

Notes de llançament per Debian 11 (bullseye), ARM EABI

El projecte de documentació de Debian (<https://www.debian.org/doc/>)

21 d'octubre de 2021

Notes de llançament per Debian 11 (bullseye), ARM EABI

Aquest document és programari lliure, el podeu redistribuir i/o modificar sota els termes de la Llicència Pública General de GNU versió 2 de la llicència, tal com la publica la Free Software Foundation.

Aquest programa es distribueix amb l'esperança que serà útil, però sense cap garantia, fins i tot sense la garantia implícita de COMERCIALITZACIÓ o IDONEÏTAT PER A UN PROPÒSIT PARTICULAR. Consulteu la llicència general pública de GNU per a més detalls.

Hauríeu d'haver rebut una còpia de la llicència pública general GNU amb este programa; si no, escriviu a la Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA.

Podeu també trobar el text amb la llicència a <https://www.gnu.org/licenses/gpl-2.0.html> i a `/usr/share/common-licenses/GPL-2` en els sistemes Debian.

Índex

1	Introducció	1
1.1	Informar d'errors d'este document	1
1.2	Col·laborar amb informes d'actualització	1
1.3	Font d'este document	2
2	Què hi ha de nou a Debian 11	3
2.1	Arquitectures suportades	3
2.2	Què hi ha de nou a la distribució?	3
2.2.1	Desktops and well known packages	3
2.2.2	Driverless scanning and printing	4
2.2.2.1	CUPS and driverless printing	4
2.2.2.2	SANE and driverless scanning	4
2.2.3	New generic open command	5
2.2.4	Control groups v2	5
2.2.5	Persistent systemd journal	5
2.2.6	New Fcitx 5 Input Method	5
2.2.7	Notícies de Debian Med Blend	5
2.2.8	Kernel support for exFAT	5
2.2.9	Improved man page translations	6
2.2.10	Improved support for alternative init systems	6
3	El sistema d'instal·lació	7
3.1	Què hi ha de nou al sistema d'instal·lació?	7
3.1.1	Help with installation of firmware	7
3.1.2	Instal·lació automatitzada	7
3.2	Container and Virtual Machine images	8
4	Upgrades from Debian 10 (buster)	9
4.1	Preparació de l'actualització	9
4.1.1	Back up any data or configuration information	9
4.1.2	Inform users in advance	9
4.1.3	Prepare for downtime on services	9
4.1.4	Prepare for recovery	10
4.1.4.1	Debug shell during boot using initrd	10
4.1.4.2	Debug shell during boot using systemd	10
4.1.5	Prepare a safe environment for the upgrade	11
4.2	Start from "pure" Debian	11
4.2.1	Upgrade to Debian 10 (buster)	11
4.2.2	Remove non-Debian packages	11
4.2.3	Upgrade to latest point release	12
4.2.4	Prepare the package database	12
4.2.5	Remove obsolete packages	12
4.2.6	Clean up leftover configuration files	12
4.2.7	The security section	12
4.2.8	The proposed-updates section	12
4.2.9	Unofficial sources	12
4.2.10	Disabling APT pinning	13
4.2.11	Check package status	13
4.3	Preparing APT source-list files	13
4.3.1	Adding APT Internet sources	14
4.3.2	Adding APT sources for a local mirror	14
4.3.3	Adding APT sources from optical media	15
4.4	Upgrading packages	15
4.4.1	Recording the session	16

4.4.2	Updating the package list	16
4.4.3	Make sure you have sufficient space for the upgrade	16
4.4.4	Minimal system upgrade	18
4.4.5	Upgrading the system	19
4.5	Possible issues during upgrade	19
4.5.1	Dist-upgrade fails with “Could not perform immediate configuration”	19
4.5.2	Expected removals	19
4.5.3	Conflicts or Pre-Depends loops	19
4.5.4	File conflicts	20
4.5.5	Configuration changes	20
4.5.6	Change of session to console	20
4.6	Upgrading your kernel and related packages	20
4.6.1	Installing a kernel metapackage	21
4.7	Preparing for the next release	21
4.7.1	Purging removed packages	21
4.8	Paquets obsolets	22
4.8.1	Transitional dummy packages	22
5	Problemes a tenir en compte a bullseye	23
5.1	Actualització d’elements específics per bullseye	23
5.1.1	The XFS file system no longer supports barrier/nobarrier option	23
5.1.2	Changed security archive layout	23
5.1.3	Password hashing uses yescrypt by default	23
5.1.4	NSS NIS and NIS+ support require new packages	24
5.1.5	Config file fragment handling in unbound	24
5.1.6	rsync parameter deprecation	24
5.1.7	Vim addons handling	24
5.1.8	OpenStack and cgroups v1	24
5.1.9	OpenStack API policy files	24
5.1.10	sendmail downtime during upgrade	25
5.1.11	FUSE 3	25
5.1.12	GnuPG options file	25
5.1.13	Linux enables user namespaces by default	25
5.1.14	Linux disables unprivileged calls to bpf() by default	25
5.1.15	redmine missing in bullseye	26
5.1.16	Exim 4.94	26
5.1.17	SCSI device probing is non-deterministic	26
5.1.18	rdiff-backup require lockstep upgrade of server and client	26
5.1.19	Intel CPU microcode issues	27
5.1.20	Upgrades involving libgc1c2 need two runs	27
5.1.21	fail2ban can’t send e-mail using mail from BSD-mailx	27
5.1.22	No new SSH connections possible during upgrade	27
5.1.23	Coses a fer després de reinicialitzar	27
5.2	Items not limited to the upgrade process	27
5.2.1	Limitacions en el suport de seguretat	27
5.2.1.1	Estat de seguretat als navegadors web i els seus motors de renderització	28
5.2.1.2	OpenJDK 17	28
5.2.1.3	Go-based packages	28
5.2.2	Accedir a la configuració de GNOME sense ratolí	28
5.2.3	The <code>rescue</code> boot option is unusable without a root password	28
5.3	Obsolescence and deprecation	29
5.3.1	Paquets notables obsolets	29
5.3.2	Components desfasats a bullseye	30
5.3.3	No-longer-supported hardware	30
5.4	Known severe bugs	31

6	Més informació sobre Debian	33
6.1	Llegir més	33
6.2	Trobar ajuda	33
6.2.1	Llistes de correu	33
6.2.2	Internet Relay Chat	33
6.3	Informes d'error	33
6.4	Col·laborar amb Debian	34
7	Glossari	35
A	Gestió del vostre sistema buster abans de l'actualització	37
A.1	Actualització del vostre sistema buster	37
A.2	Comprovació dels fitxers de llista de fonts d'APT	37
A.3	Eliminar fitxers de configuració obsolets	38
B	Contribuïdors de les notes de llançament	39
	Índex alfabètic	41

Capítol 1

Introducció

Este document informa als usuaris de la distribució Debian dels principals canvis a la versió 11 (nom en clau bullseye).

Les notes de llançament ofereixen informació sobre com actualitzar de forma segura des de la versió 10 (nom en clau buster) a la versió actual, i informen als usuaris de problemes potencials coneguts que es poden trobar durant eixe procés.

You can get the most recent version of this document from <https://www.debian.org/releases/bullseye/releasenotes>.

ATENCIÓ



Tingueu en compte que és impossible fer una llista amb tots els problemes coneguts i per tant ha calgut fer una selecció basada en una combinació de l'abast i l'impacte dels problemes.

Tingueu en compte també, que tan sols es suporta i documenta l'actualització des de la versió anterior de Debian (en este cas, l'actualització des de 10). Si necessiteu actualitzar des de versions més antigues, us suggerim que llegiu primer les edicions anteriors de les notes de llançament i actualitzeu primer a 10.

1.1 Informar d'errors d'este document

S'ha intentat fer comprovacions dels diversos passos del procés d'actualització que es descriuen en este document, intentant també anticipar possibles problemes que els nostres usuaris puguen trobar.

No obstant això, si creieu que heu trobat un error en aquesta documentació (informació incorrecta o manca d'informació), envieu un informe d'error al [sistema de seguiment d'errors](https://bugs.debian.org/) (<https://bugs.debian.org/>), al paquet `release-notes`. Reviseu primer si hi ha cap [informe d'error existent](https://bugs.debian.org/release-notes) (<https://bugs.debian.org/release-notes>) en cas de que ja s'haja informat del problema. No dubteu a afegir qualsevol informació adicional als informes d'error ja existents si creieu que podeu contribuir contingut a aquest document.

Apreciem, i animem, a proporcionar informes amb fitxers de diferències de les fonts del document. Trobareu més informació que descriu com obtenir les fonts d'aquest document en Secció [1.3](#).

1.2 Col·laborar amb informes d'actualització

Qualsevol informació dels usuaris relacionada amb actualitzacions des de buster a bullseye és benvinguda. Si voleu compartir alguna informació, envieu un informe d'error amb els vostres resultats al [sistema de seguiment d'errors](https://bugs.debian.org/) (<https://bugs.debian.org/>), al paquet `upgrade-reports`. Us demanem que comprimiu qualsevol adjunt que envieu (fent ús del `gzip`).

Quan envieu l'informe d'actualització, assegureu-vos d'incloure la informació següent:

- L'estat de la vostra base de dades de paquets abans de l'actualització: La base de dades d'estat de **dpkg** que trobareu a `/var/lib/dpkg/status` i la informació d'estat d' `apt` que trobareu a `/var/lib/apt/extended_states`. Hauríeu d'haver fet un còpia de seguretat abans d'actualitzar com es descriu a Secció 4.1.1, però també podeu trobar còpies de seguretat de `/var/lib/dpkg/status` a `/var/backups`.
- Els registres de sessió creats amb **script**, tal i com es descriu a Secció 4.4.1 .
- Els registres d'`apt` els podeu trobar a `/var/log/apt/term.log`, i els registres d'**aptitude** els trobareu a `/var/log/aptitude`.

NOTA

Hauríeu de dedicar un cert temps a revisar i esborrar qualsevol informació sensible i/o confidencial als fitxers de registre abans d'incloure'ls a l'informe d'error, ja que la informació es publicarà a una base de dades pública.

1.3 Font d'este document

El codi font d'este document està en format DocBook XML. La versió HTML es genera utilitzant `docbook-xsl` i `xsltproc`. La versió en PDF es genera utilitzant `dblatex` o bé `xmlroff`. El codi font de les notes de llançament està en el dipòsit Git del *Projecte de documentació de Debian*. Podeu utilitzar la **interfície web** (<https://salsa.debian.org/ddp-team/release-notes/>) per accedir individualment als seus fitxers des de la web i veure els canvis. Per a més informació de com accedir a Git, consulteu les **pàgines d'informació de VCS del projecte de documentació de Debian** (<https://www.debian.org/doc/vcs>).

Capítol 2

Què hi ha de nou a Debian 11

Al [Viqui](https://wiki.debian.org/NewInBullseye) (<https://wiki.debian.org/NewInBullseye>) trobareu més informació d'este tema.

2.1 Arquitectures suportades

Les arquitectures suportades oficialment a Debian bullseye són les següents:

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

Podeu llegir més de l'estat dels ports, i informació específica del port de la vostra arquitectura a les [pàgines web dels ports de Debian](https://www.debian.org/ports/) (<https://www.debian.org/ports/>).

2.2 Què hi ha de nou a la distribució?

Este nou llançament de Debian ve altra vegada amb molt més programari que el seu predecessor buster, la distribució inclou més de 11294 nous paquets, d'un total de 59551 paquets. S'ha actualitzat la major part del programari de la distribució: més de 42821 paquets de programes (açò suposa el 72% de tots els paquets de buster). També, un nombre significatiu de paquets (uns 9519, un 16% dels paquets de buster) s'han esborrat de la distribució per motius diversos. No voreu cap actualització per aquests paquets i es marcaran com a «obsolete» als entorns de gestió de paquets.

2.2.1 Desktops and well known packages

Debian again ships with several desktop applications and environments. Among others it now includes the desktop environments GNOME 3.38, KDE Plasma 5.20, LXDE 11, LXQt 0.16, MATE 1.24, and Xfce 4.16.

Les aplicacions de productivitat també han sigut actualitzades, incloent la suite d'oficina:

- LibreOffice is upgraded to version 7.0;
- Calligra is upgraded to 3.2.
- GNUcash is upgraded to 4.4;

Entre altres, este llançament també inclou les actualitzacions de programari següents:

Paquet	Versió a 10 (buster)	Versió a 11 (bullseye)
Apache	2.4.38	2.4.48
BIND servidor de DNS	9.11	9.16
Cryptsetup	2.1	2.3
Dovecot MTA	2.3.4	2.3.13
Emacs	26.1	27.1
Exim servidor de correu per defecte	4.92	4.94
GNU Compiler Collection com a compilador per defecte	8.3	10.2
GIMP	2.10.8	2.10.22
GnuPG	2.2.12	2.2.27
Inkscape	0.92.4	1.0.2
la biblioteca C de GNU	2.28	2.31
lighttpd	1.4.53	1.4.59
Imatge del nucli Linux	sèrie 4.19	5.10 series
LLVM/Clang toolchain	6.0.1 i 7.0.1 (per defecte)	9.0.1 and 11.0.1 (default)
MariaDB	10.3	10.5
Nginx	1.14	1.18
OpenJDK	11	11
OpenSSH	7.4p1	8.4p1
Perl	5.28	5.32
PHP	7.3	7.4
Postfix MTA	3.4	3.5
PostgreSQL	11	13
Python 3	3.7.3	3.9.1
Rustc	1.41 (1.34 for armel)	1.48
Samba	4.9	4.13
Vim	8.1	8.2

2.2.2 Driverless scanning and printing

Both printing with CUPS and scanning with SANE are increasingly likely to be possible without the need for any driver (often non-free) specific to the model of the hardware, especially in the case of devices marketed in the past five years or so.

2.2.2.1 CUPS and driverless printing

Modern printers connected by ethernet or wireless can already use **driverless printing** (<https://wiki.debian.org/CUPSQuickPrintQueues>), implemented via CUPS and cups-filters, as was described in the **Release Notes for buster** (<https://www.debian.org/releases/buster/amd64/release-notes/ch-whats-new.html#driverless-printing>). Debian 11 “bullseye” brings the new package **ipp-usb**, which is recommended by cups-daemon and uses the vendor-neutral **IPP-over-USB** (<https://wiki.debian.org/CUPSDriverlessPrinting#ippoverusb>) protocol supported by many modern printers. This allows a USB device to be treated as a network device, extending driverless printing to include USB-connected printers. The specifics are outlined **on the wiki** (<https://wiki.debian.org/CUPSDriverlessPrinting#ipp-usb>).

The systemd service file included in the **ipp-usb** package starts the **ipp-usb** daemon when a USB-connected printer is plugged in, thus making it available to print to. By default cups-browsed should configure it automatically, or it can be **manually set up with a local driverless print queue** (<https://wiki.debian.org/SystemPrinting>).

2.2.2.2 SANE and driverless scanning

The official SANE driverless backend is provided by **sane-escl** in **libsane1**. An independently developed driverless backend is **sane-airscan**. Both backends understand the **eSCL protocol** (<https://wiki.debian.org/SaneOverNetwork#escl>) but **sane-airscan** can also use the **WSD** (<https://wiki.debian.org/SaneOverNetwork#wsd>) protocol. Users should consider having both backends on their systems.

eSCL and WSD are network protocols. Consequently they will operate over a USB connection if the device is an `IPP-over-USB` device (see above). Note that `libsane1` has `ipp-usb` as a recommended package. This leads to a suitable device being automatically set up to use a driverless backend driver when it is connected to a USB port.

2.2.3 New generic open command

A new **open** command is available as a convenience alias to **xdg-open** (by default) or **run-mailcap**, managed by the **update-alternatives(1)** (<https://manpages.debian.org//bullseye/dpkg/update-alternatives.1.html>) system. It is intended for interactive use at the command line, to open files with their default application, which can be a graphical program when available.

2.2.4 Control groups v2

In bullseye, `systemd` defaults to using control groups v2 (`cgroupv2`), which provides a unified resource-control hierarchy. Kernel commandline parameters are available to re-enable the legacy `cgroups` if necessary; see the notes for OpenStack in Secció 5.1.8 section.

2.2.5 Persistent systemd journal

`Systemd` in bullseye activates its persistent journal functionality by default, storing its files in `/var/log/journal/`. See **systemd-journald.service(8)** (<https://manpages.debian.org//bullseye/systemd/systemd-journald.service.8.html>) for details; note that on Debian the journal is readable for members of `adm`, in addition to the default `systemd-journal` group.

This should not interfere with any existing traditional logging daemon such as `rsyslog`, but users who are not relying on special features of such a daemon may wish to uninstall it and switch over to using only the journal.

2.2.6 New Fcix 5 Input Method

`Fcix 5` is an input method for Chinese, Japanese, Korean and many other languages. It is the successor of the popular `Fcix 4` in buster. The new version supports `Wayland` and has better add-on support. More information including the migration guide can be found **on the wiki** (<https://wiki.debian.org/I18n/Fcix5>).

2.2.7 Notícies de Debian Med Blend

The Debian Med team has been taking part in the fight against `COVID-19` by packaging software for researching the virus on the sequence level and for fighting the pandemic with the tools used in epidemiology. The effort will be continued in the next release cycle with focus on machine learning tools that are used in both fields.

Besides the addition of new packages in the field of life sciences and medicine, more and more existing packages have gained `Continuous Integration` support.

A range of performance critical applications now benefit from **SIMD Everywhere** (<https://wiki.debian.org/SIMDEverywhere>). This library allows packages to be available on more hardware platforms supported by Debian (notably on `arm64`) while maintaining the performance benefit brought by processors supporting vector extensions, such as `AVX` on `amd64`, or `NEON` on `arm64`.

To install packages maintained by the Debian Med team, install the metapackages named `med-*`, which are at version 3.6.x for Debian bullseye. Feel free to visit the **Debian Med tasks pages** (<https://blends.debian.org/med/tasks>) to see the full range of biological and medical software available in Debian.

2.2.8 Kernel support for exFAT

bullseye is the first release providing a Linux kernel which has support for the `exFAT` filesystem, and defaults to using it for mounting `exFAT` filesystems. Consequently it's no longer required to use the `filesystem-in-userspace` implementation provided via the `exfat-fuse` package. If you would like to

continue to use the `filesystem-in-userspace` implementation, you need to invoke the `mount.exfat-fuse` helper directly when mounting an exFAT filesystem.

Tools for creating and checking an exFAT filesystem are provided in the `exfatprogs` package by the authors of the Linux kernel exFAT implementation. The independent implementation of those tools provided via the existing `exfat-utils` package is still available, but cannot be co-installed with the new implementation. It's recommended to migrate to the `exfatprogs` package, though you must take care of command options, which are most likely incompatible.

2.2.9 Improved man page translations

The manual pages for several projects such as `systemd`, `util-linux`, `OpenSSH`, and `Mutt` in a number of languages, including French, Spanish, and Macedonian, have been substantially improved. To benefit from this, please install `manpages-xx` (where `xx` is the code for your preferred natural language).

During the lifetime of the bullseye release, backports of further translation improvements will be provided via the `backports` archive.

2.2.10 Improved support for alternative init systems

The default init system in Debian is `systemd`. In bullseye, a number of alternative init systems are supported (such as System-V-style `init` and `OpenRC`), and most desktop environments now work well on systems running alternative inits. Details on how to switch init system (and where to get help with issues related to running inits other than `systemd`) are available [on the Debian wiki](https://wiki.debian.org/Init) (<https://wiki.debian.org/Init>).

Capítol 3

El sistema d'instal·lació

El sistema d'instal·lació oficial a Debian és el Debian Installer. Ens proporciona diversos mètodes d'instal·lació. Els mètodes dels que disposeu per instal·lar el vostre sistema dependran de la vostra arquitectura.

Les imatges de l'instal·lador de bullseye es poden trobar junt a la guia d'instal·lació a la [web de Debian](https://www.debian.org/releases/bullseye/debian-installer/) (<https://www.debian.org/releases/bullseye/debian-installer/>).

La guia d'instal·lació també està inclosa al primer suport dels conjunts de DVD de Debian (CD/blu-ray), a:

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

Pot ser també voldreu comprovar les [errates](https://www.debian.org/releases/bullseye/debian-installer/index#errata) (<https://www.debian.org/releases/bullseye/debian-installer/index#errata>) del debian-installer per obtenir la llista de problemes coneguts.

3.1 Què hi ha de nou al sistema d'instal·lació?

S'ha fet molt desenvolupament a l'instal·lador de Debian des de el seu anterior llançament oficial a Debian 10 amb el resultat que s'han aconseguit millores tant al suport de maquinari com noves i excitants característiques.

Si teniu interès en un visió general dels canvis detallats produïts des de buster, proveu els anuncis de llançament de la beta de bullseye i els llançaments RC que trobareu a [l'historial de notícies](https://www.debian.org/devel/debian-installer/News/) (<https://www.debian.org/devel/debian-installer/News/>) de l'instal·lador de Debian.

3.1.1 Help with installation of firmware

More and more, peripheral devices require firmware to be loaded as part of the hardware initialization. To help deal with this problem, the installer has a new feature. If some of the installed hardware requires firmware files to be installed, the installer will try to add them to the system, based on a mapping from hardware ID to firmware file names.

This new functionality is restricted to the unofficial installer images with firmware included (see https://www.debian.org/releases/bullseye/debian-installer/#firmware_nonfree (https://www.debian.org/releases/bullseye/debian-installer/#firmware_nonfree)). The firmware is usually not DFSG compliant, so it is not possible to distribute it in Debian's main repository.

If you experience problems related to (missing) firmware, please read [the dedicated chapter of the installation-guide](https://www.debian.org/releases/bullseye/amd64/ch06s04#completing-installation) (<https://www.debian.org/releases/bullseye/amd64/ch06s04#completing-installation>).

3.1.2 Instal·lació automatitzada

Some changes 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 buster installer, you cannot expect these to work with the new installer without modification.

La [guia d'instal·lació](https://www.debian.org/releases/bullseye/installmanual) (<https://www.debian.org/releases/bullseye/installmanual>) té un apèndix separat amb una extensa documentació de com utilitzar la configuració prèvia.

3.2 Container and Virtual Machine images

Multi-architecture Debian bullseye container images are available on [Docker Hub](https://hub.docker.com/_/debian) (https://hub.docker.com/_/debian). In addition to the standard images, a “slim” variant is available that reduces disk usage.

Virtual machine images for the Hashicorp Vagrant VM manager are published to [Vagrant Cloud](https://app.vagrantup.com/debian) (<https://app.vagrantup.com/debian>).

Capítol 4

Upgrades from Debian 10 (buster)

4.1 Preparació de l'actualització

We suggest that before upgrading you also read the information in [Capítol 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 ("dotfiles") 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 bullseye, 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 Secció 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 Secció 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 bullseye 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/bullseye/installmanual) (<https://www.debian.org/releases/bullseye/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 or [live install](https://www.debian.org/CD/live/) (<https://www.debian.org/CD/live/>) image. 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.

¹If 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.

²For 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.

³This 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](https://freedesktop.org/wiki/Software/systemd/Debugging/) (<https://freedesktop.org/wiki/Software/systemd/Debugging/>) article.

4.1.5 Prepare a safe environment for the upgrade

IMPORTANT



If you are using some VPN services (such as `tinc`) consider that they might not be available throughout the upgrade process. Please see Secció [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 temporarily fails.

Users of the watchdog daemon provided by the `micro-evtd` package should stop the daemon and disable the watchdog timer before the upgrade, to avoid a spurious reboot in the middle of the upgrade process:

```
# service micro-evtd stop
# /usr/sbin/microapl -a system_set_watchdog off
```

4.2 Start from “pure” Debian

The upgrade process described in this chapter has been designed for “pure” Debian stable systems. APT controls what is installed on your system. If your APT configuration mentions additional sources besides `buster`, or if you have installed packages from other releases or from third parties, then to ensure a reliable upgrade process you may wish to begin by removing these complicating factors.

The main configuration file that APT uses to decide what sources it should download packages from is `/etc/apt/sources.list`, but it can also use files in the `/etc/apt/sources.list.d/` directory - for details see [sources.list\(5\)](https://manpages.debian.org//bullseye/apt/sources.list.5.html) (<https://manpages.debian.org//bullseye/apt/sources.list.5.html>). If your system is using multiple source-list files then you will need to ensure they stay consistent.

4.2.1 Upgrade to Debian 10 (buster)

Direct upgrades from Debian releases older than 10 (`buster`) are not supported. Display your Debian version with:

```
$ cat /etc/debian_version
```

Please follow the instructions in the [Release Notes for Debian 10](https://www.debian.org/releases/buster/releasenotes) (<https://www.debian.org/releases/buster/releasenotes>) to upgrade to Debian 10 first.

4.2.2 Remove non-Debian packages

Below there are two methods for finding installed packages that did not come from Debian, 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 '?narrow(?installed, ?not(?origin(Debian)))'
$ apt-forktracer | sort
```

4.2.3 Upgrade to latest point release

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

4.2.4 Prepare the package database

You should make sure the package database is ready before proceeding with the upgrade. If you are a user of another package manager like `aptitude` or `synaptic`, review any pending actions. A package scheduled for installation or removal might interfere with the upgrade procedure. Note that correcting this is only possible if your APT source-list files still point to *buster* and not to *stable* or *bullseye*; see Secció [A.2](#).

4.2.5 Remove obsolete packages

It is a good idea to **remove obsolete packages** from your system before upgrading. They may introduce complications during the upgrade process, and can present security risks as they are no longer maintained.

4.2.6 Clean up leftover configuration files

A previous upgrade may have left unused copies of configuration files; **old versions** of configuration files, versions supplied by the package maintainers, etc. Removing leftover files from previous upgrades can avoid confusion. Find such leftover files with:

```
# find /etc -name '*.dpkg-*' -o -name '*.ucf-*' -o -name '*.merge-error'
```

4.2.7 The security section

For APT source lines referencing the security archive, the format has changed slightly along with the release name, going from `buster/updates` to `bullseye-security`; see Secció [5.1.2](#).

4.2.8 The proposed-updates section

If you have listed the `proposed-updates` section in your APT source-list files, you should remove it before attempting to upgrade your system. This is a precaution to reduce the likelihood of conflicts.

4.2.9 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 APT source-list files, you should check if that archive also offers packages compiled for *bullseye* and change the source item accordingly at the same time as your source items for Debian packages.

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

⁴Debian’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.

4.2.10 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\(5\)](https://manpages.debian.org//bullseye/apt/apt_preferences.5.en.html) (https://manpages.debian.org//bullseye/apt/apt_preferences.5.en.html).

4.2.11 Check package 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
```

```
o
```

```
# dpkg --get-selections "*" > ~/pkts-actuals.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** 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**, 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** can be changed using:

```
# echo nom_de_paquet 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 APT source-list files still refer to buster as explained in Secció [A.2](#).

4.3 Preparing APT source-list files

Before starting the upgrade you must reconfigure APT source-list files (`/etc/apt/sources.list` and files under `/etc/apt/sources.list.d/`) to add sources for `bullseye` and typically to remove sources for `buster`.

APT will consider all packages that can be found via any configured archive, and install the package with the highest version number, giving priority to the first entry in the files. Thus, if you have multiple mirror locations, list first the ones on local hard disks, then CD-ROMs, and then remote mirrors.

A release can often be referred to both by its codename (e.g. `buster`, `bullseye`) 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 bullseye changes from e.g. testing to stable.
- 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

On new installations the default is for APT to be set up to use the Debian APT CDN service, which should ensure that packages are automatically downloaded from a server near you in network terms. As this is a relatively new service, older installations may have configuration that still points to one of the main Debian Internet servers or one of the mirrors. If you haven't done so yet, it is recommended to switch over to the use of the CDN service in your APT configuration.

To make use of the CDN service, add a line like this to your APT source configuration (assuming you are using `main` and `contrib`):

```
deb http://deb.debian.org/debian bullseye main contrib
```

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

However, if you get better results using a specific mirror that is close to you in network terms, this option is still available.

Debian mirror addresses can be found at <https://www.debian.org/distrib/ftplist> (look at the “list of Debian mirrors” section).

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/bullseye/main/binary-armel/...
http://mirrors.kernel.org/debian/dists/bullseye/contrib/binary-armel/...
```

To configure APT to use a given mirror, add a line like this (again, assuming you are using `main` and `contrib`):

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

Again, after adding your new sources, disable the previously existing archive entries.

4.3.2 Adding APT sources for a local mirror

Instead of using remote package mirrors, you may wish to modify the APT source-list files 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/bullseye/main/binary-armel/...
/var/local/debian/dists/bullseye/contrib/binary-armel/...
```

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

```
deb file:/var/local/debian bullseye 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 archive entries in the APT source-list files by placing a hash sign (#) in front of them.

4.3.3 Adding APT sources from optical media

If you want to use *only* DVDs (or CDs or Blu-ray Discs), comment out the existing entries in all the APT source-list files 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`.

NOTA



`apt` is meant for interactive use, and should not be used in scripts. In scripts one should use `apt-get`, which has a stable output better suitable for parsing.

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

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

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

NOTA



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-bullseyestep.time -a ~/upgrade-bullseyestep.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-bullseye.scri` to view the file.

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

apt will also log the changed package states in `/var/log/apt/history.log` and the terminal output in `/var/log/apt/term.log`. **dpkg** will, in addition, log all package state changes in `/var/log/dpkg.log`. If you use **aptitude**, it will also log state changes in `/var/log/aptitude`.

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

```
# scriptreplay ~/upgrade-bullseyestep.time ~/upgrade-bullseyestep.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 update
```

NOTA



Users of **apt-secure** may find issues when using **aptitude** or **apt-get**. For **apt-get**, you can use **apt-get update --allow-releaseinfo-change**.

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 Secció 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 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 -o APT::Get::Trivial-Only=true full-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.

NOTA



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 Secció 4.4.4 before running this command to estimate the disk space.

If you do not have enough space for the upgrade, **apt** 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 clean** will remove all previously downloaded package files.
- Remove forgotten packages. If you have used **aptitude** or **apt** to manually install packages in buster 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 autoremove
```

You can also use **deborphan**, **debfoster**, 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-goodies` 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, ...).

NOTA



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 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 /var/cache/apt/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 Secció 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 APT source-list files back to buster as described in Secció A.2.

4.4.4 Minimal system upgrade

IMPORTANT



If you are upgrading remotely, be aware of Secció 5.1.22.

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 Secció 4.4.5.

To do this, first run:

```
# apt upgrade --without-new-pkgs
```

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 after downloading the packages. 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 full-upgrade
```

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 install package`.

4.5 Possible issues during upgrade

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

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

In some cases the **apt full-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 full-upgrade -o APT::Immediate-Configure=0** instead should allow the upgrade to proceed.

Another possible workaround for this problem is to temporarily add both buster and bullseye sources to your APT source-list files and run **apt update**.

4.5.2 Expected removals

The upgrade process to bullseye 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 bullseye, see Secció [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** 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** 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** or

```
# dpkg --remove nom_del_paquet
```

to eliminate some of the offending packages, or

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

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

```
# dpkg --install /camí/a/nom_del_paquet.deb
```

4.5.4 File conflicts

File conflicts should not occur if you upgrade from a “pure” buster 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` 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 systems with a graphical interface 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 full-upgrade from buster to bullseye, 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 “4.9.0-8-amd64”, it is recommended that you install `linux-image-amd64`. You may also use `apt` to see a long description of each package in order to help choose the best one available. For example:

```
# apt show linux-image-amd64
```

You should then use `apt 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 Secció 5.1.23 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](https://kernel-team.pages.debian.net/kernel-handbook/) (<https://kernel-team.pages.debian.net/kernel-handbook/>), 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 `full-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 Secció 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 Secció 4.4.3 and Secció 4.8. You should review which configuration files they use and consider purging the packages to remove their configuration files. See also Secció 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 buster) or they were provided by third-party vendors. In particular, old `init.d` scripts have been known to cause issues.

ATENCIÓ



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 purge**. Assuming you want to purge all of them in one go, you can use the following command:

```
# apt 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 Paquets obsolets

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

There are many reasons why packages might have been removed from the distribution: they are no longer maintained upstream; there is no longer a Debian Developer interested in maintaining the packages; the functionality they provide has been superseded by different software (or a new version); or they are no longer considered suitable for bullseye due to bugs in them. In the latter case, packages might still be present in the “unstable” distribution.

Some package management front-ends provide easy ways of finding installed packages that are no longer available from any known repository. The **aptitude** textual user interface lists them in the category “Obsolete and Locally Created Packages”, and they can be listed and purged from the commandline with:

```
# aptitude search '~o'
# aptitude purge '~o'
```

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 Bullseye, please refer to Secció **5.3.1**.

4.8.1 Transitional dummy packages

Some packages from buster may have been replaced in bullseye by transitional dummy packages, which are empty placeholders designed to simplify upgrades. If for instance an application that was formerly a single package has been split into several, a transitional package may be provided with the same name as the old package and with appropriate dependencies to cause the new ones to be installed. After this has happened the redundant dummy package can be safely removed.

The package descriptions for transitional dummy packages usually indicate their purpose. However, they are not uniform; in particular, some “dummy” packages are designed to be kept installed, in order to pull in a full software suite, or track the current latest version of some program. You might also find **deborphan** with the `--guess-*` options (e.g. `--guess-dummy`) useful to detect transitional dummy packages on your system.

⁵Or for as long as there is not another release in that time frame. Typically only two stable releases are supported at any given time.

Capítol 5

Problemes a tenir en compte a bullseye

De vegades, els canvis introduïts en un nou llançament poden tenir efectes col·laterals que no podem evitar raonablement, o exposen errors en un altre lloc. Aquesta secció documenta els problemes dels tenim coneixement. Llegiu també l'errata, la documentació dels paquets, els informes d'error i altra informació que s'esmenta a Secció [6.1](#).

5.1 Actualització d'elements específics per bullseye

Aquesta secció inclou elements relacionats amb l'actualització de buster a bullseye.

5.1.1 The XFS file system no longer supports barrier/nobarrier option

Support for the `barrier` and `nobarrier` mount options has been removed from the XFS file system. It is recommended to check `/etc/fstab` for the presence of either keyword and remove it. Partitions using these options will fail to mount.

5.1.2 Changed security archive layout

For bullseye, the security suite is now named `bullseye-security` instead of `codename/updates` and users should adapt their APT source-list files accordingly when upgrading.

The security line in your APT configuration may look like:

```
deb https://deb.debian.org/debian-security bullseye-security main contrib
```

If your APT configuration also involves pinning or `APT::Default-Release`, it is likely to require adjustments as the codename of the security archive no longer matches that of the regular archive. An example of a working `APT::Default-Release` line for bullseye looks like:

```
APT::Default-Release "/^bullseye(|-security|-updates)$/";
```

which takes advantage of the undocumented feature of APT that it supports regular expressions (inside `/`).

5.1.3 Password hashing uses yescrypt by default

The default password hash for local system accounts **has been changed** (<https://tracker.debian.org/news/1226655/accepted-pam-140-3-source-into-unstable/>) from SHA-512 to **yescrypt** (<https://www.openwall.com/yescrypt/>) (see **crypt(5)** (<https://manpages.debian.org//bullseye/libcrypt-dev/crypt.5.html>)). This is expected to provide improved security against dictionary-based password guessing attacks, in terms of both the space and time complexity of the attack.

To take advantage of this improved security, change local passwords; for example use the **passwd** command.

Old passwords will continue to work using whatever password hash was used to create them.

Yescrypt is not supported by Debian 10 (buster). As a result, shadow password files (`/etc/shadow`) cannot be copied from a bullseye system back to a buster system. If these files are copied, passwords that have been changed on the bullseye system will not work on the buster system. Similarly, password hashes cannot be cut&pasted from a bullseye to a buster system.

If compatibility is required for password hashes between bullseye and buster, modify `/etc/pam.d/common-password`. Find the line that looks like:

```
password [success=1 default=ignore] pam_unix.so obscure yescrypt
```

and replace `yescrypt` with `sha512`.

5.1.4 NSS NIS and NIS+ support require new packages

NSS NIS and NIS+ support has been moved to separate packages called `libnss-nis` and `libnss-nisplus`. Unfortunately, `glibc` can't depend on those packages, so they are now only recommended.

On systems using NIS or NIS+, it is therefore recommended to check that those packages are correctly installed after the upgrade.

5.1.5 Config file fragment handling in unbound

The DNS resolver `unbound` has changed the way it handles configuration file fragments. If you are relying on an `include:` directive to merge several fragments into a valid configuration, you should read [the NEWS file](https://sources.debian.org/src/unbound/bullseye/debian/NEWS/) (<https://sources.debian.org/src/unbound/bullseye/debian/NEWS/>).

5.1.6 rsync parameter deprecation

The `rsync` parameters `--copy-devices` and `--noatime` have been renamed to `--write-devices` and `--open-noatime`. The old forms are no longer supported; if you are using them you should see [the NEWS file](https://sources.debian.org/src/rsync/bullseye/debian/rsync.NEWS/) (<https://sources.debian.org/src/rsync/bullseye/debian/rsync.NEWS/>). Transfer processes between systems running different Debian releases may require the buster side to be upgraded to a version of `rsync` from the [backports](https://backports.debian.org/) (<https://backports.debian.org/>) repository.

5.1.7 Vim addons handling

The addons for `vim` historically provided by `vim-scripts` are now managed by Vim's native "package" functionality rather than by `vim-addon-manager`. Vim users should prepare before upgrading by following the instructions in [the NEWS file](https://sources.debian.org/src/vim-scripts/bullseye/debian/NEWS/) (<https://sources.debian.org/src/vim-scripts/bullseye/debian/NEWS/>).

5.1.8 OpenStack and cgroups v1

OpenStack Victoria (released in bullseye) requires `cgroup v1` for block device QoS. Since bullseye also changes to using `cgroupv2` by default (see Secció 2.2.4), the `sysfs` tree in `/sys/fs/cgroup` will not include `cgroup v1` features such as `/sys/fs/cgroup/blkio`, and as a result `cgcreate -g blkio:foo` will fail. For OpenStack nodes running `nova-compute` or `cinder-volume`, it is strongly advised to add the parameters `systemd.unified_cgroup_hierarchy=false` and `systemd.legacy_systemd_cgroup_controller` to the kernel command line in order to override the default and restore the old `cgroup` hierarchy.

5.1.9 OpenStack API policy files

Following upstream's recommendations, OpenStack Victoria as released in bullseye switches the OpenStack API to use the new YAML format. As a result, most OpenStack services, including Nova, Glance, and Keystone, appear broken with all of the API policies written explicitly in the `policy.json` files. Therefore, packages now come with a folder `/etc/PROJECT/policy.d` containing a file `00_default_policy.yaml`, with all of the policies commented out by default.

To avoid the old `policy.json` file staying active, the Debian OpenStack packages now rename that file as `disabled.policy.json.old`. In some cases where nothing better could be done in time for the release the `policy.json` is even simply deleted. So before upgrading, it is strongly advised to back up the `policy.json` files of your deployments.

More details are available in the [upstream documentation](https://governance.openstack.org/tc/goals/selected/wallaby/migrate-policy-format-from-json-to-yaml.html) (<https://governance.openstack.org/tc/goals/selected/wallaby/migrate-policy-format-from-json-to-yaml.html>).

5.1.10 sendmail downtime during upgrade

In contrast to normal upgrades of `sendmail`, during the upgrade of buster to bullseye the `sendmail` service will be stopped, causing more downtime than usual. For generic advice on reducing downtime see Secció [4.1.3](#).

5.1.11 FUSE 3

Some packages including `gvfs-fuse`, `kio-fuse`, and `sshfs` have switched to FUSE 3. During upgrades, this will cause `fuse3` to be installed and `fuse` to be removed.

In some exceptional circumstances, e.g., when performing the upgrade by only running `apt-get dist-upgrade` instead of the recommended upgrade steps from Capítol [4](#), packages depending on `fuse3` might be kept back during upgrades. Running the steps discussed in Secció [4.4.5](#) again with bullseye's `apt` or upgrading them manually will resolve the situation.

5.1.12 GnuPG options file

Starting with version 2.2.27-1, per-user configuration of the GnuPG suite has completely moved to `~/.gnupg/gpg.conf`, and `~/.gnupg/options` is no longer in use. Please rename the file if necessary, or move its contents to the new location.

5.1.13 Linux enables user namespaces by default

From Linux 5.10, all users are allowed to create user namespaces by default. This will allow programs such as web browsers and container managers to create more restricted sandboxes for untrusted or less-trusted code, without the need to run as root or to use a `setuid-root` helper.

The previous Debian default was to restrict this feature to processes running as root, because it exposed more security issues in the kernel. However, as the implementation of this feature has matured, we are now confident that the risk of enabling it is outweighed by the security benefits it provides.

If you prefer to keep this feature restricted, set the `sysctl`:

```
user.max_user_namespaces = 0
```

Note that various desktop and container features will not work with this restriction in place, including web browsers, WebKitGTK, Flatpak and GNOME thumbnailing.

The Debian-specific `sysctl` `kernel.unprivileged_usersns_clone=0` has a similar effect, but is deprecated.

5.1.14 Linux disables unprivileged calls to bpf() by default

From Linux 5.10, Debian disables unprivileged calls to `bpf()` by default. However, an admin can still change this setting later on, if needed, by writing 0 or 1 to the `kernel.unprivileged_bpf_disabled` `sysctl`.

If you prefer to keep unprivileged calls to `bpf()` enabled, set the `sysctl`:

```
kernel.unprivileged_bpf_disabled = 0
```

For background on the change as default in Debian see [bug 990411](https://bugs.debian.org/990411) (<https://bugs.debian.org/990411>) for the change request.

5.1.15 redmine missing in bullseye

The package `redmine` is not provided in bullseye, as it was too late migrating over from the old version of `rails` which is at the end of upstream support (receiving fixes for severe security bugs only) to the version which is in bullseye. The Ruby Extras Maintainers are following upstream closely and will be releasing a version via [backports](https://backports.debian.org/) (<https://backports.debian.org/>) as soon as it is released and they have working packages. If you can't wait for this to happen before upgrading, you can use a VM or container running buster to isolate this specific application.

5.1.16 Exim 4.94

Please consider the version of Exim in bullseye a *major* Exim upgrade. It introduces the concept of tainted data read from untrusted sources, like e.g. message sender or recipient. This tainted data (e.g. `$local_part` or `$domain`) cannot be used among other things as a file or directory name or command name.

This *will break* configurations which are not updated accordingly. Old Debian Exim configuration files also will not work unmodified; the new configuration needs to be installed with local modifications merged in.

Typical nonworking examples include:

- Delivery to `/var/mail/$local_part`. Use `$local_part_data` in combination with `check_local_user`.
- Using

```
data = ${lookup{$local_part}lsearch{/some/path/$domain/aliases}}
```

instead of

```
data = ${lookup{$local_part}lsearch{/some/path/$domain_data/aliases}}
```

for a virtual domain alias file.

The basic strategy for dealing with this change is to use the result of a lookup in further processing instead of the original (remote provided) value.

To ease upgrading there is a new main configuration option to temporarily downgrade taint errors to warnings, letting the old configuration work with the newer Exim. To make use of this feature add

```
.ifdef _OPT_MAIN_ALLOW_INSECURE_TAINTED_DATA
allow_insecure_tainted_data = yes
.endif
```

to the Exim configuration (e.g. to `/etc/exim4/exim4.conf.localmacros`) *before* upgrading and check the logfile for taint warnings. This is a temporary workaround which is already marked for removal on introduction.

5.1.17 SCSI device probing is non-deterministic

Due to changes in the Linux kernel, the probing of SCSI devices is no longer deterministic. This could be an issue for installations that rely on the disk probing order. Two possible alternatives using links in `/dev/disk/by-path` or a `udev` rule are suggested in [this mailing list post](https://lore.kernel.org/lkml/59eedd28-25d4-7899-7c3c-89fe7fdd4b43@acm.org/) (<https://lore.kernel.org/lkml/59eedd28-25d4-7899-7c3c-89fe7fdd4b43@acm.org/>).

5.1.18 rdiff-backup require lockstep upgrade of server and client

The network protocol of versions 1 and 2 of `rdiff-backup` are incompatible. This means that you must be running the same version (either 1 or 2) of `rdiff-backup` locally and remotely. Since buster ships version 1.2.8 and bullseye ships version 2.0.5, upgrading only the local system or only the remote system from buster to bullseye will break `rdiff-backup` runs between the two.

Version 2.0.5 of `rdiff-backup` is available in the buster-backports archive, see [backports](https://backports.debian.org/) (<https://backports.debian.org/>). This enables users to first upgrade only the `rdiff-backup` package on their buster systems, and then independently upgrade systems to bullseye at their convenience.

5.1.19 Intel CPU microcode issues

The `intel-microcode` package currently in bullseye and buster-security (see [DSA-4934-1](https://www.debian.org/security/2021/dsa-4934) (<https://www.debian.org/security/2021/dsa-4934>)) is known to contain two significant bugs. For some CoffeeLake CPUs this update [may break network interfaces](https://github.com/intel/Intel-Linux-Processor-Microcode-Data-Files/issues/56) (<https://github.com/intel/Intel-Linux-Processor-Microcode-Data-Files/issues/56>) that use firmware-iwlwifi, and for some Skylake R0/D0 CPUs on systems using a very outdated firmware/BIOS, [the system may hang on boot](https://github.com/intel/Intel-Linux-Processor-Microcode-Data-Files/issues/31) (<https://github.com/intel/Intel-Linux-Processor-Microcode-Data-Files/issues/31>).

If you held back the update from DSA-4934-1 due to either of these issues, or do not have the security archive enabled, be aware that upgrading to the `intel-microcode` package in bullseye may cause your system to hang on boot or break iwlwifi. In that case, you can recover by disabling microcode loading on boot; see the instructions in the DSA, which are also in the `intel-microcode` `README.Debian`.

5.1.20 Upgrades involving `libgc1c2` need two runs

Packages that depend on `libgc1c2` in buster (e.g. `guile-2.2-libs`) may be held back during the first full upgrade run to bullseye. Doing a second upgrade normally solves the issue. The background of the issue can be found in [bug #988963](https://bugs.debian.org/988963) (<https://bugs.debian.org/988963>).

5.1.21 `fail2ban` can't send e-mail using mail from `bsd-mailx`

The `fail2ban` package can be configured to send out e-mail notifications. It does that using `mail`, which is provided by multiple packages in Debian. A security update (needed on systems that use `mail` from `mailutils`) just before the release of bullseye broke this functionality for systems that have `mail` provided by `bsd-mailx`. Users of `fail2ban` in combination with `bsd-mailx` who wish `fail2ban` to send out e-mail should either switch to a different provider for `mail` or manually unapply [the upstream commit](https://github.com/fail2ban/fail2ban/commit/410a6ce5c80dd981c22752da034f2529b5eee8) (<https://github.com/fail2ban/fail2ban/commit/410a6ce5c80dd981c22752da034f2529b5eee8>) (which inserted the string `"-E 'set escape'"` in multiple places under `/etc/fail2ban/action.d/`).

5.1.22 No new SSH connections possible during upgrade

Although existing Secure Shell (SSH) connections should continue to work through the upgrade as usual, due to unfortunate circumstances the period when new SSH connections cannot be established is longer than usual. If the upgrade is being carried out over an SSH connection which might be interrupted, it's recommended to upgrade `openssh-server` before upgrading the full system.

5.1.23 Coses a fer després de reinicialitzar

Quan `apt full-upgrade` ha acabat, el procés "formal" d'actualització s'ha completat. Per l'actualització a bullseye, no hi ha accions especials a fer abans de reinicialitzar el sistema.

5.2 Items not limited to the upgrade process

5.2.1 Limitacions en el suport de seguretat

Hi ha alguns paquets pels quals Debian no pot prometre proveir els requisits mínims de seguretat via backports. Aquests paquets s'esmenten a les subseccions següents.

NOTA

El paquet `debian-security-support` ajuda a fer seguiment de l'estat del suport de seguretat dels paquets instal·lats.

5.2.1.1 Estat de seguretat als navegadors web i els seus motors de renderització

Debian 11 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 and engines with backported security fixes. Additionally, library interdependencies make it extremely difficult to update to newer upstream releases. Therefore, browsers built upon e.g. the webkit and khtml engines¹ are included in bullseye, but not covered by security support. These browsers should not be used against untrusted websites. The webkit2gtk and wpewebkit engines *are* covered by security support.

Per l'ús generalitzat del navegador web recomanem Firefox o Chromium. Aquests s'actualitzaran re-construint els llançaments ESR actuals per al llançament estable. La mateixa estratègia serà aplicada a Thunderbird.

5.2.1.2 OpenJDK 17

Debian bullseye comes with an early access version of OpenJDK 17 (the next expected OpenJDK LTS version after OpenJDK 11), to avoid the rather tedious bootstrap process. The plan is for OpenJDK 17 to receive an update in bullseye to the final upstream release announced for October 2021, followed by security updates on a best effort basis, but users should not expect to see updates for every quarterly upstream security update.

5.2.1.3 Go-based packages

The Debian infrastructure currently has problems with rebuilding packages of types that systematically use static linking. Before buster this wasn't a problem in practice, but with the growth of the Go ecosystem it means that Go-based packages will be covered by limited security support until the infrastructure is improved to deal with them maintainably.

If updates are warranted for Go development libraries, they can only come via regular point releases, which may be slow in arriving.

5.2.2 Accedir a la configuració de GNOME sense ratolí

Sense un dispositiu de punter, no hi ha manera directa de canviar la configuració de l'aplicació de configuració de GNOME proveïda pel paquet `gnome-control-center`. Com a sol·lució temporal, podeu navegar amb la barra lateral al contingut principal pressionant la tecla **Right Arrow** dues vegades. Per tornar a la barra lateral, podeu començar una cerca amb `Ctrl+F`, teclegeu alguna cosa, aleshores premeu **Esc** per a cancel·lar la cerca. Ara podeu utilitzar les tecles **Up Arrow** i **Down Arrow** per a navegar la barra lateral. No és possible seleccionar els resultats de la cerca amb el teclat.

5.2.3 The rescue boot option is unusable without a root password

With the implementation of `sulogin` used since buster, booting with the `rescue` option always requires the root password. If one has not been set, this makes the rescue mode effectively unusable. However it is still possible to boot using the kernel parameter `init=/sbin/sulogin --force`

¹These engines are shipped in a number of different source packages and the concern applies to all packages shipping them. The concern also extends to web rendering engines not explicitly mentioned here, with the exception of webkit2gtk and the new wpewebkit.

To configure `systemd` to do the equivalent of this whenever it boots into rescue mode (also known as single mode: see [systemd\(1\)](https://manpages.debian.org//bullseye/systemd/systemd.1.html) (<https://manpages.debian.org//bullseye/systemd/systemd.1.html>)), run `sudo systemctl edit rescue.service` and create a file saying just:

```
[Service]
Environment=SYSTEMD_SULOGIN_FORCE=1
```

It might also (or instead) be useful to do this for the `emergency.service` unit, which is started *automatically* in the case of certain errors (see [systemd.special\(7\)](https://manpages.debian.org//bullseye/systemd/systemd.special.7.html) (<https://manpages.debian.org//bullseye/systemd/systemd.special.7.html>)), or if emergency is added to the kernel command line (e.g. if the system can't be recovered by using the rescue mode).

For background and a discussion on the security implications see [#802211](https://bugs.debian.org//802211) (<https://bugs.debian.org//802211>).

5.3 Obsolescence and deprecation

5.3.1 Paquets notables obsolets

La següent llista de paquets notablement coneguts han sigut descatalogats (vegeu Secció 4.8 per a una descripció).

La llista de paquets obsolets inclou:

- The `lilo` package has been removed from bullseye. The successor of `lilo` as boot loader is `grub2`.
- The Mailman mailing list manager suite version 3 is the only available version of Mailman in this release. Mailman has been split up into various components; the core is available in the package `mailman3` and the full suite can be obtained via the `mailman3-full` metapackage.

The legacy Mailman version 2.1 is no longer available (this used to be the package `mailman`). This branch depends on Python 2 which is no longer available in Debian.

For upgrading instructions, please see [the project's migration documentation](https://docs.mailman3.org/en/latest/migration.html). (<https://docs.mailman3.org/en/latest/migration.html>)

- The Linux kernel no longer provides `isdn4linux (i4l)` support. Consequently, the related user-land packages `isdnutils`, `isdnactivecards`, `drdsl` and `ibod` have been removed from the archives.
- The deprecated `libappindicator` libraries are no longer provided. As a result, the related packages `libappindicator1`, `libappindicator3-1` and `libappindicator-dev` are no longer available. This is expected to cause dependency errors for third-party software that still depends on `libappindicator` to provide system tray and indicator support.

Debian is using `libayatana-appindicator` as the successor of `libappindicator`. For technical background see [this announcement](https://lists.debian.org/debian-devel/2018/03/msg00506.html) (<https://lists.debian.org/debian-devel/2018/03/msg00506.html>).

- Debian no longer provides `chef`. If you use Chef for configuration management, the best upgrade path is probably to switch to using the packages provided by [Chef Inc](https://www.chef.io/) (<https://www.chef.io/>).

For background on the removal, see [the removal request](https://bugs.debian.org/cgi-bin/bugreport.cgi?bug=963750) (<https://bugs.debian.org/cgi-bin/bugreport.cgi?bug=963750>).

- Python 2 is already beyond its End Of Life, and will receive no security updates. It is not supported for running applications, and packages relying on it have either been switched to Python 3 or removed. However, Debian bullseye does still include a version of Python 2.7, as well as a small number of Python 2 build tools such as `python-setuptools`. These are present only because they are required for a few application build processes that have not yet been converted to Python 3.
- The `aufs-dkms` package is not part of bullseye. Most `aufs-dkms` users should be able to switch to `overlayfs`, which provides similar functionality with kernel support. However, it's possible to have a Debian installation on a filesystem that is not compatible with `overlayfs`, e.g. `xfs`

without `d_type`. Users of `aufs-dkms` are advised to migrate away from `aufs-dkms` before upgrading to bullseye.

- The network connection manager `wicd` will no longer be available after the upgrade, so to avoid the danger of losing connectivity users are recommended to switch before the upgrade to an alternative such as `network-manager` or `connman`.

5.3.2 Components desfasats a bullseye

Al proper llançament de Debian 12 (anomenada bookworm) algunes de les funcionalitats quedaran desfasades. Els usuaris hauran de migrar a altres alternatives per tal d'evitar problemes a l'hora d'actualitzar a Debian 12.

Això inclou les funcionalitats següents:

- The historical justifications for the filesystem layout with `/bin`, `/sbin`, and `/lib` directories separate from their equivalents under `/usr` no longer apply today; see the [Freedesktop.org summary](https://www.freedesktop.org/wiki/Software/systemd/TheCaseForTheUsrMerge) (<https://www.freedesktop.org/wiki/Software/systemd/TheCaseForTheUsrMerge>). Debian bullseye will be the last Debian release that supports the non-merged-usr layout; for systems with a legacy layout that have been upgraded without a reinstall, the `usrmerge` package exists to do the conversion if desired.
- bullseye is the final Debian release to ship **apt-key**. Keys should be managed by dropping files into `/etc/apt/trusted.gpg.d` instead, in binary format as created by `gpg --export` with a `.gpg` extension, or ASCII armored with a `.asc` extension.

A replacement for **apt-key list** to manually investigate the keyring is planned, but work has not started yet.

- The slapd database backends [slapd-bdb\(5\)](https://manpages.debian.org//bullseye/slapd/slapd-bdb.5.html) (<https://manpages.debian.org//bullseye/slapd/slapd-bdb.5.html>), [slapd-hdb\(5\)](https://manpages.debian.org//bullseye/slapd/slapd-hdb.5.html) (<https://manpages.debian.org//bullseye/slapd/slapd-hdb.5.html>), and [slapd-shell\(5\)](https://manpages.debian.org//bullseye/slapd/slapd-shell.5.html) (<https://manpages.debian.org//bullseye/slapd/slapd-shell.5.html>) are being retired and will not be included in Debian 12. LDAP databases using the `bdb` or `hdb` backends should be migrated to the [slapd-mdb\(5\)](https://manpages.debian.org//bullseye/slapd/slapd-mdb.5.html) (<https://manpages.debian.org//bullseye/slapd/slapd-mdb.5.html>) backend.

Additionally, the [slapd-perl\(5\)](https://manpages.debian.org//bullseye/slapd/slapd-perl.5.html) (<https://manpages.debian.org//bullseye/slapd/slapd-perl.5.html>) and [slapd-sql\(5\)](https://manpages.debian.org//bullseye/slapd/slapd-sql.5.html) (<https://manpages.debian.org//bullseye/slapd/slapd-sql.5.html>) backends are deprecated and may be removed in a future release.

The OpenLDAP Project does not support retired or deprecated backends. Support for these backends in Debian 11 is on a best effort basis.

5.3.3 No-longer-supported hardware

For a number of armel-based devices that were supported in buster, it is no longer viable for Debian to build the required Linux kernel, due to hardware limitations. The unsupported devices are:

- QNAP Turbo Station (TS-xxx)
- HP Media Vault mv2120

Users of these platforms who wish to upgrade to bullseye nevertheless should keep the buster APT sources enabled. Before upgrading they should add an APT preferences file containing:

```
Package: linux-image-marvell
Pin: release n=buster
Pin-Priority: 900
```

The security support for this configuration will only last until buster's End Of Life.

5.4 Known severe bugs

Although Debian releases when it's ready, that unfortunately doesn't mean there are no known bugs. As part of the release process all the bugs of severity serious or higher are actively tracked by the Release Team, so an [overview of those bugs](https://bugs.debian.org/cgi-bin/pkgreport.cgi?users=release.debian.org@packages.debian.org;tag=bullseye-can-defer) (<https://bugs.debian.org/cgi-bin/pkgreport.cgi?users=release.debian.org@packages.debian.org;tag=bullseye-can-defer>) that were tagged to be ignored in the last part of releasing bullseye can be found in the [Debian Bug Tracking System](https://bugs.debian.org/) (<https://bugs.debian.org/>). The following bugs were affecting bullseye at the time of the release and worth mentioning in this document:

Bug number	Package (source or binary)	Description
922981 (https://bugs.debian.org/922981)	ca-certificates-java	ca-certificates-java: /etc/ca-certificates/update.d/jks-keystore doesn't update /etc/ssl/certs/java/cacerts
990026 (https://bugs.debian.org/990026)	cron	cron: Reduced charset in MAILTO causes breakage
991081 (https://bugs.debian.org/991081)	gir1.2-diodon-1.0	gir1.2-diodon-1.0 lacks dependencies
990318 (https://bugs.debian.org/990318)	python-pkg-resources	python-pkg-resources: please add Breaks against the unversioned python packages
991449 (https://bugs.debian.org/991449)	fail2ban	fix for CVE-2021-32749 breaks systems with mail from bsd-mailx
990708 (https://bugs.debian.org/990708)	mariadb-server-10.5, galera	mariadb-server-10.5: upgrade problems due to galera-3 -> galera-4 switch
980429 (https://bugs.debian.org/980429)	src:gcc-10	g++-10: spurious c++17 mode segmentation fault in append_to_statement_list_1 (tree-iterator.c:65)
980609 (https://bugs.debian.org/980609)	src:gcc-10	missing i386-cpuinfo.h
984574 (https://bugs.debian.org/984574)	gcc-10-base	gcc-10-base: please add Breaks: gcc-8-base (<< 8.4)
984931 (https://bugs.debian.org/984931)	git-el	git-el,elpa-magit: fails to install: /usr/lib/emacs-common/packages/install/git emacs failed at /usr/lib/emacs-common/lib.pl line 19, <TSORT> line 7.
987264 (https://bugs.debian.org/987264)	git-el	git-el: fails to install with xemacs21
991082 (https://bugs.debian.org/991082)	gir1.2-gtd-1.0	gir1.2-gtd-1.0 has empty Depends
948739 (https://bugs.debian.org/948739)	gparsed	gparsed should not mask .mount units
984714 (https://bugs.debian.org/984714)	gparsed	gparsed should suggest exfat-progs and backport the commit that rejects exfat-utils
968368 (https://bugs.debian.org/968368)	ifenslave	ifenslave: Option bond-master fails to add interface to bond
990428 (https://bugs.debian.org/990428)	ifenslave	ifenslave: Bonding not working on bullseye (using bond-slaves config)

Bug number	Package (source or binary)	Description
991113 (https://bugs.debian.org/991113)	libpam-chroot	libpam-chroot installs pam_chroot.so into the wrong directory
989545 (https://bugs.debian.org/989545)	src:llvm-toolchain-11	libgl1-mesa-dri: si_texture.c:1727 si_texture_transfer_map - failed to create temporary texture to hold untiled copy
982459 (https://bugs.debian.org/982459)	mdadm	mdadm --examine in chroot without /proc,/dev,/sys mounted corrupts host's filesystem
981054 (https://bugs.debian.org/981054)	openipmi	openipmi: Missing dependency on kmod
948318 (https://bugs.debian.org/948318)	openssh-server	openssh-server: Unable to restart sshd restart after upgrade to version 8.1p1-2
991151 (https://bugs.debian.org/991151)	procps	procps: dropped the reload option from the init script, breaking corekeeper
989103 (https://bugs.debian.org/989103)	pulseaudio	pulseaudio regressed on control = Wave configuration
984580 (https://bugs.debian.org/984580)	libpython3.9-dev	libpython3.9-dev: missing dependency on zlib1g-dev
990417 (https://bugs.debian.org/990417)	src:qemu	openjdk-11-jre-headless: running java in qemu s390 gives a SIGILL at C [linux-vdso64.so.1+0x6f8] _kernel_getcpu+0x8
859926 (https://bugs.debian.org/859926)	speech-dispatcher	breaks with pulse-audio as output when spawned by speechd-up from init system
932501 (https://bugs.debian.org/932501)	src:squid-deb-proxy	squid-deb-proxy: daemon does not start due to the conf file not being allowed by apparmor
991588 (https://bugs.debian.org/991588)	tpm2-abrmd	tpm2-abrmd should not use Requires = systemd-udev-settle.service in its unit
991939 (https://bugs.debian.org/991939)	libjs-bootstrap4	libjs-bootstrap4: broken symlinks: /usr/share/javascript/bootstrap4/css/bootstrap*.css.map -> ../../../../nodejs/bootstrap/dist/css/bootstrap*.css.map
991822 (https://bugs.debian.org/991822)	src:wine	src:wine: dh_auto_clean deletes unrelated files outside of package source
988477 (https://bugs.debian.org/988477)	src:xen	xen-hypervisor-4.14-amd64: xen dmesg shows (XEN) AMD-Vi: IO_PAGE_FAULT on sata pci device
991788 (https://bugs.debian.org/991788)	xfce4-settings	xfce4-settings: black screen after suspend when laptop lid is closed and re-opened

Capítol 6

Més informació sobre Debian

6.1 Llegir més

Més enllà d'estes notes de llançament i de la guia d'instal·lació, hi ha més documentació de Debian a la vostra disposició al Projecte de Documentació de Debian (DDP), que té com a objectiu crear documentació de gran qualitat pels usuaris i desenvolupadors de Debian. Esta documentació inclou la guia de referència de Debian, la guia del nou mantenidor, les PMF de Debian i molt més. Per obtenir informació detallada dels recursos al vostre abast, visiteu el [lloc web del projecte de documentació de Debian](https://www.debian.org/doc/) (<https://www.debian.org/doc/>) i el lloc [Viqui de Debian](https://wiki.debian.org/) (<https://wiki.debian.org/>).

Trobareu informació de cada paquet instal·lada a `/usr/share/doc/paquet`. Esta pot incloure informació dels drets d'autor, detalls específics de Debian i la documentació del desenvolupament principal.

6.2 Trobar ajuda

Hi ha moltes fonts d'ajuda, consells i suport pels usuaris de Debian, però tan sols hauria de considerar-se si no l'heu aconseguit a cap lloc després de la recerca dins la documentació sobre problema. Esta secció proporciona una introducció curta a les que poden ser d'ajuda pels usuaris de Debian.

6.2.1 Llistes de correu

Les llistes de correu més interessants pels usuaris de Debian són la llista `debian-user` (en anglès) i altres llistes com `debian-user-catalan` (en català). Per obtenir més informació d'estes llistes i detalls de com subscriure's, vegeu <https://lists.debian.org/>. Abans d'enviar la vostra pregunta, comproveu els arxius amb les respostes i respecteu les normes de la llista.

6.2.2 Internet Relay Chat

Debian té un canal IRC dedicat al suport i ajuda als usuaris de Debian que es troba a la xarxa d'IRC OFTC. Per accedir al canal, dirigiu el vostre client IRC a `irc.debian.org` i uniu-vos a `#debian`.

Seguiu les normes del canal, respectant completament els altres usuaris. Les normes les podeu trobar al [Viqui de Debian](https://wiki.debian.org/DebianIRC) (<https://wiki.debian.org/DebianIRC>).

Per obtenir més informació d'OFTC visiteu la pàgina web del [lloc web](http://www.oftc.net/) (<http://www.oftc.net/>).

6.3 Informes d'error

Ens esforcem per fer de Debian un sistema operatiu de gran qualitat, però això no vol dir que els paquets que subministrem estiguen totalment lliures d'errors. D'acord amb la filosofia de “desenvolupament obert” de Debian, i com un servei als nostres usuaris, facilitem tota la informació dels errors que s'han rebut en el nostre sistema de seguiment d'errors (BTS). Podeu navegar pel BTS a <https://bugs.debian.org/>.

Si trobeu alguna errada a la distribució o a algun paquet de programari que forme part d'ella, informeu-ne de manera que es pugui resoldre de la millor manera en algun llançament posterior. L'enviament

d'informes d'errors requereix d'una adreça vàlida de correu electrònic. El demanem perquè així podrem fer seguiment d'errors i els desenvolupadors podran posar-se en contacte amb els que l'han enviat per si és necessària alguna informació addicional.

Podeu enviar un informe d'error utilitzant el programa **reportbug** o bé utilitzant manualment el correu electrònic. Podeu llegir més informació del sistema de seguiment d'errors i de com utilitzar-lo llegint la documentació de referència (que trobareu a `/usr/share/doc/debian` si teniu el paquet `doc-debian` instal·lat) o en línia al **sistema de seguiment d'errors** (<https://bugs.debian.org/>).

6.4 Col·laborar amb Debian

No necessiteu ser experts per col·laborar amb Debian. Ajudant els usuaris en problemes a les diferents **l·listes** (<https://lists.debian.org/>) de suport pels usuaris, ja esteu contribuint a la comunitat. Identificant (i també resolent) els problemes associats al desenvolupament de la distribució mitjançant la participació a les **l·listes** (<https://lists.debian.org/>) de desenvolupament també és de molta ajuda. Per mantenir l'alta qualitat de la distribució Debian, **envieu els informes d'errors** (<https://bugs.debian.org/>) i ajudeu els desenvolupadors a fer el seu seguiment per corregir-los. Si teniu traça amb les paraules aleshores potser voldreu col·laborant de forma més activa ajudant a escriure **documentació** (<https://www.debian.org/doc/vcs>) o **traduint** (<https://www.debian.org/international/>) la que ja existeix en el vostre idioma.

Si podeu dedicar més temps, podríeu gestionar una de les peces de programari lliure de la col·lecció a Debian. És de molta ajuda que la gent adopte o mantinga elements que la gent ha demanat per que s'incloga a Debian. La **base de dades de paquets en perspectiva i amb feina pendent** (<https://www.debian.org/devel/wnpp/>) detalla tota esta informació. Si teniu interès en grups especials aleshores trobareu més diversió col·laborant amb alguns **subprojectes** (<https://www.debian.org/devel/#projects>) de Debian que inclouen ports a arquitectures concretes, **barreges pures de Debian** (<https://wiki.debian.org/DebianPureBlends>) per a grups d'usuaris específics, entre molts altres.

De qualsevol manera, si esteu treballant a la comunitat de programari lliure d'esta manera, com a usuari, programador, escrivint o traduint, ja esteu ajudant a l'esforç pel programari lliure. Col·laborar és gratificant i divertit, i també us permet conèixer nova gent que us donarà un càlid sentiment de comunitat.

Capítol 7

Glossari

ACPI

Advanced Configuration and Power Interface (Configuració avançada i interfície d'energia)

ALSA

Advanced Linux Sound Architecture (Arquitectura avançada de so de Linux)

BD

Disc Blu-ray

CD

Compact Disc (Disc compacte)

CD-ROM

Disc compacte de només lectura

DHCP

Protocol dinàmic de configuració de màquina

DLBD

Disc Blu-ray de dues capes

DNS

Domain Name System (Sistema de noms de domini)

DVD

Digital Versatile Disc

GIMP

GNU Image Manipulation Program (Programa de manipulació d'imatges de GNU)

GNU

GNU's Not Unix (GNU no és Unix)

GPG

GNU Privacy Guard

LDAP

Lightweight Directory Access Protocol (Protocol lleuger d'accés al directori)

LSB

Linux Standard Base

LVM

Logical Volume Manager (Gestor de volums lògics)

MTA

Mail Transport Agent (Agent de transport de correu)

NBD

Network Block Device (Dispositiu de blocs en xarxa)

NFS

Network File System (Sistema de fitxers en xarxa)

NIC

Network Interface Card (Targeta d'interfície de xarxa)

NIS

Network Information Service (Servei d'informació de xarxa)

PHP

PHP: Processador d'hipertext

RAID

Redundant Array of Independent Disks (Matriu redundant de discs independents)

SATA

Serial Advanced Technology Attachment (Tecnologia avançada d'accessoris en sèrie)

SSL

Secure Sockets Layer (Capa de connexió segura)

TLS

Transport Layer Security (Capa de seguretat de transport)

UEFI

Unified Extensible Firmware Interface (Interfície unificada i extensible de microprogramari)

USB

Universal Serial Bus (Bus sèrie universal)

UUID

Universally Unique Identifier (Identificador únic universal)

WPA

Wi-Fi Protected Access (Accés protegit a Wi-Fi)

Apèndix A

Gestió del vostre sistema buster abans de l'actualització

Este apèndix conté informació de com instal·lar o actualitzar els paquets de buster abans d'actualitzar a bullseye. Açò tan sols serà necessari en algunes situacions concretes.

A.1 Actualització del vostre sistema buster

Bàsicament no hi ha diferències entre qualsevol altra actualització de buster que ja hàgiu fet. L'única diferència és que necessitareu primer assegurar-vos que la vostra llista de paquets encara conté referències a buster tal com s'explica a Secció [A.2](#).

Si actualitzeu el vostre sistema utilitzant una rèplica de Debian, s'actualitzarà de forma automàtica a l'últim revisió del llançament de buster.

A.2 Comprovació dels fitxers de llista de fonts d'APT

If any of the lines in your APT source-list files (see [sources.list\(5\)](#) (<https://manpages.debian.org//bullseye/apt/sources.list.5.html>)) contain references to “stable”, this is effectively pointing to bullseye already. This might not be what you want if you are not yet ready for the upgrade. If you have already run **apt update**, you can still get back without problems by following the procedure below.

If you have also already installed packages from bullseye, there probably is not much point in installing packages from buster 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.

As root, open the relevant APT source-list file (such as `/etc/apt/sources.list`) with your favorite editor, and check all lines beginning with `deb http:`, `deb https:`, `deb tor+http:`, `deb tor+https:`, `URIs: http:`, `URIs: https:`, `URIs: tor+http:` or `URIs: tor+https:` for a reference to “stable”. If you find any, change `stable` to `buster`.

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

IMPORTANT



Do not change any lines that begin with `deb cdrom:` or `URIs: 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.

Si heu fet algun canvi, guardeu el fitxer i executeu

```
# apt update
```

per refrescar la llista de paquets.

A.3 Eliminar fitxers de configuració obsolets

Abans d'actualitzar el vostre sistema a bullseye és recomanable treure els fitxers de configuració antics (com ara `*.dpkg-{new,old}`) a sota el directori `/etc` del sistema.

Apèndix B

Contribuidors de les notes de llançament

Molta gent ha ajudat en les notes de llançament, estan incloses, però no limitades a

D. Barratt, Adam, Di Carlo, Adam, Barth, Andreas, Popescu, Andrei, Bezemer, Anne, Hilliard, Bob, Plessy, Charles, Perrier, Christian, Berg, Christoph, Baumann, Daniel, Prévot, David, Petrișor, Eddy, Kasper, Emmanuel, Arajärvi, Esko, Pop, Frans, Rapagnani, Giovanni, Farquharson, Gordon, Yamane, Hideki, Wansing, Holger, Fernández-Sanguino Peña, Javier, Seidel, Jens, Meurer, Jonas, Nieder, Jonathan, van Baal-Ilić, Joost, Rodin, Josip, Cristau, Julien, Rye, Justin B, Jones, LaMont, Claes, Luk, Michlmayr, Martin, Biebl, Michael, Mühlenhoff, Moritz, Thykier, Niels, Meyerhans, Noah, Kobayashi, Noritada, Aoki, Osamu, Gevers, Paul, Green, Peter, Bradford, Rob, Thibault, Samuel, Bienlein, Simon, Paillard, Simon, Fritsch, Stefan, Langasek, Steve, McIntyre, Steve, Scherer, Tobias, victory, McIntyre, Vincent, i Borgert, W. Martin.

Este document s'ha traduït a molts idiomes. Moltes gràcies als traductors!

Índex alfabètic

A

Apache, 4

B

BIND, 4

C

Calligra, 3

Cryptsetup, 4

D

DocBook XML, 2

Dovecot, 4

E

Exim, 4

G

GCC, 4

GIMP, 4

GNOME, 3

GNUCash, 3

GnuPG, 4

I

Inkscape, 4

K

KDE, 3

L

LibreOffice, 3

LXDE, 3

LXQt, 3

M

MariaDB, 4

MATE, 3

N

Nginx, 4

O

OpenJDK, 4

OpenSSH, 4

P

packages

apt, 2, 14, 25

apt-listchanges, 18

aptitude, 12, 17, 22

aufs-dkms, 29, 30

bsd-mailx, 27

ca-certificates-java, 31

chef, 29

cinder-volume, 24

connman, 30

cron, 31

cups-browsed, 4

cups-daemon, 4

cups-filters, 4

dblatex, 2

debian-goodies, 17

debian-kernel-handbook, 21

debian-security-support, 28

doc-debian, 34

docbook-xsl, 2

drdsl, 29

exfat-fuse, 5

exfat-utils, 6

exfatprogs, 6

fail2ban, 27, 31

firmware-iwlwifi, 27

fuse, 25

fuse3, 25

gcc-10-base, 31

gir1.2-diodon-1.0, 31

gir1.2-gtd-1.0, 31

git-el, 31

glibc, 24

gnome-control-center, 28

gparted, 31

grub2, 29

guile-2.2-libs, 27

gvfs-fuse, 25

ibod, 29

ifenslave, 31

initramfs-tools, 10, 21

intel-microcode, 27

ipp-usb, 4, 5

isdnactivecards, 29

isdnutils, 29

kio-fuse, 25

libappindicator-dev, 29

libappindicator1, 29

libappindicator3-1, 29

libayatana-appindicator, 29

libgc1c2, 27

libjs-bootstrap4, 32

libnss-nis, 24

libnss-nisplus, 24

libpam-chroot, 32

libpython3.9-dev, 32

libsane1, 4, 5

lilo, 29

linux-image-*, 20

linux-image-amd64, 21

linux-source, 21

localepurge, 17

mailman, 29

mailman3, 29

mailman3-full, 29

mailutils, 27

mariadb-server-10.5, galera-4, 31

mdadm, 32
micro-evtd, 11
network-manager, 30
nova-compute, 24
openipmi, 32
openssh-server, 27, 32
popularity-contest, 17
procps, 32
pulseaudio, 32
python-pkg-resources, 31
python-setuptools, 29
rails, 26
rdiff-backup, 26, 27
redmine, 26
release-notes, 1
rsync, 24
rsyslog, 5
sane-airscan, 4
sendmail, 25
slapd, 30
speech-dispatcher, 32
src:gcc-10, 31
src:llvm-toolchain-11, 32
src:qemu, 32
src:squid-deb-proxy, 32
src:wine, 32
src:xen, 32
sshfs, 25
synaptic, 12
systemd, 6
tinc, 11
tpm2-abrmd, 32
udev, 21, 26
unbound, 24
upgrade-reports, 1
usrmerge, 30
vim, 24
vim-addon-manager, 24
vim-scripts, 24
wicd, 30
xfce4-settings, 32
xmlroff, 2
xsltproc, 2

Perl, 4
PHP, 4
Postfix, 4
PostgreSQL, 4

X
Xfce, 3