
Release-Notes for Debian 13 (trixie)

リリース *13.0*

Debian Documentation Team

2023-09-03

目次

第 1 章	はじめに	3
1.1	この文書に関するバグを報告する	3
1.2	アップグレードについての報告をする	4
1.3	この文書のソース	4
第 2 章	Debian 13 の最新情報	5
2.1	サポートするアーキテクチャ	5
2.2	アーカイブエリア	5
2.3	ディストリビューションの最新情報	6
2.3.1	デスクトップとよく知られているパッケージ	6
2.3.2	さらに翻訳された manpage	7
2.3.3	Debian Med Blend からのニュース	8
2.3.4	Debian Astro Blend からのニュース	8
2.3.5	ARM64 でのセキュアブート	9
第 3 章	インストール用システム	11
3.1	インストールシステムの変更点	11
3.2	クラウドへのインストール	11
3.3	コンテナおよび仮想マシンイメージ	12
第 4 章	Debian 12 (bookworm) からのアップグレード	13
4.1	アップグレードの準備	13
4.1.1	あらゆるデータや設定情報をバックアップする	13
4.1.2	事前にユーザに通知する	14
4.1.3	サービスのダウン期間の準備	14
4.1.4	復旧の準備	14
4.1.5	アップグレード用の安全な環境の準備	16
4.2	"純粹"な Debian からの作業開始	16
4.2.1	Debian 12 (bookworm) からのアップグレード	16
4.2.2	最新のポイントリリースへのアップグレード	17
4.2.3	Debian Backports	17
4.2.4	パッケージデータベースの準備	17
4.2.5	利用されなくなったパッケージ	17
4.2.6	Debian 由来でないパッケージを削除する	18
4.2.7	残っている設定ファイルを取り除く	18
4.2.8	non-free コンポーネントと non-free-firmware コンポーネント	18
4.2.9	proposed-updates セクション	18
4.2.10	非公式なソース	18
4.2.11	APT の pin 機能を無効にする	19
4.2.12	gpgv がインストールされているのを確認する	19

4.2.13	パッケージの状態をチェックする	19
4.3	APT source-list ファイルの準備	20
4.3.1	APT のインターネットソースの追加	20
4.3.2	APT のローカルミラーソースの追加	21
4.3.3	APT の光学メディアソースの追加	21
4.4	パッケージのアップグレード	22
4.4.1	セッションの記録	22
4.4.2	パッケージリストの更新	23
4.4.3	アップグレードするのに十分な領域があることを確認する	23
4.4.4	監視システムの停止	25
4.4.5	システムの最小アップグレード	26
4.4.6	システムのアップグレード	26
4.5	アップグレード中の注意点	27
4.5.1	「即時設定は動作しません」で full-upgrade が失敗する	27
4.5.2	予期されるパッケージの削除	27
4.5.3	衝突 (Conflicts) あるいは事前依存 (Pre-Depends) のループ	27
4.5.4	ファイルの衝突	28
4.5.5	設定の変更	28
4.5.6	コンソール接続へセッションの変更	28
4.6	カーネルと関連パッケージのアップグレード	29
4.6.1	カーネルメタパッケージのインストール	29
4.7	次のリリースへの準備	30
4.7.1	削除したパッケージを完全削除する	30
4.8	利用されなくなったパッケージ	30
4.8.1	移行用ダミーパッケージ	31
第 5 章	trixie で注意すべき点	33
5.1	trixie へのアップグレード特有の事項	33
5.1.1	non-free なファームウェアがアーカイブの独自コンポーネントへ移動しました	33
5.1.2	システムクロックの設定をするパッケージについての変更	33
5.1.3	設定管理システム puppet はバージョン 7 にアップグレードされました	34
5.1.4	youtube-dl は yt-dlp にリプレースされました	34
5.1.5	fcitx は複数バージョンをインストールできなくなりました	34
5.1.6	MariaDB パッケージはパッケージ名にバージョン番号を含まなくなりました	35
5.1.7	システムのロギング設定への変更	35
5.1.8	rsyslog での変更が logcheck のようなログ分析ソフトに影響を与えます	36
5.1.9	rsyslog が生成するログファイルが少なくなっています	36
5.1.10	slapd のアップグレードには手動での介入が必要になる可能性があります	37
5.1.11	GRUB はデフォルトでは os-prober を実行しなくなりました	38
5.1.12	GNOME はスクリーンリーダーへのアクセシビリティサポートを縮小しています	38
5.1.13	Baseline for 32-bit PC is now i686	38
5.1.14	polkit 設定についての変更点	38
5.1.15	"merged-/usr" が必須となりました	39
5.1.16	buster からのサポート外のアップグレードが libcrypt1 で失敗する	40
5.1.17	openssh-server no longer reads ~/.pam_environment	40
5.1.18	OpenSSH no longer supports DSA keys	40
5.2	アップグレード後、再起動前にすること	41

5.2.1	アップグレード後も影響がある項目	41
5.2.2	セキュリティサポートにおける制限事項	41
5.2.3	python インタプリタは externally-managed としてマークされます	42
5.2.4	VLC でのハードウェアアクセラレーションを利用したビデオエンコードとデコードのサポートは制限されています	42
5.2.5	systemd-resolved は分割されて独立したパッケージになりました	43
5.2.6	systemd-boot は分割されて独立したパッケージになりました	43
5.2.7	systemd-journal-remote は GnuTLS を使わなくなりました	43
5.2.8	bookworm に向けた adduser の大幅な変更点	43
5.2.9	予測可能な Xen のネットワークインターフェイス名	44
5.2.10	dash でのサーカムフレックス (^) の扱いの変更	44
5.2.11	netcat-openbsd が抽象ソケットをサポートしました	44
5.3	廃止および非推奨となった事柄について	44
5.3.1	特記すべき廃止されたパッケージたち	44
5.3.2	trixie で非推奨となったコンポーネント	45
5.4	既知の重大なバグ	45
第 6 章	Debian に関するさらなる情報	47
6.1	もっと読みたい	47
6.2	手助けを求めるには	47
6.2.1	メーリングリスト	47
6.2.2	インターネットリレーチャット (IRC)	47
6.3	バグを報告する	48
6.4	Debian に貢献する	48
第 7 章	アップグレードの前に bookworm システムを調整する	49
7.1	bookworm システムのアップグレード	49
7.2	APT source-list ファイルのチェック	49
7.3	Performing the upgrade to latest bookworm release	50
7.4	古く不要になった設定ファイルを削除する	50
第 8 章	リリースノートの貢献者たち	51

The Debian Documentation Project <<https://www.debian.org/doc>>.

この文書はフリーソフトウェアです。あなたは、Free Software Foundation が公表した GNU 一般公衆ライセンスの第二版の条件に基づいて、本文書の再頒布および変更を行うことができます。

本プログラムはその有用性が期待されて頒布されるものですが、市場性や特定の目的への適合性に関する暗黙の保証も含め、いかなる保証も行いません。詳細については GNU 一般公衆ライセンスをご覧ください。

あなたは、このプログラムとともに、GNU 一般公衆ライセンスの写しを受け取っているはずです。そうでなければ Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA に手紙でお知らせください。

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

第1章 はじめに

この文書は Debian ディストリビューションのユーザーに、バージョン 13 (コードネーム trixie) での大きな変更点を知らせるものです。

このリリースノートでは、リリース 12 (コードネーム bookworm) から今回のリリースへの安全なアップグレード方法や、その際ユーザーが遭遇する可能性がある既知の問題点についての情報をユーザーに提供しています。

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

注意: 既知の問題点をすべて列挙するのは不可能なので、問題点の予想される広がり具合と影響の大きさの双方に基づいて取舍選択していることに注意してください。

Debian の 1 つ前のリリースからのアップグレード (この場合、bookworm からのアップグレード) のみがサポート・記述されていることに注意してください。さらに古いリリースからのアップグレードが必要な場合は、過去のリリースノートを読み、まず bookworm へとアップグレードすることをお勧めします。

1.1 この文書に関するバグを報告する

私たちは、この文書で説明されているすべての異なるアップグレード手順を試し、また、ユーザーが直面する可能性のある、すべての問題を想定しました。

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

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

1.2 アップグレードについての報告をする

We welcome any information from users related to upgrades from bookworm to trixie. If you are willing to share information please file a bug in the [bug tracking system](#) against the **upgrade-reports** package with your results. We request that you compress any attachments that are included (using gzip).

アップグレードについての報告を提出する際には、以下の情報を含めてください。

- The status of your package database before and after the upgrade: **dpkg**'s status database available at `/var/lib/dpkg/status` and **apt**'s package state information, available at `/var/lib/apt/extended_states`. You should have made a backup before the upgrade as described at [あらゆるデータや設定情報をバックアップする](#), but you can also find backups of `/var/lib/dpkg/status` in `/var/backups`.
- `script` を使用して作成したセッションのログ。セッションの記録で説明します。
- `/var/log/apt/term.log` にある `apt` のログか、`/var/log/aptitude` にある `aptitude` のログ。

注釈: バグ報告に情報を含める前に、慎重に扱うべき情報や機密情報がログに含まれていないかある程度時間をかけて検査し、ログから削除してください。なぜなら、バグ報告に含まれる情報は公開データベースで公表されるからです。

1.3 この文書のソース

The source of this document is in reStructuredText format, using the sphinx converter. The HTML version is generated using `sphinx-build -b html`. The PDF version is generated using `sphinx-build -b latex`. Sources for the Release Notes are available in the Git repository of the *Debian Documentation Project*. You can use the [web interface](#) to access its files individually through the web and see their changes. For more information on how to access Git please consult the [Debian Documentation Project VCS information pages](#).

第2章 Debian 13 の最新情報

この章のより詳しい情報は [Wiki](#) を参照してください。

2.1 サポートするアーキテクチャ

Debian trixie で公式にサポートされているアーキテクチャは以下のとおりです。

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

Baseline bump for 32-bit PC to i686

The 32-bit PC support (known as the Debian architecture i386) now requires the "long NOP" instruction. Please refer to [Baseline for 32-bit PC is now i686](#) for more information.

You can read more about port status, and port-specific information for your architecture at the [Debian port web pages](#).

2.2 アーカイブエリア

以下の、社会契約と Debian ポリシーで言及されているアーカイブエリアが長期間に渡って存在しています:

- main: Debian ディストリビューションそのもの
- contrib: Debian ディストリビューションと互換性がある補助パッケージ群だが、ビルドまたは動作にディストリビューション外のソフトウェアを必要とする
- non-free: DFSG (Debian フリーソフトウェアガイドライン) に適合しない、あるいは配布を難しくする他の問題があるが、Debian ディストリビューションと互換性がある補助パッケージ群

non-free なファームウェアに関する [2022 年の一般決議](#) によって、社会契約の第 5 条へ以下の文章が追加されました:

Debian の公式メディアには、Debian システムの一部ではないファームウェアが含まれていることがあります。これはそのようなファームウェアを必要とするハードウェアで Debian を利用可能とするのに他の方法がないためです。

未だ社会契約や Debian ポリシーには明示的に反映されていませんが、新しいアーカイブエリアが導入されたことで、non-free なファームウェアを他の non-free なパッケージから隔離できています。

- non-free-firmware

ほとんどの non-free なファームウェアパッケージが Debian 13 リリースの準備中に non-free から non-free-firmware へ移動されました。このクリーンな分離によって contrib や non-free を利用しない、main および non-free-firmware のパッケージを利用した公式インストールイメージのビルドが可能になりました。同様に、これらのインストールイメージで contrib や non-free 無しで main および non-free-firmware のみでシステムをインストール可能となっています。

bookworm からのアップグレードについては *non-free* コンポーネントと *non-free-firmware* コンポーネントを参照してください。

2.3 ディストリビューションの最新情報

Debian のこの新しいリリースには、一つ前のリリースである bookworm に含まれていたよりさらに多くのソフトウェアが含まれています。このディストリビューションには、11294 以上の新しいパッケージが含まれており、全体のパッケージ数は 59551 以上になりました。ディストリビューション中のほとんどのソフトウェア、すなわち約 42821 ものソフトウェアパッケージ (これは bookworm のパッケージ全体の 72% にあたります) が更新されました。また、かなりの数のパッケージ (bookworm のパッケージの 16% にあたる 9519 以上) が、様々な理由でディストリビューションから取り除かれました。これらのパッケージは更新されることはなく、パッケージ管理用のフロントエンドでは 'obsolete' というマークが付けられます。これについては [利用されなくなったパッケージ](#) を参照してください。

2.3.1 デスクトップとよく知られているパッケージ

Debian は今回も複数のデスクトップアプリケーションとデスクトップ環境をサポートしています。中でも GNOME 43, KDE Plasma 5.27, LXDE 11, LXQt 1.2.0, MATE 1.26, Xfce 4.18 があります。

事務用アプリケーションもオフィススイートを含めてアップグレードされています:

- LibreOffice が 7.4 へアップグレードされました。
- GNUCash が 4.13 へアップグレードされました。

またこのリリースには、特に挙げるなら、以下のソフトウェアの更新も含まれています:

パッケージ	12 (bookworm) でのバージョン	13 (trixie) でのバージョン
Apache	2.4.54	2.4.57
Bash	5.1	5.2.15
BIND DNS Server	9.16	9.18
Cryptsetup	2.3	2.6
Emacs	27.1	28.2
Exim default e-mail server	4.94	4.96
GNU Compiler Collection as default compiler	10.2	12.2
GIMP	2.10.22	2.10.34
GnuPG	2.2.27	2.2.40
Inkscape	1.0.2	1.2.2
the GNU C library	2.31	2.36
Linux カーネルイメージ	5.10 シリーズ	6.1 シリーズ
LLVM/Clang ツールチェイン	9.0.1, 11.0.1 (デフォルト) そして 13.0.1	13.0.1, 14.0 (デフォルト) そして 15.0.6
MariaDB	10.5	10.11
Nginx	1.18	1.22
OpenJDK	11	17
OpenLDAP	2.4.57	2.5.13
OpenSSH	8.4p1	9.2p1
OpenSSL	1.1.1n	3.0.8
Perl	5.32	5.36
PHP	7.4	8.2
Postfix MTA	3.5	3.7
PostgreSQL	13	15
Python 3	3.9.2	3.11.2
Rustc	1.48	1.63
Samba	4.13	4.17
Systemd	247	252
Vim	8.2	9.0

2.3.2 さらに翻訳された manpage

翻訳者らのおかげで、man-page 形式でのさらに多くのドキュメントがこれまでより多くの言語で利用できるようになっています。例として、多くの manpage がチェコ語・デンマーク語・ギリシャ語・フィンランド語・インドネシア語・マケドニア語・ノルウェー語 (ブークモール)・ロシア語・セルビア語・スウェーデン語・ウクライナ語・ベトナム語で利用できるようになり、さらに `systemd` の全ての manpage がドイツ語になっています。

To ensure the `man` command shows the documentation in your language (where possible), install the right `manpages-lang` package and make sure your locale is correctly configured by using

```
# dpkg-reconfigure locales
```

2.3.3 Debian Med Blend からのニュース

リリースごとに医療・ライフサイエンス分野の新規パッケージが追加されています。新パッケージ **shiny-server** は R を使った科学向けウェブアプリケーションを簡易に作成してくれるので特に言及する価値があるでしょう。また、我々は Debian Med team がメンテナンスしているパッケージ群へ継続的インテグレーション (CI) サポートの提供に向けた努力を続けています。

Debian Med team は常にユーザーからのフィードバックに興味を持っています。パッケージ化されていないフリーソフトウェアのパッケージ化リクエストや、testing にある新規パッケージやより新しいバージョンのパッケージを backports へ投入することについては特にです。

Debian Med team がメンテナンスしているパッケージをインストールするには、`med-*` という名前の Debian bookworm ではバージョン 3.8.x のメタパッケージをインストールしてください。Debian で入手可能な全範囲の生物・医療関連ソフトウェアを参照するには [Debian Med tasks pages](#) にお気軽にお越しください。

2.3.4 Debian Astro Blend からのニュース

Debian bookworm は Debian Astro Pure Blend バージョン 4.0 として提供されます。Debian Astro Pure Blend は天文学の専門家、天文愛好家、そして天文学に興味を持つ全ての人々へ素晴らしいワンストップソリューションを提供し続けています。Debian Astro のほぼ全てのパッケージは新しいバージョンへ更新されていますが、新しいソフトウェアパッケージもいくつか存在します。

電波天文学者向けにオープンソースの相関器である **openvlbi** が含まれるようになりました。新規パッケージの **astap** と **planetary-system-stacker** はイメージ・スタッキング法や天体位置測定学に役立ちます。INDI プロトコルをサポートする大量の新しいドライバとライブラリがパッケージ化されて Debian に含まれており、利用できるようになっています。

The new Astropy affiliated packages **python3-extinction**, **python3-sncosmo**, **python3-specreduce**, and **python3-synphot** are included, as well as packages created around **python3-yt** and **python3-sunpy**. Python support for the ASDF file format is much extended, while the Java ecosystem is extended with libraries handling the ECSV and TFCAT file formats, primarily for use with topcat.

完全な一覧とさらなる情報を得たい方は [the Astro Blend page](#) を確認してください。

2.3.5 ARM64 でのセキュアブート

ARM64 でのセキュアブートサポートが trixie にて再導入されました。UEFI が利用可能な ARM64 ハードウェアのユーザーはセキュアブートモードを有効にして起動が可能となり、セキュリティ機能の恩恵を最大限に受けることができます。セキュアブートを有効にしたシステムを使うには、**grub-efi-arm64-signed** および **shim-signed** パッケージがインストールされているのを確認してから、デバイスのファームウェア設定画面でセキュアブートを有効にして再起動してください。

[Wiki](#) にはセキュアブートの使い方とデバッグ方法についてのさらなる情報があります。

第3章 インストール用システム

Debian Installer は公式の Debian インストールシステムです。このインストーラーは、様々なインストール方法を提供しています。お使いのシステムにインストールするのにどの方法が利用できるかは、使っているアーキテクチャに依存します。

Images of the installer for trixie can be found together with the Installation Guide on the Debian website (<https://www.debian.org/releases/trixie/debian-installer/>).

インストールガイドは、Debian 公式 DVD セット (CD/blu-ray) の 1 枚目の、次の場所にも含まれています。

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

You may also want to check the errata for debian-installer at <https://www.debian.org/releases/trixie/debian-installer#errata> for a list of known issues.

3.1 インストールシステムの変更点

There has been a lot of development on the Debian Installer since its previous official release with Debian 12, resulting in improved hardware support and some exciting new features or improvements.

If you are interested in an overview of the changes since bookworm, please check the release announcements for the trixie beta and RC releases available from the Debian Installer's [news history](#).

3.2 クラウドへのインストール

The [cloud team](#) publishes Debian trixie for several popular cloud computing services including:

- アマゾン ウェブ サービス (AWS)
- Microsoft Azure
- OpenStack
- 仮想マシン (Plain VM)

クラウド用イメージでは cloud-init 経由の自動フックを提供し、特別に最適化されたカーネルパッケージと grub 設定を使ってインスタンスを高速に起動することを優先します。様々なアーキテクチャをサポートするイメージを必要に応じて提供し、クラウドサービスが提供するすべての機能をサポートするようクラウドチームは努力しています。

The cloud team will provide updated images until the end of the LTS period for trixie. New images are typically released for each point release and after security fixes for critical packages. The cloud team's full support policy can be found [here](#).

More details are available at <https://cloud.debian.org/> and on the [wiki](#).

3.3 コンテナおよび仮想マシンイメージ

Multi-architecture Debian trixie container images are available on [Docker Hub](#). 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](#).

第4章 Debian 12 (bookworm) からのアップグレード

4.1 アップグレードの準備

アップグレードの前には、[trixie](#) で注意すべき点 に書かれている情報も読むことをお勧めします。この章に書かれている問題点は、アップグレードの過程と直接は関係がないかもしれませんが、それでもアップグレードを開始する前に知っておくべき重要事項である可能性があります。

4.1.1 あらゆるデータや設定情報をバックアップする

システムをアップグレードする前に、完全なバックアップを取っておくよう強くお勧めします。少なくとも、失いたくないデータや設定情報だけでもバックアップしておきましょう。アップグレードのツールや処理はきわめて信頼性の高いものですが、アップグレードの最中にハードウェア障害が起こると、システムに大きなダメージを与えることがあります。

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.

あらゆるパッケージのインストール処理はスーパーユーザ特権で実行されなければならないため、`root` としてログインするか `su` や `sudo` を使って、必要なアクセス権限を得てください。

アップグレードにあたって事前に整えなければならない条件がいくつかあります。実際にアップグレードを実行する前にそれらを確認してください。

4.1.2 事前にユーザに通知する

アップグレードの前には、その予定をすべてのユーザに知らせるとよいでしょう。ただ、システムに ssh 接続などでアクセスしてきているユーザが、アップグレードの最中にそう気付くことはほとんどないはずで、また、作業を続行できるはずで

万一の対策をしたければ、アップグレードの前に /home パーティションをバックアップするか、アンマウントしておきましょう。

trixie にアップグレードするときはおそらくカーネルをアップグレードしなければならないので、通常は再起動が必要です。通常、これはアップグレード完了後に実施します。

4.1.3 サービスのダウン期間の準備

システムが提供しているサービスで、アップグレードに含まれるパッケージが関連するサービスがあるかもしれません。この場合、注意して欲しいのですが、アップグレード作業中に関連パッケージが置換・設定される際、これらのサービスが停止します。この間、サービスは利用できなくなります。

これらのサービスに対する実際のダウン期間は、システム中でアップグレードされるパッケージ数に応じて違いますし、このダウン期間には (もしあれば) システム管理者が各パッケージのアップグレードに対する設定の質問への回答に費やす時間も含まれます。アップグレード作業が放置されたままでいて、システムがアップグレード中に入力が必要とした場合、非常に長期間サービスが利用ができなくなる可能性が非常に高いでしょう¹

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 *Minimal system upgrade*, 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 *Upgrading the system*. 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 復旧の準備

Debian はシステムがブートできる状態を常に確保するように努めていますが、アップグレード後のシステム再起動で問題に遭遇する可能性は常にあります。既知の潜在的な問題点の多くは、このリリースノートの本章と次章で述べられています。

上述の理由により、システムが再起動に失敗したり、リモート管理されているシステムならネットワーク接続の確立に失敗した場合に備え、復旧できる手立てを整えておくことが大切です。

ssh 接続経由でリモートからアップグレードを行うのなら、リモートのシリアル端末からサーバにアクセスできるよう、必要な事前準備をしておくことをお勧めします。カーネルをアップグレードして再起動した後、ローカルコンソール経由でシステム設定を修正しなければならないことがあります。また、アップグレード中に誤ってシステムが再起動された場合にも、ローカルコンソールを使って復旧する必要に迫られることがあります。

¹ debconf の優先度を、とても高いレベルに設定していると設定プロンプトを抑制できますが、デフォルト値があなたのシステムに合わない場合、サービスはそのままでは起動に失敗することでしょう。

² 例: DNS や DHCP サービス、特に冗長性やフェイルオーバー機能が無い場合。DHCP の例では、リースタイムがアップグレード作業が完了する時間よりも短い場合、エンドユーザはネットワークから切り離されるでしょう。

For emergency recovery we generally recommend using the *rescue mode* of the trixie 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 (at <https://www.debian.org/releases/trixie/installmanual>) and the [Debian Installer 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* 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.

initrd を使った起動中のデバッグシェル

initramfs-tools パッケージは生成した *initrd* にデバッグシェルを収録します³。例えば、*initrd* がルートファイルシステムをマウントできなければ、デバッグシェル内に移るでしょう。このデバッグシェルは、問題の追跡、そしておそらくは修正の手助けとなる基本的なコマンドを備えています。

チェックすべき基本的事項としては、次のようなものがあります。/dev 内に適切なデバイスファイルが存在するか、どのモジュールがロードされているか (`cat /proc/modules`)、`dmesg` の出力にドライバのロード失敗のエラーが出ていないか、など。`dmesg` の出力はまた、どのデバイスファイルがどのディスクに割り当てられているのかも示してくれます。ルートファイルシステムが期待通りのデバイス上にあるかを確認するために、`echo $ROOT` の出力もチェックすべきでしょう。

問題点を何とか解決できたなら、`exit` とタイプすることでデバッグシェルを終了させ、起動プロセスを失敗した時点から継続できます。もちろん次回の起動時に再び失敗することが無いよう、根本的な問題を修正して *initrd* を再生成する必要があるでしょう。

systemd を使った起動中のデバッグシェル

起動が *systemd* において失敗する場合、カーネルコマンドラインを変更することでデバッグ用の *root* シェルを追加できます。基本的な起動は成功するがサービスが起動に失敗する場合は、カーネルパラメーターに `systemd.unit=rescue.target` を追加すると解決の役に立つかもしれません。

それ以外の場合、カーネルパラメーターとして `systemd.unit=emergency.target` を指定することによって、可能な限り早い段階で *root* シェルが使えるようになります。ですが、これは *root* ファイルシステムを読み書き可能な権限でマウントする前に実行されます。以下を手動で実行する必要があります:

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

もう一つのアプローチは `debug-shell.service` 経由で *systemd* の "早い段階でのデバッグシェル" を有効にすることです。次の起動時にこのサービスは起動プロセスの初期段階で `tty9` にて *root* のログインシェルを立ち上げます。カーネルの起動パラメーターで `systemd.debug-shell=1` と指定して有効にするか、あるいは `systemctl enable debug-shell` で設定を永続的にします (この場合、デバッグが完了した際には再度無効にできます)。

systemd 環境下で起動がおかしいのをデバッグする詳細な情報については、[Freedesktop.org の Diagnosing Boot Problems](https://www.debian.org/bugs/tracking/Freedesktop.org/Diagnosing-Boot-Problems) という記事で参照できます。

³ この機能は、ブートパラメータに `panic=0` を付加することで無効にできます。

4.1.5 アップグレード用の安全な環境の準備

重要: If you are using some VPN services (such as **tinc**) consider that they might not be available throughout the upgrade process. Please see *Prepare for downtime on services*.

リモートでのアップグレード時にさらなる安全マージンを得るため、screen プログラムが提供する仮想コンソール内でアップグレード作業を行うことを提案します。このプログラムは安全な再接続を可能にし、リモート接続プロセスが一時的に切断された場合でもアップグレード作業が中断しないようにしてくれます。

micro-evtd パッケージにより提供される watchdog デーモンのユーザは、アップグレードの前にデーモンを止めて watchdog タイマーを無効化し、アップグレード作業の途中で誤ってリブートが起きないようにすべきです:

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

4.2 "純粹"な Debian からの作業開始

この章で説明しているアップグレードのプロセスは、"純粹"な安定版の Debian システムを想定して書かれています。もし、APT の設定が bookworm 以外で追加のソースを指定している、あるいは他のリリースやサードパーティからパッケージをインストールしている場合、確実にアップグレード作業を遂行するため、事態をややこしくするこれらの要因を取り除くことから始めると良いでしょう。

APT がどのソースからパッケージをダウンロードすべきかを判断するのに使っている主要設定ファイルは /etc/apt/sources.list ですが、/etc/apt/sources.list.d/ ディレクトリ内のファイルを使用することもできます。詳細は [sources.list\(5\)](#) を参照してください。もしシステムで複数の source-list ファイルを使用しているのであれば、設定に一貫性があることを確認する必要があります。

4.2.1 Debian 12 (bookworm) からのアップグレード

12 (bookworm) からのアップグレードのみがサポートされています。Debian のバージョンを表示するには以下を実行します:

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

Please follow the instructions in the Release Notes for Debian 12 at <https://www.debian.org/releases/bookworm/releasenotes> to upgrade to Debian 12 first if needed.

4.2.2 最新のポイントリリースへのアップグレード

This procedure assumes your system has been updated to the latest point release of bookworm. If you have not done this or are unsure, follow the instructions in *bookworm システムのアップグレード*.

4.2.3 Debian Backports

Debian Backports allows users of Debian stable to run more up-to-date versions of packages (with some tradeoffs in testing and security support). The Debian Backports Team maintains a subset of packages from the next Debian release, adjusted and recompiled for usage on the current Debian stable release.

bookworm-backports から取得したパッケージは trixie にあるバージョンよりも小さいバージョン番号なので、ディストリビューションのアップグレードの作業中、"純粋な" bookworm パッケージと同じやり方で trixie へと問題なくアップグレードできるはずですが、今の所は潜在的な問題は特定されていないものの backports 経由のアップグレードはテストが少なく、それに応じてよりリスクがあります。

注意: While regular Debian Backports are supported, there is no clean upgrade path from *sloppy* backports (which use APT source-list entries referencing bookworm-backports-sloppy).

As with *Unofficial sources*, users are advised to remove "bookworm-backports" entries from their APT source-list files before the upgrade. After it is completed, they may consider adding "trixie-backports" (see <https://backports.debian.org/Instructions/>).

For more information, consult the [Backports Wiki page](#).

4.2.4 パッケージデータベースの準備

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 "bookworm" and not to "stable" or "trixie"; see *APT source-list ファイルのチェック*.

4.2.5 利用されなくなったパッケージ

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 Debian 由来でないパッケージを削除する

Debian 由来でないパッケージを見つけるには、以下の apt または apt-forktracer を使った 2 つの手法があります。どちらも 100% 正確ではない点を留意して下さい (例: apt の例では、古いカーネルパッケージのように一度は Debian によって提供されていたが今は提供されていないパッケージを表示します)。

```
$ apt list '?narrow(?installed, ?not(?origin(Debian)))'
$ apt-forktracer | sort
```

4.2.7 残っている設定ファイルを取り除く

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'
```

4.2.8 non-free コンポーネントと non-free-firmware コンポーネント

If you have non-free firmware installed it is recommended to add `non-free-firmware` to your APT sources-list. For details see [アーカイブエリア](#) and *non-free* なファームウェアがアーカイブの独自コンポーネントへ移動しました。

4.2.9 proposed-updates セクション

APT source-list ファイルに `proposed-updates` セクションを含めている場合は、システムのアップグレードを試みる前に、それらのセクションをファイルから削除してください。これは衝突の可能性を減らすための予防策です。

4.2.10 非公式なソース

システムに Debian 以外のパッケージがインストールされている場合、依存関係の衝突のためアップグレード中に削除されるかもしれないことに注意してください。当該パッケージが APT source-list ファイルに Debian 以外のパッケージアーカイブを追加することでインストールされたのなら、そのアーカイブが trixie 用にコンパイルされたパッケージも提供しているかをチェックし、Debian パッケージ用のソース項目と同時にそれも適切に修正してください。

Some users may have *unofficial* backported "newer" versions of packages that *are* in Debian installed on their bookworm system. Such packages are most likely to cause problems during an upgrade as they may result in file conflicts⁴. *Possible issues during upgrade* has some information on how to deal with file conflicts if they should occur.

⁴ Debian のパッケージ管理システムにおいて、別のパッケージを置き換えるように指定されていないパッケージは、通常、別のパッケージの所有ファイルを削除したり置き換えたりすることはできません。

4.2.11 APT の pin 機能を無効にする

特定のパッケージを安定版以外のディストリビューション(テスト版など)からインストールするように APT を設定している場合、そのパッケージが新しい安定版リリース内のバージョンにアップグレードできるように、(/etc/apt/preferences および /etc/apt/preferences.d/ 内に保存されている) APT の pin 設定を変更しなければならないかもしれません。APT の pin 機能に関する、より詳しい情報は、[apt_preferences\(5\)](#) にあります。

4.2.12 gpgv がインストールされているのを確認する

APT は trixie のリリース署名に使われている鍵を照合するのに gpgv バージョン 2 以降を必要とします。gpgv1 でも技術的には依存関係を満たしますが、これは特定の状況下でのみ有用なため、ユーザーは正しいバージョンがインストールされているのを保証したくなることでしょう。以下のコマンドを実施ください:

```
# apt install gpgv
```

4.2.13 パッケージの状態をチェックする

アップグレードの方法に関係なく、まず全パッケージの状態を調べ、全パッケージがアップグレード可能な状態にあるのを確認することをお勧めします。次のコマンドは、インストールが未完了のパッケージ (Half-Installed) や設定に失敗したパッケージ (Failed-Config)、何らかのエラー状態にあるパッケージを表示します:

```
$ dpkg --audit
```

aptitude や次のようなコマンドを使ってシステムの全パッケージの状態を検査することもできます。

```
$ dpkg -l | pager
```

または

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

別の方法としては apt を使うこともできます。

```
# apt list --installed > ~/curr-pkgs.txt
```

アップグレード前に、あらゆる hold 状態を解除しておいたほうがよいでしょう。アップグレードに不可欠なパッケージが hold 状態にある場合、アップグレードに失敗します。

```
$ apt-mark showhold
```

パッケージをローカルで変更・再コンパイルしており、パッケージの名前を変えたりバージョン番号に epoch フィールドを追加していないなら、アップグレードしないよう hold 状態にしておかなければなりません。

apt でパッケージを "hold" 状態に変更するには、以下のように実行してください。

```
# apt-mark hold package_name
```

Replace hold with unhold 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 bookworm as explained in *APT source-list ファイルのチェック*.

4.3 APT source-list ファイルの準備

アップグレードを始める前に、APT の source-list ファイル (/etc/apt/sources.list および /etc/apt/sources.list.d/ 以下のファイル) に |RELEASENAME| を追加し、 |OLDRELEASENAME| を削除する必要があります。

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. "bookworm", "trixie") 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 は、Debian のリリースに関わる関連情報について最新の状態を保つために役立つ 2 つのアナウンス用メーリングリストを提供しています:

- By [subscribing to the Debian announcement mailing list](#), you will receive a notification every time Debian makes a new release. Such as when "trixie" changes from e.g. "testing" to "stable".
- By [subscribing to the Debian security announcement mailing list](#), you will receive a notification every time Debian publishes a security announcement.

4.3.1 APT のインターネットソースの追加

新規インストールではデフォルトはネットワークの条件によってあなたに近いサーバから自動的にパッケージをダウンロードできる Debian APT CDN サービスを使うように APT を設定します。これは比較的新しいサービスのため、古いインストールでは以前としてメインの Debian インターネットサーバのひとつまたはミラーのひとつを設定しているかもしれません。もしまだ設定を行っていない場合、APT 設定において CDN サービスを使うように切り替えることを推奨します。

CDN サービスを利用するには、あなたの APT ソース設定にこのような 1 行を追加してください (main と contrib を使用していると仮定します):

```
deb https://deb.debian.org/debian trixie main contrib
```

新しいソースを追加した後、"deb" 行の先頭に、ハッシュ記号 (#) を追加して無効にしてください。

しかしながら、もしあなたのネットワークの条件から近い特定のミラーを使用して良い結果が得たいならば、このオプションはまだ利用可能です。

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

例えば、一番近くにある Debian ミラーが <http://mirrors.kernel.org> だったとしましょう。このミラーをウェブブラウザで見ると、主なディレクトリが以下のような構成になっていることがわかります。

```
http://mirrors.kernel.org/debian/dists/trixie/main/...
http://mirrors.kernel.org/debian/dists/trixie/contrib/...
```

与えられたミラーを使うよう APT 設定をするには、このような 1 行を追加してください (再度、main と contrib を使用していると仮定します):

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

"dists" を書かなくても、暗黙のうちに追加します。リリース名の後の各引数は、パスの末尾につけて、複数のディレクトリに展開するのに用います。

再度、あなたの新しいソースを追加した後、既存のアーカイブエントリを無効にしてください。

4.3.2 APT のローカルミラーソースの追加

HTTP パッケージミラーを使うのではなく、ローカルディスク (おそらくは NFS マウントされたもの) にあるミラーを使うよう、APT source-list ファイルを変更したいことがあるかもしれません。

例えばパッケージのミラーが `/var/local/debian/` にあり、主なディレクトリの配置が次のようになっているとします。

```
/var/local/debian/dists/trixie/main/...
/var/local/debian/dists/trixie/contrib/...
```

これを `apt` で使うには、次の行を `sources.list` ファイルに追加します。

```
deb file:/var/local/debian trixie main contrib
```

"dists" を書かなくても、暗黙のうちに追加します。リリース名の後の各引数は、パスの末尾につけて、複数のディレクトリに展開するのに用います。

新しいソースを追加した後、APT source-list ファイル内の既存のアーカイブエントリ先頭にハッシュ記号 (#) を追加して無効にしてください。

4.3.3 APT の光学メディアソースの追加

DVD (や CD、Blu-ray ディスク) だけを使いたい場合は、すべての APT source-list ファイル内の既存エントリ先頭にハッシュ記号 (#) を置き、それらを無効にしてください。

CD-ROM ドライブをマウントポイント `/media/cdrom` にマウントできるようにしている行が `/etc/fstab` にあるかどうかを確認してください。例えば `/dev/sr0` が CD-ROM ドライブなら、`/etc/fstab` には次のような行が必要です。

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

第 4 フィールドの `noauto,ro` の単語の間には、スペースを入れてはいけません。

これが正しく機能しているか調べるには、CD を挿入して以下を実行してみてください。

```
# 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
```

問題がなければ

```
# apt-cdrom add
```

を、Debian Binary CD-ROM それぞれに対して実行してください。各 CD に関するデータが APT のデータベースに追加されます。

4.4 パッケージのアップグレード

推奨する方法はパッケージ管理ツール `apt` を使って前の Debian リリースからアップグレードすることです。

注釈: `apt` は対話式な用途を目的としており、スクリプトの中で使うべきではありません。スクリプトの中では字句解析に適していて安定した出力をもつ `apt-get` を使うべきです。

まず、必要なすべてのパーティション (特にルートパーティションと `/usr` パーティション) を `read-write` モードでマウントするのを忘れずに行いましょう。それには以下のようなコマンドを使います。

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

次に、(`/etc/apt/sources.list` や `/etc/apt/sources.list.d/` 以下のファイル内の) APT ソースのエントリが `"trixie"` と `"stable"` のいずれか一方を指定していることを念入りにチェックしてください。bookworm を指し示すソースエントリが含まれてはいけません。

注釈: CD-ROM のソース行は `"unstable"` を指定していることがよくあります。これは混乱の元かもしれませんが、変更すべきではありません*。

4.4.1 セッションの記録

ここで強くお勧めしたいのですが、`/usr/bin/script` プログラムを使って、このアップグレードセッションの記録を取るようにしましょう。こうすれば、何らかの問題が生じたときに何が起きたかを記録しておくことができ、必要に応じてバグ報告に正確な情報を含めることができます。記録を開始するには次のように入力します。

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

to view the file.

アップグレード完了後に `script` を停止するには、プロンプトから `exit` と入力してください。

`apt` は `/var/log/apt/history.log` に変更されたパッケージの状態を、`/var/log/apt/term.log` に端末の出力を記録します。

`script` に `-t` スイッチをつけておいた場合は、以下のように `scriptreplay` プログラムでセッション全体をリプレイできます。

```
# scriptreplay ~/upgrade-trixie-step.time ~/upgrade-trixie-step.script
```

4.4.2 パッケージリストの更新

まず、新しいリリースで利用可能なパッケージの一覧を取得する必要があります。そのためには以下のコマンドを実行してください。

```
# apt update
```

注釈: `apt-secure` のユーザは `aptitude` や `apt-get` を使うと問題を見つけることができるかもしれません。`apt-get` の場合、`apt-get update --allow-releaseinfo-change` を使うことができます。

4.4.3 アップグレードするのに十分な領域があることを確認する

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 *Upgrading the system*. 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` で、インストールに必要なディスク領域の詳細な情報が表示できます。アップグレードを実行する前に、次のように実行して必要な領域の推定値を見ることができます。

```
# 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.
```

注釈: 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 *Minimal system upgrade* before running this command to estimate the disk space.

アップグレードをするのに十分な領域がない場合は、apt が以下のような警告メッセージを出します。

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

この場合、事前に領域を解放するのを忘れないようにしてください。以下のことを実行するとよいでしょう。

- インストールのために、以前 (/var/cache/apt/archives に) ダウンロードしたパッケージを削除する。apt clean を実行してパッケージキャッシュを一掃すると、以前ダウンロードしたパッケージファイルをすべて削除できます。
- 忘れ去られたパッケージを削除する。さらに、bookworm で手作業でパッケージをインストールするのに aptitude や apt を使っていたのなら、手作業でインストールされたパッケージの記録が取られています。依存関係のみによって引きずられてインストールされたパッケージに対して、依存元パッケージが削除されたためにもう不要となった場合に、余分だというマークをつけることができます。手作業でインストールしたパッケージには削除されるマークをつけません。自動的にインストールされたがもはや使われていないパッケージを削除するには、以下を実行してください:

```
# apt autoremove
```

余分なパッケージを見つけるのに、deborphan や debfoster、cruft も使えます。これらのツールが表示したパッケージをやみくもに削除しないでください。特に、誤検出しやすい非デフォルトの凶暴なオプションを使っている場合はなおさらです。実際に削除する前に、削除を提案されたパッケージ(の内容やサイズ、説明など)を手作業で調べなおすことを強くお勧めします。

- 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.
- 翻訳や地域化用ファイルが不要なら、それらをシステムから削除する。**localepurge** パッケージをインストールして設定すれば、選んだ少数のロケールのみがシステムに残るようにすることが可能です。これによって、`/usr/share/locale` の消費するディスク領域を減らせるでしょう。
- `/var/log/` の下にあるシステムログを、一時的に他のシステムに移動するか、永久に削除する。

- 仮設の `/var/cache/apt/archives` を使用する。すなわち、別のファイルシステム (USB ストレージデバイス、一時的なハードディスク、既に使用されているファイルシステムなど) を仮設のキャッシュディレクトリとして拝借することができます。

注釈: アップグレード中にネットワーク接続が途切れる可能性があるので、NFS マウントは使用しないでください。

以下は、`/media/usbkey` にマウントされた USB ドライブがある場合を例とします。

1. 今までに、インストールのためにダウンロードされたパッケージを削除します。

```
# apt clean
```

2. ディレクトリ `/var/cache/apt/archives` を、USB ドライブにコピーします。

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

3. 現在のキャッシュディレクトリに、仮のキャッシュディレクトリをマウントします。

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

4. アップグレード後に、元々の `/var/cache/apt/archives` ディレクトリを復活させます。

```
# umount /var/cache/apt/archives
```

5. 残っている `/media/usbkey/archives` を削除します。

仮設のキャッシュディレクトリは、システムにマウントされているファイルシステムであれば何にでも作成できます。

- Do a minimal upgrade of the system (see *Minimal system upgrade*) 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.

パッケージを安全に削除するための注意として、[APT source-list ファイルのチェック](#) で説明するように、APT source-list ファイルが `bookworm` を指し示すよう設定を戻しておくことが望ましいです。

4.4.4 監視システムの停止

`apt` があなたのコンピュータで動作しているサービスを一時的に停止する必要があるかもしれないため、アップグレードの最中に終了された他のサービスを再起動できるような監視用サービスを予め停止しておくのが良い考えでしょう。Debian では、`monit` がそのようなサービスの例です。

4.4.5 システムの最小アップグレード

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 *Upgrading the system*.

これをまず行うには、以下のコマンドを実行してください。

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

このコマンドには、アップグレードしても他のパッケージをインストール・削除する必要がないパッケージだけをアップグレードする、という効果があります。

システムの容量が少なく、容量による制約のため完全アップグレードが実行できない場合にも、システムの最小アップグレードは有用です。

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.

4.4.6 システムのアップグレード

これまでの手順を実行し終わったら、アップグレードの主要な部分が続ける準備ができています。以下のコマンドを実行してください。

```
# apt full-upgrade
```

これによってシステムの完全なアップグレードが行われ、すべてのパッケージの最新版がインストールされ、リリース間で発生しうるパッケージの依存関係の変化すべてが解決されます。必要に応じて、新しいパッケージ (通常は、新しいバージョンのライブラリや、名前が変わったパッケージ) がインストールされたり、衝突した古いパッケージが削除されたりもします。

CD/DVD/BD のセットからアップグレードする場合には、アップグレードの最中に、おそらく特定のディスクを入れるよう何回か指示されることになるでしょう。同じディスクを複数回入れなければならないかもしれませんが、これは、相互に依存しているパッケージが別々のディスクに分散しているためです。

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 アップグレード中の注意点

以下の章では、trixie へのアップグレードの最中に現れるかもしれない既知の問題を記述しています

4.5.1 「即時設定は動作しません」で full-upgrade が失敗する

apt full-upgrade の途中でパッケージをダウンロードした後に失敗となり、

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

と表示することがあります。これが起きた場合は、代わりに apt full-upgrade -o APT::Immediate-Configure=0 を実行することでアップグレードを進められるはずです。

この問題の暫定的な別の対処の可能性として、bookworm と trixie の両方のソースを一時的に sources.list に追加して apt update を実行する方法があります。

4.5.2 予期されるパッケージの削除

The upgrade process to trixie 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 trixie, see *Obsolete packages*.

4.5.3 衝突 (Conflicts) あるいは事前依存 (Pre-Depends) のループ

場合によっては衝突や事前依存のループのために、APT の APT::Force-LoopBreak オプションを有効にして、必須パッケージを一時的に削除しなければならないかもしれません。その場合 apt はこのことを警告してアップグレードを中断します。apt のコマンドラインにオプション -o APT::Force-LoopBreak=1 を指定すれば、この状態を回避できます。

システムの依存関係の構造があまりに問題だらけで、手動での介入が必要となることもあります。通常、手動での介入とは、apt を用いるか、あるいは

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

で問題の原因となるパッケージを消す作業になります。または次の方法を用いてもよいかもしれません。

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

極端な場合には、コマンドラインから次のように入力して、再インストールしなければならないかもしれません。

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

4.5.4 ファイルの衝突

"純粹" な bookworm システムからのアップグレードでは、ファイルの衝突は起こらないはずですが、非公式のバックポートパッケージをインストールしているなら起こるかもしれません。ファイルの競合が起こると、次のようなエラーになります:

```
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
```

問題が修正できたら、先程説明した apt コマンドを再度入力すれば、アップグレードを再開できます。

4.5.5 設定の変更

アップグレードの最中に、いくつかのパッケージの設定・再設定に関する質問が表示されます。/etc/init.d ディレクトリと /etc/manpath.config ファイルに関しては、パッケージメンテナのバージョンに置き換えるようにしてください。システムの整合性を保つためには "yes" と答えることが必要になります。古いバージョンも .dpkg-old という拡張子をつけられて保存されていますので、戻すのはいつでもできます。

どうすればよいかわからなくなったら、そのパッケージやファイルの名前を書き留めておいて、その問題解決は後回しにしましょう。typescript ファイルを検索すれば、アップグレードの最中に画面に表示された情報をもう一度見ることもできます。

4.5.6 コンソール接続へセッションの変更

システムのローカルコンソールを使ってアップグレードを実行している場合、アップグレードの最中に何回かコンソールが別の画面へ移動してしまい、アップグレード作業が見えなくなることにも気づくかもしれません。例えば、グラフィカルインターフェイスがあるシステムではディスプレイマネージャが再起動した際に起こります。

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 カーネルと関連パッケージのアップグレード

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.

このセクションに書かれている多くの情報は、ユーザが Debian のモジュラーカーネルのいずれかを **initramfs-tools** や **udev** とともに使用しているのを前提にしている、ということに注意してください。initrd を必要としないカスタムカーネルを使用するを選択した場合や、initrd 生成ユーティリティとして異なるものを使用している場合は、このセクションの情報の一部は適切ではないかもしれません。

4.6.1 カーネルメタパッケージのインストール

When you full-upgrade from bookworm to trixie, 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
```

何も出力されない場合は、新しい **linux-image** パッケージを手作業でインストールするか、**linux-image** メタパッケージをインストールする必要があります。利用可能な **linux-image** メタパッケージの一覧を見るには次のように実行してください:

```
$ 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 [アップグレード後、再起動前にすること](#) before performing the first reboot after the upgrade.

少し勇気のある人には、Debian 上で簡単に自分のカスタムカーネルをコンパイルするやり方があります。**linux-source** パッケージで提供されるカーネルソースをインストールしてください。バイナリパッケージの構築には、ソース中の Makefile 中の `deb-pkg` ターゲットが使えます。さらなる情報は、[Debian Linux カーネルハンドブック](#) にあります。**debian-kernel-handbook** パッケージでも利用できます。

可能であればカーネルパッケージのアップグレードをメインの `full-upgrade` と分けることで、一時的にでも起動不能なシステムにしてしまうことを極力避けられます。カーネルパッケージのアップグレードは、[Minimal system upgrade](#) で説明した最小アップグレードの手順の後以外では行うべきでないことに注意してください。

4.7 次のリリースへの準備

アップグレードの後で、次のリリースに向けてできるいくつかの準備があります。

- Remove newly redundant or obsolete packages as described in *Make sure you have sufficient space for the upgrade* and *Obsolete packages*. You should review which configuration files they use and consider purging the packages to remove their configuration files. See also *Purging removed packages*.

4.7.1 削除したパッケージを完全削除する

一般的に、削除したパッケージを完全に削除 (purge) するのは賢明なことです。以前のリリースアップグレード (つまりは bookworm へのアップグレード) の際に削除されているパッケージである、あるいはパッケージがサードパーティベンダーから提供されたものである場合、尚のこととなります。特に、古い init.d スクリプトは問題を起すことが知られています。

注意: パッケージの完全削除 (purge) は通常ログファイルについても完全に削除を行うので、まずはこのバックアップを行ったほうが良いでしょう。

以下のコマンドは、設定ファイルをシステムに残して削除されたパッケージの一覧を (もしあれば) 表示します:

```
$ apt list '~c'
```

apt purge を実行すればパッケージを削除できます。一度でこれらのパッケージを削除したい場合は、以下のコマンドで実施できます:

```
# apt purge '~c'
```

4.8 利用されなくなったパッケージ

trixie では大量の新規パッケージが導入された一方で、bookworm に存在していた非常の少量の古いパッケージの破棄や削除が行われています。これら時代遅れのパッケージをアップグレードする手段は提供されていません。時代遅れのパッケージを使い続けても構いませんが、Debian プロジェクトでは通常、trixie がリリースされてから 1 年後にセキュリティサポートを終了します⁵。そして、その時点から他のサポートも提供しません。利用可能な代替手段で置き換えられるのであれば、そうすることをお勧めします。

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 trixie due to bugs in them. In the latter case, packages might still be present in the "unstable" distribution.

"Obsolete and Locally Created Packages" can be listed and purged from the commandline with:

⁵ あるいは、1 年以内でも別のリリースが出るときに。一般に、どの時点でも、サポートされる安定版リリースは 2 つだけです。

```
$ apt list '~o'  
# apt purge '~o'
```

Debian バグ追跡システムは、パッケージが削除された理由についての追加情報を提供してくれることがよくあります。そのパッケージ自体と ftp.debian.org 擬似パッケージの両方の、アーカイブ化されたバグ報告を調べてください。

trixie での廃止パッケージ一覧については、[特記すべき廃止されたパッケージたち](#) を参照して下さい。

4.8.1 移行用ダミーパッケージ

bookworm からのいくつかのパッケージは trixie においてアップグレードを簡単にできるよう設計された空の代用品である移行用ダミーパッケージによって置き換えられるかもしれません。以前は1つのパッケージであったアプリケーションがいくつかのパッケージに分割された場合、移行用パッケージは古いパッケージと同じ名前で、新しいパッケージをインストールするための適切な依存関係をもって提供されるかもしれません。これをインストールした後は冗長なダミーパッケージを安全に削除できます。

(すべてではないのですが) 大半のダミーパッケージの説明には、その目的が記されています。しかしダミーパッケージの説明文は統一されておらず、パッケージをインストールするためや、最新のバージョンを追跡するために"ダミー"パッケージがインストールされたままとなるように設計されているものもあります。自分のシステム内のダミーパッケージを検出するには、`deborphan` を `--guess-*` オプションつきで使うのが便利でしょう (例: `--guess-dummy`)。一部のダミーパッケージは、アップグレード後に削除されることを意図したものではなく、プログラムのどのバージョンが現在利用可能な最新版かを長期にわたって追跡するのに使われる、ということに注意してください。

第5章 trixie で注意すべき点

新しいリリースで導入された変更点には副作用が避けられず、どこか他の場所でバグを出してしまうことがあります。この章では、現時点で私たちが知っている問題点を記載しています。正誤表・関連パッケージの付属文書・バグ報告や、[もっと読みたい](#)で触れられているその他の情報も読んでください。

5.1 trixie へのアップグレード特有の事項

この項では bookworm から trixie へのアップグレードに関連した項目を取り扱います。

5.1.1 non-free なファームウェアがアーカイブの独自コンポーネントへ移動しました

[アーカイブエリア](#) で説明したように、non-free なファームウェアパッケージは non-free-firmware と呼ばれる独自のアーカイブコンポーネントから提供されるようになりました。インストール済みの non-free なファームウェアパッケージがきちんとアップグレードされるのを保証するため、APT の設定変更が必要となっています。ファームウェアをインストールするのに APT sources-list に ``non-free`` コンポーネントだけが追加されている場合は、更新した APT sources-list の記述は以下の様になります：

```
deb https://deb.debian.org/debian bookworm main non-free-firmware
```

apt によってこの章を案内された場合、`/etc/apt/apt.conf.d/no-bookworm-firmware.conf` という名前の `apt.conf(5)` ファイルを以下の内容で作成することで、この変更について何度も通知されるのを防げます。

```
APT::Get::Update::SourceListWarnings::NonFreeFirmware "false";
```

5.1.2 システムクロックの設定をするパッケージについての変更

Network Time Protocol (NTP) サーバーからシステムクロックを設定するデフォルトの手法だった `ntp` パッケージは `ntpsec` に置き換えられました。

`ntp` から `ntpsec` への移行について、ほとんどのユーザーは何も特別な作業をする必要はありません。

bookworm では、他にも同様のサービスを提供する複数のパッケージがあります。Debian でのデフォルトは `systemd-timesyncd` となっていますが、これは `ntp` クライアントをクロック設定の用途のみに必要なユーザーには十分なものでしょう。bookworm は、NTP サーバーの運用など、より高度な機能をサポートする `chrony` や `openntpd` も含んでいます。

5.1.3 設定管理システム puppet はバージョン 7 にアップグレードされました

Puppet はバージョン 5 から 7 へとアップグレードされ、Puppet 6 シリーズは完全にスキップされました。これは Puppet のエコシステムに大きな変更をもたらします。

クラシックな Ruby ベースの Puppet Master 5.5.x アプリケーションは、upstream によって非推奨にされ、Debian では利用できなくなりました。これは Puppet Server 7.x へと置き換えられ、**puppetserver** パッケージで提供されています。このパッケージは移行用の **puppet-master** パッケージの依存関係によって自動的にインストールされます。

いくつかのケースでは Puppet サーバーは Puppet マスターの完全互換品となりますが、新しいデフォルト値がデプロイ環境に適合するのを確認するため、`/etc/puppet/puppetserver` 以下にある設定ファイルをチェックすべきでしょう。特に `auth.conf` ファイルでのレガシーな形式は非推奨となっています。詳細は [auth.conf documentation](#) を参照してください。

推奨方法はクライアントの前にサーバーをアップグレードすることです。Puppet 7 サーバーは過去のクライアントと後方互換性を持っています。Puppet 5 サーバーはアップグレードされたエージェントを扱いつづけることはできますが、新規の Puppet 7 エージェントを登録できません。つまり、サーバーをアップグレードする前に新規に Puppet 7 エージェントをデプロイすると、フリートにこのエージェントを追加できなくなってしまいます。

puppet パッケージは **puppet-agent** パッケージによって置き換えられ、スムーズなアップグレードを保証するために移行パッケージとなっています。

最後に、**puppetdb** パッケージは `bullseye` で削除されましたが `bookworm` で再導入されています。

5.1.4 youtube-dl は yt-dlp にリプレースされました

広く様々なウェブサイト (YouTube を含みますがこれに限りません) からビデオをダウンロードできることで人気のツール、**youtube-dl** は Debian に含まれなくなりました。その代わりに、**yt-dlp** を引っ張ってくる中身が空の移行パッケージとしてリプレースされています。**yt-dlp** は **youtube-dl** のフォークで、現在新規の開発が行われています。

完全な互換を持つラッパーは提供されていないので、スクリプトや自身の挙動を `youtube-dl` ではなく `yt-dlp` を呼び出すように調整する必要があります。機能はほぼ同じですが、オプションの一部や細かな挙動が変更されています。詳細は `yt-dlp` の `url-man-stable:yt-dlp.1`、特に `yt-dlp.1.html#Differences_in_default_behavior` セクションを必ず確認してください。

5.1.5 fcitx は複数バージョンをインストールできなくなりました

fcitx そして **fcitx5** パッケージは、人気の Fcix インпутメソッドフレームワークのバージョン 4 およびバージョン 5 を提供しています。upstream からのアドバイスに従い、同一のオペレーティングシステム上にこれらを同時にインストールできなくなりました。以前に **fcitx** と **fcitx5** を同時にインストールしてあった場合、ユーザーはどちらのバージョンの Fcix をキープするかを決める必要があります。

アップグレード作業の前に、ユーザーは希望しない Fcix バージョンの関連パッケージ全て (Fcix 4 の場合は `fcitx-*`、Fcix 5 は `fcitx5-*`) の `purge` を強く推奨します。アップグレード完了後、選択した希望のインพุットメソッドフレームワークがシステムで使われるよう、`im-config` の再実行を検討してください。

さらなる背景情報を [メーリングリストへ投稿されたアナウンス](#) (簡体中国語です) を読むことができます。

5.1.6 MariaDB パッケージはパッケージ名にバージョン番号を含まなくなりました

bullseye では MariaDB バージョンがパッケージ名に含まれていましたが (例: `mariadb-server-10.5` や `mariadb-client-10.5`)、bookworm では MariaDB 10.11 同等のパッケージ名は完全にバージョンが無いものとなります (例: `mariadb-server` または `mariadb-client`)。MariaDB バージョンはパッケージバージョンのメタデータで確認できます。

バージョン番号なしのパッケージ名への移行が失敗すると判明しているアップグレード時の流れの一つ (Bug #1035949) は以下です

```
# apt-get install default-mysql-server
```

`mariadb-admin` コマンドを使う MariaDB サーバーの SysV init サービスがシャットダウンを実行する前に `mariadb-client-10.5` 及びパッケージ内の `/usr/bin/mariadb-admin` ファイルが削除された場合に失敗することになります。回避策は以下を実行します。

```
# apt upgrade
```

を実行する前に以下を実行します

```
# apt full-upgrade
```

以上です。

MariaDB でのパッケージ名の変更の詳細な情報については、</usr/share/doc/mariadb-server/NEWS.Debian.gz> を参照してください。

5.1.7 システムのロギング設定への変更

`rsyslog` パッケージはほとんどのシステムで不要になっており、削除しても構いません。

多くのプログラムが何をしているかをユーザーに知らせるためにログメッセージを生成します。これらのメッセージは、`systemd` の "journal" が `rsyslog` のような "syslog デモン" によって取り扱えます。

bookworm では、デフォルトで `rsyslog` がインストールされ、`systemd` ジャーナルは `/var/log/syslog` のような様々なテキストファイルにメッセージを書き込む `rsyslog` にログメッセージを転送するように設定されていました。

trixie より `rsyslog` はデフォルトではインストールされなくなりました。 `rsyslog` を使い続けたくない場合、アップグレード後に自動的にインストールされたというマークを以下のコマンドで付けることができます。

```
# apt-mark auto rsyslog
```

を実施、それから可能な場合は以下で削除を実行します

```
# apt autoremove
```

さらに以前の Debian のリリースからアップグレードしていて、デフォルトの設定を受け入れていなかった場合、ジャーナルは永続ストレージにメッセージを保存するようには設定されていません: [journal.conf\(5\)](#) に有効にする指示手順があります。

`rsyslog` から離れるのを決めた場合ですが、`journalctl` コマンドを使えば `/var/log/journal` にバイナリ形式で保存されているログメッセージを読むことができます。

```
# journalctl -e
```

上記のコマンドは、ジャーナル内の最も直近のログメッセージを表示します

```
# journalctl -ef
```

このコマンドは書き込みされた新しいメッセージを表示します (同様のコマンドを実行する場合は以下のようになります)

```
# tail -f /var/log/syslog
```

).

5.1.8 `rsyslog` での変更が `logcheck` のようなログ分析ソフトに影響を与えます

`rsyslog` は"高精度タイムスタンプ"の利用をデフォルトにしましたが、この変更はシステムログを分析する他のプログラムに影響を与える可能性があります。この設定をどのように調整するかについての詳細な情報が [rsyslog.conf\(5\)](#) にあります。

タイムスタンプの変更は、ローカルで作成された `logcheck` ルールの更新が必要になる場合があります。`logcheck` は、ルールとして知られている正規表現データベースに照らし合わせて (`systemd-journald` または `rsyslog` が生成した) システムログ内のメッセージをチェックします。メッセージが生成された時間にマッチするルールは、新しい `rsyslog` 形式にマッチするように更新する必要があります。`logcheck-database` パッケージによって提供されているデフォルトのルールは更新されていますが、ローカルで作成したなどの他のルールについては新しい形式を認識するように更新する必要があるかもしれません。ローカルの `logcheck` ルールを更新するのに役立つスクリプトについては [/usr/share/doc/logcheck-database/NEWS.Debian.gz](#) を参照してください。

5.1.9 `rsyslog` が生成するログファイルが少なくなっています

`rsyslog` が生成するログファイルについて変更を行ったので、`/var/log` 以下のいくつかのファイルは削除しても構わなくなりました。

`rsyslog` を使い続ける場合 ([Changes to system logging](#) 参照)、`/var/log` 内のログファイルのいくつかはデフォルトでは生成されなくなります。これらのファイルに書き込まれていたメッセージは `/var/log/syslog` にもありますが、デフォルトでは生成されなくなっています。これらのファイルに書き込まれていたものはすべて、`/var/log/syslog` 以下から入手可能です。

作成されなくなったファイル一覧:

- `/var/log/mail.{info,warn,err}`

これらのファイルは、ローカルのメール転送エージェント (MTA) からのメッセージを含み、プライオリティによって分割されていました。

`/var/log/mail.log` がメール関連のメッセージを全て含んでいるので、これらのファイル (およびローテーションされたコピー) は安全に削除できます。このファイル群をアノマリー検知に利用していた場合ですが、適切な代替は `logcheck` などになるでしょう。

- `/var/log/lpr.log`

これらのファイルには印刷に関連するログメッセージが含まれていました。Debian でのデフォルトの印刷システム `cups` はこのファイルを使わないので、他の印刷システムをインストールしているのではない限り、このファイル (およびローテーションされたコピー) は削除できます。

- `/var/log/{messages,debug,daemon.log}`

これらのファイル (およびローテーションされたコピー) は削除して構いません。ここに書き込まれていたものはすべて `/var/log/syslog` 以下にあります。

5.1.10 slapd のアップグレードには手動での介入が必要になる可能性があります

OpenLDAP 2.5 は新規のメジャーリリースで、[upstream のリリースアナウンス](#) に記述されているように複数の非互換性を含んでいます。設定によりませんが、`slapd` サービスはアップグレード後に必要な設定更新が完了するまで停止し続ける可能性があります。

以下が既知となっている互換性のない変更です:

- `slapd-bdb(5)` および `slapd-hdb(5)` データベースバックエンドは削除されました。bookworm でこれらのバックエンドのうちの一つを利用していた場合、trixie への*アップグレード前に* `slapd-mdb(5)` への移行を強く推奨します。
- `slapd-shell(5)` データベースバックエンドは削除されました。
- `slapo-ppolicy(5)` オーバーレイはスキーマをモジュールに内包するようになりました。以前の外部スキーマが存在していると、新しい内蔵スキーマとコンフリクトを起こします。
- The contrib password module `pw-argon2` (see `slapd-pw-argon2.5`) has been renamed to `argon2`.

アップグレードの完了と `slapd` サービスの再開に関する作業手順は、[/usr/share/doc/slapd/README.Debian.gz](#) で確認できます。さらに [upstream のアップグレードに関する記述](#) についても調べておくのが良いでしょう。

5.1.11 GRUB はデフォルトでは os-prober を実行しなくなりました

長らく `grub` はコンピューターにインストールされている他のオペレーティングシステム (OS) を検出するのに `os-prober` パッケージを利用しており、検出した OS を起動メニューに追加できていました。残念なことにこの機能は特定のケース (例: ゲストバーチャルマシンが動作している状態) で問題になるため、最新の upstream でのリリースでデフォルトでは無効にされました。

システムを起動するのに GRUB を使っていて、他の OS を起動メニューにそのまま載せておきたい場合は、この挙動を変更できます。 `/etc/default/grub` ファイルを編集し、設定が `GRUB_DISABLE_OS_PROBER=false` となっていることを確認して `update-grub` を実行する、あるいは

```
# dpkg-reconfigure <GRUB_PACKAGE>
```

を実行して、この設定やその他の GRUB の設定をよりユーザーフレンドリーなやり方で変更します。

5.1.12 GNOME はスクリーンリーダーへのアクセシビリティサポートを縮小しています

数多くの GNOME アプリケーションが GTK3 グラフィックスツールキットから GTK4 へ切り替えをしました。残念なことに、多くのアプリケーションが `orca` のようなスクリーンリーダーでとても使いづらくなっています。

あなたがスクリーンリーダーに頼っている場合は、よりマシなアクセシビリティサポートがある `Mate` などの他のデスクトップ環境への切り替えを検討する必要があるでしょう。この切り替えを行うには `mate-desktop-environment` パッケージのインストールを行います。Mate 環境での Orca の使い方についての情報は `こちら` になります。

5.1.13 Baseline for 32-bit PC is now i686

Debian's support for 32-bit PC (known as the Debian architecture i386) now no longer covers any i586 processor. The new minimum requirement is i686. This means that the i386 architecture now requires the "long NOP" (NOPL) instruction, while bullseye still supported some i586 processors without that instruction (e.g. the "AMD Geode").

あなたのマシンがこの要求に適合しない場合、サポートのライフサイクルが存在する間は bullseye に留まることをお勧めします。

5.1.14 polkit 設定についての変更点

upstream や他のディストリビューションとの整合性のため、非特権プログラムが特権システムサービスにアクセスするのを許可する `polkit` サービス (以前は PolicyKit) は、ローカルのポリシールールについて記述方法と配置場所を変更しました。これからは、セキュリティポリシーのカスタマイズするローカルルールを JavaScript で記述し、 `/etc/polkit-1/rules.d/*.rules` に配置する必要があります。新しい形式を使ったルールの例は `/usr/share/doc/polkitd/examples/` で、詳細は `polkit.8.html#AUTHORIZATION_RULES` で参照できます。

これまで、ルールを `pkla` 形式で記述し、`/etc/polkit-1/localauthority` または `/var/lib/polkit-1/localauthority` 以下のサブディレクトリに配置できました。しかし、`.pkla` ファイルは非推奨 (deprecated) として扱われるようになり、**polkitd-pkla** パッケージがインストールされている場合のみ動作しつづけることができます。このパッケージは、通常 `bookworm` にアップグレードする際に自動的にインストールされますが、将来の Debian リリースには含まれなくなると考えられるため、上書きするローカルポリシーは JavaScript 形式に移行する必要があるでしょう。

5.1.15 "merged-/usr" が必須となりました

Debian は "merged-/usr" と呼ばれるファイルシステムレイアウトを採用し、従来のディレクトリである `/bin` `/sbin`, `/lib` そして `/lib64` のようなオプションを含まなくなりました。新しいレイアウトでは、従来のディレクトリはそれぞれ対応する場所である `/usr/bin`, `/usr/sbin`, `/usr/lib`, and `/usr/lib64` に置き換えられています。つまり、例えば `/bin/bash` でも `/usr/bin/bash` でも `bash` を起動することになります。

`buster` または `bullseye` としてインストールされたシステムでは、新しいファイルシステムレイアウトがこれらのリリースでは既にデフォルトとなっていたので、何も変更はありません。しかし、`bookworm` へのアップグレードした時に

The conversion to the new layout should have no impact on most users. All files are automatically moved to their new locations even if they were installed locally or come from packages not provided by Debian, and hardcoded paths such as `/bin/sh` continue to work. There are, however, some potential issues:

- `# dpkg --search`

このコマンドは新しい位置に移動したファイルについて誤った答えを出します:

```
# dpkg --search /usr/bin/bash
```

上記は `bash` がどのパッケージからきたのか判別できません (ですが、以下は今までどおりで期待通りに動作します。)

```
# dpkg --search /bin/bash
```

still works as expected.)

- Local software not provided by Debian may not support the new layout and may, for example, rely on `/usr/bin/name` and `/bin/name` being two different files. This is not supported on merged systems (including new installations since `buster`), so any such software must be fixed or removed before the upgrade.
- Systems that rely on a "base layer" that is not directly writable (such as [WSL1](#) images or container systems using multi-layer [overlayfs](#) filesystems) cannot be safely converted and should either be replaced (e.g., by installing a new `WSL1` image from the store) or have each individual layer upgraded (e.g., by upgrading the base Debian layer of the `overlayfs` independently) rather than `dist-upgraded`.

For further information, see [The Case for the /usr merge](#) and [the Debian Technical Committee resolution](#).

5.1.16 buster からのサポート外のアップグレードが libcrypt1 で失敗する

Debian は公式には安定版リリースからその次の安定版リリースへのアップグレードのみをサポートしています。つまり、bullseye から bookworm に、ということです。buster から bookworm へのアップグレードはサポートしておらず、Bug #993755 のため以下のようなエラーで失敗することになります:

```
Setting up libc6:ARCH (2.36-9) ...
/usr/bin/perl: error while loading shared libraries: libcrypt.so.1: cannot open
↳shared object file: No such file or directory
dpkg: error processing package libc6:ARCH (--configure):
installed libc6:ARCH package post-installation script subprocess returned error exit
↳status 127
```

このような特定の状況からではありますが、強制的に新しい **libcrypt1** をインストールすることで手動でリカバリ可能となります:

```
# cd $(mktemp -d)
# apt download libcrypt1
# dpkg-deb -x libcrypt1_*.deb .
# cp -ra lib/* /lib/
# apt --fix-broken install
```

5.1.17 openssh-server no longer reads ~/.pam_environment

The Secure Shell (SSH) daemon provided in the **openssh-server** package, which allows logins from remote systems, no longer reads the user's `~/.pam_environment` file by default; this feature has a [history of security problems](#) and has been deprecated in current versions of the Pluggable Authentication Modules (PAM) library. If you used this feature, you should switch from setting variables in `~/.pam_environment` to setting them in your shell initialization files (e.g. `~/.bash_profile` or `~/.bashrc`) or some other similar mechanism instead.

Existing SSH connections will not be affected, but new connections may behave differently after the upgrade. If you are upgrading remotely, it is normally a good idea to ensure that you have some other way to log into the system before starting the upgrade; see [復旧の準備](#).

5.1.18 OpenSSH no longer supports DSA keys

Digital Signature Algorithm (DSA) keys, as specified in the Secure Shell (SSH) protocol, are inherently weak: they are limited to 160-bit private keys and the SHA-1 digest. The SSH implementation provided by the **openssh-client** and **openssh-server** packages has disabled support for DSA keys by default since OpenSSH 7.0p1 in 2015, released with Debian 9 ("stretch"), although it could still be enabled using the `HostKeyAlgorithms` and `PubkeyAcceptedAlgorithms` configuration options for host and user keys respectively.

The only remaining uses of DSA at this point should be connecting to some very old devices. For all other purposes, the other key types supported by OpenSSH (RSA, ECDSA, and Ed25519) are superior.

As of OpenSSH 9.8p1 in trixie, DSA keys are no longer supported even with the above configuration options. If you have a device that you can only connect to using DSA, then you can use the `ssh1` command provided by the `openssh-client-ssh1` package to do so.

In the unlikely event that you are still using DSA keys to connect to a Debian server (if you are unsure, you can check by adding the `-v` option to the `ssh` command line you use to connect to that server and looking for the "Server accepts key:" line), then you must generate replacement keys before upgrading. For example, to generate a new Ed25519 key and enable logins to a server using it, run this on the client, replacing `username@server` with the appropriate user and host names:

```
$ ssh-keygen -t ed25519
$ ssh-copy-id username@server
```

5.2 アップグレード後、再起動前にすること

`apt full-upgrade` が完了した時点で、"正規"のアップグレードは完了しています。trixie へのアップグレードについては、再起動の実行前に必要となる特別な作業はありません。

5.2.1 アップグレード後も影響がある項目

5.2.2 セキュリティサポートにおける制限事項

Debian がセキュリティ問題に対する最小限のバックポートを約束できないパッケージがいくつか存在しています。これらについては以下の章で触れられています。

注釈: `debian-security-support` パッケージが、インストールされたパッケージのセキュリティサポート状況を確認するのに役立ちます。

ウェブブラウザとそのレンダリングエンジンにおけるセキュリティ更新の状態

Debian 13 は複数のブラウザエンジンを含んでおり、これらは一定の割合でセキュリティ脆弱性の影響を受けます。高い脆弱性率と長期プランチ形式での upstream でのサポートが限定的なことによって、セキュリティ修正をバックポートしてこれらのブラウザならびにブラウザエンジンをサポートする事が難しくなっています。さらに、ライブラリとの相互依存性のため、開発元での新しいリリースへの更新を極めて難しくしています。`webkit2gtk` ソースパッケージを使ったアプリケーション (例: `epiphany`) はセキュリティサポートの対象ですが、`qtwebkit` (`qtwebkit-opensource-src` ソースパッケージ) を使っているアプリケーションはセキュリティサポートの対象外です。

一般的なウェブブラウザ利用として我々は Firefox または Chromium を推奨しています。安定版向けに現行の ESR リリースをリビルドすることで最新を維持します。同じ手法が Thunderbird にも適用されます。

一旦リリースが `oldstable` となると、公式サポート対象のブラウザは標準的な保証期間の更新を受け続けられないかもしれません。例えば、Chromium は `oldstable` では通常の 12 ヶ月ではなく 6 ヶ月のセキュリティサポートのみを受けます。

Go および Rust 言語ベースのパッケージ

現在、Debian のインフラは静的リンクを行うパッケージをリビルドすることに問題を抱えています。Go および Rust のエコシステムの成長に伴い、インフラが強化されメンテナンスが行き届くようになるまでは、限定的なセキュリティサポートとなります。

多くの場合ですが Go あるいは Rust の開発用ライブラリへの更新は、定期的なポイントリリースでのみ提供されます。

5.2.3 python インタプリタは `externally-managed` としてマークされます

Debian が提供する `python3` インタプリタパッケージ (`python3.11` および `pypy3`) は、[PEP-668](#) に従って `externally-managed` とマークされるようになりました。Debian で提供されるバージョンの `python3-pip` はこれに伴って、`--break-system-packages` オプションが指定されない限り、Debian の `python` インタプリタ環境へ手作業でのパッケージインストールを拒否します。

Debian ではパッケージ化されていない Python アプリケーション (あるいはバージョン) をインストールする必要がある場合、(`pipx` Debian パッケージにある) `pipx` を使ったインストールを推奨します。`pipx` は他のアプリケーションやシステムの Python モジュールから隔離された環境をセットアップし、アプリケーションとその依存関係をその環境へインストールします。

Debian ではパッケージ化されていない Python ライブラリモジュール (あるいはバージョン) をインストールする必要がある場合、可能であれば `virtualenv` へインストールするのを推奨しています。`virtualenv` は Python 標準ライブラリモジュールの `venv` (`python3-venv` Debian パッケージ内)、あるいは Python サードパーティツールの `virtualenv` (`virtualenv` Debian パッケージにあります) で作成できます。例えば、`pip install --user foo` を実行するのではなく、任意の `virtualenv` にインストールするのに `mkdir -p ~/.venvs && python3 -m venv ~/.venvs/foo && ~/.venvs/foo/bin/python -m pip install foo` を実行します。

詳細については `/usr/share/doc/python3.11/README.venv` を参照してください。

5.2.4 VLC でのハードウェアアクセラレーションを利用したビデオエンコードとデコードのサポートは制限されています

VLC ビデオプレイヤーは、`VA-API` と `VDPAU` を使ってハードウェアアクセラレーションを有効にしたビデオエンコーディング・デコーディングをサポートしています。しかし、VLC の `VA-API` サポートは `FFmpeg` のバージョンと強く結びついてしまっています。`FFmpeg` が 5.x ブランチにアップグレードされたので、VLC の `VA-API` サポートは無効にされています。ネイティブな `VA-API` サポートを持つ GPU (Intel や AMD の GPU) のユーザーは、ビデオの再生やエンコーディング時に高い CPU 負荷に遭遇するかもしれません。

ネイティブな `VDPAU` サポートを提供している GPU (non-free なドライバを利用している NVIDIA) はこの問題の影響を受けません。

Support for VA-API and VDPAU can be checked with `vainfo` and `vdpauintfo` (each provided in a Debian package of the same name).

5.2.5 `systemd-resolved` は分割されて独立したパッケージになりました

The new `systemd-resolved` package will not be installed automatically on upgrades. If you were using the `systemd-resolved` system service, please install the new package manually after the upgrade, and note that until it has been installed, DNS resolution might no longer work since the service will not be present on the system. Installing this package will automatically give `systemd-resolved` control of `/etc/resolv.conf`. For more information about `systemd-resolved`, consult the official [documentation](#). Note that `systemd-resolved` was not, and still is not, the default DNS resolver in Debian. If you have not configured your machine to use `systemd-resolved` as the DNS resolver, no action is required.

5.2.6 `systemd-boot` は分割されて独立したパッケージになりました

新規パッケージ `systemd-boot` はアップグレード時に自動的にインストールされません。 `systemd-boot` を利用していた場合は、この新規パッケージを手動でインストールしてください。インストールするまではブートローダーとして `systemd-boot` の以前のバージョンが利用される点に注意ください。このパッケージをインストールすると、自動的に `systemd-boot` をこのマシンのブートローダーとして設定します。Debianでのデフォルトのブートローダーは依然として GRUB です。以前にこのマシンがブートローダーとして `systemd-boot` を使うように設定していない場合、特に作業は要りません。

5.2.7 `systemd-journal-remote` は GnuTLS を使わなくなりました

任意のサービス `systemd-journal-gatewayd` および `systemd-journal-remote` は GnuTLS サポート無しでビルドされるようになりました。これはどちらのプログラムでも `--trust` オプションが無くなり、もし指定した場合はエラーが返されるようになったということです。

5.2.8 `bookworm` に向けた `adduser` の大幅な変更点

様々な変更が `adduser` にあります。最も目立つ変更点は `--disabled-password` および `--disabled-login` が文字通り動作するようになったことです。より詳細については `/usr/share/doc/adduser/NEWS.Debian.gz` を参照してください。

5.2.9 予測可能な Xen のネットワークインターフェイス名

systemd でのネットワークインターフェースに関する予測可能な命名規則ですが、Xen の netfront デバイス情報での固定名の生成にまで拡張されています。これはインターフェイスが以前のカーネルにアサインされたシステム名ではなく、enX# という形式の固定名を持つということを意味しています。アップグレード後、再起動の前にシステムを調整してください。他の情報については [NetworkInterfaceNames wiki page](#) を参照してください。

5.2.10 dash でのサーカムフレックス (^) の扱いの変更

Debian でシステムシェル /bin/sh をデフォルトで提供する dash ですが、これまでのように POSIX 互換の挙動を意図しているところから、サーカムフレックス (^) をリテラル文字として扱うように変更しました。これは bookworm では [^0-9] は "0 から 9 ではない" という意味ではなく、"0 から 9 と ^" という意味になるということです。

5.2.11 netcat-openbsd が抽象ソケットをサポートしました

The netcat utility for reading and writing data across network connections supports abstract sockets (see [unix.7.html#Abstract_sockets](#), and uses them by default in some circumstances.

デフォルトの状態では netcat は **netcat-traditional** によって提供されています。しかし、netcat が **netcat-openbsd** パッケージによって提供されていて AF_UNIX ソケットを使っている場合、この新しいデフォルト設定が適用されます。この場合、nc コマンドの -U オプションは @ で始まる引数をカレントディレクトリにある @ で始まるファイル名ではなく、抽象ソケットの要求であると解釈するようになっています。ファイルシステムのパーミッション設定が抽象ソケットへのアクセス制御に使えなくなるため、これはセキュリティに影響を与える可能性があります。./ をファイル名のプレフィックスに使う、あるいは絶対パスを指定すれば、@ から始まるファイル名を使い続けることができます。

5.3 廃止および非推奨となった事柄について

5.3.1 特記すべき廃止されたパッケージたち

以下は、よく知られていて特に廃止されたパッケージの一覧です (説明については [利用されなくなったパッケージ 参照](#))、

廃止パッケージの一覧には以下が含まれます:

- **libnss-ldap** パッケージは trixie から削除されました。この機能は **libnss-ldapd** ならびに **libnss-sss** によってカバーされています。
- **libpam-ldap** パッケージは trixie から削除されました。代替は **libpam-ldapd** となります。
- **fdflush** パッケージは trixie から削除されました。代わりとして、**util-linux** の `blockdev --flushbufs` を利用してください。

- GDAL の Perl バインディングは upstream でサポートされなくなったため、**libgdal-perl** パッケージは trixie から削除されています。GDAL の Perl サポートを必要とする場合は、CPAN で入手可能な **Geo::GDAL::FFI** パッケージが提供する FFI インターフェイスに移行できます。[BookwormGdalPerl Wiki ページ](#) に記述されているように、自前でバイナリをビルドする必要があります。

5.3.2 trixie で非推奨となったコンポーネント

次のリリースである Debian 14 (コードネーム forky) では、いくつかの機能が非推奨となります。14 へ更新する際にトラブルを防ぐためには、ユーザーは他の選択肢へ移行する必要があります。

これには以下の機能が含まれます:

- NSS サービス `gw_name` の開発は 2015 年に停止しました。関連パッケージ **libnss-gw-name** は将来の Debian リリースにて削除される可能性があります。upstream の開発者は代わりに **libnss-myhostname** の利用を推奨しています。
- **dmraid** は upstream での活動が 2010 年終わりから見られず、Debian でのサポートを受けている状態でした。bookworm がこれを含む最後のリリースとなるので、**dmraid** を利用している場合は対応の計画を適宜してください。
- **request-tracker4** はこのリリースで **request-tracker5** に置き換えられました。そして **request-tracker4** は将来のリリースで削除されます。今回のリリースのサポート期間内で **request-tracker4** から **request-tracker5** への以降を計画しておくことをお勧めします。
- The **isc-dhcp** suite has been deprecated by the ISC. The [Debian Wiki](#) has a list of alternative implementations, see the [DHCP Client](#) and [DHCP Server](#) pages for the latest. If you are using NetworkManager or systemd-networkd, you can safely remove the **isc-dhcp-client** package as they both ship their own implementation. If you are using the **ifupdown** package, you can experiment with `udhcpc` as a replacement. The ISC recommends the **Kea** package as a replacement for DHCP servers.

The security team will support the **isc-dhcp** package during the bookworm lifetime, but the package will likely be unsupported in the next stable release, see [bug #1035972 \(isc-dhcp EOL'ed\)](#) for more details.

5.4 既知の重大なバグ

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](#) that were tagged to be ignored in the last part of releasing trixie can be found in the [Debian Bug Tracking System](#). The following bugs were affecting trixie at the time of the release and worth mentioning in this document:

バグ 番号	パッケージ名 (ソースある いはバイナリ)	説明
1032240	akonadi-backend-mysql	akonadi server fails to start since it cannot connect to mysql database
1032177	faketime	faketime が (i386 では) 動作しない
918984	src:fuse3	provide upgrade path fuse -> fuse3 for bookworm
1016903	g++-12	tree-vectorize: O2 レベルで間違ったコードが生成される (-fno-tree-vectorize は動作している)
1020284	git-daemon-run	purge に失敗する: deluser -f: Unknown option: f
919296	git-daemon-run	以下のメッセージで失敗する 'warning: git-daemon: unable to open supervise/ok: file does not exist'
1034752	src:gluegen2	non-free なヘッダを含む

第6章 Debian に関するさらなる情報

6.1 もっと読みたい

Beyond these release notes and the installation guide (at <https://www.debian.org/releases/trixie/installmanual>) further documentation on Debian is available from the Debian Documentation Project (DDP), whose goal is to create high-quality documentation for Debian users and developers, such as the Debian Reference, Debian New Maintainers Guide, the Debian FAQ, and many more. For full details of the existing resources see the [Debian Documentation website](#) and the [Debian Wiki](#).

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

6.2 手助けを求めるには

Debian ユーザ向けのヘルプ・アドバイス・サポートなどは、いろいろな場所から得られます。しかしこれらを頼りにするのは、入手できるドキュメントでその問題について調査してからにしましょう。この章では新しく Debian ユーザになった人にとって役立つであろう、これらのリソースを簡単に紹介します。

6.2.1 メーリングリスト

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

6.2.2 インターネットリレーチャット (IRC)

Debian には、Debian ユーザのサポートや援助のために専用の IRC チャンネルが OFTC IRC ネットワークにあります。このチャンネルにアクセスするには、お好みの IRC クライアントを `irc.debian.org` に接続し、`#debian` に join してください。

チャンネルのガイドラインに従い、他のユーザをきちんと尊重してください。ガイドラインは [Debian Wiki](#) で参照できます。

For more information on OFTC please visit the [website](#).

6.3 バグを報告する

私たちは Debian を高品質な OS にするよう努めていますが、だからといって私たちの提供するパッケージにバグが皆無というわけではありません。Debian の "オープンな開発体制" という考え方に合致し、また、ユーザに対するサービスとして、私たちは報告されたバグに関するすべての情報をバグ追跡システム (Bug Tracking System: BTS) で提供しています。この BTS は <https://bugs.debian.org/> で閲覧できます

もしディストリビューションや、その一部であるパッケージされたソフトウェアにバグを見つけたら、将来のリリースで修正できるよう、その問題点の報告をお願いします。バグを報告するには有効な電子メールアドレスが必要です。これをお願いしているのは、バグを追跡できるようにするため、そして追加情報が必要になった場合に開発者が報告者に連絡できるようにするためです。

You can submit a bug report using the program `reportbug` or manually using e-mail. You can find out more about the Bug Tracking System and how to use it by reading the reference documentation (available at `/usr/share/doc/debian` if you have **doc-debian** installed) or online at the [Bug Tracking System](#).

6.4 Debian に貢献する

You do not need to be an expert to contribute to Debian. By assisting users with problems on the various user support [lists](#) you are contributing to the community. Identifying (and also solving) problems related to the development of the distribution by participating on the development [lists](#) is also extremely helpful. To maintain Debian's high-quality distribution, [submit bugs](#) and help developers track them down and fix them. The tool `how-can-i-help` helps you to find suitable reported bugs to work on. If you have a way with words then you may want to contribute more actively by helping to write [documentation](#) or [translating](#) existing documentation into your own language.

If you can dedicate more time, you could manage a piece of the Free Software collection within Debian. Especially helpful is if people adopt or maintain items that people have requested for inclusion within Debian. The [Work Needing and Prospective Packages database](#) details this information. If you have an interest in specific groups then you may find enjoyment in contributing to some of Debian's [subprojects](#) which include ports to particular architectures and [Debian Pure Blends](#) for specific user groups, among many others.

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

第7章 アップグレードの前に bookworm システムを調整する

この付録には、trixie へアップグレードする前に bookworm パッケージを確実にインストールしたりアップグレードする方法についての情報が述べられています。

7.1 bookworm システムのアップグレード

Basically this is no different from any other upgrade of bookworm you've been doing. The only difference is that you first need to make sure your package list still contains references to bookworm as explained in *Checking your APT source-list files*.

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

7.2 APT source-list ファイルのチェック

APT source-list ファイル (`sources.list(5)` 参照) 内の行で "stable" を指定している行があるなら、trixie への準備が事実上できています。もしアップグレードへの準備がまだできていない場合には、これはお望みの設定ではないかもしれません。すでに `apt update` を実行済みでも、以下の手順に従えば問題なく元に戻すことができます。

trixie からパッケージのインストールもしてしまっているなら、おそらくこれ以上 bookworm からパッケージをインストールしても無意味でしょう。この場合、続けるかどうかを自分で決断しなければなりません。パッケージをダウングレードすることはできますが、その方法はここでは扱いません。

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:`

- URIs: `tor+https:`

for a reference to "stable". If you find any, change "stable" to "bookworm".

`deb file:` または URIs: `file:` で始まっている行があるなら、その行が指定している場所が `bookworm` か `trixie` のどちらのアーカイブなのかを自分で調べなければなりません。

重要: `deb cdrom:` または URIs: `cdrom:` で始まっている行は、絶対に変更しないでください。変更するとその行は無効になって、もう一度 `apt-cdrom` を実行しなければならなくなるでしょう。 `cdrom:` ソースが "unstable" を指定していても心配しないでください。混乱するかもしれませんが、これで正常なのです。

変更が済んだら、ファイルを保存してから

```
# apt update
```

と実行して、パッケージリストを更新してください。

7.3 Performing the upgrade to latest bookworm release

To upgrade all packages to the state of the latest point release for bookworm, do

```
# apt full-upgrade
```

7.4 古く不要になった設定ファイルを削除する

システムを `trixie` へアップグレードする前に、古い設定ファイル (`/etc` 以下にある `*.dpkg-{new,old}` ファイルなど) をシステムから削除することを推奨します。

第8章 リリースノートの貢献者たち

たくさんの方がリリースノートを手伝ってくれました。以下の方々もそうですが、他にもいらっしゃいます。

- ADAM D. BARRAT (2013 年での様々な修正),
- ADAM DI CARLO (以前のリリース),
- ANDREAS BARTH ABA (2005 - 2007 間のリリース),
- ANDREI POPESCU (さまざまな貢献),
- ANNE BEZEMER (以前のリリース),
- BOB HILLIARD (以前のリリース),
- CHARLES PLESSY (GM965 問題の解説),
- CHRISTIAN PERRIER BUBULLE (Lenny インストールについて),
- CHRISTOPH BERG (PostgreSQL 固有の問題について),
- DANIEL BAUMANN (Debian Live),
- DAVID PRÉVOT TAFFIT (Wheezy リリースについて),
- EDDY PETRIȘOR (さまざまな貢献),
- EMMANUEL KASPER (バックポート),
- ESKO ARAJÄRVI (X11 アップグレードの書き直し),
- FRANS POP FJP (以前のリリース (Etch) について),
- GIOVANNI RAPAGNANI (数え切れない貢献),
- GORDON FARQUHARSON (ARM 移植版関連),
- HIDEKI YAMANE HENRICH (2006 年から貢献),
- HOLGER WANSING HOLGERW (2009 年から貢献),
- JAVIER FERNÁNDEZ-SANGUINO PEÑA JFS (Etch および Squeeze のリリースについて),
- JENS SEIDEL (ドイツ語翻訳、数え切れない貢献),
- JONAS MEURER (syslog 関連),
- JONATHAN NIEDER (Squeeze および Wheezy リリースについて),
- JOOST VAN BAAL-ILIĆ JOOSTVB (Wheezy および Jessie リリースについて),

- JOSIP RODIN (以前のリリース),
- JULIEN CRISTAU JCRISTAU (Squeeze および Wheezy リリースについて),
- JUSTIN B RYE (英語の修正),
- LAMONT JONES (NFS 問題の解説),
- LUK CLAES (編集者のモチベーション管理),
- MARTIN MICHLMAYR (ARM 移植版関連),
- MICHAEL BIEBL (syslog 関連),
- MORITZ MÜHLENHOFF (さまざまな貢献),
- NIELS THYKIER NTHYKIER (Jessie リリースについて),
- NOAH MEYERHANS (数え切れない貢献),
- NORITADA KOBAYASHI (日本語翻訳 (コーディネート)、数え切れない貢献),
- OSAMU AOKI (さまざまな貢献),
- PAUL GEVERS ELBRUS (buster リリースについて),
- PETER GREEN (カーネルバージョンメモ),
- ROB BRADFORD (Etch リリース),
- SAMUEL THIBAUT (d-i でのブライユ点字サポートの解説),
- SIMON BIENLEIN (d-i でのブライユ点字サポートの解説),
- SIMON PAILLARD SPAILLAR-GUEST (数え切れない貢献),
- STEFAN FRITSCH (Apache 関連の解説),
- STEVE LANGASEK (Etch リリース),
- STEVE MCINTYRE (Debian CD),
- TOBIAS SCHERER ("proposed-update" の解説),
- VICTORY VICTORY-GUEST (マークアップの修正, 2006 年より貢献),
- VINCENT MCINTYRE ("proposed-update" の解説),
- W. MARTIN BORGERT (Lenny リリースノートの編集、DocBook XML への変換).

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