Setting Up a `systemd` Service

Setting up a systemd service on SUSE Linux Enterprise Server for NVIDIA allows administrators to automate process execution, manage application lifecycles and configure custom background tasks. By defining declarative unit files, users can specify service dependencies, control startup behavior and apply security sandboxing features. This ensures reliable background operation, simplifies troubleshooting with systemctl and improves overall system stability across system reboots.

Prerequisites

- Basic understanding of Linux processes, daemons and control groups

Publication Date: 26 Nov 2026

Contents

1 What is `systemd`? 3

2 Structure of a unit file 3

3 Unit file types 5

4 Unit dependencies and order 6

5 Creating a Linux service with `systemd` 7

6 The **`sysctemctl edit`** command 8

7 Securing `systemd` services 10

8 Debugging a `systemd` service 13

9 **`systemctl`** commands overview 15

10 Legal Notice 17

A GNU Free Documentation License 17

1 What is systemd?

systemd is a system and service manager for Linux operating systems. It is the default initialization system for major Linux distributions. systemd is not directly initiated by the user, but installed through the /sbin/init and started during the early boot. systemd acts as the init system that brings up and maintains user space services when run as the first process on boot (PID 1). PID 1 is known as *init* and is the first Linux user-mode process created. It runs until the system shutdown.

systemd owns PID 1, and is started directly by the kernel. All other processes are started directly by systemd or one of its child processes. systemd mounts the host's file system and manages temporary files. It is backward compatible with the SysV init scripts. SysV is an initialization system that predates systemd.

In systemd, a unit is a resource that the system knows how to operate on and manage. This is the primary object that the systemd tools use. These resources are defined with configuration files called unit files.

systemctl is the central management tool for controlling the init system. It is used to examine and control the state of the systemd system and service manager.

Targets in systemd are groups of related units that act as synchronization points during a system boot. Target unit files have a .target file extension. Target units group together various systemd units through a chain of dependencies.

For troubleshooting, you can use **journalctl**, which is used to query and display log messages from the systemd journal.

For more information on systemd, you can refer to https://systemd.io and **man 1 systemd**.

2 Structure of a unit file

In systemd, a unit refers to any resource that the system knows how to operate on and manage. This is the primary object that the systemd tools use. These resources are defined using configuration files called unit files. Administration is easier when you understand unit files when working with systemd. Unit files use a simple declarative syntax that allows you to see easily the purpose and effects of a unit upon activation. Unit files have sections with directives, for example:

```
[Section]
Directive1=value
Directive2=value
```

```
        . . .
```

Unit file types include the following sections:

`[Unit]`

The first section found in most unit files is the `[Unit]` section. This section is used to define the metadata of the unit file and configure the relationship of the unit file to other unit files. This section is usually placed at the top because it provides an overview of the unit file.

`[Automount] / [Mount] / [Path] / [Service] / [Slice] / [Socket] /[Swap] / [Timer]`

Sections containing directives that are specific to the respective type. See *Section 3, "Unit file types"* for a list of available types. Note that the types `device`, `target`, `snapshot` and `scope` do not have a type-specific section.

`[Install]`

This is often the last section in the unit file and is optional. This section is used to define the behavior of a unit file when it is enabled or disabled. When you enable a unit file, it automatically starts at boot. Based on the specific unit, there could be a dependency on other related units to work properly. For example, `chrony` requires the directives `After`, `Wants`, and `Before`, which are all dependencies for `chrony` to work with.

EXAMPLE 1: A `systemd` SERVICE FILE

```
[Unit]
Description=usbguard ❶

[Service]
ExecStart=/usr/sbin/usb-daemon ❷

[Install]
WantedBy=multi-user.target ❸
```

❶ A brief and meaningful description explaining the service file's purpose.

❷ Specifies the program to be executed when the service is started.

❸ Starts a multi-user system with networking, and no graphical environment. This directive allows you to specify a dependency relationship.

3 Unit file types

You can determine the type of unit by its file extension. `systemd` categorizes units according to the type of resource they describe.

Types of unit files available for `systemd`:

`.service`
: Describes how to manage a service or application. This includes how to start or stop the service, reload its configuration file (if applicable), under what conditions the service starts automatically, and the dependency or the hierarchy information for related unit files.

`.scope`
: This unit file is automatically created by `systemd` from the information received from the D-Bus interface and is used to manage sets of system processes that are created externally.

`.path`
: Defines a path for path-based activation. By default, a `.service` unit file of the same base name is activated. `inotify` is a kernel API that is used by programs that want to be notified about changes to files.

`.snapshot`
: The **`systemctl snapshot`** command automatically creates a `.snapshot` unit file. This command creates temporary snapshots of the current state of the system. You can modify the current state of the system after making changes. Snapshots are used for rolling back temporary states.

`.timer`
: Defines a timer that is managed by `systemd`. This is similar to a cron job for delayed or scheduled activation. A unit file with the same name, but with file extension `.service` is started when the timer is reached.

`.slice`
: Associate Linux Control Group nodes, which allow resources to be assigned or restricted to any processes associated with the slice. The name indicates the hierarchy within the control group tree. Units are placed in slices by default depending on their type.

`.target`
: Provides synchronization for other units during a boot up or a change in state, or brings the system to a new state. Other units specify their relation to targets in order to sync with the target's operations.

`.socket`

Describes a network, an IPC socket, or a FIFO buffer that `systemd` uses for socket-based activation. There is an associated `.service` file that starts when an activity is seen on the socket that this unit defines.

`.device`

Defines a device that has been designated for `systemd` management by `udev` or `sysfs` file system. Not all devices have the `.device` file. This unit file is required when ordering, mounting, or accessing a device.

`.swap`

Defines the swap space on the system. The name of the unit file must reflect the device or file path of the space.

`.mount`

Defines a mount point on the system to be managed by `systemd`. This file is named after the mount path, with the slashes changed to dashes. Entries within `/etc/fstab` can have units created automatically.

`.automount`

Defines a mount point that is automatically mounted. Name the file after the mount point that it refers to. A matching `.mount` unit file is required to define the specifics of the mount.

4 Unit dependencies and order

`systemd` has two types of dependencies: requirement and order dependencies. Requirement dependencies specify which other units must be either started or stopped when activating a unit. Order dependencies specify the order in which units must be started.

Unit dependencies

Unit files have the dependencies feature. A unit may want or require one or more other units before it can run. These dependencies are set in unit files with the directives `Wants` and `Requires`.

`Wants`

For example, if unit A has `Wants=unit B`, when unit A is run, unit B runs as well. But if unit B starts successfully or not, does not have an influence on unit A running successfully.

`Requires`

If unit A has `Requires=unit B`, both units run but if unit B does not run successfully, unit A is deactivated. It does not matter if the processes of unit A would have run successfully.

Unit order

Without proper instructions, `systemd` can run a group of units at the same time. Starting services in the right order is important for a good functioning of the Linux system. You can arrange the order with the unit file directives `Before` and `After`.

`Before`

For example, if unit A has `Before=unit B`, when both units are run, unit A is executed fully before unit B.

`After`

If unit A has `After=unit B`, when both units are run, unit B is executed fully before unit A.

5 Creating a Linux service with `systemd`

Create an auto-start task or program that executes every time that you boot or reboot your system by creating a Linux service with `systemd`.

EXAMPLE 2: AUTOMATING TASK EXECUTION WITH `systemd`

Creating a custom `systemd` service, involves creating a service unit file, which defines the service and its behavior.

1. Create a new script at the specified location, that is used by the `systemd` service:

```
vi /usr/local/bin/FILE_NAME.sh
```

The `/usr/local/bin` is the standard location for installing custom scripts and executables specific to the system. By placing the script in this location, you make it accessible to all system users without a need to specify the full path.

2. Paste the following in the file:

```
#!/bin/bash
     echo "Hello, Everyone!"
```

3. Make the script executable:

```
> sudo  chmod +x /usr/local/bin/FILE_NAME.sh
```

4. Create a `systemd` unit file at the specified location:

```
vi /etc/systemd/system/FILE_NAME.service
```

5. Paste the following in the file:

```
[Unit]
Description= Name service

[Service]
ExecStart=/usr/local/bin/FILE_NAME.sh

[Install]
WantedBy=multi-user.target
```

   The `Unit` section gives you a description of the service. The `Service` section defines the service and its behavior. The `ExecStart` directive specifies the command to start the service. The `Install` section specifies when the service should start.

6. Save and exit the file.

7. To make `systemd` aware of the new service, run:

```
> sudo  systemctl SERVICE_NAME
```

8. Start,enable and check the status of the service:

```
systemctl start SERVICE_NAME
```

```
systemctl enable SERVICE_NAME
```

```
systemctl status SERVICE_NAME
```

6 The `sysctemctl edit` command

You can use the **`systemctl`** command to edit and modify an existing service file.

There are three main directories where unit files are stored on the system:

`/usr/lib/systemd/system/`
: When the RPM packages are installed, `systemd` unit files reside here.

`/run/systemd/system/`
: `systemd` unit files created at run time. This directory takes precedence over the directory with installed service unit files.

`/etc/systemd/system/`
: `systemd` unit files that are created by the **`systemctl enable`** command and also unit files added for extending a service. This directory takes precedence over the directory with unit files created at run time.

By default, the **`sysctemctl edit`** command opens a unit file snippet, for example:

```
> sudo  systemctl edit testhttp.service
```

This creates a blank file that is used to override or add directives to the unit file definition. A directory is created in `/etc/systemd/system`, which contains the name of the unit file with `.d` appended. For example, a directory called `testhttp.service.d` is created.

Within the directory, a snippet called `override.conf` is created. `systemd` merges the override snippet with the full unit file when the unit is loaded. This snippet's directives take precedence over those directives in the original unit file.

With the `--full` flag, you can edit the full unit file instead of creating a snippet. For example:

```
> sudo  systemctl edit --full testhttp.service
```

This loads the current unit file into the editor, allowing you to make modifications. When you save and exit the editor, the modified file is written to `/etc/systemd/system`, which takes precedence over the system's unit definition, usually located in `/lib/systemd/system`.

To remove any additions you have made, you can delete the unit's `.d` configuration directory or the the modified service file from `/etc/systemd/system`. After deleting the file or directory, reload the `systemd` process so that it reverts back to the initial system process.

7 Securing `systemd` services

Linux increases its security by separating privileges between individual components of the operating system. System services already have a default level of security. For example, their processes run under their own user ID, which limits the changes they can perform on the system.

The default level of privilege separation provides only a basic protection. Services can still perform as many changes as normal local users, though not as many as `root`. A higher level of system security requires limiting what services can perform and prevents them from certain privileges that normal users are allowed to use.

7.1 Why is securing `systemd` services important?

Securing `systemd` services increases the security of the whole operating system and protects sensitive data contained on its file system. With `systemd`, you can configure your system in many ways. `systemd` runs as the first process on boot (PID1) so it has a lot of power on your Linux environment.

`systemd` can apply additional restrictions to local services by using technologies included in the kernel. These restrictions are activated by adding specific options to the systemd service definition and restarting the service. `systemd` has a command-line tool **`systemd-analyze security`**. This command analyzes the services and checks whether the services are using its security options.

7.2 What is the **`systemd-analyze security`** command?

The command analyzes the security and sandboxing settings of the specified service units. A detailed analysis of the security settings is executed and displayed. If a service unit is not specified, all currently loaded, long-running service units are inspected and the results are displayed in a terse table.

Upon checking the security settings, the command assigns a numeric value , also known as *exposure level*. This value depends on how important a setting is. It then calculates the overall exposure level for the entire unit. This value ranges from 0.0 to 10.0, which is an indicator of how exposed a service is security-wise. High exposure levels indicate that the service might benefit from additional security settings. Low exposure levels indicate tight security restrictions.

7.3 Using the `systemd-analyze security` command

Use the **`systemd-analyze security`** command to analyze the security settings of a `systemd` service. The `security` option analyzes the security and the sandboxing settings of one or more specified services.

1. Create a `systemd` service in the `/etc/systemd/system`.
2. Reload the service files to include the new service:

```
> sudo  systemctl daemon-reload
```

3. Start, enable and check the status of the service:

```
> sudo  systemctl start SERVICE_NAME
```

```
> sudo systemctl enable SERVICE_NAME
```

```
> sudo  systemctl status SERVICE_NAME
```

4. Analyze the security settings of the service:

```
> sudo  systemd-analyze security SERVICE_NAME
```

   For example:

```
> sudo  systemd-analyze security test.service
  NAME                DESCRIPTION                                EXPOSURE
  ✗ PrivateNetwork=     Service has access to the host's network      0.5
  ✗ User=/DynamicUser=  Service runs as root user                     0.4
  ✗ DeviceAllow=        Service has no device ACL
  ...
  → Overall exposure level for test.service: 9.6 UNSAFE #
```

7.4 How to improve the overall exposure

If you get *9.6 UNSAFE*, you can use `[Service]` part of the service definition file to add any of the below options. For example:

```
[Service]
NoNewPrivileges=yes
PrivateTmp=yes
PrivateNetwork=yes
InaccessibleDirectories=/home
```

```
.....
```

NoNewPrivileges=yes

New privileges are not required.

PrivateTmp=yes

Private directory for temporary files. This option provides the service with a private `/tmp` isolated from the host system's `/tmp`. The shared host `/tmp` directory is a major source of security problems, such as symlink attacks and DoS `/tmp` temporary files.

PrivateNetwork=yes

This option isolates the service and its processes from networking. This prevents external network requests from reaching the protected service. Be aware that certain services require the network to be operational.

InaccessibleDirectories=/home

This option makes the specified directories inaccessible to the service. This option narrows the range of directories that can be read or modified by the service, for example, to secure users' private files.

ReadOnlyDirectories=/var

This option makes the specified directories inaccessible for writing to the service. The example configuration makes the whole tree below `/var` read-only. This option prevents the service from damaging the system files.

CapabilityBoundingSet=CAP_CHOWN CAP_KILL

This option restricts the kernel capabilities that a service can retain. In the example above, only the `CAP_CHOWN` and `CAP_KILL` capabilities are retained by the service, and the service and any processes it creates cannot obtain any other capabilities, not even via setuid binaries.



Tip: The **pscap** command tool

To easily identify which processes on your system retain which capabilities, use the **`pscap`** command tool from the `libcap-ng-utils` package.

The `~` prefix inverts the meaning of the option—. Instead of listing all capabilities that the service retains, you can list the ones it does not retain:

```
...
  [Service]
  CapabilityBoundingSet=~CAP_SYS_PTRACE
  ...
```

LimitNPROC=1, LimitFSIZE=0

You can use *resource limits* to apply security limits on services. Two of them can disable specific operating system features: `RLIMIT_NPROC=1` disables process forking, while `RLIMIT_FSIZE=0` disables creating non-empty files on the file system.

DeviceAllow=/dev/null rw

This option limits access to `/dev/null`, disallowing access to any other device nodes.

These are some options you can use.

EXAMPLE 3: `NoNewPrivileges=yes` OPTION EXAMPLE

This example shows a basic script configured to run as a specific user with privilege escalation strictly disabled.

```
[Unit]
Description=Simple Secure Service

[Service]
# Run as a non-root user
User=john
Group=john

# Disable gaining new privileges (setuid, setgid, etc.)
NoNewPrivileges=yes
```

8 Debugging a `systemd` service

Use the **`systemctl`** and **`journalctl`** commands to investigate the reasons for a failed system service.

When `systemd` fails to start a service, a generic error message is displayed, for example:

```
# systemctl start apache2
Job for apache2.service failed because the control process exited with error code.
See "systemctl status apache2.service" and "journalctl -xe" for details.
```

List all available services and their current status

```
# systemctl list-units -t service --all
```

For example:

```
# systemctl list-units -t service --all
```

```
  UNIT                                    LOAD      ACTIVE   SUB     DESCRIPTION>
  accounts-daemon.service                 loaded    active   running Accounts Se>
  apache2.service                         loaded    failed   failed  The Apache HTTP
 server>
\u25cf acpid.service                      not-found inactive dead    acpid.servi>
 after-local.service                     loaded    inactive dead    /etc/init.d>
  alsa-restore.service                    loaded    active   exited  Save/Restor>
  alsa-state.service                      loaded    inactive dead    Manage Soun>
\u25cf amavis.service                     not-found inactive dead    amavis.serv>
  apparmor.service                        loaded    active   exited  Load AppArm>
  appstream-sync-cache.service            loaded    inactive dead    Synchronize>
  auditd.service                          loaded    active   running Security Au>
  augenrules.service                      loaded    active   exited  auditd rule>
```

Check whether a specific service has failed

```
# systemctl is-failed apache2.service
              failed
```

Check the status of the failed service

```
# systemctl status --full --lines=50 apache2
  ● apache2.service - Apache HTTP Server
  Loaded: loaded (/usr/lib/systemd/system/apache2.service; enabled; vendor preset:
 disabled)
  Active: failed (Result: exit-code) since Fri 2021-08-20 10:24:15 CEST; 2min 24s
 ago
    Docs: https://httpd.apache.org/docs
 Process: 2491 ExecStart=/usr/bin/apache2 --add-runtime oci=/usr/sbin/apache-runc
 $DOCKER_NETWORK_OPTIONS $DOCKER_OPTS >
Main PID: 2491 (code=exited, status=1/FAILURE)
```

In the above example, the service ran as process ID 2491, but failed. Error messages give you a hint on what to do.

Check the logs of the failed service

```
#  journalctl --catalog --pager-end --unit=apache2
[...]
  Aug 20 10:24:15 localhost.localdomain dockerd[2479]: unable to configure the
 Docker daemon with file /etc/apache2/daemon.json:
cannot unmarshal string into Go value of type map[string]interface {}
```

The option `--unit` limits the log entries only to the failed Apache2 service. The error message suggests looking into the file `/etc/docker/daemon.json`. In this scenario, the error was caused by a wrong syntax. You can fix this and restart the Apache2 service.

9 `systemctl` commands overview

The **`systemctl`** command is used to examine and control the state of `systemd` and service manager.

You can use the following common **`systemctl`** commands and refer to the *man systemctl* page.

9.1 Viewing `systemd` information

To view information about `systemd` components, you can use the following commands:

systemctl list-units

Lists the `systemd` units. You can use the optional arguments: **`--state=running`** to show the active units and **`--type=service`** to show the exited and active units.

systemctl list-unit-files

Lists the `systemd` units and the status, such as static, generated, disabled, alias, masked, and enabled.

systemctl list-dependencies

Lists the dependency tree.

systemctl list-dependencies *`UNIT_FILE`*

Lists the dependencies of a unit file.

9.2 Managing `systemd` services

The **`systemctl`** command enables you to perform the following tasks with services.

systemctl status *`SERVICE`*

Checks the status of the specific service.

systemctl show *`SERVICE`*

Displays the service information.

systemctl start *`SERVICE`*

Instead of manually starting the service, use the **`start`** command. When a change is made to the configuration file, the related service must be started again.

systemctl stop *`SERVICE`*

Stops a specific running service.

systemctl restart *SERVICE*

Instead of manually restarting the service, use the **`restart`** command. When a change is made to the configuration file, the related service must be restarted again.

systemctl enable *SERVICE*

Enables the service on boot.

systemctl disable *SERVICE*

Disables the service on boot.

systemctl reload-or-restart *SERVICE*

Reload the service if it supports reloading, otherwise it restarts the service. If the service is not running, it is restarted.

systemctl mask *SERVICE*

When a service is masked, this means the unit file is symlinked to `/dev/null`. A symlink for a masked service is created from `/etc/systemd/system` to point to `/dev/null`. This makes it impossible to load the service even if another enabled service requires it. It must be stopped manually, or it continues to run in the background. You can use **`--runtime`** option to only mask temporarily until the next reboot of the system.

```
Created symlink /etc/systemd/system/FOSSLinux.service → /dev/null.
```

systemctl unmask *SERVICE*

Unmasks the service. It is effective when the system is started or restarted manually.

9.3 Managing system states

The **`systemctl`** command enables you to perform power management processes on your system, like restarting, shutting down and so on, as described below.

systemctl reboot

Reboots the system `reboot.target`.

systemctl poweroff

Powers off the system `poweroff.target`.

systemctl emergency

Goes into the emergency mode `emergency.target`.

systemctl default

Goes back to default target `multi-user.target`.

10 Legal Notice

Copyright© 2006– 2026 SUSE LLC and contributors. All rights reserved.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or (at your option) version 1.3; with the Invariant Section being this copyright notice and license. A copy of the license version 1.2 is included in the section entitled “GNU Free Documentation License”.

For SUSE trademarks, see https://www.suse.com/company/legal/. All other third-party trademarks are the property of their respective owners. Trademark symbols (®, ™ etc.) denote trademarks of SUSE and its affiliates. Asterisks (*) denote third-party trademarks.

All information found in this book has been compiled with utmost attention to detail. However, this does not guarantee complete accuracy. Neither SUSE LLC, its affiliates, the authors, nor the translators shall be held liable for possible errors or the consequences thereof.

A GNU Free Documentation License

Copyright (C) 2000, 2001, 2002 Free Software Foundation, Inc. 51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA. Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

0. PREAMBLE

The purpose of this License is to make a manual, textbook, or other functional and useful document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or non-commercially. Secondarily, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you". You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The "Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not "Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image for-

mats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

A section "Entitled XYZ" means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as "Acknowledgements", "Dedications", "Endorsements", or "History".) To "Preserve the Title" of such a section when you modify the Document means that it remains a section "Entitled XYZ" according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or non-commercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally

prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.

B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.

C. State on the Title page the name of the publisher of the Modified Version, as the publisher.

D. Preserve all the copyright notices of the Document.

E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.

F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.

G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.

H. Include an unaltered copy of this License.

I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.

J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.

K. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.

L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.

M. Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.

N. Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.

O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties--for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled "History" in the various original documents, forming one section Entitled "History"; likewise combine any sections Entitled "Acknowledgements", and any sections Entitled "Dedications". You must delete all sections Entitled "Endorsements".

6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an "aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided for under this License. Any other attempt to copy, modify, sublicense or distribute the Document is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See https://www.gnu.org/copyleft/.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation.

ADDENDUM: How to use this License for your documents

```
Copyright (c) YEAR YOUR NAME.
Permission is granted to copy, distribute and/or modify this document
under the terms of the GNU Free Documentation License, Version 1.2
or any later version published by the Free Software Foundation;
with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts.
A copy of the license is included in the section entitled “GNU
Free Documentation License”.
```

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the “with...Texts.” line with this:

```
with the Invariant Sections being LIST THEIR TITLES, with the
Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST.
```

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.