Tanja Roth
Christopher-Julian Zwickl

DAPS Quick Start Guide

May 24, 2012 http://daps.sourceforge.net

The DocBook Authoring and Publishing Suite (DAPS) is developed as open source software for Linux operating systems and is licensed under the GPL. A number of integrated tools, stylesheets, scripts, and makefiles help technical writers in the editing, translation and publishing process. DAPS supports single source publishing into a number of different output formats, and is suited both for small documentation projects as well as for comprehensive projects, created by multiple authors in a collaborative effort.

Target Audience

This document is intended for users who want to make efficient use of DocBook-XML for editing and publishing their documentation—be it documentation sets, individual books, or articles. Key knowledge of XML and DocBook is required, as well as key knowledge of using the Bash Shell (or command line interfaces in general).

Supported DocBook Versions

Currently, DAPS supports only DocBook 4.x. Support for DocBook 5.x is planned for version 2.0.

Requirements and Additional Software

DAPS is a lean solution that does not require a lot of system resources.

Hardware Requirements

RAM
: The required amount of RAM mostly depends on the volume of your documentation projects. For creation of PDF output, 2 GB of RAM are recommended.

CPU
: If you have multiple or very large documentation projects, a machine with multiple cores is recommended.

Hard Disk Space
: The disk space consumed mostly depends on the amount of your documentation sources and the number of output formats you want to generate.

Software Requirements

In addition to DAPS, you need the following software:

- An XML (or text) editor of your choice.
- For generating PDF output: an FO formatter, like FOP [`http://projects.apache.org/projects/fop.html`] or XEP [`http://www.renderx.com`]. Antenna House Formatter [`http://www.antennahouse.com`] is currently not supported. Whereas FOP is an open source product, both XEP and Antenna House are commercial products.

When installing DAPS as an RPM package (on any SUSE-based system), dependencies on other software packages are automatically resolved during installation.

For installing the DAPS sources on other Linux distributions (with `configure`, `make`, and `make install`), make sure the following packages or tools are installed on your system. Otherwise the installation scripts will fail.

- convert (included in the `ImageMagick` package)
- DocBook 4
- DocBook 4 Stylesheets (usually a separate package)
- make
- xmlcatalog (usually part of the package `libxml2`, `libxml-utils`, or `libxml-tools`)
- xmllint (usually part of the package `libxml2`, `libxml-utils`, or `libxml-tools`)
- xsltproc (if not available as a separate package, it may be included in `libxslt`)

During the installation procedure, the `convert` script informs you about any further missing software packages. Refer to Procedure 1, “Installing the DAPS Sources” (page 3) for more information.

Directory Structure

For DAPS to work out of the box, it requires a certain organization of your XML files and images within your documentation directory. For details, refer to Section “Directory Structure” (page 4). You can generate the necessary structure with the DAPS initialization script, `daps-init`. For instructions on how to make existing DocBook projects compatible with DAPS, refer to Section “Migration of Existing DocBook Projects” (page 9).

Additional Software

To add further components like version management or a workflow mechanism for your projects, use DAPS in combination with the following software:

- Any version management system, like CVS, Subversion, Mercurial or Git.
- Docmanager, a command-line tool for adding and retrieving meta-information for all files belonging to a documentation project. Docmanager is especially useful for larger, collaborative projects where it helps you to keep track of owners (authors) and editing status of all files. However, Docmanager requires to host your documentation files on a Subversion server.

Together with the software components mentioned above, DAPS can be used as a fully-fledged authoring and content management system for documentation projects based on DocBook.

Installation

The DocBook Authoring and Publishing Suite can be installed and used on any Linux distribution. Currently, DAPS is available as RPM package for the openSUSE distribution. Eventually, RPM packages for further distributions may become available. In the meantime, you can download a tarball with the DAPS sources and install them on any distribution as described in Procedure 1, “Installing the DAPS Sources” (page 3).

Installing DAPS on openSUSE

For openSUSE, the `daps` package is available from the `Documentation:Tools` repository at `http://download.opensuse.org/repositories`. For the complete path to the repository, add the version number of your openSUSE installation. For example, the complete repository URL for openSUSE 12.1 is: `http://download.opensuse.org/repositories/Documentation:/Tools/openSUSE_12.1/`

For basic details on how to add software repositories and how to install software on openSUSE, refer to the openSUSE *Reference Guide*. The guide is available from `http://www.suse.com/documentation`.

Choose one of the following installation methods. Dependencies to other packages that are required by DAPS are automatically resolved.

1-Click Installation from the Internet
: Go to `http://software.opensuse.org/` and search for the `daps`. On the resulting page, click *daps* → *Show other versions* → *Show unstable packages*. Select the version that is provided by the repository `Documentation:Tools`. To start installation of the package click the *1-Click Install* link next to the selected package.

Installation with YaST (GUI)
: Add the `Documentation:Tools` repository that matches your openSUSE version—see the introduction to this section. From this repository, install the `daps` package. For more information on installing packages with YaST refer to the openSUSE *Reference Guide*, chapter *Installing Or Removing Software*.

Installation with `zypper` (command line)
: Add the `Documentation:Tools` repository that matches your openSUSE version—see the introduction to this section. From this repository, install the `daps` package. For more information on installing packages with YaST refer to the openSUSE *Reference Guide*, chapter *Managing Software with Command Line Tools*.

Installing DAPS on Other Linux Distributions

For installation on other Linux distributions, the DAPS sources are available as tarball. They can be installed with `configure`, `make`, and `make install`.

Procedure 1 *Installing the DAPS Sources*

Before starting the installation, check the DAPS Requirements and Additional Software (page 1) and make sure to have all required packages and tools installed.

1. Go to `http://sourceforge.net/projects/daps/files/` and download the DAPS source tarball, `daps-`*`versionnumber`*`.tar.bz2` .
2. Unpack the tarball:

   ```
   tar xvf daps-versionnumber.tar.bz2
   ```
3. Change to the newly created `daps` subdirectory and start the configure script:

   ```
   $ ./configure
   ```

   If you want to adjust the DAPS installation paths:

   a. View the available options with

      ```
      $ ./configure --help
      ```
   b. Run the configure script with the desired option.

   The script checks your system for any software relevant to DAPS or the DAPS installation process. It also creates a makefile that will be used during installation. Based on the analysis, the script shows a summary that includes the following information:

   - the DAPS installation paths,
   - an overview of DAPS features that will be available on your system if you install DAPS now, and
   - which software is still missing to enable the remaining DAPS features as well.
4. Check the summary carefully.
5. Install missing packages, if necessary. After installing new packages, repeat Step 3 (page 3) and check the summary again.
6. If everything is prepared according to your wishes, enter:

   ```
   $ make
   ```
7. Start the installation process with:

   ```
   $ sudo make install
   ```

Installing and Configuring the FO Formatter

For installation and configuration of an FO formatter (for generating PDF output), refer to its installation instructions (or to your system administrator). For FOP, you usually only need to install the respective FOP package. However, not all FOP packages contain hyphenation pattern files.

Defining Documentation Projects

The easiest way to set up a new documentation project from scratch is to use the DAPS initialization script `daps-init`. For instructions how to do so, refer to Procedure 2, “Using `daps-init`” (page 4). The script automatically creates the Key Files (page 3) and Directory Structure (page 4) that you need to get started with DAPS.

Key Files

The following key files define a documentation project so that it can be processed by DAPS:

Main File
: An XML file containing the “starting point” (the highest-level object) of your documentation project (for example, `book` or `article`). For larger documentation projects, it is good practice to name the file `MAIN-`*`PROJECTNAME`*`.xml` , but you can use any other filename as well.

Doc Config (DC) File
: A configuration file defining a number of parameters for your documentation deliverable (for example, the main file, layout variants, or which profiling information to use). Of the multiple parameters that can be set in the DC file, the only one required is *`MAIN`*, pointing to the XML file that you want to process. Usually, you create one DC file per book or article. For a documentation `set` (a collection of books), multiple DC files

can be defined. This allows you to set different parameters and different values for individual books in the set.

Directory Structure

For DAPS to work out of the box, your XML files and images must be organized in a specific structure within your documentation directory. Example 1, “Directory Structure” (page 4) shows the required structure including the key files for a DAPS documentation project. You can also create multiple documentation directories for individual documentation projects, but they all need the substructure outlined below.

Example 1 *Directory Structure*

```
YOUR_DOC_DIR/❶
  |--DC*❷
     |--images/
     |    |--src/❸
     |    |  |--dia/
     |    |  |--eps/
     |    |  |--fig/
     |    |  |--pdf/
     |    |  |--png/
     |    |  |--svg/
     |--xml/❹
     |    |--MAIN*.xml❺
```

❶ “Working directory” for the respective documentation project.
❷ DC file defining the documentation project.
❸ Top-level directory for any original images that you want to use in the documentation project. Contains subdirectories for images in various formats. Any images to be referenced in the XML sources must be put in the respective subdirectories. For information about referencing images, refer to Section “Referencing Images” (page 6).
❹ Directory holding the XML files for the documentation project. If you declare entities in an external file (for example, in `entity-decl.ent`), put the entity declaration file here, too.
❺ The Main file of the documentation project. It contains “references” to other books, chapters, appendices, etc.

A Documentation Project From Scratch

Use `daps-init` to set up a new documentation project from scratch. The initialization script automatically creates the key files and directory structure you need to get started with DAPS. View the available options with the command `daps-init --help`. By default, the script creates a `book` as example document.

Procedure 2 *Using daps-init*

1. Create a directory that you want to use as your documentation directory:

   ```
   $ mkdir DOC_DIR
   ```
2. Create the default directory structure for DAPS (containing an example `book`):

   ```
   $ daps-init -d PATH_TO_DOC_DIR
   ```

   With the `-r` option, you can also set another root element, such as `article`, for example. To create the default directory structure with an `article` example:

   ```
   $ daps-init -d PATH_TO_DOC_DIR -r article
   ```
3. In case the specified directory does not exists, DAPS asks if to create it. Proceed by pressing `y` or `n`.
4. To see what the output of the XML example file looks like, follow the instructions on the screen. For example, the following command specifies the `DC*` file (`DC-daps-example`) to use and the output format (color PDF) to create:

   ```
   $ daps -d PATH_TO_DOC_DIR/DC-daps-example color-pdf
   ```

   For creation of the PDF, DAPS uses FOP by default (if no other formatter is specified) and applies the default DocBook stylesheets (if no custom layout options are defined). At the end of the transformation

process, DAPS shows a message where to find the generated PDF. By default, all contents generated by DAPS is located in the `build` subdirectory. It is automatically created within your documentation directory.

5. Check your documentation directory for the new files: The text file `DC-daps-example` is annotated and gives you a rough idea which options can be defined in a DC file. For having a look at the XML source code of the `book` or `article`, change to the `xml` subdirectory and open the file `MAIN-DC-daps-example.xml` in a text or XML editor.

Editing DocBook XML Files

As DAPS does not include any editor software, you are completely free in the choice of your XML editor. Basically, you can use any text editor, but it is helpful if the editor supports editing XML in accordance with the DTD you use. A number of open source editors can be extended with plug-ins for automatic tag insertion and completion, insertion of `xref` elements and for checks if the XML document is well-formed. If you are already familiar with vi or Emacs, you can configure them to support XML editing mode. If your prefer an editor with a graphical user interface, jEdit [`www.jedit.org/`] is a good choice.

If you already worked with DocBook, you know about the typical top-level elements (or root elements) for documents: `book` or `article`. For larger documentation projects, another typical root element is `set` (a collection of books).

To define the individual components of a book, you use structural elements such as `part`, `chapter`, `preview` or `appendix`. Chapters are usually subdivided into sections (`section` elements or `sect1`, `sect2` etc.). Smaller structural units are `para` (for paragraphs), or list elements such as `orderlist`, `itemizedlist`, or `variablelist`.

If you have set up your documentation project from scratch with `daps-init`, you can explore the example documents that are installed within the directory structure. They show the most commonly used DocBook XML constructs.

Image Handling

Depending on the output format you generate with DAPS (PDF or HTML for example), the source images you provide and reference in your XML sources are automatically transformed into the appropriate output formats. For example, SVG images are converted to PNG for HTML builds, or color images to grayscale for black-and-white PDFs.

Supported Image Types

DAPS supports the following types of images:

- `DIA`
- `EPS` (experimental)
- `FIG`
- `PDF` (experimental)
- `PNG`
- `SVG`

Location of the Images

DAPS requires a certain directory structure for your XML sources and for your images that you want to reference in the XML files. For details, refer to Directory Structure (page 4). Depending on the file type of your source images, add them to the respective subdirectories in *`YOUR_DOC_DIR`* `/images/src` .

IMPORTANT: Unique Image Names

The base names of your images need to be unique throughout the image subdirectories. DAPS processes images in parallel processes. For example, if both `images/src/svg/AAA.svg` and `images/src/png/AAA.png` exist, it is unpredictable if the SVG or the PNG will appear in your output document. Therefore, use different base names: `images/src/svg/AAA.svg` and `images/src/svg/BBB.png` .

To search for common image names, use the `warn-images` subcommand. It lists all images with non-unique base names for a particular documentation project.

```
$ daps -d PATH_TO_DOC_DIR/DC-FILE warn-images
```

Referencing Images

Provided your images are located in the required default directory, DAPS automatically finds the path to your images. Therefore referencing images in your XML sources is very straightforward: you do not need to include any path in the `fileref` attribute—the file name is enough.

Furthermore DocBook allows you to reference more than one image to distinguish between different output formats. For example, you can add two references pointing to the same file, but using different images widths for PDF and HTML output. Use the `role` attribute to specify the output format, for example `fo` or `html`. See Example 2, “Image Reference in an XML File” (page 6).

Example 2 *Image Reference in an XML File*

```
<figure>
  <title>Main Window</title>
  <mediaobject>
    <imageobject role="fo">
      <imagedata fileref="screenshot.png" width="70%"/>
    </imageobject>
    <imageobject role="html">
      <imagedata fileref="screenshot.png" width="75%"/>
    </imageobject>
  </mediaobject>
</figure>
```

Basic DAPS Syntax

Before introducing the key `daps` commands to create output formats from your XML documents, let's get familiar with the basic syntax of the `daps` command:

```
$ daps [--global-options] subcommand [--command-options] [arguments]
```

Example 3, “DAPS Syntax” (page 6) shows an example command that generates HTML output. Global options are used to specify the level of verbosity, and the Doc Config file for creating the output.

Example 3 *DAPS Syntax*

```
daps❶ --debug❷ -d❸ DC-daps-example html❹ --static❺
```

❶ Main command: `daps`
❷ Global Option `--debug`: Sets the highest verbosity level (number of messages shown during the transformation process from XML to HTML).
❸ Global Option `-d`: Defines the relative or absolute path to the Doc Config file. In this example, `daps` is called in the same directory that holds the Doc Config file.
❹ Subcommand `html`: Defines the output format to create.
❺ Command option `--static`: Tells DAPS to copy CSS and image files to the same location like the HTML files. For more information, see Table 1, “DAPS Output Commands and Formats” (page 7).

Generally, DAPS can be executed with or without options. To view the global options and the available subcommands, use the command:

```
$ daps help
```

For a short help text on a specific *subcommand*, use:

```
$ daps help subcommand
```

For example, if you want more information about generating HTML output, run:

```
$ daps help html
```

The following section introduces the key `daps` commands needed to create output formats from XML files. All examples are based on the example files that are provided by the DAPS initialization script. For more information, refer to Section “A Documentation Project From Scratch” (page 4).

Output Formats

DAPS supports a number of different output formats, including “exotic” formats like man pages or ASCII text.

By default, the DocBook stylesheets are used for generating output formats. But DAPS also allows you to customize your output formats in a very flexible way.

Validation

Generating any output requires that your XML files are well-formed and can be validated. As soon as any output command is executed, DAPS automatically runs a validation check first. If it fails, DAPS returns the parser errors, including information about the type of error, the respective file name and the line number where the error occured.

Example 4 *Parser Output For Validation Errors (link to unknown ID)*

```
daps_user_concept.xml:60: element xref: validity error:
IDREF attribute linkend references an unknown ID "itl.daps.user.inst.other.req"
Document /local/svn/daps-svn/daps/doc/build/.profiled/x86-amd64-em64t_osuse_/
MAIN.DAPS.xml does not validate
make: *** [validate] Error 3
```

Basic Syntax for Generating Output

Independent of the individual output format you want to create, you need to specify the Doc Config file to use:

```
$ daps -d DC_FILE OUTPUT_FORMAT
```

For example:

```
$ daps -d DC-daps-example color-pdf
```

If your current directory is not the documentation directory where the DC file is located, also specify the (absolute or relative) path to the DC file. For example:

```
$ daps -d /svn/daps/example/DC-daps-example color-pdf
```

At the end of the transformation process, DAPS shows a message where to find the generated output.

Generating Different Output Formats

The following table gives an overview of the DAPS subcommands for generating output formats.

Table 1 *DAPS Output Commands and Formats*

Subcommand	Output	Note
color-pdf	Creates a color PDF (without any crop marks). Open the result in a PDF viewer.	Requires an FO formatter.
pdf	Creates a black-and-white PDF with crop marks, suitable for hand-off to a printing shop. Open the result in a PDF viewer.	Requires an FO formatter. Creation of crop marks is currently only supported by the XEP FO formatter. All color images are automatically converted to grayscale images.
html	Creates a subdirectory containing individual HTML files for all chapters of a book (including also preface, glossary or appendix files). Open the generated `index.html` file in a Web browser to view the generated HTML from the starting point (ROOTID of the top-level element).	Images and CSS files are only linked in the resulting directory that contains the HTML files. To copy these files to the same location like the HTML files, use the `--static` option. This is useful for creating distributable HTML builds.
html-single	Creates a single HTML file, named after the DC file used to create the	Single HTML files are more convenient for full text searches. Images and CSS

Subcommand	Output	Note
	output. Open the generated `*.html` file in a Web browser.	files are only linked in the resulting directory that contains the HTML files. To copy these files to the same location like the HTML files, use the `--static` option. This is useful for creating distributable HTML builds.
epub	Creates an ePUB document. Open the result in an portable e-book reader (or with a software like Calibre).	
webhelp	Creates a DocBook Web Help output. Open the generated `index.html` file in a Web browser to view the generated HTML from the starting point (ROOTID of the top-level element).	Experimental feature. Requires a very recent version of the DocBook stylesheets. DocBook Web Help consists of HTML pages with an additional pane, featuring a table of contents and a search function. The table of contents can be expanded and collapsed and is automatically synchronized with the contents pane. The search function weights the search results so that the most relevant results are listed first.
txt	Creates an ASCII text output. Open the result in a text editor.	All images are removed from the output, but their location is indicated in the text by the respective image base name printed in square brackets. A table of contents is automatically generated and is available at the beginning of the text document.
man	Creates one or multiple man pages.	To create man pages, your XML files must contain at least one `refentry`— be it in a `chapter`, `appendix`, or collected in a `reference` element. When processing a DocBook document with multiple `refentry` elements (regardless where they appear), DAPS generates one man page file per `refentry` element. All other parts of the document will be ignored.
jsp	Creates Java Server Pages (JSP files). The generated `jsp` subdirectory contains individual JSP files for all chapters of a book (including also preface, glossary or appendix files).	A JSP file is an HTML page containing a reference to Java servlets or Java Server Side Applets.
wiki	Creates a MediaWiki output in a single file. Open the result in a text or MediaWiki editor, and use it as raw input for a MediaWiki article.	Experimental feature. The output is very basic: Section titles are converted in headlines according to their hierarchy, itemized lists, ordered lists and `screen` elements are converted to the respective MediaWiki elements.

Advanced Output Options

Instead of always building your complete documentation project (`set`, `book`, or `article`), DAPS also allows you to build an individual `chapter` or `part`. If you want the output format to include `remark` elements or to be marked as a `draft` version, you can tell DAPS to do so.

Partial Builds

The "starting point" of your documentation project is usually the root element defined in the Main file that is referenced in the respective Doc Config. Alternatively, specify a ROOTID on the command line by using the `--rootid` option. This also allows you to build only a part of your documentation project by using the ID of a `book`, `article`, `glossary`, `appendix`, `part`, or `chapter` element.

For example, if you have set up your working environment with `daps-init` and an example `book`, use the following command to build the first chapter of the book (without the book's title page, table of contents etc.):

```
$ daps -d DC-daps-example color-pdf --rootid=cha.template.examples
```

Output with Remarks or Draft Watermark

For publishing a pre-release version of a document that you might want to send to a proofreader for review, use the `--draft` command option to mark the document as draft version. For example, use the following command to build a color PDF that has a `DRAFT` watermark printed on each page:

```
$ daps -d DC-daps-example color-pdf --draft
```

If you used `remark` elements in your XML files (for editorial remarks or questions to the proofreader), include the remarks in the output with the `--remarks` option:

```
$ daps -d DC-daps-example color-pdf --remarks
```

By default, the content of remark elements is shown in italics in the output format. Enabling remarks automatically turns draft mode on.

> **NOTE: Output with Remarks or Draft Watermark**
> The command options `--draft` option and `--remarks` are supported for HTML and PDF, but not for all DAPS output formats.

By default, DAPS adds a string to the base name of the output file to flag output formats generated with special options. Example file names are `daps-example_draft_en.pdf` or `daps-example_remarks_draft_en.pdf` .

Migration of Existing DocBook Projects

This section provides instructions how to migrate existing DocBook projects so that you can use DAPS for managing and publishing them.

Procedure 3 *Making DocBook Projects Compatible with DAPS*

1. If your XML files are distributed across several subdirectories, flatten the hierarchy and put all XML files directly into the `xml` subdirectory that is required by DAPS. See Directory Structure (page 4). Hosting multiple documentation projects in the same `xml` directory is fine as long as the file names are unique. You can put multiple Main files there.
2. If you have any XIncludes or entity declaration files, also put them into the `xml` subdirectory.
3. Depending on the file type of your source images, add them to the respective subdirectories in *`YOUR_DOC_DIR`*`/images/src` . The `image` directory and its substructure is required by DAPS. For details, refer to Directory Structure (page 4).
4. Make sure that the base names of your image files are unique. For details, refer to Section "Location of the Images" (page 5).
5. Adjust all references of image files, Xincludes, and entity declarations, in the existing XML files to match the structure required by DAPS. The references must not include any absolute or relative path, the plain file name is enough.
6. For each deliverable (book, article, set) that you want to generate from your XML files, create a Doc Config file. For more information, refer to Section "Key Files" (page 3). Find a template for DC files in your installed system in `/usr/share/daps/init_templates/DC-file.template` .

If you already used DAPS' predecessor `susedoc`, use the `/usr/bin/daps-envconvert` script for migrating your ENV files to DC files. For a short overview of the main changes, refer to `/usr/share/doc/packages/daps/README.upgrade_from_susedoc_4.x`. In contrast to susedoc, DAPS uses the DocBook layout by default. The SUSE-stylesheets have been moved to a separate package, `suse-xsl-stylesheets`. It is available from the Documentation:Tools repository. If you want to continue using the SUSE-layout for your documentation projects, install this package in addition to DAPS. To make DAPS use the SUSE layout, adjust the *`STYLEROOT`* parameter in the DC files of your documentation projetcs.

DAPS Configuration

DAPS can be customized to a large degree: per system, per user, and per document. The configuration file `/etc/daps/config` lists all parameters that can be configured, including a short description for each parameter. Parameters are always defined as `KEY="VALUE"` pairs. Any parameter can be set in various locations, which are listed below in ascending order with regards to their hierarchy. If conflicting values are set for the same parameter, the value defined in the next higher hierarchy level takes precedence. Values defined on the command line always take precedence over values set in any other locations.

- `/etc/daps/config` (system-wide configuration file)
- `~/.daps/config` (user-specific configuration file)
- DC (doc config) file of the documentation project (for settings specific to a document or documentation set)
- on the fly at the command line by specifing options to a `daps` command.

For More Information

This guide gave you a short introduction toDAPS and guided you through the key tasks. To discover more, refer to the other manuals available on DAPS at `https://sourceforge.net/p/daps/home/Home/` .

For general discussions and technical support, join the forum at `https://sourceforge.net/p/daps/discussion/` . You do not need a user account at `sourceforge.net` for this.

For bugs or enhancement requests, please open a ticket at `https://sourceforge.net/p/daps/tickets/add` . A user account at `sourceforge.net` is recommended, but you may also open tickets anonymously.

For a complete DocBook reference see *DocBook: The Definitive Guide* [`http://www.docbook.org/tdg/en/html/docbook.html`].

If you encounter problems with DAPS, check *Troubleshooting* (↑User Guide) for a list of common problems and their solution.

GNU Free Documentation License

Version 1.2, November 2002

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 noncommercially. 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 in order 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.

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 formats 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.

VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, 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.

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.

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.

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".

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.

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.

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.

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.

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 http://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

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

```
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.

Created by SUSE® with XSL-FO