Quick Start

DAPS 3.1

Stefan Knorr, Tanja Roth, Manuel Schnitzer, and Christopher-Julian Zwickl

DAPS (DocBook Authoring and Publishing Suite) helps technical writers to author and publish documentation written in DocBook XML. DAPS is a command line based software for Linux* and released as open source.

The DAPS Quick Start is a short introduction to DAPS for technical writers. It includes step-by-step instructions for key editing and publishing tasks.

Publication Date: May 11, 2021

Contents

1 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 and of using the Bash Shell (or command line interfaces in general) is required.

2 Supported DocBook Versions

DAPS supports DocBook 4.x and DocBook 5.x.

3 System Requirements

DAPS itself is a lean solution that does not require a lot of system resources. However, it does use components that may need a stronger processor and more RAM, for example, for creating PDF output files. Although not required, DAPS benefits from multi-core processors.

3.1 Hardware Requirements

RAM

> The required amount of RAM mostly depends on the volume of your documentation projects. For creation of PDF output, at least 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.

3.2 Software Requirements

DAPS runs on any modern Linux system. It has not been attempted to port DAPS to Windows* or Mac OS X* yet.

When installing DAPS as an RPM package (on any SUSE-based system), dependencies on other software packages are automatically resolved during installation. No additional action is required.

For installing DAPS from the sources on other Linux distributions, refer to https://github.com/openSUSE/daps/blob/main/INSTALL.adoc ↗ where the respective requirements are covered in detail.

3.3   Additional Software

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) ↗. The FO formatter 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.

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.

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.

3.4   Directory Structure

For DAPS to work out of the box, it requires a certain organization of your DocBook XML files and images within your documentation directory. For details, refer to *Section 5.1, "Directory Structure"*. 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 12, "Migration of Existing DocBook Projects"*.

4 Installation

The DocBook Authoring and Publishing Suite can be installed and used on any Linux distribution. Currently, DAPS is available as an RPM package for the openSUSE distribution and for SUSE Linux Enterprise products. Eventually, packages for other distributions may become available. For the latest status update and installation instructions, refer to https://github.com/openSUSE/daps/blob/main/INSTALL.adoc ↗.

4.1 Installing DAPS on openSUSE

There are a few ways to install DAPS on openSUSE. To always stay up-to-date with the latest version of DAPS install the `daps` package from the `Documentation:Tools` repository as outlined below.

You may also use the `daps` package that shipped with your version of openSUSE. However, you then might miss the latest features and bug fixes in DAPS.

The quickest way to install DAPS is using the **`zypper`** command.

PROCEDURE 1: INSTALLING DAPS VIA ZYPPER FROM DOCUMENTATION:TOOLS

1. Open a browser and enter the following URL: http://download.opensuse.org/repositories/Documentation:/Tools ↗

2. Select your distribution and product number to make the browser show the URL for the respective repository.

3. Copy the URL from the address bar.

4. Open a terminal.

5. Add the repository with the following zypper command:

   ```
   root # zypper ar -f URL Documentation:Tools
   ```

   Replace _URL_ with the URL you pasted from your browser.

6. Install DAPS with the following zypper command:

   ```
   root # zypper in --from Documentation:Tools daps
   ```

   In order to install DAPS you have to trust the `Documentation:Tools` repository.

4.2 Installing DAPS on SUSE Linux Enterprise

Starting with SUSE Linux Enterprise 12, DAPS is also available for SUSE Linux Enterprise. The DAPS package is provided by the SUSE Software Development Kit (SDK), a free extension for SUSE Linux Enterprise. You need to install it as add-on (or extension). You can install the SDK (without any physical media) as an extension after the registration of your system at SUSE Customer Center. If you prefer to install from a physical medium, proceed as follows:

1. Download the installation media for the SDK from http://download.suse.com/ ↗.

2. Install the SDK as an add-on product to SUSE Linux Enterprise.
   For details on how to install add-on products (with or without physical media), see the *SUSE Linux Enterprise 12 Deployment Guide*, available at http://www.suse.com/documentation/ ↗. Refer to chapter *Installing Add-On Products*, https://www.suse.com/documentation/sles-12/book_sle_deployment/data/cha_add-ons.html ↗.

3. Install the `daps` package, using either the YaST *Software Management* module or the following command:

   ```
   root # zypper in daps
   ```

   Dependencies on other software packages are automatically resolved during installation.

4.3 Installing DAPS on Other Linux Distributions

For the latest status update and installation instructions, refer to https://github.com/openSUSE/daps/blob/main/INSTALL.adoc ↗.

4.4 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 the hyphenation pattern files. Using the hyphenation patterns is recommended.

5 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*. The script automatically creates the *Directory Structure* and *Key Files* that you need to get started with DAPS.

5.1 Directory Structure

DAPS requires your XML files and images to be organized in a specific structure within your documentation directory. *Example 1, "Directory Structure"* 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/
    |       |--jpg/
    |       |--pdf/
    |       |--png/
    |       |--svg/
  |--xml/ ❹
      |--MAIN*.xml ❺
      |--*.xml
```

❶ "Working directory" for the respective documentation project (in the following also called `project directory` or `documentation directory`).

❷ Doc Config file (or files) defining the documentation project (books, articles).

❸ Top-level directory for any original images that you want to use in the documentation project. This directory 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 9.4, "Referencing Images"*.

④ Directory holding the XML MAIN file and all other XML files for the documentation project. If you declare entities in one or more external files (for example, in `entity-decl.ent`), put the entity declaration files here, too.

⑤ The MAIN file of the documentation project. It contains the "starting point" (the highest-level object) of your documentation project and includes "references" to other books, chapters, appendixes, etc. For more information, refer to *Section 5.2, "Key Files"*.

> **❗ Important: Structure of the `xml` and `image/src/*` Directories**
>
> Avoid subdirectories within the `xml` and `image/src/*` directories. Referencing or including files from subdirectories within those directories can lead to unpredictable results with DAPS.

5.2 Key Files

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

MAIN File

A DocBook XML file in the `xml` directory. It contains 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 file name as well. Other XML files may be included into the MAIN file via `<xi:include/>` statements.

Doc Config (DC) File

A configuration file in the project directory. It defines several 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.

5.3 File Name Requirements

XML extension required

Always use the .xml extension, so DAPS can recognize the files, which are needed to build the document.

No Spaces and Colons In File and Directory Names

Avoid spaces and colons in file and directory names. The `make` command in DAPS has trouble understanding them. Use underscores (`_`) or hyphens (`-`) instead.

It is good practice to only use the following characters for file or directory names: alphabetic characters [a-z] or [A-Z], numerical characters [0-9], hyphens (`-`), or underscores (`_`).

Unique Image Names

Always store just one file with a particular name within the `images/src` directory of a project. As DAPS tries to create any missing image formats from original images, it will otherwise not know which one of the duplicate files to use for converting to the missing formats.

Additionally, having a file called `example.png` and another called `example.svg` in the same documentation project will often lead to questions like: Which file to use where? Do both files display the same content? Are both files current, or is one outdated?

When invoking DAPS with the parameter `-v`, a warning will be printed whenever a file name appears twice within a project. To specifically check for image name clashes upfront, use the `daps list-images-multisrc` subcommand.

It is a good idea to find a consistent file naming scheme. For example, when documenting software, it might prove helpful to include the name of the application at the beginning of the file name. You can also use prefixes like `screenshot_` and `diagram_` to separate between different types of images.

> ### Note: Hint on Using UTF-8 characters in File Names
>
> Depending on your environment, the build process could fail if you are using file names with UTF-8 characters. Run the `locale` command and check if the LANG environment variable contains UTF-8. Otherwise adjust the LANG environment variable accordingly.

6 A Documentation Project From Scratch

Use **daps-init** to set up a new documentation project from scratch. The init script automatically creates the key files and directory structure you need to get started with DAPS. By default, it creates a DocBook article as example file. By adding options you can modify parameters (such as changing the root element to book or specifying the DocBook version in which you want the example to be generated). View the available options with **daps-init -h**.

PROCEDURE 2: USING **daps-init**

1. To create a working environment for DAPS , including an example document, enter the following:

   ```
   tux:~> daps-init --docdir PATH_TO_DOC_DIR
   ```

   Specifying the project directory with --docdir is mandatory. If the directory does not exist, DAPS prompts you to create it.
   If you want to modify the file name and the title for the document, use the options --name and --title:

   ```
   tux:~> daps-init --docdir PATH_TO_DOC_DIR \
       --name "my_document" \
       --title "Example Documentation"
   ```

2. If you want to create an example book file in addition to the article, enter the following:

   ```
   tux:~> daps-init
       --docdir PATH_TO_DOC_DIR --rootelement book
   ```

3. To see what the output of the XML example file looks like, follow the instructions on the screen.
   At the end of the transformation process, DAPS shows a message where to find the generated output file. By default, all contents generated by DAPS is located in the build subdirectory. It is automatically created within your project directory.

4. Check your project directory for the new files: The text file DC-* is annotated and gives you a general idea which options can be defined in a DC file. For having a look at the XML source code of the example document, change to the xml subdirectory and open the file MAIN-*.xml in a text editor or XML editor.

7 Editing DocBook XML Files

As DAPS does not include any editor software, you are completely free in the choice of your XML editor. While you can use your text editor of choice, it is helpful if the editor supports editing XML in accordance with the schema you use. Several 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 you prefer an editor with a graphical user interface, jEdit (http://www.jedit.org/) ↗ is a good choice.

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

To define the individual components of a book, use structural elements such as `part`, `chapter`, `preface` 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.

8 Validation

Generating any output requires that your XML files are valid. 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 occurred. In addition, DAPS shows the path to the profiled XML sources and the total number of errors.

EXAMPLE 2: PARSER OUTPUT FOR VALIDATION ERRORS (`xref` 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
```

> **Note: Validating is Done in Build Directory**
>
> Validation is always done in the `build` directory and in the profiled sources, as indicated by the path above (`/local/[...]/build/.profiled/[...]/MAIN.DAPS.xml`). However, you need to fix the validation error in the sources located in your `xml` directory, otherwise the fixes will not take effect. Profiling is similar to conditional text. For details, refer to the chapter about modularizing document projects in the DAPS User Guide.

9   Image Handling

Depending on the output format you generate (PDF or HTML, for example), DAPS automatically transforms the source images you provide (which are also referenced in your XML sources) 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. You only need to decide which file format to use as source format. Of course, this decision depends on the purpose of the image. For more details, see the DAPS User Guide.

9.1   Supported Image Types

DAPS supports the following types of images:

- DIA (input format only)

- EPS (experimental)

- FIG (input format only)

- JPEG

- PDF (experimental - only works for PDF output and with XEP formatter)

- PNG

- SVG

9.2 Organization of the `images` Directory

DAPS requires you to use a specific directory structure for images. All images that you reference from your DocBook files must be stored in a subdirectory of the project directory named `images/src/file_extension`. For example, PNG files must be stored under `images/src/png`. If you used **daps-init** to set up your project, the appropriate directories should already exist. For a more detailed reference to the directory structure, see *Section 5.1, "Directory Structure"*.

9.3 Image File Name Requirements

For details, see *Section 5.3, "File Name Requirements"*.

9.4 Referencing Images

As your images need to be located in a defined directory structure, DAPS automatically finds the path to your images. Therefore, referencing images in your XML sources is very straightforward: you must not 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`.

EXAMPLE 3: IMAGE REFERENCE IN AN XML FILE

Let us assume you have a source image file named `graphic.dia`. To make DAPS use an SVG version of your image for PDF output (`role="fo"`) and a PNG version for HTML output (`role="html"`), reference the images as follows:

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

10 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:

```
tux:~> daps [--global-options] subcommand [--command-options] [arguments]
```

Example 4, "DAPS Syntax" 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 4: DAPS SYNTAX

```
daps ❶  --debug ❷  -d ❸ DC-daps-example html ❹  --static ❺
```

❶ Main command: **daps**

❷ Global Option `--debug`: Sets the highest verbosity level (number of messages shown during the conversion 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`: Instructs 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"*.

💡 Tip: Specifying the DC File

For execution of most commands, DAPS needs to know which DC file to use. Usually, you do so by specifying a DC file with the global option `-d`. For example:

```
tux:~> daps -d PATH_TO_DC_FILE pdf
```

Only in the following cases you may omit the `-d` option:

- If your documentation directory contains only one DC file. In that case, DAPS automatically uses the corresponding file.

- If you have specified a default DC file to use in `~/.config/daps/dapsrc` (as a value for `DOCCONF_DEFAULT`). In that case, DAPS automatically uses the corresponding file, unless you specify a different one on the command line.

- If you want to call the help function.

To view the global options and the available subcommands for DAPS, use the command:

```
tux:~> daps help
```

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

```
tux:~> daps help  subcommand
```

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

```
tux:~> daps help html
```

The following section introduces the key `daps` commands for generating output formats from XML files. All examples are based on the files generated by the DAPS init script. For more information, refer to *Section 6, "A Documentation Project From Scratch"*.

11   Output Formats

By default, the DocBook stylesheets are used for generating output formats. But DAPS also allows you to easily customize your output formats. For more details, refer to *Book "User Guide", Chapter 9 "Customizing Layout of the Output Formats"*.

11.1   Basic Syntax for Generating Output

DAPS supports various different output formats, including also "exotic" formats like man pages or simple text. *Table 1* gives an overview.

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

```
tux:~> daps -d PATH_TO_DC_FILE OUTPUT_FORMAT
```

For example:

```
tux:~> daps -d DC-daps-example pdf
```

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

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:

```
tux:~> daps -d /svn/daps/example/DC-daps-example pdf
```

11.2 Generating Different Output Formats

The following table lists the main output formats and their characteristics, and the DAPS subcommands to generate them. Refer to *Section 11.1* for the commands' basic syntax.

TABLE 1: DAPS OUTPUT COMMANDS AND FORMATS

Subcommand	Output/Note
pdf	Creates a color PDF. Open the result in a PDF viewer. Requires an FO formatter.
pdf --grayscale	Creates a black-and-white PDF. Open the result in a PDF viewer. Requires an FO formatter. All color images are automatically converted to grayscale images. If you need a PDF for a printing shop, add the `--cropmarks` option. Creation of crop marks is currently only supported by the XEP FO formatter.
html	Creates a subdirectory containing individual HTML files for all chapters of a book (including also preface, glossary or appendix files). The HTML files are named according to the ID of the

Subcommand	Output/Note
	respective root element. 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 as 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 output. Open the generated `*.html` file in a Web browser. Single HTML files are more convenient for full text searches. 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.
epub	Creates an EPUB 2 document. Open the resulting file in a portable e-book reader (or with a software like Calibre). If you need an EPUB 3 document, add the `--epub3` option.
mobi	Creates an Amazon Kindle e-book in Mobipocket format. Open the resulting file in a portable e-book reader (or with a software like Calibre). Requires Calibre. DAPS first generates an EPUB file which is then converted to `*.mobi` format with Calibre.
webhelp	Creates a DocBook Web Help output. Open the resulting `index.html` file in a Web browser to view the generated document from the starting point (ROOTID of the top-level element). Experimental feature. Requires the most 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

Subcommand	Output/Note
	collapsed, and is automatically synchronized with the contents pane. The search function orders the search results so that the most relevant results are listed first.
text	Creates an ASCII text output. Open the resulting file 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 inserted 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.

The number of output formats may be extended in the future, depending on the output formats that are supported by DocBook stylesheets. For an overview of all output formats, run `daps help`. The available output formats are listed below *Subcommands › Generate Books*.

DAPS allows you to fine-tune the output in many ways . For example, you can include remarks or a `DRAFT` watermark in your output, or you can build parts of your documentation project only. Find some examples in the sections below.

11.3 Partial Builds

Instead of always building your complete documentation project (`set`, `book`, or `article`), DAPS also allows you to build only individual parts. The "starting point" of your documentation project is usually defined by the root element of the MAIN file that is referenced in the respective Doc Config. To build only a part of your documentation project, use the `--rootid` option to specify the ID of an individual `book`, `article`, `glossary`, `appendix`, `part`, or `chapter`.

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:

```
tux:~> daps -d DC-daps-example pdf --rootid=cha.template.examples
```

11.4 Output with Remarks or Draft Watermark

For publishing a pre-release version of a document that you can send to a proofreader for review, use the `--draft` option to mark the document accordingly. For example:

```
tux:~> daps -d DC-daps-example pdf --draft
```

This command creates a color PDF that has a `DRAFT` watermark printed on each page.

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

```
tux:~> daps -d DC-daps-example pdf --remarks
```

When generating PDFs with FOP, the contents of the remark elements is shown in italics within the text. XEP supports conversion of remark elements into PDF annotations. This feature is enabled in DAPS by default, but if you want XEP to treat remark elements like FOP does, you can change the respective DAPS parameter. In HTML, HTML-single and EPUB output, the contents of the remark elements is shown in red within the text.

> 📝 Note: Availability of Advanced Output Options
>
> Advanced output options are only supported for selected formats. For example, `--draft` and `--remarks` are only available in HTML, HTML-single, PDF, and EPUB output.
>
> Using the `--remarks` option automatically turns draft mode on.
>
> 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 `*_draft_en.pdf` or `*_remarks_*_draft_en.pdf`.

12 Migration of Existing DocBook Projects

To migrate existing DocBook projects so that you can manage and publish them with DAPS, follow the step-by-step instructions in *Book "User Guide"*, .

13 For More Information

This guide gave you a short introduction to DAPS and guided you through the key tasks. To discover more, refer to the other manuals available on DAPS at http://opensuse.github.io/daps/ ↗ .

We want to hear your comments and suggestions about DocBook Authoring and Publishing Suite (including this guide and the other documentation included with DAPS). You can contact us on the `#opensuse-doc` IRC channel on `irc.freenode.net` or in the discussion forum at https://github.com/openSUSE/daps/discussions ↗ . For bugs or enhancement requests, open an issue at https://github.com/openSUSE/daps/issues/new ↗ . A user account at https://github.com ↗ is needed.

Patches and user contributions are welcome!

For a complete DocBook reference, see *DocBook: The Definitive Guide* (http://www.docbook.org/tdg/en/html/docbook.html) ↗ .

For an overview of the key terms used in the context of DAPS and DocBook, refer to .

If you encounter problems with DAPS, check *Book "User Guide", Chapter 11 "Troubleshooting"* for a list of common problems and their solutions.

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 "Acknowledgments", "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 "Acknowledgments" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgments 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 "Acknowledgments", 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 "Acknowledgments", "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.