User Guide

Tanja Roth <ta-ro@opensuse.org>
Technical review: Thomas Schraitle, Frank Sundermeyer

User Guide

by Tanja Roth

Technical review: Thomas Schraitle, Frank Sundermeyer

Table of Contents

List of Examples

About This Guide

The DocBook Authoring and Publishing Suite (DAPS) is an open-source authoring and publishing environment for DocBook XML. It is command-line oriented and can be used on Linux operating systems. It consists of integrated tools that support technical writers in the editing, translation and publishing process of their XML documents.

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).

Available Documentation

This guide contains links to additional documentation resources. The following manuals are available for DAPS:

DAPS Quick Start Guide
: Short introduction to DAPS for end-users. Includes step-by-step instructions for the key tasks in editing and publishing DocBook documents.

User Guide
: Comprehensive guide for end-users. It guides you through creating, editing, managing and publishing your DocBook documents with DAPS—be it a short article by a single author or a larger documentation project, written by multiple authors.

Feedback

We want to hear your comments and suggestions about DocBook Authoring and Publishing Suite (including this guide and the other documentation included with DAPS).

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.

Patches and user contributions are welcome!

Documentation Conventions

The following typographical conventions are used in this manual:

- `/etc/passwd`: directory names and filenames
- *`placeholder`*: replace *`placeholder`* with the actual value
- `PATH`: the environment variable PATH
- **ls**, `--help`: commands, options, and parameters

- `user`: users or groups
- **Alt**, **Alt**+**F1**: a key to press or a key combination; keys are shown in uppercase as on a keyboard
- File, File+Save As: menu items, buttons
- *Dancing Penguins* (Chapter *Penguins*, #Another Manual): This is a reference to a chapter in another manual.

About the Making of This Document

This documentation is written in DocBook (see http://www.docbook.org) and edited and generated with the open-source tools provided by the DocBook Authoring and Publishing Suite. The XML source files were validated by **xmllint**, processed by **xsltproc**, and converted into XSL-FO using a customized version of the DocBook stylesheets. The final PDF is formatted through XEP.

Chapter 1. System Requirements and Installation

DAPS is a lean solution that does not require a lot of system resources. It can be installed on any Linux distribution. For detailed system requirements and step-by-step installation instructions, refer to DAPS Quick Start Guide.

Chapter 2. Conceptual Overview

Supported DocBook Versions

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

Key Features

DAPS supports technical writers in the editing, translation and publishing process of DocBook XML files:

Output Formats

DAPS lets you publish your XML sources in a number of different output formats, for example: HTML, HTML-single, PDF, ePUB, Web Help, text, man pages, or MediaWiki. For details, refer to Chapter 4, *Generating Output Formats*.

Custom Layouts

By default, DAPS uses the DocBook stylesheets to generate the output formats. But DAPS also supports custom layouts for your documentation projects (or for individual books within your set).

Apart from that, DAPS allows you to change individual layout parameters by passing string parameters to xsltproc for HTML or PDF builds —without even touching the stylesheets. For details about custom layouts, refer to Chapter 8, *Customizing Layout of the Output Formats*.

Editor Macros

For Emacs, DAPS includes a set of macros for easy insertion of complex DocBook elements like `variablelist`, `figure`, `table` or `indexterm`. Instead of inserting the child elements successively, you will get a "skeleton" that includes all required child elements and is ready to be filled with contents. For details, refer to Chapter 3, *Editing DocBook-XML*.

Validating

Validating XML files within in a book or set exceeds validation of the current XML file, as links (`xref` elements) or XIncludes need to be resolved, too. With DAPS, you can check validity of all files that belong to a documentation project with a single command. For details, refer to Chapter 3, *Editing DocBook-XML*.

Spell Check

DAPS supports spell checking of your XML sources with aspell from the command line. Depending on the XML editor you use, you can also integrate a custom aspell dictionary into your editor. For details, refer to Chapter 3, *Editing DocBook-XML*.

Link Checker

To make sure that all links in your XML sources are still available (and do not give a `404` error or similar), DAPS also includes a link checker (based on `checkbot`). Use it to create a report of all links that caused some kind of warning or error. For details, refer to Chapter 3, *Editing DocBook-XML*.

Image Handling

DAPS provides sophisticated image handling support. For example, it can transform images referenced in your XML files into different formats, list all source images referenced in your XML files, list any missing images or list the *generated* images used for the various output formats. You can also forward those lists to your preferred image viewer to conveniently browse through the images, or check if all image names are unique. For details, refer to Chapter 5, *Image Handling*.

Profiling (Conditional Text)

If you have similar products to document and want to generate multiple documentation variants from your XML files, you can do so with the help of conditional text (or `profiling`, as it is called in DocBook). For example, you can profile certain parts of your XML texts for different (processor) architectures, operating systems, vendors or target groups. Use the `PROF*` keys defined in `/etc/daps/config` to define which information should be included in the output. For details, refer to Chapter 3, *Editing DocBook-XML*.

Dynamic Product Names and Numbers

DAPS allows to set product name and product number dynamically. This enables you to use the same XML sources for different products. Use the entities &productname; and &productnumber; in your XML sources, and define the respective values in an entity (using profiling for multiple products). For details, refer to Chapter 3, *Editing DocBook-XML*.

Review and Translation Processes

DAPS offers a number of features to simplify review and translation processes. For example, you can insert `remark` elements in the source code and generate an output format that either includes or suppresses these remarks. You can also generate preview versions of your documentation with a `DRAFT` watermark appearing on the HTML or PDF output. If you use Docmanager in addition to DAPS, you can “flag” your XML files with meta-information (like workflow status). DAPS offers an option to also display this meta-information in the generated output. For handing over your files to review or translation, DAPS can create tarballs of the XML sources and graphics. For details, refer to Chapter 6, *Review and Translation Processes*.

Packaging and Deployment

For deploying the documentation as RPM packages and integrating it into KDE and GNOME desktop environments as well as into Web user interfaces (via JSP), DAPS offers a number of options to produce the corresponding output: For example, you can create source packages, HTML tarballs, color PDFs and desktop and document files with the **daps package-*** commands. For details refer to Chapter 7, *Packaging and Deployment*.

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.

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

To migrate existing DocBook projects so that you can manage and publish them with DAPS, follow the step-by-step instructions in Procedure 3, “Making DocBook Projects Compatible 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.

In the following sections, find examples for Main and DC files, together with background information on some key parameters that can be used in DC files. The examples are sorted according to use cases:

- Small documentation projects, consisting of Single Deliverables (Article or Book)
- Larger documentation projects, consisting of Multiple Deliverables: Articles or Books in a Set

Single Deliverables (Article or Book)

The most elementary case of a documentation project is probably a white paper or article. Typically, its contents is covered in a single XML file with `article` as root element. In this case, the single XML file would be the Main file as it specifies the highest-level object in your documentation project (`article`). Apart from document title and body, the file can contain other information such as legal notice, release information, author data etc. An article may be structured into sections (by use of `section` elements or `sect1`, `sect2` etc.).

Creating an Example Document

Using **daps-init** allows you to automatically set up an example article or book, together with a DC file. The examples below are based on the output of **daps-init**, but vary deliberately in some details to show key parameters that you might want to add or change.

Find a simple example in Example 2.1, “Main File of an Article (DocBook 4.x)”.

Example 2.1. Main File of an Article (DocBook 4.x)

```
<?xml version="1.0" encoding="UTF-8"?>
[...]

<article lang="en" id="art.template">
 <title>Article Template</title>
 <subtitle>generated by DAPS</subtitle>
 <articleinfo>
  <releaseinfo>Version 0.1</releaseinfo>
  <releaseinfo>Revision: 0</releaseinfo>
  <releaseinfo>
   Build Date: <?dbtimestamp format="B d, Y"?>
  </releaseinfo>
  <legalnotice>
   <para>
    <ulink url="http://www.gnu.org/licenses/fdl-1.3-standalone.html">
    GNU Free Documentation License</ulink>
   </para>
  </legalnotice>
 </articleinfo>
 <abstract>
  <para>
   You may use this file as a template. For a complete DocBook reference
   see <citetitle>DocBook: The Definitive Guide</citetitle>, available at
   <ulink url="http://www.docbook.org/tdg/en/html/docbook.html"/>.
  </para>
 </abstract>
 <sect1 id="sec.template.examples">
  <title>Examples: The most commonly used DocBook XML constructs</title>
  <para>
   I am a paragraph in a section 1.
  </para>
  <sect2 id="sec.template.examples.lists">
   <title>Lists</title>
   <para>
    This section 2 showcases 3 types of lists.
   </para>
   [...]
  </sect2>
 </sect1>
</article>
```

Let us assume the XML file shown in Example 2.1, “Main File of an Article (DocBook 4.x)” is named `MAIN-DAPS-example-article.xml` and you want to publish it in default DocBook layout, without any special requirements. To generate output, you usually create a DC file per article or book, specifying a number of parameters such as the Main file or which layout 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. Therefore, a very basic DC file for the article in Example 2.1, “Main File of an Article (DocBook 4.x)” could look as follows:

Example 2.2. Basic DC File for an Article

```
## Doc config file for the DAPS example article
## See /etc/daps/config for documentation of the settings below
##

## Mandatory Parameter
MAIN="MAIN-DAPS-example-article.xml" ❶
```

❶ Specifies the Main XML file. It contains the highest-level object (root element) of your documentation project. The Main file must be located in *YOUR_DOC_DIR*/xml/. Therefore, you only need to specify the Main's file name in the DC file (no path).

The example above is a bit artificial, though: If you do not want to specify any further parameters (apart from the Main file), you can also set the *MAIN* parameter as command line option during generation of the output format. In that case, you can do completely without DC file. For details, refer to Chapter 4, *Generating Output Formats*.

In case your documentation project does not consist of an article, but a single book, the basic setup of Main file and DC file is similar:

Example 2.3. Main File of a Book (DocBook 4.x)

```
<?xml version="1.0" encoding="UTF-8"?>
[...]

<book id="book.template" lang="en">
 <bookinfo>
  <title>Book Template</title>
  <subtitle>generated by daps</subtitle>
  <productname>Book Template</productname>
   <legalnotice>
   <para>
    <ulink url="http://www.gnu.org/licenses/fdl-1.3-standalone.html">
    GNU Free Documentation License</ulink>
   </para>
  </legalnotice>
 </bookinfo>
 <chapter id="cha.template.examples">
  <title>Examples: the most commonly used DocBook XML constructs</title>
  <abstract>
   <para>
    You may use this file as a template. For a complete reference on DocBook
    see <citetitle>&tdg;</citetitle>, available at
    <ulink url="http://www.docbook.org/tdg/en/html/docbook.html"/>.
   </para>
  </abstract>
  <para>
   I am a paragraph in a chapter.
  </para>
  <sect1 id="sec.template.examples.lists">
   <title>Lists</title>
   <para>
    This is a section 1.
   </para>
  </sect1>
 </chapter>
</book>
```

In the example above, the book contents is also covered in a single XML file with `book` as root element. In contrast to an article, books can have more structure levels: they are usually divided into `chapter` elements (that may contain sections and subsections) as outlined in Example 2.3, “Main File of a Book (DocBook 4.x)”. In addition to chapters, books may also contain other structure elements such as `preface`, `glossary`, or `appendix`. As additional structure level, books may also contain parts. For a complete reference, see *DocBook: The Definitive Guide*, available at http://www.docbook.org/tdg/en/html/docbook.html.

Let us assume the XML file shown in Example 2.3, “Main File of a Book (DocBook 4.x)” is named `MAIN-DAPS-example-book.xml` and you want to publish it in a custom layout. To generate output, you would create a DC file pointing to the Main file of the book, and additionally specifying a set of custom stylesheets.

Example 2.4. DC File For a Book with Custom Layout

```
## Doc config file for the DAPS example book
## See /etc/daps/config for documentation of the settings below

## Mandatory Parameter
MAIN="MAIN-DAPS-example-book.xml" ❶

## Optional Parameters

## Custom Stylesheets
## (if not defined the DocBook stylesheets will be used)

STYLEROOT="/usr/share/xml/docbook/stylesheet/custom/xslt" ❷
```

❶ Specifies the Main XML file. It contains the highest-level object (root element) of your documentation project. The Main file must be located in *`YOUR_DOC_DIR`*`/xml/`. Therefore, you only need to specify the Main's file name in the DC file (no path).

❷ For a custom layout, use the *`STYLEROOT`* parameter to specify the (absolute or relative) path to the directory containing the custom stylesheets. Using absolute paths is recommended for DC files.

Multiple Deliverables: Articles or Books in a Set

If your documentation project consists of multiple books in a `set`, the Main file is the one that contains the `set` element. In the following example, the components of the set (individual books) are not part of the Main file, but have been put into separate document files (`book*.xml`), that are then assembled in the Main file using XIncludes. Note that this is not specific for a set—it is mainly a means of modularizing your documents. You can also use XIncludes for splitting up books, articles or chapters into separate files. For more information, refer to *Physical Divisions: Breaking a Document into Separate Files* [http://www.docbook.org/tdg51/en/html/ch02.html].

Example 2.5. Main File of a Set (DocBook 4.x)

```
<?xml version="1.0" encoding="UTF-8"?>
[...]
```

```
<set lang="en">
 <title>DAPS Documentation</title>
 <xi:include href="book_daps_user.xml"
             xmlns:xi="http://www.w3.org/2001/XInclude"/>
 <xi:include href="book_daps_quickstarts.xml"
             xmlns:xi="http://www.w3.org/2001/XInclude"/>
 <!--<xi:include href="book_daps_developer.xml"
             xmlns:xi="http://www.w3.org/2001/XInclude"/>-->
</set>
```

For a documentation `set` (a collection of books), multiple DC files can be defined. This allows you to set different parameters and values for individual books in the set. By specifying a different *`ROOTID`* in the DC file, you define which book of the set is to be built. You can also specify different layouts or output modes (such as draft or annotated versions) for individual books in the same documentation set.

The following two DC files are those of the DAPS Quick Start Guide and the DAPS User Guide— both guides belong to the same documentation set, but use different layouts.

Example 2.6. DC File For a Book in a Set

```
## Doc config file for DAPS Quick Start Guide
## See /etc/daps/config for documentation of the settings below

## Mandatory Parameter
MAIN="MAIN.DAPS.xml" ❶

## Optional Parameters
## ROOTID
## If MAIN contains a set with several books and/or articles, use
## a separate DC-file for each book/article and set ROOTID to
## the id of the respective <book>/<article> element of the document
## This will enable you to build individual books/articles rather than
## the whole set
## See http://www.docbook.org/tdg/en/html/set.html for more information
## on sets
ROOTID="art.daps.quick" ❷

## Custom Stylesheets
## (if not defined the DocBook stylesheets will be used)

STYLEROOT="/usr/share/xml/docbook/stylesheet/suse/xslt/flyer" ❸
#FALLBACK_STYLEROOT="" ❹
HTML_CSS="./daps.css" ❺
EPUB_CSS="./daps.css" ❺
```

❶ Specifies the Main XML file. It contains the highest-level object (root element) of your documentation project. The Main file must be located in *`YOUR_DOC_DIR`*`/xml/`. Therefore, you only need to specify the Main's file name in the DC file (no path).

❷ Defines the root ID of the element to be used for creating an output format. Usually, you specify the root ID of a `book` or `article` element here.

In this example, `art.daps.quick` is the root ID of the DAPS Quick Start Guide, contained in `MAIN.DAPS.xml`.

❸ For a custom layout, use the *STYLEROOT* parameter to specify the (absolute or relative) path to the directory containing the custom stylesheets. Using absolute paths is recommended for DC files.

In this example, the parameter specifies the path to a directory with SUSE-specific stylesheets for the flyer layout that is used by the DAPS Quick Start Guide.

❹ Allows you to define a fallback which is used in case the custom stylesheets defined with *STYLEROOT* cannot be accessed. In case neither the stylesheets specified with *STYLEROOT* nor with *FALLBACK_STYLEROOT* can be accessed, DAPS uses the default DocBook layout.

In this example, no fallback is specified and the parameter is disabled.

❺ If not specified, DAPS will use the default DocBook stylesheets for production of HTML and ePUB. For custom CSS styles, specify the (absolute or relative) path to the respective CSS file. Using absolute paths is recommended for DC files.

In this example, custom CSS files are specified for both HTML and ePUB output.

Example 2.7. DC File For Another Book in the Same Set

```
## Doc config file for DAPS User Guide
## See /etc/daps/config for documentation of the settings below

## Mandatory Parameter
MAIN="MAIN.DAPS.xml" ❶

## Optional Parameters
## ROOTID
## If MAIN contains a set with several books and/or articles, use
## a separate DC-file for each book/article and set ROOTID to
## the id of the respective <book>/<article> element of the document
## This will enable you to build individual books/articles rather than
## the whole set
## See http://www.docbook.org/tdg/en/html/set.html for more information
## on sets
ROOTID="book.daps.user" ❷

## Custom Stylesheets
## (if not defined the DocBook stylesheets will be used)

STYLEROOT="/usr/share/xml/docbook/stylesheet/suse/xslt/" ❸
#FALLBACK_STYLEROOT="" ❹
HTML_CSS="./daps.css" ❺
EPUB_CSS="./daps.css" ❺

## Formatter
# Specify which PDF formatter to use. Currently only fop or xep are supported

FORMATTER="xep" ❻

##Draft Mode
# Turns on DRAFT watermarks in PDF or HTML builds when set to "yes"
# Is ignored for any other output format and has no effect on profiling.
# This value can be set to "yes" using the -d switch on the command line
# Also see COMMENTS and REMARKS
#
```

```
DRAFT="yes" ❼
```

❶ Specifies the Main XML file. It contains the highest-level object (root element) of your documentation project. The Main file must be located in *`YOUR_DOC_DIR/`*`xml/`. Therefore, you only need to specify the Main's file name in the DC file (no path).

❷ Defines the root ID of the element to be used for creating an output format. Usually, you specify the root ID of a `book` or `article` element here.

In this example, `book.daps.user` is the root ID of the DAPS User Guide, contained in `MAIN.DAPS.xml`.

❸ For a custom layout, use the *`STYLEROOT`* parameter to specify the (absolute or relative) path to the directory containing the custom stylesheets. Using absolute paths is recommended for DC files.

In this example, the parameter specifies the path to a directory with SUSE-specific stylesheets that is used by the DAPS User Guide.

❹ Allows you to define a fallback which is used in case the custom stylesheets defined with *`STYLEROOT`* cannot be accessed. In case neither the stylesheets specified with *`STYLEROOT`* nor with *`FALLBACK_STYLEROOT`* can be accessed, DAPS uses the default DocBook layout.

In this example, no fallback is specified and the parameter is disabled.

❺ If not specified, DAPS will use the default DocBook stylesheets for production of HTML and ePUB. For custom CSS styles, specify the (absolute or relative) path to the respective CSS file. Using absolute paths is recommended for DC files.

In this example, custom CSS files are specified for both HTML and ePUB output.

❻ Specifies the PDF formatter to use.

For supported formatters, refer to the section called “Software Requirements”. In this example, XEP is specified as PDF formatter.

❼ When set to `yes`, a DRAFT watermark appears in PDF or HTML outputs of the document.

If your documentation project contains cross-references between the individual books in a set, it is useful to define an additional DC file —*without* the *`ROOTID`* parameter. Use this (generic) DC to generate HTML outputs containing all hyperlinks between the individual books (or for creating file lists of all source files and images used in the set). Find an example DC file in Example 2.8, “DC File for a Set”.

Example 2.8. DC File for a Set

```
## Doc config file for the DAPS Documentation Set
## See /etc/daps/config for documentation of the settings below

## Mandatory Parameter
MAIN="MAIN.DAPS.xml" ❶

## Optional Parameters
## ROOTID
## If MAIN contains a set with several books and/or articles, use
## a separate DC-file for each book/article and set ROOTID to
## the id of the respective <book>/<article> element of the document
## This will enable you to build individual books/articles rather than
## the whole set
## See http://www.docbook.org/tdg/en/html/set.html for more information
## on sets
#ROOTID="" ❷
```

```
## Custom Stylesheets
## (if not defined the DocBook stylesheets will be used)

STYLEROOT="/usr/share/xml/docbook/stylesheet/suse/xslt/" ❸
#FALLBACK_STYLEROOT="" ❹
HTML_CSS="./daps.css" ❺
EPUB_CSS="./daps.css" ❺

## enable sourcing
export DOCCONF=$BASH_SOURCE ❻
```

❶ Specifies the Main XML file. It contains the highest-level object (root element) of your documentation project. The Main file must be located in *YOUR_DOC_DIR/*`xml/`. Therefore, you only need to specify the Main's file name in the DC file (no path).

❷ Defines the root ID of the element to be used for creating an output format. Usually, you specify the root ID of a `book` or `article` element here.

In this example, no *ROOTID* is set. This allows to build the complete documentation set, with the output containing all hyperlinks between the individual books.

❸ For a custom layout, use the *STYLEROOT* parameter to specify the (absolute or relative) path to the directory containing the custom stylesheets. Using absolute paths is recommended for DC files.

In this example, the parameter specifies the path to a directory with SUSE-specific stylesheets.

❹ Allows you to define a fallback which is used in case the custom stylesheets defined with *STYLEROOT* cannot be accessed. In case neither the stylesheets specified with *STYLEROOT* nor with *FALLBACK_STYLEROOT* can be accessed, DAPS uses the default DocBook layout.

In this example, no fallback is specified and the parameter is disabled.

❺ If not specified, DAPS will use the default DocBook stylesheets for production of HTML and ePUB. For custom CSS styles, specify the (absolute or relative) path to the respective CSS file. Using absolute paths is recommended for DC files.

In this example, custom CSS files are specified for both HTML and ePUB output.

❻ When set to `yes`, a DRAFT watermark appears in PDF or HTML outputs of the document.

❻ Enabling this parameter allows you to source the DC file on the Bash with DAPS. Sourcing a DC file (formerly called `ENV` file) was necessary to work with the documentation environment provided by susedoc, DAPS's predecessor.

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 2.9, “Required 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 2.9. Required 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 On the topmost level of your documentation directory, store the DC file defining your documentation project. You can store multiple DC files here (for multiple books belonging to the same documentation project, or DC files for various documentation projects). For more information, refer to the section called “Key Files”.

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 basic information about referencing images, refer to the section called “Referencing Images”.

❹ Directory holding the Main XML 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.

❺ The Main file of the documentation project. It contains the “starting point” (the highest-level object) of your documentation project. For more information, refer to the section called “Key Files”.

The `build` Directory

To strictly discriminate between all source content added by users and the content generated by DAPS, DAPS uses a `build` directory. When generating output from your documentation project for the first time, DAPS adds a `build` directory to your documentation directory. It is located parallel to the `xml` and `images` subdirectories. (If desired, the name and path of the `build` directory can be changed with the parameter *`BUILD_DIR`* in `/etc/daps/config` or `~/.daps/config`.)

The `build` directory is structured as follows:

Example 2.10. Build Directory

```
YOUR_DOC_DIR ❶
  |--build/ ❷
     |--NAME_OF_DC1/ ❸
     |--NAME_OF_DC2/ ❸
     |--.images/ ❹
     |--.profiled/ ❺
     |--.tmp/ ❻
```

❶ “Working directory” for the respective documentation project.

❷ Directory that holds all contents build by DAPS.

❸ For each of your documentation deliverables, DAPS creates a subdirectory, named after the respective DC from which you build the book, article or set. All formats that have been generated from the DC (PDF, HTML, TXT, ePUB etc.) can be found there. A `log` subdirectory stores log files for each output format that has been generated by DAPS.

❹ Directory holding the images created by DAPS.

❺ Directory holding the profiled XML sources created by DAPS.

❻ Directory holding temporary files created by DAPS (for example, the FO files).

Chapter 3. Editing DocBook-XML

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.

Basic Structural Elements

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.

Macros for Automatic Insertion of Complex Elements

XML elements can be nested to a high extend. Some constructs like `variablelist`, `table` or `image` have a lot of required child elements. If you have an editor with DTD support, it will tell you which elements are allowed at the current cursor position, but nevertheless it is cumbersome if you need to insert the child elements of complex XML constructs consecutively.

Most editors allow you to define or record macros which you can use for automatically inserting empty “skeletons” for a complex XML construct as illustrated by Example 3.1, “A `varlistentry` ”.

Example 3.1. A `varlistentry`

```
<varlistentry>
  <term></term>
  <listitem>
    <para></para>
  </listitem>
</varlistentry>
```

For Emacs, DAPS already includes macros for adding elements such as `listitem`, `figure`, `indexterm` etc. The macros are defined in `docbook_macros.el` and are added to your system during the installation of DAPS. They require that you execute Emacs in `psgml`-mode.

Procedure 3.1. Configuring Emacs to Use the DB Macros

1. To load the DocBook macros, open your Emacs customization file (`~/.emacs` or `~/.gnu-emacs`).

2. Insert the following line:

   ```
   (load "/usr/share/emacs/site-lisp/docbook_macros.el" t t)
   ```

3. Save the Emacs customization file.

For an overview, which macros are available and how to use them, refer to http://en.opensuse.org/openSUSE:Documentation_Emacs_Docbook_Macros.

Spell Checking

DAPS comes with a spell checker that is optimized for DocBook documents: Tags and attributes are excluded from the check so that you can focus on the content of the document. The spell checker is based on aspell and can be run from the command line. By default, it starts in interactive mode, but you can also run it in “batch” mode where it dumps a sorted list of misspelled words to standard output. DAPS also allows you to specify a custom dictionary and the language to use for spelling.

In the following, find some examples for using daps-susespell:

Spell Checking One or Multiple XML Files

```
daps-susespell xml/file1.xml xml/file2.xml
```

Checks the specified files with the default dictionary (en_US). Suggests alternative spellings for each misspelled word and waits for user interaction.

Spell Checking XML Files in Languages Other than English

```
daps-susespell --lang=de_DE xml/filename.xml
```

Checks the specified files with the de_DE dictionary (make sure the specified aspell dictionary is installed). Suggests alternative spellings for each misspelled word and waits for user interaction. For information about the language code to use for the `--lang` option, run daps-susespell --help.

Spell Checking XML Files in Batch Mode

```
daps-susespell --list xml/filename.xml > aspell_output.tx
```

Checks the specified files with the default dictionary (en_US). Returns a list of misspelled words to standard output and writes them to the file `aspell_output.txt`. You can use the `--list` option to easily collect a list of words that are unknown to aspell and use this output as basis for a custom aspell wordlist or dictionary.

Spell Checking XML Files with an Additional Custom Dictionary

```
daps-susespell --dict /absolute_path_to_custom_dict xml/
```

Checks the specified files with the default dictionary (en_US) plus the additional custom dictionary specified with `--dict`.

Validating Your XML Sources

Validating XML files within in a book or set often exceeds validation of the current XML file, as links (`xref` elements) or XIncludes need to be resolved, too. If you use conditional text (`profiling`) in your XML sources (for creating variants), your XML editor cannot check validity of your XML files. However, DAPS can handle all those cases due to the built-in xmllint validator.

Procedure 3.2. Validating XML Files

To validate all files that belong to your documentation project, DAPS only needs to know which Doc Config file to use. If you have specified a value for *`DOCCONF_DEFAULT`* in `~/.daps/config` or if your documentation directory contains only one DC file, DAPS automatically uses the corresponding DC file. Otherwise, specify the path to the DC file with the `-d` option as described below.

By default, `remark` elements and XML comments are ignored during validation. However, if you intend to create a (draft) output including remarks or comments, you need to include them for validation by specifying the respective DAPS option.

1. To validate all XML files in your book, article or set, enter:

   ```
   daps -d PATH_TO_DC_FILE validate
   ```

   If the XML files are not valid, DAPS will return the parser errors. They include information about the type of error, the respective file name and the line number where the error occured. In addition, DAPS shows the path to the profiled XML sources and the total number of errors.

   If validation was successful, DAPS returns: `All files are valid.`

2. To validate your files including remarks, enter:

   ```
   daps -d PATH_TO_DC_FILE validate -r
   ```

3. To validate your files including XML comments, enter:

   ```
   daps -d PATH_TO_DC_FILE validate -c
   ```

Example 3.2. Parser Output For Validation Errors

```
daps_user_concept.xml:60: element xref: validity error : IDREF attribute linkend r
Document /local/svn/daps-svn/daps/doc/build/.profiled/x86-amd64-em64t_osuse_/MAIN.
make: *** [validate] Error 3
```

Chapter 4. Generating Output Formats

DAPS supports a number of different output formats, including also “exotic” formats like man pages or simple text. Generating any output requires that your XML files are well-formed and can also be validated. You can build several output formats in parallel, build your complete documentation project (set, book, or article) or only a part of it (for example, a specific chapter). If you want the output format to contain meta-data about the XML files (for example, file name or workflow status), to include `remark` elements or to be marked as a `draft` version, you can tell DAPS to do so by specifying options on the command line. By default, DAPS uses the regular DocBook stylesheets, but DAPS also allows you to customize your output formats in a very flexible way.

Supported Output Formats

DAPS currently lets you publish your XML sources in the following output formats:

- PDF
- HTML
- HTML-single
- ePUB
- text
- man page
- Web Help

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 the available output formats, run **daps --help** and have a look at the subsection in *Subcommands* that is entitled *Generate Books*.

Basic Syntax

Work in progress

Chapter 5. 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. For basic information about image handling (such as supported formats, where to store the images in your documentation directory, and how to reference them in your XML files), refer to the section called “Image Handling”.

Supported Image Types

DAPS supports the following types of images:

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

Chapter 6. Review and Translation Processes

Work in progress

Chapter 7. Packaging and Deployment

For deploying the documentation as RPM packages and integrating it into KDE and GNOME desktop environments as well as into Web user interfaces (via JSP), DAPS offers a number of options to produce the corresponding output: For example, you can create source packages, HTML tarballs, color PDFs and desktop and document files with the **daps package-*** commands.

Work in progress

Chapter 8. Customizing Layout of the Output Formats

Work in progress

Chapter 9. Customizing/Configuring DAPS

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.

Chapter 10. Troubleshooting

DAPS is less verbose than its predecessor `susedoc`. If you should run into problems with DAPS, check the DAPS log files in *`YOUR_DOC_DIR`*`/build/`*`BOOKNAME`*`/log`. A complete log file of the latest **daps *`subcommand`*** that was executed is available in *`YOUR_DOC_DIR`*`/build/`*`BOOKNAME`*`/log/make_`*`SUBCOMMAND`*`.log`

In case of an error the complete log will be shown on the screen (STDOUT).

To get the same level of output as with susedoc 4.x, run **daps** with the `-v` option. For more details, use the `--debug` option.

Appendix A. 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 the section called “Migration of Existing DocBook Projects”.

Glossary

Conditional Text	See Profiling.
Document Type Definition	
DOCTYPE Declaration	
DocBook Authoring and Publishing Suite	
DocBook	
Entity	
Formatter	
FO	See Also Formatter.
Portable Document Format	
Processing Instruction	
Profiling	
Scalable Vector Graphics	
Stylesheet	
Validation	
XInclude	
XML Catalog	
XML Parser	
Extensible Markup Language	
Extensible Stylesheet Language	See Also Extensible Stylesheet Language for Transformations, FO.
XSL-FO	See Also Extensible Stylesheet Language for Transformations, FO.
Extensible Stylesheet Language for Transformations	

Appendix B. GNU Licenses

This appendix contains the GNU General Public License version 2 and the GNU Free Documentation License version 1.2.

GNU General Public License

Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc. 59 Temple Place - Suite 330, Boston, MA 02111-1307, USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation’s software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author’s protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors’ reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone’s free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

GNU GENERAL PUBLIC LICENSE TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The “Program”, below, refers to any such program or work, and a “work based on the Program” means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term “modification”.) Each licensee is addressed as “you”.

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program’s source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a). You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b). You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c). If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a). Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b). Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c). Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program 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.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public 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.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and “any later version”, you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM “AS IS” WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR

LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the “copyright” line and a pointer to where the full notice is found.

```
one line to give the program’s name and an idea of what it does.
Copyright (C) yyyy name of author



This program is free software; you can redistribute it and/or
modify it under the terms of the GNU General Public License
as published by the Free Software Foundation; either version 2
of the License, or (at your option) any later version.



This program is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
GNU General Public License for more details.



You should have received a copy of the GNU General Public License
along with this program; if not, write to the Free Software
Foundation, Inc., 59 Temple Place - Suite 330, Boston, MA 02111-1307, USA.
```

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

```
Gnomovision version 69, Copyright (C) year name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details
type `show w’. This is free software, and you are welcome
to redistribute it under certain conditions; type `show c’
for details.
```

The hypothetical commands `show w’ and `show c’ should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w’ and `show c’; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a “copyright disclaimer” for the program, if necessary. Here is a sample; alter the names:

```
Yoyodyne, Inc., hereby disclaims all copyright
interest in the program `Gnomovision’
(which makes passes at compilers) written
by James Hacker.



signature of Ty Coon, 1 April 1989
Ty Coon, President of Vice
```

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Lesser General Public License [http://www.fsf.org/licenses/lgpl.html] instead of this License.

GNU Free Documentation License

Version 1.2, November 2002

Copyright (C) 2000,2001,2002 Free Software Foundation, Inc. 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

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.