

# How to Author Core Documentation (2.1 legacy document)

## Table of contents

1 Comments.....	5
-----------------	---

## Table of contents

1 Overview.....	3
2 Purpose.....	3
3 Intended Audience.....	3
4 Prerequisites.....	3
5 Steps.....	3
5.1 1. Join the cocoon-docs email list.....	3
5.2 2. Find a job.....	3
5.3 3. Announce your effort.....	3
5.4 4. Do the work.....	4
5.5 5. Ask for help when you need it.....	4
5.6 6. Get some feedback.....	4
5.7 7. Review your work.....	4
5.8 8. Validate your document(s).....	4
5.9 9. Update any related pages.....	4
5.10 10. Prepare any related patches.....	4
5.11 11. Submit via Bugzilla.....	5
6 Summary.....	5
7 Comments.....	5

**Warning:**

This document was copied as is from the Cocoon 2.1 documentation, but has not yet been fully reviewed or moved to its new home.

## 1. Overview

This How-To describes the steps necessary to author or revise core documentation for Cocoon. Documentation is considered "core" if it is included in any Cocoon guide. Currently there are three guides: CTWIG, user, and developer. Please note that this guide structure may be revised soon. As you probably have discovered by now, many gaps exist in current Cocoon documentation. Authoring new core documents is a valuable way contribute to the Cocoon community.

## 2. Purpose

Writing about Cocoon is not a trivial exercise. Understanding the Cocoon CVS and document organization takes time. In other words, many potential obstacles stand in your way. These guidelines were written to help you make the most productive use of your "volunteer" time. Following them not only saves your time but also saves the time of committers who will apply your work to the CVS.

## 3. Intended Audience

Cocoon users who would like to improve Cocoon's documentation. This includes authors, editors, proofreaders, and quality assurance testers.

## 4. Prerequisites

- Spend some time familiarizing yourself with status of existing Cocoon documents.
- When evaluating doc-related issues, make sure you are referring to the most recent document versions from the CVS or Cocoon web site. Please note that sometimes the most recent version of any document will be found only in CVS HEAD.

## 5. Steps

Here's how to proceed.

### 5.1. 1. Join the cocoon-docs email list

Find out what documentation efforts are already in process among other users and committers. Join the Cocoon [Docs List](#).

### 5.2. 2. Find a job

Look through Cocoon's documentation set in order to find content holes to fill or existing documents to improve. You don't have to be an author to contribute: editors, proofreaders, and quality assurance testers provide vital work as well.

### 5.3. 3. Announce your effort

Let others know about your efforts. Announce your plans on the cocoon-docs list. This will help to prevent any duplication of effort. It will also attract the interest of and valuable feedback from other users who have similar interests or expertise in your document's subject area.

#### 5.4. 4. Do the work

Perhaps the easiest way to start a new document is to use an existing document as a template. If you are revising an existing document, make sure you are using the most recent copy of the document from CVS HEAD. If you aren't working with a local CVS repository, you can access all CVS files through [ViewCVS](#).

#### 5.5. 5. Ask for help when you need it

Writing effectively about a "glue" framework like Cocoon, which integrates many diverse technologies, is difficult, no doubt about it. Navigating your way through the CVS, Bugzilla, the patch process, as well as all of the steps of document creation can be overwhelming at first. Many of us have "been there" already and are available to help you. Don't hesitate to ask for assistance on cocoon-docs when you are confronted with a conceptual or technical problem you can't solve on your own. Don't waste your precious "volunteer" time pulling your hair out. Still, if you reach a point in your work where you are hopelessly stuck, you can always leave concise comments in a <fixme> element for others to fill down the road. This is in tune with open source, community-based development. Contribute what you can, when you have an irresistible "itch" to "scratch". Others will pick up where you left off.

#### 5.6. 6. Get some feedback

If you are working on a new or revised document, please post a text version of it to the cocoon-docs list to receive valuable comments from developers and users. Again, if you have technical holes in your content, be sure to add fixme comments to make it clear for others what particular areas you'd like for them to address.

#### 5.7. 7. Review your work

Look over your work for embarrassing spelling or grammatical errors. At least check your document with a spell checker before submitting it.

#### 5.8. 8. Validate your document(s)

Use the appropriate and most recent dtd to validate any new or revised documents. You will find all dtds in the src/documentation/xdocs/dtd directory. While a "docs" target (during Cocoon's build process) is capable of validating your files, it is far more efficient to troubleshoot validation and well-formedness problems with your own tools.

#### 5.9. 9. Update any related pages

If your work impacts the content of other files, for example the menu file (known as book.xml), consider updating these documents as well. You can validate and check link targets within all your documents by performing a docs build. If you have a working copy of the cvs HEAD, run the appropriate build script inside the cocoon-2.1 directory, specifying docs as the build target. Here's an example:

```
./build.[sh|bat] docs
```

#### 5.10. 10. Prepare any related patches

Any new document file is already a patch, at least as far as Bugzilla is concerned. However, if you also

edited any existing documents, you will need to create a patch for them before submitting all files. If you don't know how to create a patch, follow the instructions in [How to Prepare a Patch](#).

## 5.11. 11. Submit via Bugzilla

Create an attachment for any documents, and submit it via Bugzilla. If you don't know how to submit via Bugzilla, follow the instructions in [How to Contribute a Patch via Bugzilla](#).

## 6. Summary

The Cocoon documentation effort is based on the belief that open source documents can be as first-rate as the software on which they are based. However, such an endeavor is dependent upon vital forms of contributions from developers and users alike. This How-To gives you the information you need to join like-minded individuals in the effort to improve Cocoon docs. Take a minute to imagine how advanced the Cocoon community would become if more Cocoon users played a role in improving Cocoon docs. Think how much more we would learn. Think how many more innovative Cocoon-based solutions we could develop, based on our extended knowledge. Think about it.

## 7. Comments

Care to comment on this How-To? Got another tip? Help keep this How-To relevant by passing along any useful feedback to the author, [Diana Shannon](#). Even better, join the Cocoon [Docs List](#) and share your feedback and ideas with other people who are committed to producing high quality Cocoon documentation!

### 1. Comments

add your comments