

# How to prepare a unified patch

## Overview

A "patch" is the set of differences between two versions of the same file. Patches are used to send someone the exact changes that you have made to your version of a program or a document. They can then apply that patch to their version to merge the changes and bring their version up-to-date with your version.

As our example we use the contribution of a simple documentation patch for the Castor project. The principles apply to any project and to any type of file, e.g. \*.xml, \*.java, \*.xsd, etc.

## Intended Audience

Anyone who wants to contribute to a project. This document addresses the basics, so as to get new people started.

Our example describes the use of command-line tools for a UNIX system. Other tools can be used, as long as they produce a "unified diff".

## Prerequisites

Contributors should have:

- The source code of the documents as a local working copy of the CVS repository. If you are working with the current CVS HEAD then you will have already done a 'cvs checkout castor' (see CVS Usage Precis). However, see below for other ways of obtaining source for diff comparison.
- The tools with which to prepare a patch. On UNIX the "cvs" program has the `cvs diff -u` command.

## Steps

Here is how to proceed.

## Understand what a patch is

A "Patch" is the set of differences between two versions of the same file. A patch comprises one or more "diff" files. These diffs are produced by the program of the same name: diff.

Here is an example of a single diff for one of the Castor How-to pages, where we are suggesting a minor text change. Do not get frightened. These are just human-readable instructions to the "patch" program.

```
Index: contrib.xml
=====
RCS file: /home/projects/castor/src/doc/jdo-howto.xml,v
retrieving revision 1.7
diff -u -r1.7 contrib.xml
--- jdo-howto.xml 30 Apr 2002 07:44:52 \-0000      1.7
\+\+\+ jdo-howto.xml 26 May 2002 04:08:23 \-0000
@@ \-208,7 \+208,7 @@
  to create a patch. (The commands are for Linux.)
  </p>
- <s2 title="How to Establish your Local Repository">
+ <s2 title="How to Establish your Local Working Copy">

  <p>
  This will checkout the current copy of the master cvs repository and
```

That is a "unified diff" ... there are some lines of context on each side of the changes. This patch is basically saying "Change the text on line 208".

- lines to be deleted are preceded with -
- lines to be added are preceded with +
- contextual lines with no leader remain the same

## Modify your document and ensure consistency

Let us now go through the process of preparing that patch. Go ahead and edit your local copy of the document at `$CASTOR_HOME/src/doc/jdo-howto.xml`.

Ensure that it is valid XML using your favourite XML editor or an external validating parser. Please do not leave it up to the poor committer to fix broken XML.

Run the `'build doc'` target to be sure that links are not broken and that the new document is presented as you intend it to be.

## Get ready

If you are using the HEAD of CVS then ensure that your working copy is up-to-date. Of course, if you are using a previous public release version of Cocoon, then it is already up-to-date.

## Generate the differences

Prepare the diff file. CVS will contact the remote repository, ensure that your working copy is up-to-date, then compare your local copy with the master repository.

```
cd src/doc
cvs diff \-u jdo-howto.xml > $TEMP/castor/patch/jdo-howto.xml.diff
```

## Describe the patch

Prepare a brief explanation of what your patch does. Get this ready in a text file before you go to Jira. See further hints about this in the "Description" section of the How-to Jira.

- What version of CVS did you patch against?
- Was it HEAD branch?
- Was it a nightly build?
- Was it a public release?

## Submit via Jira

To contribute your patch to a specific project, use Jira - The Codehaus Issue Database. The procedure is explained in How to Contribute a Patch via Jira.

## Real World Extension

### Multiple diffs in a single patch

A patchfile can contain the differences to various individual documents. For example, the following command does that ...

```
cd src
cvs diff \-u > $WORK/castor/patch/src.dir.diff
```

However, be careful not to go overboard with this technique. When producing multiple diffs in one patchfile, try to limit it to one particular topic, i.e. when fixing the same broken external link in various pages, then it would be fine to produce a single diff. Consider the committer - they will find it hard to apply your patch if it also attempts to fix other things.

## Other ways of obtaining source for diff comparison

Ideally you will prepare your patches against a CVS repository. There are other ways to do this. They do create more work for the committers, however it may be the only way that you can do it. We would certainly rather receive your patch however it comes. As a matter of fact, we would politely ask you first to send us a unified patch.

You could get the source document via the web interface to CVS. Here are the steps ...

- get the relevant XML file via ViewCVS
- save the file to your local disk: `./jdo-howto.xml.orig`
- create a copy of the file: `./jdo-howto.xml`
- make your modifications and validate the XML
- use the "diff" command (i.e. not 'cvs diff') as follows
- `diff -u jdo-howto.xml.orig jdo-howto.xml > $WORK/castor/patch/jdo-howto.xml.diff`
- proceed as for Step 5.

## Tips

- Please review your diffs before you submit your patch to Bugzilla

## References

- The UNIX manual pages 'man diff' and 'man patch'.
- CVS usage information is at [www.cvshome.org](http://www.cvshome.org) and your local 'info cvs' pages or 'man cvs' pages or user documentation. This includes guidance about using cvs diff