

Writing How- Tos

Table of contents

1 Writing a How-To Document.....	2
1.1 Introduction.....	2
1.2 Content.....	2
1.3 Procedure.....	2
1.4 Contributing.....	3

1. Writing a How-To Document

1.1. Introduction

One of the best ways to communicate is through a targeted tutorial. The How-Tos section of this site is intended to be a group of documents describing how specific aspects of meta-jb can be utilized or extended.

1.2. Content

Typically, a "How-To" document will start with an introduction section describing the basic use-case of the how-to, a brief overview of the components involved, and perhaps information on a target audience.

Subsequent sections will indicate different major steps along the way. To whatever extent possible, these sections should be self-contained with each one building on the previous.

1.3. Procedure

Described here are the specific steps to adding a How-To document. I prefer the jspwiki mark-up for my own editing because I find it easier to type than xdoc XML. My instructions will be structured accordingly but one could just as easily create xdoc.

Steps:

1. Plan the basic content you want to add. If you can't clearly articulate the use-case or if you think you might have trouble implementing it yourself then some more research may be in-order. I'd prefer it if works in progress don't get linked into the main site but they can still be added to the module for source tracking if desired.
2. Create your document with the appropriate name. JSP-Wiki pages will split words on camel case, so: FooBar will become "Foo Bar" at the top of your page. Give the file the .jspwiki extension and it will automatically be converted to HTML during site building.
3. When finished, add a link to your page in the appropriate how-to section of the site.xml file. Tags can be named anything and are really just used for convenient internal referencing. We could call them all <how-to> I think and it would still work. When referencing your document use the .html extension since that's what the output will be.
4. Run "forrest site" and look at the resulting site to verify you've hooked everything up correctly.
5. Add your file to CVS and commit.

6. Pat yourself on the back for contributing to documentation.

1.4. Contributing

If you are not a committer to the project, you can still write documentation. In fact, I think that would be very nifty. Just develop your content in a local version of the repository if you want to check that it works with forrest. Alternately, you can just write up whatever you can in whatever format you want and one of us will commit it. Things just go quicker if it is already a drop-in addition.