AsiaBSDcon 2016:
Getting Started with the FreeBSD Documentation and Translation Projects

Dru Lavigne (dru@freebsd.org), Benedict Reuschling (bcr@freebsd.org)

February 8, 2016

Abstract

The FreeBSD Documentation Project (FDP) maintains the FreeBSD documentation set, including man pages, the Handbook and FAQ, articles, and web pages.

In this hands-on workshop, participants will learn the tools and methods of contributing to the FreeBSD Documentation Project as well as how to create translations for existing FreeBSD documentation. The goal for this workshop is to provide each participant with the knowledge and some hands-on experience for submitting patches and translations to the FreeBSD Documentation Project. By the end of the workshop, participants will understand the documentation workflow as a whole and will have everything they need to continue working after the workshop is over.

At the end of this workshop, participants will understand the following:

- how to install and use the documentation tools,

- the FreeBSD documentation layout and workflow,

- how to edit existing and contribute new documentation,

- how to submit a patch,

- how to use the PO/gettext translation system,

- and how to work with other members of the doc team.

By the end of this session, participants will have created at least one patch and will know where to go for further assistance.

Since this is a hands-on workshop, participants should bring a laptop and are encouraged to have FreeBSD already installed on their system or in a virtual environment. A graphical environment (X windows system with a window manager of your choice) is recommended. You will need to install additional software, so make sure networking is enabled and you can connect to the Internet.

Contents

1 Conventions & Trademarks Used in This Document

Whenever we show a % sign as the prompt, it means that this is a normal user executing the command that follows it. A # prompt indicates that the command needs to be run with superuser (i.e. `root`) privileges.

We will refer to the FreeBSD Documentation Project as the FDP.

FreeBSD is a registered trademark of The FreeBSD Foundation. FreeBSD's documentation is provided under the FreeBSD Documentation License[1].

2 Introduction to the FDP

The FreeBSD Documentation Project (FDP) maintains the documentation for the FreeBSD project. Its purpose is to provide FreeBSD users with instructions for configuring and using the features provided by the FreeBSD operating system. It is intended for new users who have had little to no experience with a Unix-like system as well as users who want to learn more about the FreeBSD operating system.

FreeBSD's documentation set includes a number of documents of different types. Some of the most important ones are described here:

The FreeBSD Handbook: a resource for users to learn about the features this operating system provides. Its main focus is towards users and administrators of FreeBSD systems.

FAQ: A collection of frequently asked questions and their answers. This resource provides users with a written source of consistently maintained answers.

Manual pages: These are the manual pages that are provided with the operating system using the `man(1)` format. In FreeBSD, manual pages are managed outside the `doc` repository as they reside within the `src` repository. Manual pages are collaboratively maintained by both `src` and `doc` committers.

Articles: The FreeBSD Project maintains a collection of smaller articles that describe certain things that do not fit into other documentation categories but are worth keeping around, sometimes for historical reasons.

Books: This category includes the Porter's Handbook, with instructions on how to work with the ports system as a developer or `ports` committer, as well as the FreeBSD Developers Handbook, intended for people

---

[1] `http://www.freebsd.org/copyright/freebsd-doc-license.html`

who want to know more about how FreeBSD is developed and how some of its subsystems work.

The FreeBSD Website: The FreeBSD Website is maintained by the Documentation Project and can be built like any other document using a `Makefile`.

The FDP also maintains the FDP Primer[2], which contains instructions for people that want to get involved in the FreeBSD documentation project. Starting from the tools that are being used, to formatting and indentation rules, it covers everything a person involved in FreeBSD's documentation needs to know. It is used as a reference even by documentation committers who have been around for a long time.

Translation projects for various countries and languages exist to translate FreeBSD's documentation into their respective languages. Translated documentation helps users who do not speak English and who find it easier to understand the system when the instructions are written in their native language.

---

[2]`https://www.freebsd.org/doc/en_US.ISO8859-1/books/fdp-primer/`

3   Get the Tools

The FDP maintains the textproc/docproj meta-port which provides the software needed to edit and build the FreeBSD documentation. To save compilation time, we will install this software using the package. Since the default package does not include Japanese (CJK) support, the translation tools, or a translation file editor, we will also install those packages.

NOTE: Although manual editing of a `.po` (translation) file is possible, using a PO editor provides more comfort and error checking. We will use **PO edit**[3] in this workshop.

To begin the package installations, become the superuser and type:

```
# pkg install textproc/docproj
```

Whenever prompted, type **y** to proceed. Depending upon which dependencies are already installed on your system, this package installation may take a few minutes so wait until it is finished. If you receive an error with the installation, raise your hand so that an instructor can assist you in completing the installation.

Next, install the following packages. Again, type **y** when prompted. If you receive a message indicating that a package is already installed, skip that package and proceed to the next package installation.

```
# pkg install textproc/html2text
# pkg install japanese/font-ipa
# pkg install textproc/itstool
# pkg install editors/poedit
```

Once all of the package installations are complete, leave the superuser account and type **rehash** to load the new binaries into your path. You are now ready to move on to the next section to install the documentation source code.

---

[3]Available from `https://poedit.net/`

4   Get the Source

Most of the FreeBSD documentation, in all of the translated languages, lives in the doc repository. This includes the FreeBSD articles, the various handbooks, the FAQ, and the website, but not the man pages as those are included in the base repository as part of the command being documented.

In order to edit existing documentation, add new documentation, or translate documentation, you need to maintain a copy of that doc repository on your system and ensure that it is kept synchronized with the official FreeBSD repository.

Before downloading the doc repository, decide where on your computer to store the local copy of the repository. Since you will be using your regular user account when making doc changes, the location is typically somewhere in your home directory. The first time you download the repository, it will be saved to your current working directory unless you specify where to save it. We will create a subdirectory named *freebsddoc* in our home directory and specify that location when downloading the repository. Ensure that you are logged in as your regular user account and in your home directory before running these commands:

```
% svnlite co https://svn.freebsd.org/doc/head ~/freebsddoc
```

It will take a few minutes to complete the download. Note that you only need to use the **svnlite co** (short for **checkout**) command once, for the initial download. However, you should always verify that you are synchronized with the repository before you start to make any doc changes. To check for any changes and to synchronize with the repository, run the **up** command from within the directory that you downloaded the repository to. In our case, you would do this:

```
% cd ~/freebsddoc
% svnlite up
Updating '.':
At revision 48138.
```

In this example, no changes have been downloaded since none have occurred since you last downloaded the repository. If there had been changes, you would see something like this:

```
Updating '.':
U    zh_TW.UTF-8/books/porters-handbook/Makefile
U    zh_TW.UTF-8/books/porters-handbook/book.xml
At revision 48138.
```

In this example, there have been two updates (U) to the Chinese translation of the Porter's Handbook. It is important to note the names of the files that have changed if you are making edits to those same files and have not yet submitted your changes for review. You should also make note of the revision number as this may be needed during the translation process.

5 Create Accounts

Now that you have the necessary software and the documentation source, you should create a user account on the bug tracker and the review website. You will need to be logged into the bug tracker in order to submit a patch and you will need to be logged into the review website in order to request a review of a patch. It is also recommended to join the doc mailing list and, if you plan to assist with translations, the translations mailing list.

5.1 Bug Tracker

To create an account on the bug tracker, go to `https://bugs.freebsd.org/bugzilla/createaccount.cgi`. In the screen shown in Figure 1, input a valid email address and click the "Send" button. Use an email address that you check often as any comments and updates regarding your patch will be sent to that email address.

Figure 1: Create an Account on the FreeBSD Bug Tracker

Once you input your email address, the web page will display a message indicating that you need to check your inbox for an activation email and to click on the link within that email to activate the account within three days. When you click the link in the activation email, you will be prompted to set your password. If you do not see this email, check your SPAM filter. Raise your hand and let an instructor know if you do not receive the activation email.

Once you have activated your account, click the "Log In" link at the top of the page and enter your email address and password. If you do not want your browser to cache the password, uncheck the "Remember" checkbox. Click the "Log in" button. You should now see a page similar to the one shown in Figure 2.

Take a few minutes to click the "Preferences" link and to review the available preferences.

7

Figure 2: Main Page of FreeBSD Bug Tracker While Logged In
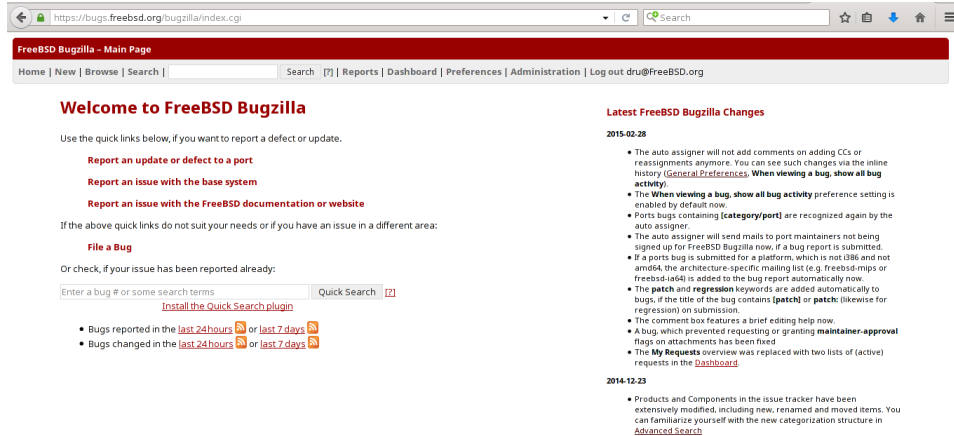
5.2 Review Website

Next, go to `https://reviews.freebsd.org/auth/register/`. In the screen shown in Figure 3, input a username into the "Real Name" field, enter and confirm a password, and input an email address. This will often be the same email address you use for the bug tracker, but does not have to be. Make sure to use a valid email address that you check often as all reviewer comments regarding your patch will be sent to that email address. Finally, type the text that appears in the CAPTCHA field and click the "Register Phabricator Account" button. The web page will display a message indicating that you need to check your inbox for an activation email and to use the instructions in the email to activate your account.

Figure 3: Create an Account on the FreeBSD Review Website

5.3 FDP Mailing List

To subscribe to the FDP mailing list, go to `http://lists.freebsd.org/mailman/listinfo/freebsd-doc`. In the screen shown in Figure 4, input a valid email address into the "Your email address" field. If you wish, you can also enter your name and pick and confirm a password, or leave those fields empty. By default, you will receive an email for every message that is sent to this list. This is a low-traffic list which averages a few messages a day. If you prefer to only receive one email a day that contains all of the messages from that day, click the "Yes" button next to "Would you like to receive list mail batched in a daily digest?". When finished, click the "Subscribe" button. Again, you should receive a confirmation email with a link that you must click in order to finish subscribing to this list.

Figure 4: Subscribe to the FDP Mailing List

If you wish, you can send an introductory email to this list to tell the other members of the list a little bit about yourself and your interest in FreeBSD documentation.

If you would like to review the archived messages for this mailing list, go to `http://lists.freebsd.org/pipermail/freebsd-doc/`. The archives are listed by month, with this month's messages at the top of the list.

5.4 Translations Mailing List

If you are interested in translating existing FreeBSD documentation into your native language, you should also subscribe to the translations mailing list. To do so, go to `http://lists.freebsd.org/mailman/listinfo/freebsd-translators`. This page is laid out the same as the FDP mailing list page so that you can subscribe to the list and view its archives. Again, do not forget to reply to the confirmation email. If you join this mailing list

you should send an introduction email indicating which language(s) you are interested in translating so that other translators of that language know that another person is available to assist with the translation.

5.5   IRC Channel

The FDP has its own `#bsddocs` channel on EFNET. The IRC channel provides an opportunity to have near real-time conversations with other members of the documentation team. This is useful if you want to discuss the best way to word a documentation improvement or have someone review a small patch before submitting it. If you hang out on this channel frequently, you will also get a good idea of who is working on which bits of documentation.

In order to join this IRC channel, you will need an IRC client. Once you have an IRC client, input an EFNET server from the list at `http://www.efnet.org/?module=servers` and join the `#bsddocs` channel. If you need assistance in selecting an IRC client or in joining the channel, raise your hand so an instructor can assist you.

6 Review Existing Bugs

You now have everything you need to start contributing to the FDP. Given the amount of existing documentation, it can be a bit overwhelming deciding where to start. In this workshop, we will work in small teams. Each team will be assigned an existing documentation bug to work on, make the necessary edits to resolve the bug, submit a patch, and have their patch reviewed and committed. This allows everyone to see the entire documentation workflow as well as experience the satisfaction of seeing a real change that benefits FreeBSD users applied to the "live" documentation.

To see which documentation bugs are available, go to `https://bugs.freebsd.org`. Make sure you are logged in (so that you can interact with an existing bug) and click "Search" (4th option from the top left). In the screen shown in Figure 5, select "Documentation" in the "Product" drop-down menu, then click the "Search" button.

Figure 5: Search for Documentation Bugs

The results of the search will be displayed, as seen in the example in Figure 6.

To sort the bug list by date (oldest first), click the "ID" link; if you click it again, it will reverse the sort (newest first). Also, note who is assigned to each bug. If a person's email address is listed, that person is handling the bug. Typically, this means that you don't work on that bug unless you have talked to that person and have agreed to help them. If it is assigned to "freebsd-doc@FreeBSD.org", noone has "taken" the bug yet to work on it. To sort the bug list by assignee, click the "Assignee" link.

To see the details of a bug, click its number. The details will include the initial report, any existing patches, and all comments about the bug.

Work with your team and an instructor to find a bug that your team can fix. This will require some discussion and may require reviewing an existing patch to see if it is correct or deciding upon some text that will address a documentation deficiency.

Figure 6: List of Documentation Bugs

7 Make and Test Edits

Once your team has decided upon a bug to fix, an instructor can assist you in finding which source file to edit in order to address the patch.

7.1 Find the File to Edit

As an example, consider a bug that requires a fix to the FreeBSD Handbook. If you do a listing of your local copy of the doc repository, you will see this:

```
% cd ~/freebsddoc
% ls
Makefile          de_DE.ISO8859-1   hu_HU.ISO8859-2   nl_NL.ISO8859-1   share
README            el_GR.ISO8859-7   it_IT.ISO8859-15  no_NO.ISO8859-1   sr_YU.ISO8859-2
bn_BD.UTF-8       en_US.ISO8859-1   ja_JP.eucJP       pl_PL.ISO8859-2   tr_TR.ISO8859-9
chapter.diff      es_ES.ISO8859-1   ko_KR.UTF-8       pt_BR.ISO8859-1   zh_CN.UTF-8
da_DK.ISO8859-1   fr_FR.ISO8859-1   mn_MN.UTF-8       ru_RU.KOI8-R      zh_TW.UTF-8
```

Note that there is a directory for each available language. Here is a listing for the English language subdirectory:

```
% ls en_US.ISO8859-1/
Makefile      articles      captions      htdocs      slides
Makefile.inc  books         flyer         share
```

Here, the English documentation has been divided by articles, the various handbooks, the website (htdocs), and XML elements which are used by all of the English documentation.

Here is a listing of the books subdirectory:

```
% ls en_US.ISO8859-1/books/
Makefile          bibliography      developers-handbook handbook
Makefile.inc      design-44bsd      faq                 pmake
arch-handbook     dev-model         fdp-primer          porters-handbook
```

And a listing for the FreeBSD Handbook's subdirectory:

```
% ls en_US.ISO8859-1/books/handbook/
Makefile             config                introduction         ports
advanced-networking  cutting-edge          jails                ppp-and-slip
audit                desktop               kernelconfig         preface
basics               disks                 l10n                 printing
bibliography         dtrace                linuxemu             security
book.xml             eresources            mac                  serialcomms
boot                 filesystems           mail                 txtfiles.ent
bsdinstall           firewalls             mirrors              virtualization
catalog-cwd.xml      geom                  multimedia           x11
chapters.ent         imagelib              network-servers      zfs
colophon.xml         index.xml             pgpkeys
```

Note that there is a subdirectory for each chapter of the Handbook and that the name usually matches the name of the chapter. If you do a listing of each subdirectory, you will see that it (usually) contains a file named *chapter.xml*. So, if you needed to edit the Security chapter of the FreeBSD Handbook, you would edit this file:

```
~/freebsddoc/en_US.ISO8859-1/books/handbook/security/chapter.xml
```

7.2   Configure Editor

The *.xml* files in the doc repository are ASCII text files that are marked up using DocBook XML. This means that you should use an ASCII text editor to edit these files. Before making any edits, double-check that the editor your team is using has been configured as follows:

- Word wrap set to 70 characters.

- Tab stops set to 2.

- Replace each group of 8 leading spaces with a single tab.

How to configure those settings varies by text editor. Configurations for vim, emacs, and nano are listed at `https://www.freebsd.org/doc/en_US.ISO8859-1/books/fdp-primer/editor-config.html`. Graphical text editors such as kate or kwrite can be configured using their "Settings" menu. Raise your hand if you require assistance from an instructor in configuring your editor.

7.3   Make Your Edits

Once the editor is configured, open the file to edit and work with your team to make your changes. Depending upon the edit, you may need assistance in determining which formatting tags are required. Again, an instructor can assist you with your edits.

13

NOTE: for most edits, you can use the formatting tags and indentation in the surrounding text to determine how to format your edits. The XML Primer at `https://www.freebsd.org/doc/en_US.ISO8859-1/books/fdp-primer/xml-primer.html` is available for reference. However, you will find that there are only a dozen or so tags that you will use regularly.

7.4 Run igor

The FDP provides a utility named **igor** to assist you in finding formatting, grammatical, and spelling errors. Once you have saved your edits, run **igor** and specify the name of the file you made edits to. This example checks the XML file of the Security chapter and **igor** provides some feedback about that file:

```
igor ~/freebsddoc/en_US.ISO8859-1/books/handbook/security/chapter.xml
chapter.xml:374:spelling:Alternatively, if [noone] else can see your terminal now, you can
chapter.xml:1589:bad phrase:        will work, but [the A] and PTR records must be correct
and
chapter.xml:2532:contraction:The authenticity of host 'example.com (10.0.0.1)' [can't] be
established.
chapter.xml:2573:repeated:       <screen>&prompt.root; <userinput>scp
<replaceable>user@example.com:/[COPYRIGHT COPYRIGHT]</replaceable></userinput>
chapter.xml:2866:use tabs instead of spaces:[       ]use the &man.service.8;
command:</para>
chapter.xml:2869:blank line with whitespace:[       ]
chapter.xml:2869:use tabs instead of spaces:[       ]
chapter.xml:3060:repeated:       <programlisting>drwx------  2 [robert
robert]  512 Dec 27
11:54 private
chapter.xml:3061:repeated:drwxrwx---+ 2 [robert   robert]  512 Dec 23 10:57 directory1
chapter.xml:3062:repeated:drwxrwx---+ 2 [robert   robert]  512 Dec 22 10:20 directory2
chapter.xml:3063:repeated:drwxrwx---+ 2 [robert   robert]  512 Dec 27 11:57 directory3
chapter.xml:3064:repeated:drwxr-xr-x  2 [robert   robert]  512 Nov 10 11:54
public_html</programlisting>
```

When you are checking a file that you made edits to, you should only concern yourself with the lines that you edited. For example, if you made edits to lines 2860-2870, you should review those lines in the **igor** output. In this example, **igor** indicates that tabs were not properly used in lines 2866 and 2869 and that in line 2869 there is extra whitespace in a blank line. Work with your team members to resolve any issues introduced by your edits and, if needed, ask an instructor for assistance. Rerun **igor** until you are satisfied that your edits are formatted correctly.

7.5 Build Test

Next, you want to ensure that your edits will not break the build of the FreeBSD documentation and that the formatting tags you used will render correctly. If you made a change to the FreeBSD Handbook, run this command from the root directory for the handbook:

```
% cd ~/freebsddoc/en_US.ISO8859-1/books/handbook
% make
```

This will produce a lot of build output but should not fail with an error. If the build is successful, it will create an HTML file for each chapter of the Handbook. For example, to test a change to the Security chapter, open this file in a web browser and verify that your edits look properly formatted:

```
file://usr/home/YOURNAME/freebsd/en_US.ISO8859-1/books/handbook/security.html
```

NOTE: if your build fails with an error, an instructor can assist you in troubleshooting the error so that you can complete a successful build.

8   Create and Submit a Patch

Once you have verified your build, you are ready to create a patch for submission. To create the patch, run **svnlite diff** from the root directory of your local doc repository and specify the name of the patch to create. When naming the patch, use the format *chaptername.diff.txt*. This example creates a patch of the edits to the Security chapter of the Handbook.

```
% cd ~/freebsddoc
% svnlite diff > security.diff.txt
% more security.diff.txt
Index: en_US.ISO8859-1/books/handbook/security/chapter.xml
===================================================================
--- en_US.ISO8859-1/books/handbook/security/chapter.xml (revision 48145)
+++ en_US.ISO8859-1/books/handbook/security/chapter.xml (working copy)
@@ -27,6 +27,8 @@
   <sect1 xml:id="security-synopsis">
     <title>Synopsis</title>

+    <para>This is where I made my changes.</para>
+
     <para>Security, whether physical or virtual, is a topic so broad
       that an entire industry has evolved around it.  Hundreds of
       standard practices have been authored about how to secure
```

Note that it is important to create the patch in the root directory of the local repository. This ensures that the "Index" line in the patch contains the full path to the file that was edited so that the patch reviewer has this information. In this simple example, a new paragraph followed by an empty line was added to the Synopsis section of this chapter.

8.1   Submit the Patch

Once you have a patch, open up the bug ticket you are working on. Click the "Add an Attachment" link which will open the screen shown in Figure 7. Make sure you complete the following:

- Click the "Browse" button to browse to the location of the patch.

- Check the box next to "If the attachment is a patch, check the box below".

- Enter a comment describing the patch in the "Comment" box.

- Keep the "Add me to CC list" box checked so that you will be notified of any comments about your patch.

Figure 7: Attach the Patch to the Bug Report

File: *Enter the path to the file on your computer (or paste text as attachment).*

[Browse...] No file selected.

(File size limit: 1000 KB)

* Description: *Describe the attachment briefly.*

Content Type: *If the attachment is a patch, check the box below.*

☐ patch

Otherwise, choose a method for determining the content type.

◉ auto-detect

○ select from list: [plain text (text/plain) ▾]

○ enter manually: [   ]

Flags: **Requestee:**

maintainer-approval   [▾]

Reassignment: *If you want to assign this bug to yourself, check the box below.*

☐ take bug and set the bug status to [New (current) ▾]

Comment: *(optional) Add a comment about this attachment to the bug.*

Comment	Preview

Quick Editing Help

☑ Add me to CC list

[Submit]

When finished, click the "Submit" button. Raise your hand so the instructor can review the patch and make any further editing suggestions. Once the review is complete, the instructor will commit your patch and close the bug report for you.

Congratulations! You have just make a change to the FreeBSD documentation!

9 Using the Review System

When starting out with code changes, it is good to let someone else have a look at your patch before submitting it. FreeBSD uses the **Phabricator open source review system**[4]. We will demonstrate how to use FreeBSD's Phabricator to create a code review. Code reviews allow contributors to be sure that someone looked over their change and approved it. This builds confidence and encourages further patch submissions.

9.1 Client Setup

FreeBSD's Phabricator review system supports a command line client as well as a web browser interface. You already used that web interface when you created your account on Phabricator in section 5.2 on page 8.

The client component of Phabricator is called Arcanist (PHP-based) and can be installed from packages with this command:

```
# pkg install php5-arcanist
```

The client needs some configuration settings to know how to find and access the FreeBSD Review server. This involves copying a cookie that is provided during the setup process. Enter the following and follow the on-screen instructions:

```
% arc install-certificate https://reviews.freebsd.org
```

Once the cookie-setup is complete, the only thing left is to configure Arcanist to use the URL for FreeBSD's review system by default:

```
% arc set-config default https://reviews.freebsd.org/
```

9.2 Create a Revision using the Command Line Client

To create a revision from the top-level directory of your Freebsd documentation tree checkout for the given path, execute this command:

```
% arc diff --create /path/within/doc/tree
```

A template will be opened in the default editor with the following content:

```
<<Replace this line with your Revision Title>>

Summary:

Test Plan:

Reviewers:
```

---

[4]Available from `http://phabricator.org/`

18

```
Subscribers:

# NEW DIFFERENTIAL REVISION
# Describe the changes in this new revision.
#
# arc could not identify any existing revision in your working copy.
# If you intended to update an existing revision, use:
#
#   $ arc diff --update <revision>
```

Use the following instructions when filling out the template to match the review you want to create:

Revision Title (mandatory): This should be a short title for the review, describing what this review is for.

Summary (optional): Provide a summary of the intended changes. This should be in commit message style, but is optional and can be refined later. This should give reviewers more information than the title alone.

Test Plan (optional): Describe how you have tested the patch or how reviewers can do that. For documentation, it usually suffices to say that when the patch is applied, the doc build should still work without errors. Make sure that is true!

Reviewers (optional): This should list a couple of people that should review the patch before it is committed. If these people have Phabricator accounts, they will be notified (usually via email) of that new review. You can also add reviewers later using the web interface if you don't yet know who should review your patch.

When the template has been filled out, save your changes and exit your editor. The Phabricator system will create a new review on the FreeBSD server and the client will print a link for you referencing it in the following form:

```
https://reviews.freebsd.org/DXXXX
```

The XXXX parts will be replaced with the actual number of your review, similar to a ticket ID. The D represents the documentation tree. You can enter that URL into a web browser to see how your review looks like in the web application or to make changes to it.

9.3 Create a Revision using the Web Interface

To create a review using a web browser instead of the command line client, follow these steps:

1. Use a web browser to go to `https://reviews.freebsd.org/differential/`. Make sure you are logged into your account to create reviews, so that people know who this patch came from.

2. In the upper right corner, there is a button called **Create Diff**.

   Click that button to get to the screen to create new diffs.

3. On the **Create New Diff** page, you can either chose to

   (a) paste the patch directly into the input field (Raw Diff), or

   (b) upload your patch from a file (Raw Diff from File)

   In the **Repository** field, click on the small lens icon on the right. A new window will open, from which you need to select the documentation (doc) repository.
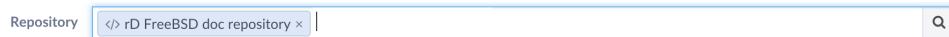
   Make sure that the **Visible To** field is set to **Public**, which is the default. When you are done, click on the **Create Diff** button.

4. A new page will appear with a preview of your patch at the bottom. At the top, make sure that the **AttachTo** pull-down menu has **Create a new Revision...** selected.

   Then, click on the **Continue** button.

5. The next page allows you to enter detailed information about your patch, a title, test plan, and who should review it. It has the same information as described in the template in section 9.2 on the previous page. When you are finished, create the review by clicking on the **Save** button.

10 Using the Translation System

This section takes a closer look at FreeBSD's new PO/gettext translation system. We will show how translators can automatically extract strings from a document that needs to be translated. The strings are then loaded into a PO editor, which supports editing these strings and offers suggestions of previously provided strings for re-use.

10.1 Initial Workspace Setup

For a new translation, the doc build system needs to have a certain directory structure, as well as a `Makefile`. The easiest way is to copy an existing directory structure and adapt it to our own translation project. In this case, we will copy the article about the handling of leap seconds in FreeBSD from the English (original) version. In the following example, replace `doc` with the local top level directory of your source checkout:

```
% svnlite mkdir --parents ~/freebsddoc/ja_JP.eucJP/articles/leap-seconds/
```

We now have the directory structure in place. Next, we copy the `Makefile` from the English version of that article to our newly created directory structure under `ja_JP.eucJP`:

```
% svnlite cp ~/freebsddoc/en_US.ISO8859-1/articles/leap-seconds/Makefile \
    ~/freebsddoc/ja_JP.eucJP/articles/leap-seconds/
```

10.2 Extracting strings

We now need to extract all the strings to be translated from the document. This can be done with the following command:

```
% cd ~/freebsddoc/ja_JP.eucJP/articles/leap-seconds/
% make po
```

We now have a new file in that directory called `ja_JP.po`. It contains all strings from the document that we need to translate.

10.3 Using a PO Editor

For this section, work in your team so that each person translates a portion of the document then reviews the portions of the document translated by other team members.

To start translating, open the file `ja_JP.po` in the `PO edit` editor by typing:

```
% poedit ja_JP.po
```

This will load the document and display the strings to be translated. Alternately, you can start `poedit` and use the "Open" dialog to browse to

that file and open it. On the left side of your editor, you will find all the strings from the english document. The right side lists the strings for which a translation already exists (empty at the beginning). At the bottom of the editor window, the currently selected string is shown. The field below is where you enter the translation in your local language that matches that string.

The following tips and rules should help you with the translation process:

Preserve XML tags: Do not translate strings that are XML tags like `<para>` or `</para>`. They are needed by the doc build system to markup the output in a certain way.

Preserve spaces: Do not change the spacing at the beginning and end of the strings in your translation. This is especially true for everything within `<programlisting>` and `<screen>` tags. All spaces must remain as they are in the translated version.

Preserve verbatim tags: Some tags do not need to be translated. Here is a list:

- `<citerefentry>`
- `<command>`
- `<filename>`
- `<literal>`
- `<manvolnum>`
- `<orgname>`
- `<package>`
- `<programlisting>`
- `<prompt>`
- `<refentrytitle>`
- `<screen>`
- `<userinput>`
- `<varname>`

Version Control strings: Things like `$FreeBSD$` do not need to be translated, as they are being replaced with a version number and other information upon checking the file into FreeBSD's version control system.

Names: Names of products, places, author names, and filenames like `Makefile` do not need to be translated. When in doubt, do not translate and discuss these within your translation group or ask a documentation committer.

Word list: Over time, you will find that you are translating the same string over and over again. In this case, it is good to create a word list that other translators can use. This is especially helpful for difficult words and will create better consistency over time as the same word can be reused in the same context, but in other sections or documents.

10.4   Generating the Translated Document

After translating a couple of strings or even the whole document, follow these instructions to see what the translated document will look like.

First, save your edited document and exit the PO editor. Then, generate the translated document:

```
% ~/freebsddoc/ja\_JP.eucJP/articles/leap-seconds/
% make tran
```

If everything works, you should now have an updated `article.xml` in your directory. It contains all the translated strings so far in the source document.

To create an HTML document in order to view it in a web browser, specify that format. This example generates the HTML then opens it in the Firefox web browser:

```
% make FORMATS=html
% firefox article.html
```

This assumes that you are in a X11 environment and can display the document using a graphical web browser.

Repeat these steps, beginning with editing the po document, to update the translation.

10.5   Creating a Patch for Submission

When the translation is finished, it should be submitted to the FreeBSD project for inclusion into their repository. That way, it becomes part of the official documentation set and other people can read in their native language what you translated for them.

Follow these steps to create a patch for submission.

1. First, add the following line to the PO document:

   ```
   #$FreeBSD$
   ```

   This will later be replaced by the version control system with a date timestamp and author information so that other people know when this file was last changed.

2. Add the PO file, the `Makefile` and generated XML translation to version control as part of your patch:

```
% ~/freebsddoc/ja_JP.eucJP/articles/leap-seconds/
% ls
Makefile article.xml ja_JP.eucJP.po
% svn add Makefile article.xml ja_JP.eucJP.po
A          Makefile
A          article.xml
A          ja_JP.eucJP.po
```

3. Subversion `svn:keywords` properties need to be set on these files to
   the following:

```
% svn propset svn:keywords FreeBSD=%H Makefile article.xml ja_JP.eucJP.po
property 'svn:keywords' set on 'Makefile'
property 'svn:keywords' set on 'article.xml'
property 'svn:keywords' set on 'ja_JP.eucJP.po
```

4. Set the proper MIME types for the files. Usually, this is the UTF-8
   character set, but check for your local language encoding. To prevent
   the version control system from mistaking these files as binary data,
   the `fbsd:notbinary` property must also be set:

```
% svn propset svn:mime-type 'text/x-gettext-translation; charset=UTF-8' ja_JP.eucJP.po
property 'svn:mime-type' set on 'ja_JP.eucJP.po'
% svn propset fbsd:notbinary yes ja_JP.eucJP.po
property 'fbsd:notbinary' set on 'ja_JP.eucJP.po'
% svn propset svn:mime-type 'text/xml; charset=UTF-8' article.xml
property 'svn:mime-type' set on 'article.xml'
% svn propset fbsd:notbinary yes article.xml
property 'fbsd:notbinary' set on 'article.xml'
```

   The `Makefile` is properly recognized and there is no need to add the
   MIME types for this file.

5. Finally, the patch is created from the top level directory:

```
% cd ~/freebsddoc
svn diff a_JP.eucJP/articles/leap-seconds > /tmp/a_JP.eucJP-leap-seconds.diff
```

After these steps, you will find a patch file in `/tmp/a_JP.eucJP-leap-seconds.diff`
that contains all information to generate the translated document. Submit
your patch using the instructions in section 8.1 on page 15.

11   Class Discussion