Apache POI - Contribution Guidelines

by Nick Burch, David Fisher

1. Introduction

1.1. Disclaimer

Any information in here that might be perceived as legal information is informational only. We're not lawyers, so consult a legal professional if needed.

1.2. The Licensing

The POI project is OpenSource and developed/distributed under the Apache Software License. Unlike other licenses this license allows free open source development; however, it does not require you to release your source or use any particular license for your source. If you wish to contribute to POI (which you're very welcome and encouraged to do so) then you must agree to release the rights of your source to us under this license.

1.3. Publicly Available Information on the file formats

In early 2008, Microsoft made a fairly complete set of documentation on the binary file formats freely and publicly available. These were released under the Open Specification Promise, which does allow us to use them for building open source software under the Apache Software License.

You can download the documentation on Excel, Word, PowerPoint and Escher (drawing) from http://www.microsoft.com/interop/docs/OfficeBinaryFormats.mspx. Documentation on a few of the supporting technologies used in these file formats can be downloaded from http://www.microsoft.com/interop/docs/supportingtechnologies.mspx.

Previously, Microsoft published a book on the Excel 97 file format. It can still be of plenty of use, and is handy dead tree form. Pick up a copy of "Excel 97 Developer's Kit" from your favourite second hand book store.

The newer Office Open XML (ooxml) file formats are documented as part of the ECMA / ISO standardisation effort for the formats. This documentation is quite large, but you can

normally find the bit you need without too much effort! This can be downloaded from http://www.ecma-international.org/publications/standards/Ecma-376.htm, and is also under the OSP.

It is also worth checking the documentation and code of the other open source implementations of the file formats.

1.4. I just signed an NDA to get a spec from Microsoft and I'd like to contribute

In short, stay away, stay far far away. Implementing these file formats in POI is done strictly by using public information. Most of this Public Information currently comes from the documentation that Microsoft makes freely available (see above). The rest of the public information includes sources from other open source projects, books that state the purpose intended is for allowing implementation of the file format and do not require any non-disclosure agreement and just hard work. We are intent on keeping it legal, by contributing patches you agree to do the same.

If you've ever received information regarding the OLE 2 Compound Document Format under any type of exclusionary agreement from Microsoft, or received such information from a person bound by such an agreement, you cannot participate in this project. Sorry. Well, unless you can persuade Microsoft to release you from the terms of the NDA on the grounds that most of the information is now publically available. However, if you have been party to a Microsoft NDA, you will need to get clearance from Microsoft before contributing.

Those submitting patches that show insight into the file format may be asked to state explicitly that they have only ever read the publicly available file format information, and not any received under an NDA or similar, and have only made us of the public documentation.

2. I just want to get involved but don't know where to start

- Read the rest of the website, understand what POI is and what it does, the project vision, etc.
- Use POI a bit, look for gaps in the documentation and examples.
- Join the mailing lists and share your knowledge with others.
- Get Subversion and check out the POI source tree
- Documentation is always the best place to start contributing, maybe you found that if the documentation just told you how to do X then it would make more sense, modify the documentation.
- Contribute examples - if there's something people are often asking about on the user list which isn't covered in the documentation or current examples, try writing an example of this and uploading it as a patch.

- Get used to building POI, you'll be doing it a lot, be one with the build, know its targets, etc.
- Write Unit Tests. Great way to understand POI. Look for classes that aren't tested, or aren't tested on a public/protected method level, start there.
- Download the file format documentation from Microsoft - OLE2 Binary File Formats or OOXML XML File Formats
- Submit patches (see below) of your contributions, modifications.
- Check the bug database for simple problem reports, and write a patch to fix the problem
- Review existing patches in the bug database, and report if they still apply, if they need unit tests atc.
- Take a look at all the unresolved issues in the bug database, and see if you can help with testing or patches for them
- Add in new features, see Bug database for suggestions.

The Apache Contributors Tech Guide gives a good overview how to start contributing patches.

The Nutch project also have a very useful guide on becoming a new developer in their project. While it is written for their project, a large part of it will apply to POI too. You can read it at http://wiki.apache.org/nutch/Becoming_A_Nutch_Developer. The Apache Community Development Project also provides guidance and mentoring for new contributors.

3. Submitting Patches

Patches are submitted via the Bug Database. Create a new bug, set the subject to [PATCH] followed by a brief description. Explain you patch and any special instructions and submit/save it. Next, go back to the bug, and create attachements for the patch files you created. Be sure to describe not only the files purpose, but its format. (Is that ZIP or a tgz or a bz2 or what?).

Ideally, patches should be submitted early and often. This is for two key reasons. Firstly, it's much easier to review smaller patches than large ones. This means that smaller patches are much more likely to be applied to SVN in a timely fashion. Secondly, by sending in your patches earlier rather than later, it's much easier to get feedback on your coding and direction. If you've missed an easier way to do something, or are duplicating some (probably hidden) existing code, or taking things in an unusual direction, it's best to get the feedback sooner rather than later! As such, when submitting patches to POI, as with other Apache Software Foundation projects, do please try to submit early and often, rather than "throwing a large patch over the wall" at the end.

You may create your patch file using either of the following approaches (the committers

recommend the first):

3.1. Approach 1 - use Ant

Use Ant to generate a patch file to POI:

```
   ant -f patch.xml
```

This will create a file named patch.tar.gz that will contain a unified diff of files that have been modified and also include files that have been added. Review the file for completeness and correctness. This approach is recommended because it standardizes the way in which patch files are constructed. It also eliminates the chance of you missing to submit new files that constitute part of the patch.

3.2. Approach 2 - the manual way

Patches to existing files should be generated with svn diff filename and save the output to a file. if you want to get the changes made to multiple files in a directory , just use svn diff. then, tar and gzip the patch file as well as any new files that you have added.

If you use a unix shell, you may find the following following sequence of commands useful for building the files to attach.

```
# run this in the root of the checkout, i.e. the directory holding
#  build.xml and poi.pom

# build the directory to hold new files
mkdir /tmp/poi-patch/
mkdir /tmp/poi-patch/new-files/

# get changes to existing files
svn diff > /tmp/poi-patch/diff.txt

# capture any new files, as svn diff won't include them
# preserve the path
svn status | grep "^\?" | awk '{printf "cp --parents %s /tmp/poi-patch/new-files/\n",

# tar up the new files
cd /tmp/poi-patch/new-files/
tar jcvf ../new-files.tar.bz2
cd ..

# upload these to bugzilla
echo "please upload to bugzilla:"
echo "   /tmp/poi-patch/diff.txt"
echo "   /tmp/poi-patch/new-files.tar.bz2"
```

3.3. checklist before submitting a patch

- added code complies with coding standards
- added code compiles and runs on java 1.5
- new java files begin with the apache software license statement.
- the code does not depend on gpl or lgpl code.
- the code includes the @author tag on any files you've altered or created.
- existing test cases succeed.
- new test cases written and succeed.
- documentation page extended as appropriate.
- diff files generated using svn diff
- message to dev contains [patch], task name and patch reason in subject.
- message body contains a rationale for the patch.
- message attachment contains the patch file(s).

4. Mentoring and Committership

The POI project will generally offer committership to contributors who send in consistently good patches over a period of several months.

The requirement for "good patches" generally means patches which can be applied to SVN with little or no changes. These patches should include unit test, and appropriate documentation. Whilst your first patch to POI may require quite a bit of work before it can be committed by an existing committer, with any luck your later patches will be applied with no / minor tweaks. Please do take note of any changes required by your earlier patches, to learn for later ones! If in doubt, ask on the dev mailing list.

The requirement for patches over several months is to ensure that committers remain with the project. It's very easy for a good developer to fire off half a dozen good patches in the couple of weeks that they're working on a POI powered project. However, if that developer then moves away, and stops contributing to POI after that spurt, then they're not a good candidate for committership. As such, we generally require people to stay around for a while, submitting patches and helping on the mailing list before considering them for committership.

Where possible, patches should be submitted early and often. For more details on this, please see the "Submitting Patches" section above.

Where possible, the existing developers will try to help and mentor new contributors. However, everyone involved in POI is a volunteer, and it may happen that your first few patches come in at a time when all the committers are very busy. Do please have patience, and remember to use the dev mailing list so that other contributors can assist you!

For more information on getting started at Apache, mentoring, and local Apache Committers near you who can offer advice, please see the Apache Community Development Project website.