

# Apache POI - Contribution Guidelines

by Nicola Ken Barozzi, Marc Johnson, Andrew C. Oliver, Tetsuya Kitahata

## 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.msp>. Documentation on a few of the supporting technologies used in these file formats can be downloaded from <http://www.microsoft.com/interop/docs/supportingtechnologies.msp>.

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. 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 (possibly illegally) received such information from a person bound by such an agreement, you cannot participate in this project. (Sorry)

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.

#### **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.

## Apache POI - Contribution Guidelines

- 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
- Add in new features, see [Bug database](#) for suggestions.

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](http://wiki.apache.org/nutch/Becoming_A_Nutch_Developer)

### 3. Submitting Patches

Create patches by getting the latest sources from Subversion. Alter or add files as appropriate. Then, from the poi directory, type `svn diff > mypatch.patch`. This will capture all of your changes in a patch file of the appropriate format. However, `svn diff` won't capture any new files you may have added. So, if you've added any files, create an archive (tar.bz2 preferred as its the smallest) in a path-preserving archive format, relative to your poi directory. You'll attach both files in the next step.

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 attachments 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?).

Make sure your patches include the @author tag on any files you've altered or created. Make sure you've documented your changes and altered the examples/etc to reflect them. Any new additions should have unit tests. Lastly, ensure that you've provided appropriate javadoc. (see [Coding Standards](#)). Patches that are of low quality may be rejected or the contributor may be asked to bring them up to spec.

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", $2 }' | sh -s

# 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"
```