Help Wanted

We are looking for volunteers to do the following tasks. But please do not feel limited to these items. For inspiration on other ideas we recommend reading "14 Ways to Contribute to Open Source without Being a Programming Genius or a Rock Star".

If you are interested in working on any of these items, please send a note to the dev@openoffice.apache.org mailing list and we can give you hints on how to accomplish the task.

Easy

1. Submit feedback about the product or website(s). Subscribe to the users mailing list or the developer mailing list. Skills needed: e-mail.
2. Submit problem reports about the product or website(s). Register for an Apache OpenOffice Bugzilla account and submit an "issue". Skills needed: ability to learn Bugzilla.
3. Help existing users. Skills needed: a working knowledge of OpenOffice, and a desire to help other users. Subscribe to the users mailing list, or join the support forum, or help users that write to the Facebook page.
4. Test Developer Builds. See the source code webpage for more information on testing developer builds. The easiest way to test is to use "Full Installation Sets" for your OS environment. Install to a separate area (directory) so as not to overwrite your existing AOO installation.
5. Improve the look of our website(s), especially the default landing page. Skills needed: graphic design, web design, good taste, possible svn.
6. Create a more friendly "Develop" landing page for the website (related to the previous item). Skills needed: webpage authoring including graphic design, svn for committers, or the Apache CMS to submit a webpage for others. See discussion and proposed mockup from the developer mailing list.
7. Help market Apache OpenOffice by developing artwork or other marketing materials. See the marketing webpage or review the marketing mailing list for more information. Get creative!
8. Help plan the future of Apache OpenOffice. Subscribe to the developer mailing list and apply for a Apache OpenOffice Planning Wiki account. Skills needed: e-mail, Confluence wiki desirable (but easy to learn). Skills gained: Confluence wiki, cooperation.

Medium

1. Write Documentation. This includes user guides, install guides and developer documentation. Documentation is found in the documentation area of the OpenOffice wiki. To participate through the OpenOffice Documentation Wiki, register for an Apache OpenOffice Community Wiki account. Skills needed: familiarity with, or willing to learn MediaWiki. Skills gained: more wiki experience, group collaboration.
2. Update the OpenOffice Administration Guide.
3. Develop a trademark approval process. This would be used by the PMC for external entities that want to use our trademarks. Write up the process, review with Apache Branding, and get consensus on the ooo-dev list. Add the process definition to the Community FAQ's on the website. Things to consider include criteria for approval and record keeping. A good starting point might be this email. Skills gained: greater understanding of Apache branding requirements.
4. Write up a tutorial (or link to a good existing one) that explains how new project volunteers should create and submit patches (for publication on the Project website). Skills needed: an understanding of Markdown and probably committer rights to the project website. (Already somewhat done on the Developer FAQs webpage).
5. Write a tutorial on how committers should review and apply patches. Add a link to this tutorial from the Developer FAQs webpage. Skills needed: SVN, technical writing. Skills gained: Apache CMS.
6. Document the procedures used by the PMC. This would include the processes: how to elect new committers and PMC members, including form letters, userid generation, tracking, etc. Link it from the PMC FAQs webpage on the website. Consider the steps that the PMC needs to do, as well as the new committer, so this can be a reference for both. Skills needed: writing. Skills gained: Apache CMS. Note: this task is suitable for existing PMC members only.

Challenging

1. Document or find tools. Specifically, external diff and merge tools for Subversion that can process ODF documents.
2. Fix bugs, create and update OpenOffice. Skills needed: an Apache OpenOffice Bugzilla account, git or github knowledge, C++ (Apache OpenOffice is mainly written in C++, though there is some code in Java and Python). See the Developer FAQ to get started.
3. Build ApacheOffice from source. We can use feedback on the build process. Skills needed include familiarity with C++, and your system's batch or shell processing capabilities. See the Source Control webpage for more information on obtaining and building Apache OpenOffice.