Chukwa How To Contribute

Guide for Chukwa Committers

This page contains Chukwa specific guidelines for committers.

New committers

New committers are encouraged to first read Apache's generic committer documentation:

- Apache New Committer Guide
- Apache Committer FAQ

The first act of a new core committer is typically to add their name to the credits page. This requires changing the XML source in http://svn.apache.org/repos/asf/incubator/chukwa/site/author/src/documentation/content/xdocs/credits.xml. Once done, update the Chukwa website as described here.

Review

Chukwa committers should, as often as possible, attempt to review patches submitted by others. Ideally every submitted patch will get reviewed by a committer within a few days. If a committer reviews a patch they've not authored, and believe it to be of sufficient quality, then they can commit the patch, otherwise the patch should be cancelled with a clear explanation for why it was rejected.

The list of submitted patches is in the Chukwa Review Queue. This is ordered by time of last modification. Committers should scan the list from top-to-bottom, looking for patches that they feel qualified to review and possibly commit.

For non-trivial changes, it is best to get another committer to review your own patches before commit. Use "Submit Patch" like other contributors, and then wait for a "+1" from another committer before committing.

Reject

Patches should be rejected which do not adhere to the guidelines in HowToContribute and to the CodeReviewChecklist. Committers should always be polite to contributors and try to instruct and encourage them to contribute better patches. If a committer wishes to improve an unacceptable patch, then it should first be rejected, and a new patch should be attached by the committer for review.

Commit

When you commit a patch, please:

1. Add an entry in CHANGES.txt, at the end of the appropriate section. This should include the Jira issue id, and the name of the contributor.
2. Include the Jira issue id in the commit message, along with a short description of the change and the name of the contributor if it is not you. Be sure to get the issue id right, as this causes Jira to link to the change in Subversion (use the issue's "All" tab to see these).
3. Resolve the issue as fixed, thanking the contributor. Always set the "Fix Version" at this point, but please only set a single fix version, the earliest release in which the change will appear. **Special case**- when committing to a *non-mainline* branch (such as branch-0.5.0), please set fix-version to either 0.5.x or 0.6.x appropriately too.
4. Use the -E option to make sure that empty files are removed during the commit.

Committing Documentation

Chukwa's official documentation is authored using Maven APT format. To commit documentation changes you must have Maven installed and the `mvn` executable on your `$PATH`. Documentation is of two types:

1. End-user documentation, versioned with releases; and,
2. The website. This is maintained separately in subversion, republished as it is changed.

To commit end-user documentation changes to trunk or a branch, ask the user to submit only changes made to the *.apt files in `src/docs`. Apply that patch, run `mvn site` to generate the html, and then `mvn site-deploy` to publish. End-user documentation is only published to the web when releases are made, as described in Chukwa_How_To_Release.

To commit changes to the website and re-publish them:

```
svn co https://svn.apache.org/repos/asf/incubator/chukwa/site
cd site
mvn site
firefox target/index.html # preview the changes
svn stat                   # check for new pages
svn add                    # add any new pages
svn commit
mvn site-deploy
```

Changes to website (*via mvn site-deploy*) might take upto an hour to be reflected on Chukwa site.

Backporting commits to previous branches

If a patch needs to be backported to previous branches, follow these steps.

1. Commit the changes to trunk and note down the revision number, say 4001. (Revision number is displayed as response to your *svn commit* command). 2. Check out the desired branch and execute this command from the root directory.

   ```
   svn merge -r 4000:4001 https://svn.apache.org/repos/asf/incubator/chukwa/trunk .
   # Now resolve any merge conflicts.
   # If major edits are needed, produce a new patch and upload it to the Jira.
   svn diff CHANGES.txt # get all Jira numbers included in this merge
   svn commit -m "merge <list all Jira numbers here>"
   ```

Please be sure to include Jira number(s) in the commit message for merge commits. Sometimes developers just put something like "merge -r 4000:4001" in the merge message, which fails to trigger the Jira/Subversion integration, so the Jira doesn't record the branch commit. It is important to link to the Jira number, so that when looking at the Jira it will be clear that this patch has been merged to this branch.

Dialog

Committers should hang out in the #chukwa room on irc.freenode.net for real-time discussions. However any substantive discussion (as with any off-list project-related discussion) should be re-iterated in Jira or on the developer list.