Working With Git Source Control

*** This page is in a draft state to be reviewed by others before becoming official. ***

Overview

The VCL project as adopted the Gitflow Workflow that is explained very thoroughly elsewhere. We recommend reading over that documentation before reading this page as it assumes an understanding of the Gitflow model.

VCL's Git repository is officially found at the ASF Git repo:

https://git-wip-us.apache.org/repos/asf/vcl.git

It is also mirrored on GitHub:

https://github.com/apache/vcl

The **master** branch of VCL should always be stable and only contains released code.

The **develop** branch of VCL should normally be fairly stable. It should not be developed on directly. Development should be done in feature branches that are then merged back in to the develop branch once work on the feature is complete. Obviously, there could be bugs in the feature, but an effort should be made to test things before merging a feature branch back in to develop.

Working on a Feature or Bug

When working on a feature or bug fix, a JIRA issue must be created to for tracking it.

A Git branch should be created whose name includes the JIRA issue number and a name the give an idea of what is being done. If the JIRA issue name is short, it should match the name, leaving out any characters not allowed in branch names and replacing spaces with underscores.

Example: The issue named **VCL-1087 VCL CAS SSO** would have a branch named **VCL-1087_VCL_CAS_SSO**.

If the issue is more than a small amount of work, the branch should be pushed to the ASF repository while work is being done so that other committers can view the work.

Once the work is complete and tested, the branch should be merged into the develop branch. A **merge commit** should be done rather than a squash commit so that comments from the work can be retained.

It is advisable to continue to follow the develop branch while working on a feature branch so that the merge back into develop with go smoothly. Whether this is done by merging develop into the feature branch or rebasing the feature branch is left up to the committer at the time the work is done.

Releases

When it is time for a release to be done, the release manager will create a release branch, ensuring it is pushed to the ASF repo. Only bug fixes and documentation type changes should be made to the release branch. Release candidates will be made from this branch. Once a vote on a release candidate passes, the branch will be merged into the master branch and the final release will be cut.

Hot Fixes

Hot fixes should be handled by branching a new release branch directly off of the master branch. As long as the fix required for the bug is not particularly involved, work can be done directly on this release branch to fix the bug. If the work is fairly involved, a bug fix branch should be created off of the release branch for the work. Once the bug is fixed, the normal release process should be followed. After a release candidate has passed a vote, the release manager must ensure that the bug fix is also merged into the develop branch if it has not already been merged in.