How to Merge a Feature Branch with Master for Test and UAT

On how to merge a feature development branch with the master branch for test and UAT, using Stash and SourceTree.

Introduction

This merging is done after the feature development is completed the code is ready to be merged with the master branch for test and UAT.

What is involved is, pulling the latest changes from the master branch onto the feature development branch, resolving any conflicts with the merge, building, debugging and testing the merged changes locally and then pushing the merged changes back to the remote feature branch. Afterwards a pull request is created to merge the feature branch with the master branch to finalise the merge.

The master branch represents the production code and development is usually done in feature branches. A feature branch would have the name:

feature/EN-xxxxxx-Feature-Name

The ‘xxxxxx’ represents the actual Jira issue number and ‘Feature-Name’ is a description of the feature being developed.

Merging with Master

Using SourceTree, ensuring that the feature branch is the active branch, do a Pull request. In the Pull dialog that opens, ‘Pull from remote’ should have the value ‘origin’, ‘Remote branch to pull’ should have the value ‘master‘ and ‘Commit merged changes immediately’ should be checked, then press OK.

For xxx only, there are three different master branches, so care needs to be taken to ensure that the correct master branch is pulled from. For the xxx, xxx, ‘master’ is the correct master branch, for xxx the master branch to use is ‘xxx’ and for version 3 it is ‘xxx’.

Once the pull from the master branch finishes, the ‘Unstaged files’ list should show any conflicting files with a yellow warning icon next to it and the code view on the right on SourceTree should show the code details of the conflict.

These conflicting files can be opened in a code editor such as Visual Studio so that the conflicting lines of code can be edited and resolved. Once the conflicts are resolved, mark them as resolved in SourceTree by right-clicking on the file and selecting ‘Resolve Conflicts’ > ‘Mark Resolved’. Confirm when prompted.

Once any conflicts are resolved, build the solution and re-test the changes locally to ensure that there are no issues after merging the feature branch with the master branch locally. Now the changes can be pushed back to the same remote feature branch.

Stage the files that need to pushed and select the Push command. Ensure that the Push checkbox is checked and the ‘Local branch’ and the ‘Remote branch’ both refer the same feature branch. Ensure that ‘Push all tags’ is checked and press the ‘Push’ button. The local feature branch merge changes with master will then be pushed to the remote feature branch.

Now that the master branch has been merged with the feature branch on Stash, a pull request can be created to finalise this merge.

Creating the Pull Request

Using the web browser, go to the related Stash repository, e.g. ‘xxx’. From the ‘ACTIONS’ menu on the left side click on ‘Create pull request’. In the ‘Create pull request’ screen, click on “Select branch” and select the feature branch to merge back with the master branch. Once the feature branch is selected, it will show the changes that would be merged with the master branch.

Click on ‘Continue’ and a description should be provided along with the name of the developer who would review and approve the pull request. Then click on ‘Create’ to create the pull request.

Once the reviewer approves and merges the pull request, there will be notifications by email.

Meeting Recordings

There were meetings on the 08th of June 2022 and this document is based on those meetings. The recordings of these meetings are available at this network location:

xxx

They are in the folder named “2022-06-08 How to Merge a Feature Branch with Master for Test and UAT” at the above network location.