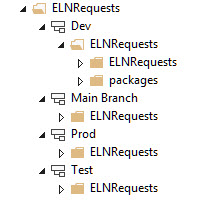
This post was published to Code Blog at 11:41:00 AM 9/26/2016

Branching Technique

Category VSTS

​Here is the branching technique recommended by Nick from Microsoft.

Here is a simple example of what the TFS folder structure should look like.  Note that the top-level Dev/Test/Prod/Main nodes are branch nodes, not folder nodes.



**Steps to create a branching technique for a new project**

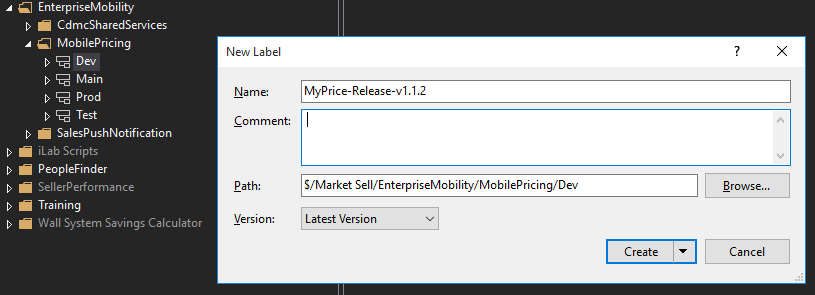
1. Create a folder in TFS with the name of your project
2. Create a subfolder called main. Check in the folders you just created.
3. Now right click on the main folder > branching and merging > convert to branch
4. The main branch is where all the stable code lives. It is also the branch that gets deployed to the dev web farm server. Everything branches off    of main.
5. Create a new project under the main branch (file > new > project), or add an existing project (rt click main > Add items to folder)
6. Check project in
7. Now you are ready to create the dev branch from main. This is where all the developers will do there work.
8. Right click main branch > branching and merging > branch
   1. source branch = main
   2. target branch = dev
9. Now you are ready to create the test branch from main. This is where all the stable code lives that gets deployed to the test web farm.
10. Right click main branch > branching and merging > branch
    1. source branch = main
    2. target branch = test
11. Now you are ready to create the prod branch from main. This is where all the stable code lives that gets deployed to the prod web farm.
12. Right click main branch > branching and merging > branch
    1. source branch = main
    2. target branch = prod

**Developer Branch Merging Steps for a Release**

1. The developers gets the latest code on dev that has all the feature checked in for the release.
2. Think of a good name for the release like 'MyPrice-Release-v1.1.2'. If your build enviornment has a build number, use that in the release.
3. Each project should have a librarian who is repsonsible for merging completed features from dev to main. Notify the Librarian that you new feature is ready to be merged to main. The steps beloiw are the step for the Librarian.

**Steps for the Librarian to merge and label the release**

1. You are ready to build a release (dev/test/prod...), so its time to do a reverse integration to main from dev.
2. Right click dev branch > branching and meging > merge source branch = dev target branch = main
3. Next > Next > Finish
4. Now your dev code is merged into the main branch...but its not on the server yet. At this point you should test the main branch locally and make sure it runs and build.
5. Right click Main branch > Check in > use descriptive comment (maybe use the same description as the label, i.e "MyPrice-Release-v1.1.2")
6. Apply a label to the dev branch with the name of the release or feature. Right click the branch > advanced > apply label > use a descriptive name, maybe follow this standard AppName-FeatureDescription-ReleaseX for example, MyPrice-Release-v1.1.2 (Make sure to choose the appropraite path)



1. This will check in your release to source control. It you have continuous integration and a build defintion created in VSTS it should auto build the app on the VSTS build server. If the build succeeds and you have Continuous Deployment and a release confiugred it will deploy your code to an azure website.
2. If code works properly in main, then its time to build to test and let users test the new feature.
3. You are ready to build to test web farm, do a forward integration from main, by right click main branch > branching and meging > merge source branch = main target branch = test
4. Next > Next > Finish
5. Right click Test branch > Check in > use descriptive comment (maybe use the same description as the label, i.e "CdmcSharedServices-EmailPdfFeature-Release1")
6. Run CI (Continuous Integration i.e. VSTS build on check in) and CD (Continuous Deployment/Delivery i.e. VSTS release) process.
7. If the users say your app is awesome then build to prod. If there is a bug then fix the issue on the main branch and merge the fix to test. Try not to fix the bug in dev because another developer may have implemented new features in the time (maybe a few days) that it took the user to test. After you prove the bug is fixed then merge the fix from main to dev.
8. You are ready to build to prod web farm, do a forward integration from main, by right click main branch > branching and meging > merge source branch = main target branch = prod
9. Next > Next > Finish
10. Right click Prod branch > Check in > use descriptive comment (maybe use the same description as the label, i.e "CdmcSharedServices-EmailPdfFeature-Release1")
11. Run CI (Continuous Integration i.e. VSTS build on check in) and CD (Continuous Deployment/Delivery i.e. VSTS release) process. most likely this process is a bit more gated when moving to prod...you should have an approval process/email in place in VSTS.
12. Done, the release is pushed, pat yourself on the back and go get some coffee

**Steps to do a forward integration (i.e Merge code from Main back to Dev)**

You should only have to do this if you had to quickly fix a bug you fixed in the main/test or production branch. Almost all of your code updates should be in the dev branch.

1. Get latest/ Check in your code in the main branch
2. Get latest on dev branch.
3. Right click main branch >branching and mergeing >merge
   1. - Source Branch = main
   2. All Changes up to a specific version
   3. Target Branch = dev
4. Next > latest version > next > finish
5. Fix any merge conflicts , or let them resolve manually