AIG DevOps PlayBook

This document provides the guidance to operate the DevOps pipeline built for Retirement Day-In and Day out by roles

PSO reference

# Table of Contents

[Table of Contents 2](#_Toc479176783)

[Introduction 5](#_Toc479176784)

[DevOps high level process flow \* 5](#_Toc479176785)

[Solution View 5](#_Toc479176786)

[User Access Set-up 5](#_Toc479176787)

[Git Notification Set-up 5](#_Toc479176788)

[What tasks do I need to for my role 5](#_Toc479176789)

[Developer 5](#_Toc479176790)

[Database Code 5](#_Toc479176791)

[Setup 5](#_Toc479176792)

[Create Local Copy 5](#_Toc479176793)

[Work on a new element 7](#_Toc479176794)

[Checkout Feature Branch 7](#_Toc479176795)

[Develop 7](#_Toc479176796)

[Make Changes 7](#_Toc479176797)

[Fetch upstream updates 8](#_Toc479176798)

[Commit/Push 8](#_Toc479176799)

[Build 9](#_Toc479176800)

[Request Merge 10](#_Toc479176801)

[.NET code 11](#_Toc479176802)

[Setup 11](#_Toc479176803)

[Create Local Copy 11](#_Toc479176804)

[Work on a new element 12](#_Toc479176805)

[Synchronize Feature Branch 12](#_Toc479176806)

[Develop 12](#_Toc479176807)

[Make Changes 12](#_Toc479176808)

[Fetch upstream updates 13](#_Toc479176809)

[Commit/Push 14](#_Toc479176810)

[Request Merge 15](#_Toc479176811)

[Tester – N/A 15](#_Toc479176812)

[Tech Lead 15](#_Toc479176813)

[Database Code 15](#_Toc479176814)

[Create Feature Branch 15](#_Toc479176815)

[Create Release Branch 16](#_Toc479176816)

[Process Pull requests to Release 17](#_Toc479176817)

[Create Pull Request in INT 18](#_Toc479176818)

[Remove Release branch 18](#_Toc479176819)

[Remove Feature Branch 18](#_Toc479176820)

[.NET Code 18](#_Toc479176821)

[Create Feature Branch 18](#_Toc479176822)

[Create Release Branch 19](#_Toc479176823)

[Process Pull requests 20](#_Toc479176824)

[Build 21](#_Toc479176825)

[Create Pull Request in INT 22](#_Toc479176826)

[Remove Feature Branch 22](#_Toc479176827)

[CM/RM 22](#_Toc479176828)

[Branch Set-up 22](#_Toc479176829)

[Jenkins Set-up for branch 23](#_Toc479176830)

[Database Code 23](#_Toc479176831)

[Process Pull requests to QA 23](#_Toc479176832)

[.NET Code 23](#_Toc479176833)

[Process Pull requests to INT 23](#_Toc479176834)

[DBA 23](#_Toc479176835)

[Database code 23](#_Toc479176836)

[Process Pull requests to INT 23](#_Toc479176837)

[Create Pull Request in QA 23](#_Toc479176838)

[Conflict Resolutions 24](#_Toc479176839)

[Administrative 24](#_Toc479176840)

[GITHUB 24](#_Toc479176841)

[Tech Contact 24](#_Toc479176842)

[Admin Contact 24](#_Toc479176843)

[Jenkins 24](#_Toc479176844)

[Tech Contact 24](#_Toc479176845)

[Admin Contact 24](#_Toc479176846)

[JIRA 24](#_Toc479176847)

[Tech Contact 24](#_Toc479176848)

[Admin Contact 24](#_Toc479176849)

[XLR 24](#_Toc479176850)

[Tech Contact 24](#_Toc479176851)

[Admin Contact 24](#_Toc479176852)

[XLD 24](#_Toc479176853)

[Tech Contact 24](#_Toc479176854)

[Admin Contact 24](#_Toc479176855)

[FAQ 24](#_Toc479176856)

# Introduction

# DevOps high level process flow \*

Provide a sequence or process flow diagram going from JIRA to XLD and XLR

# Solution View

Must cover JIRA, TFS, GIT, Artifactory, XLD and XLR

# User Access Set-up

* Roles and Responsibilities
* Access controls, permissions and grants

# Git Notification Set-up

# What tasks do I need to for my role

## Developer

### Database Code

This section provides step-by-step instruction for a developer to perform their actions working with git repositories

#### Setup

* Please ensure that you have access to “github.aig.net”
* Also ensure that you have the GIT desktop downloaded and installed on your local machine
* Supply the Enterprise login ID and password along with the necessary information to GIT desktop so that it can establish the necessary connections for further work

##### Create Local Copy

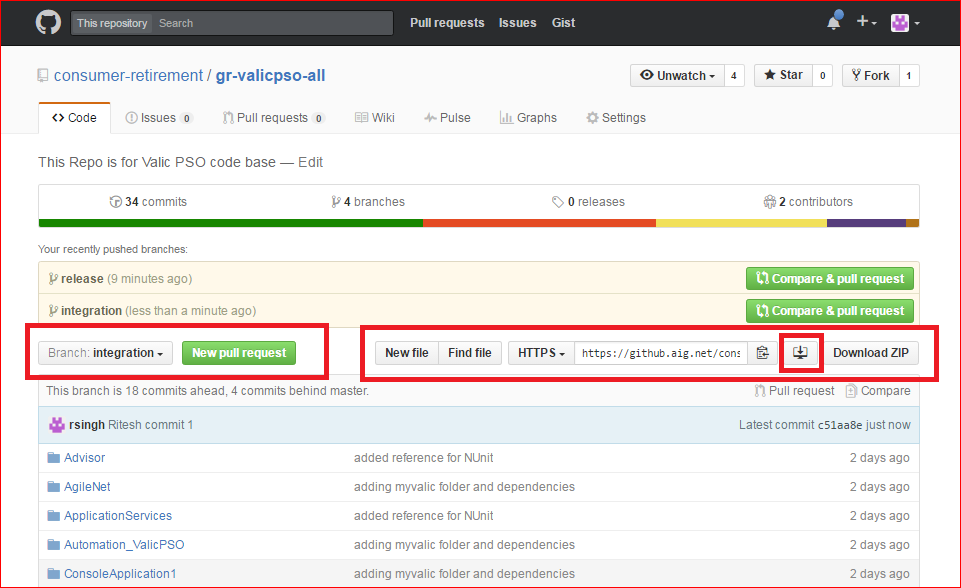
Git UI Steps:

A local copy is created by the process of “cloning” the remote repository from the branch that was created by the tech lead

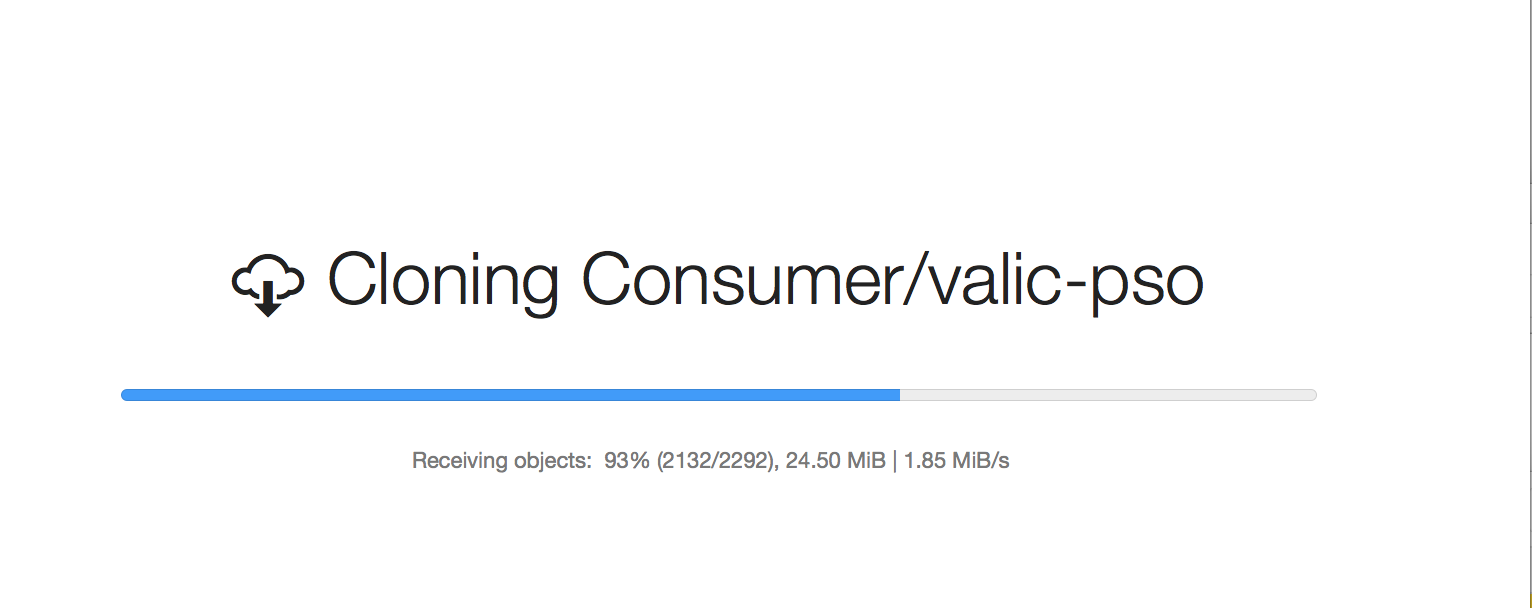
Command:

**Git UI Steps**

1. If you have a previous local copy (clone) for which work has already been completed, please delete the local copy using Windows Explorer of Mac Finder
   1. **WARNING:** If you have not completed the previous work and start cloning then you will lose the changes that were made to the local copy from the previous time
2. Go to <https://github.aig.net/consumer-retirement/gr-valicpsoall>
3. Click on the “download” icon on the RIGHT of the frame as shown below



1. Upon download the GIT desktop should automatically open up to show the cloning progress

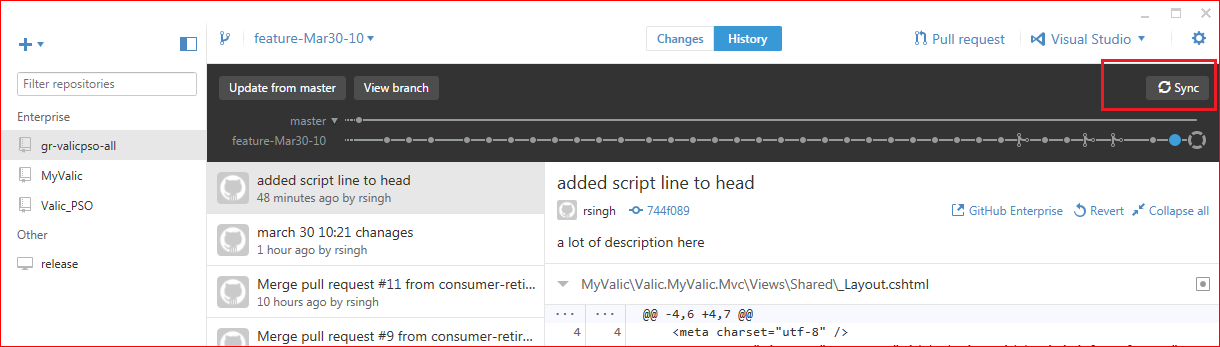


#### Work on a new element

##### Checkout Feature Branch

Git UI Steps

1. The “Synch” action helps synchronize your work with the remote repository
2. “Synch” is a two-way action whereby it would fetch changes in remote repository to your local repository and also helps to push your changes to remote repository
   1. This is the equivalent of “git fetch” and “git push” commands

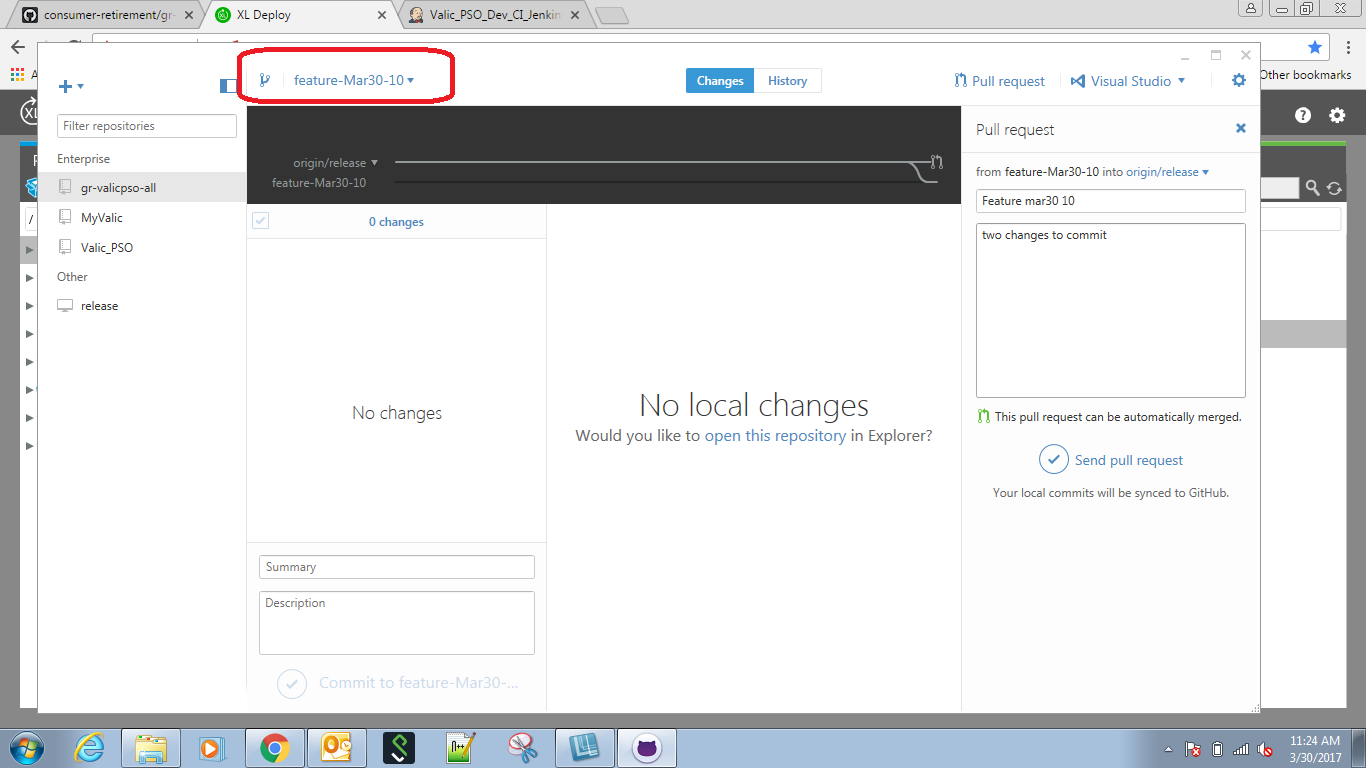


##### Develop

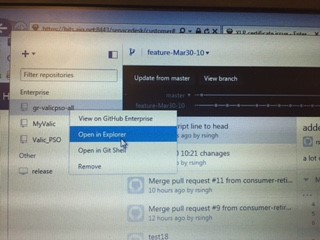
###### Make Changes

Git UI Steps

1. Choose the branch on your local repository in which you want to work on



1. Open the local repository in either Windows Explorer or Mac Finder

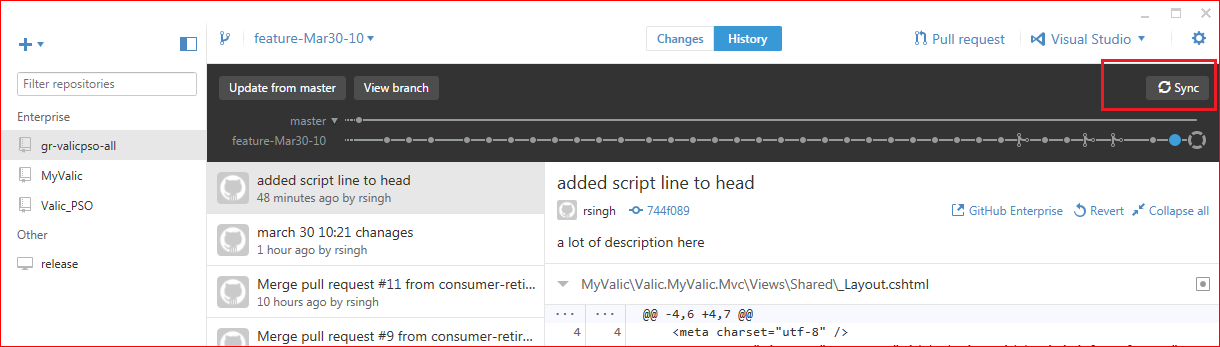


1. Complete the development activities using the tools that are needed and preferred and save your changes as usual

###### Fetch upstream updates

Git UI Steps:

1. At this point you want to make sure that you are able to merge your development/changes along with changes that done by other developers who may have been working on the same code as yours. The “Synch” action helps synchronize your work with the remote repository

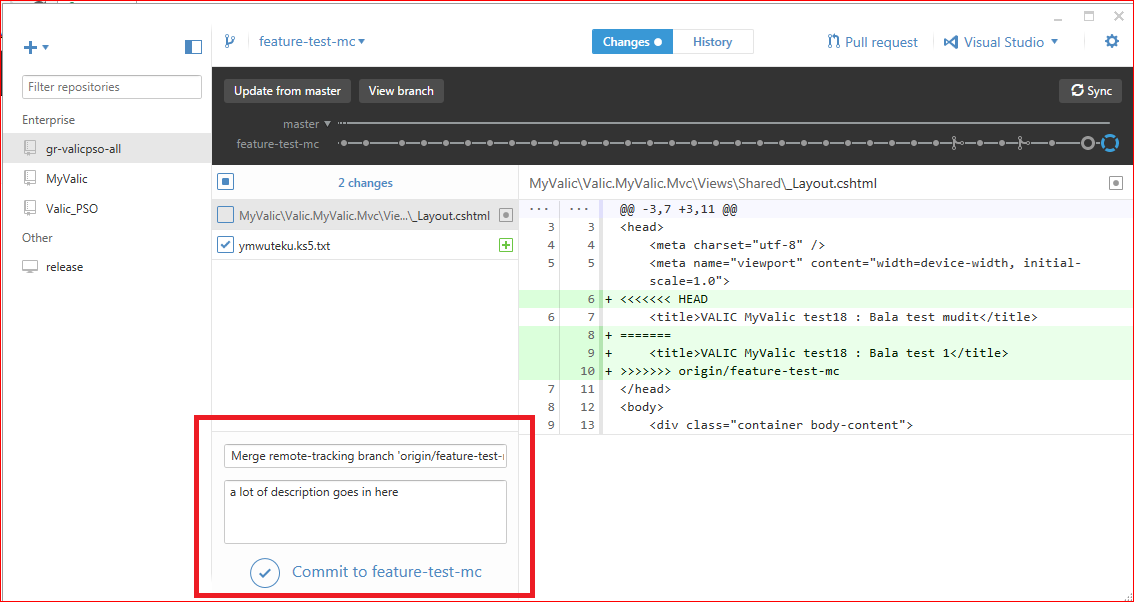


1. Merge all changes as required and proceed to next step

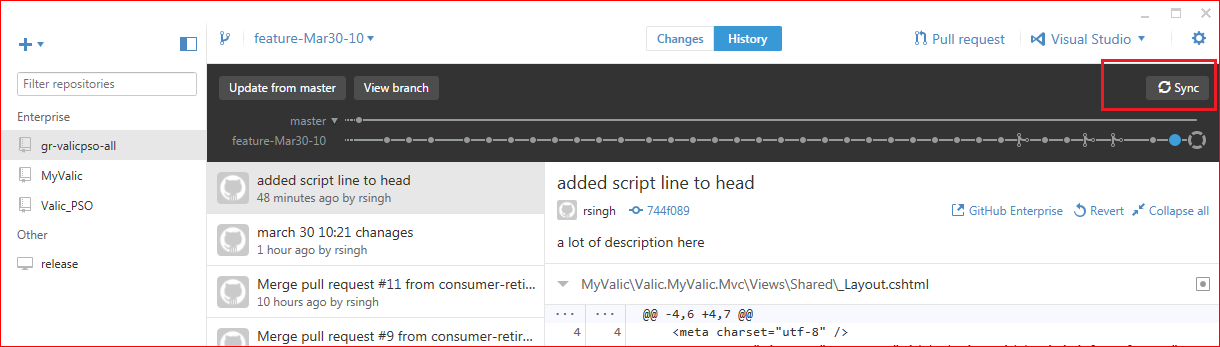
###### Commit/Push

Git UI Steps

1. Commit your changes



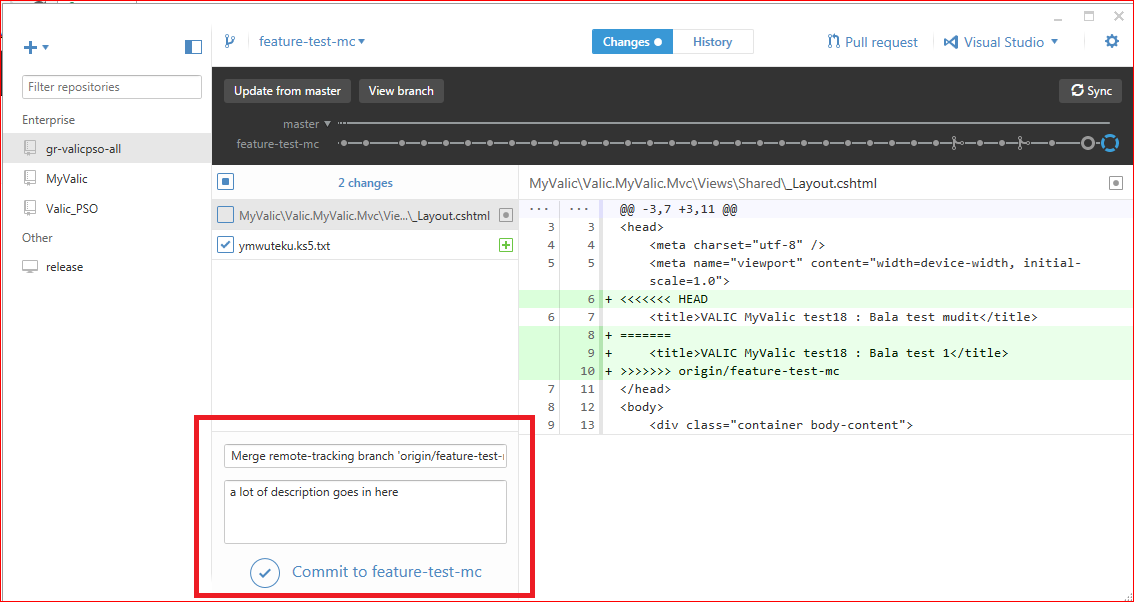
1. Push changes to remote repository



###### Build

Git UI Steps

1. Commit your changes

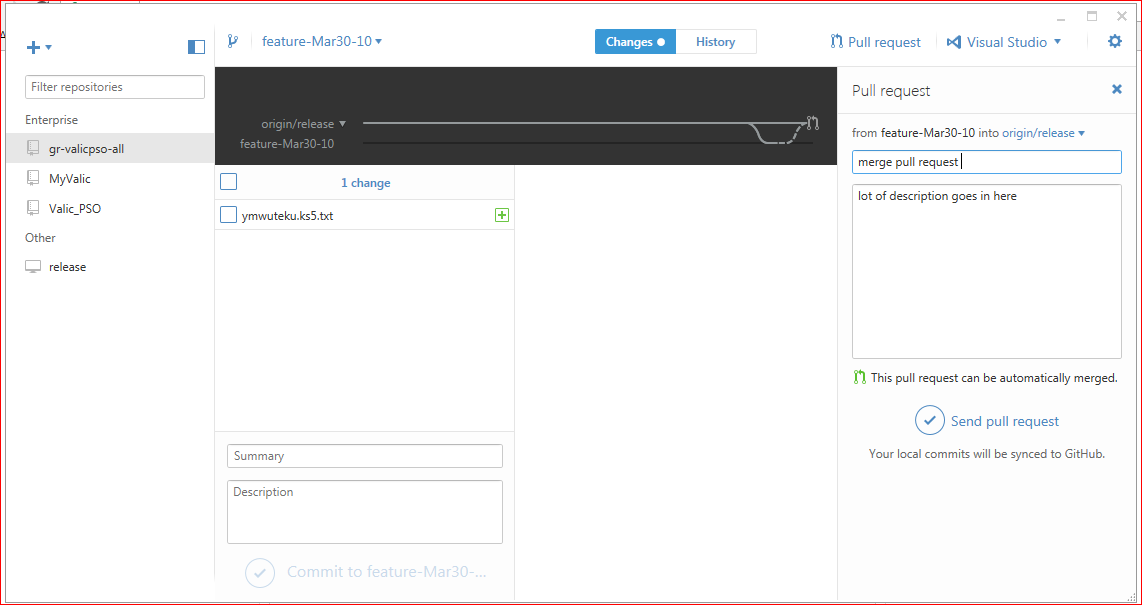


1. Push changes to remote repository

##### Request Merge

Git UI Steps

1. Upon completing committing and pushing your work, create a “Pull Request” to inform the Tech Lead to “pull” the changes that you have done into the Release branch



### .NET code

This section provides step-by-step instruction for a developer to perform their actions working with git repositories

#### Setup

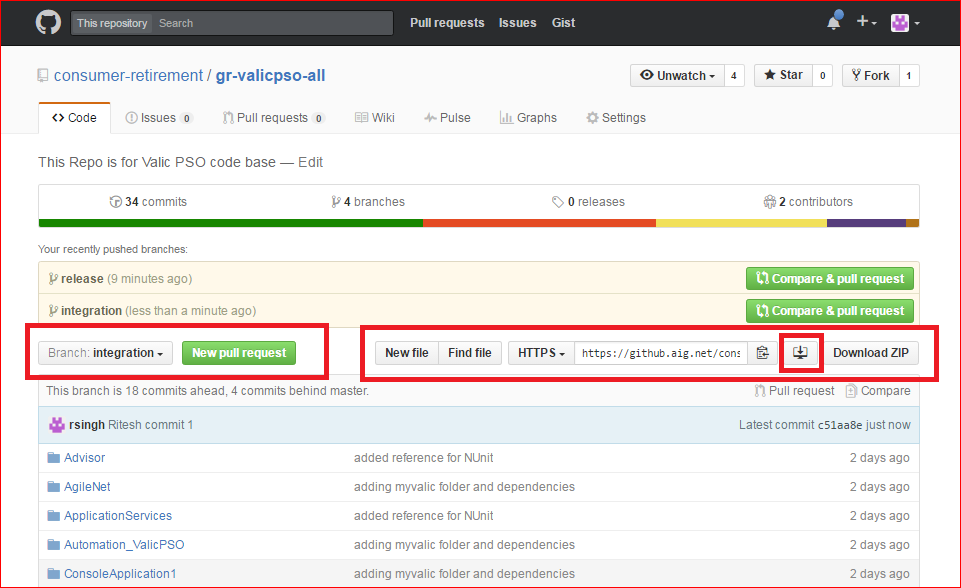
* Please ensure that you have access to “github.aig.net”
* Also ensure that you have the GIT desktop downloaded and installed on your local machine
* Supply the Enterprise login ID and password along with the necessary information to GIT desktop so that it can establish the necessary connections for further work

##### Create Local Copy

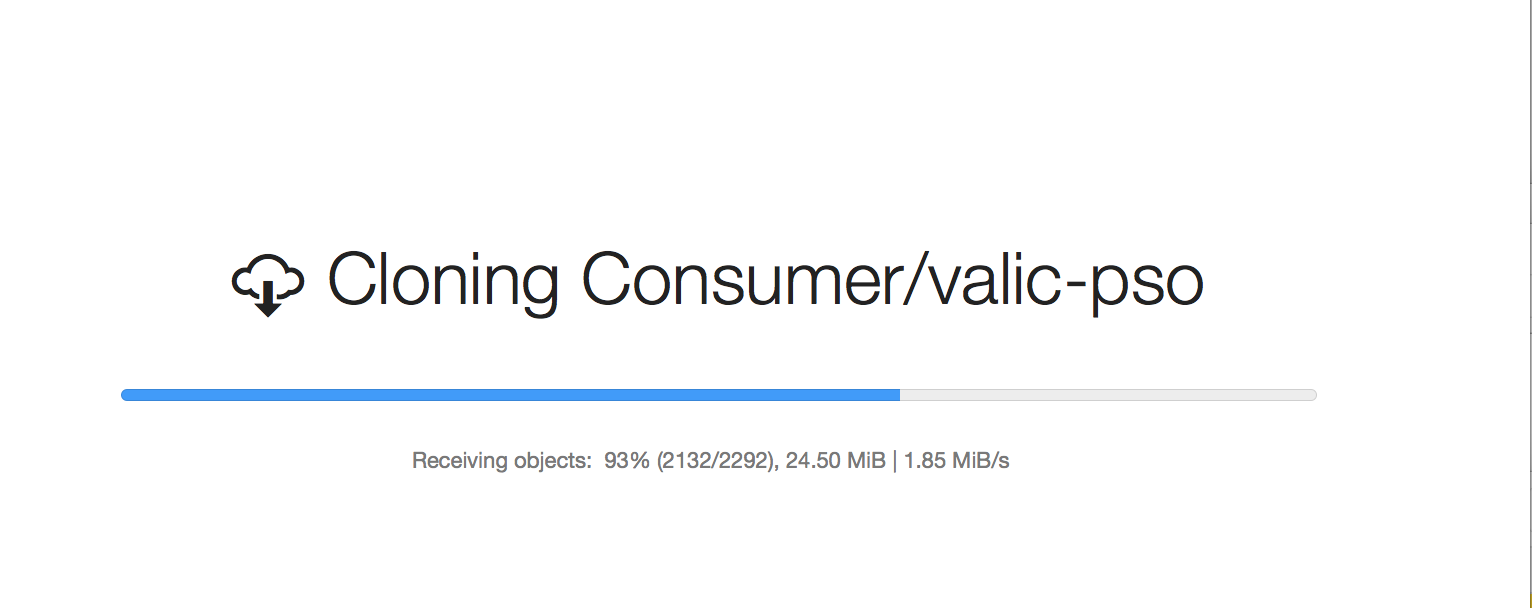
A local copy is created by the process of “cloning” the remote repository from the branch that was created by the tech lead

**Git UI Steps**

1. If you have a previous local copy (clone) for which work has already been completed, please delete the local copy using Windows Explorer of Mac Finder
   1. **WARNING:** If you have not completed the previous work and start cloning then you will lose the changes that were made to the local copy from the previous time
2. Go to <https://github.aig.net/consumer-retirement/gr-valicpsoall>
3. Click on the “download” icon on the RIGHT of the frame as shown below



1. Upon download the GIT desktop should automatically open up to show the cloning progress



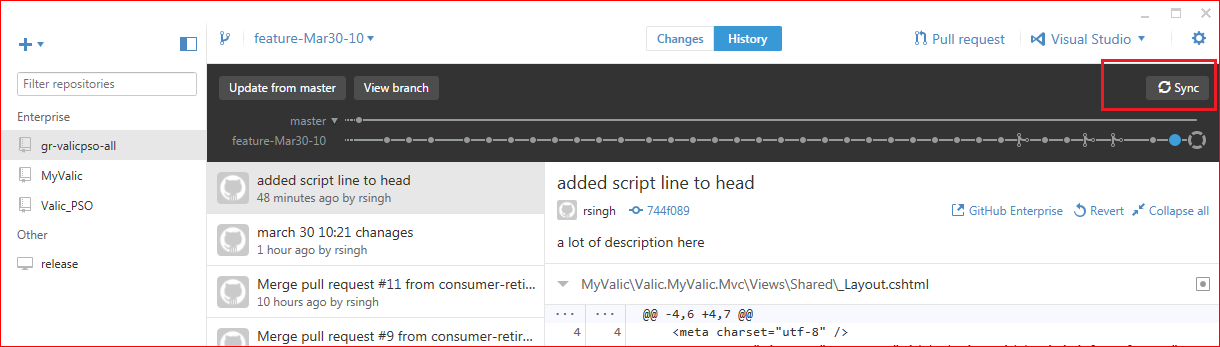
#### Work on a new element

##### Synchronize Feature Branch

Command:

Git UI Steps

1. The “Synch” action helps synchronize your work with the remote repository
2. “Synch” is a two-way action whereby it would fetch changes in remote repository to your local repository and also helps to push your changes to remote repository
   1. This is the equivalent of “git fetch” and “git push” commands

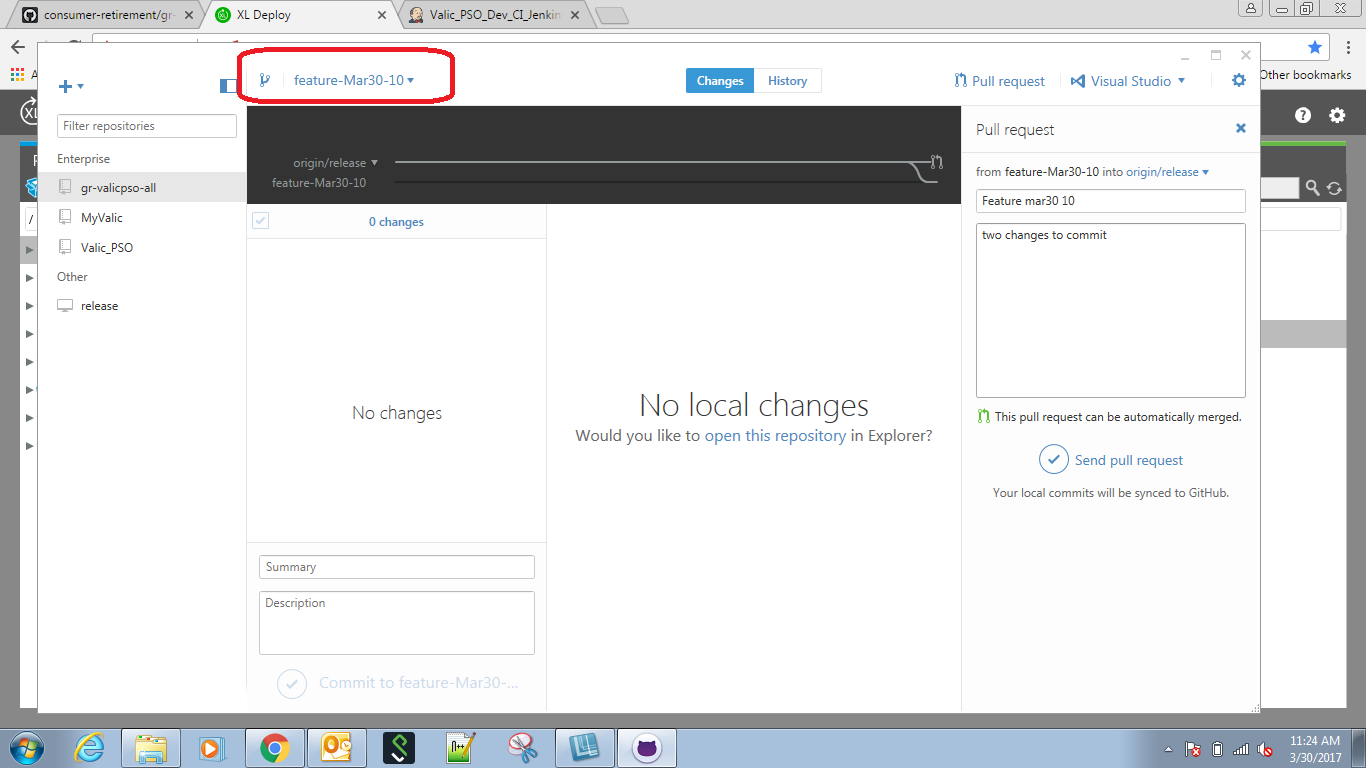


##### Develop

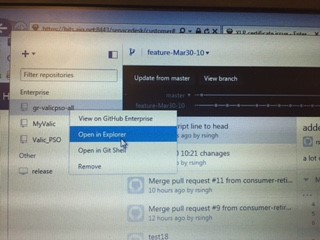
###### Make Changes

Git UI Steps

1. Choose the branch on your local repository in which you want to work on



1. Open the local repository in either Windows Explorer or Mac Finder

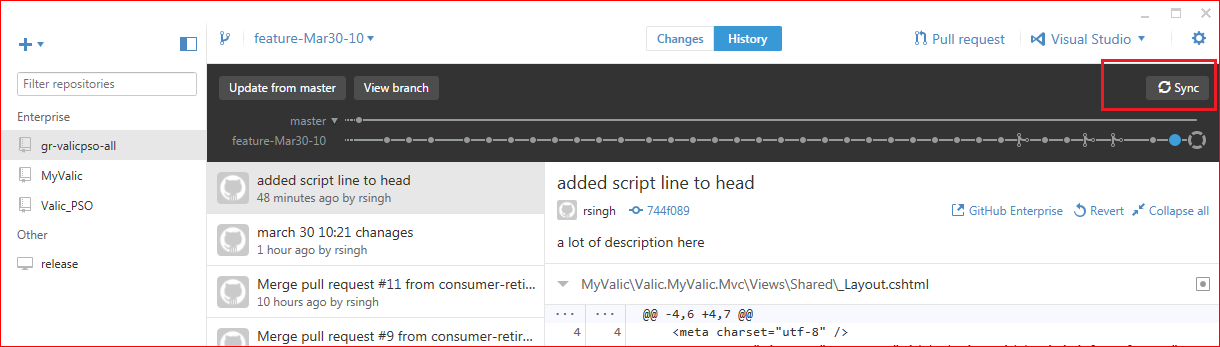


1. Complete the development activities using the tools that are needed and preferred and save your changes as usual

###### Fetch upstream updates

Git UI Steps

1. At this point you want to make sure that you are able to merge your development/changes along with changes that done by other developers who may have been working on the same code as yours. The “Synch” action helps synchronize your work with the remote repository

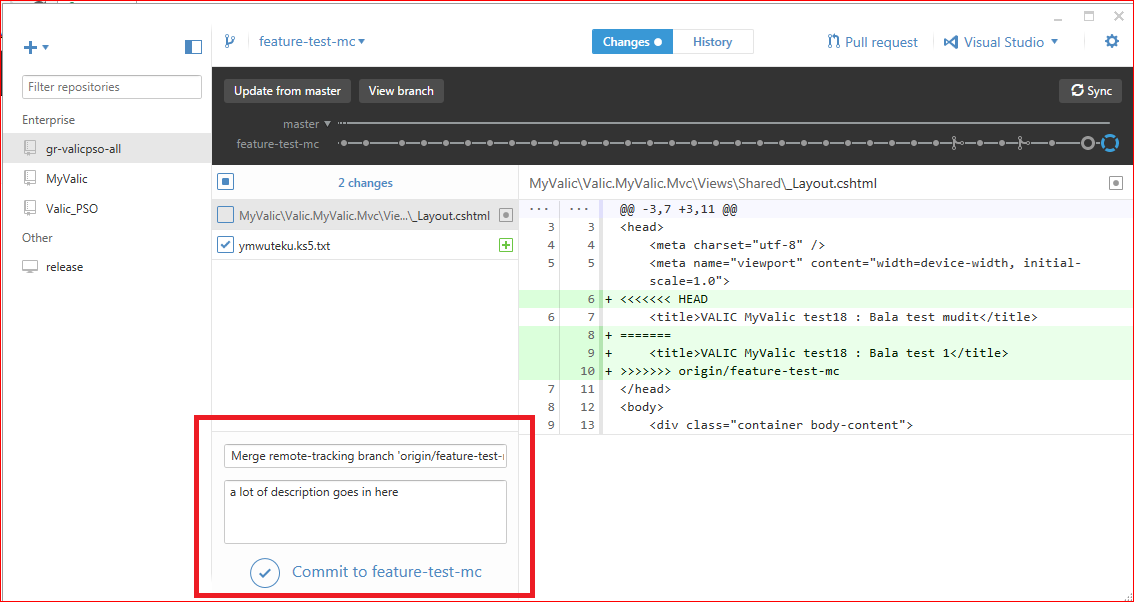


1. Merge all changes as required and proceed to next step

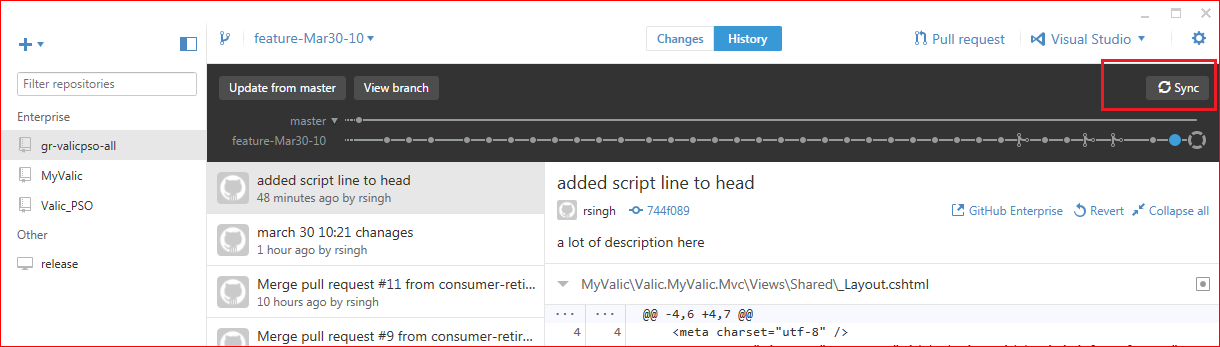
###### Commit/Push

Git UI Steps

1. Commit your changes



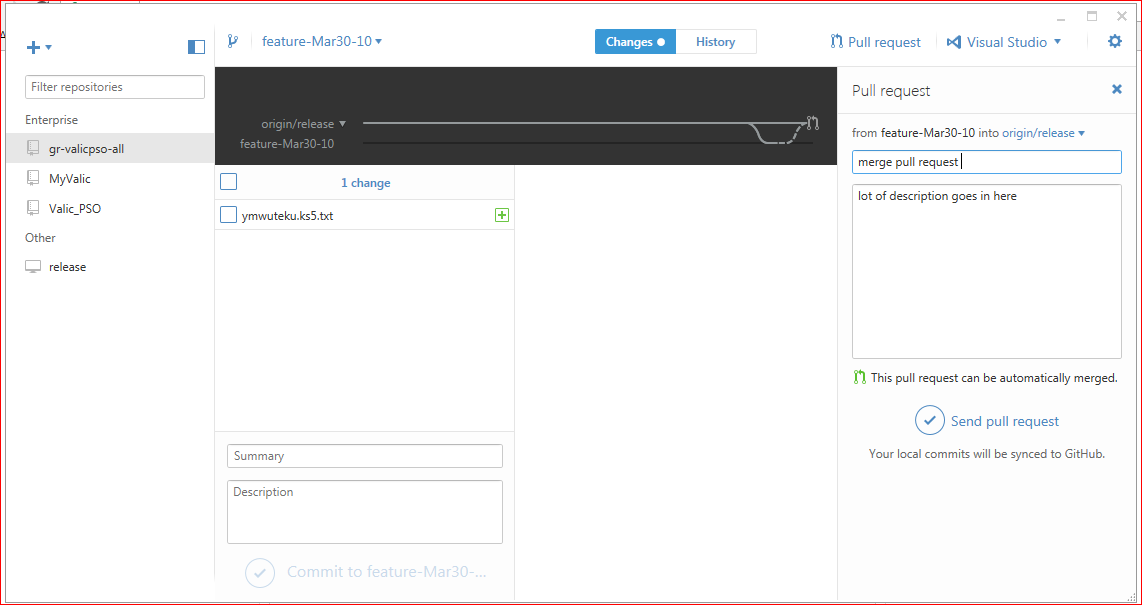
1. Push changes to remote repository



##### Request Merge

Git UI Steps

1. Upon completing committing and pushing your work, create a “Pull Request” to inform the Tech Lead to “pull” the changes that you have done into the Release branch



## Tester – N/A

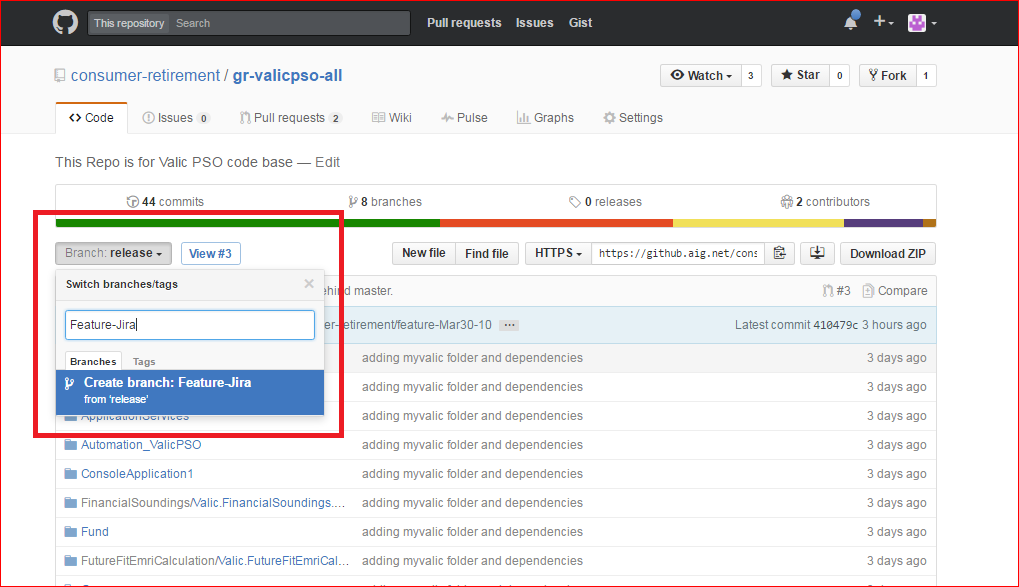
## Tech Lead

### Database Code

#### Create Feature Branch

Git UI Steps

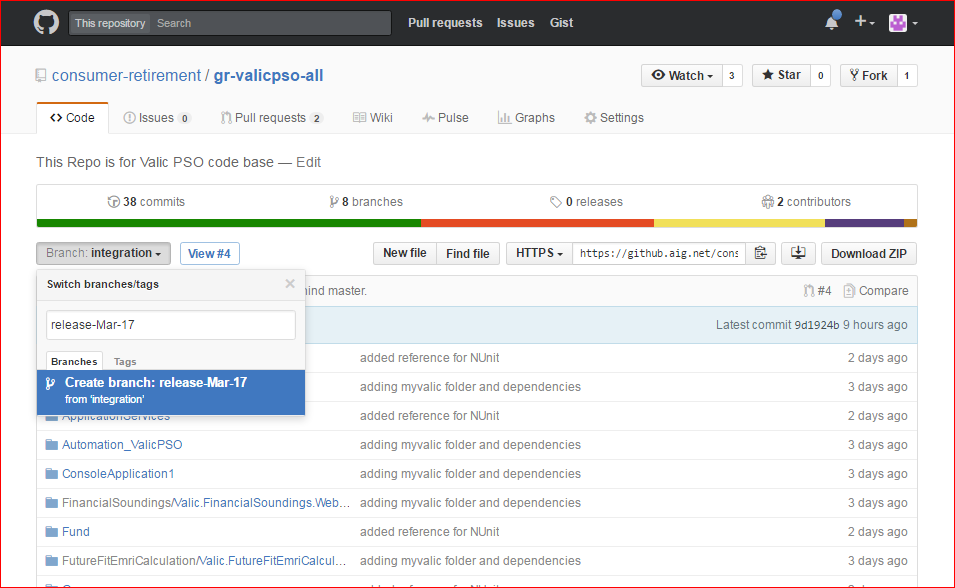
1. Ensure that you are on the correct branch (“release”) from which you want to create the “feature” branch
2. Name the “feature” branch you want to create
3. Click on “Create <feature branch name> from release” message that shows up
   1. **WARNING:** Hitting “Enter” key after typing name does not seem to work in multiple instances



#### Create Release Branch

Git UI Steps

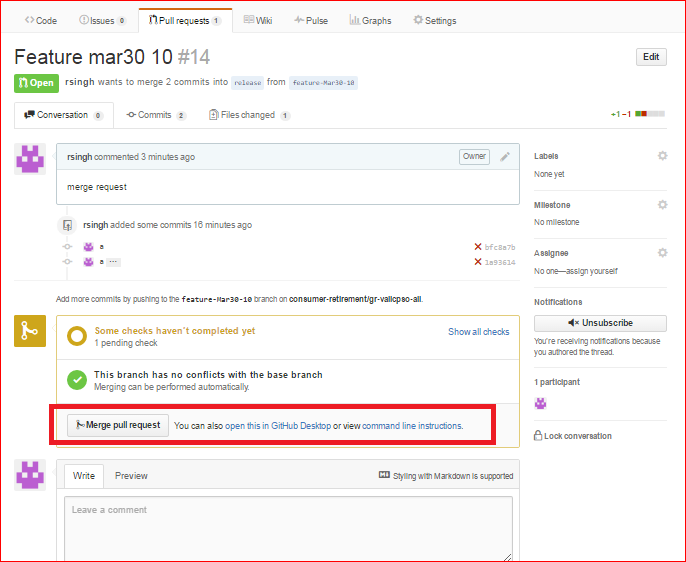
1. Ensure that you are on the correct branch (“integration”) from which you want to create the “release” branch
2. Name the “release” branch you want to create
3. Click on “Create <release branch name> from integration” message that shows up
   1. **WARNING:** Hitting “Enter” key after typing name does not seem to work in multiple instances

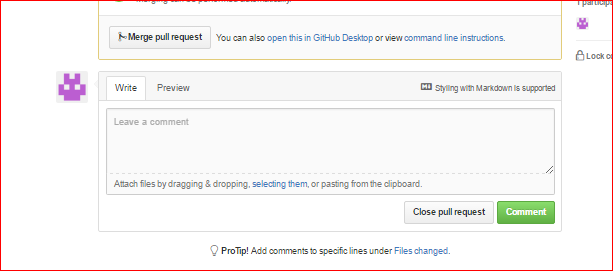


#### Process Pull requests to Release

Git UI Steps

1. Go to <https://github.aig.net/consumer-retirement/gr-valicpsoall>
2. Click on the “Pull requests” tab on top





#### Create Pull Request in INT

Git UI Steps

#### Remove Release branch

Command:

Git UI Steps

#### Remove Feature Branch

Command:

Git UI Steps

### .NET Code

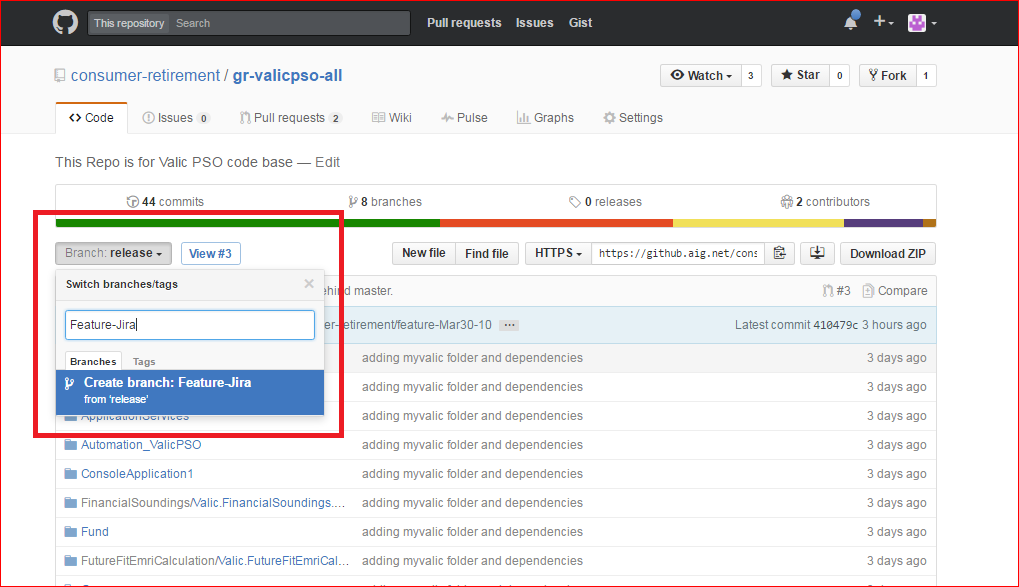
#### Create Feature Branch

Tech Leads are required to have created the “Feature Branch” from which the developers would start cloning their local copies for development

Command:

Git UI Steps

1. Ensure that you are on the correct branch (“release”) from which you want to create the “feature” branch
2. Name the “feature” branch you want to create
3. Click on “Create <feature branch name> from release” message that shows up
   1. **WARNING:** Hitting “Enter” key after typing name does not seem to work in multiple instances

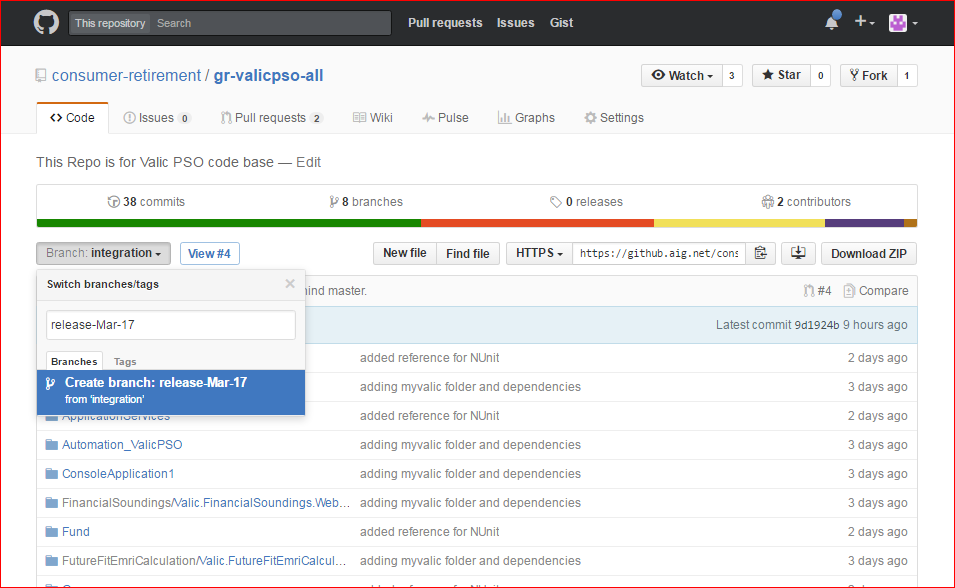


#### Create Release Branch

Command:

Git UI Steps

1. Ensure that you are on the correct branch (“integration”) from which you want to create the “release” branch
2. Name the “release” branch you want to create
3. Click on “Create <feature branch name> from integration” message that shows up
   1. **WARNING:** Hitting “Enter” key after typing name does not seem to work in multiple instances

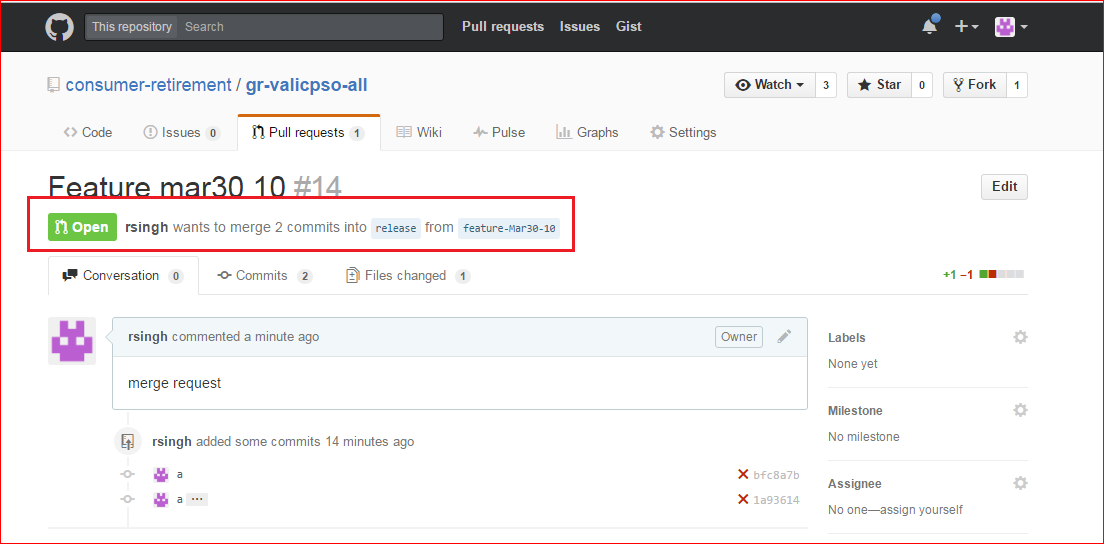


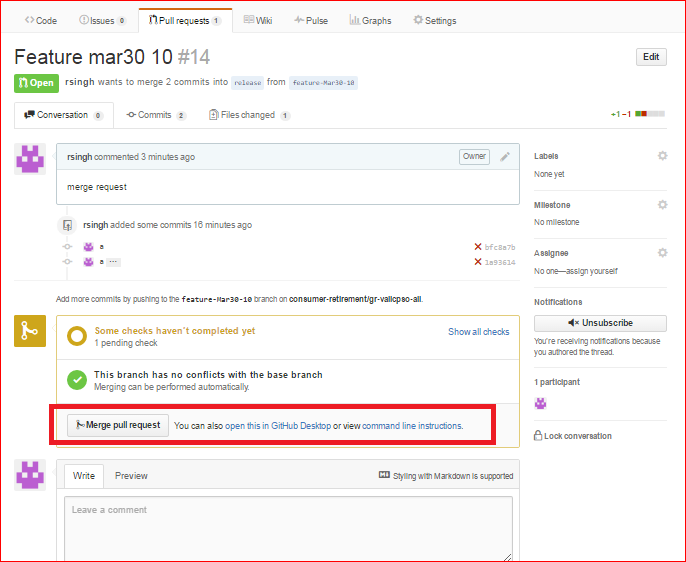
#### Process Pull requests

Command:

Git UI Steps

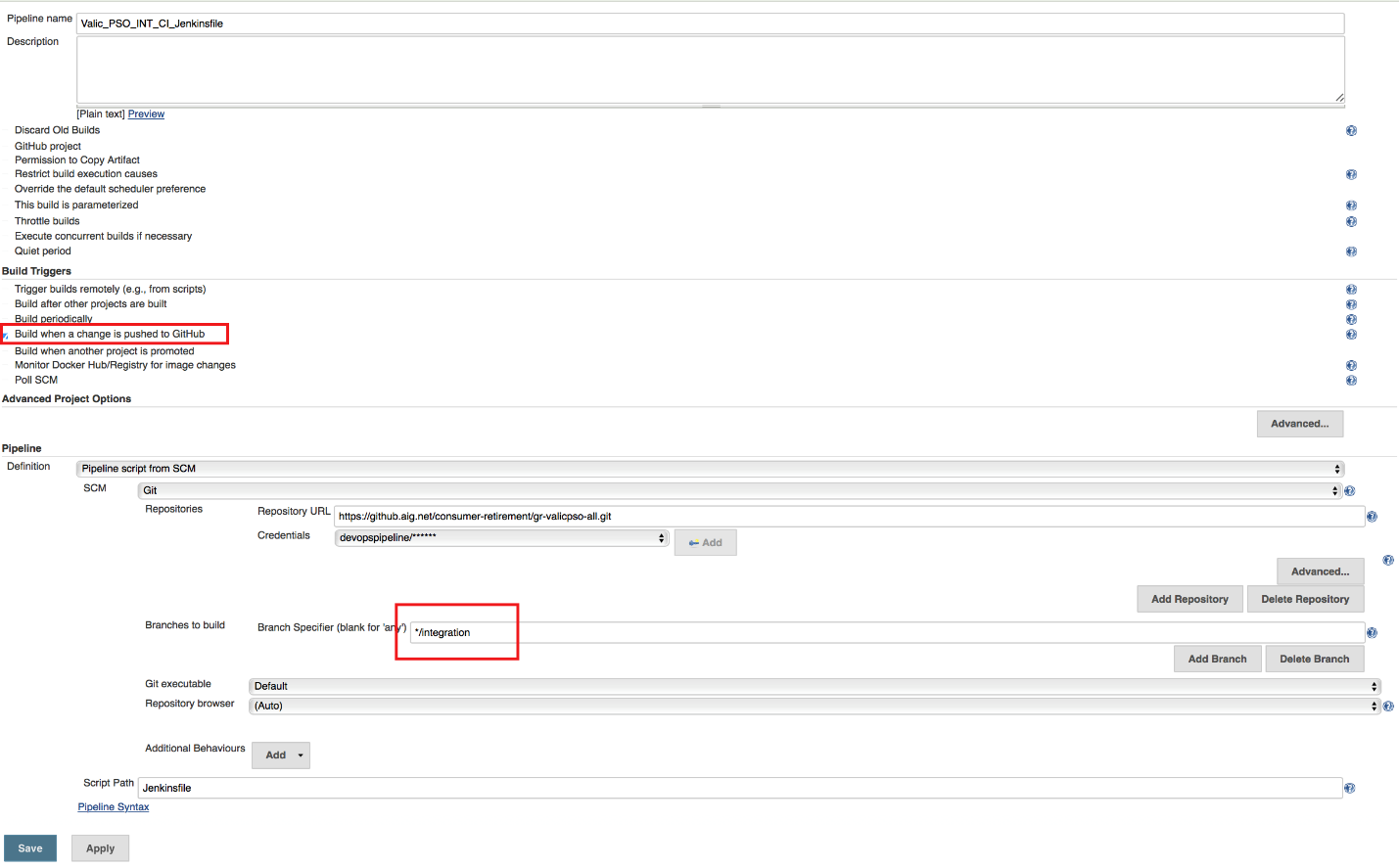
1. Go to <https://github.aig.net/consumer-retirement/gr-valicpsoall>
2. View pull requests and process them





###### Build

Build is configured to be kicked-off automatically upon processing pull request and code is merged.



#### Create Pull Request in INT

Command:

Git UI Steps

#### Remove Feature Branch

Command:

Git UI Steps

## CM/RM

### Branch Set-up

Naming Conventions

Git

Git-TFS

Branch level properties

#### Jenkins Set-up for branch

Create Jenkinsfile

### Database Code

#### Process Pull requests to QA

Command:

Git UI Steps

### .NET Code

#### Process Pull requests to INT

Command:

Git UI Steps

## DBA

### Database code

#### Process Pull requests to INT

Command:

Git UI Steps

#### Create Pull Request in QA

Command:

Git UI Steps

# Conflict Resolutions

## Administrative

### GITHUB

#### Tech Contact

#### Admin Contact

### Jenkins

#### Tech Contact

#### Admin Contact

### JIRA

#### Tech Contact

#### Admin Contact

### XLR

#### Tech Contact

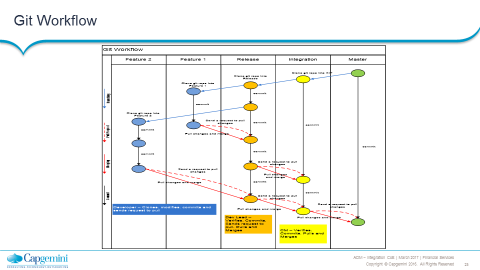
#### Admin Contact

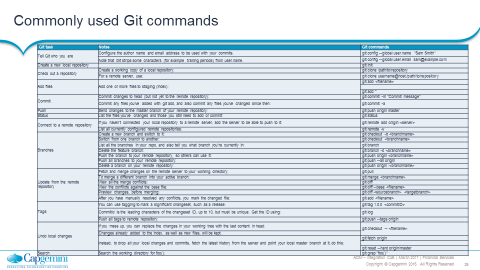
### XLD

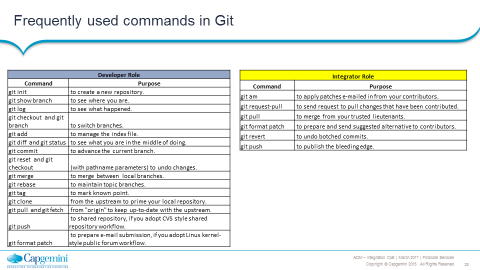
#### Tech Contact

#### Admin Contact

## FAQ





Reference

