Contents

[1. Bitbucket setup and client workflow 2](#_Toc453170503)

[a. Workflow process 2](#_Toc453170504)

[2. Git Client 2](#_Toc453170505)

[a. Setup 2](#_Toc453170506)

[b. Git client usage 5](#_Toc453170507)

[3. Sourcetree 8](#_Toc453170508)

[a. Setup 8](#_Toc453170509)

[b. Client usage 9](#_Toc453170510)

[4. Eclipse Client 14](#_Toc453170511)

[a. Setup 14](#_Toc453170512)

[b. Client Usage 19](#_Toc453170513)

[c. Pulling from remote repository 26](#_Toc453170514)

[d. Pushing your changes to remote repository 29](#_Toc453170515)

[5. Server Pull request workflow 31](#_Toc453170516)

[6. Appendix 35](#_Toc453170517)

[a. Access on BitBucket Server 35](#_Toc453170518)

[b. Install GIT client 36](#_Toc453170519)

[c. Install Source Tree GIT Client. 37](#_Toc453170520)

# Bitbucket setup and client workflow

1. Take access on the bit bucket server as a user. The cmp process is mentioned in the Appendix 1
2. Install once of the clients on your machine: Git Client, SourceTree, Eclipse or Sublime. The installation and usage steps are mentioned in different sections in the document.
3. Make changes in the local repository and merge with the central repository based on the below Git Workflow.
4. Once the changes are pushed to the central repository, login to the server and create a pull request. The pull request is detailed in separate section

Bitbucket server: <https://cedt-icg-bitbucket.nam.nsroot.net/bitbucket/projects>

1. Merge the changes once the pull request is approved.

## Workflow process



# Git Client

## Setup

1. Raise cmp for installation of Git client. The cmp details are mentioned in Appendix 2
2. Setup name and email id

Open Git Bash which will open a command line and enter the below 2 commands for configuring your Name & E-mail ID details in GIT Config file.

git config --global user.name “<soeid>"

git config --global user.email <email id>

1. SSH key generation

You can use SSH keys to establish a secure connection between your computer and Stash when you are performing Git operations. The option to clone using SSH only becomes available once you've added an SSH key to your user profile in Stash. Supported key types are DSA and RSA2.

Check if a key file already exists in C:/Users/[Your SOEID]/.ssh/id\_rsa.pub

If it exists, reuse the same key, else, generate a new one using the below commands

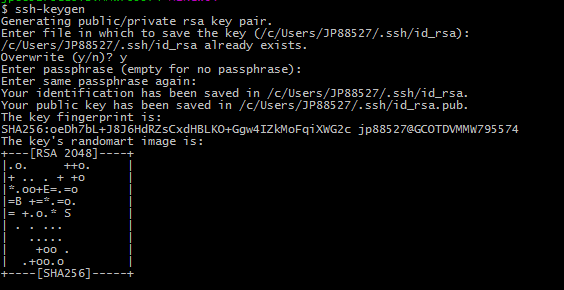
1. Open **Git Bash** and run the following command:-

ssh-keygen

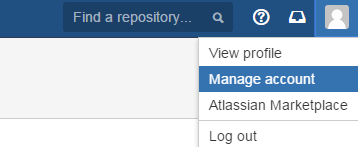
You shall see the following response:-



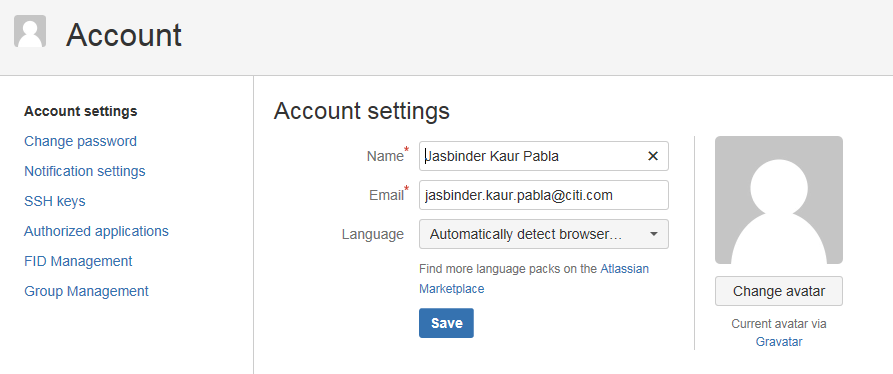
1. Press <Enter> to accept the default location and file name. If the .ssh directory doesn't exist, the system creates one for you.
2. Enter, and re-enter, a passphrase when prompted [ Passphrase is an optional value which acts as another layer of credentials while you connect to the Git repository]. Please check the link <https://help.github.com/articles/working-with-ssh-key-passphrases/> which discusses about passphrases.

The whole interaction will look similar to this:-

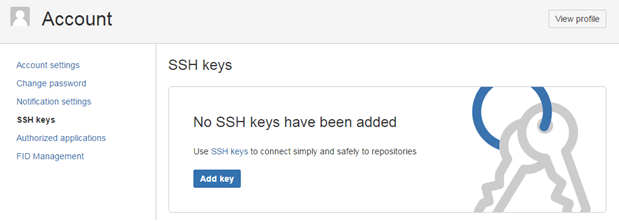
1. You can find your generated key here: **C:/Users/[Your SOEID]/.ssh/id\_rsa.pub**
2. Enter **C:/Users/[Your SOEID]/.ssh/id\_rsa.pub** to display the public key content.
3. Copy contents of Public key
4. Log in to Bitbucket
   * Click on "Manage account" from top right corner of GUI)



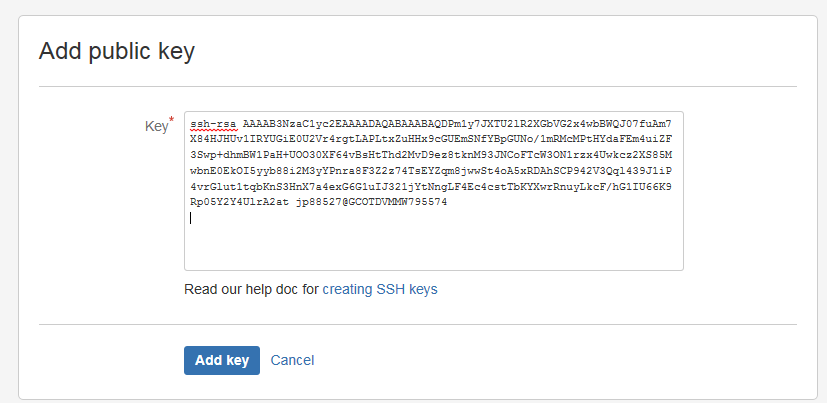
* + Click on "SSH Keys"



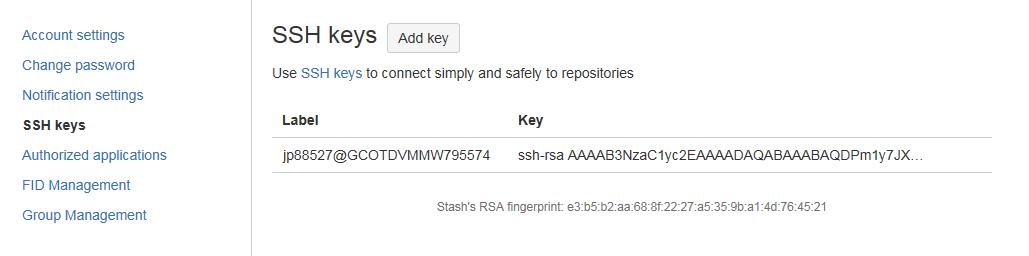
* + Click on "Add Key"



* + Paste the complete SSH keys generated from the Step above:-



* + Click on "Add key", You can see your key is now configured to use (Label - SOE ID@VDI machine name)



## Git client usage

1. Clone remote repository to get a local working copy
2. Open Git CMD or Bash -> Go to the directory on disk where you want to clone remote git repository
3. Go to your repository in bitbucket server -> Select Clone from sidebar-> Copy Clone url -> Execute – “git clone <remote\_repository\_url>”
4. Checkout the branch from develop branch for making changes

Execute – Git checkout –b < branch name> develop

1. Modify a file in the working directory
2. Move to the cloned repository folder. Check current status by executing “git status”. Presently there are no modified files.
3. Modify some files & check git status. It should show modifies files.
4. Add these files to the staging area
5. Execute “git add .” to add all modified files to staging area or “git add <filename.txt>” to add one specific file or “git add \*.txt” to add only particular type of files.
6. Perform commit operation that moves the files from the staging area and commits changes to the local repository
7. Run “git commit –m “<message>” ” to commit changes locally. Only staged changes (staged in previous step) will be committed.
8. Push changes committed in local repository to remote, it stores the changes permanently to the remote Git repository
9. Run “git push origin <remote\_destination\_branch \_name>”. You can now browse through your changes on bitbucket server.

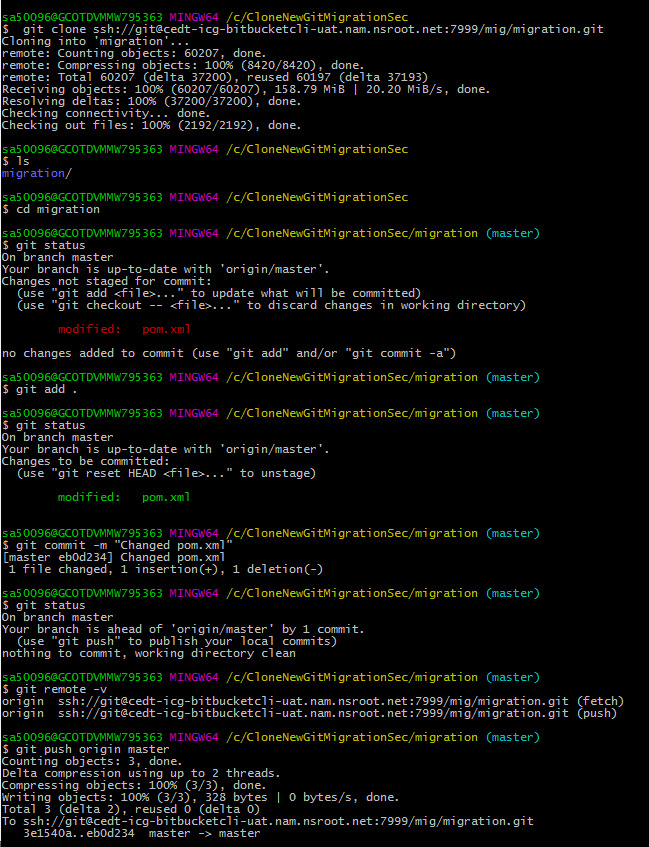
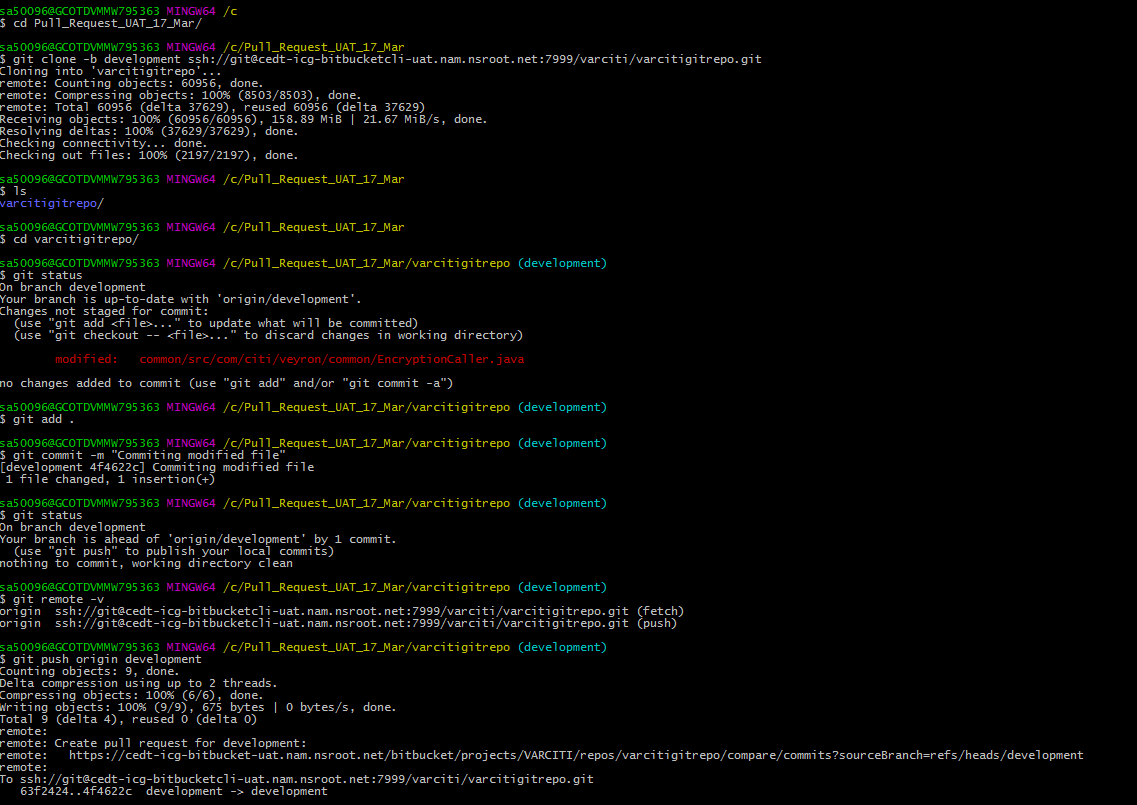
Fig : Cloning remote repositories master branch, modifying & pushing back to master branch

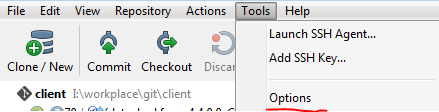
Fig: Cloning development branch, modifying & pushing back to development branch

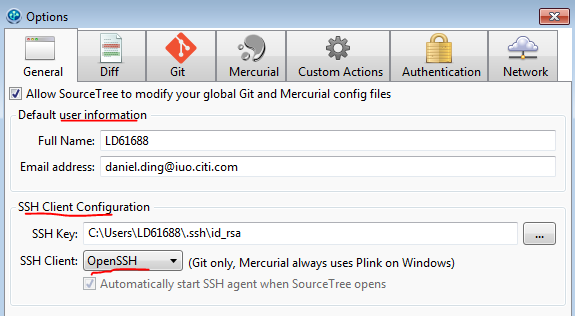


# Sourcetree

## Setup

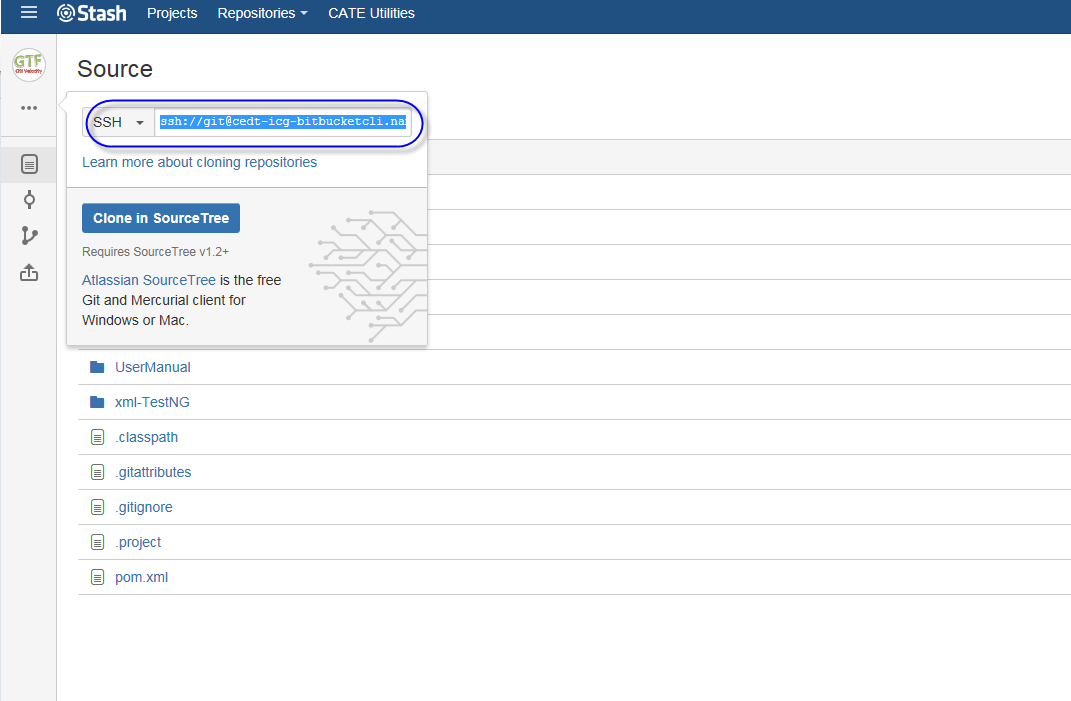
1. Raise cmp for installation of Git client. The cmp details are mentioned in Appendix 3
2. Configure SSH keys



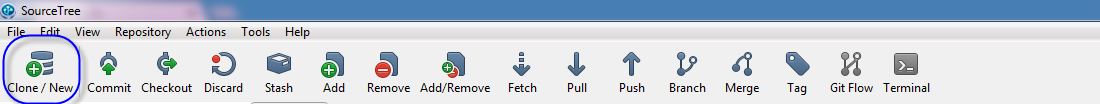


## Client usage

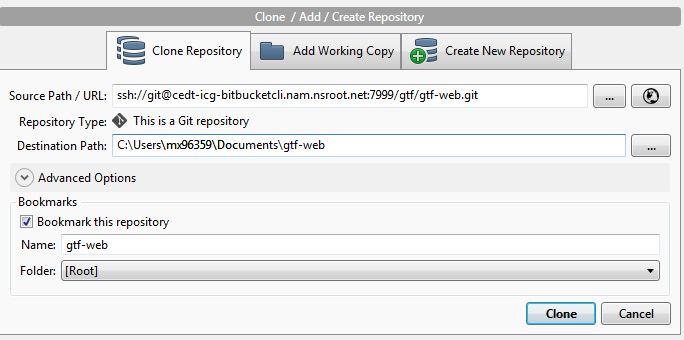
1. Clone the remote repository



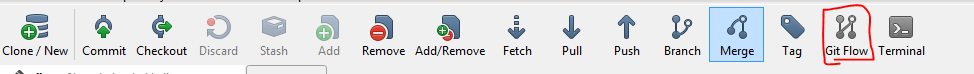
In SourceTree, click “Clone/New” button.

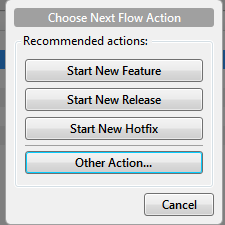


Clone Repository – Paste source URL into it and define the destination path, click “Clone” button.

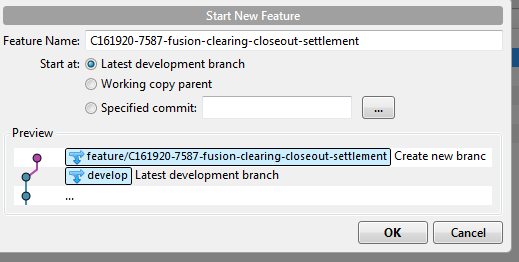


1. Create a new branch in local repository





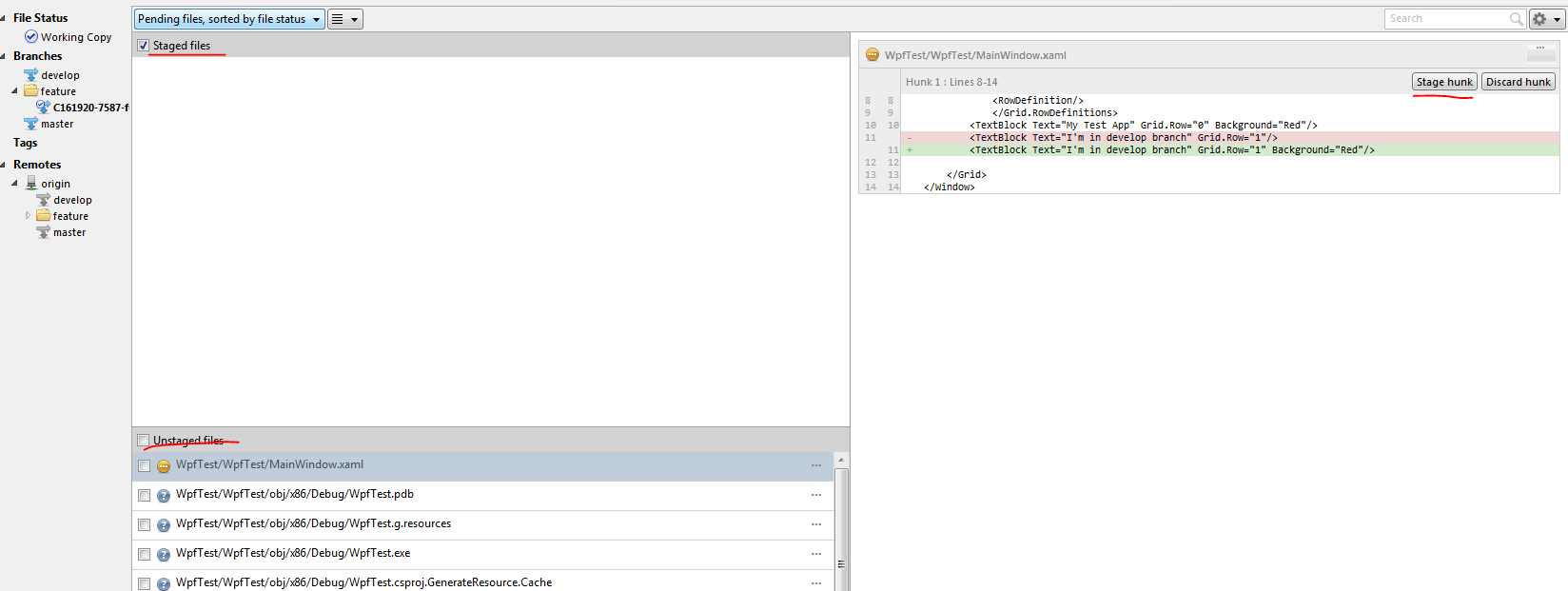
Click on ‘Start New Feature’



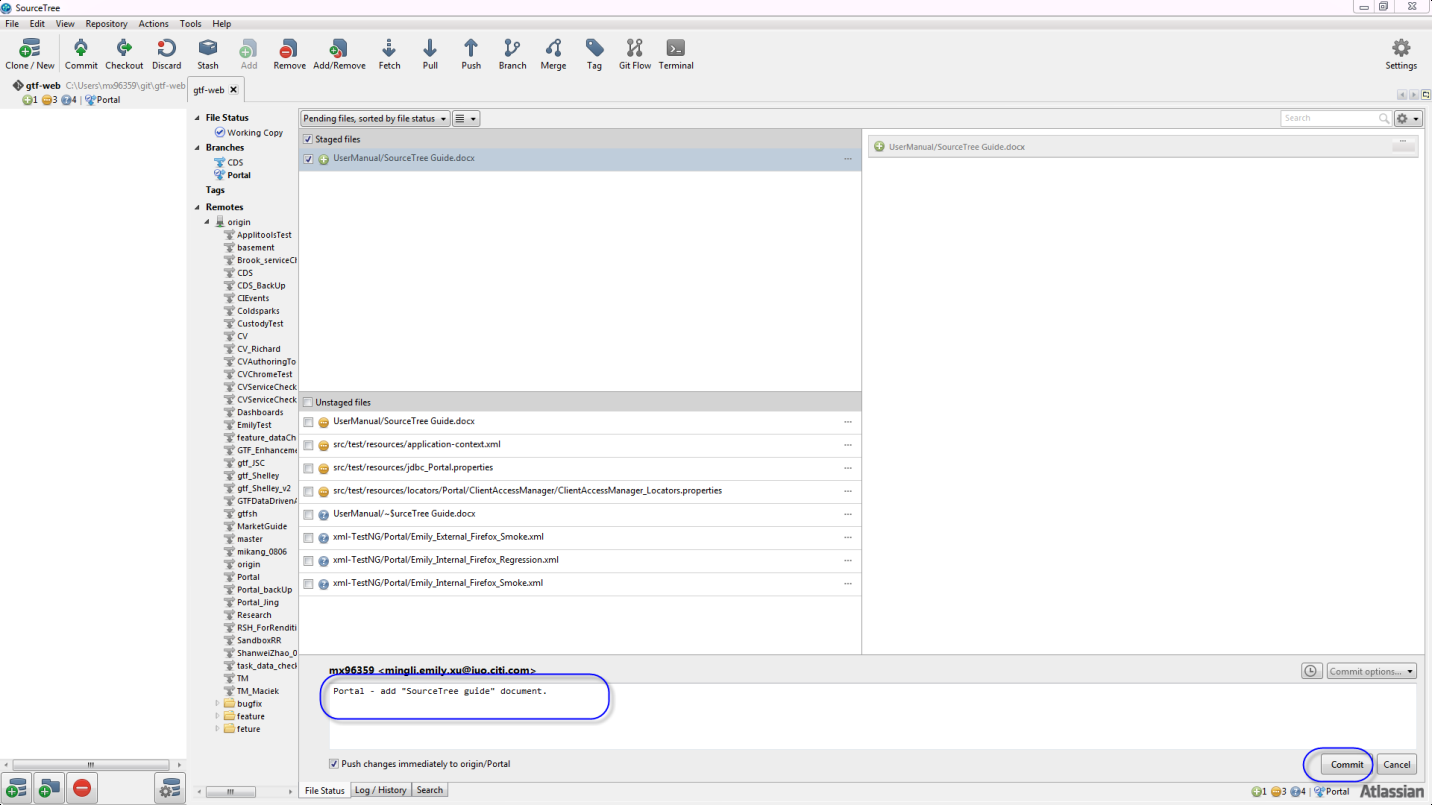
This will automatically check out the feature branch

1. Modify files and commit changes to the new branch in local repository

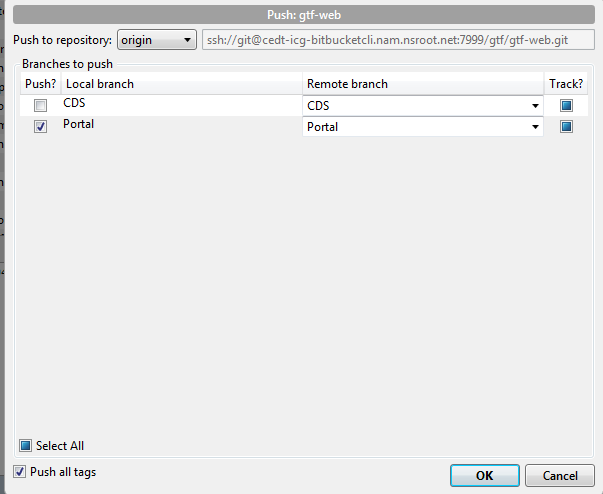
* Modify files in your working directory
* Stage the files, adding snapshots of them to your staging area
* Commit the changes which takes the files as they are in the staging area and store that snapshot permanently to local Git repository
* In order to commit changes into local git directory first move the file from Unstaged files to Staged files by checking the checkbox at the beginning



* Write commit comments in correct format and then click Commit button.



* Push commit to the remote branch.

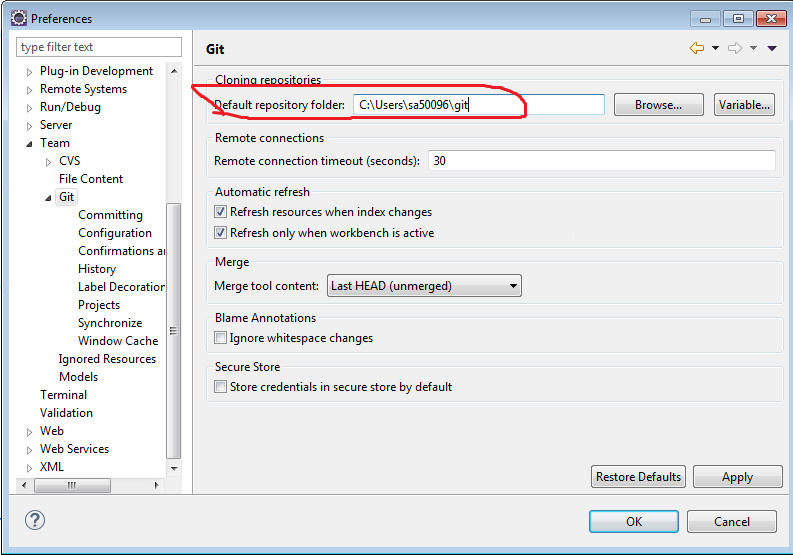


* On the server, create pull request for the branch

# Eclipse Client

## Setup

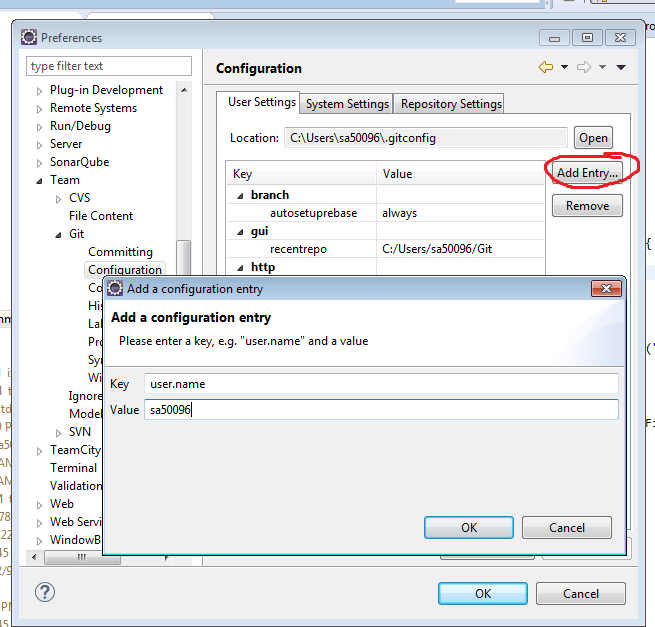
* 1. Go to Windows -> Preferences -> Team -> Git -> Change the default repository folder if required (This is where your cloned repositories sub-folders would be created)



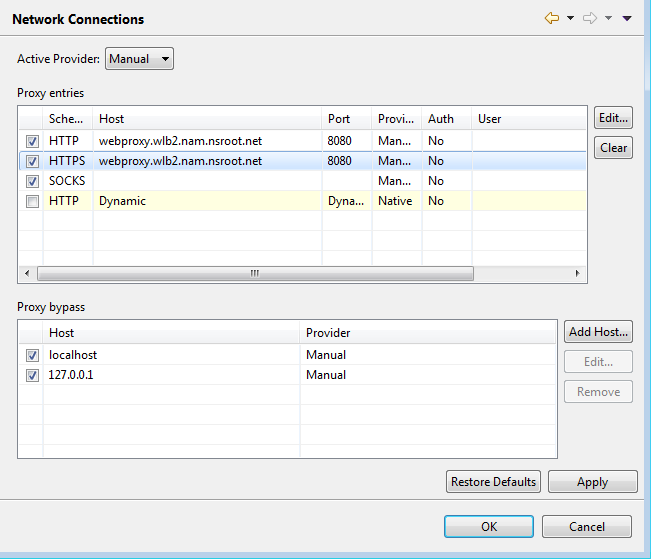
* 1. Setup user id . Go to Configuration -> User Settings Tab -> Add Entry

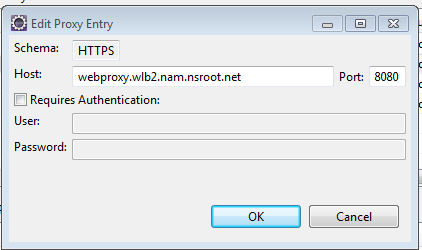
Key – user.name, value – [YOUR SOEID]

Key - user.email, value – [YOUR CITI EMAIL ID]



Check if existing network settings is as below



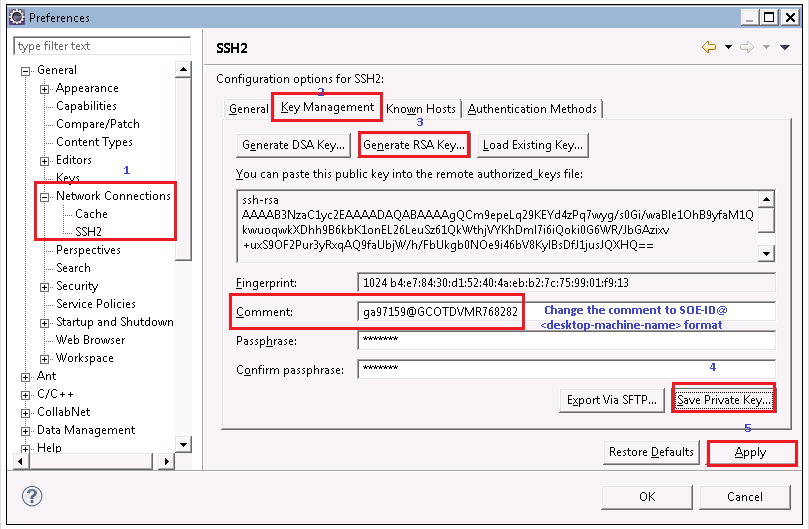


* 1. Setup SSH Key

Check if a key file already exists in C:/Users/[Your SOEID]/.ssh/id\_rsa.pub

If it exists, reuse the same key, else, generate a new one using the below process

* Go to Windows -> Preferences -> General -> Network Connections -> SSH2-> Key Managemnet
* Change the comment field to <your soeid>@<desktop-machine-name>
* Save the Private Key
* Eclipse restart may be required once this step is completed

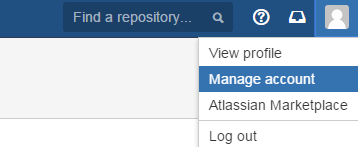


Add key to client using the below steps

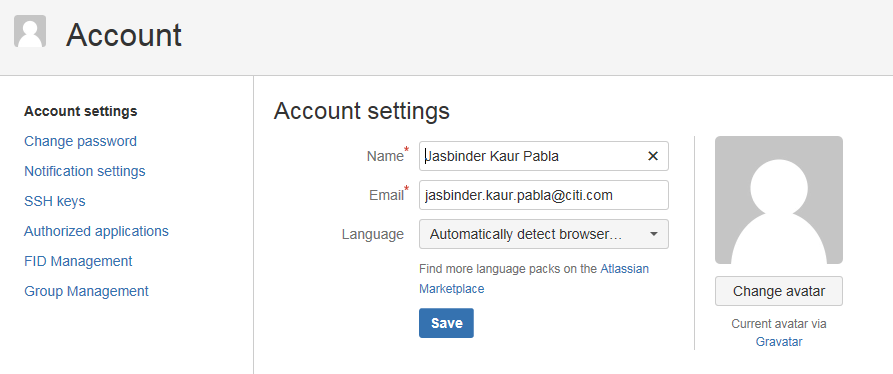
* Go to Windows -> Preferences -> General -> Network Connections -> SSH2
* In the General Tab, SSH2 home should point to “C:\Users\[YOUR SOEID]\.ssh” and Private Keys should point to “id\_dsa,id\_rsa”
* If kys already exist, in the Key Management Tab, select Load Existing Key -> Point to your id\_rsa file in .ssh folder -> Open -> (Leave passphrase as empty) Apply -> OK.
* Restart Eclipse for the changes to take effect.

Add the keys generated above to Bitbucket server .

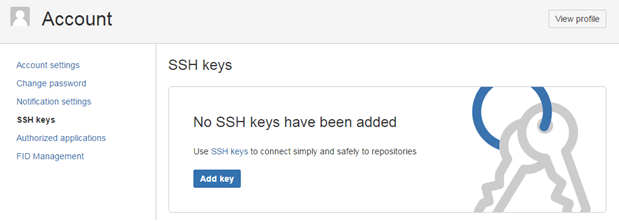
* Log in to Bitbucket
  + Click on "Manage account" from top right corner of GUI)



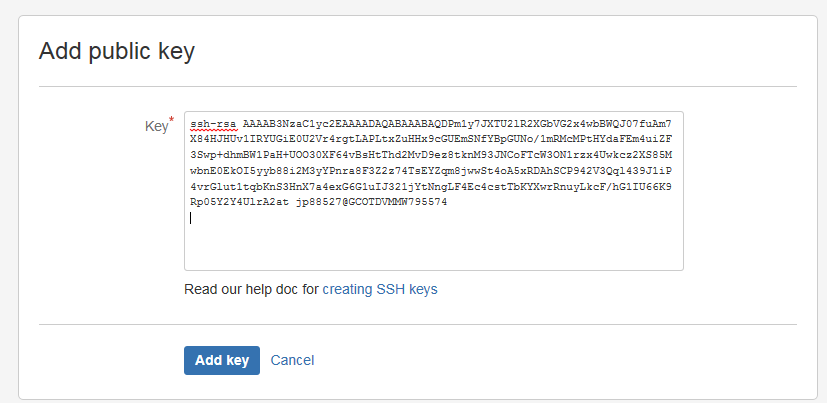
* + Click on "SSH Keys"



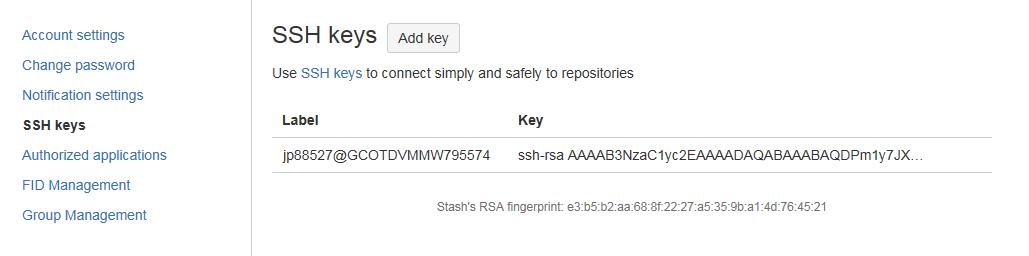
* + Click on "Add Key"



* + Paste the complete SSH keys generated from the Step above:-



* + Click on "Add key", You can see your key is now configured to use (Label - SOE ID@VDI machine name)

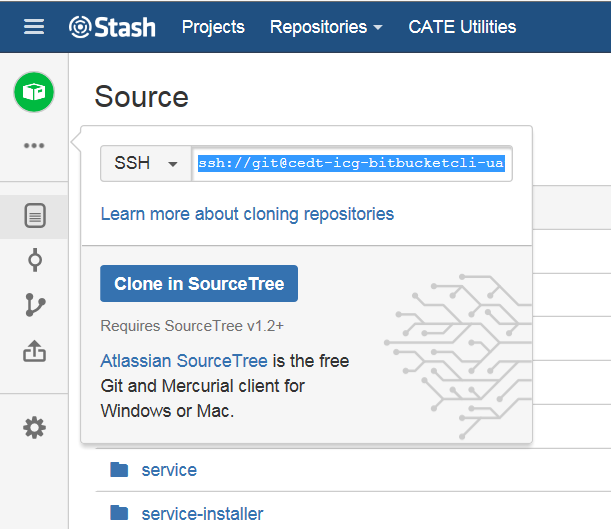


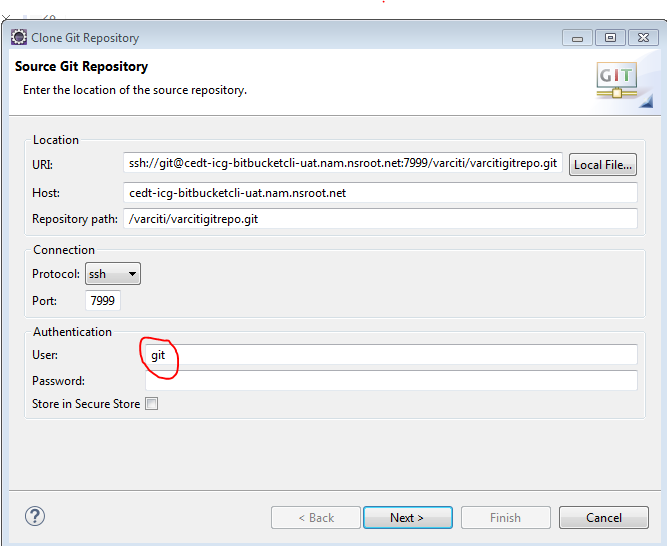
## Client Usage

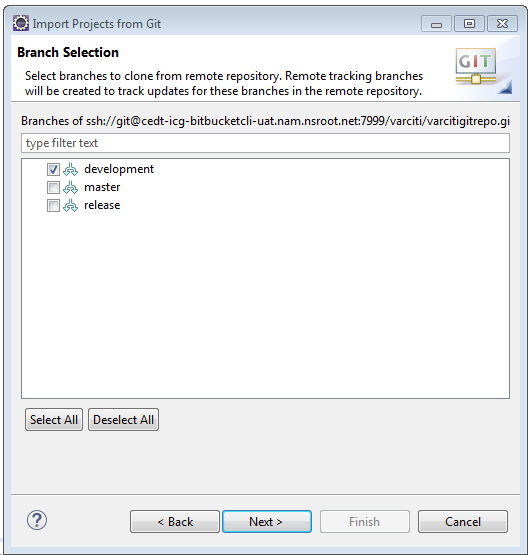
* 1. Clone the remote repository

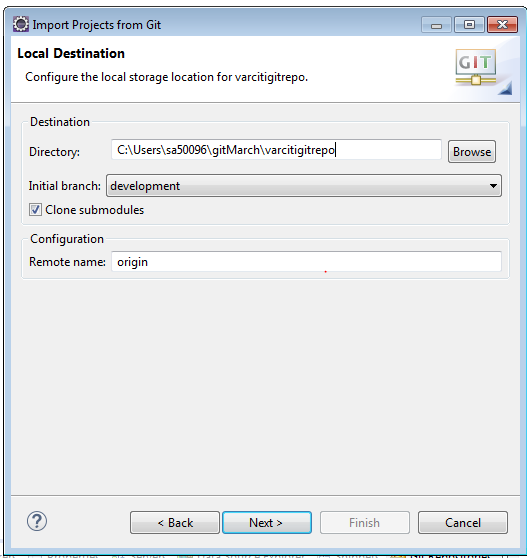
In your Eclipse workspace go to File -> Import -> Git -> Clone URI -> Press Next -> Enter the repository url (Go to your repository in bitbucket server -> select clone & copy the url)

Note: User is “git” when cloning via ssh

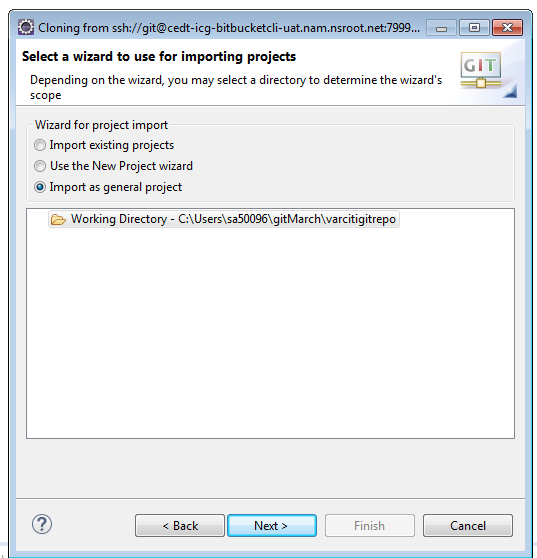




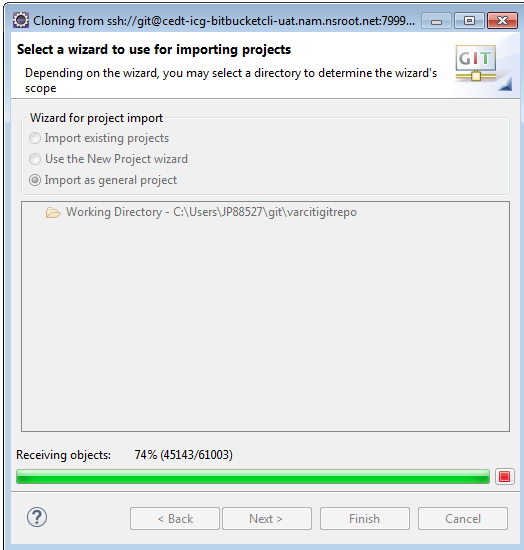
* Click Next -> Select Branches you want to clone (Developers would want to clone just the “development branch”)
* Click Next -> Make sure you check Clone submodules check box



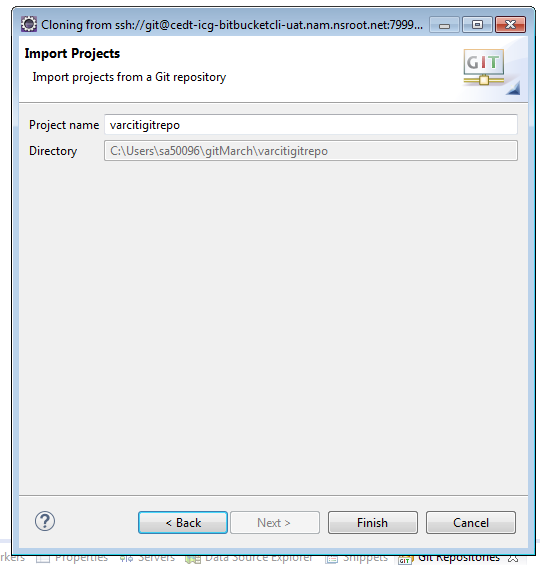
* Click Next -> Select Import as a general project



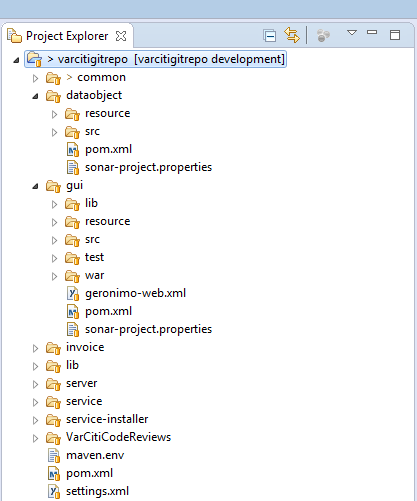
* Click Next



* Click Finish

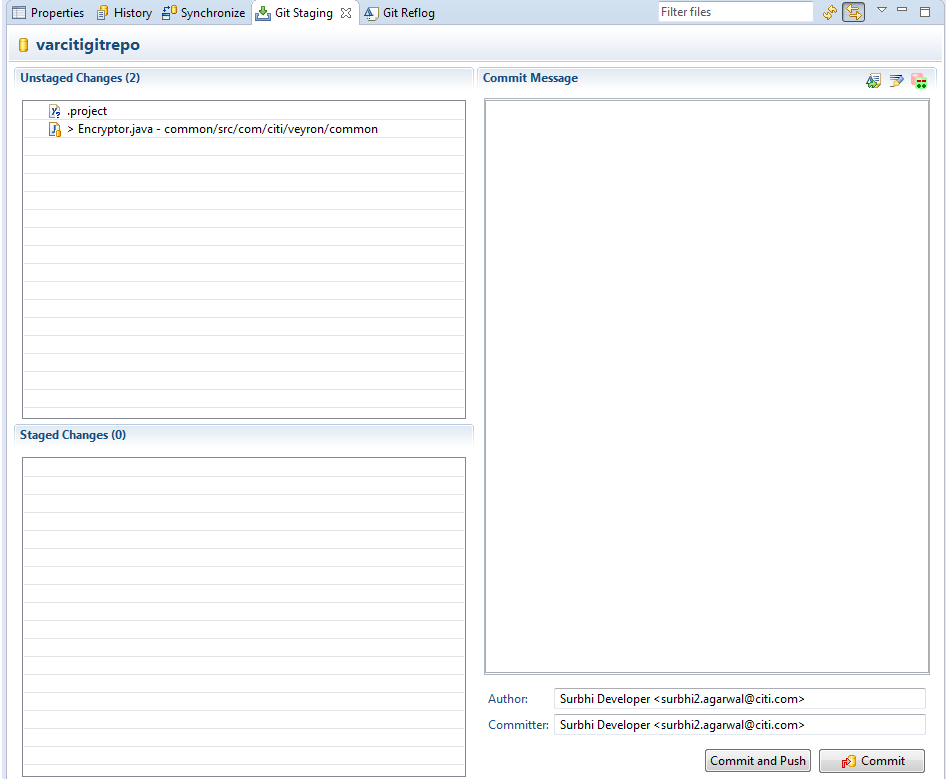


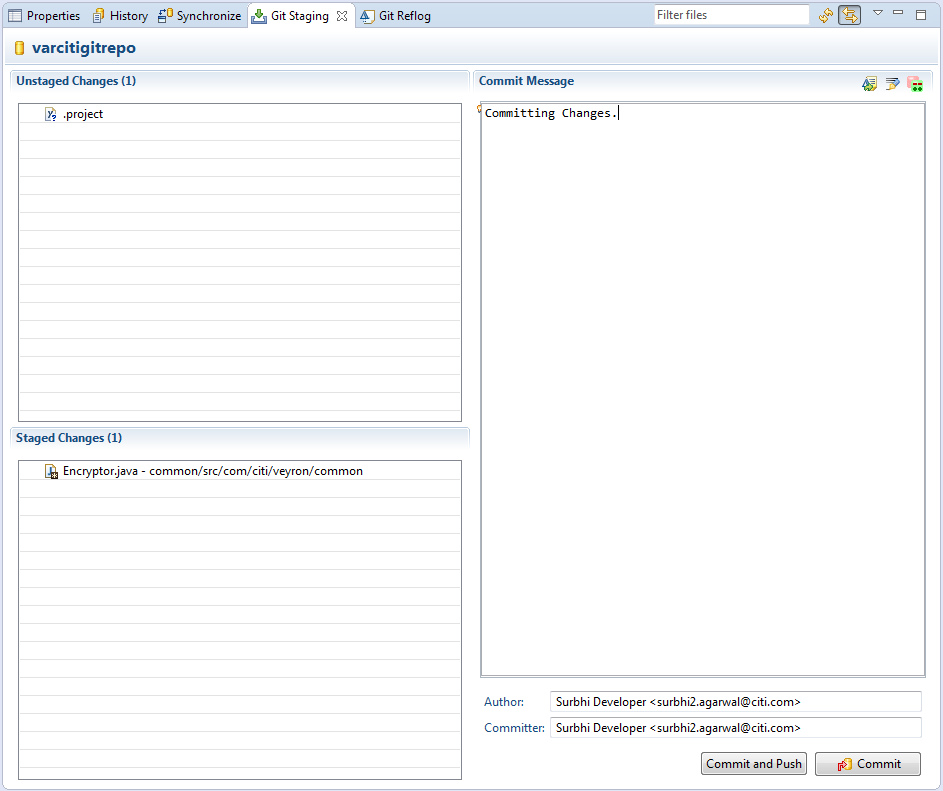
* Project Explorer should look like the following:



* 1. Make changes to the branch
* Go to Git Perspective (Window -> Open Perspective -> Git) -> Git Staging area
* Drag files from Unstaged Changes area to Staged Changes area, give a commit message and hit Commit to commit locally.

Note: Hitting Commit and push will open push dialog to allow pushing the changes to remote repository. Ideally you should first do commits locally then later push all commits to the remote repo as explained below. So for now just commit.



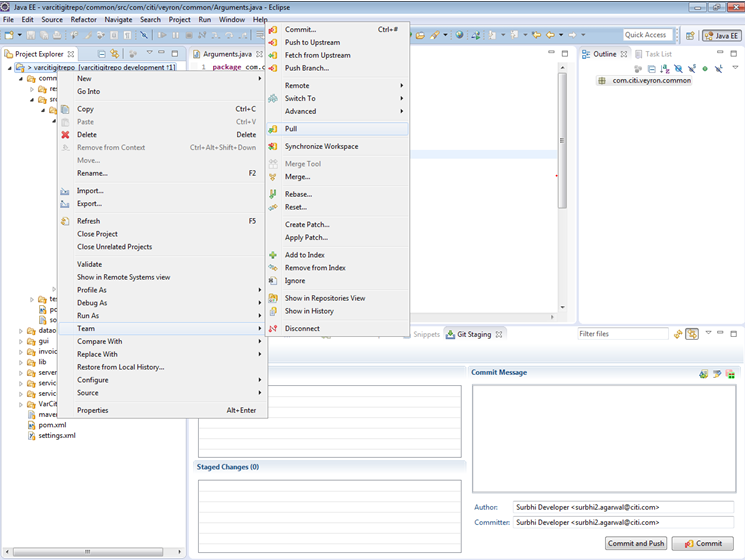


* Before pushing your changes to repository ensure your local repository is in sync with remote repo. Follow steps below to first sync your local branch to remote development branch and then push to it.

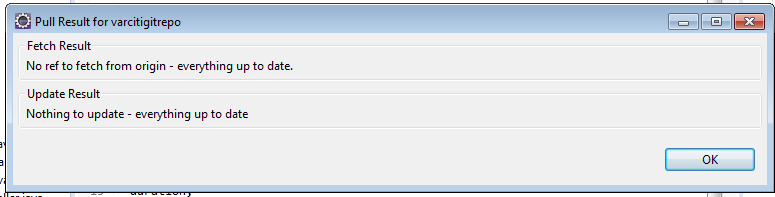
## Pulling from remote repository

Note: You should have committed your changes to local repository before pulling in new updates from remote (Based on the workflow suggested in above pdf attachment)

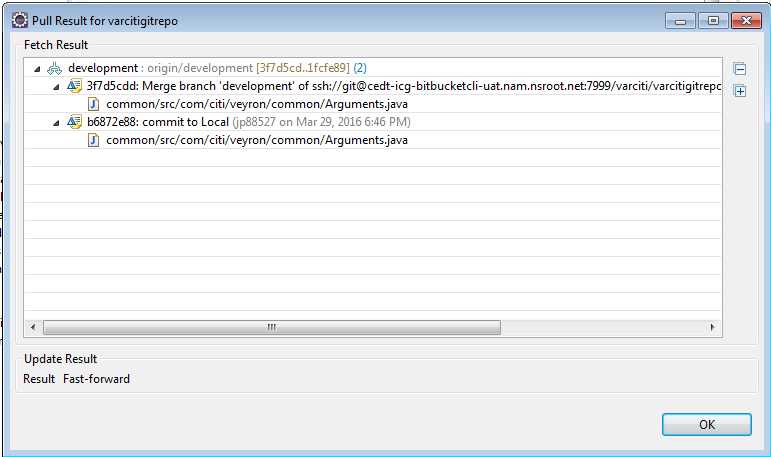
* Pull from remote repository: Go to root folder in your package explorer & right click -> Team -> Pull



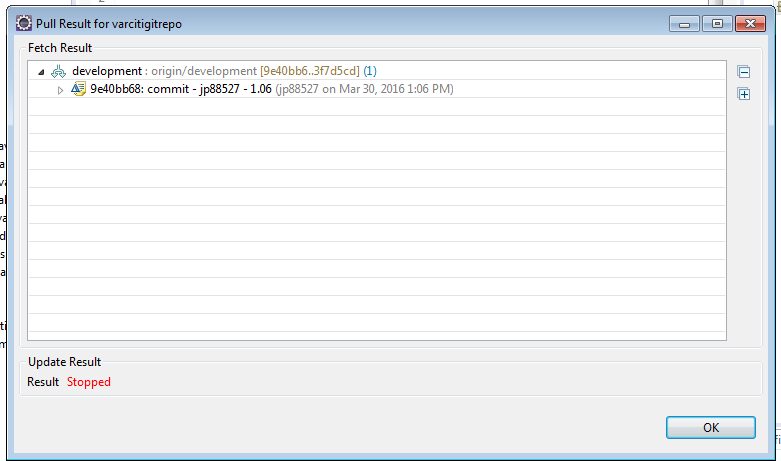
* If you do not have any updates to pull you would see something like :



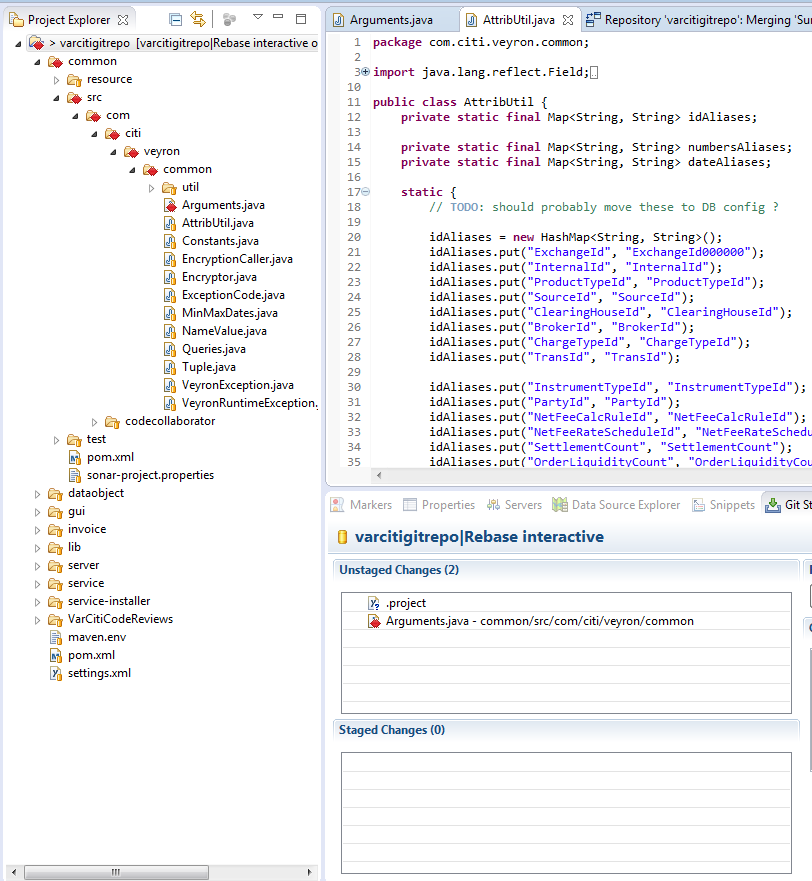
* If in case you have some updates to pull you would see something like:



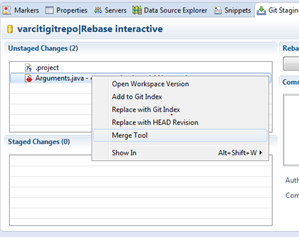
* Click OK. If you have no conflicting changes the changes would be pulled (fetched + merged). No further action required
* If you have any conflicts:

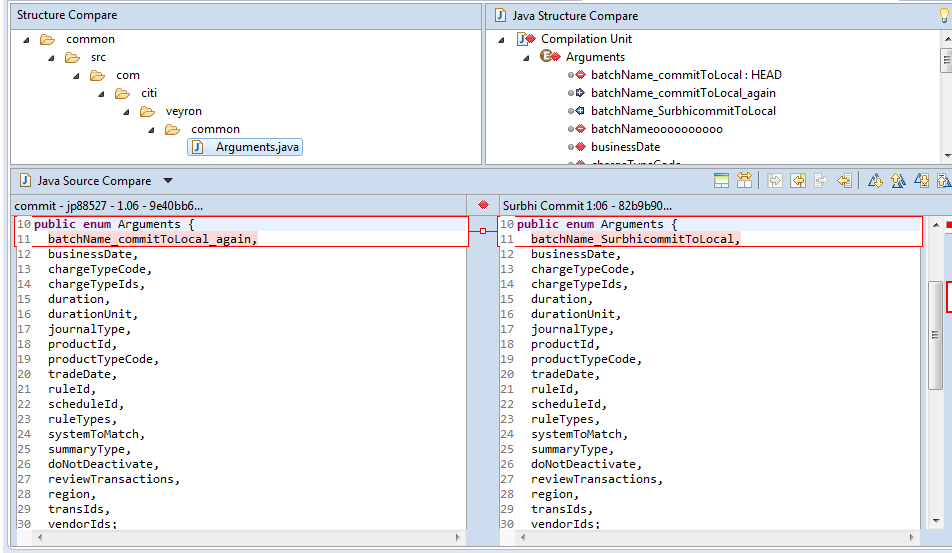


* Click ok. Your Project Explorer will show the conflict as below and in Git Staging area also you’ll see the conflict highlighted.



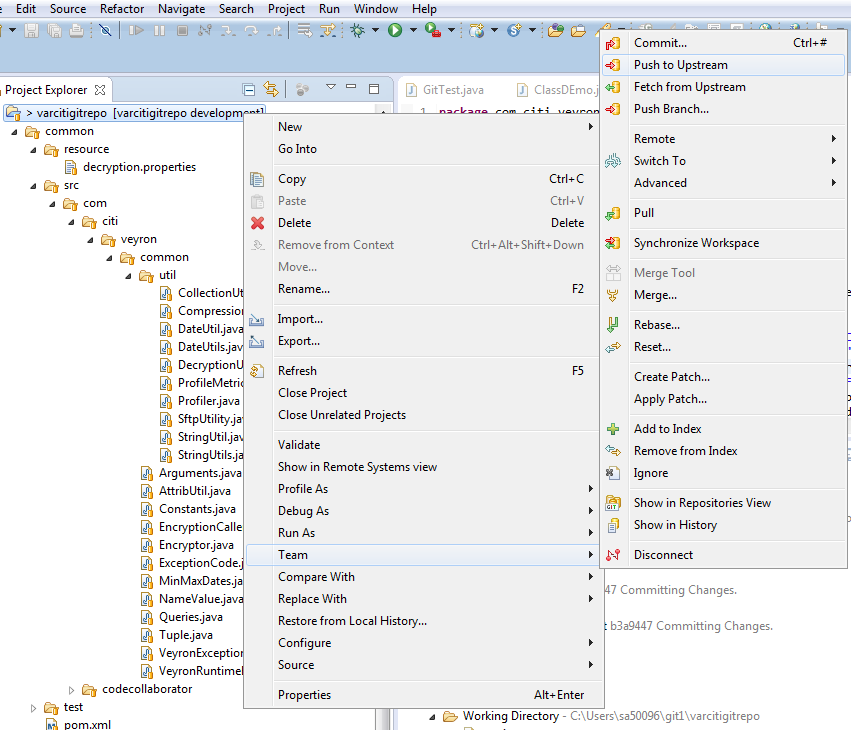
* To resolve the conflict, In your staging area right click the file with conflict -> Select Merge tool

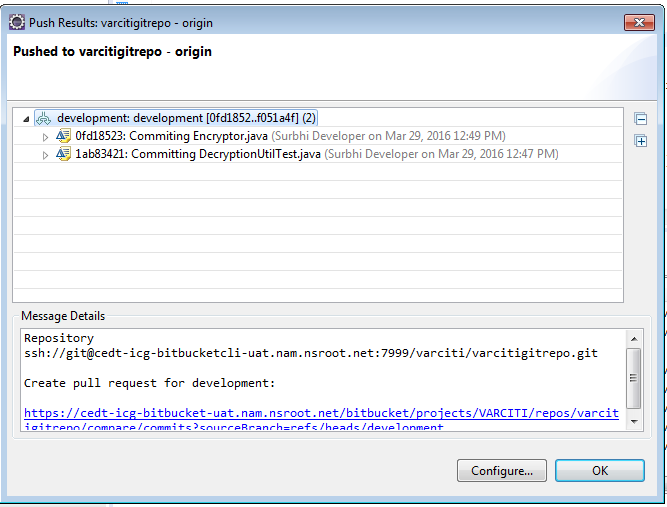


* You would see the conflicting changes in the compare editor of Merge Tool
* After Comparing you could take either of the two paths below
  + Accepting the changes the pulled file from remote repository suggests
    - After accepting the changes -> Stage the file -> Commit.
  + Disregarding those changes and keeping your changes
    - Just stage ->Hit “Commit & Push”

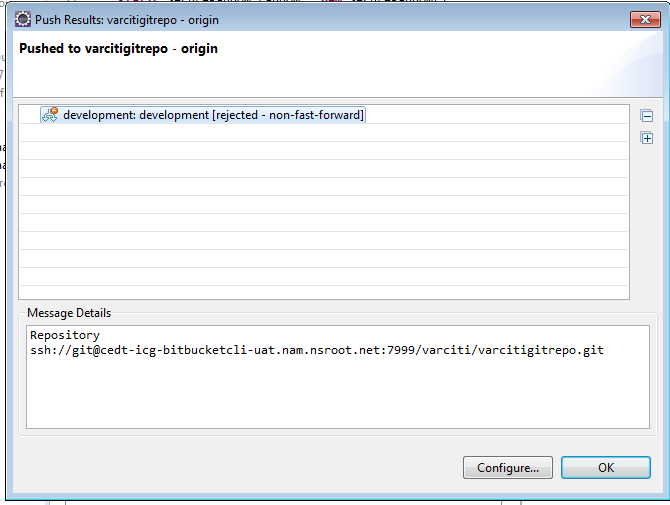
## Pushing your changes to remote repository

* + Go to the folder containing all your projects in explorer & right click, select Team -> Push to Upstream





* Select ok. This pushes to the same remote branch which was cloned (“development” in this example)
* If in case you haven’t followed above steps of pulling from remote repository and there are changes to be pulled then Egit won’t let you push. You should get a message like below. In such a case follow pulling steps above

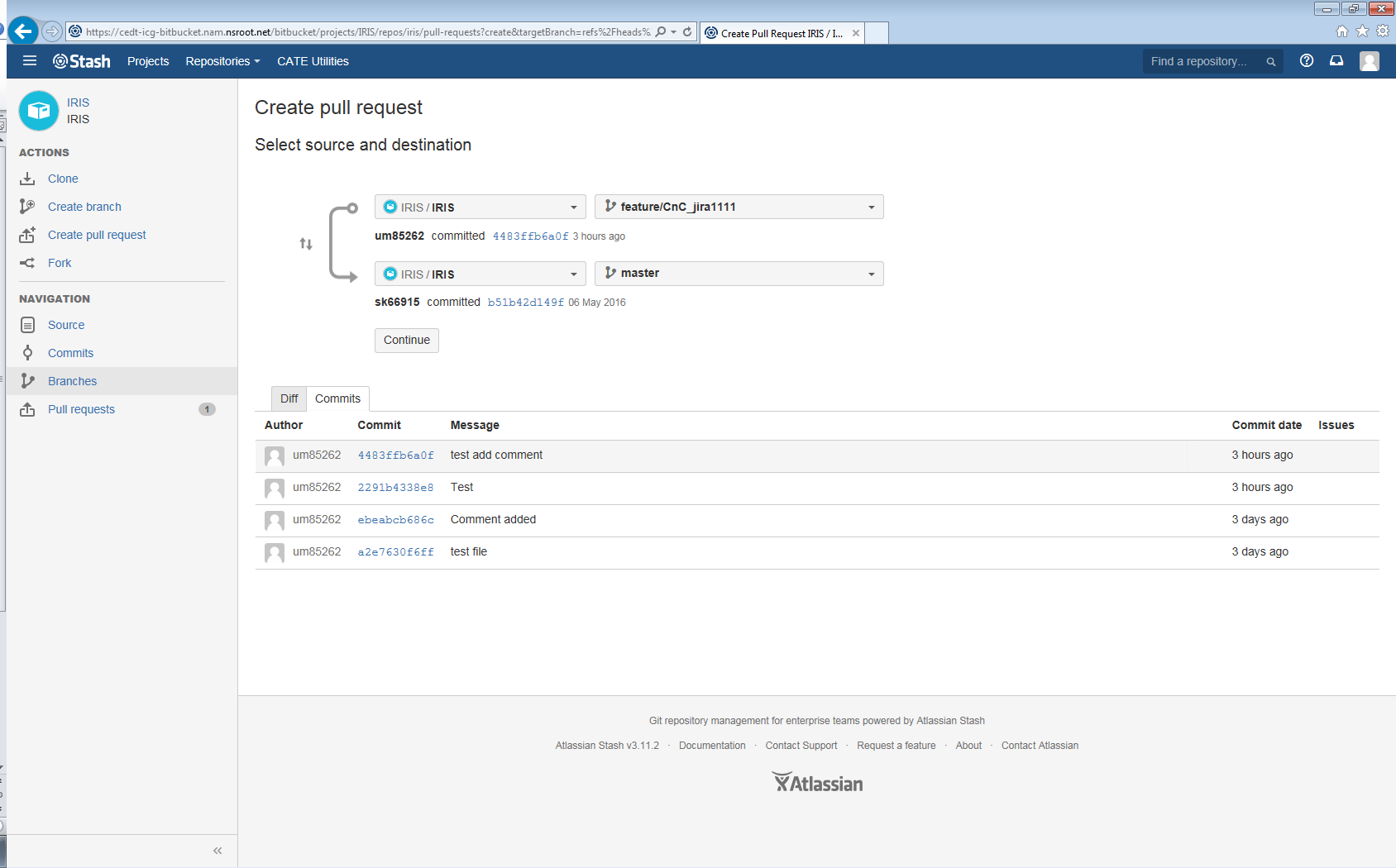


# Server Pull request workflow

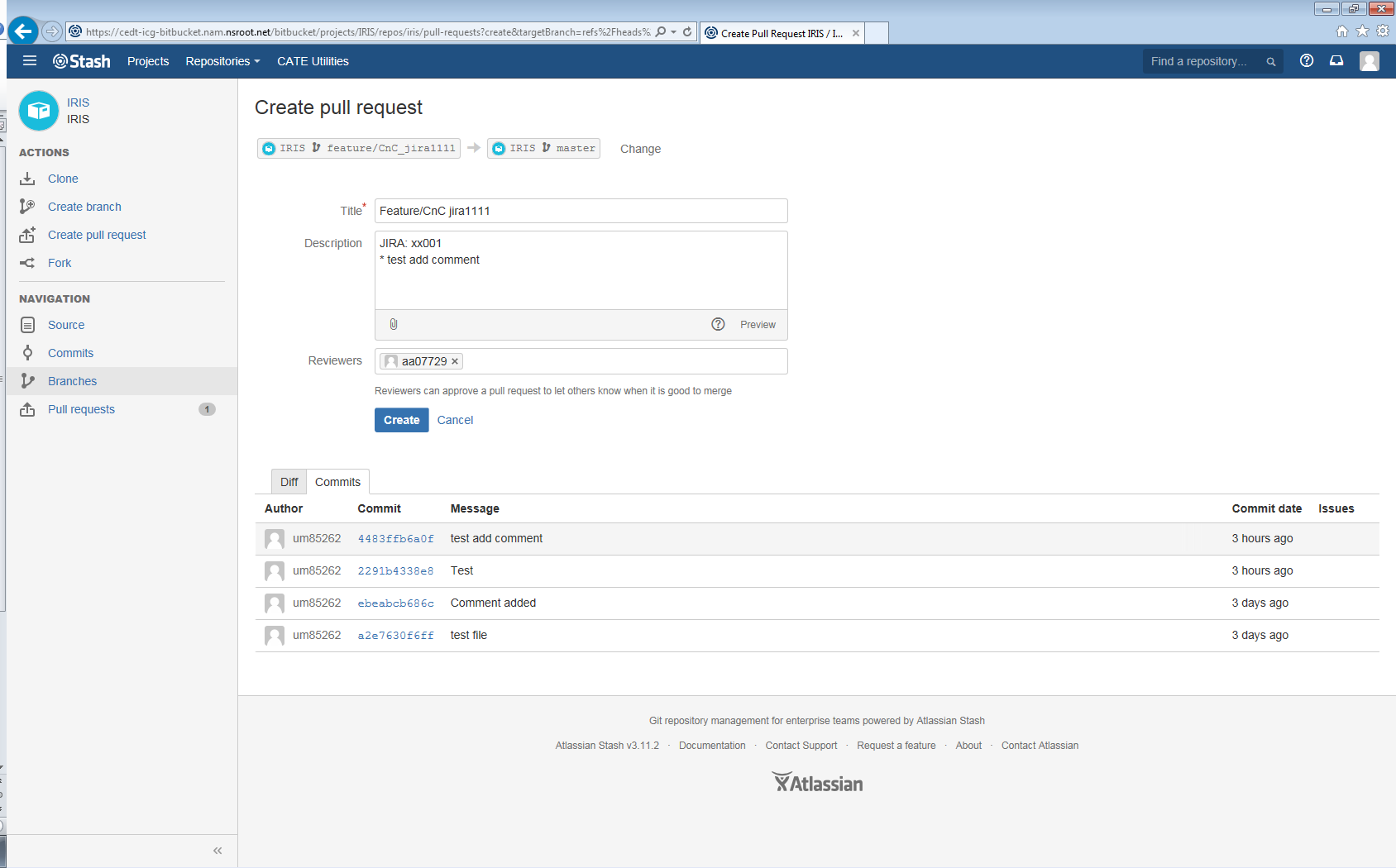
After the changes are pushed in remote branch, the developer should follow the below steps for merging the code in master branch.

**Developer Task:**

1. Developer creates a pull request for merging from a specific branch to master.

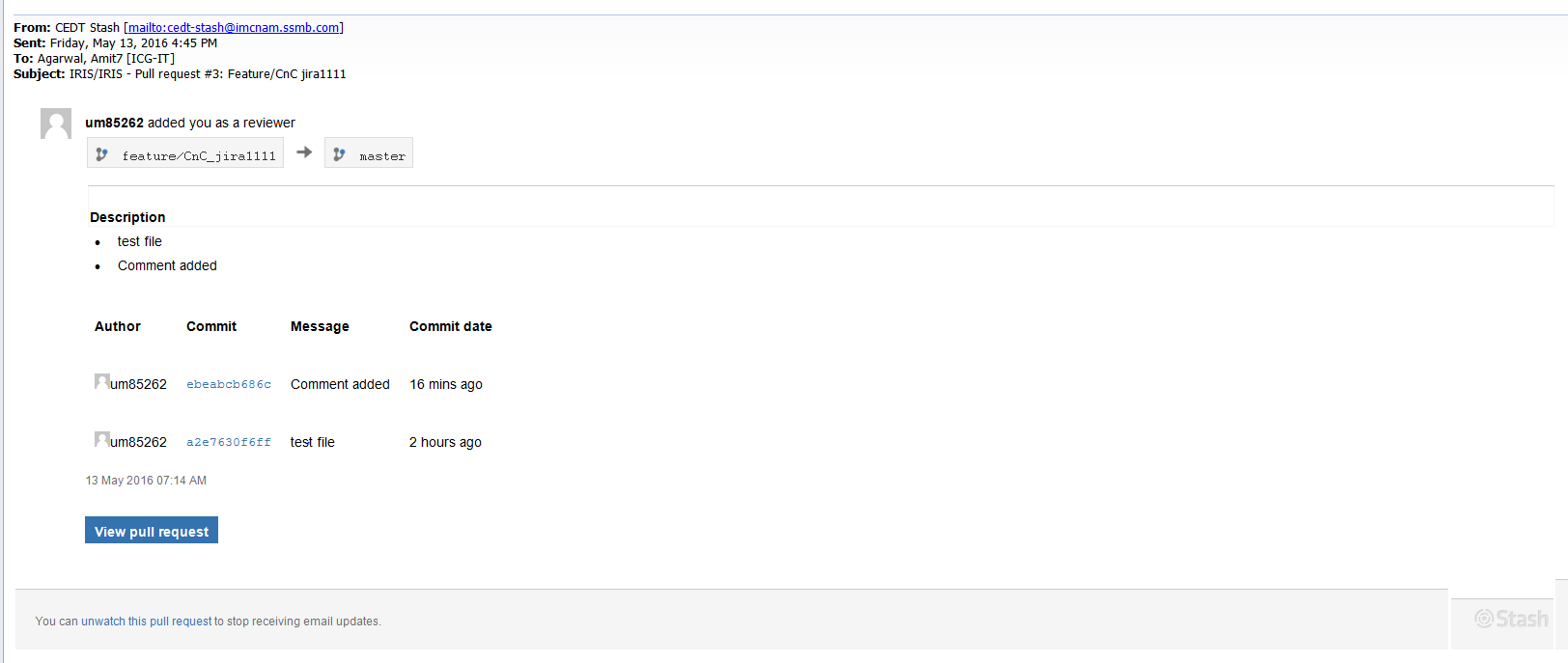


1. Add reviewers for approving the pull requests.

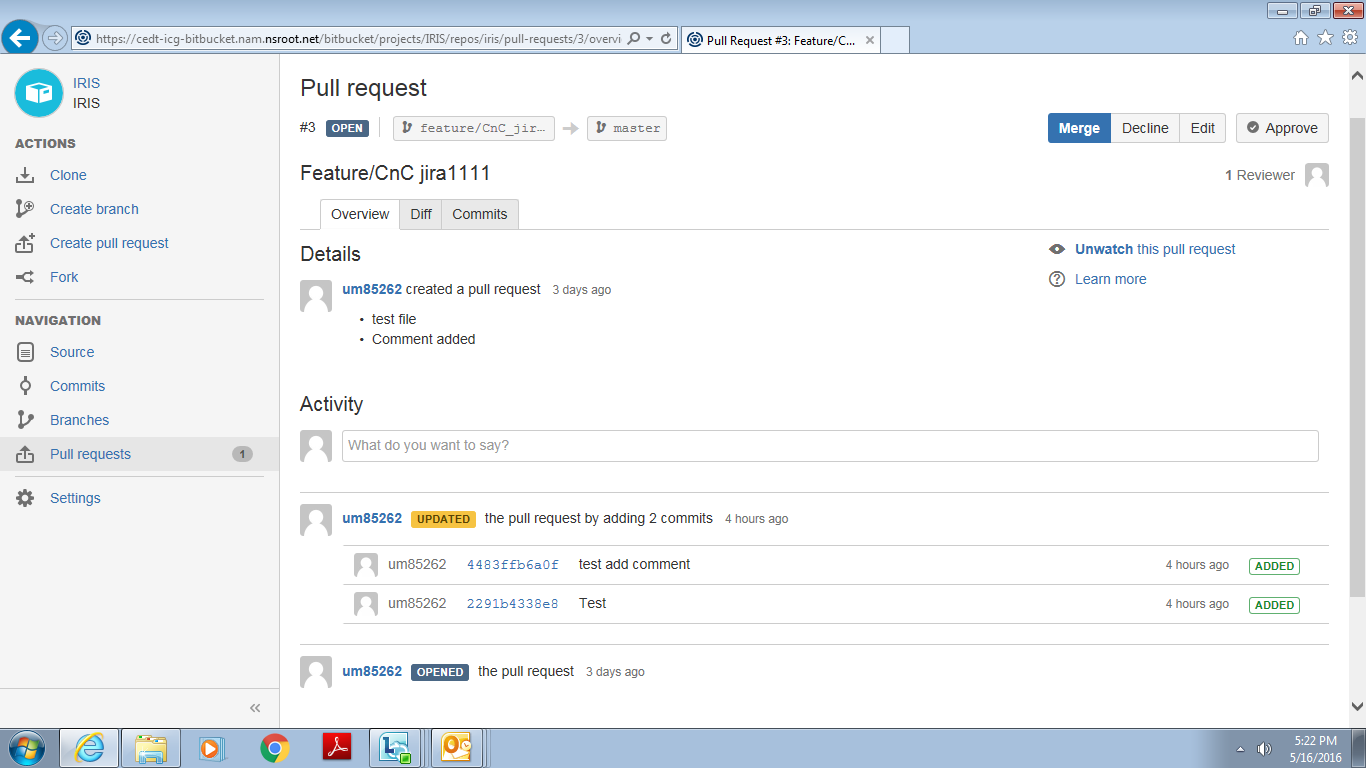


**Reviewer Task:**

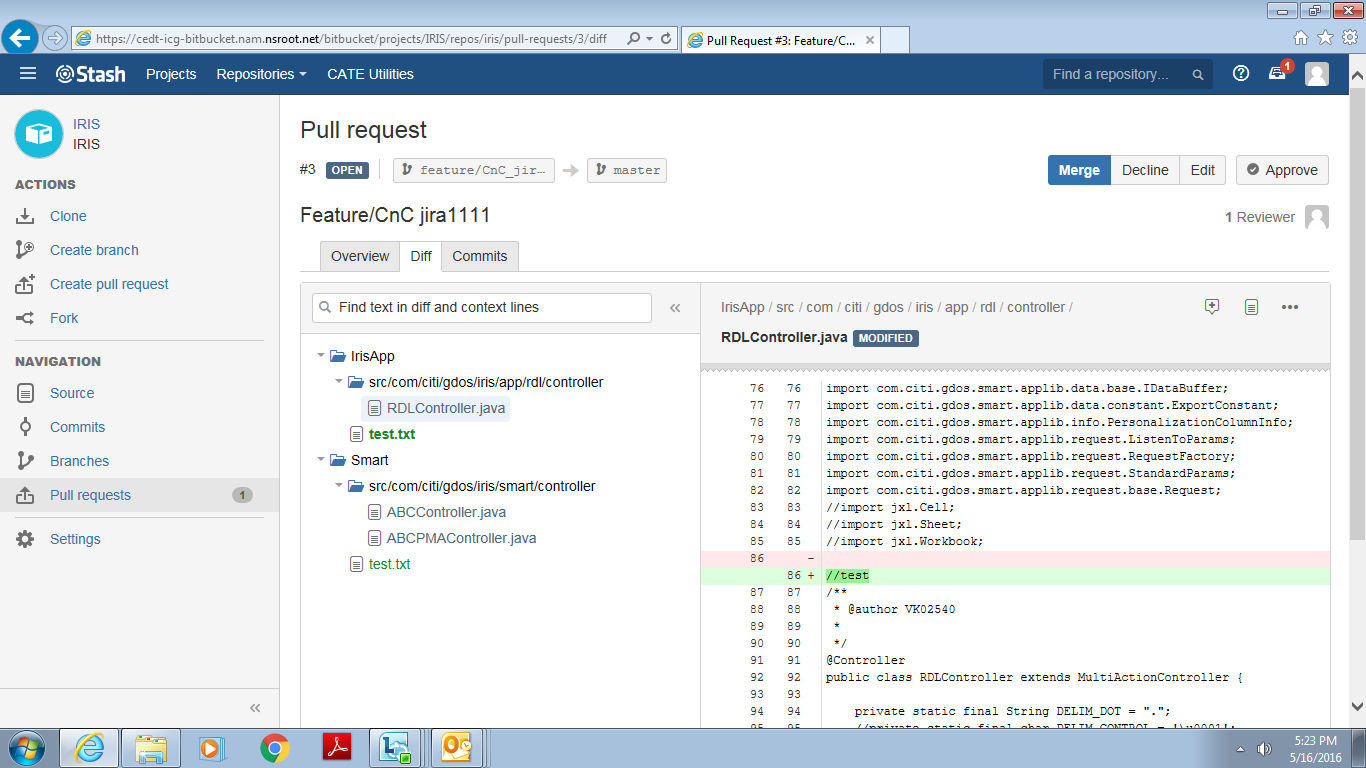
1. Reviewer gets a mail notification in outlook.



1. Reviewer can review the changes. Screen shows the overview of the changes, details of the changes as well as historical commits.



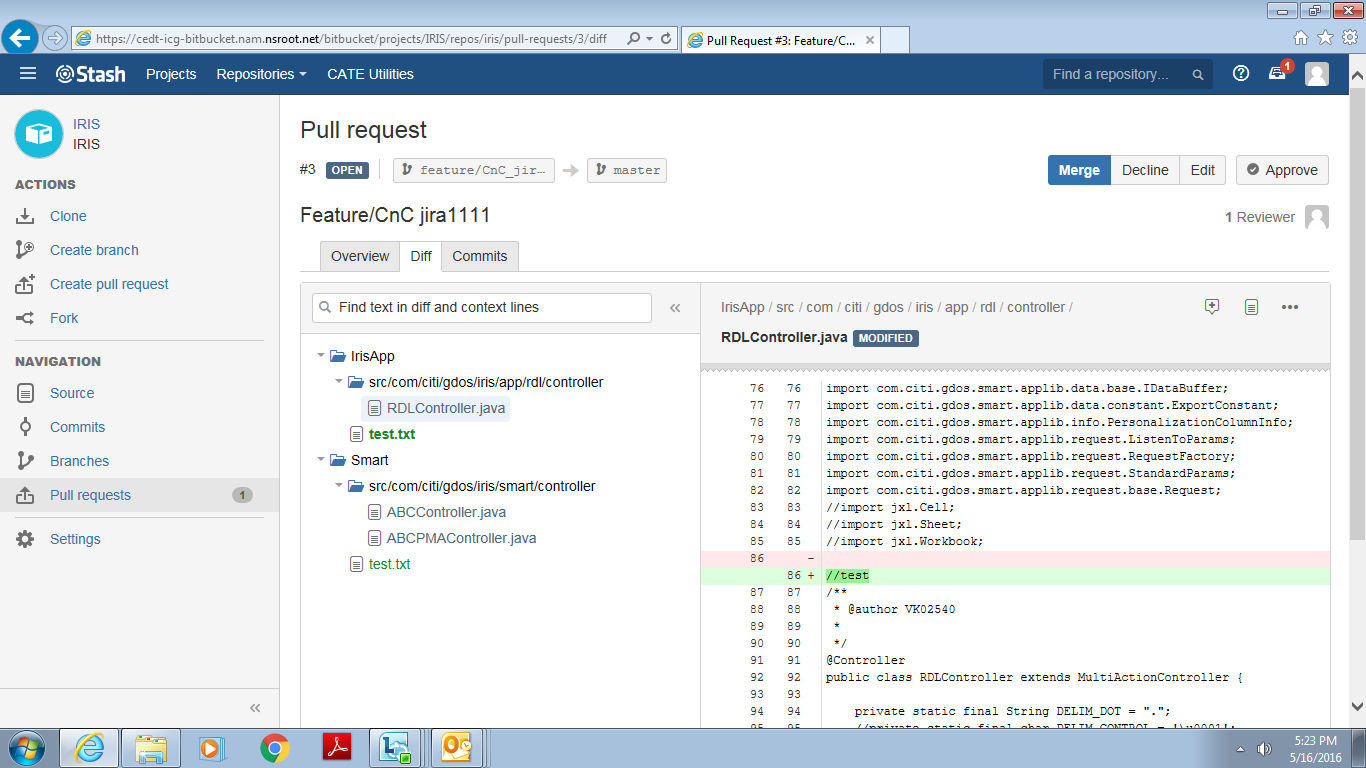
1. Review can see the difference in Bit Bucket as below and approve, edit or decline the request from here.



**Developer task:**

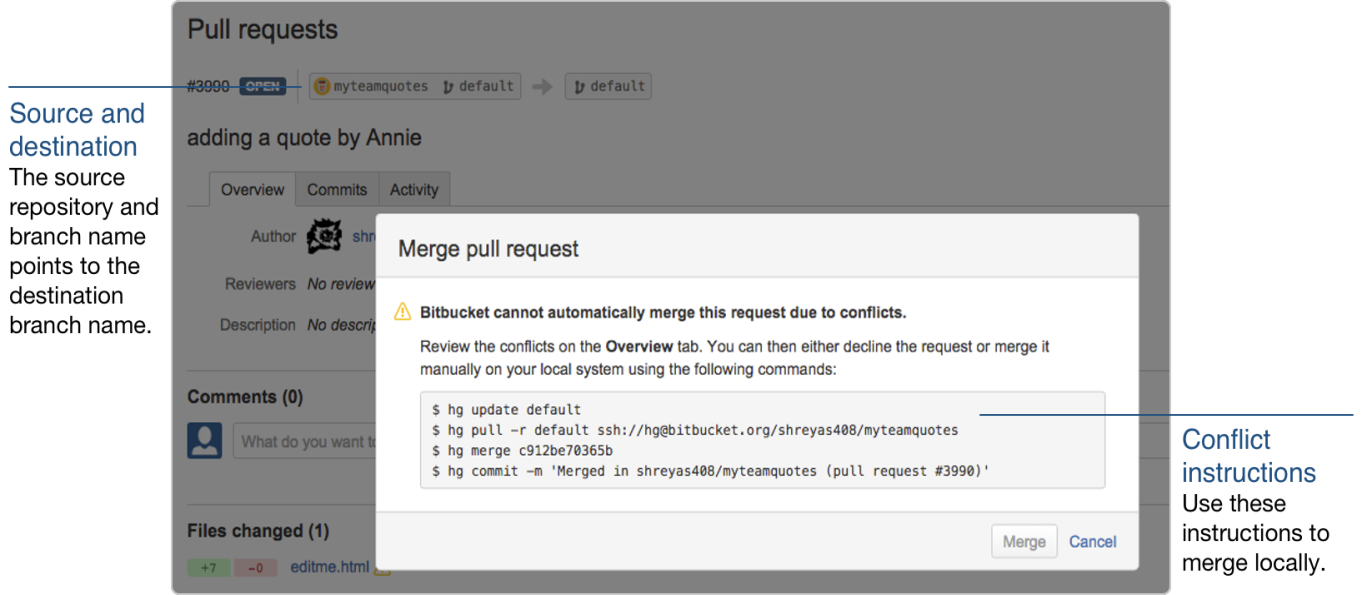
1. After getting Approved notification Developer have to merge his pull request by just clicking the merge button. Developer may get conflicts before merge remove the conflicts then merge.

Without conflicts:



1. With Conflicts

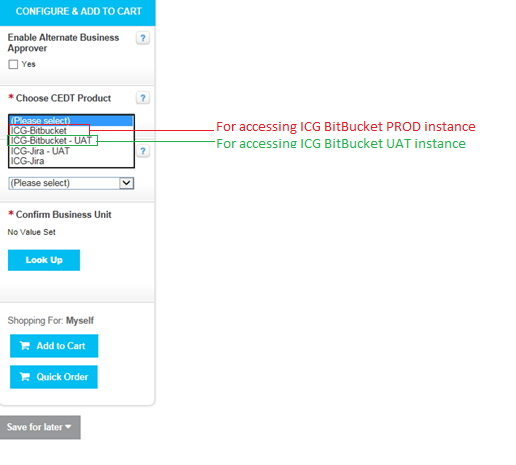
Steps to resolving conflicts popup come like this as shown below image:



# Appendix

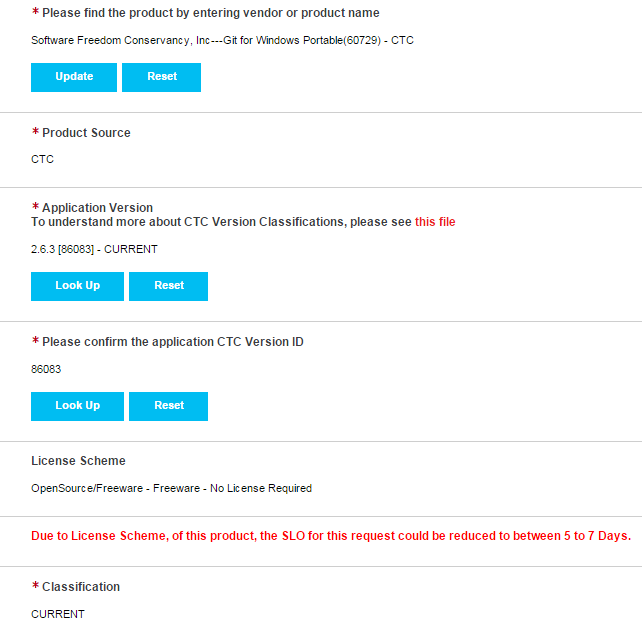
## Access on BitBucket Server

[https://cmp.nj.ssmb.com/marketplace/control/product/~productId=48645\_436286\_GLOBAL#](https://cmp.nj.ssmb.com/marketplace/control/product/~productId=48645_436286_GLOBAL)



## Install GIT client

[https://cmp.nj.ssmb.com/marketplace/control/product/~productId=47268\_465603\_GLOBAL?SEARCH\_STRING\_CAT=software#](https://cmp.nj.ssmb.com/marketplace/control/product/~productId=47268_465603_GLOBAL?SEARCH_STRING_CAT=software)



## Install Source Tree GIT Client.

[https://cmp.nj.ssmb.com/marketplace/control/product/~productId=47268\_465603\_GLOBAL?SEARCH\_STRING\_CAT=software#](https://cmp.nj.ssmb.com/marketplace/control/product/~productId=47268_465603_GLOBAL?SEARCH_STRING_CAT=software)

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  | | --- | |  | | [Global Other Desktop Software](https://cmp.nj.ssmb.com/marketplace/control/product/~productId=47268_465603_GLOBAL) Qty: 1   |  |  | | --- | --- | | Configuration Options | | | **Select your P2P Approver:** | Ravilla, Ushapriya R - 0003434057 | | **Pricing Disclaimer:** | Acknowledged | | **Please select your region:** | NAM | | **Click "Lookup" to confirm who you are requesting for:** | Saiganesh, Aswin - 1010580920 | | **SOEID:** | AS80920 | | **Country:** | India | | **Domain Name:** | NAM | | **OU:** | OU=GRN\_SW,OU=Users,OU=GCOT\_Users,OU=Accounts\_ICGCORP,OU=ICGCORP,DC=nam,DC=nsroot,DC=net | | **Please find the product by entering vendor or product name:** | ATLASSIAN SOFTWARE SYSTEMS PTY LTD---SourceTree (50271) - CTC | | **Product Source:** | CTC | | **Application Version To understand more about CTC Version Classifications, please see** [**this file**](https://catecollaboration.citigroup.net/entarch/eagovernanceinternal/Tools/Support/VERSION%20CLASSIFICATIONS.pdf)**:** | 2.2.4 [90065] - RESTRICTED | | **Please confirm the application CTC Version ID:** | 90065 | | **License Scheme:** | OpenSource/Freeware - Freeware - No License Required | | **Classification:** | RESTRICTED | | **Please attach approval email here:** | [CTC Product Deviation Request.msg](https://cmp.nj.ssmb.com/marketplace/services/public/getResource.jsp?dataResourceId=ATT3821552) | | **Where do you require the software to be installed?:** | Windows 7 (GCOTDVMR753805) VDI | | **Lookup the package based on your device selection:** |  | | **Business Justification:** | Need to connect with Git Server | | |

The following can be filled for the deviation request

