**Glossary Manager**

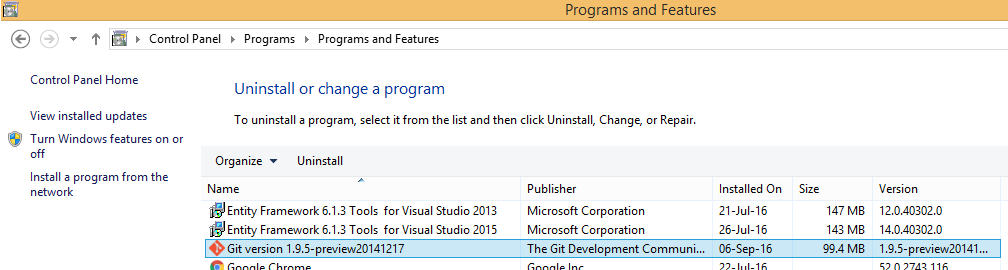
**Code and Environment Setup Document:**

* **List of Software required**

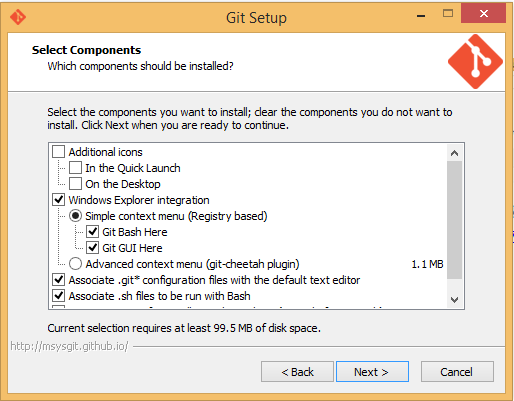
1. Visual Studio 2015 (C# version 6.0, .NET FRAMEWORK 4.6, ASP.NET MVC 5 FRAMEWORK)
2. SQL Server 2014 Management Studio.
3. Angular JS: 1.4.5
4. jQuery: 1.10.2
5. Node.js: 4.4.7
6. Bootstrap: 3.0.0
7. Jasmine Core: 2.2.0
8. Grunt: 1.0.1
9. Karma Test Case Runner: 1.0.0

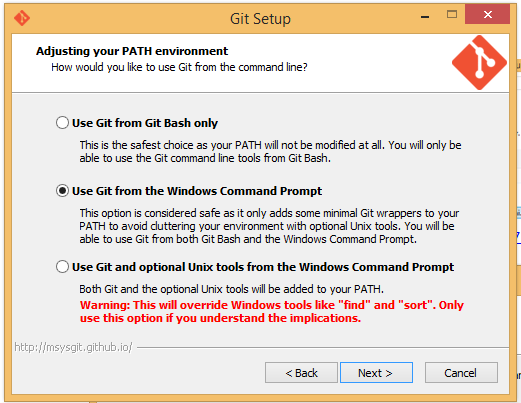
* **Below are the steps to Code and Environment Setup**

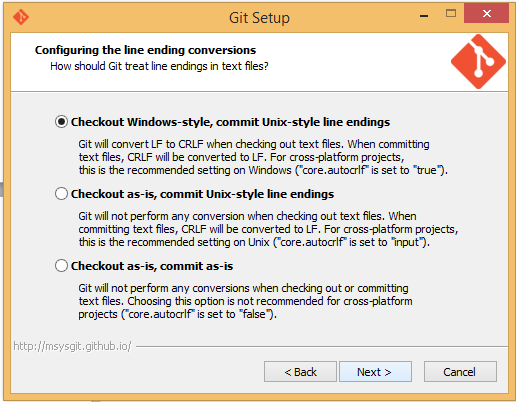
1. Remove/uninstall old version of GIT



1. Install git from [\\ct-share\UtilitiesForAll\GIT\Git-1.9.5-preview20141217.exe](file:///\\ct-share\UtilitiesForAll\GIT\Git-1.9.5-preview20141217.exe)
2. Select options as shown below and click Next

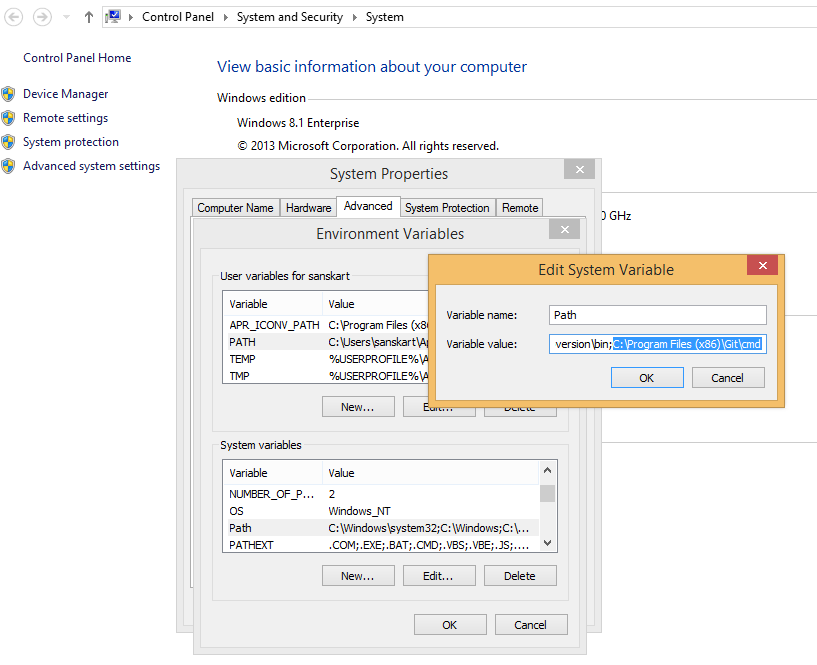




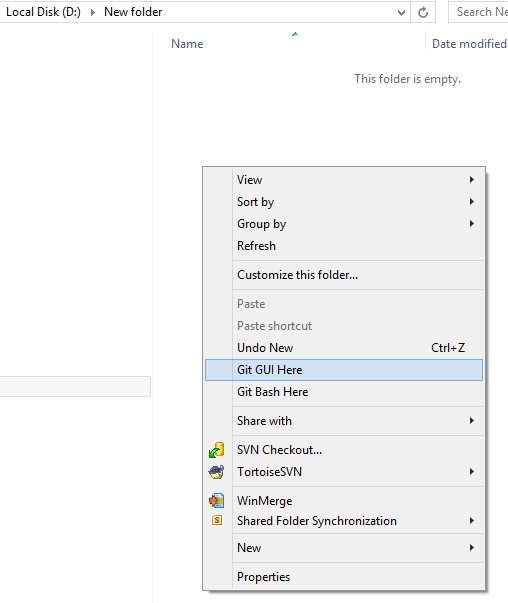


1. Goto ‘This PC/Properties/Advanced System Settings

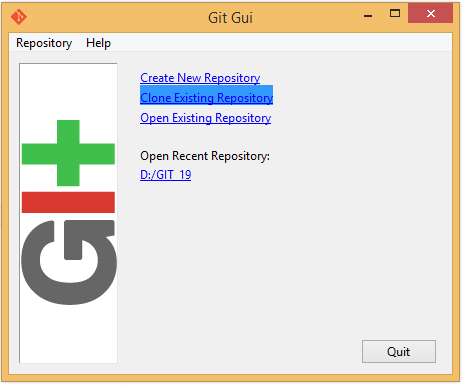
Then click ‘Advanced’ tab, click ‘Environment Variables’, select ‘Path’ in system variables and paster ‘;C:\Program Files (x86)\Git\cmd’ at the end of Variable value’. Please refer screenshot below



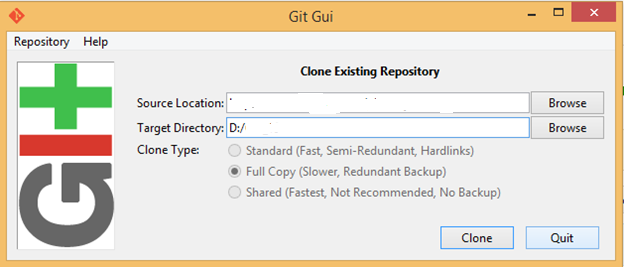
1. After installation is complete, right click in any new folder on d: and click ‘Git GUI Here’ from popup



1. Click ‘Clone Existing Repository’ link from popup



1. Give Source location and Target Directory as below and click ‘Clone’ button



1. You will be prompted for username and password for 2/3 times. Need to give details.
2. You will get **.git** hidden folder created and latest git master checkout at your local D:/GIT\_19 location.
3. From here you can use either ‘Git Bash’ or ‘Git GUI’ as per your convenience.

**Below are few quick commands to be used from Git Bash**

1. **Config**

git config --global user.name "yourfirstname-yourlastname"

git config --global user.email "youremailaddress"

1. **Creating Branch (use lowercase only)**

We already have below branches created here. NO need to create now.

1. release
2. sprint19

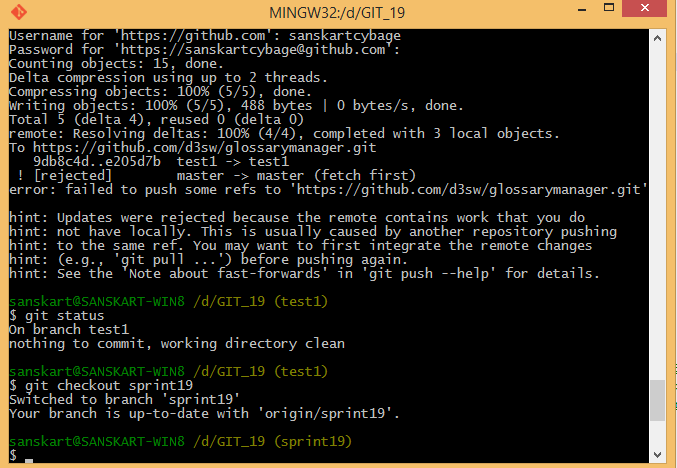
**Note:**

1. release branch will be merged by individual one day prior to every planned release. He/she should merge from sprint19 to release
2. sprint19 will be development branch which will be used for development of sprint19. This will be changed for each sprint.
3. **How to get checkout of sprint19 at local?**

git branch –a

git pull origin sprint19

git checkout sprint19



You will see (sprint19) as current active branch in ()

1. **Use GM Application for development**

You can use this solution to make changes in visual studio.

1. **Few more git commands**

git branch sprint19 (create branch)

git checkout sprint19 (move to that branch)

git branch -D sprint19 (delete unused branch)

git pull origin sprint19 (to pull branches from other users)

git branch -a

git branch --all (to see branches created by other users)

git help commit (help on commit)

git add <filename> (to add file for commit)

git add . (to add all modified files)

git commit -m 'message'

git rm <filename> (to remove a file)

------------

git status (to verify the branch pending changes)

git add 'GM\_APP/Version.html'

git commit -m 'any comment' (to commit changes only after git add command)

git status (to verify the branch pending changes)

git merge sprint19 (use this from release to get sprint19 changes in release)

git push origin master (origin(to), master(from))

git push origin sprint19:master (origin(to), sprint19(from))

-----------

git stash save myChanges(to save aside changes b4 switching branch, it does not stash newly added file until used git add command)

git stash show (gives stashed filename, changeset)

git stash list (gives list of stash names .. myChanges, etc)

git stash drop myChanges (deletes the specified stash entry)

git stash apply (to recall the stashed changes after re entering stashed branch)

git stash clear (which deletes all entries of the stash)

-------

git tag (to see tags instead of branches)

git tag '2016\*'

----------

git log

git log --pretty=oneline (press just 'q' to exit <END>/scroll of log)