**Git Developer notes**

1. **Request an issue** – email [cmsupport@usf.edu](mailto:cmsupport@usf.edu) using this format below.

Note: repository names will be the same as the JIRA project names

(aisdw, absfam, absfbog, absgam, absgbog, oashits, oasadms, oasam, oasban, oasbdmsam, oasbdms, oasbog, oasdgwks, itam, faam)

**Please create an issue:**

**Project/Repository:** OASIS Banner Development (OASBAN)

**Issue Type:** Improvement, Fix, etc.

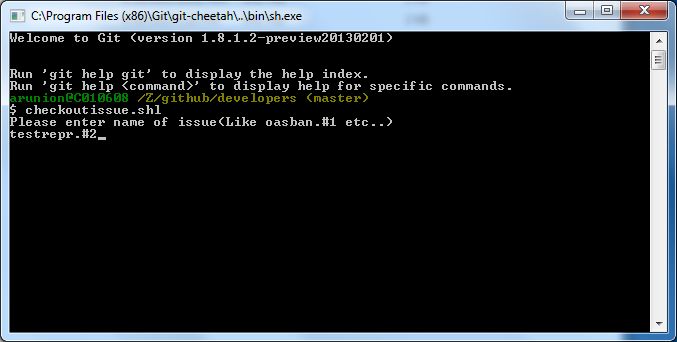
**Project/Issue number:**  12-0232

**Description:** SSFAD DER maintenance

**Technical Manager:** Praveen Sadagopan

1. **Work on an issue** –

From Z🡪github🡪developers right click checkoutissue.shl, highlight & click gitbash, type checkoutissue.shl in the gitbash window and hit enter. You are then prompted for the issue number. Enter the issue number (i.e. oasban.#1) – repository name, dot/period, pound sign, issue number (format is very important). Example below shows issue testrepr.#2:



To verify, go to C🡪github🡪oasban (or whatever the repository name is) right click and point your mouse on git branch and it should show the issue name (i.e. oasban.#1).

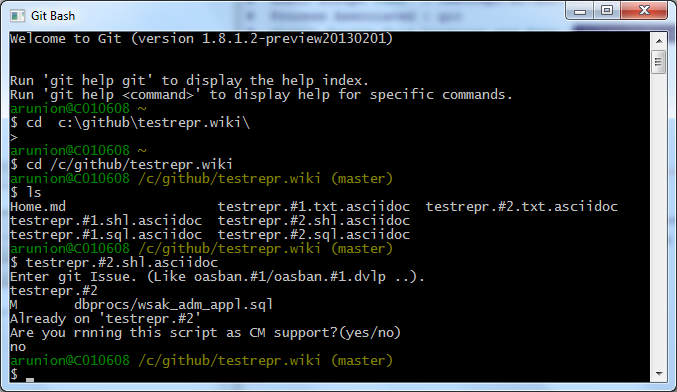
1. **Modify code -** Modify your code and save under c:\repositoryname\objecttype (i.e. c:\oasban\dbprocs).
2. **Create supporting documents like shell, sql, and text files -** , create .shl file, .sql file, and .txt file and save in c:\github\repositoryname.wiki (i.e. c:\github\oasban.wiki)

**Templates are available here: Z:\github\developers\template.**

**(5) Test:**

**Open wiki folder (**c:\github\oasban.wiki) in Git Bash and run <issue>.shl.asciidoc. (oasban.#1.shl.asciidoc).

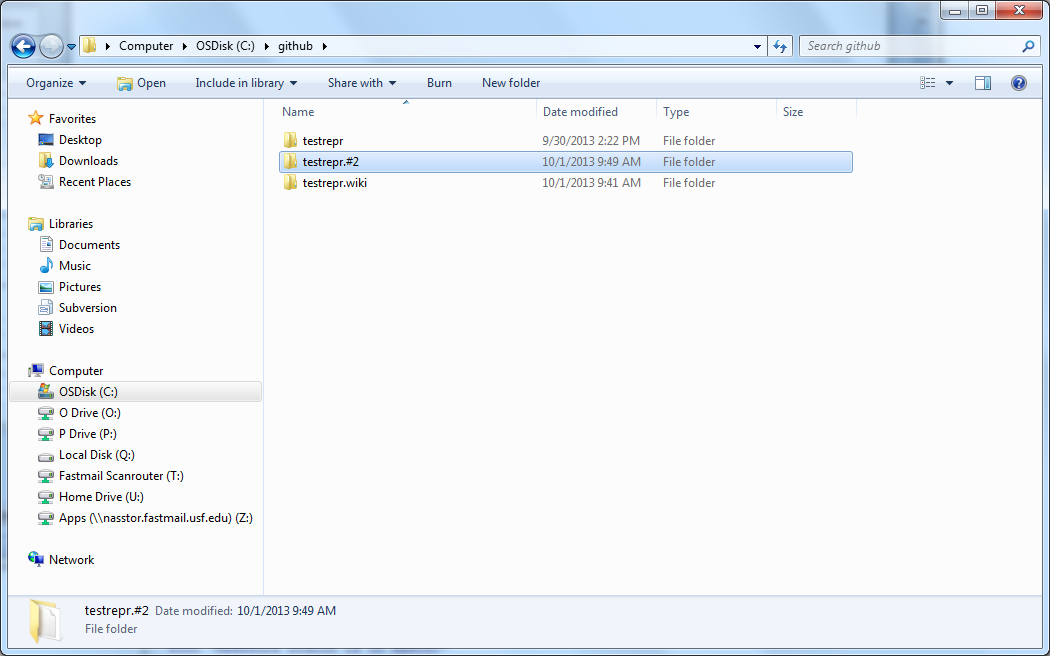
In the example below, testrepr.#2.shl.asciidoc is run from Git Bash.



This creates <issue> folder(oasban.#1).

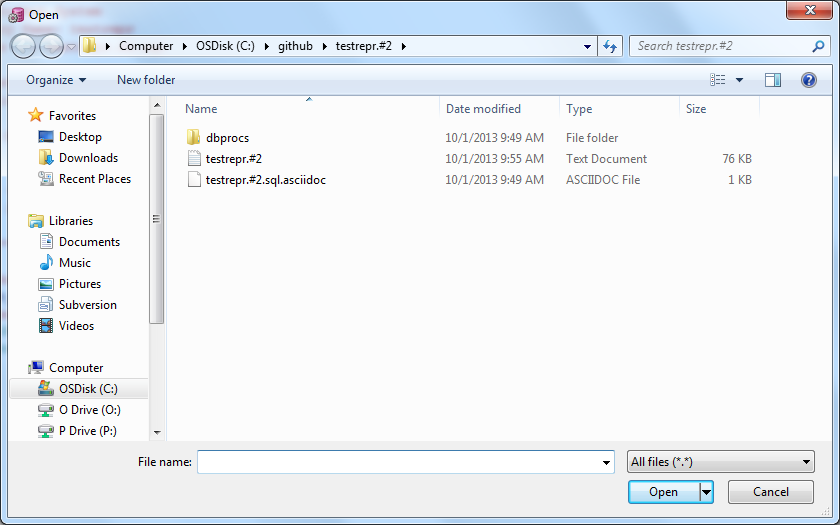
Note: <issue> folder (oasban.#1) is not Git Repository. This is for you to test whether your objects are copied into <issue> folder. Once you are done with test, please remove issue folder that is having issue number extension(oasban.#1).

In the example below, issue folder testrepr.#2 is created. This folder (testrepr.#2, NOT testrepr or testrrepr.wiki ) should be deleted after testing.



**Compile: If you have database objects, then compile objects to test.**

Open SQl window and open <issue> (oasban.#1) folder. Run <issue>.sql.asciidoc (oasban.#1.asciidoc) testrepr.#2.sql.asciidoc that is under c:\github\testrepr.#2 folder). In the example below, testrepr.#2.sql.asciidoc is opend from plsql developer tool. You can open the script using any tool of your choice.

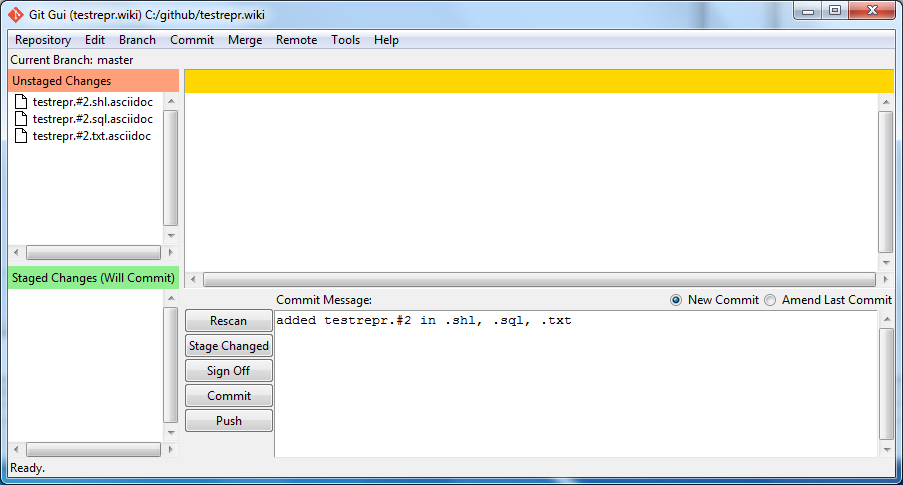


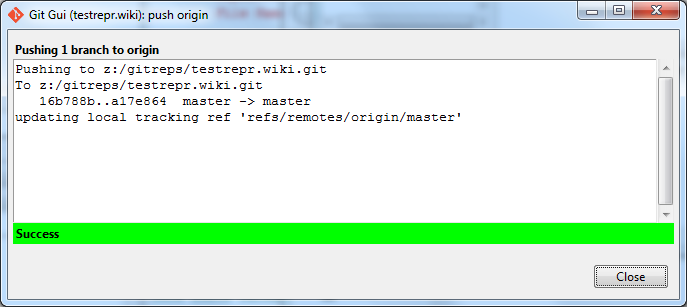
1. **Remove <issue> (oasban.#1).** After running driver script and checking log, if everything looks good, remove <issue> (oasban.#1) folder.
2. **Commit supporting documents – O**pen <project>.wiki (c:\github\oasban.wiki) in Git Gui, put message "<issue>(oasban.#1) ", click on stage change, respond yes to ‘stage untracked files’ , click on commit, and click push. This commits the code you’re your local C drive. If you get error when you push to local server, then open z:\github\developers folder in Git Bash and run updatewiki.shl. Again open <project>.wiki (c:\github\oasban.wiki) in Git Gui and Push. This time you should not see error.

**NOTE:** When you do a commit to your local repository, please put the entire issue name instead of just the issue number.

For Example, when committing to the oasban.#2 issue, instead of just putting “ #2  changes made to wsak\_adm\_appl”, now you need to use “oasban.#2 changes made to wsak\_adm\_appl”.

In the example below, supporting documents for testrerpr.#2 issue is committed. (Note that the folder is testrepr.wiki NOT testrepr).

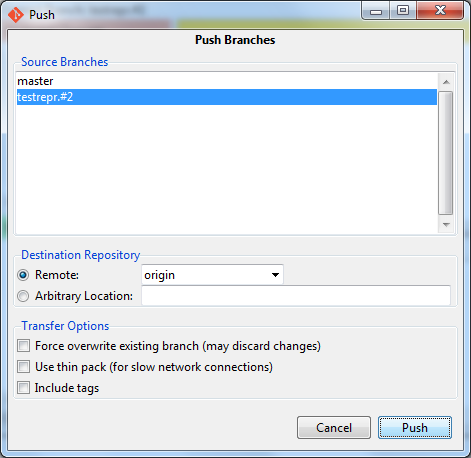


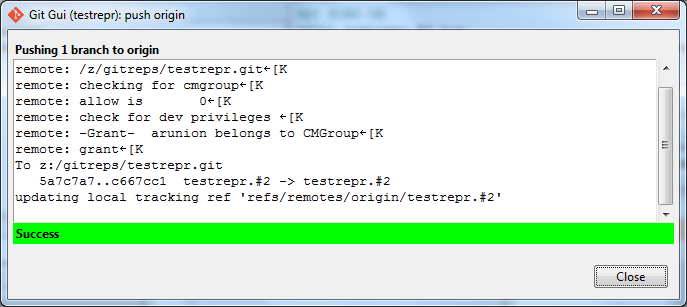


1. **Commit code – O**pen <project> (c:\github\oasban) in Git Gui, put message "<issue>(oasban.#1) Added …", click on stage change, respond yes to ‘stage untracked files’ , click on commit, and click push. This commits the code you’re your local C drive.

**NOTE:** When you do a commit to your local repository, please put the entire issue name instead of just the issue number.

For Example, when committing to the oasban.#2 issue, instead of just putting “ #2  changes made to wsak\_adm\_appl”, now you need to use “oasban.#2 changes made to wsak\_adm\_appl”.

In the example below, code for testrepr.#2 is committed and pushed. 



1. **Request code to be promoted –**

email [cmsupport@usf.edu](mailto:cmsupport@usf.edu) and ask them to promote to ready for DVLP/TEST/etc.

**10) If you need to change code after issue Is promoted (demoting):**

Request CM Support to demote your issue from dvlp/test environment (**specify which environment your code is in**).

Checkout issue.

Make changes to code. If changes required for supporting documents, do changes in wiki folder.

Commit and push.

**NOTE: After you merge your branch to DVLP, you should be cleaning it up in your repository.**

**If you need to work on the same issue again, you should send a request to CM support asking for the issue to be demoted from DVLP/Test.**

**To cleanup merged branches from your c:\github repository, use these steps:**

1. Go to z:\github\developers folder and open Git Bash.
2. Run cleanlocalbranches.shl. Enter <repository> when promoted. This will remove all branches except master.