Git-Blit Tutorial

We use Git-Blit to allow automation teams to maintain their automation assets within a central repo. This ensures that all of the automation in sync and no one is overriding another person's changes while working on the same project. Git-Blit allows for version control across the automation assets- if the code breaks it can be reverted back to a recent version without losing too much data.

It is an open source tool that we use extensively because it gives us complete control over installation and security. Security is in check because it can easily be deployed within the firewall. Below are the steps that walk you through the Git-Blit Setup and basic usage (checking-in, checkin-out, etc.):

Git Fetch

- 1. Open Git GUI
- 2. Click on Remote\Fetch From\Origin

🧃 Git Gui (relms_scripts) C:/Users/U363074/De	esktop/git_repo/relms_scripts	. D X
Repository Edit Branch Commit Merge	Remote Tools Help	
Current Branch: master	Fetch from origin	
Unstaged Changes	Prune from	
	Remove Remote	~
	Add Ctrl-A	
	Push Ctrl-P	
	Delete Branch	
-		
< >		
Staged Changes (Will Commit)		
	Commit Message:	nd Last Commit
	Rescan	
	Stage Changed	
	Sign Off	
	Commit	
-	Push	
< >		*
Ready.		

3. Wait till you see Fetching new changes from Origin Success message as shown in the following screenshot.

 Git Gui (relms scripts) C·/Users/U363074/Des 	ktop/ait repo/relms scripts	x
President Eth Prest, Constitution		
Repository Edit Branch Commit Werge	Remote Tools Help	
Current Branch: master		
Unstaged Changes		
Â	Git Gui (relms_scripts): fetch origin	23
	Fetching new changes from origin	
	From ssh://wnl-svr081.wellsfargo.com:29418/relms_scripts 521b9ebab5d65d master -> origin/master	
< >		
Staged Changes (Will Commit)		-
	Success	
	۲ Close	e
	Kescan	-
	Stage Changed Sign Off Commit Push	
*		~
Ready.		

Merge Changes Locally

1. Click on Merge Tab and click Local Merge

Git Gui (relms_scripts) C:/Users/U363074/D	lesktop/git_repo/relms_scripts	_ D X
Repository Edit Branch Commit Merg	e Remote Tools Help	
Current Branch: master	cal Merge Ctrl-M	
Unstaged Changes Al	port Merge	
۲۰ المعالم الم Staged Changes (Will Commit)		
	Commit Manager	,
	Bercan	end Last Commit
	Stage Changed Sign Off Commit Push	Î
Ready.		-

2. Choose the "**Tracking Branch**" where you want the merge to take place and click **Merge** button

Repository Edit Branch Commit Merge Remote Tools Help	🖣 Git Gui (relms_scripts) C:/Users/U363074/Desktop/git_repo/relms_scripts	
Merge Image Merge Into master Revision To Merge Revision Expression: Local Branch @ Tracking Branch @ Tag Origin/HEAD Origin/Master	Repository Edit Branch Commit Merge Remote Tools Help	
Merge Into master Revision To Merge Revision Expression: Local Branch Tracking Branch Tag origin/HEAD origin/HEAD origin/master	t Merge	
Revision To Merge Revision Expression: Local Branch Origin/HEAD origin/master	Merge Into master	
 Revision Expression: Local Branch Tag origin/HEAD origin/master 	Revision To Merge	A
Local Branch Tag origin/HEAD origin/master	Revision Expression:	
origin/HEAD origin/master	💿 Local Branch 💿 Tracking Branch 💿 Tag	
origin/master	origin/HEAD	
	origin/master	
		*
Visualize Cancel Merge	Visualize Cancel Merge	
© New Commit © Amend Last Commit	Commut Message.	New Commit O Amend Last Commit
Rescan	Rescan	
Stage Changed	Stage Changed	
Sign Off	Sign Off	
Commit	Commit	
Push	Push	
	· · · · · · · · · · · · · · · · · · ·	
Ready	* Peady	Ψ

3. Wait to see the Merge Success message and close the Merge pop up window.



Pushing Your Local Changes to Git Repository

1. Click "**Rescan**" to see your local changes.

🗯 Git Gui (relms_scripts) C:/Users/U363074/Des	sktop/git_repo/relms_scripts		x
Repository Edit Branch Commit Merge	Remote Tools Help		
Current Branch: master			
Unstaged Changes	Modified, not staged File: features/step_definitions/relms_uat.xls		
features/DP_Features/LDBP_Global Comi features/DP_Features/LDBP_Purpose Coc features/step_definitions/Relms_edit.rb features/step_definitions/relms_uat.xls	Binary files a/features/step_definitions/relms_uat.xls and b/features/st	ep_defin	iti 🔺
4 III >			
Staged Changes (Will Commit)			
	 III Commit Message: New Commit © A Rescan Stage Changed Sign Off Commit Push 	mend Last (Ecommit
-			-
Ready.			

2. Click on "Stage Changed" to stage your local changes to the Git Repository. Click "YES" on the browser alert.

🗧 Git Gui (relms_scripts) C:/Users/U363074/Des	ktop/git_repo/relms_scripts	
Repository Edit Branch Commit Merge	Remote Tools Help	
Current Branch: master		
Unstaged Changes	Modified, not staged File: featur	es/step_definitions/relms_uat.xls
features/DP_Features/LDBP_Global Com features/DP_Features/LDBP_Purpose Coc features/step_definitions/Relms_edit.rb features/step_definitions/relms_uat.xds	Binary files a/features/step_definitions/relm	s_uat.xls and b/features/step_definiti
4 III >		
Staged Changes (Will Commit)	Git Gui (relms_scripts)	 New Commit Amend Last Comm
* *		
Ready.		

3. Once you click "Yes" you should be able to see all the new changes marked under the Staged Changes Column.

an dar (tenns_serba) er ösers/öses/ösesköl/grejebo/renns_serba		
Repository Edit Branch Commit Merge Remote Tools Help		
Current Branch: master		
Unstaged Changes		
Staged Changes (Will Commit) Staged Changes (Will Commit) features/DP_Features/LDBP_Global Com features/step_definitions/relms_uat.xls features/step_definitions/relms_uat.xls Commit Message: New Commit Am Stage Changed Sign Off Commit Push	nd Last (Commit
Ready to commit.		

4. After staging the changes, click "**Commit**" to commit the changes to the Git Repository and then click "OK" on the browser alert.



5. Add your custom Commit Message relevant to the changes that you are willing to commit to Git.



6. Click "**Commit**" again and wait for your commit message to appear at the bottom of Git GUI page.

🧃 Git Gui (relms_scripts) C:/Users/U363074/D	esktop/git_repo/re	elms_scripts	
Repository Edit Branch Commit Merge	e Remote Tool	ls Help	
Current Branch: master			
Unstaged Changes			
4			*
Staged Changes (Will Commit)			
	<		
		Commit Message:	New Commit
	Rescan Stage Changed Sign Off Commit Push		
			-
Created commit 3ecbffe7: Deepty: Added New	Feature Files and c	hanges to Custom steps	

7. Click "**Push**" on Git GUI and click "**Push**"again after verifying that the correct source branch is selected on the Push Branches pop up window. The application might ask you to enter your password for authentication so enter your password.

	🧓 Push	
🚔 Git Gui (relms_scripts) C:/Users/U363074/Desktop/git	Push Branches	— — X
Repository Edit Branch Commit Merge Remot	Source Branches	
Current Branch: master	master	
Unstaged Changes		
Staged Changes (Will Commit)		ommit Amend Last Commit
Created commit 3ecbffe7: Deepty: Added New Feature File	es and changes to Custom steps	

8. Wait to See "Success" on Push Origin pop up window.

🗧 Git Gui (relms_scripts) C:/Users/U363074/Desktop/git_repo	/relms_scripts	— — X
Repository Edit Branch Commit Merge Remote To	ols Help	
Current Branch: master		
Unstaged Changes	Bit Gui (relms_scripts): push origin	
<	Pushing 1 branch to origin Pushing to ssh://deeptymayi@wnl-svr081.wellsfargo.com:29418/reremote: Resolving deltas: 100% (6/6) +[Kremote: Resolving deremote: Updating references: 100% (1/1) +(Kremote: Updating To ssh://deeptymayi@wnl-svr081.wellsfargo.com:29418/relms_scriab5d65d3ecbffe master -> master m Success m	elms_scripts.git eltas: 100% (6/6)+[references: 100% (pts.git
	Commit Message:	mend Last Commit
Rescan Stage Chang Sign Off Commit Push		
Created commit 3echffe7: Deenty: Added New Feature Files and	changes to Custom steps	*
created commit Seconer, Deepty, Added New Feature Files and	changes to custom steps	

Validating that Changes have been Pushed Successfully

Once you have pushed your changes to the Git Repo, it is important to validate that you have all the changes pushed successfully and your changes are merged with changes made by other members of your team in the same Git branch. We can follow the steps below to validate the same:

1. Click on "Repository" tab on Git GUI and click "Visualize All Branch History"



2. You should see the recent commit message added by you and your name at the top of the branch. The "**Yellow**" dot is your update. If the yellow dot is at the top, it means you have synced all the changes made by you and your team members and you have made the latest update/commit to Git Repository.



3. You can also verify that your changes have been successfully pushed to Git Repository by logging in to Gitblit. The first commit under the recent activity list in "**My dashboard**" should be you.

Here is a screenshot:

ä	Blit my dat	shboard repositories a	ctivity search	filters -					-
Ā	Welcome to Gitl	blit pr view your own Git reposito	ies.			starred owr	ied active	projects	
r	recent activity last 7								
	adive repositories	active authors 63% truary 5, 2015 of relms_scripts by Deepty ty: Added New Feature Files vy. February 4, 2015 of relms_scripts by 3 auth- ut comment ty/Added new feature files b branch master of ssn/ideute files by Adding 3 new feature files	and changes to vs phymayi@wnl-sv	235 Custom steps #081.wellsfargo.com/294	418ir				
	2 days ago, Monday, 7 commits to mateur based Sarat eseration Sarat eseration Sarat eseration Sarat eseration Sarat eseration Sarat sarat eseration Sarat eseration Sarat eserat eseration Sarat eseration Sarat eseration Sarat eseration Sar	February 2, 2015 r of relms_scripts by 2 auth h Add one feaure file and up. Feature files and updated Rt u-ort-by: Sarath <u3903556 a branch 'master' of ssh./ijoh ad John's excel changes and</u3903556 	ors ate on excel file c by steps for 2nd DTCD282442CU n@wnl-svr081.w ruby fie changes	on 02/02/15 I Feb 660A.ent.wfb.bank.corp≻ rell5argo.com:29418/reli s.	ms_s				