Bitbucket

SVN to Git Migration

&

Operations using GitHub Desktop

Table of Contents

Document Information

Project Name:	Project LPKM		
Prepared By:	Swapna Munnangi	**Document Version No:**	0.1
Title:	SVN to Git Migration & Operations using GitHub Desktop	**Document Version Date:**	12-Apr-18
Reviewed By:	Molakala Reddy Praveen	**Review Date:**	

Document Version History

Version Number	**Version Date**	**Prepared by**	**Revised By**	**Description**
1	12-Apr-18	Swapna Munnangi	Molakala Reddy Praveen	SVN to Git Migration & Operations using GitHub Desktop

SVN to Git Migration

1. Prerequisites for Migration

- SVN installation setup on local system
  SVN: https://tortoisesvn.net/downloads.html

- Ordering Bit bucket [Using below URL]
  https://bmwprod.service-now.com

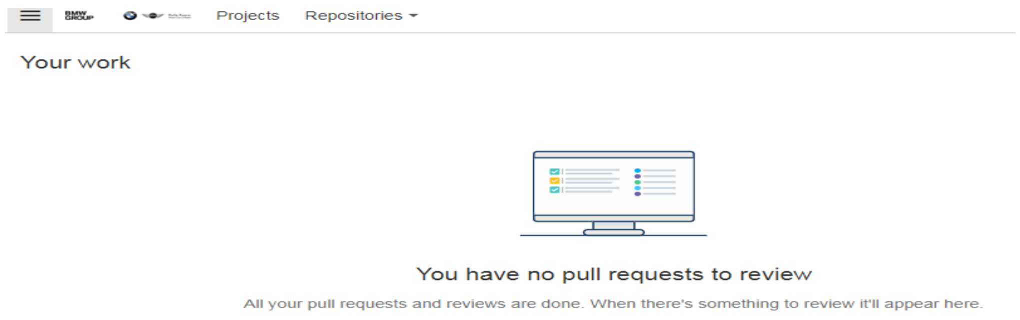
- Access to SVN Repository
- Access to Bit bucket

2. Implementation steps involved in migration

1. Creation of Git REPOSITORY

   - Login to the below URL using QX number and password to see the below screen

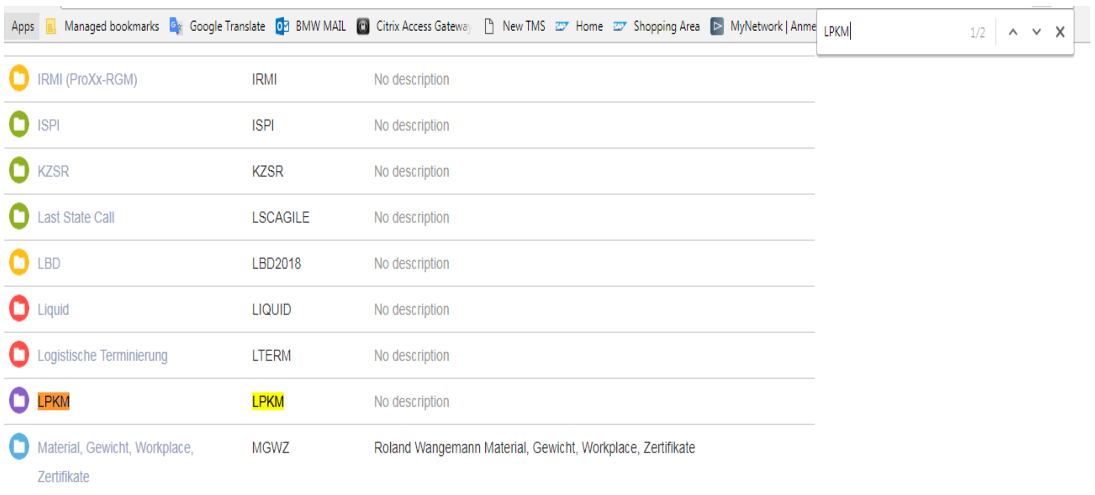
     URL : https://atc.bmwgroup.net/bitbucket/



You have no pull requests to review

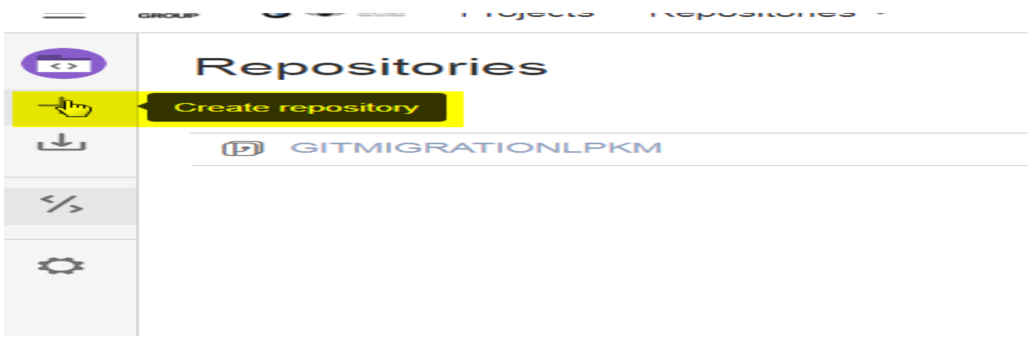
All your pull requests and reviews are done. When there's something to review it'll appear here.

2. Now click on Projects, search for the application(LPKM) and click on it

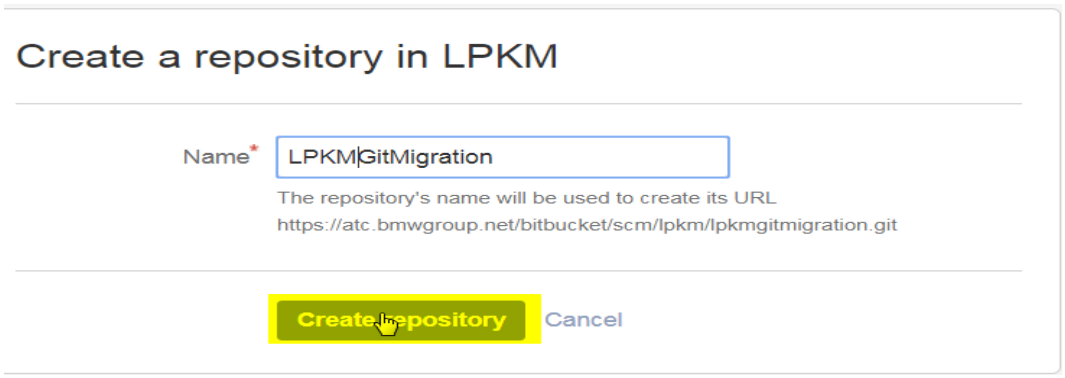


IRMI (ProXx-RGM)	IRMI	No description
ISPI	ISPI	No description
KZSR	KZSR	No description
Last State Call	LSCAGILE	No description
LBD	LBD2018	No description
Liquid	LIQUID	No description
Logistische Terminierung	LTERM	No description
LPKM	LPKM	No description
Material, Gewicht, Workplace, Zertifikate	MGWZ	Roland Wangemann Material, Gewicht, Workplace, Zertifikate

3. Click on 'Create repository'



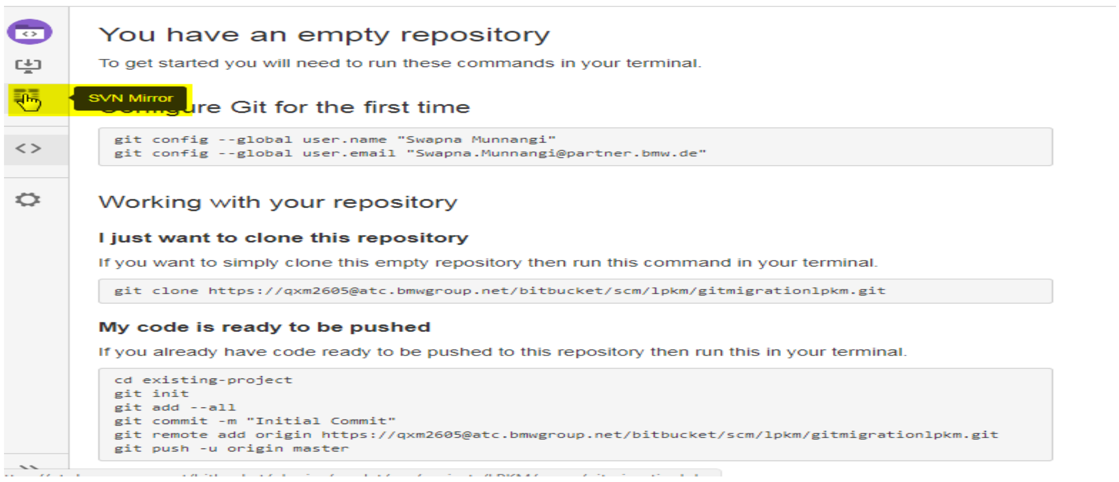
4. Specify a name to your repository and click on ‘Create Repository’ button.


Create a repository in LPKM

Name* LPKMGitMigration

The repository's name will be used to create its URL https://atc.bmwgroup.net/bitbucket/scm/lpkm/lpkmgitmigration.git

5. Go to SVN Mirror settings in the newly created Repository



You have an empty repository

To get started you will need to run these commands in your terminal.

Configure Git for the first time

```
git config --global user.name "Swapna Munnangi"
git config --global user.email "Swapna.Munnangi@partner.bmw.de"
```

Working with your repository

I just want to clone this repository

If you want to simply clone this empty repository then run this command in your terminal.

```
git clone https://qxm2605@atc.bmwgroup.net/bitbucket/scm/lpkm/gitmigrationlpkm.git
```

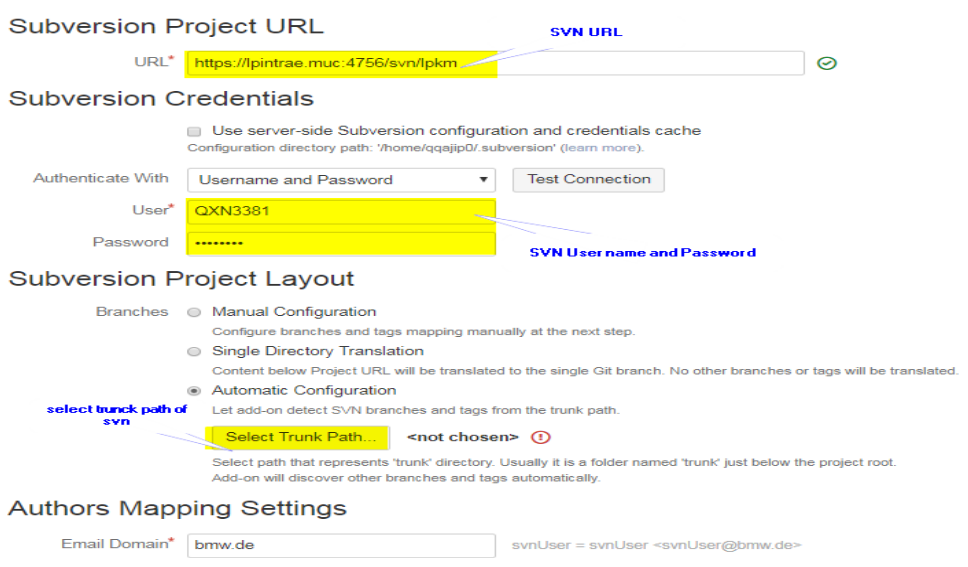
My code is ready to be pushed

If you already have code ready to be pushed to this repository then run this in your terminal.

```
cd existing-project
git init
git add --all
git commit -m "Initial Commit"
git remote add origin https://qxm2605@atc.bmwgroup.net/bitbucket/scm/lpkm/gitmigrationlpkm.git
git push -u origin master
```

6. Provide the SVN URL, SVN username, password, email domain and select trunk path of SVN.

   [Password need to be updated for the corresponding user whenever there is change in password]


Subversion Project URL

URL*: https://lpintrae.muc:4756/svn/lpkm

Subversion Credentials

☐ Use server-side Subversion configuration and credentials cache
Configuration directory path: '/home/qqajip0/.subversion' (learn more).

Authenticate With: Username and Password

User*: QXN3381

Password: ••••••••

Subversion Project Layout

Branches:
○ Manual Configuration
Configure branches and tags mapping manually at the next step.
○ Single Directory Translation
Content below Project URL will be translated to the single Git branch. No other branches or tags will be translated.
● Automatic Configuration
Let add-on detect SVN branches and tags from the trunk path.

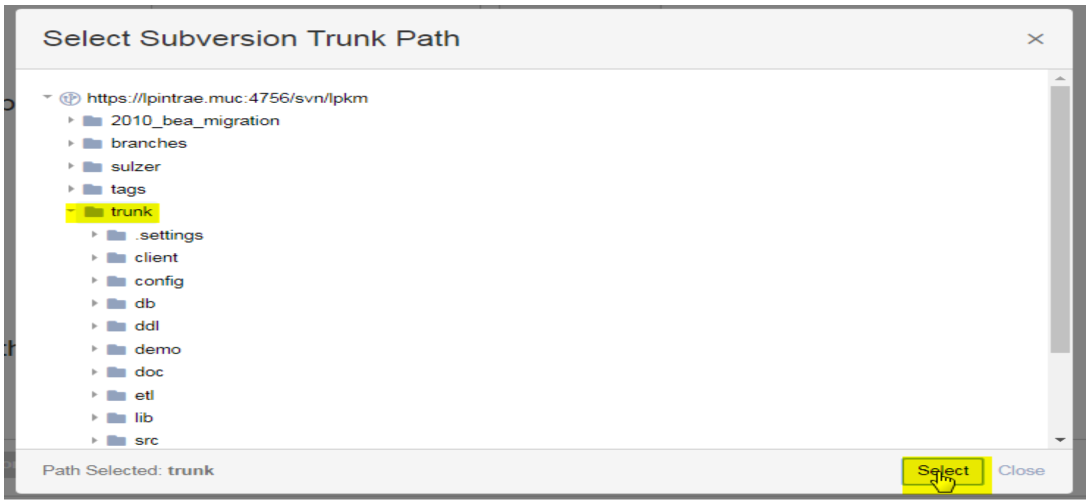
<not chosen>
Select path that represents 'trunk' directory. Usually it is a folder named 'trunk' just below the project root.
Add-on will discover other branches and tags automatically.

Authors Mapping Settings

Email Domain*: bmw.de

svnUser = svnUser <svnUser@bmw.de>

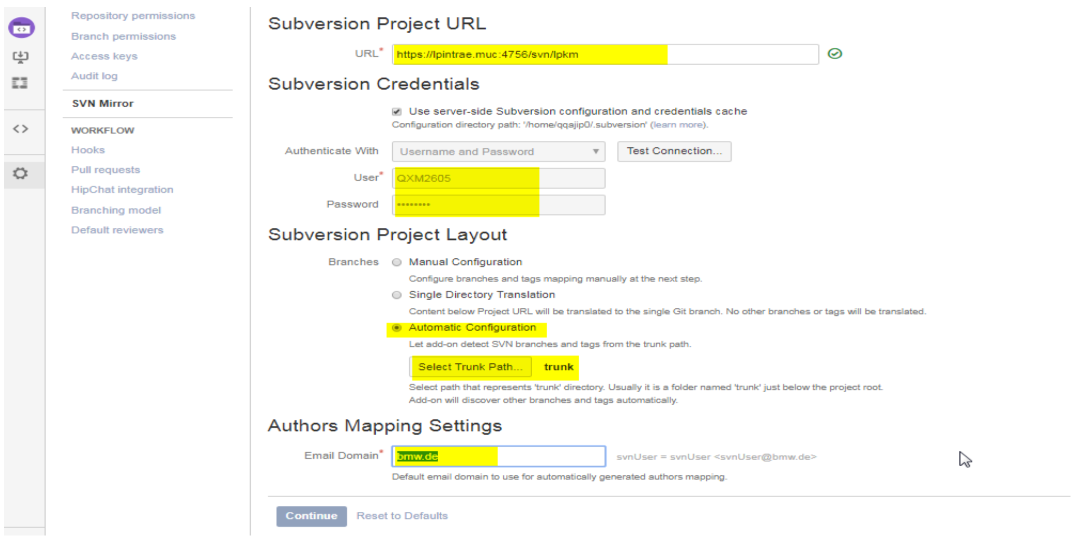
7. Selecting the trunk path of SVN as shown below,


Select Subversion Trunk Path

```
https://lpintrae.muc:4756/svn/lpkm
  2010_bea_migration
  branches
  sulzer
  tags
  trunk
    .settings
    client
    config
    db
    ddl
    demo
    doc
    etl
    lib
    src
```

Path Selected: **trunk**

8. Click on continue button



Repository permissions
Branch permissions
Access keys
Audit log
SVN Mirror
WORKFLOW
Hooks
Pull requests
HipChat integration
Branching model
Default reviewers

Subversion Project URL

URL* https://lpintrae.muc:4756/svn/lpkm

Subversion Credentials

Use server-side Subversion configuration and credentials cache
Configuration directory path: '/home/qqajip0/.subversion' (learn more).

Authenticate With: Username and Password | Test Connection...
User*: QXM2605
Password: ••••••••

Subversion Project Layout

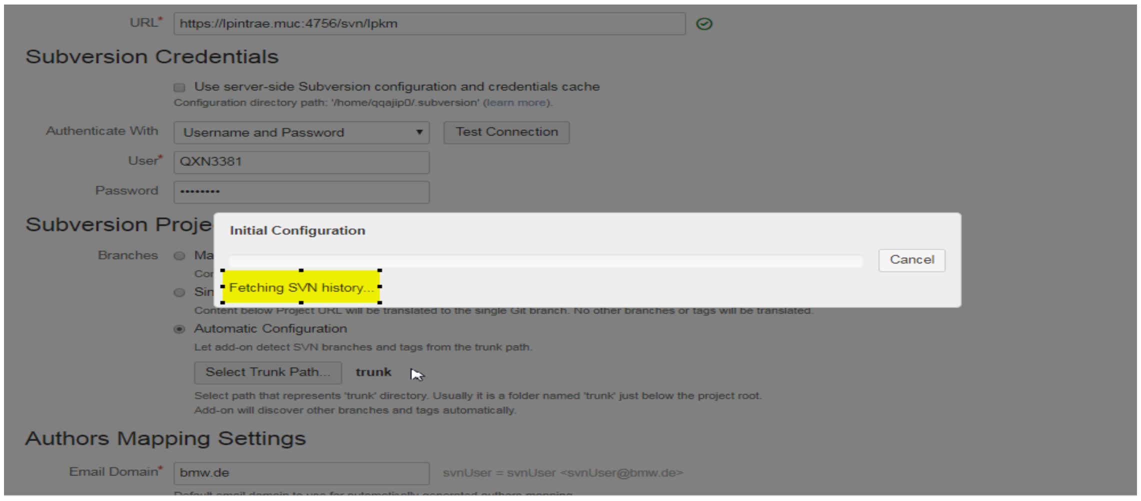
Branches
- Manual Configuration
  Configure branches and tags mapping manually at the next step.
- Single Directory Translation
  Content below Project URL will be translated to the single Git branch. No other branches or tags will be translated.
- [✓ Automatic Configuration]
  Let add-on detect SVN branches and tags from the trunk path.
  Select Trunk Path... **trunk**
  Select path that represents 'trunk' directory. Usually it is a folder named 'trunk' just below the project root.
  Add-on will discover other branches and tags automatically.

Authors Mapping Settings

Email Domain*: bmw.de   svnUser = svnUser <svnUser@bmw.de>
Default email domain to use for automatically generated authors mapping.

Continue | Reset to Defaults

9. It connects to SVN and below messages appears once configuration started.



URL* https://lpintrae.muc:4756/svn/lpkm

Subversion Credentials

Use server-side Subversion configuration and credentials cache
Configuration directory path: '/home/qqajip0/.subversion' (learn more).

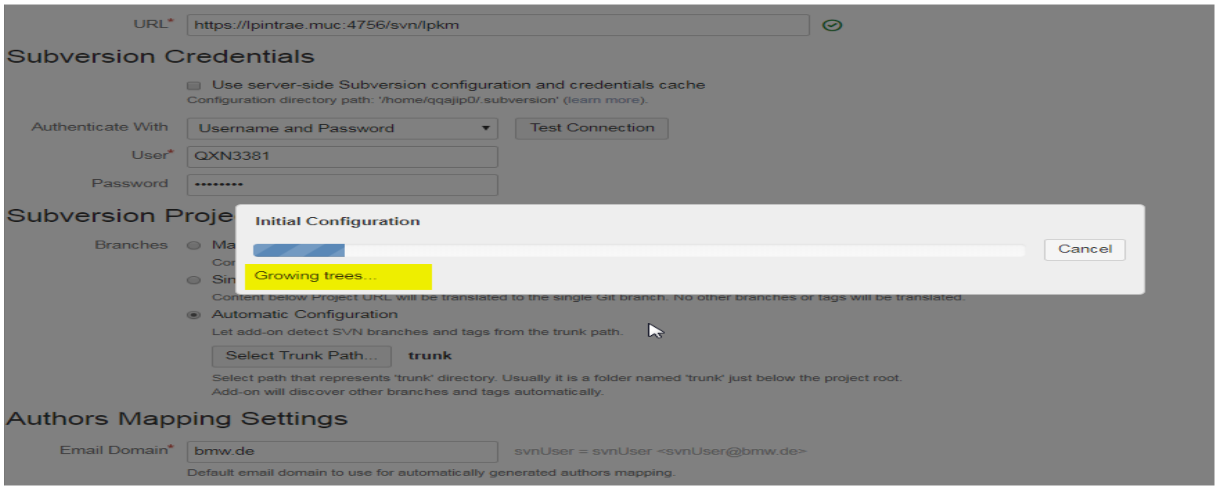
Authenticate With: Username and Password | Test Connection
User*: QXN3381
Password: ••••••••

Initial Configuration
Fetching SVN history...
Cancel

Content below Project URL will be translated to the single Git branch. No other branches or tags will be translated.
[✓ Automatic Configuration]
Let add-on detect SVN branches and tags from the trunk path.
Select Trunk Path... **trunk**
Select path that represents 'trunk' directory. Usually it is a folder named 'trunk' just below the project root.
Add-on will discover other branches and tags automatically.

Authors Mapping Settings

Email Domain*: bmw.de   svnUser = svnUser <svnUser@bmw.de>



URL* https://lpintrae.muc:4756/svn/lpkm

Subversion Credentials

Use server-side Subversion configuration and credentials cache
Configuration directory path: '/home/qqajip0/.subversion' (learn more).

Authenticate With: Username and Password — Test Connection
User*: QXN3381
Password: ••••••••

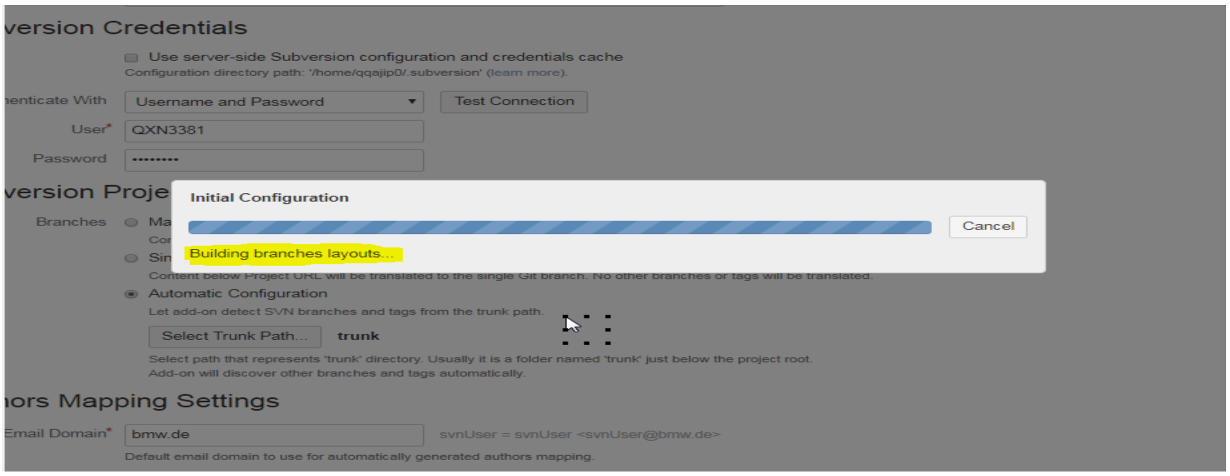
Subversion Proje

Branches: Ma / Cor / Sin
Content below Project URL will be translated to the single Git branch. No other branches or tags will be translated.
Automatic Configuration
Let add-on detect SVN branches and tags from the trunk path.
Select Trunk Path... trunk
Select path that represents 'trunk' directory. Usually it is a folder named 'trunk' just below the project root.
Add-on will discover other branches and tags automatically.

Initial Configuration
Cancel
Growing trees...

Authors Mapping Settings

Email Domain*: bmw.de — svnUser = svnUser <svnUser@bmw.de>
Default email domain to use for automatically generated authors mapping.



version Credentials

Use server-side Subversion configuration and credentials cache
Configuration directory path: '/home/qqajip0/.subversion' (learn more).

henticate With: Username and Password — Test Connection
User*: QXN3381
Password: ••••••••

version Proje

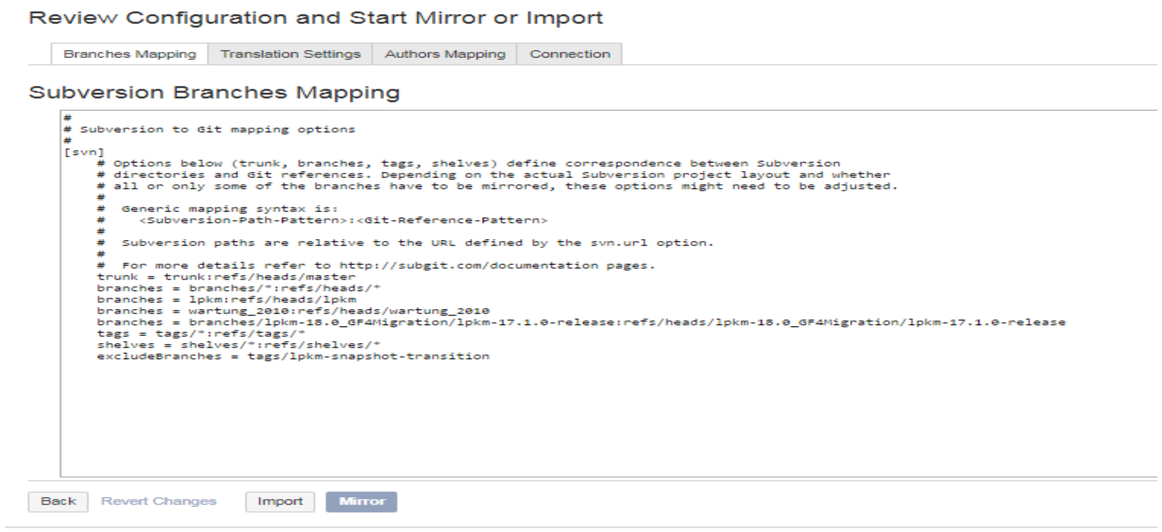
Branches: Ma / Cor / Sin
Content below Project URL will be translated to the single Git branch. No other branches or tags will be translated.
Automatic Configuration
Let add-on detect SVN branches and tags from the trunk path.
Select Trunk Path... trunk
Select path that represents 'trunk' directory. Usually it is a folder named 'trunk' just below the project root.
Add-on will discover other branches and tags automatically.

Initial Configuration
Cancel
Building branches layouts...

hors Mapping Settings

Email Domain*: bmw.de — svnUser = svnUser <svnUser@bmw.de>
Default email domain to use for automatically generated authors mapping.

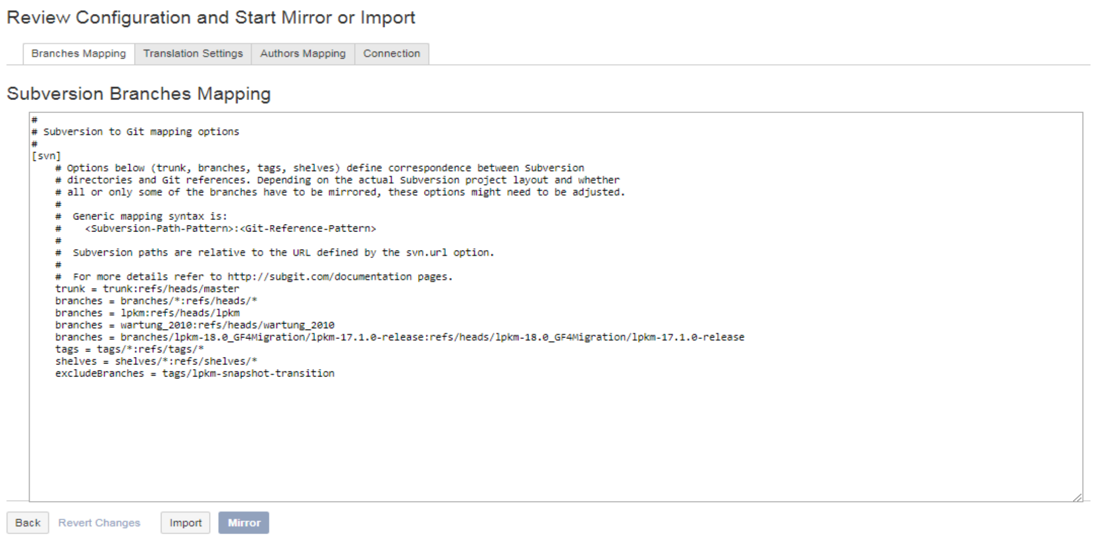
10. Once the configuration process is completed, the below screen appears



```
#
# Subversion to Git mapping options
#
[svn]
    # Options below (trunk, branches, tags, shelves) define correspondence between Subversion
    # directories and Git references. Depending on the actual Subversion project layout and whether
    # all or only some of the branches have to be mirrored, these options might need to be adjusted.
    #
    #  Generic mapping syntax is:
    #    <Subversion-Path-Pattern>:<Git-Reference-Pattern>
    #
    #  Subversion paths are relative to the URL defined by the svn.url option.
    #
    #  For more details refer to http://subgit.com/documentation pages.
    trunk = trunk:refs/heads/master
    branches = branches/*:refs/heads/*
    branches = lpkm:refs/heads/lpkm
    branches = wartung_2010:refs/heads/wartung_2010
    branches = branches/lpkm-18.0_GF4Migration/lpkm-17.1.0-release:refs/heads/lpkm-18.0_GF4Migration/lpkm-17.1.0-release
    tags = tags/*:refs/tags/*
    shelves = shelves/*:refs/shelves/*
    excludeBranches = tags/lpkm-snapshot-transition
```

11. Now adjust the Branch Mapping, Authors Mapping and translation modifications are required as mappings are automatically picked up.

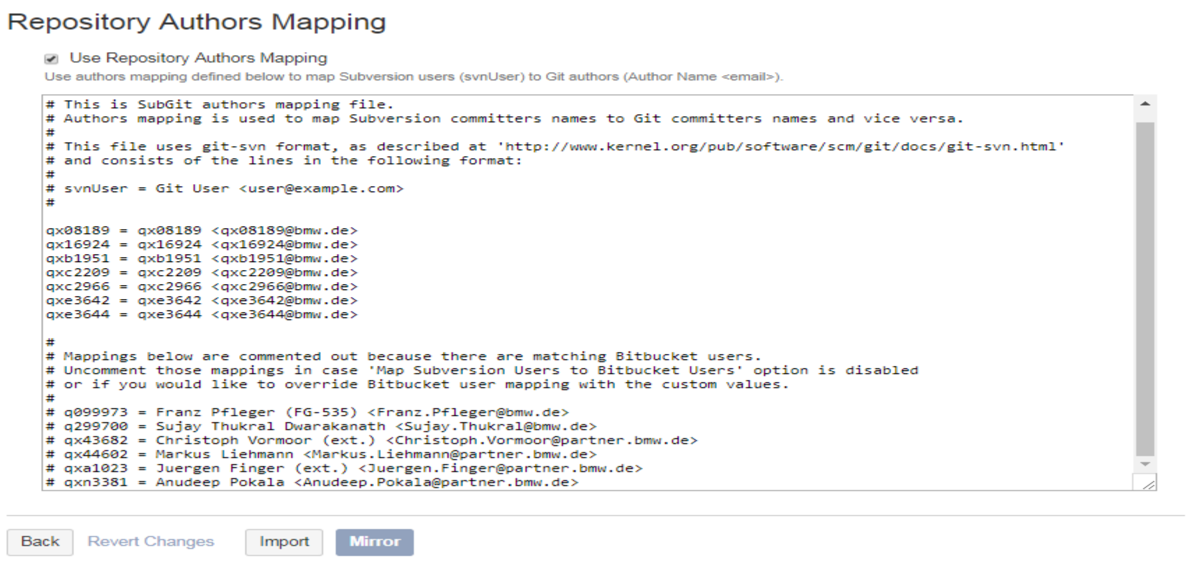
- **Branch Mapping**



```
#
# Subversion to Git mapping options
#
[svn]
    # Options below (trunk, branches, tags, shelves) define correspondence between Subversion
    # directories and Git references. Depending on the actual Subversion project layout and whether
    # all or only some of the branches have to be mirrored, these options might need to be adjusted.
    #
    #  Generic mapping syntax is:
    #    <Subversion-Path-Pattern>:<Git-Reference-Pattern>
    #
    #  Subversion paths are relative to the URL defined by the svn.url option.
    #
    #  For more details refer to http://subgit.com/documentation pages.
    trunk = trunk:refs/heads/master
    branches = branches/*:refs/heads/*
    branches = lpkm:refs/heads/lpkm
    branches = wartung_2010:refs/heads/wartung_2010
    branches = branches/lpkm-18.0_GF4Migration/lpkm-17.1.0-release:refs/heads/lpkm-18.0_GF4Migration/lpkm-17.1.0-release
    tags = tags/*:refs/tags/*
    shelves = shelves/*:refs/shelves/*
    excludeBranches = tags/lpkm-snapshot-transition
```

- **Authors Mapping**

  Before Adjusting the Authors Mapping, screen appears as below



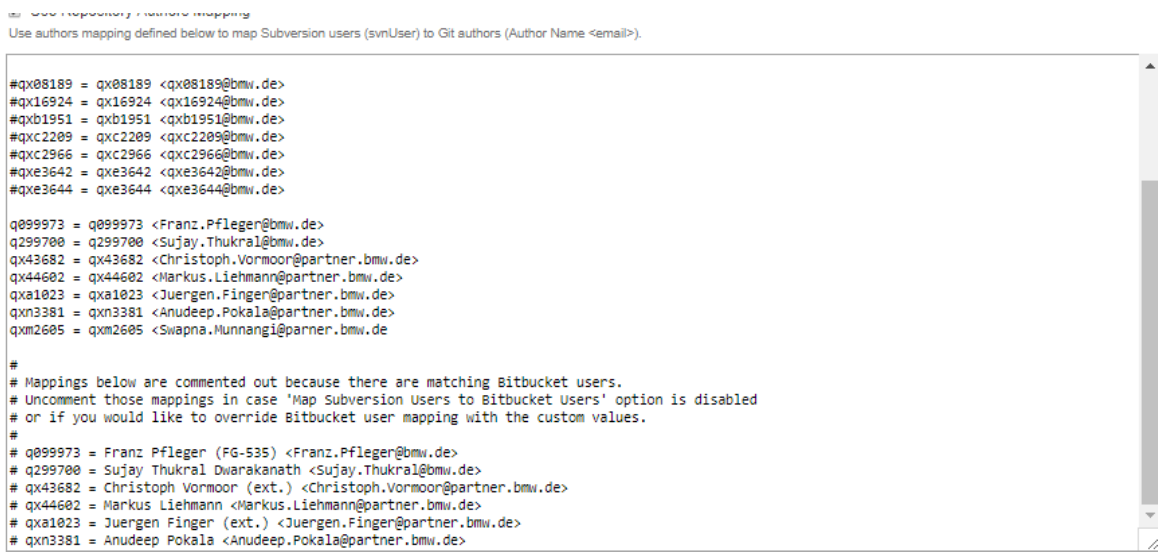
Repository Authors Mapping

☑ Use Repository Authors Mapping

Use authors mapping defined below to map Subversion users (svnUser) to Git authors (Author Name <email>).

```
# This is SubGit authors mapping file.
# Authors mapping is used to map Subversion committers names to Git committers names and vice versa.
#
# This file uses git-svn format, as described at 'http://www.kernel.org/pub/software/scm/git/docs/git-svn.html'
# and consists of the lines in the following format:
#
# svnUser = Git User <user@example.com>
#

qx08189 = qx08189 <qx08189@bmw.de>
qx16924 = qx16924 <qx16924@bmw.de>
qxb1951 = qxb1951 <qxb1951@bmw.de>
qxc2209 = qxc2209 <qxc2209@bmw.de>
qxc2966 = qxc2966 <qxc2966@bmw.de>
qxe3642 = qxe3642 <qxe3642@bmw.de>
qxe3644 = qxe3644 <qxe3644@bmw.de>

#
# Mappings below are commented out because there are matching Bitbucket users.
# Uncomment those mappings in case 'Map Subversion Users to Bitbucket Users' option is disabled
# or if you would like to override Bitbucket user mapping with the custom values.
#
# q099973 = Franz Pfleger (FG-535) <Franz.Pfleger@bmw.de>
# q299700 = Sujay Thukral Dwarakanath <Sujay.Thukral@bmw.de>
# qx43682 = Christoph Vormoor (ext.) <Christoph.Vormoor@partner.bmw.de>
# qx44602 = Markus Liehmann <Markus.Liehmann@partner.bmw.de>
# qxa1023 = Juergen Finger (ext.) <Juergen.Finger@partner.bmw.de>
# qxn3381 = Anudeep Pokala <Anudeep.Pokala@partner.bmw.de>
```

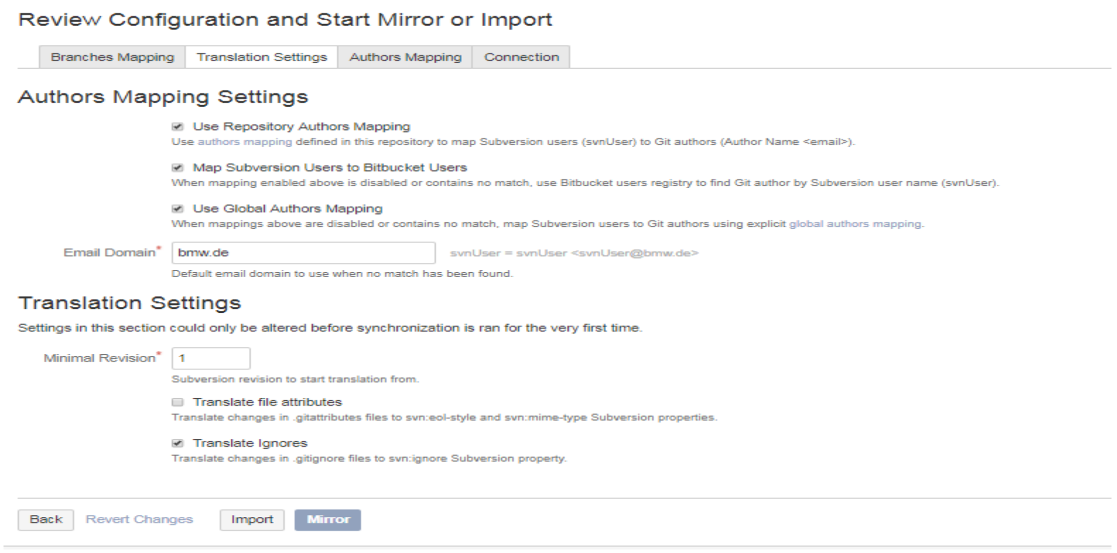
After adjusting the Authors Mapping, screen appears as below



Use authors mapping defined below to map Subversion users (svnUser) to Git authors (Author Name <email>).

```

#qx08189 = qx08189 <qx08189@bmw.de>
#qx16924 = qx16924 <qx16924@bmw.de>
#qxb1951 = qxb1951 <qxb1951@bmw.de>
#qxc2209 = qxc2209 <qxc2209@bmw.de>
#qxc2966 = qxc2966 <qxc2966@bmw.de>
#qxe3642 = qxe3642 <qxe3642@bmw.de>
#qxe3644 = qxe3644 <qxe3644@bmw.de>

q099973 = q099973 <Franz.Pfleger@bmw.de>
q299700 = q299700 <Sujay.Thukral@bmw.de>
qx43682 = qx43682 <Christoph.Vormoor@partner.bmw.de>
qx44602 = qx44602 <Markus.Liehmann@partner.bmw.de>
qxa1023 = qxa1023 <Juergen.Finger@partner.bmw.de>
qxn3381 = qxn3381 <Anudeep.Pokala@partner.bmw.de>
qxm2605 = qxm2605 <Swapna.Munnangi@parner.bmw.de

#
# Mappings below are commented out because there are matching Bitbucket users.
# Uncomment those mappings in case 'Map Subversion Users to Bitbucket Users' option is disabled
# or if you would like to override Bitbucket user mapping with the custom values.
#
# q099973 = Franz Pfleger (FG-535) <Franz.Pfleger@bmw.de>
# q299700 = Sujay Thukral Dwarakanath <Sujay.Thukral@bmw.de>
# qx43682 = Christoph Vormoor (ext.) <Christoph.Vormoor@partner.bmw.de>
# qx44602 = Markus Liehmann <Markus.Liehmann@partner.bmw.de>
# qxa1023 = Juergen Finger (ext.) <Juergen.Finger@partner.bmw.de>
# qxn3381 = Anudeep Pokala <Anudeep.Pokala@partner.bmw.de>
```

- **Translation Settings**



Review Configuration and Start Mirror or Import

Authors Mapping Settings

☑ Use Repository Authors Mapping
Use authors mapping defined in this repository to map Subversion users (svnUser) to Git authors (Author Name <email>).

☑ Map Subversion Users to Bitbucket Users
When mapping enabled above is disabled or contains no match, use Bitbucket users registry to find Git author by Subversion user name (svnUser).

☑ Use Global Authors Mapping
When mappings above are disabled or contains no match, map Subversion users to Git authors using explicit global authors mapping.

Email Domain*: bmw.de   svnUser = svnUser <svnUser@bmw.de>
Default email domain to use when no match has been found.

Translation Settings

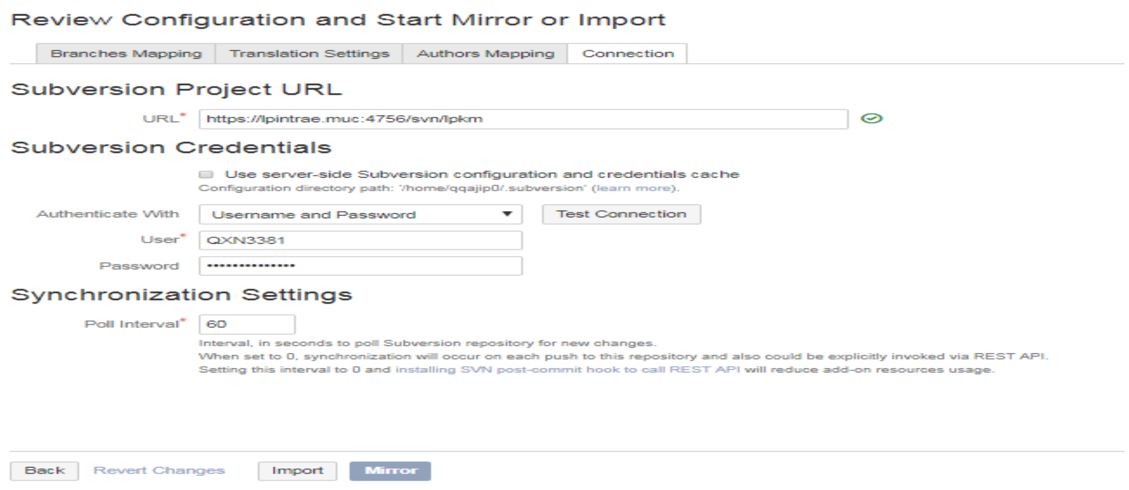
Settings in this section could only be altered before synchronization is ran for the very first time.

Minimal Revision*: 1
Subversion revision to start translation from.

☐ Translate file attributes
Translate changes in .gitattributes files to svn:eol-style and svn:mime-type Subversion properties.

☑ Translate Ignores
Translate changes in .gitignore files to svn:ignore Subversion property.

- **Connection Settings**



Review Configuration and Start Mirror or Import

Subversion Project URL

URL*: https://lpintrae.muc:4756/svn/lpkm

Subversion Credentials

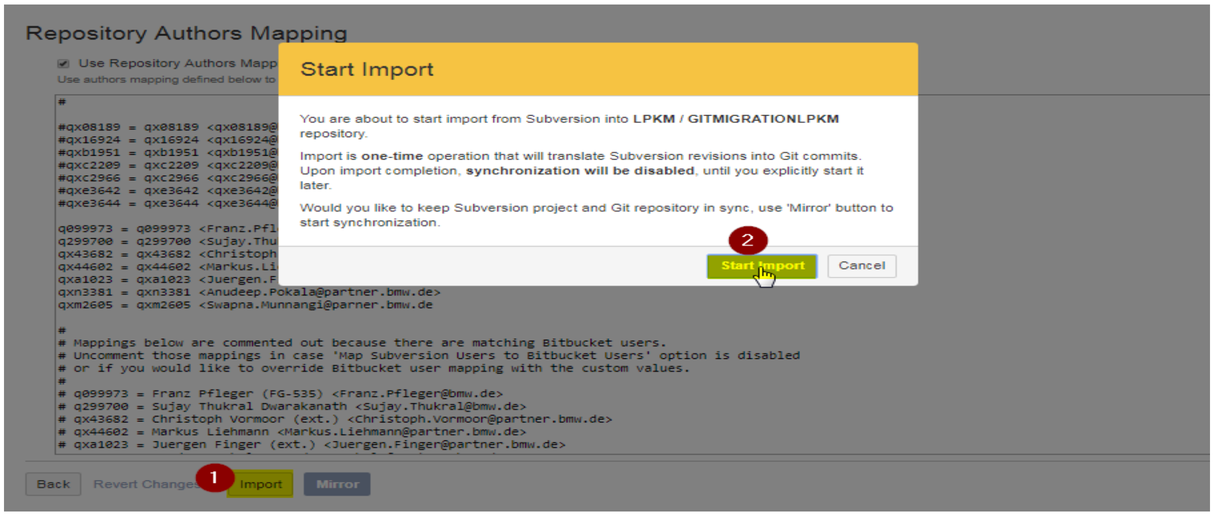
☐ Use server-side Subversion configuration and credentials cache
Configuration directory path: '/home/qqajip0/.subversion' (learn more).

Authenticate With: Username and Password
User*: QXN3381
Password: •••••••••••••••

Synchronization Settings

Poll Interval*: 60
Interval, in seconds to poll Subversion repository for new changes.
When set to 0, synchronization will occur on each push to this repository and also could be explicitly invoked via REST API.
Setting this interval to 0 and installing SVN post-commit hook to call REST API will reduce add-on resources usage.

12. Finally click on ‘Import’ button and then click on ‘Start import’ button to import the code from SVN repository to Git



Repository Authors Mapping

☑ Use Repository Authors Mapp

Use authors mapping defined below to

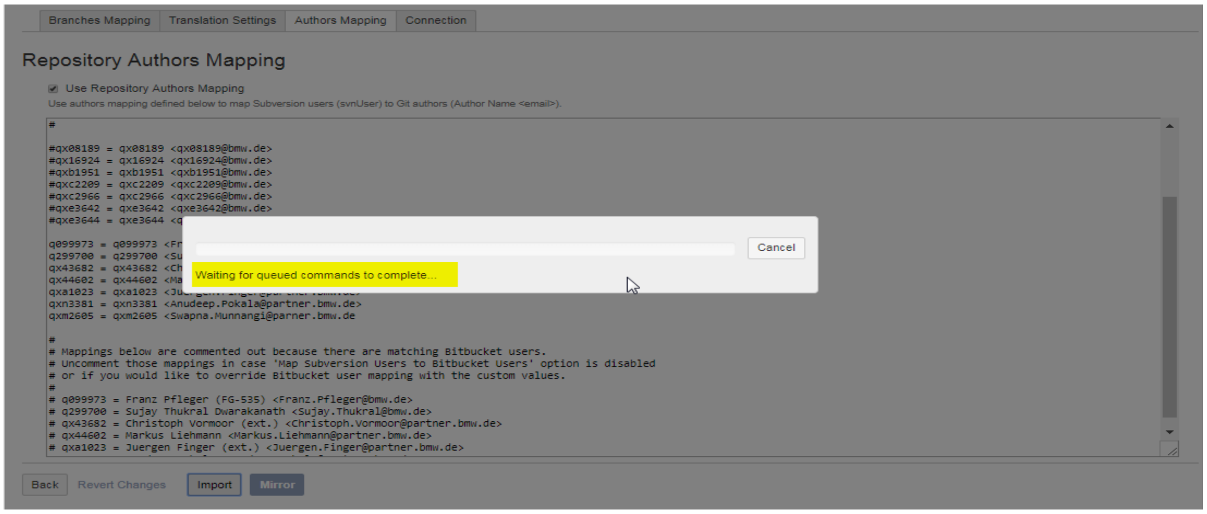
```
#

#qx08189 = qx08189 <qx08189@
#qx16924 = qx16924 <qx16924@
#qxb1951 = qxb1951 <qxb1951@
#qxc2209 = qxc2209 <qxc2209@
#qxc2966 = qxc2966 <qxc2966@
#qxe3642 = qxe3642 <qxe3642@
#qxe3644 = qxe3644 <qxe3644@

q099973 = q099973 <Franz.Pfl
q299700 = q299700 <Sujay.Thu
qx43682 = qx43682 <Christoph
qx44602 = qx44602 <Markus.Li
qxa1023 = qxa1023 <Juergen.F
qxn3381 = qxn3381 <Anudeep.Pokala@partner.bmw.de>
qxm2605 = qxm2605 <Swapna.Munnangi@parner.bmw.de

#
# Mappings below are commented out because there are matching Bitbucket users.
# Uncomment those mappings in case 'Map Subversion Users to Bitbucket Users' option is disabled
# or if you would like to override Bitbucket user mapping with the custom values.
#
# q099973 = Franz Pfleger (FG-535) <Franz.Pfleger@bmw.de>
# q299700 = Sujay Thukral Dwarakanath <Sujay.Thukral@bmw.de>
# qx43682 = Christoph Vormoor (ext.) <Christoph.Vormoor@partner.bmw.de>
# qx44602 = Markus Liehmann <Markus.Liehmann@partner.bmw.de>
# qxa1023 = Juergen Finger (ext.) <Juergen.Finger@partner.bmw.de>
```

Start Import

You are about to start import from Subversion into **LPKM / GITMIGRATIONLPKM** repository.

Import is **one-time** operation that will translate Subversion revisions into Git commits. Upon import completion, **synchronization will be disabled**, until you explicitly start it later.

Would you like to keep Subversion project and Git repository in sync, use 'Mirror' button to start synchronization.



Repository Authors Mapping

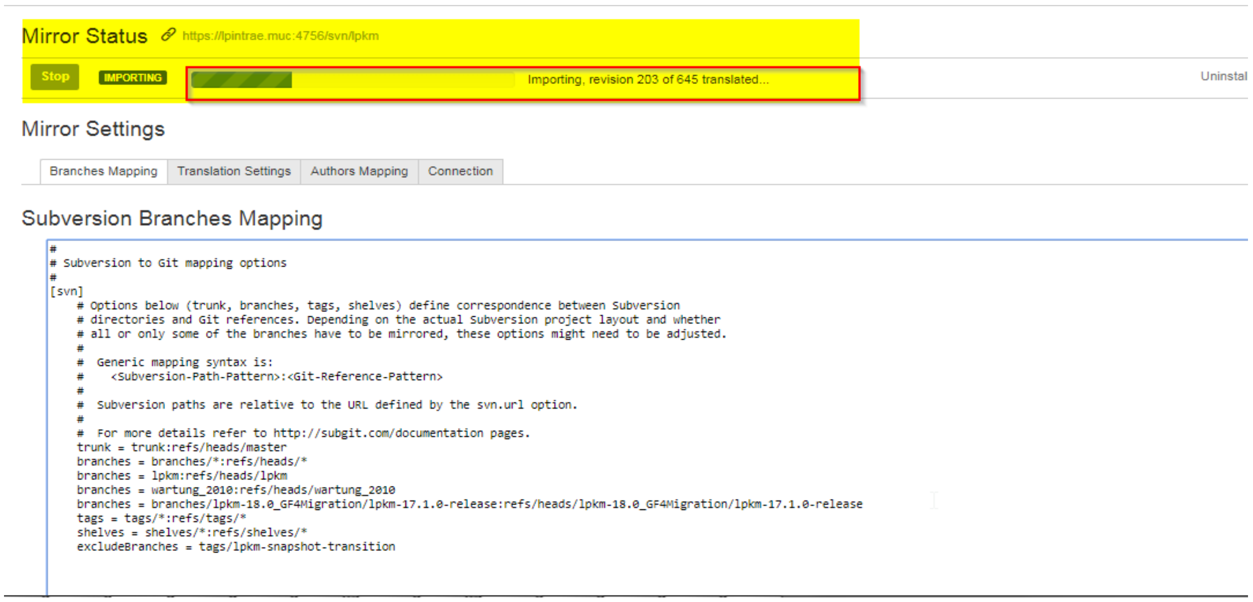
☑ Use Repository Authors Mapping

Use authors mapping defined below to map Subversion users (svnUser) to Git authors (Author Name <email>).

```
#

#qx08189 = qx08189 <qx08189@bmw.de>
#qx16924 = qx16924 <qx16924@bmw.de>
#qxb1951 = qxb1951 <qxb1951@bmw.de>
#qxc2209 = qxc2209 <qxc2209@bmw.de>
#qxc2966 = qxc2966 <qxc2966@bmw.de>
#qxe3642 = qxe3642 <qxe3642@bmw.de>
#qxe3644 = qxe3644 <q

q099973 = q099973 <Fr
q299700 = q299700 <Su
qx43682 = qx43682 <Ch
qx44602 = qx44602 <Ma
qxa1023 = qxa1023 <Ju
qxn3381 = qxn3381 <Anudeep.Pokala@partner.bmw.de>
qxm2605 = qxm2605 <Swapna.Munnangi@parner.bmw.de

#
# Mappings below are commented out because there are matching Bitbucket users.
# Uncomment those mappings in case 'Map Subversion Users to Bitbucket Users' option is disabled
# or if you would like to override Bitbucket user mapping with the custom values.
#
# q099973 = Franz Pfleger (FG-535) <Franz.Pfleger@bmw.de>
# q299700 = Sujay Thukral Dwarakanath <Sujay.Thukral@bmw.de>
# qx43682 = Christoph Vormoor (ext.) <Christoph.Vormoor@partner.bmw.de>
# qx44602 = Markus Liehmann <Markus.Liehmann@partner.bmw.de>
# qxa1023 = Juergen Finger (ext.) <Juergen.Finger@partner.bmw.de>
```

Waiting for queued commands to complete...

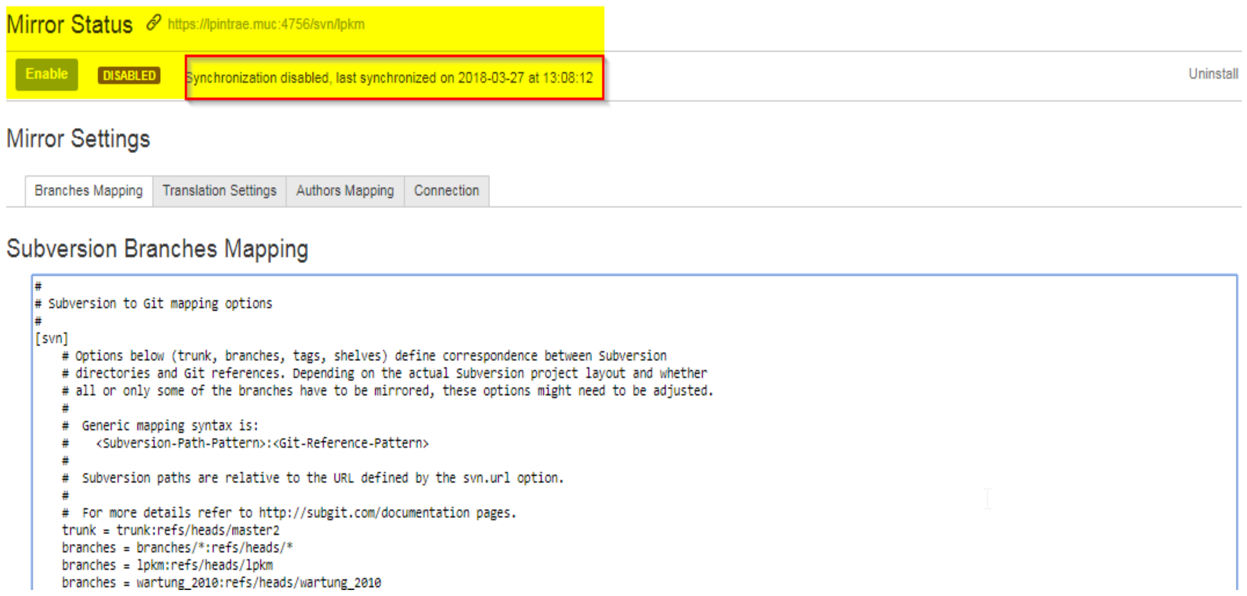
13. Importing process is visible on the top of the Mirror settings window.



Importing, revision 203 of 645 translated...

```
#
# Subversion to Git mapping options
#
[svn]
    # Options below (trunk, branches, tags, shelves) define correspondence between Subversion
    # directories and Git references. Depending on the actual Subversion project layout and whether
    # all or only some of the branches have to be mirrored, these options might need to be adjusted.
    #
    #  Generic mapping syntax is:
    #    <Subversion-Path-Pattern>:<Git-Reference-Pattern>
    #
    #  Subversion paths are relative to the URL defined by the svn.url option.
    #
    #  For more details refer to http://subgit.com/documentation pages.
    trunk = trunk:refs/heads/master
    branches = branches/*:refs/heads/*
    branches = lpkm:refs/heads/lpkm
    branches = wartung_2010:refs/heads/wartung_2010
    branches = branches/lpkm-18.0_GF4Migration/lpkm-17.1.0-release:refs/heads/lpkm-18.0_GF4Migration/lpkm-17.1.0-release
    tags = tags/*:refs/tags/*
    shelves = shelves/*:refs/shelves/*
    excludeBranches = tags/lpkm-snapshot-transition
```

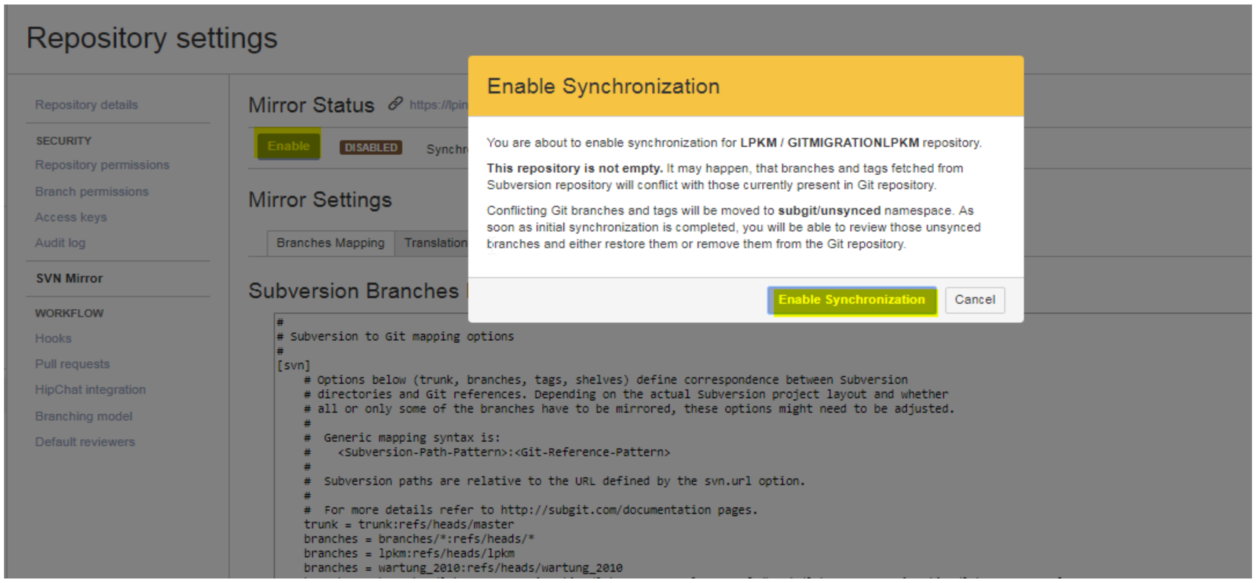
14. Once the import is done, ENABLE button is viewed on left top corner.



Synchronization disabled, last synchronized on 2018-03-27 at 13:08:12

```
#
# Subversion to Git mapping options
#
[svn]
    # Options below (trunk, branches, tags, shelves) define correspondence between Subversion
    # directories and Git references. Depending on the actual Subversion project layout and whether
    # all or only some of the branches have to be mirrored, these options might need to be adjusted.
    #
    #  Generic mapping syntax is:
    #    <Subversion-Path-Pattern>:<Git-Reference-Pattern>
    #
    #  Subversion paths are relative to the URL defined by the svn.url option.
    #
    #  For more details refer to http://subgit.com/documentation pages.
    trunk = trunk:refs/heads/master2
    branches = branches/*:refs/heads/*
    branches = lpkm:refs/heads/lpkm
    branches = wartung_2010:refs/heads/wartung_2010
```

15. Click on ‘Enable’ button and then click on ‘Enable Synchronization’.



Enable Synchronization

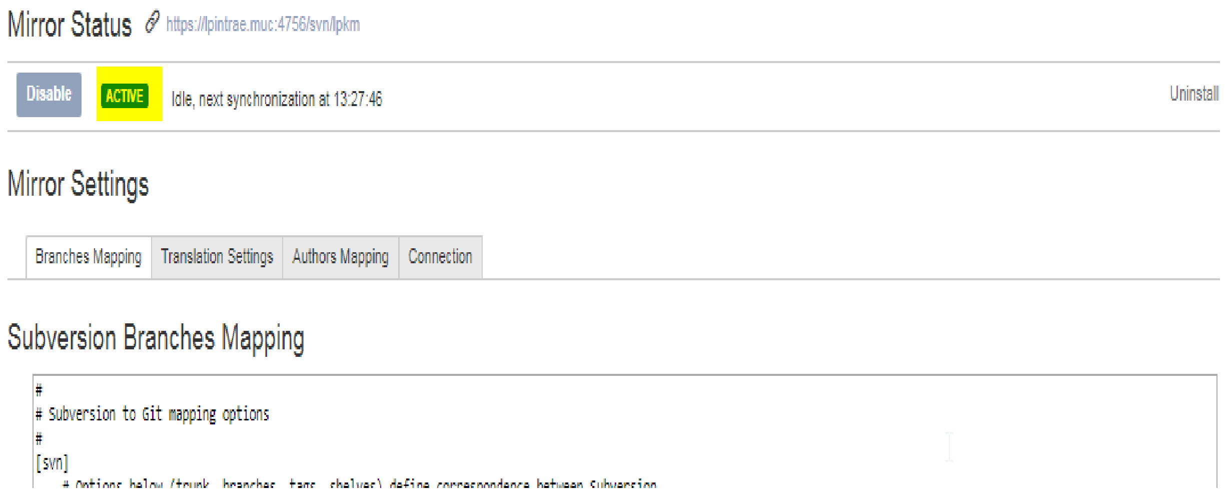
You are about to enable synchronization for **LPKM / GITMIGRATIONLPKM** repository.

This repository is not empty. It may happen, that branches and tags fetched from Subversion repository will conflict with those currently present in Git repository.

Conflicting Git branches and tags will be moved to **subgit/unsynced** namespace. As soon as initial synchronization is completed, you will be able to review those unsynced branches and either restore them or remove them from the Git repository.

```
#
# Subversion to Git mapping options
#
[svn]
    # Options below (trunk, branches, tags, shelves) define correspondence between Subversion
    # directories and Git references. Depending on the actual Subversion project layout and whether
    # all or only some of the branches have to be mirrored, these options might need to be adjusted.
    #
    #  Generic mapping syntax is:
    #    <Subversion-Path-Pattern>:<Git-Reference-Pattern>
    #
    #  Subversion paths are relative to the URL defined by the svn.url option.
    #
    #  For more details refer to http://subgit.com/documentation pages.
    trunk = trunk:refs/heads/master
    branches = branches/*:refs/heads/*
    branches = lpkm:refs/heads/lpkm
    branches = wartung_2010:refs/heads/wartung_2010
```

16. The progress bar starts and once the process is completed, the below screen appears. If there are any conflicts, then ‘UNSYNCED’ button appears next to ‘ACTIVE’ button.

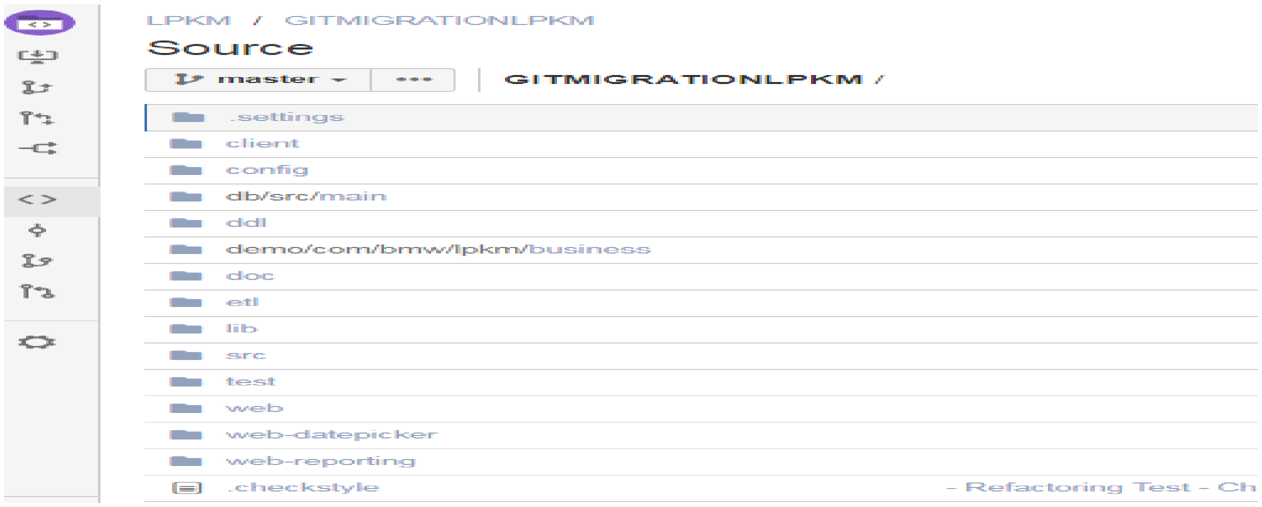


Idle, next synchronization at 13:27:46

```
#
# Subversion to Git mapping options
#
[svn]
```

17. Now we have the Git repository with the contents of SVN repository

Master [Master is trunk in SVN]

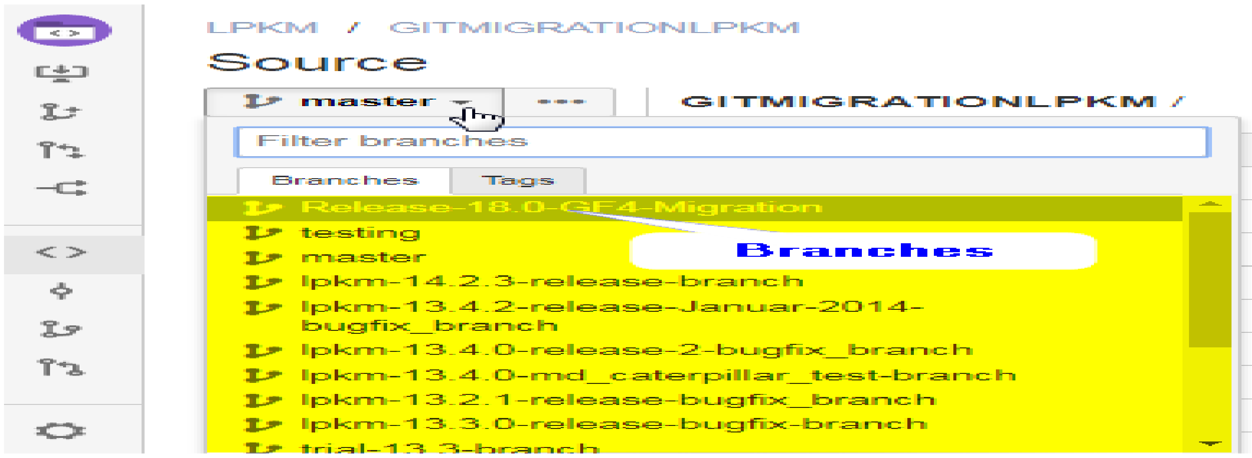

LPKM / GITMIGRATIONLPKM

Source

master ▾ ··· | GITMIGRATIONLPKM /

.settings
client
config
db/src/main
ddl
demo/com/bmw/lpkm/business
doc
etl
lib
src
test
web
web-datepicker
web-reporting
.checkstyle - Refactoring Test - Ch

Branches


LPKM / GITMIGRATIONLPKM

Source

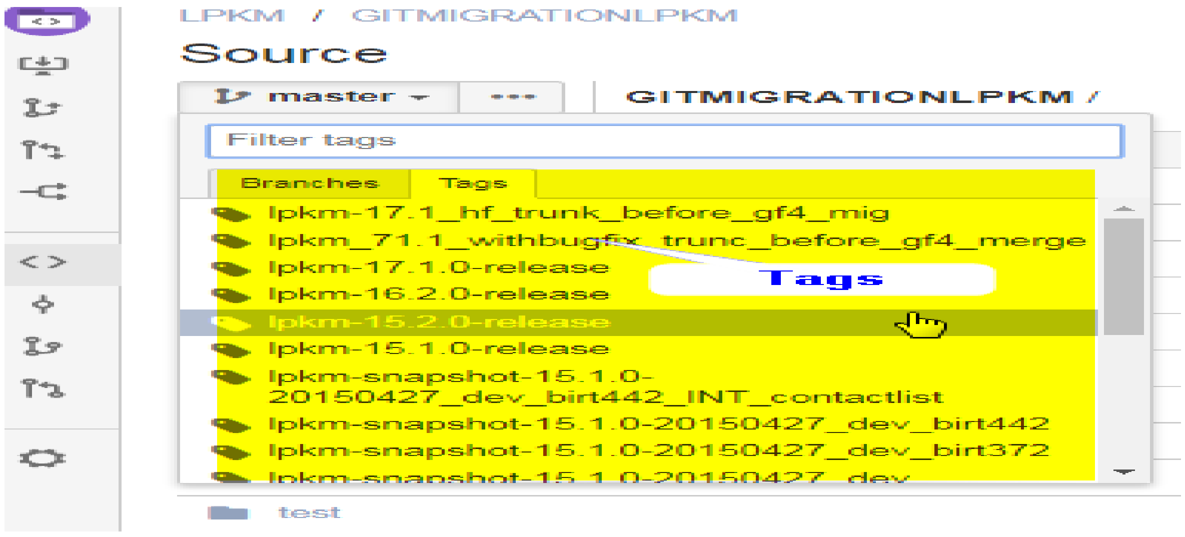
master ▾ ··· | GITMIGRATIONLPKM /

Filter branches

Branches | Tags

Release-18.0-GF4-Migration
testing
master
lpkm-14.2.3-release-branch
lpkm-13.4.2-release-Januar-2014-bugfix_branch
lpkm-13.4.0-release-2-bugfix_branch
lpkm-13.4.0-md_caterpillar_test-branch
lpkm-13.2.1-release-bugfix_branch
lpkm-13.3.0-release-bugfix-branch
trial-13.3-branch

Tags



lpkm-17.1_hf_trunk_before_gf4_mig
lpkm_71.1_withbugfix_trunc_before_gf4_merge
lpkm-17.1.0-release
lpkm-16.2.0-release
lpkm-15.2.0-release
lpkm-15.1.0-release
lpkm-snapshot-15.1.0-20150427_dev_birt442_INT_contactlist
lpkm-snapshot-15.1.0-20150427_dev_birt442
lpkm-snapshot-15.1.0-20150427_dev_birt372
lpkm-snapshot-15.1.0-20150427_dev

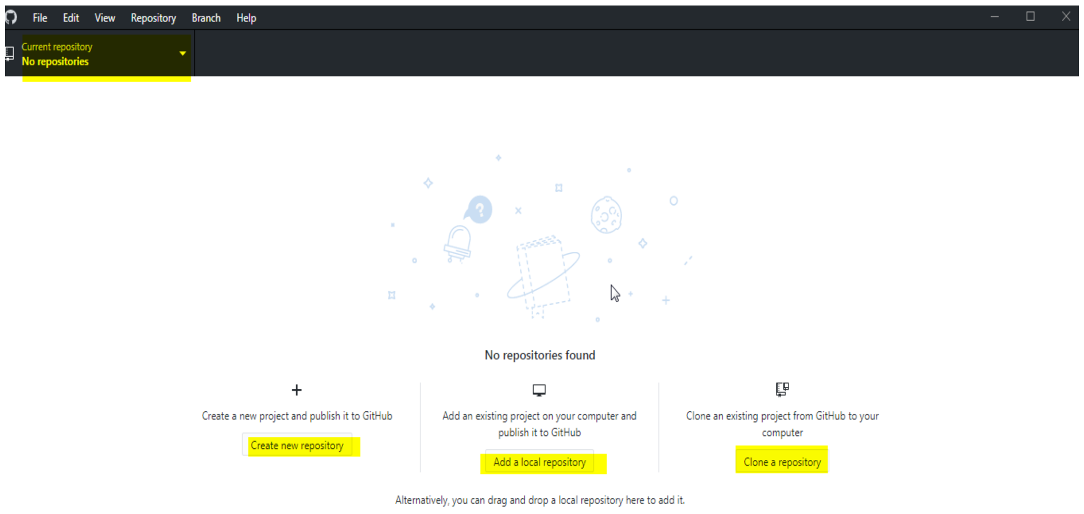
3. Operations using GitHub Desktop

GitHub Desktop Installation

Install GitHub Desktop using the below URL,

URL : https://desktop.Github.com/

Initial screen of the GitHub Desktop appears as in below screenshot,



No repositories found

Create a new project and publish it to GitHub

Add an existing project on your computer and publish it to GitHub

Clone an existing project from GitHub to your computer

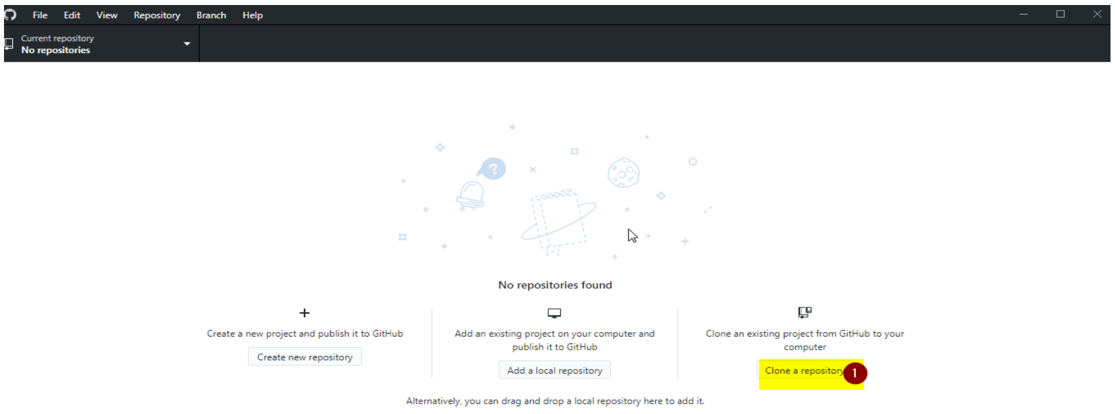
Alternatively, you can drag and drop a local repository here to add it.

3.1 Clone

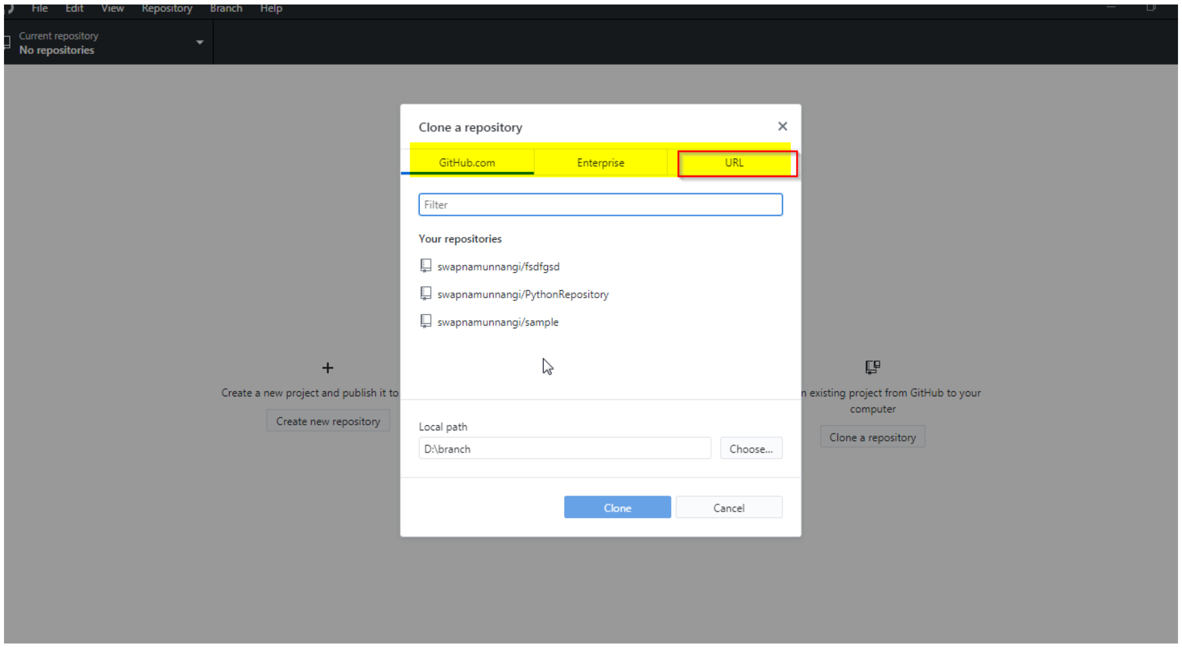
- Current repository → Display the Git repository name
- Current Branch → Display the list of master & other Branches in Git repository
- Post cloning, folder name of the cloned Master or Branch appears as Git repository name by default [For example, 'gitmigrationlpkm']
- Once we clone the repository using Git repository URL, all the Branches including Master will be cloned locally.
- Still the Current Branch points to 'Master' initially.
- But based on our selection, we can switch to required Branch (Refer the contents in the cloned repository folder by switching to Master or a Branch)]

a) Cloning of a Master:

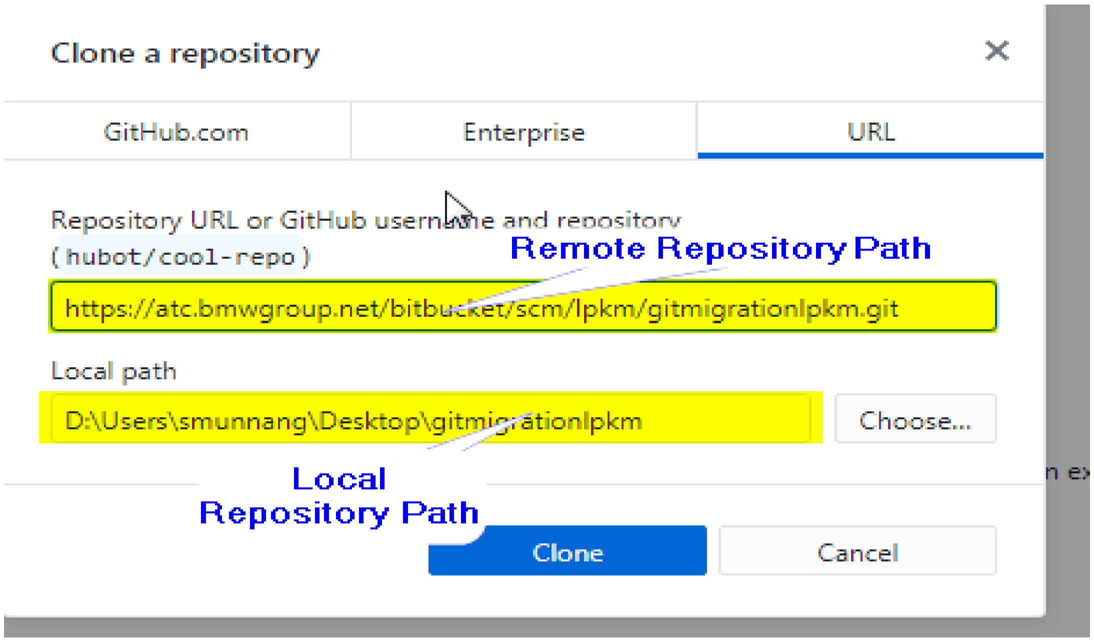
- Click on clone a Repository


File
Edit
View
Repository
Branch
Help
Current repository
No repositories
No repositories found
Create a new project and publish it to GitHub
Create new repository
Add an existing project on your computer and publish it to GitHub
Add a local repository
Clone an existing project from GitHub to your computer
Clone a repository
1
Alternatively, you can drag and drop a local repository here to add it.

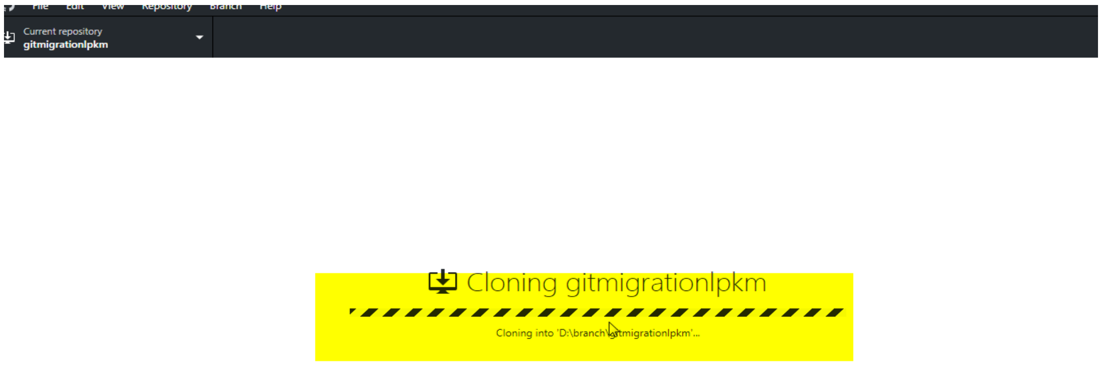
- Below screen appears and then select URL



- Enter the Git repository URL so that local path of the repository is selected automatically [If required, local path can be changed]


Clone a repository
GitHub.com
Enterprise
URL
Repository URL or GitHub username and repository (hubot/cool-repo)
Remote Repository Path
https://atc.bmwgroup.net/bitbucket/scm/lpkm/gitmigrationlpkm.git
Local path
D:\Users\smunnang\Desktop\gitmigrationlpkm
Choose...
Local Repository Path
Clone
Cancel

- Then click on ‘Clone’ so that cloning process starts

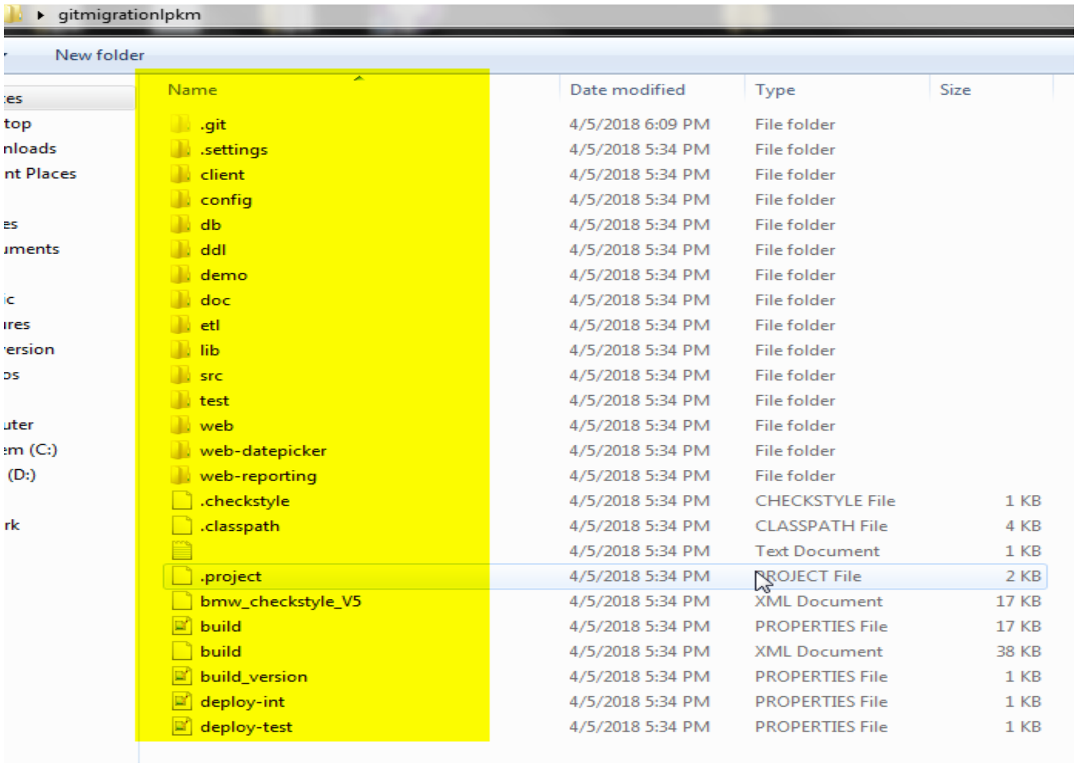


Cloning gitmigrationlpkm

Cloning into 'D:\branch\gitmigrationlpkm'...

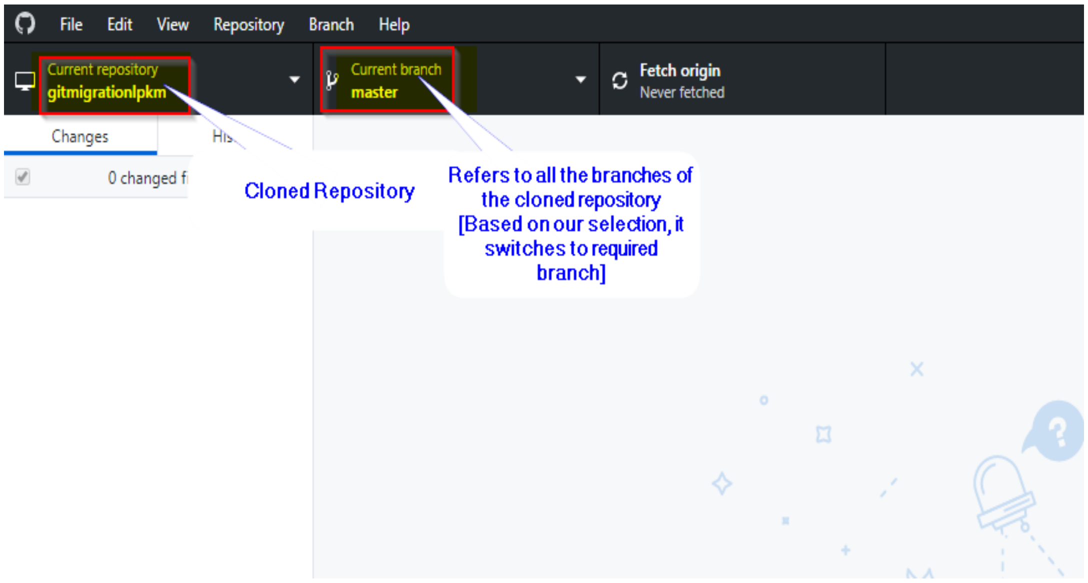
- Once the cloning process is completed a folder appears in the local system with name of the repository.

- Contents inside the Master appears as below



Name	Date modified	Type	Size
.git	4/5/2018 6:09 PM	File folder	
.settings	4/5/2018 5:34 PM	File folder	
client	4/5/2018 5:34 PM	File folder	
config	4/5/2018 5:34 PM	File folder	
db	4/5/2018 5:34 PM	File folder	
ddl	4/5/2018 5:34 PM	File folder	
demo	4/5/2018 5:34 PM	File folder	
doc	4/5/2018 5:34 PM	File folder	
etl	4/5/2018 5:34 PM	File folder	
lib	4/5/2018 5:34 PM	File folder	
src	4/5/2018 5:34 PM	File folder	
test	4/5/2018 5:34 PM	File folder	
web	4/5/2018 5:34 PM	File folder	
web-datepicker	4/5/2018 5:34 PM	File folder	
web-reporting	4/5/2018 5:34 PM	File folder	
.checkstyle	4/5/2018 5:34 PM	CHECKSTYLE File	1 KB
.classpath	4/5/2018 5:34 PM	CLASSPATH File	4 KB
	4/5/2018 5:34 PM	Text Document	1 KB
.project	4/5/2018 5:34 PM	PROJECT File	2 KB
bmw_checkstyle_V5	4/5/2018 5:34 PM	XML Document	17 KB
build	4/5/2018 5:34 PM	PROPERTIES File	17 KB
build	4/5/2018 5:34 PM	XML Document	38 KB
build_version	4/5/2018 5:34 PM	PROPERTIES File	1 KB
deploy-int	4/5/2018 5:34 PM	PROPERTIES File	1 KB
deploy-test	4/5/2018 5:34 PM	PROPERTIES File	1 KB

- The below screen describes Master Branch is cloned

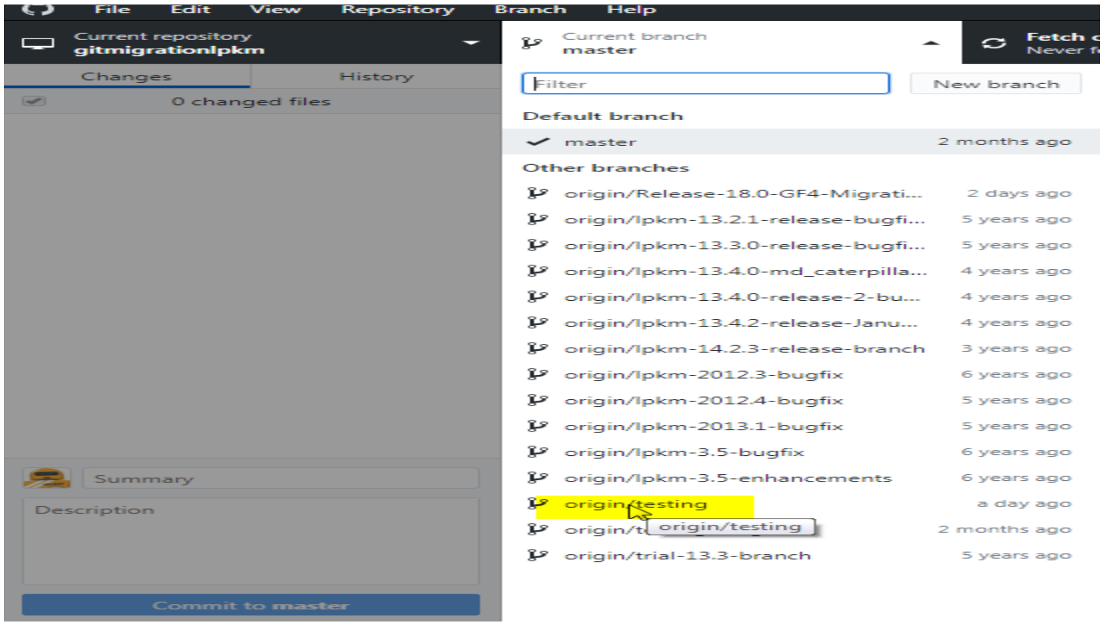


Cloned Repository

Refers to all the branches of the cloned repository [Based on our selection, it switches to required branch]

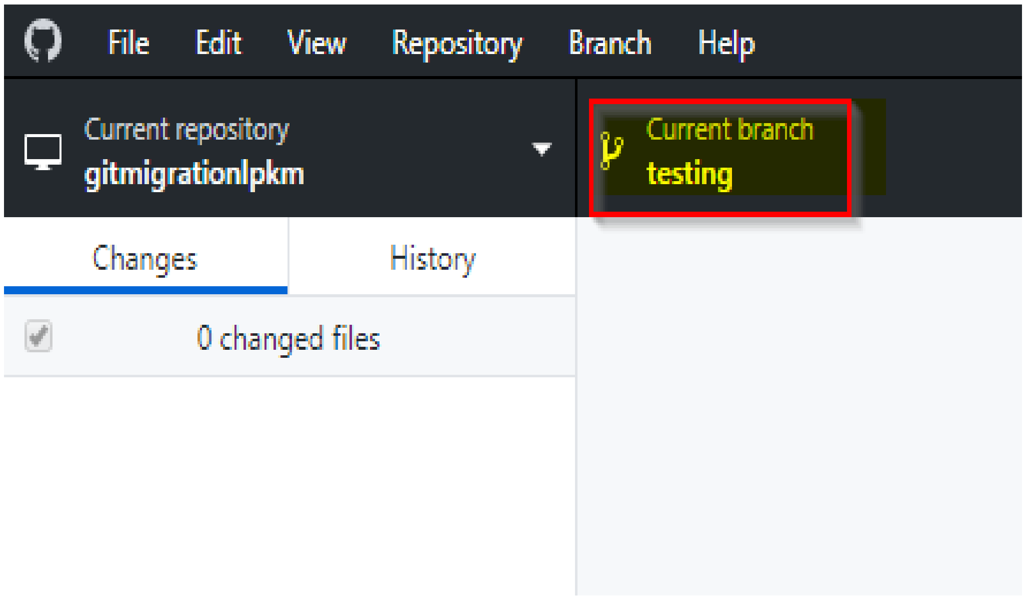
b) Cloning of a Branch:

- Go to GitHub Desktop and just change the Current Branch to the required Branch which automatically filled with the contents of the selected Branch

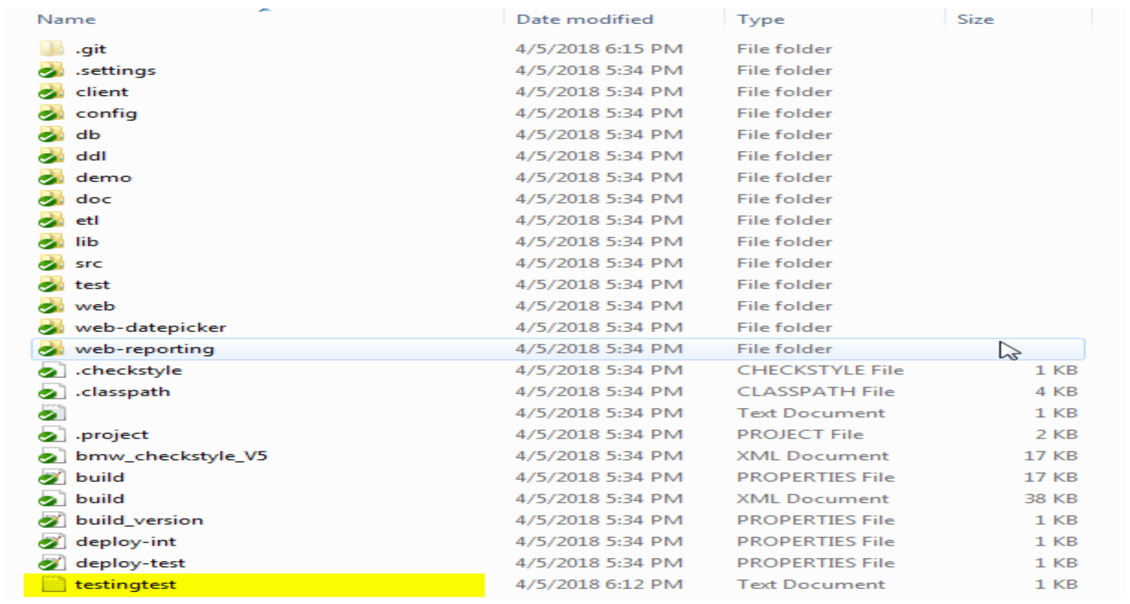


master	2 months ago
origin/Release-18.0-GF4-Migrati...	2 days ago
origin/lpkm-13.2.1-release-bugfi...	5 years ago
origin/lpkm-13.3.0-release-bugfi...	5 years ago
origin/lpkm-13.4.0-md_caterpilla...	4 years ago
origin/lpkm-13.4.0-release-2-bu...	4 years ago
origin/lpkm-13.4.2-release-Janu...	4 years ago
origin/lpkm-14.2.3-release-branch	3 years ago
origin/lpkm-2012.3-bugfix	6 years ago
origin/lpkm-2012.4-bugfix	5 years ago
origin/lpkm-2013.1-bugfix	5 years ago
origin/lpkm-3.5-bugfix	6 years ago
origin/lpkm-3.5-enhancements	6 years ago
origin/testing	a day ago
origin/t	2 months ago
origin/trial-13.3-branch	5 years ago

- Now the Branch is switched to '**testing**'



- Below screen shows the contents of ‘**testing’** Branch in the local repository.

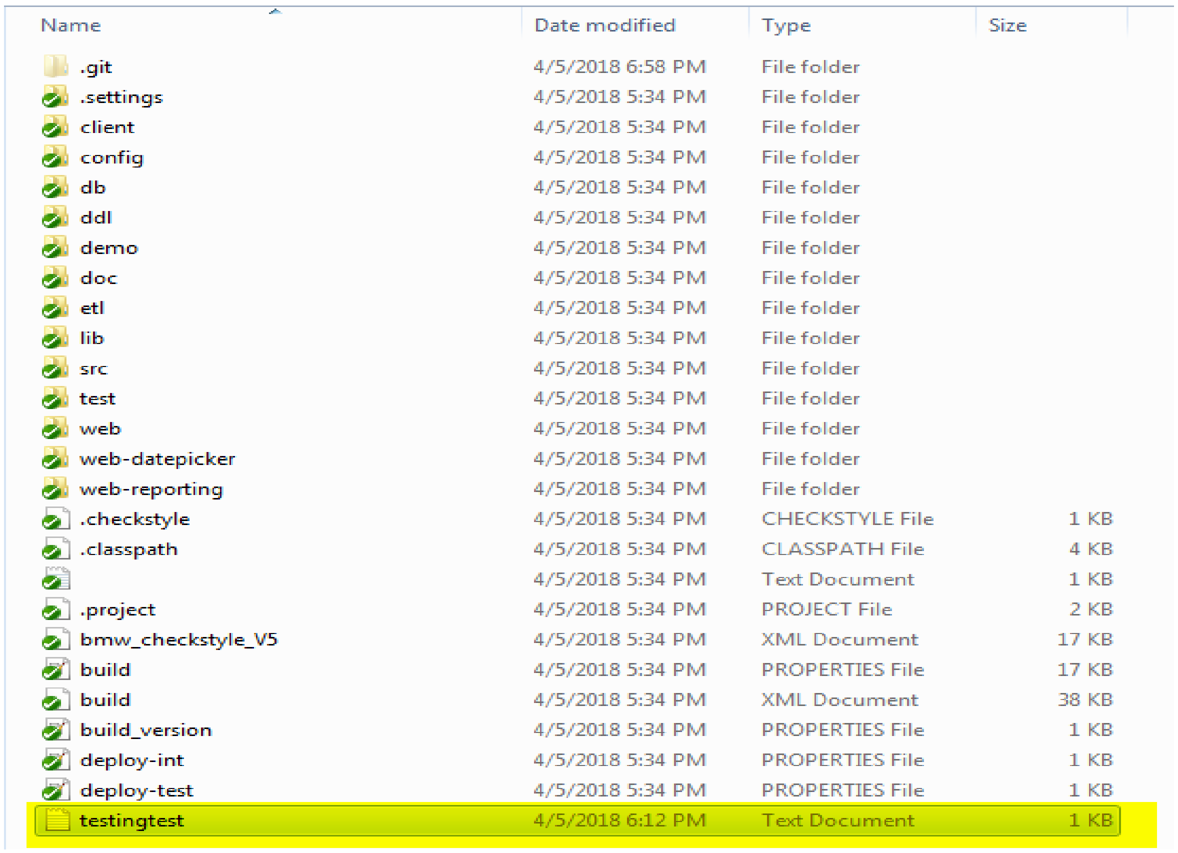


Name	Date modified	Type	Size
.git	4/5/2018 6:15 PM	File folder	
.settings	4/5/2018 5:34 PM	File folder	
client	4/5/2018 5:34 PM	File folder	
config	4/5/2018 5:34 PM	File folder	
db	4/5/2018 5:34 PM	File folder	
ddl	4/5/2018 5:34 PM	File folder	
demo	4/5/2018 5:34 PM	File folder	
doc	4/5/2018 5:34 PM	File folder	
etl	4/5/2018 5:34 PM	File folder	
lib	4/5/2018 5:34 PM	File folder	
src	4/5/2018 5:34 PM	File folder	
test	4/5/2018 5:34 PM	File folder	
web	4/5/2018 5:34 PM	File folder	
web-datepicker	4/5/2018 5:34 PM	File folder	
web-reporting	4/5/2018 5:34 PM	File folder	
.checkstyle	4/5/2018 5:34 PM	CHECKSTYLE File	1 KB
.classpath	4/5/2018 5:34 PM	CLASSPATH File	4 KB
	4/5/2018 5:34 PM	Text Document	1 KB
.project	4/5/2018 5:34 PM	PROJECT File	2 KB
bmw_checkstyle_V5	4/5/2018 5:34 PM	XML Document	17 KB
build	4/5/2018 5:34 PM	PROPERTIES File	17 KB
build	4/5/2018 5:34 PM	XML Document	38 KB
build_version	4/5/2018 5:34 PM	PROPERTIES File	1 KB
deploy-int	4/5/2018 5:34 PM	PROPERTIES File	1 KB
deploy-test	4/5/2018 5:34 PM	PROPERTIES File	1 KB
testingtest	4/5/2018 6:12 PM	Text Document	1 KB

3.2 Commit

1. Changes performed in the code can be committed from local repository to Git repository. It can be illustrated with an example below,

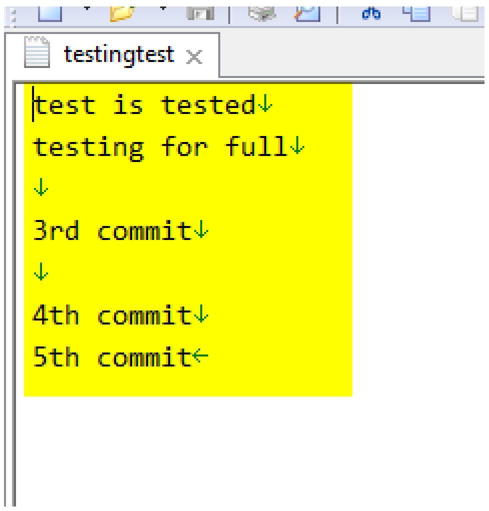
   Suppose consider the file ‘testingtest.txt’ [Inside ‘**testing’** Branch] in which changes need to be made,



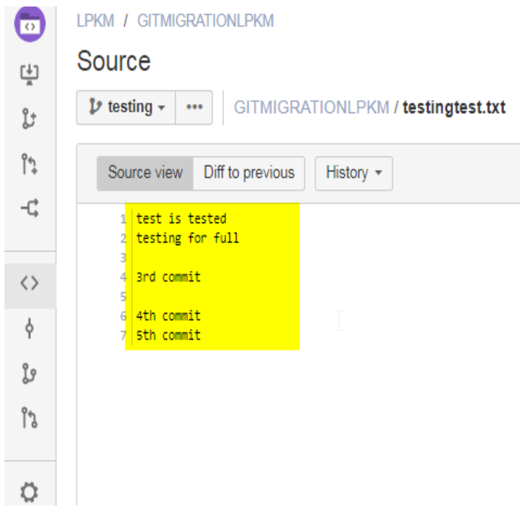
Name	Date modified	Type	Size
.git	4/5/2018 6:58 PM	File folder	
.settings	4/5/2018 5:34 PM	File folder	
client	4/5/2018 5:34 PM	File folder	
config	4/5/2018 5:34 PM	File folder	
db	4/5/2018 5:34 PM	File folder	
ddl	4/5/2018 5:34 PM	File folder	
demo	4/5/2018 5:34 PM	File folder	
doc	4/5/2018 5:34 PM	File folder	
etl	4/5/2018 5:34 PM	File folder	
lib	4/5/2018 5:34 PM	File folder	
src	4/5/2018 5:34 PM	File folder	
test	4/5/2018 5:34 PM	File folder	
web	4/5/2018 5:34 PM	File folder	
web-datepicker	4/5/2018 5:34 PM	File folder	
web-reporting	4/5/2018 5:34 PM	File folder	
.checkstyle	4/5/2018 5:34 PM	CHECKSTYLE File	1 KB
.classpath	4/5/2018 5:34 PM	CLASSPATH File	4 KB
	4/5/2018 5:34 PM	Text Document	1 KB
.project	4/5/2018 5:34 PM	PROJECT File	2 KB
bmw_checkstyle_V5	4/5/2018 5:34 PM	XML Document	17 KB
build	4/5/2018 5:34 PM	PROPERTIES File	17 KB
build	4/5/2018 5:34 PM	XML Document	38 KB
build_version	4/5/2018 5:34 PM	PROPERTIES File	1 KB
deploy-int	4/5/2018 5:34 PM	PROPERTIES File	1 KB
deploy-test	4/5/2018 5:34 PM	PROPERTIES File	1 KB
testingtest	4/5/2018 6:12 PM	Text Document	1 KB

2. The file ‘testingtest.txt’ in local & remote repositories before modification,

testingtest.txt – Local testingtest.txt – Remote

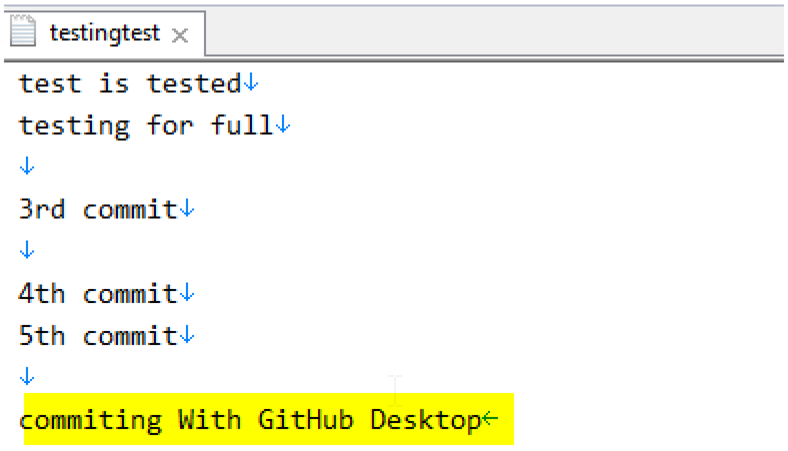


```
test is tested↓
testing for full↓
↓
3rd commit↓
↓
4th commit↓
5th commit←
```



```
test is tested
testing for full

3rd commit

4th commit
5th commit
```

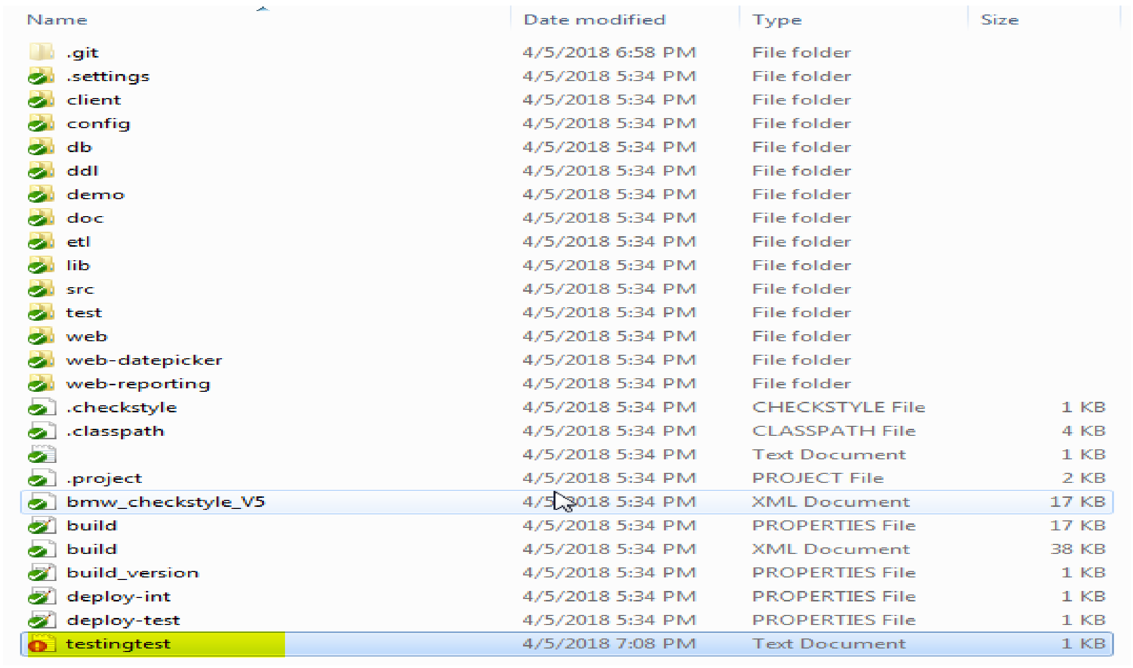
3. The file ‘testingtest.txt’ after modification,



testingtest

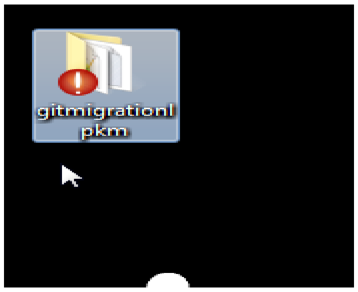
```
test is tested
testing for full

3rd commit

4th commit
5th commit

commiting With GitHub Desktop
```

4. After performing changes and saving the file ‘testingtest.txt’, it appears as in below screenshot

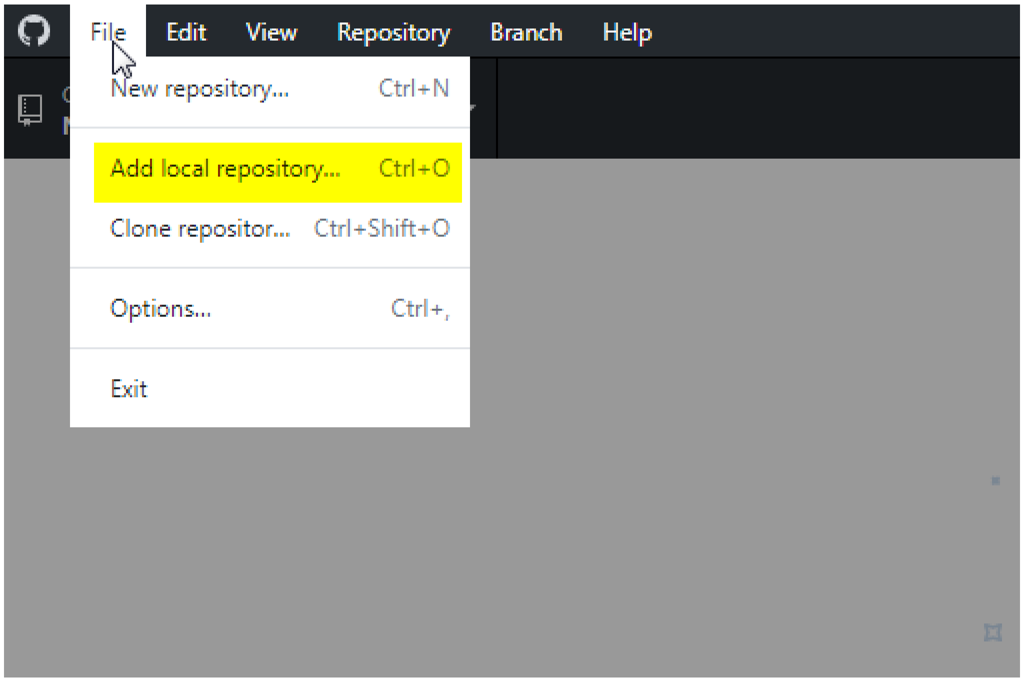


Name	Date modified	Type	Size
.git	4/5/2018 6:58 PM	File folder	
.settings	4/5/2018 5:34 PM	File folder	
client	4/5/2018 5:34 PM	File folder	
config	4/5/2018 5:34 PM	File folder	
db	4/5/2018 5:34 PM	File folder	
ddl	4/5/2018 5:34 PM	File folder	
demo	4/5/2018 5:34 PM	File folder	
doc	4/5/2018 5:34 PM	File folder	
etl	4/5/2018 5:34 PM	File folder	
lib	4/5/2018 5:34 PM	File folder	
src	4/5/2018 5:34 PM	File folder	
test	4/5/2018 5:34 PM	File folder	
web	4/5/2018 5:34 PM	File folder	
web-datepicker	4/5/2018 5:34 PM	File folder	
web-reporting	4/5/2018 5:34 PM	File folder	
.checkstyle	4/5/2018 5:34 PM	CHECKSTYLE File	1 KB
.classpath	4/5/2018 5:34 PM	CLASSPATH File	4 KB
	4/5/2018 5:34 PM	Text Document	1 KB
.project	4/5/2018 5:34 PM	PROJECT File	2 KB
bmw_checkstyle_V5	4/5/2018 5:34 PM	XML Document	17 KB
build	4/5/2018 5:34 PM	PROPERTIES File	17 KB
build	4/5/2018 5:34 PM	XML Document	38 KB
build_version	4/5/2018 5:34 PM	PROPERTIES File	1 KB
deploy-int	4/5/2018 5:34 PM	PROPERTIES File	1 KB
deploy-test	4/5/2018 5:34 PM	PROPERTIES File	1 KB
testingtest	4/5/2018 7:08 PM	Text Document	1 KB

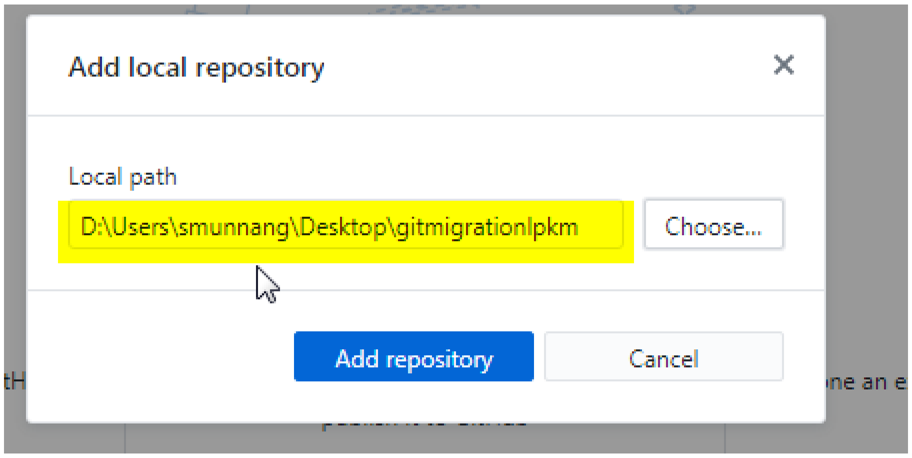
5. Post changes, cloned repository folder in the local system is visible as in below image



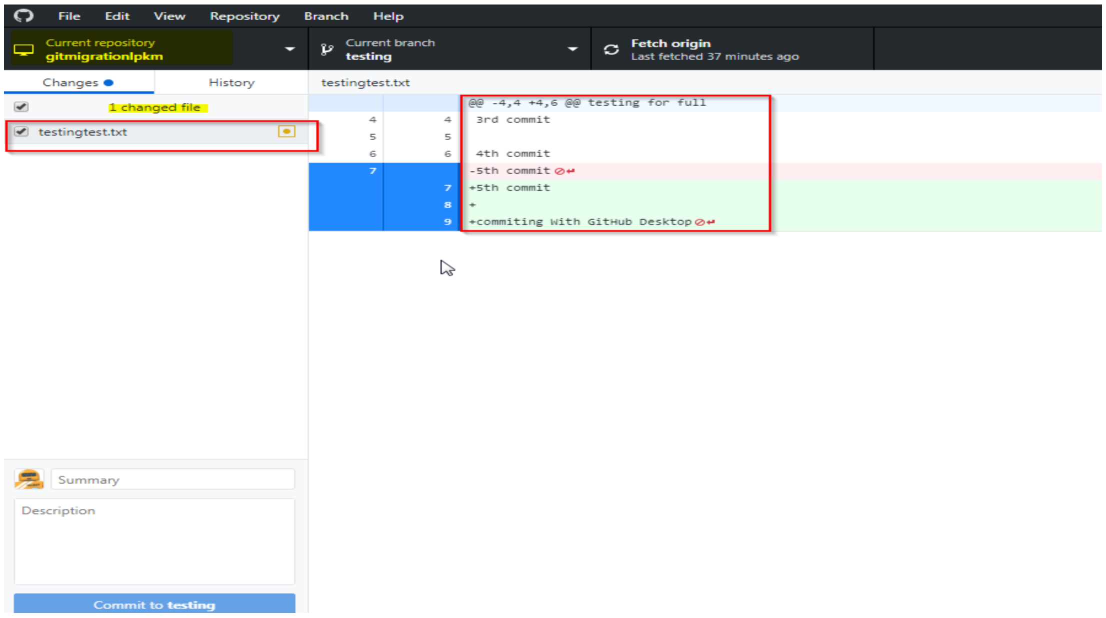
6. Go to GitHub Desktop and click on ‘Add the local repository’ which we modified[‘**testing’**]



7. Then the below screen appears, choose the local path of the modified repository and click on ‘Add repository’



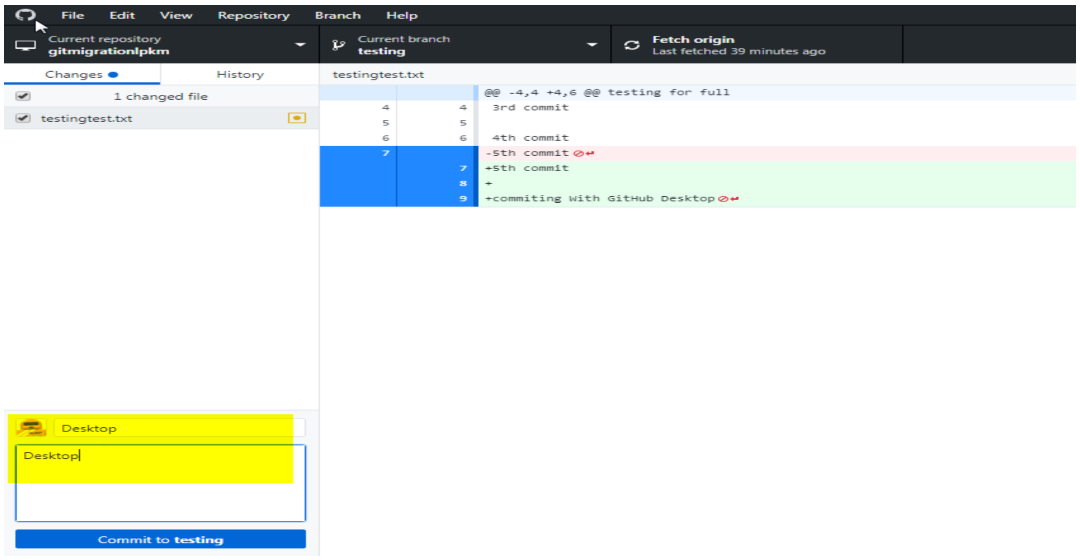
8. Then the files which are changed will appear as a list as shown in below screen. Here, the file ‘testingtest.txt’ is the modified file.



testingtest.txt

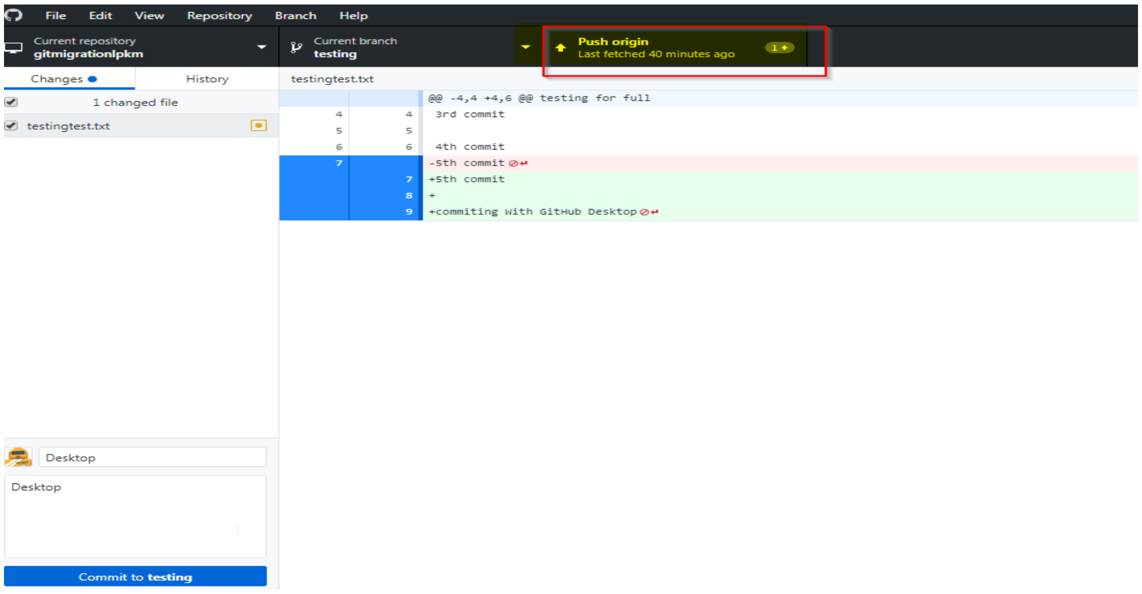
```
@@ -4,4 +4,6 @@ testing for full
 3rd commit

 4th commit
-5th commit
+5th commit
+
+commiting with GitHub Desktop
```

9. Enter the comments in the text box and click on ‘Commit to testing’



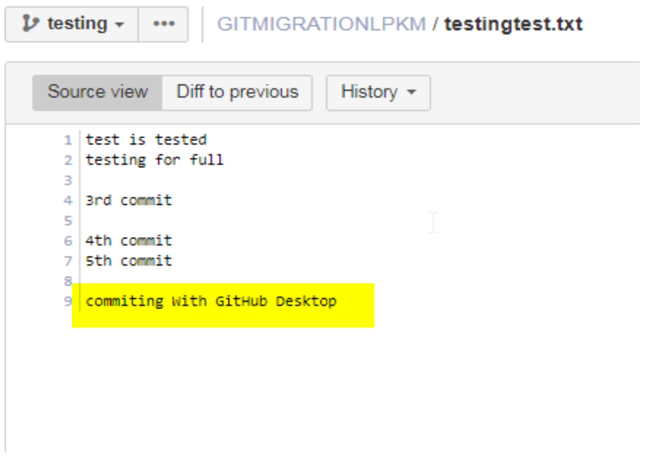
```
@@ -4,4 +4,6 @@ testing for full
 3rd commit

 4th commit
-5th commit
+5th commit
+
+commiting with GitHub Desktop
```

10. Post commit, click on ‘Push origin’ to push the changes to the remote Git repository.



```
@@ -4,4 +4,6 @@ testing for full
 3rd commit

 4th commit
-5th commit
+5th commit
+
+commiting with GitHub Desktop
```

11. Here is the remote repository screen of the ‘testingtest.txt’ file which shows that changes are pushed successfully



```
test is tested
testing for full

3rd commit

4th commit
5th commit

commiting with GitHub Desktop
```

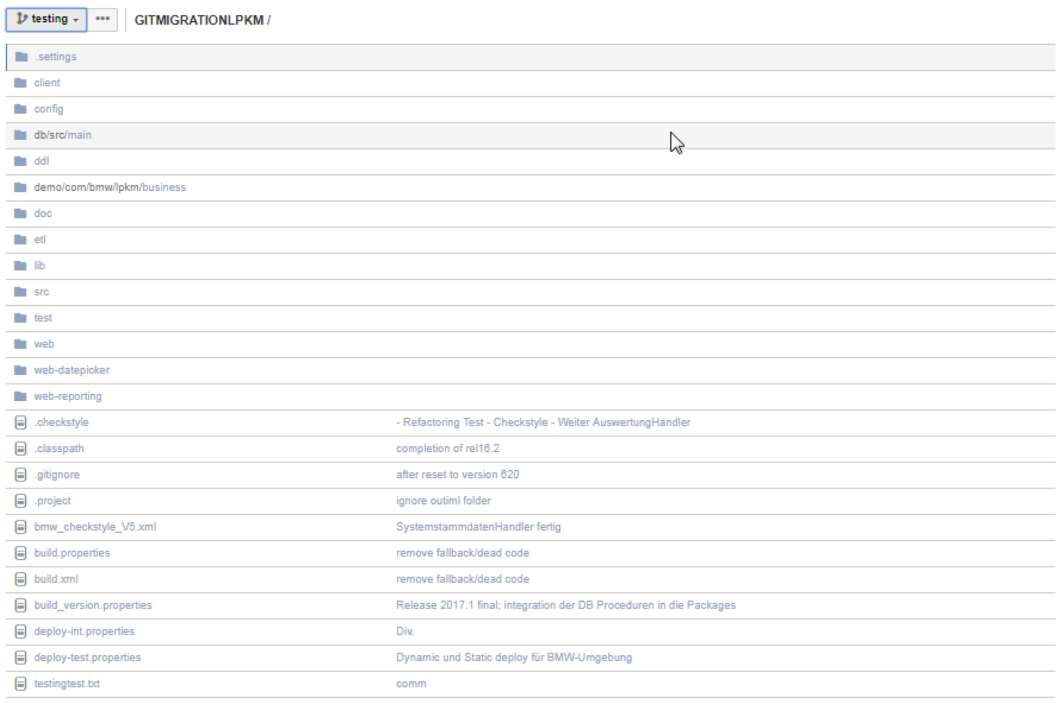
3.3 Merge

Note :

- In general, Git repository has an option to sync with SVN
- **To ‘Merge’ the Branches, the sync option in Git repository should be disabled**

Merge of ‘**testing’** Branch with ‘**testingMerge’** Branch

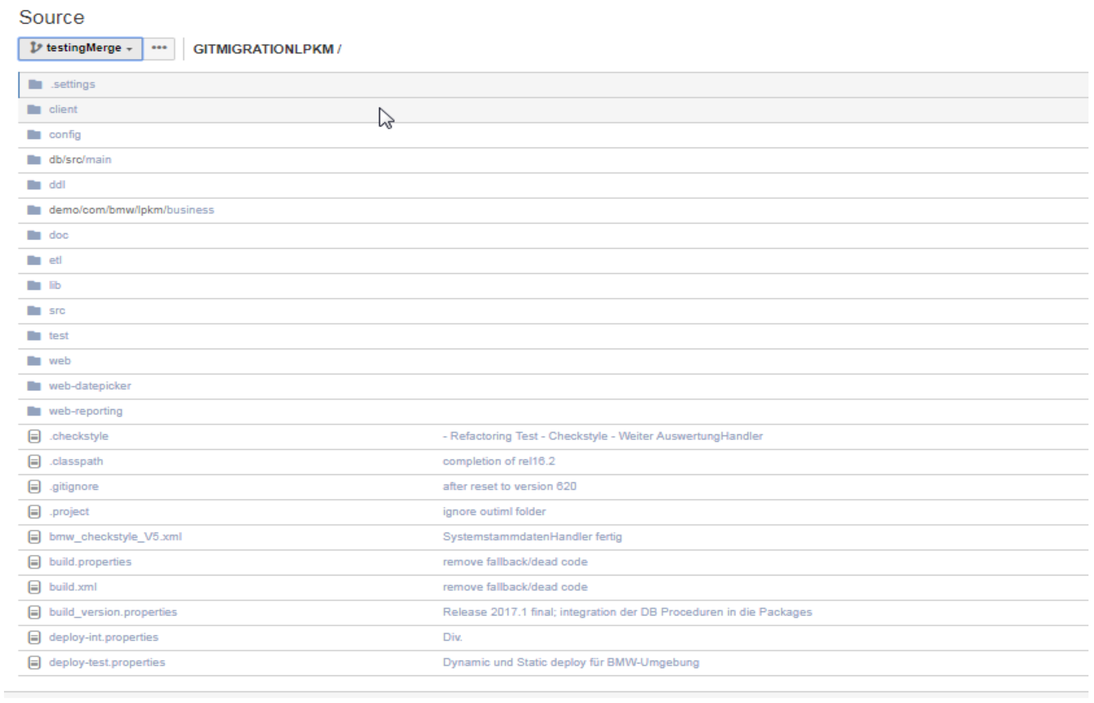
1. Refer the below screenshot for the contents of ‘**testing’** Branch in Git repository with ‘testingtest.txt’ file.



testing GITMIGRATIONLPKM /

.settings	
client	
config	
db/src/main	
ddl	
demo/com/bmw/lpkm/business	
doc	
etl	
lib	
src	
test	
web	
web-datepicker	
web-reporting	
.checkstyle	- Refactoring Test - Checkstyle - Weiter AuswertungHandler
.classpath	completion of rel16.2
.gitignore	after reset to version 620
.project	ignore outiml folder
bmw_checkstyle_V5.xml	SystemstammdatenHandler fertig
build.properties	remove fallback/dead code
build.xml	remove fallback/dead code
build_version.properties	Release 2017.1 final; integration der DB Proceduren in die Packages
deploy-int.properties	Div.
deploy-test.properties	Dynamic und Static deploy für BMW-Umgebung
testingtest.txt	comm

2. Refer the below screenshot for the contents of ‘**testingMerge’** Branch in Git repository without ‘testingtest.txt’ file.

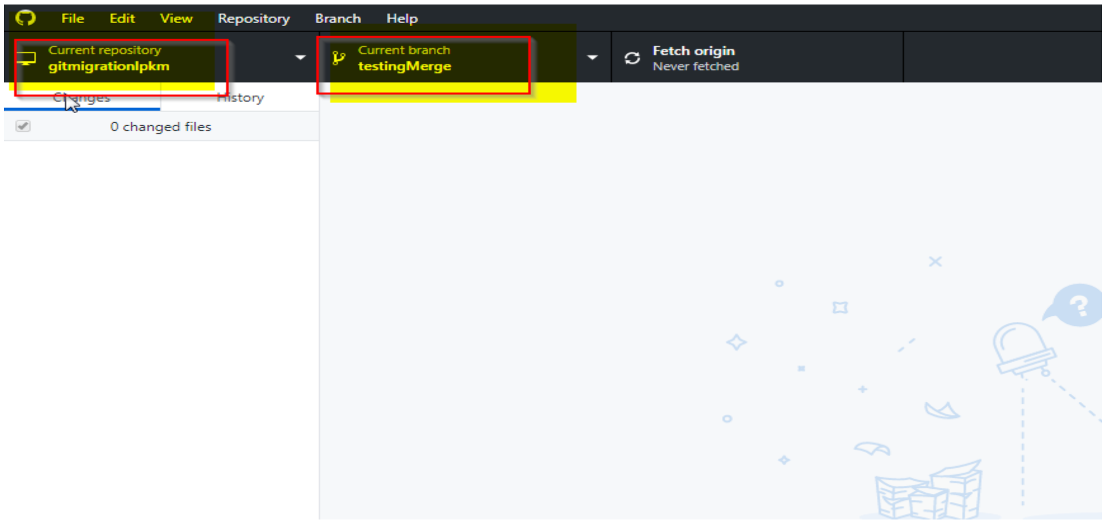


Source

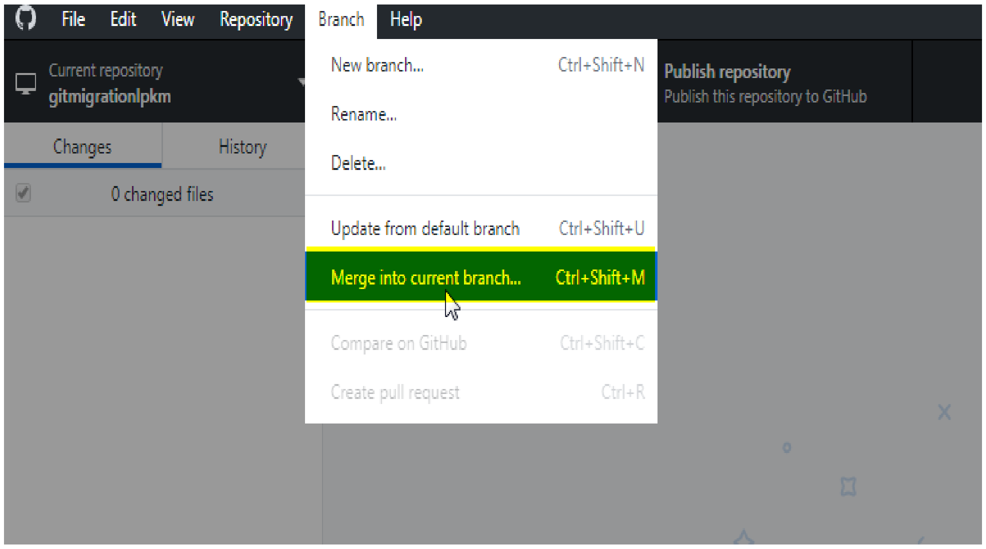
testingMerge GITMIGRATIONLPKM /

.settings	
client	
config	
db/src/main	
ddl	
demo/com/bmw/lpkm/business	
doc	
etl	
lib	
src	
test	
web	
web-datepicker	
web-reporting	
.checkstyle	- Refactoring Test - Checkstyle - Weiter AuswertungHandler
.classpath	completion of rel16.2
.gitignore	after reset to version 620
.project	ignore outiml folder
bmw_checkstyle_V5.xml	SystemstammdatenHandler fertig
build.properties	remove fallback/dead code
build.xml	remove fallback/dead code
build_version.properties	Release 2017.1 final; integration der DB Proceduren in die Packages
deploy-int.properties	Div.
deploy-test.properties	Dynamic und Static deploy für BMW-Umgebung

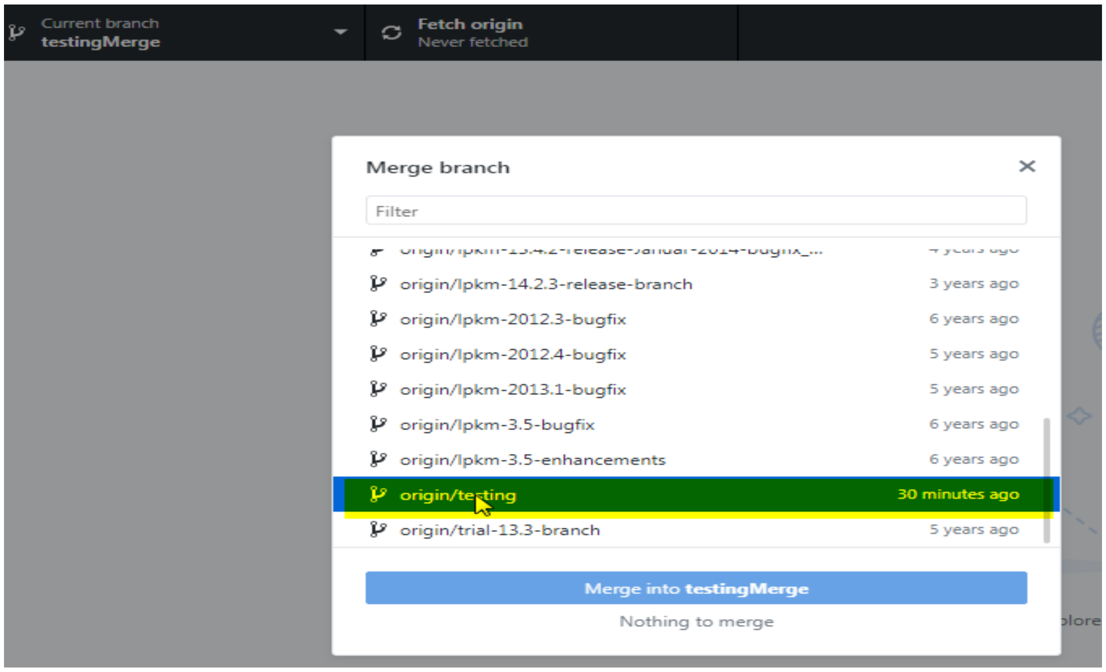
3. Open GitHubDesktop and clone the ‘**testingMerge’** so that current Branch is set to ‘**testingMerge’**



4. Go to Branch→ select ‘Merge into current Branch’


File
Edit
View
Repository
Branch
Help
Current repository
gitmigrationlpkm
Changes
History
0 changed files
New branch... Ctrl+Shift+N
Rename...
Delete...
Update from default branch Ctrl+Shift+U
Merge into current branch... Ctrl+Shift+M
Compare on GitHub Ctrl+Shift+C
Create pull request Ctrl+R
Publish repository
Publish this repository to GitHub

5. A popup will be displayed → select ‘**testing**’ Branch



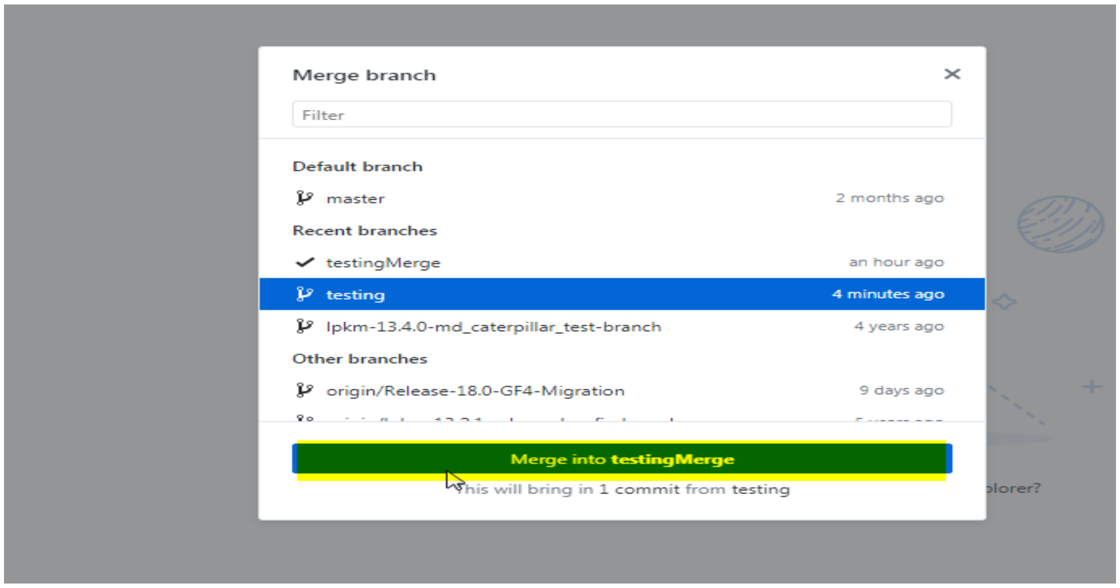
Current branch
testingMerge
Fetch origin
Never fetched

Merge branch
Filter

origin/lpkm-14.2.3-release-branch	3 years ago
origin/lpkm-2012.3-bugfix	6 years ago
origin/lpkm-2012.4-bugfix	5 years ago
origin/lpkm-2013.1-bugfix	5 years ago
origin/lpkm-3.5-bugfix	6 years ago
origin/lpkm-3.5-enhancements	6 years ago
origin/testing	30 minutes ago
origin/trial-13.3-branch	5 years ago

Merge into **testingMerge**
Nothing to merge

6. Click on ‘Merge into testingMerge’ button to merge the Branches



Merge branch
Filter

Default branch

master	2 months ago

Recent branches

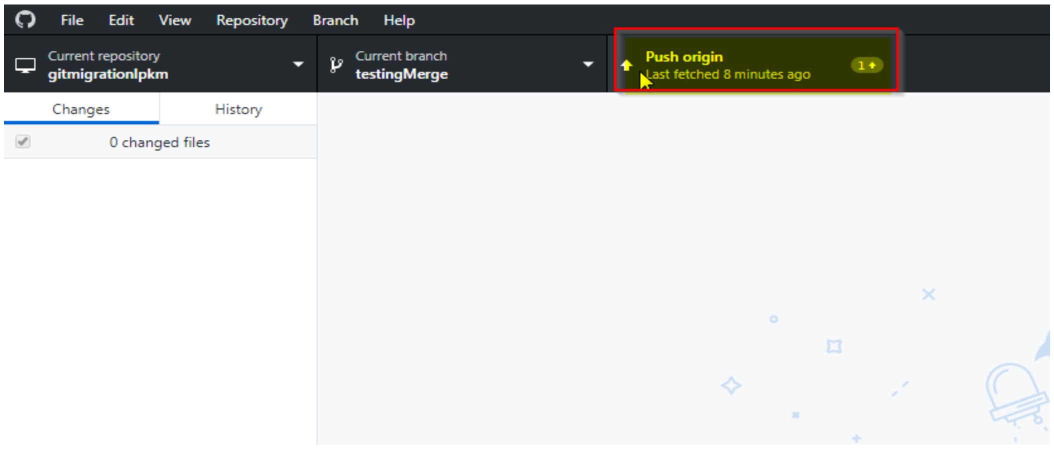
✓ testingMerge	an hour ago
testing	4 minutes ago
lpkm-13.4.0-md_caterpillar_test-branch	4 years ago

Other branches

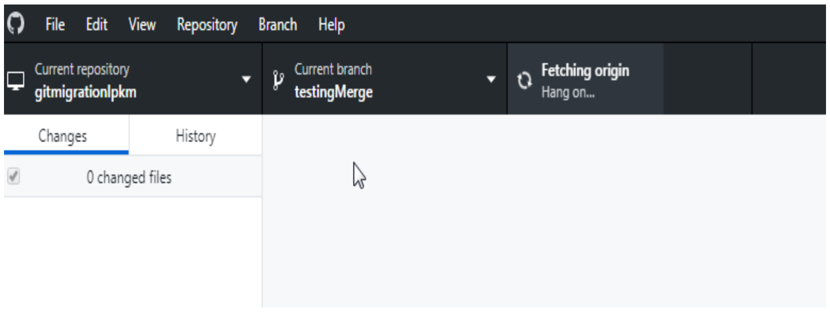
origin/Release-18.0-GF4-Migration	9 days ago

Merge into **testingMerge**
This will bring in 1 commit from testing

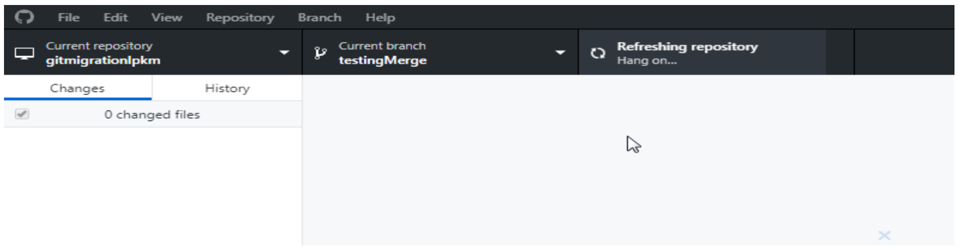
7. Then click on ‘Push origin’



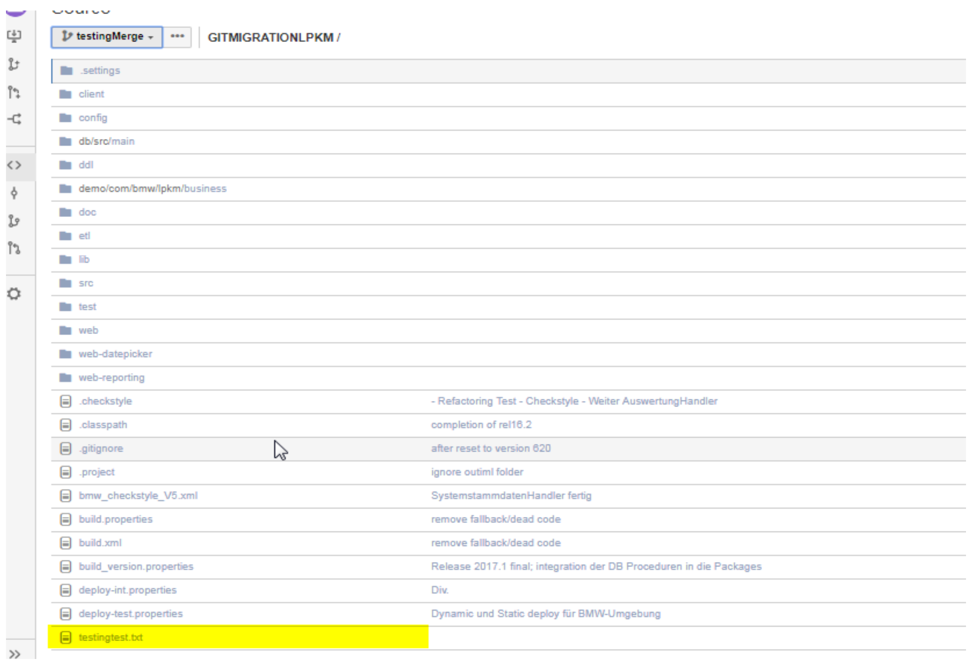
8. Fetching origin is in progress



9. Then, it refreshes the repository



10. Now the ‘testingtest.txt’ file of ‘**testing’** Branch is merged into ‘**testingMerge’** Branch in Git repository



testingMerge ▾ ··· GITMIGRATIONLPKM /

.settings	
client	
config	
db/src/main	
ddl	
demo/com/bmw/lpkm/business	
doc	
etl	
lib	
src	
test	
web	
web-datepicker	
web-reporting	
.checkstyle	- Refactoring Test - Checkstyle - Weiter AuswertungHandler
.classpath	completion of rel16.2
.gitignore	after reset to version 620
.project	ignore outiml folder
bmw_checkstyle_V5.xml	SystemstammdatenHandler fertig
build.properties	remove fallback/dead code
build.xml	remove fallback/dead code
build_version.properties	Release 2017.1 final; integration der DB Proceduren in die Packages
deploy-int.properties	Div.
deploy-test.properties	Dynamic und Static deploy für BMW-Umgebung
testingtest.txt	