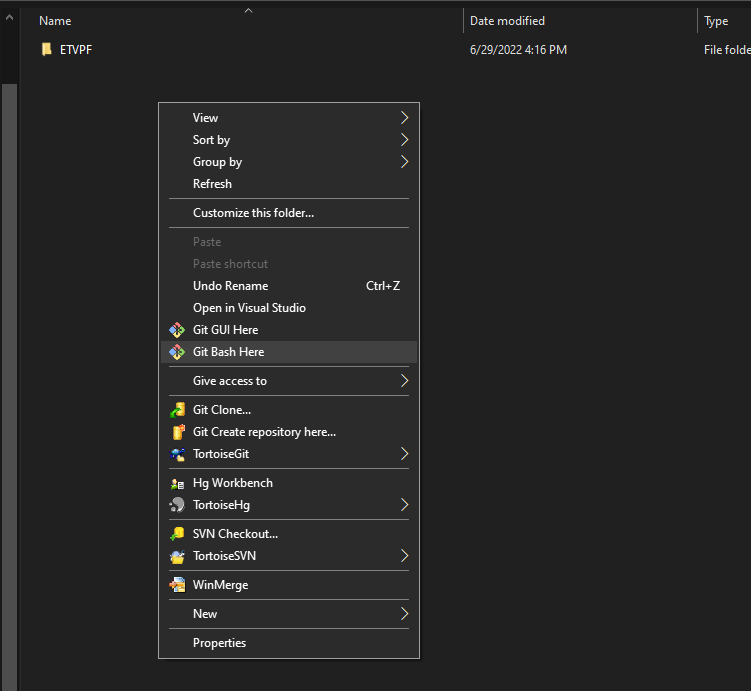
**Tutorial Using GIT-LAB**

# Init Gitlab

1. Create new directory  
   Graphical user interface

   Description automatically generated
2. Open Git Bash Here



1. Change the username and email global in gitbash

Graphical user interface, text, application

Description automatically generated

Figure 1 Check info in GITLAB

git config --global user.name "<username in GITLAB>"

git config --global user.email "<your.email@renesas.com>"

1. Disable SSL verification globally while running the git clone.

git config --global http.sslVerify false

1. Get the HTTPS on GITLAB sever  
   Graphical user interface, text, application, email

   Description automatically generated

Figure 2 Check HTTPS git

1. Using git clone to clone the project  
   git clone “HTTPS” – After using git clone, some users will popup the login account, Enter the account from GITLAB   
   A screenshot of a computer

   Description automatically generated

Figure 3 Check Git Clone Success

1. …

# How to update

* Go to directory you want to update -> Open Git Bash Here  
  Graphical user interface

  Description automatically generated
* Use command “git pull” to pull new source code

Text

Description automatically generated

# How to commit

* Refer part “I How to update” for update to new revision
* Use “git status” to see which file have been edit (In picture I modify callback\_about.m and add new source callback\_about\_new.m)

A screenshot of a computer

Description automatically generated

* Using “git add ‘file’ ” – You can add separate each file or add full with command “git add -A”

Text

Description automatically generated

* When “git status” is green it’s mean you complete add it into git
* Use git commit -m "commit message"

Text

Description automatically generated

* After commit success, using “git push” to push the commit into sever

Text

Description automatically generated

Text

Description automatically generated

# How to check revision history

* Using “git log” to check revision

Text

Description automatically generated

* If you want to check previous revision “git log -1”

Text

Description automatically generated

* Exit in “git log” press “q”

Text

Description automatically generated

# How to revert to revision history

* Using git log in chapter III, and remember the revision you want to revert like the red draw bellow

Text

Description automatically generated

* You have 2 option in this part:  
  1. You want check out new revision into new branch not impact old branch => Using “git checkout revision” => Your current working in revision a37df62517911d5a594afa1f25cc9df84edb7bd6 and in another branch, not impact main branch

Text

Description automatically generated

2. You want to revert into revision xxx and in main branch, using “git revert xxx” - xxx is revision => This way is very rick because when you revert it will cause conflict unless you have solution for conflict file

=> I prefer the option 1 checkout to new branch to do anything you want, then you commit it into main branch

# How to move to another branch

1. Git branch -> check your current branch on repo

Text

Description automatically generated

1. Git branch -a -> check all branch exist in repo

Text

Description automatically generated

1. Git checkout branch name -> move to branch

Graphical user interface

Description automatically generated with medium confidence

# How to merge branch

T.B.D

# Reset git when your git error

If your git is error, doing following command:  
- Copy your modify or important file on your git out into another location (Backup file before reset git)

* Execute command “Git clean -d -f” to clean all file which not exist on git sever
* Execute command “Git reset --hard” to reset git into current revision
* Execute command “Git pull origin” to push new revision on git