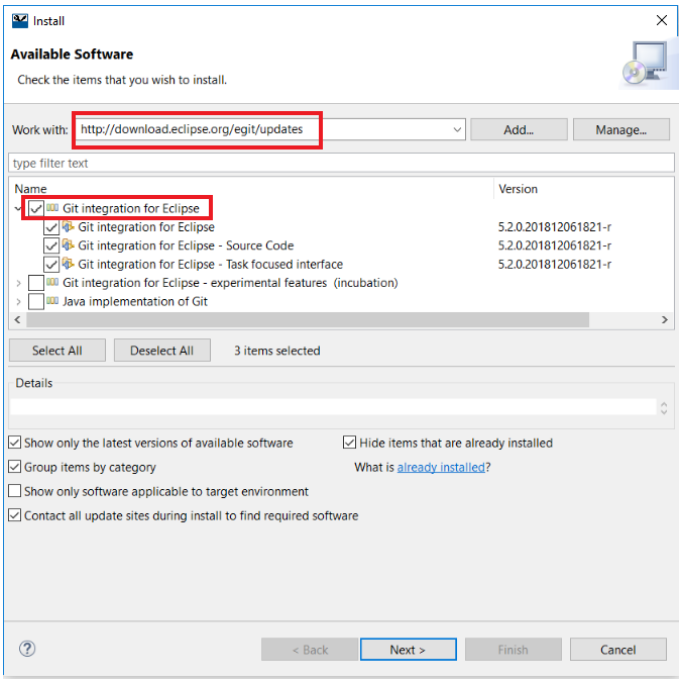
Question: How do you add Git version control system to projects in ModusToolbox™?

Answer: Version control systems keep track of modifications to a file over time so early revisions can be restored and used by teams for the source code. It acts as a special kind of database to help fix mistakes, prevent concurrent work from conflicting and allow for smooth and continuous flow of changes to code. This article uses **Git**, a popular distributed version control system designed for coordinating work among teams.

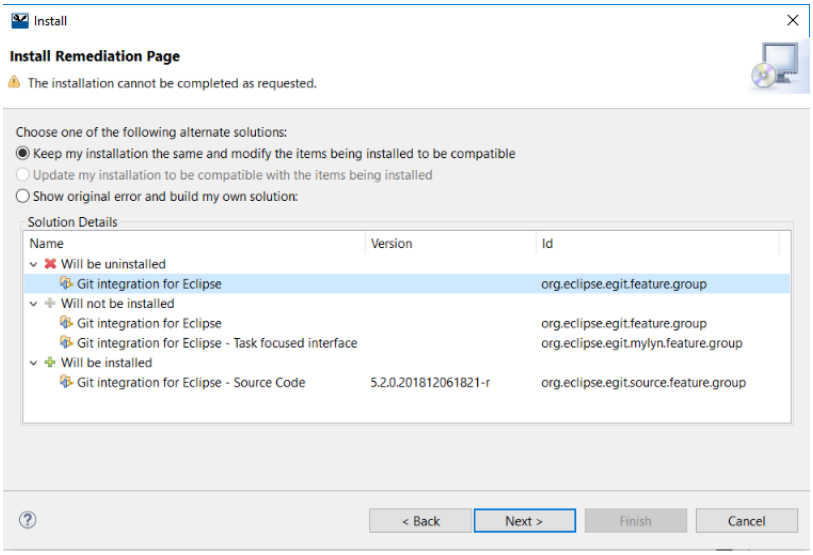
The following steps illustrate the process of integrating projects onto Git:

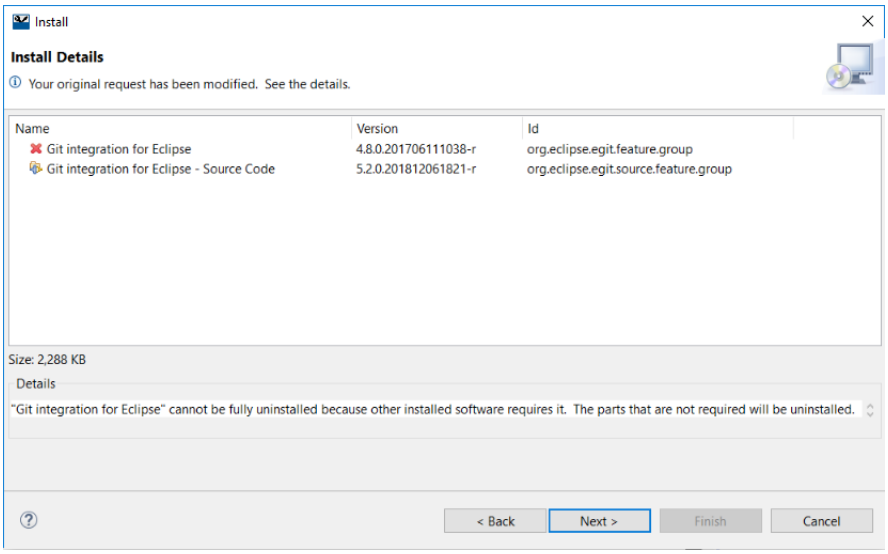
Step1: Updating the Git plugin

ModusToolbox™ comes with Git (v4.8.0) integration by default. To get the latest plugin, go to "**Help > Install New Software**" and add "http://download.eclipse.org/egit/updates" as shown:



It will list the available Git plugins that can updated/installed. Select the required plugins and click on **Next**.

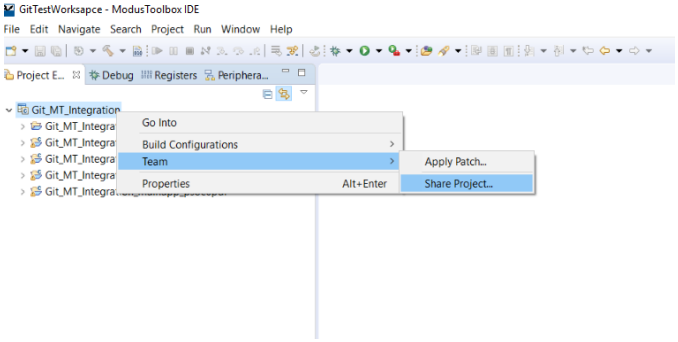




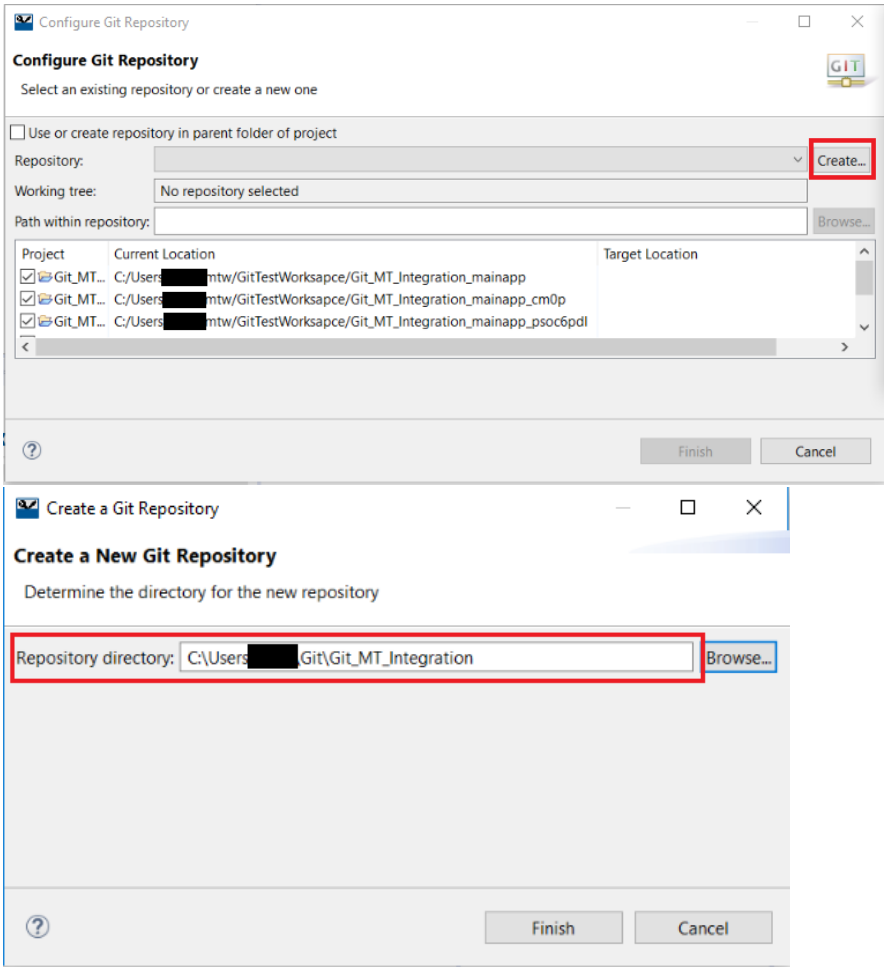
Follow the default onscreen instructions, accept the license agreement and complete the installation. Now restart ModusToolbox for the changes to take effect.

Step 2: Adding projects to Git

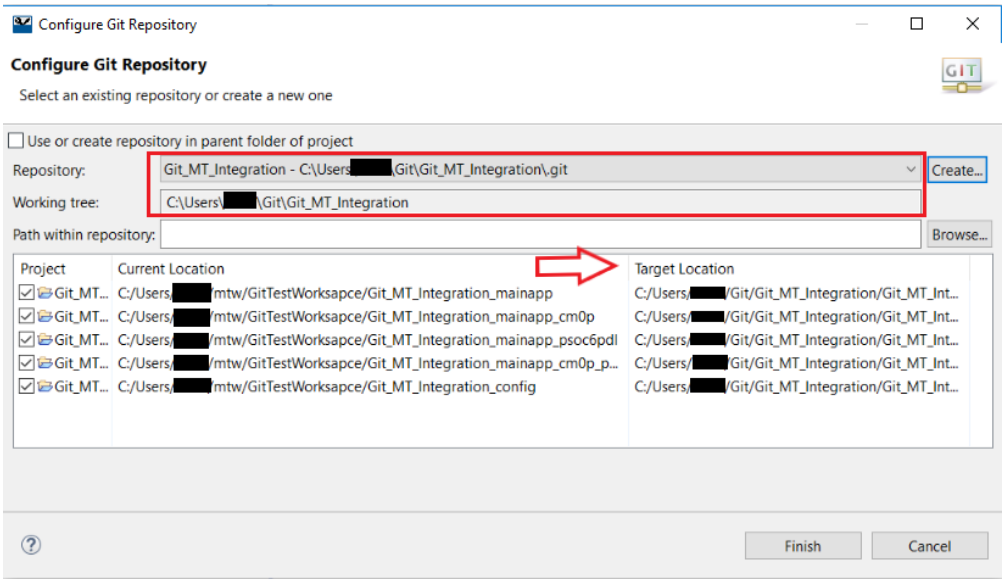
Create a working set for your project so that all the files can be added easily as mentioned here. **Right click on the working set > Team > Share Project** as shown below:



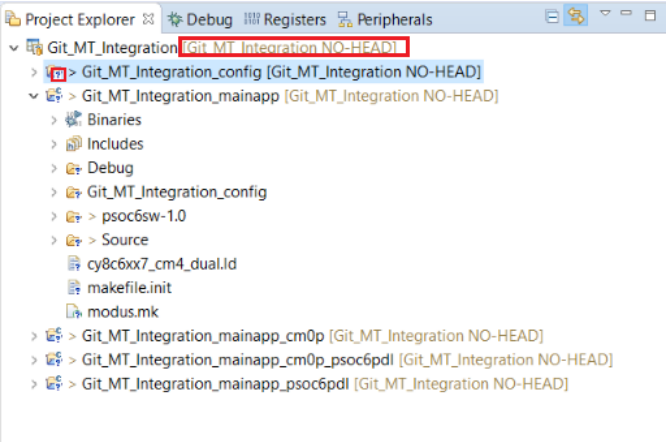
Now click on **Create** to create a new repository. Enter the directory of choice for the project files as shown below:



Please note that this will place all the project files in this new directory as shown below and will no longer be present in the workspace directory.



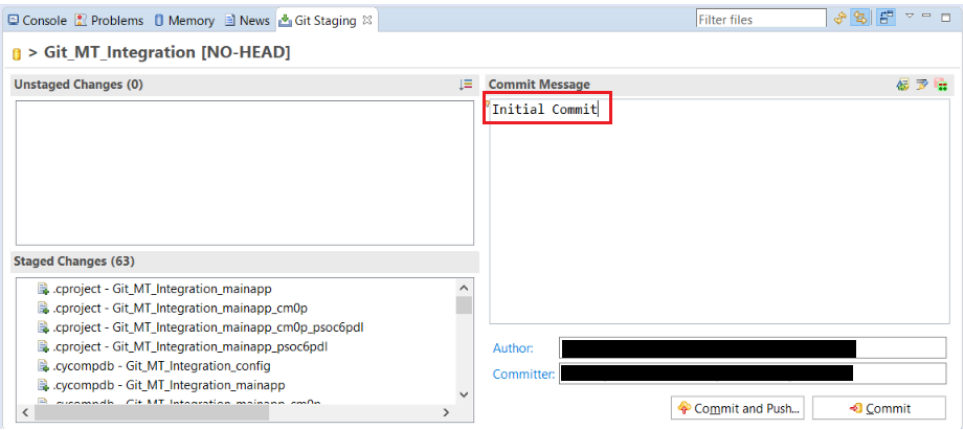
Now click on **Finish** and you should observe all the project files being decorated as shown:



The NO-HEAD label basically indicates that the Git configuration is not yet complete. The subtle question mark symbol indicates that the projects are yet to be committed.

Step 3: Committing the files

Right click on the working set > Team > Add to index. You will observe all the question mark symbols now turn to a "+" symbol. This will add all the files to the staging area as shown below. If there are Unstaged changes, it means the files haven't been added to index yet. If you don't see the staging area, go to **Window > Show view > Other > Git > Git Staging.**



Now add a message to describe the changes. This will serve as a changelog for anyone viewing the files. Now click on **Commit.**

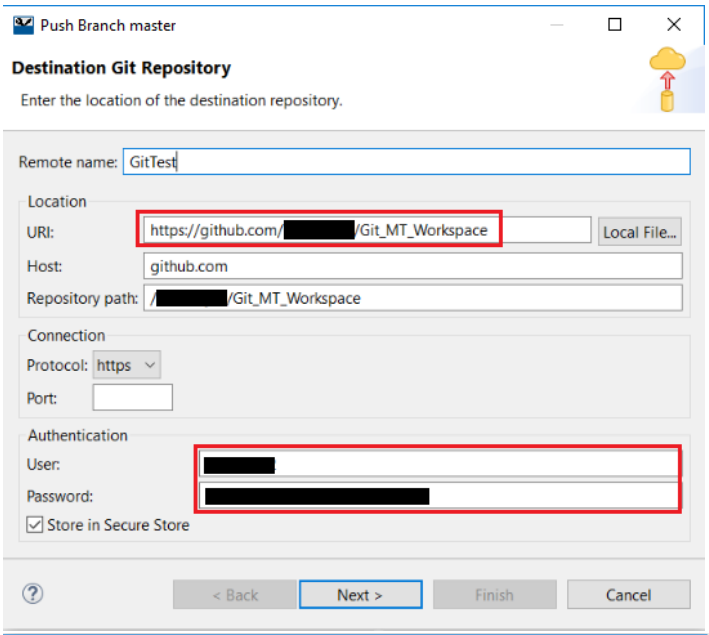
Step 4: Creating the online repository

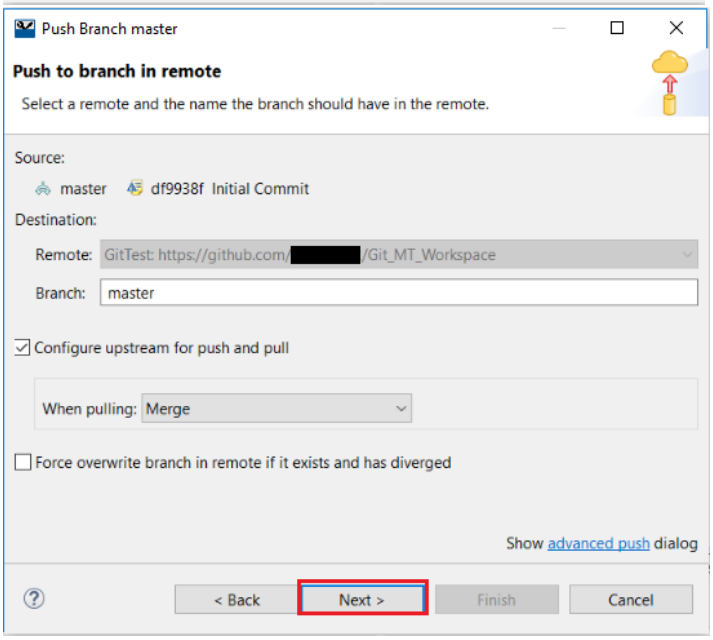
Create a public repository in any of popular the web hosting service for Git like **Github** or **Bitbucket**. Remember the credentials used to login to these sites. In this case, we will assume a repository named "*Git_MT_Workspace*" has been created.

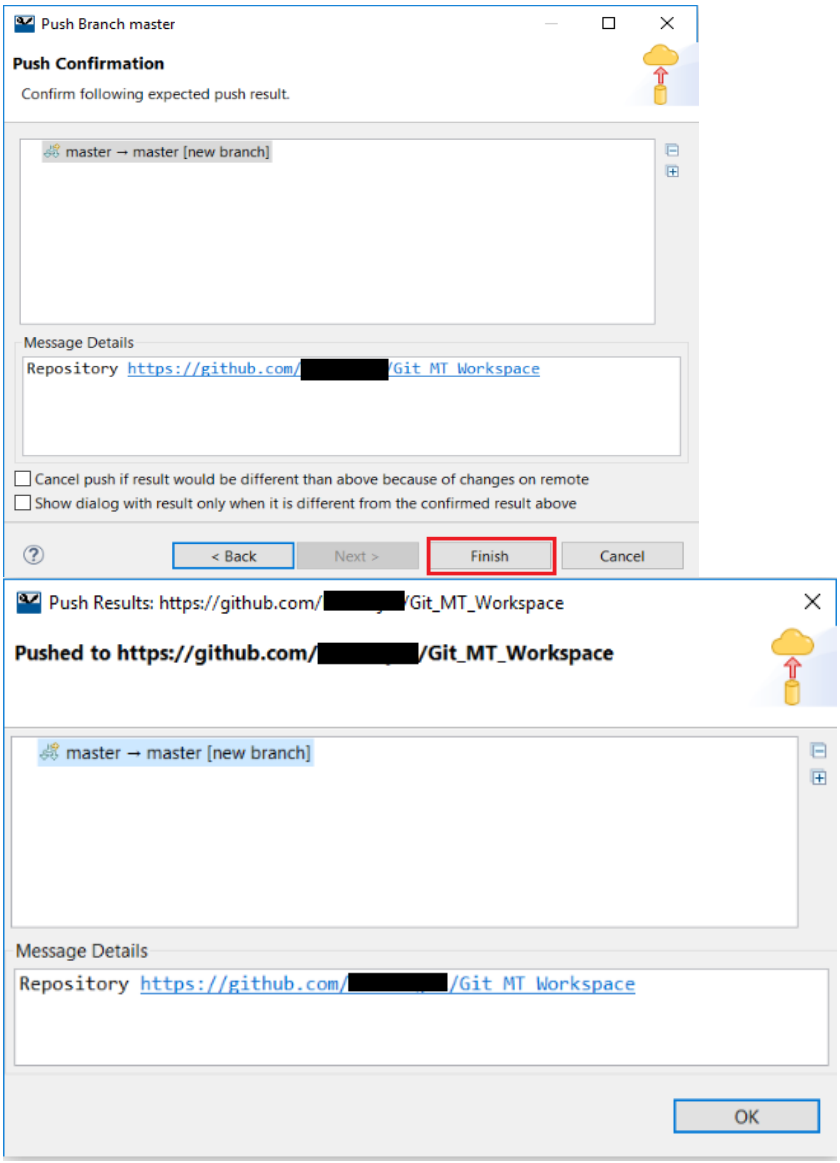
Step 5: Pushing the files to online repository

Till now all the changes were done in the local repository. Its time to push these changes to the public online repository for everyone to use. **Right click on the working set > Team > Push**. In the dialog prompt, add in the **URL to the repository** and the credentials under **Authentication**. You can choose to check the box "**Store in secure store**" to remember the credentials for future use. Follow the onscreen instructions to push the files as illustrated below.

*Please note: This is only for the first commit. For subsequent pushes, you can directly hit "Commit and Push" in the "**Git Staging**" window.*



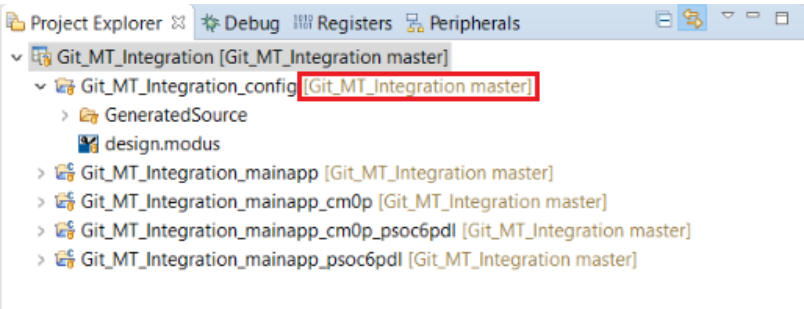




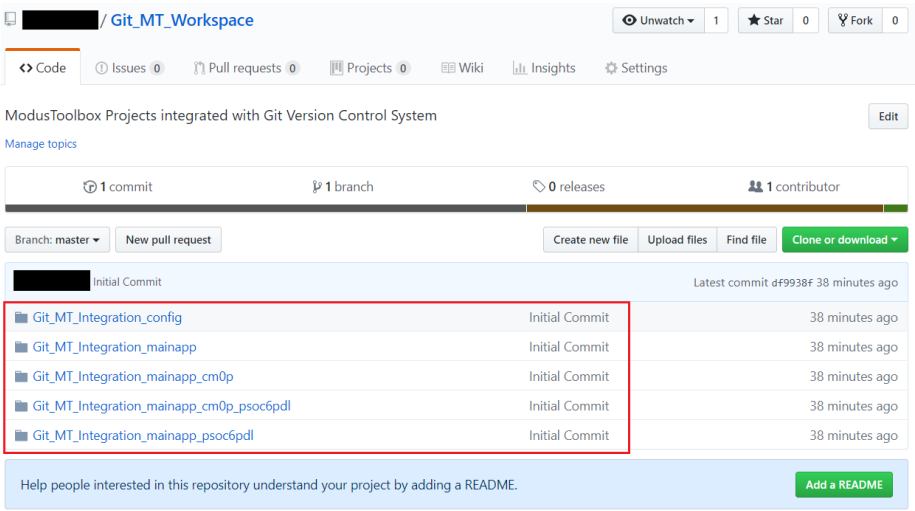
If there is a failure, the possible reasons for failure could be:

- There were changes done online to the files and the current project is not up to date. To solve this, **right click on the working set > Pull**. This will download all the latest files onto the project repository. Now you can push your changes.
- Access restrictions might be in place for the online repository. Check with the web hosting service to make the access public or for means of authentication in case of private repository.

You should see the file decorations now having a head node.

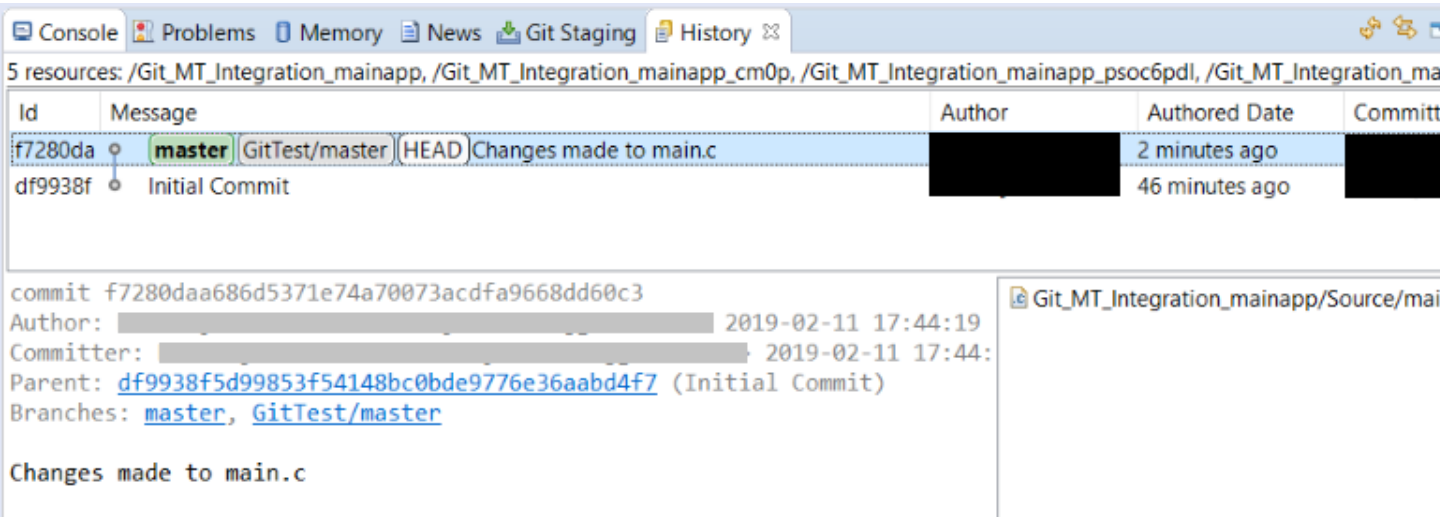


If the push was successful, the files should now be seen in the online repository along with the commit message.

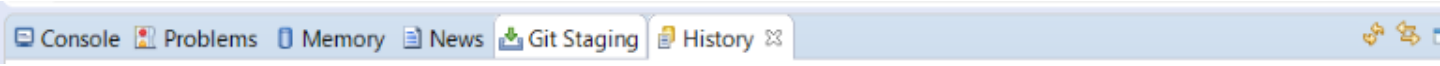


Additional Information:

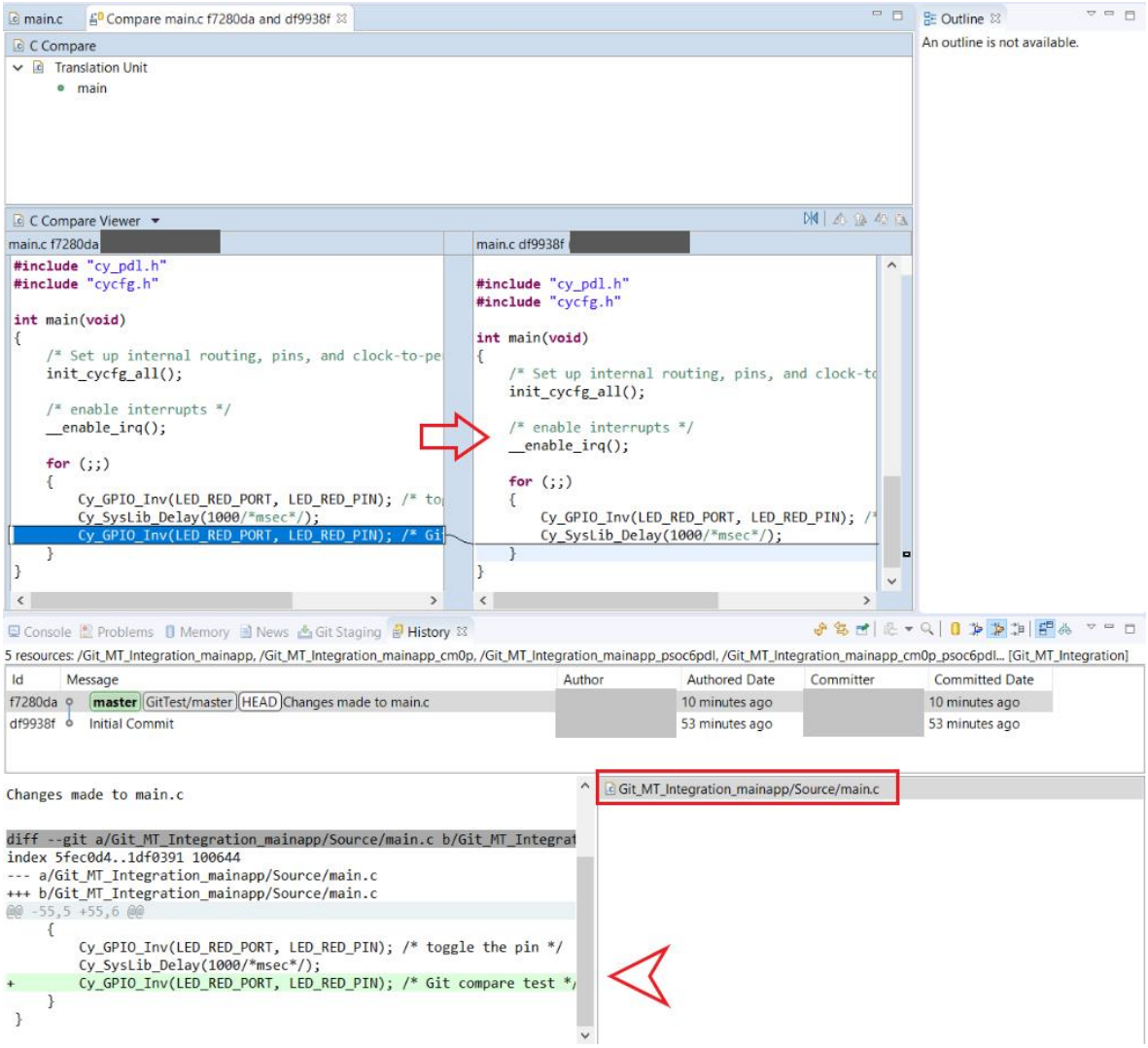
To track the changes done to the project, **right click on working set > Team > Show in history**. Selecting each commit gives details on the changes.



To compare the changes, hit the "*Compare Mode*" symbol present in the **History** window.



Double clicking the files, opens the compare window.



Lastly, to disable version control, **right click on working set > Team > Disconnect**.