

Git and GitHub for MATLAB

Baris Erkus

1 January 2015

Introduction

Git is a version control tool that is used for programming projects. It can be used locally by individual programmers and does not need to be part of a collaborative project. GitHub is for collaborative software development. If the project created in GitHub is public, then the project is open-source.

In this document, we explain one scenario for setting up Git and GitHub for MATLAB programming. There are many ways to achieve the same purpose, and these are explained at many other Internet web pages.

It is assumed that the reader is familiar with basic concepts of programming, Git, GitHub, MATLAB, command-line and Unix/Linux commands.

This document does not use the Git module provided by MATLAB, *which* is explained in <http://blogs.mathworks.com/community/2014/10/20/matlab-and-git/>.

Scenario

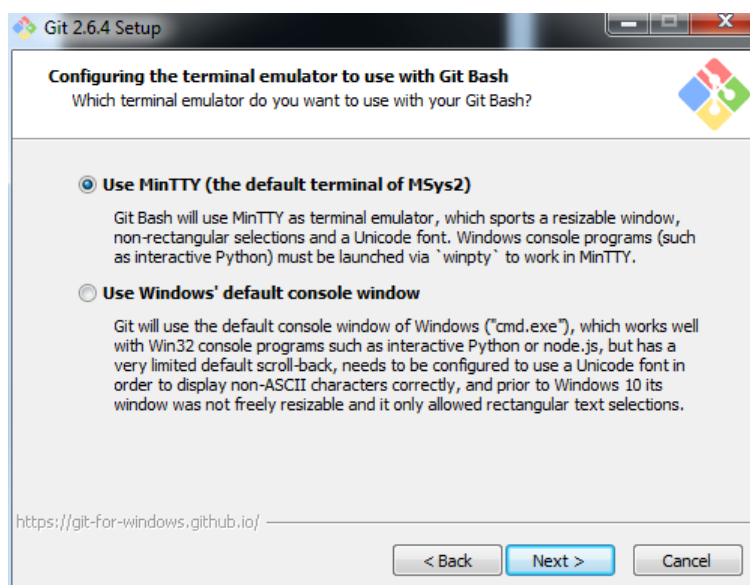
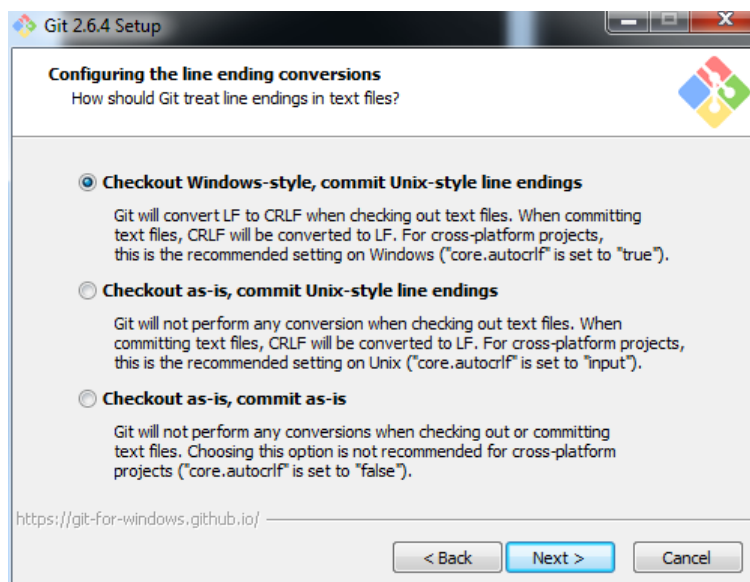
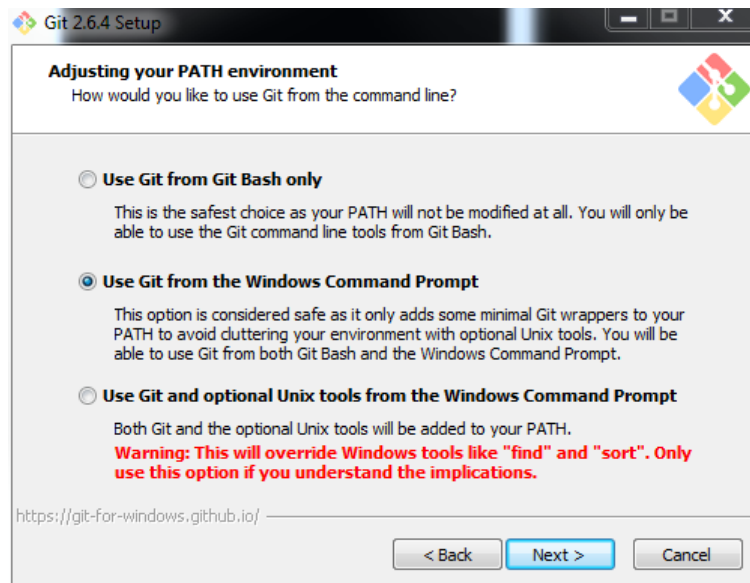
In the current scenario, a collaborative MATLAB project is developed. A lead GitHub account is setup for the main project by the project leader. Then another GitHub account is setup for the current user or contributor. If there are other users, each user should setup up their own GitHub account to contribute to the main project. Therefore, in this scenario there is a lead programmer and there are many contributor programmers under the lead programmer

This scenario can also be used for a project where one programmer exists if there is a possibility that new programmers will join to the project in the future.

In the rest of this document, all operations, except that are related to the project leader, are explained for the user.

Preliminaries

- A lead GitHub account is setup at <https://github.com> by the project leader. In this document, we call this account: bariserkusgit. The web page for this account will be <https://github.com/bariserkusgit>. Lets assume that this account is setup with an e-mail account leadgit@gmail.com.
- A user GitHub account is setup at <https://github.com> by the user. In this document, we call this account: bariserkus. The web page for this account will be <https://github.com/bariserkus>. Lets assume that this account is setup with an e-mail account usergit@gmail.com.
- User setup Git on user's local computer. We assume that the operating system is 64-bit Windows. Git can be downloaded from <https://git-scm.com/>. For 64-bit Windows, the current link is <https://github.com/git-for-windows/git/releases/download/v2.6.4.windows.1/Git-2.6.4-64-bit.exe>. It is recommended that during the setup "Use Git from the Windows Command Prompt," "Checkout Windows-style, commit Unix-style line endings" and "Use MinTTY (the default terminal of MSys2)" are selected as shown below:



- User setup basic information in Git as explained in <https://git-scm.com/book/en/v2/Getting-Started-First-Time-Git-Setup>.
- Setup user name/surname, e-mail address and the editor as follows:

```
$ git config --global user.name "Baris Erkus"
$ git config --global user.email usergit@gmail.com
$ git config --global core.editor "'C:/Program Files (x86)/Notepad++/notepad++.exe' -multiInst -nosession"
```

- Get the Git wrapper for MATLAB from the following web page: <https://github.com/manur/MATLAB-git>. Same wrapper is available at MATLAB forum as well. Place the wrapper in a folder that is in the path of MATLAB. Please note that user do not need to have this wrapper for the software versioning. User can use the Git command window for almost all the tasks that is done by the wrapper. However, wrapper provides a convenience as it runs the commands from within MATLAB. The document provided by <https://admin.kuleuven.be/icts/.../matlab/pdf/versioning-git-matlab.pdf> is very useful for understanding how wrapper is used.
- When the user wants to upload the changes made in the local computer to the user's GitHub account, the system will ask for username/password. Normally user needs to enter them each time an upload is required. If the user wants users username/password info be entered automatically, a "credential helper" can be used as explained in the page: <https://help.github.com/articles/caching-your-github-password-in-git/>. The command is as follows:

```
$ git config --global credential.helper wincred
```

- The system is ready to use the Git and GitHub.

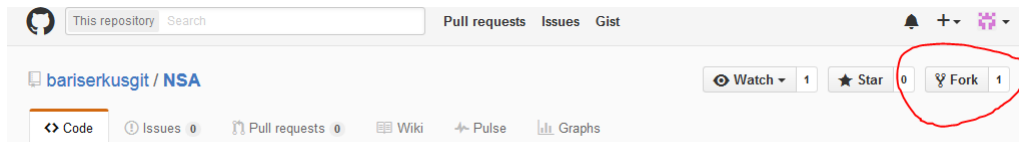
Lead GitHub Repository of the Project: Upstream

In this scenario, the leader setups the main project repository in GitHub. This repository is generally called "upstream" project. In this document the name of the project and the repository is NSA (short for Nonlinear Structural Analysis in this example). The repository creation is straightforward. After creation, the leader can place startup MATLAB files in to the repository. The project URL is <https://github.com/bariserkusgit/NSA>.

User GitHub Repository of the Project: Fork

In this scenario the user will contribute to the main project created by the leader. In this case, the user creates a copy of the upstream project. The cloned project is a "fork" of the upstream project.

- User logon to user's GitHub account.
- User goes to the leader upstream project repository web page, which in this example is <https://github.com/bariserkusgit/NSA>. On this page, user choses the "Fork" button on the upper right corner as shown below:



- Now, user will have a clone of the upstream project files on his own repository. The clone files will appear on the web page of <https://github.com/bariserkus/NSA>.

User Local Copy of the User Fork: Local Copy

User need to have a local copy of the fork on the local computer so that user can contribute user's MATLAB code to the project.

- User creates a folder on the PC. In this example, the local folder is E:\Matlab\. Note that the copy of the fork will be created as E:\Matlab\NSA\.
- Right-click to the Matlab folder and select "Git Bash Here" on the menu. A MINGW64 window will be opened and the current folder will be E:\Matlab\.
- Create a clone of the user fork using the "git clone" command as follows:

```
bariserkus@AST01 MINGW64 /e/Matlab
$ git clone https://github.com/bariserkus/NSA
```

- This will create the folder E:\Matlab\NSA\. and in that folder there will be a hidden folder named ".git".

User Branch of the Local Copy

The user local copy is called "main branch." The user needs to create a branch that is different than the main branch when working on the code. This is done by the "git checkout" command.

- Create a branch of the local copy. In this example lets call the branch ver01. The command is as follows:

```
$ git checkout ver01
```

- User develops MATLAB code. Go to MATLAB and change the folder to E:\Matlab\NSA\. Make necessary changes to the existing code or develop new code.
- The user can use the Git wrapper to execute certain Git commands as explained in the document <https://admin.kuleuven.be/icts/.../matlab/pdf/versioning-git-matlab.pdf>.
- If new M-files are developed, these files should be added to the branch. This is done by "git add" command. In this example, lets assume user added a new file called "newcode.m" to the project. Then the command on the Git Bash window is:

```
$ git add newcode.m
```

- If the contribution of the user is sufficient for uploading, the user should "commit" the changes to the current branch. This is done by "git commit" command. In the current example the command is

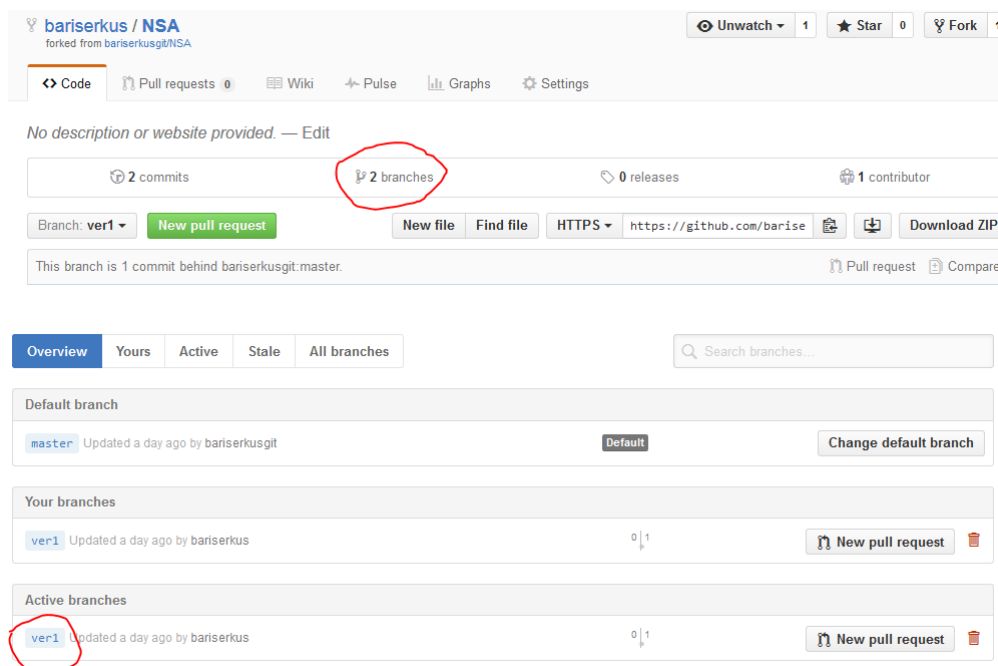
```
$ git commit
```

Uploading Local Branch to the Fork Repository

- If the user would like to propose the current branch to the lead developer, the user first needs to upload the branch to the user's fork repository. This is done by the "git push" command. The command for the current example is

```
$ git push -u origin ver01
```

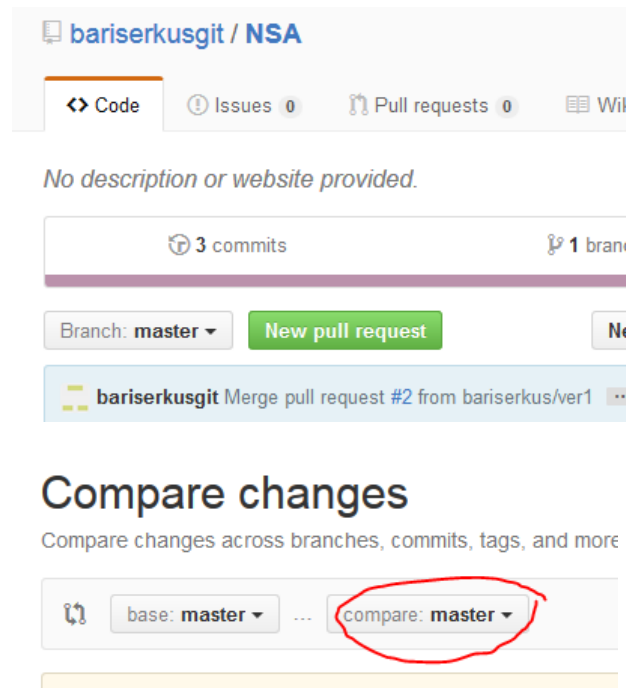
- At this point, the user should see the branch ver01 on the user's GitHub web page of the fork. This can be checked by the branch number and clicking it as shown in the following:



Requesting a Pull from the Project Leader

User needs to request a "pull" from the project leader to apply the branch to the upstream project files. Here, the user did some work on the project and he proposes to the leader that user changes are accepted and applied to the project. If the project leader considers that the changes/progress made by the user within the branch is worth to add to the main project, the pull request will be accepted.

- The user goes to the upstream project repository on the web page: <https://github.com/bariserkusgit/NSA>.
- The user selects the New Pull Request. This brings up the "Compare changes" page. At this page the upstream will be compared with the proposed branch. At the compare button, the user needs to select the proposed branch, in this case, branch name called "ver01".
- These two steps are shown in the following figures:



Accepting the Pull Request by the Leader

The leader will see the pull request on leader GitHub web page of the project repository: <https://github.com/bariserkusgit/NSA>. If the leader finds the proposed changes worth to apply to the original project, he will accept the the pull request.