GIT Installation Setup

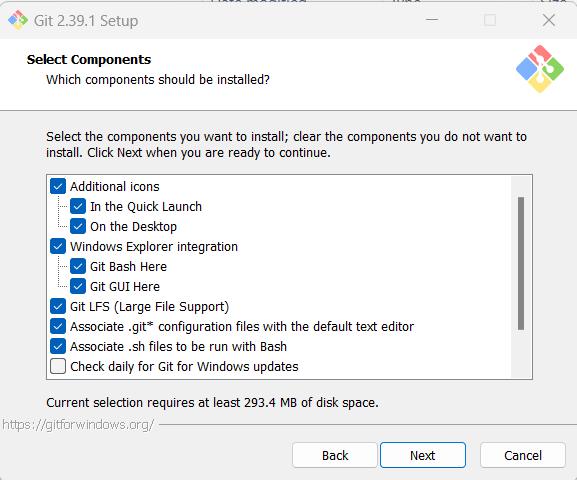
Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

* **Select Components** Make sure you leave turned on the check-box to install Git LFS (Large File Support).



Graphical user interface, text, application, email

Description automatically generated

* **Choosing the default editor used by Git** I recommend that you install Notepad++ first, and during the initial installation of Git, select that as the default editor.

Graphical user interface, text, application, email

Description automatically generated

* **Adjusting the name of the initial branch** in new repositories Override the default branch name for new repositories, and set that to main (this is what we are using for the Flux repositories).

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

* **Configuring the line ending conversions** Make sure to select Checkout as-is, commit as-is. In short, we don’t want any LF to CRLF conversions or vice versa. We have a lot of text and HTML files in our test-data set that use Unix style endings and this conversion will cause tests to fail.

Graphical user interface, text, application, email

Description automatically generated

* **Configuring the terminal emulator** to use with Git Bash I found it simpler to use Windows’ default console window than to use MinTTY.

Graphical user interface, text, application, email

Description automatically generated

* **Choose the default behavior** of "git pull" Make sure to select Only ever fast-forward. This is especially important for folks like us who are coming over from Mercurial. By default a git pull will do a merge of the incoming commit, and that is usually not what you want unless we can do a fast-forward merge. The equivalent to a hg pull is really a git fetch and then one can do a git merge appropriately. I think this is definitely better than doing silent merges of the non-fast-forward variety.

Graphical user interface, text, application, email

Description automatically generated

* **Choose a credential helper** Pick the Git Credential Manager Core – this will make it easy to log in just once to Github and you will not keep seeing name and password prompts each time.

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated