Help Topic: The Git Pull Command

Fall 2024

Michael Benjamin, mikerb@mit.edu
Oscar Viquez, oviquezr@mit.edu
Department of Mechanical Engineering
MIT, Cambridge MA 02139

The Git `pull` Command

The `git pull` command lets you refresh your locally checked out repository with any changes in the repository HEAD on the server. It also tells you *what* has been changed, added, deleted. If a change has been made to a file you have also changed locally, Git will try to merge those changes. If unsuccessful, it will report a conflict. More on this is discussed below. More info can always be found by Googling "Git book" and reading the full PDF online free, or just typing `git help pull` anytime on the command line.

Basic usage of the `git pull` command

The basic syntax for the `git pull` command is:

```
$ git pull        #(invoked somewhere within previously checked out tree)
```

The `git pull` command requires a network connection to the server and perhaps a password if it is not an anonymous read-only repository. The command may be invoked anywhere in the locally checked out tree. By default, the update will apply across the entire repository regardless of where in the tree it is called.

Deciphering the output of the `git pull` command

In the simplest case, when your locally checked out tree is up to date with the server, you will see a one-line response like the following:

```
$ git pull
Already up to date.
```

Beware of the common "gotcha". Two developers both perform an update and confirm that they have the same version - but their code is acting differently. This can drive people bonkers. The thing to remember is that, even though you both have the same version checked out, one or both of you may have local modifications that you have not yet checked into the server. Very likely one of these local modifications is causing the difference in observed behavior. Check for local modifications using the `git status` command, discussed in a separate help topic or in the Git book.

Here's an example output after an update:

```
$ git pull
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 0 (delta 0)
Unpacking objects: 100% (3/3), 270 bytes | 135.00 KiB/s, done.
From <server>:<repo>
   66051d7..15a95d0  master     -> origin/master
Updating 66051d7..15a95d0
Fast-forward
 Figure.h   | 1 +
 Object.cpp | 0
 Object.h   | 0
 3 files changed, 1 insertion(+)
 create mode 100644 Object.cpp
 create mode 100644 Object.h
```

This indicates that:

- The file `Figure.h` was updated with changes pulled in from the server. There was one line of code added to the file.
- The files `Object.h` and `Object.cpp` are newly added from the server to your local checkout. These sample files are empty, adding zero new lines of code.

If there are any files with *uncommitted* local changes which would be overwritten by a file pulled from the server, the program may inform the user of this conflict, and instruct the user to commit or stash their changes before merging changes from the server, to avoid the loss of progress.

```
$ git pull
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 3 (delta 1), reused 0 (delta 0)
Unpacking objects: 100% (3/3), 281 bytes | 140.00 KiB/s, done.
From <server>:<repo>
   18aacd8..9bc491a  master     -> origin/master
Updating 18aacd8..9bc491a
error: Your local changes to the following files would be overwritten by merge:
  README.md
Please commit your changes or stash them before you merge.
Aborting
```