GIT: Advanced Commands

GIT Default Difftool Activity:
Setting our difftool to VSCode

Brian Gorman, Author/Instructor/Trainer

Introduction

Although BASH allows us to see differences in the terminal, working in the command line can be a bit tedious. To make viewing differences more attractive, a very nice option is to use Visual Studio Code as the difftool.

Once VSCode is installed and set as the difftool, we'll be able to easily compare the differences between a couple of commits, or the difference between a commit and our working branch.

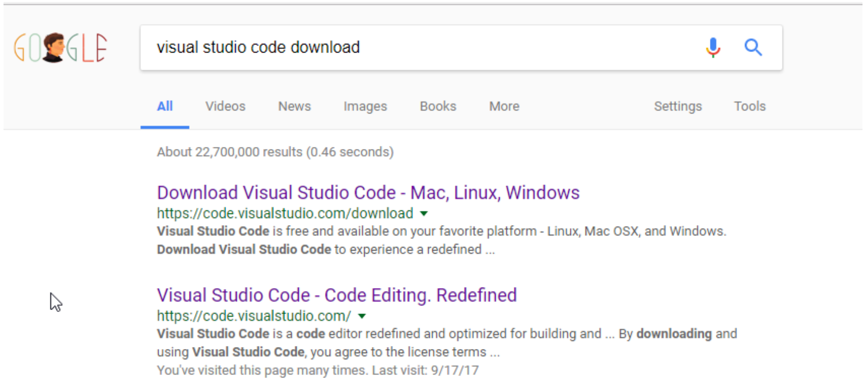
Let's gets started!

Step 1: Get Visual Studio Code Setup [if you don’t already have it]

a) Download and install Visual Studio Code onto our machine.

Go To: https://code.visualstudio.com/download

Or just do a simple google search for Visual Studio Code:



visual studio code download

About 22,700,000 results (0.46 seconds)

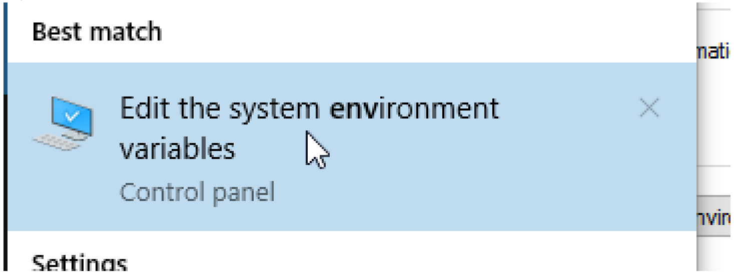
Download Visual Studio Code - Mac, Linux, Windows
https://code.visualstudio.com/download
Visual Studio Code is free and available on your favorite platform - Linux, Mac OSX, and Windows. **Download Visual Studio Code** to experience a redefined ...

Visual Studio Code - Code Editing. Redefined
https://code.visualstudio.com/
Visual Studio Code is a **code** editor redefined and optimized for building and ... By **downloading** and using **Visual Studio Code**, you agree to the license terms ...
You've visited this page many times. Last visit: 9/17/17

Install the application, start it up and make sure it works.
More information about the application can be found here:
https://code.visualstudio.com/docs

b) If on a windows machine, make sure code is in your PATH variable.

We don’t have to do this, but it will make things a lot easier. We’ll be able to reference the executable directly, rather than having to code the entire path to the executable.
Go to the start menu and type “Environment Variables” Then select “Edit System Environment variables”

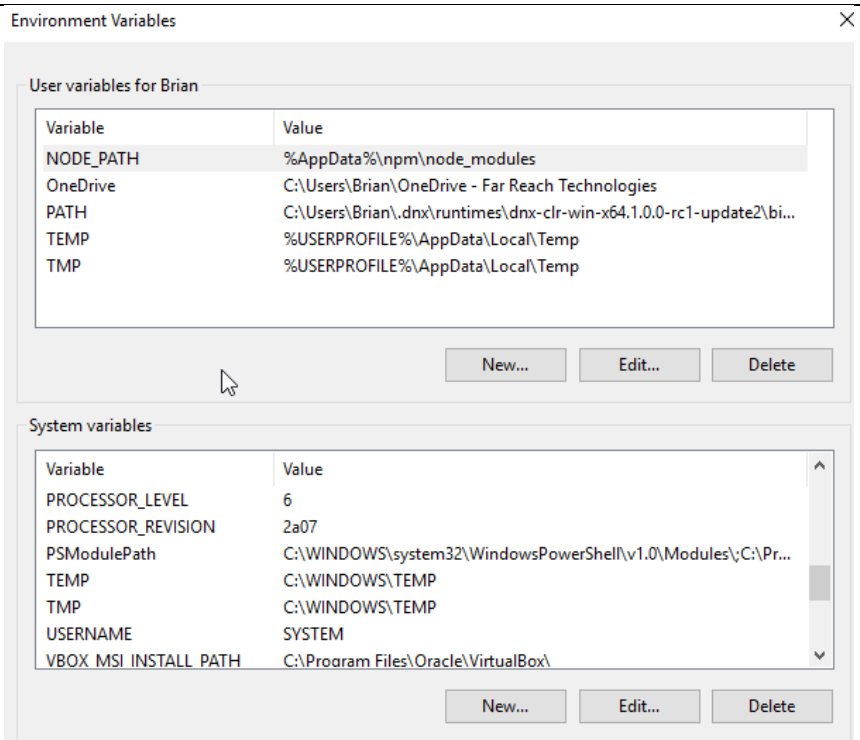


Best match

Edit the system environment variables
Control panel

Settings

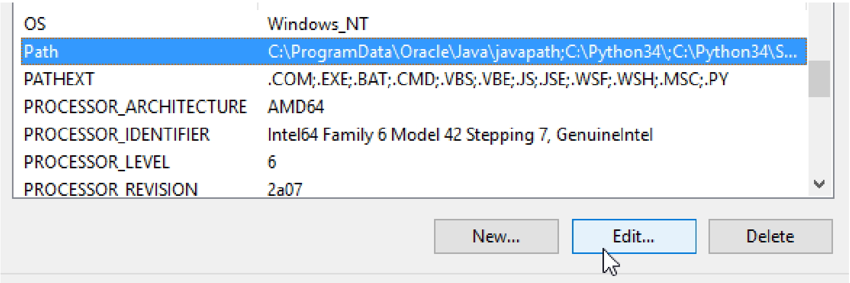
Notes



Variable	Value
NODE_PATH	%AppData%\npm\node_modules
OneDrive	C:\Users\Brian\OneDrive - Far Reach Technologies
PATH	C:\Users\Brian\.dnx\runtimes\dnx-clr-win-x64.1.0.0-rc1-update2\bi...
TEMP	%USERPROFILE%\AppData\Local\Temp
TMP	%USERPROFILE%\AppData\Local\Temp

Variable	Value
PROCESSOR_LEVEL	6
PROCESSOR_REVISION	2a07
PSModulePath	C:\WINDOWS\system32\WindowsPowerShell\v1.0\Modules\;C:\Pr...
TEMP	C:\WINDOWS\TEMP
TMP	C:\WINDOWS\TEMP
USERNAME	SYSTEM
VBOX_MSI_INSTALL_PATH	C:\Program Files\Oracle\VirtualBox\

Under 'System Variables' find "PATH"

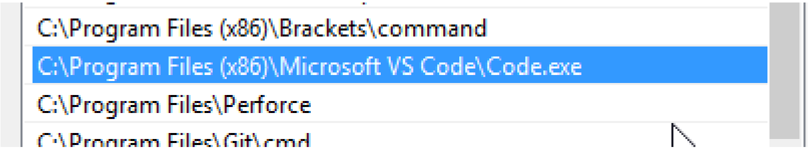


OS	Windows_NT
Path	C:\ProgramData\Oracle\Java\javapath;C:\Python34\;C:\Python34\S...
PATHEXT	.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH;.MSC;.PY
PROCESSOR_ARCHITECTURE	AMD64
PROCESSOR_IDENTIFIER	Intel64 Family 6 Model 42 Stepping 7, GenuineIntel
PROCESSOR_LEVEL	6
PROCESSOR_REVISION	2a07

Select Edit.

If you are on an older version of windows, you may need to parse the string in a text editor. It's much easier now in Windows 10:

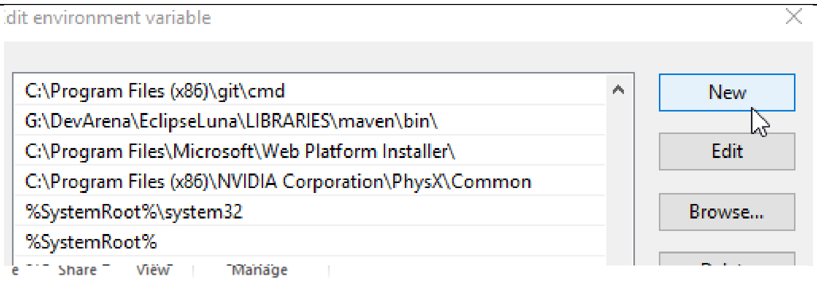
Make sure you have an entry for CODE:



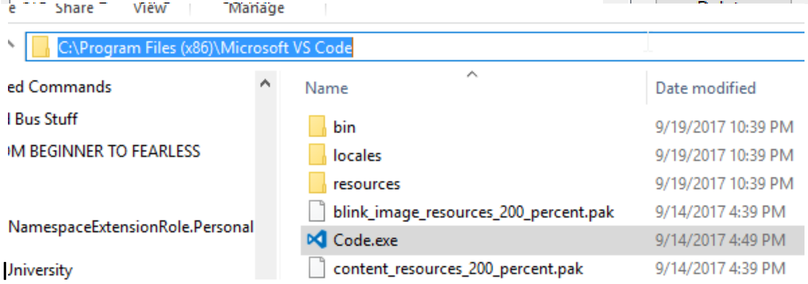
```
C:\Program Files (x86)\Brackets\command
C:\Program Files (x86)\Microsoft VS Code\Code.exe
C:\Program Files\Perforce
C:\Program Files\Git\cmd
```

If you do not, then you will want to add one using the "NEW…" button:

First find the path to your executable:



C:\Program Files (x86)\git\cmd
G:\DevArena\EclipseLuna\LIBRARIES\maven\bin\
C:\Program Files\Microsoft\Web Platform Installer\
C:\Program Files (x86)\NVIDIA Corporation\PhysX\Common
%SystemRoot%\system32
%SystemRoot%



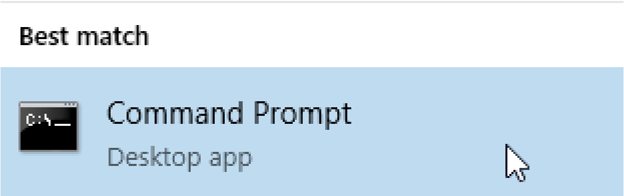
C:\Program Files (x86)\Microsoft VS Code

Name	Date modified
bin	9/19/2017 10:39 PM
locales	9/19/2017 10:39 PM
resources	9/19/2017 10:39 PM
blink_image_resources_200_percent.pak	9/14/2017 4:39 PM
Code.exe	9/14/2017 4:49 PM
content_resources_200_percent.pak	9/14/2017 4:39 PM

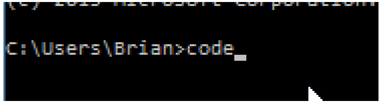
Then place that path into your Environment variables.

c) Test that it works:

Go to the start menu and type ‘cmd’
Then select “Command Prompt”



In the command prompt, type “code”:



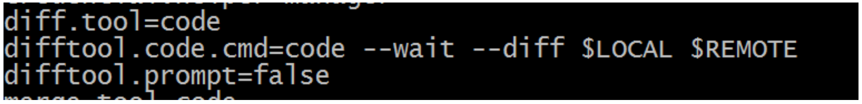
```
C:\Users\Brian>code_
```

If VS Code launches, you are all set. If not, you can either try again or just use the full path from above when setting values in the git config for difftool.

Step 2: Go to any repository

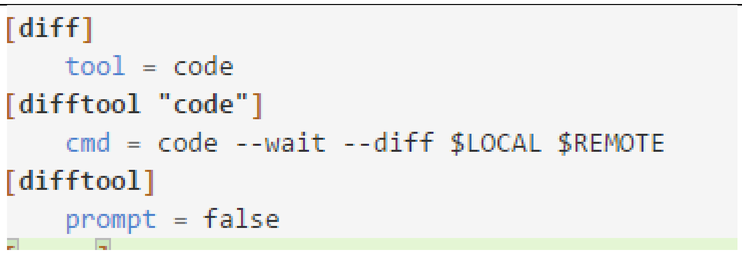
a) Make sure you are on any working repository. It doesn’t even have to be up to date. We are not going to be affecting anything.

Our goal is to get our global config to have three entries for difftool:



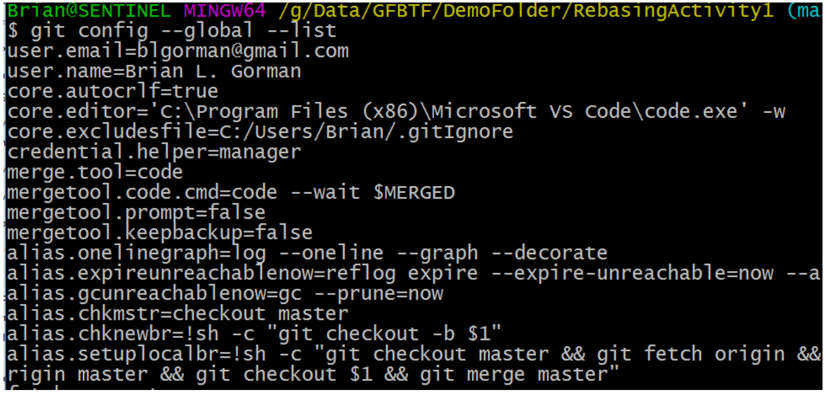
```
diff.tool=code
difftool.code.cmd=code --wait --diff $LOCAL $REMOTE
difftool.prompt=false
```

Additionally, if we were to look at our file with our editor, we should see entries like this:



```
[diff]
    tool = code
[difftool "code"]
    cmd = code --wait --diff $LOCAL $REMOTE
[difftool]
    prompt = false
```

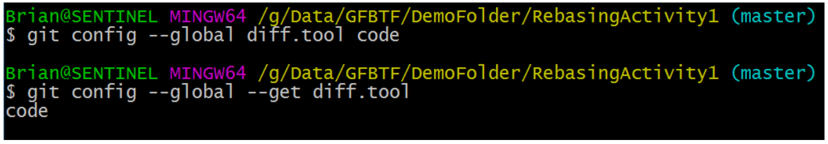
On the working repository
[git config --global --list]
//I've removed it for now to do this activity:



```
Brian@SENTINEL MINGW64 /g/Data/GFBTF/DemoFolder/RebasingActivity1 (ma
$ git config --global --list
user.email=blgorman@gmail.com
user.name=Brian L. Gorman
core.autocrlf=true
core.editor='C:\Program Files (x86)\Microsoft VS Code\code.exe' -w
core.excludesfile=C:/Users/Brian/.gitIgnore
credential.helper=manager
merge.tool=code
mergetool.code.cmd=code --wait $MERGED
mergetool.prompt=false
mergetool.keepbackup=false
alias.onelinegraph=log --oneline --graph --decorate
alias.expireunreachablenow=reflog expire --expire-unreachable=now --a
alias.gcunreachablenow=gc --prune=now
alias.chkmstr=checkout master
alias.chknewbr=!sh -c "git checkout -b $1"
alias.setuplocalbr=!sh -c "git checkout master && git fetch origin &&
rigin master && git checkout $1 && git merge master"
```

b) Set and verify the difftool variable in our global config

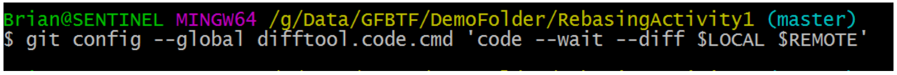
[git config –global diff.tool <variable>]
[git config --global --get diff.tool]



```
Brian@SENTINEL MINGW64 /g/Data/GFBTF/DemoFolder/RebasingActivity1 (master)
$ git config --global diff.tool code

Brian@SENTINEL MINGW64 /g/Data/GFBTF/DemoFolder/RebasingActivity1 (master)
$ git config --global --get diff.tool
code
```

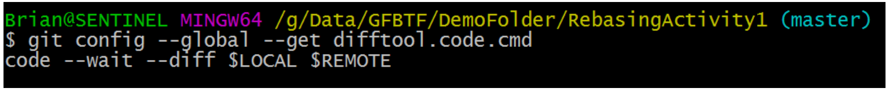
c) Set the actual execution command for code as the difftool, and verify:

[git config --global difftool.code.cmd 'code --wait --diff $LOCAL $REMOTE']



```
Brian@SENTINEL MINGW64 /g/Data/GFBTF/DemoFolder/RebasingActivity1 (master)
$ git config --global difftool.code.cmd 'code --wait --diff $LOCAL $REMOTE'
```

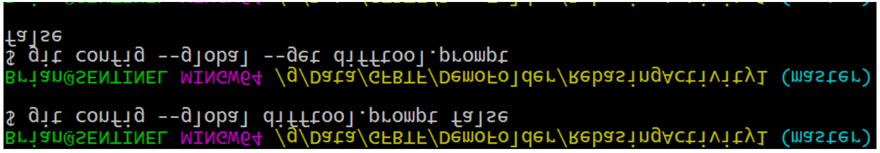
[git config --global --get difftool.code.cmd]



```
Brian@SENTINEL MINGW64 /g/Data/GFBTF/DemoFolder/RebasingActivity1 (master)
$ git config --global --get difftool.code.cmd
code --wait --diff $LOCAL $REMOTE
```

d) Optional, turn off the annoying prompt for every diff to open:

[git config –global difftool.prompt false]

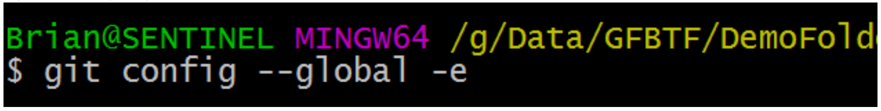


```
Brian@SENTINEL MINGW64 /g/Data/GFBTF/DemoFolder/RebasingActivity1 (master)
$ git config --global difftool.prompt false

Brian@SENTINEL MINGW64 /g/Data/GFBTF/DemoFolder/RebasingActivity1 (master)
$ git config --global --get difftool.prompt
false
```

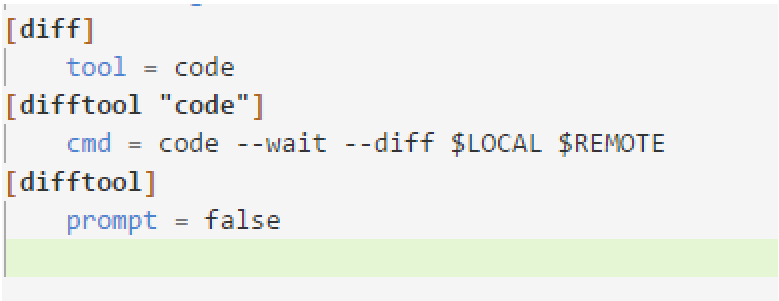
Step 3: Review the actual file entries

a) Open the file and find the entries

[git config --global –e]



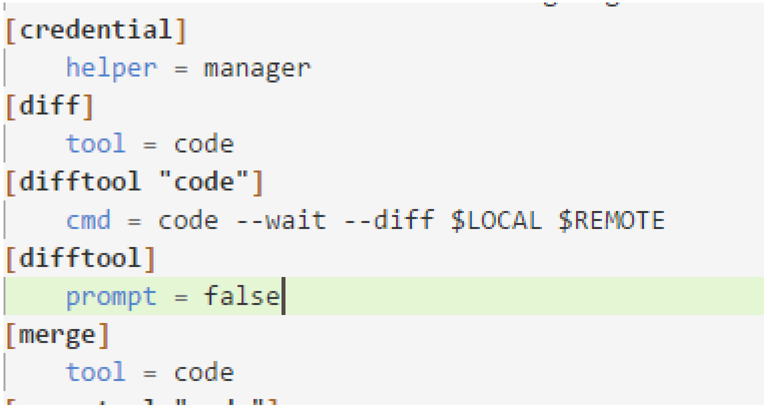
```
Brian@SENTINEL MINGW64 /g/Data/GFBTF/DemoFold
$ git config --global -e
```



```
[diff]
    tool = code
[difftool "code"]
    cmd = code --wait --diff $LOCAL $REMOTE
[difftool]
    prompt = false
```

b) Optional – Move the entries in the file.

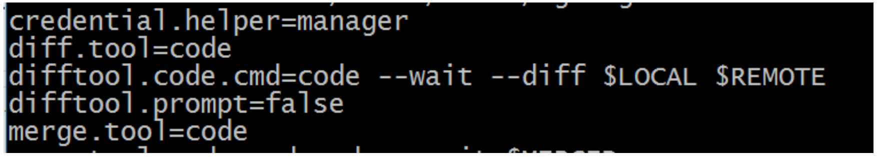
I don't want these on the bottom. I'm going to move them to be between credential manager and merge tool {don't worry if you don't have a mergetool set yet…that's another activity…}



```
[credential]
    helper = manager
[diff]
    tool = code
[difftool "code"]
    cmd = code --wait --diff $LOCAL $REMOTE
[difftool]
    prompt = false
[merge]
    tool = code
```

The order makes absolutely no difference, it is up to you what you want it to be. Just remember that this order determines what you see on --list from global config.

[git config --global --list]



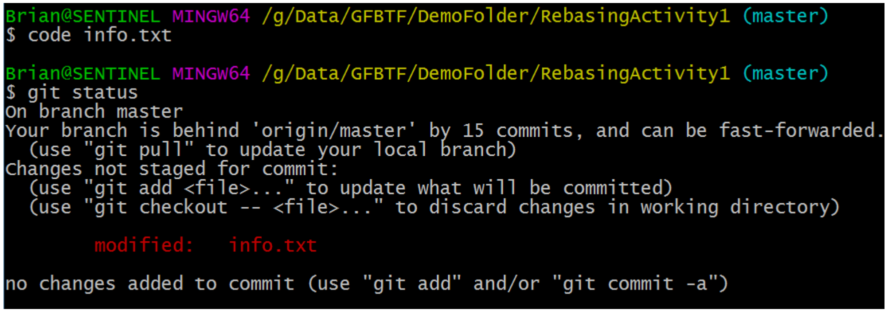
```
credential.helper=manager
diff.tool=code
difftool.code.cmd=code --wait --diff $LOCAL $REMOTE
difftool.prompt=false
merge.tool=code
```

Step 4: Make sure the difftool works!

a) Need to be on a repository, change any file

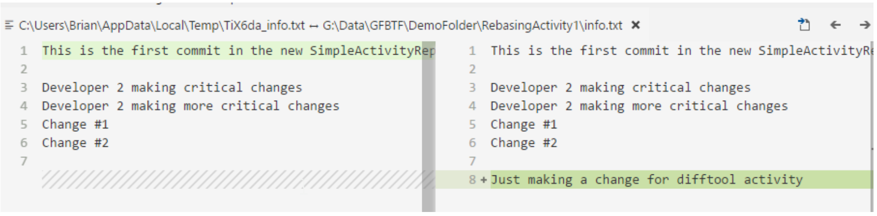
[code info.txt] or [vim info.txt] if you haven't setup code as default editor.

[git status]



```
Brian@SENTINEL MINGW64 /g/Data/GFBTF/DemoFolder/RebasingActivity1 (master)
$ code info.txt

Brian@SENTINEL MINGW64 /g/Data/GFBTF/DemoFolder/RebasingActivity1 (master)
$ git status
On branch master
Your branch is behind 'origin/master' by 15 commits, and can be fast-forwarded.
  (use "git pull" to update your local branch)
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

        modified:   info.txt

no changes added to commit (use "git add" and/or "git commit -a")
```

[git difftool head]



C:\Users\Brian\AppData\Local\Temp\TiX6da_info.txt ↔ G:\Data\GFBTF\DemoFolder\RebasingActivity1\info.txt

```
This is the first commit in the new SimpleActivityRep

Developer 2 making critical changes
Developer 2 making more critical changes
Change #1
Change #2

```

```
This is the first commit in the new SimpleActivityR

Developer 2 making critical changes
Developer 2 making more critical changes
Change #1
Change #2

+ Just making a change for difftool activity
```

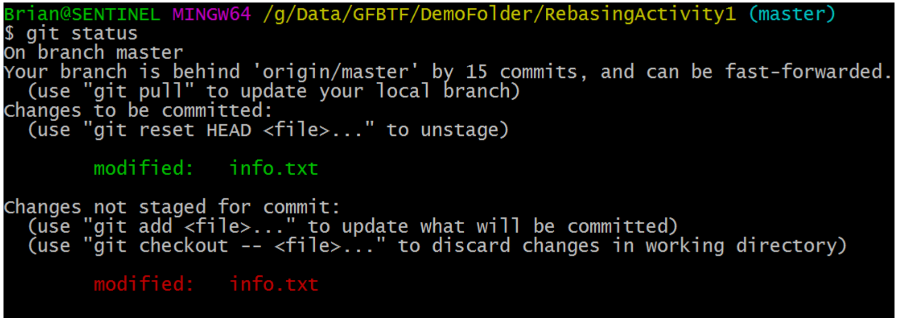
It's working!

b) Show modified and staged changes in difftool

[git add .]

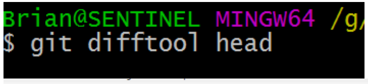
[code info.txt] or [vim info.txt]

[git status]

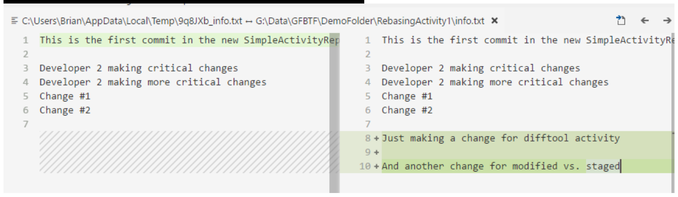


```
Brian@SENTINEL MINGW64 /g/Data/GFBTF/DemoFolder/RebasingActivity1 (master)
$ git status
On branch master
Your branch is behind 'origin/master' by 15 commits, and can be fast-forwarded.
  (use "git pull" to update your local branch)
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

        modified:   info.txt

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

        modified:   info.txt
```

[git difftool head]



```
Brian@SENTINEL MINGW64 /g/
$ git difftool head
```

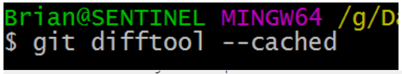


C:\Users\Brian\AppData\Local\Temp\9q8JXb_info.txt ↔ G:\Data\GFBTF\DemoFolder\RebasingActivity1\info.txt

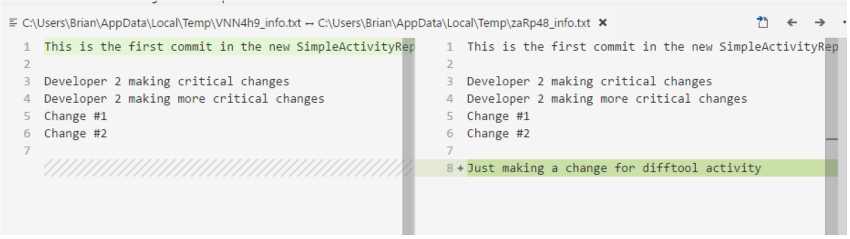
```
This is the first commit in the new SimpleActivityRep

Developer 2 making critical changes
Developer 2 making more critical changes
Change #1
Change #2

```

```
This is the first commit in the new SimpleActivityR

Developer 2 making critical changes
Developer 2 making more critical changes
Change #1
Change #2

+ Just making a change for difftool activity
+
+ And another change for modified vs. staged
```

[git difftool --cached]



```
Brian@SENTINEL MINGW64 /g/D
$ git difftool --cached
```



C:\Users\Brian\AppData\Local\Temp\VNN4h9_info.txt ↔ C:\Users\Brian\AppData\Local\Temp\zaRp48_info.txt

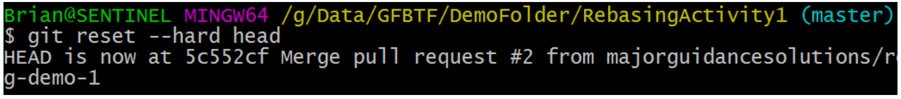
```
This is the first commit in the new SimpleActivityRep

Developer 2 making critical changes
Developer 2 making more critical changes
Change #1
Change #2

```

```
This is the first commit in the new SimpleActivityRep

Developer 2 making critical changes
Developer 2 making more critical changes
Change #1
Change #2

+Just making a change for difftool activity
```

c) Here we can either just commit the changes or reset back to where we were.

If you are on an important repo, don’t run this command, instead, just check in your changes or undo them manually. If you are in a playground, then run the following

[git reset --hard head]



```
Brian@SENTINEL MINGW64 /g/Data/GFBTF/DemoFolder/RebasingActivity1 (master)
$ git reset --hard head
HEAD is now at 5c552cf Merge pull request #2 from majorguidancesolutions/r
g-demo-1
```

[we’ll cover resetting in a later activity]

This concludes our GIT difftool Activity.

Closing Thoughts

Setting VSCode [or another tool] to use for the difftool is very powerful and effective. If you don't mind seeing the changes in the console, you don't need to do this. Also note even with the difftool you can still use the console by just using the default 'diff' command rather than the difftool command we setup in this activity.

Take a few minutes to make some notes about the various commands we've learned about in this activity, and practice using them.

Notes

____

____

____

____

____

____

____

____

____

____

____

____