**Instruction**

**Auto Install (tested)**

1. Run “install.bat” with administrator. (right-click -> run as administrator)
2. Run “autcomplete-1-run-only.bat” 1 time only, no need to run again when updating new version. Unless asked to do so.
3. Reset Git console

**Manual**

1. Copy 3 scripts file: **gc,gg** and **gm** to folder below

“C:\Program Files\Git\mingw64\bin”

1. To turn on branch name autocomplete

* Go to “C:\Users\UserName”
* Copy “gc-completion.bash” there
* Append “.bash\_profile” with the text below

“source ~/gc-completion.bash”

1. Reset Git console
2. **Show help**

For example:

* gc -h
* gg -h
* gm -h

1. **Automatic checkout new branch**

* gc full\_branch\_name

**or**

* gc jobnumber- then TAB

For safety, you need to confirm continue. Current files could be lost if they haven’t been committed.

1. **Automatic checkout / commit /merge with option to rebase against specific branch**

* **gc** branch\_name **-m** “master\_branch”
* **gg -m** “master\_branch”
* **gm** branch\_name **-m** “master\_branch”

1. **Automatic get old commit back with option rebase/stash**

* gc branch\_name **-o**
* Type “Hard (**h**) or Soft (**s**) or No (**n**)” to confirm how you want to reset to get old commit back
* Choose one of the options: Rebase(r) or Stash (s) or No (n) to continue

1. **Automatic commit**

* check files you want to commit before doing anything
* type "**gg**" / or / "**gg -p patch\_job\_number**” if it is a patch job
* **manually enter** the commit message **/ OR /** press “enter” to use suggestion

(the suggestion comes from the job name)

* do rebase (squash, fixup...)
* type "**git push**"

(must do it **manually**, make sure everything looks right, don’t blame me 😊)

1. **Semi-automatic code review (code review only)**

* **gm branch\_name** [ -m master\_branch]
* Review the changes in Visual Studio
* Confirm to continue rebase against master branch
* Solve conflicts if any
* Confirm to merge changes into master branch
* Confirm push
* Check current HEAD