UHEAA – GitHub (<https://github>)

# Install Bash Git

Download URL:

* <https://github.com/git-for-windows/git/releases/download/v2.6.3.windows.1/Git-2.6.3-32-bit.exe>

## Install Options:

* Select Components: default
* Path Environment: Git and optional Unix tools
* Line Ending Conversions: Checkout as-is, commit as-is
* Terminal Emulator: MinTTY
* Performance Tweaks: Enable File System Caching

# Setup your SSH key

This ensures secure and uniquely identifiable communication between your machine and GitHub.

Launch Git Bash and follow Steps 1 – 3 here:

<https://help.github.com/enterprise/2.3/user/articles/generating-ssh-keys/>

Use this command for step 3.1: eval $(ssh-agent –s) or if that fails to return a process id, use eval `ssh-agent -s`

Paste the contents of your **id\_rsa.pub** file (copied to your clipboard in step 4) into and email and send it to me. Make the subject contains your name and machine id: E.g. Eric Desktop - LPP-2543

Wait for Eric to add your ssh key to GitHub.

# Configuration

Run three configuration commands below substitution your information:

git config –-global http.sslverify false

git config user.name “John Doe”

git config user.email [Jdoe@utahsbr.edu](mailto:Jdoe@utahsbr.edu)

Commands can be aliased through Git configurations or the bash.bashrc file if desired.

## Git:

E.g. git config --global alias.st “status”

Typing git st is now the same as typing git status

## Bash:

Edit your C:\Program Files\Git\etc\bash.bashrc file adding your desired aliases to the bottom of the file.

E.g.

alias gst="git status"

alias gco="git checkout"

alias gcm="git commit -m"

Once you’ve restarted Git Bash, typing gst will be the same as typing git status

# Clone Projects from GitHub

Make a folder on your desktop called “Projects”

From Git Bash change directories to your newly created “Projects” folder and run the following commands:

git clone <https://github/UHEAA/SSRS-Reporting.git>

change to the SSRS-Reporting folder and run the following command:

git remote add origin https://github/UHEAA/SSRS-Reporting.git

git push --set-upstream origin master

## Repeat for each project:

* <https://github/UHEAA/Queries.git>
* <https://github/UHEAA/Applications.git>
* <https://github/UHEAA/CommonLibraries.git>

# Migrating a Solution from SVN to a GitHub Project

Right-click on the solution you wish to export > TortoiseSVN > Export

Move the exported script folder to your new projects folder: E.g. /Desktop/Projects/Applications/<SCRIPT\_ID>

Use SVN-rename to add an underscore to the beginning of the folder name. This is to signal that the solution’s source code is no longer being managed within SVN and Git should be used going forward. E.G. …\UHEAA\StandAloneApplications\\_CodeSearch\

git add .

git commit -m "Initial commit of CodeSearch (migration from SVN)"

git push origin master

**Branches**

All development should be done on a branch. The branch name should match the request type followed by an underscore and then the request number. E.g. NH\_1234, SR\_1234, SASR\_1234

Using the –b option will both create the branch and checkout the branch.

E.g. git checkout –b SR\_1234

This will create a new branch and checkout that branch for you.

Your development work should be committed (with a comment) to your local repository and push to GitHub at the end of every day at a minimum. More frequent commits and pushes are encouraged. The work you complete on a branch should be pushed to GitHub on a branch of the same name. E.g. git push origin SR\_1234

The request should be merged back into the master branch and pushed to GitHub once development is complete and ready for promotion.

I.e.

git checkout master

git merge SR\_1234

git push origin master