GitHub is a web-based hosting service for projects that use the Git revision control system. It is written using Ruby on Rails by GitHub, Inc

This document describes:

- How to set-up a GitHub account for access to OPBM.
- How to create a copy of the repository locally.
- How to download the latest build for testing.
- How to track and add bugs/issues.

In order to…	You will…
Set-up a GitHub account	Create a GitHub account. Generate a personal private/public key to identify yourself for secure Git transactions.
Copy a repository locally for development	Install TortoiseGit. Generate TortoiseGit private/public key with PuTTYgen. Copy a repository with the **Gitclone** function in Windows explorer.
Download the latest build for testing	Download the latest build from your GitHub account.
Track/create issues	Use the GitHub Track Issues …..

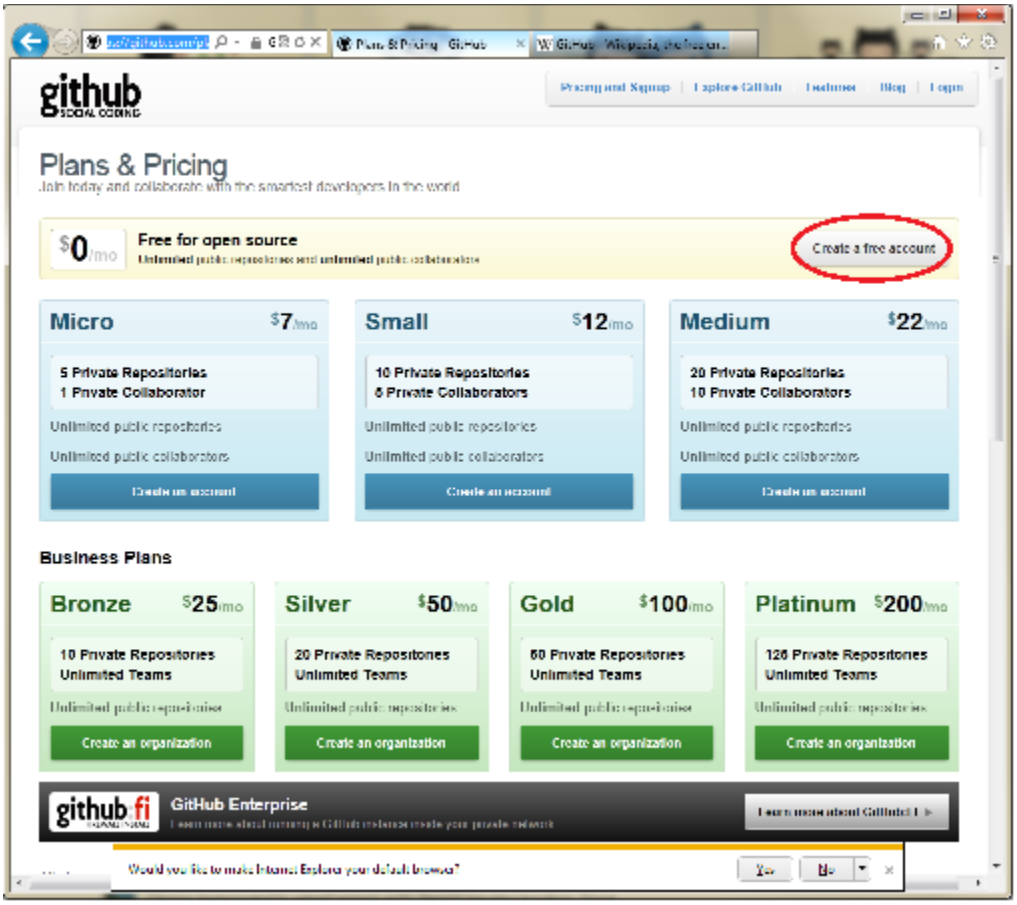
You will be installing and working with the following applications on your computer.

- MSysGit
- TortoiseGit (The Git version of TortoiseSvn)
- Puttygen (Puttygen is automatically installed as part of the TortiseGit installation)

Setting up a GitHub Account

Creating a GitHub Account

1. Go to https://github.com/plans
2. Click on **Create a free account**.



3. Enter your Username, Email Address and Password.
4. Click on the **Create an account** button. You do **not** need to enter an SSH public key at this time.
5. Send your GitHub Username to van@canalabs.com. Van (GitHub user *van-smith*) will add you as an OPBM collaborator.

Generating a personal private/public key to for secure Git transactions

GitHub uses the Git distributed version control system. Git has gained a great deal of momentum recently at the expense of SVN, mainly because Git is often much faster than SVN. However, Git is more complicated to set up.

Git requires that you create an SSH private/public key pair to identify yourself for secure Git transactions. You will need to produce these keys before you can use GitHub.

1. Download MSysGit from one of this sites

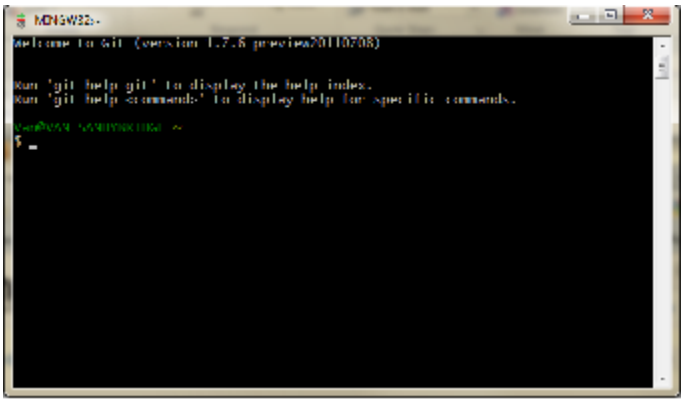
   http://code.google.com/p/msysgit/downloads/detail?name=Git-1.7.6-preview20110708.exe&can=2&q=

   http://code.google.com/p/msysgit/downloads/list

   OR follow the instructions here:

   https://git.wiki.kernel.org/index.php/MSysGit:InstallMSysGit

2. Install MSysGit. Install with default settings.
3. Click on the **Start** menu and find the Git submenu. Click on Git/Git Bash. A command window will appear.



4. Type the command:

   ```
   ssh-keygen -C "myemailaddress@amd.com" -t rsa <enter>
   ```

   Of course, replace `myemailaddress` with your own email address. The command generates your SSH private/public key pair for Git.

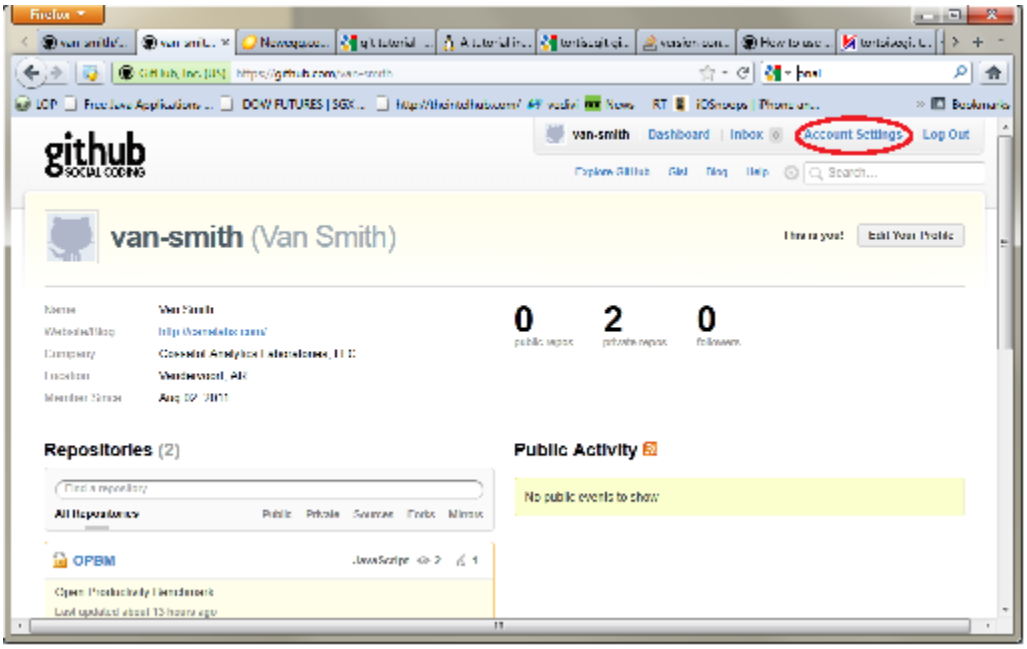
5. `ssh-keygen` will ask you for a file path. Press <enter> to accept the default directory.

   Your private/public key pair should now be in the `.ssh` directory located in your home directory. For instance, my keys are here:

`C:\Users\Van\.ssh`

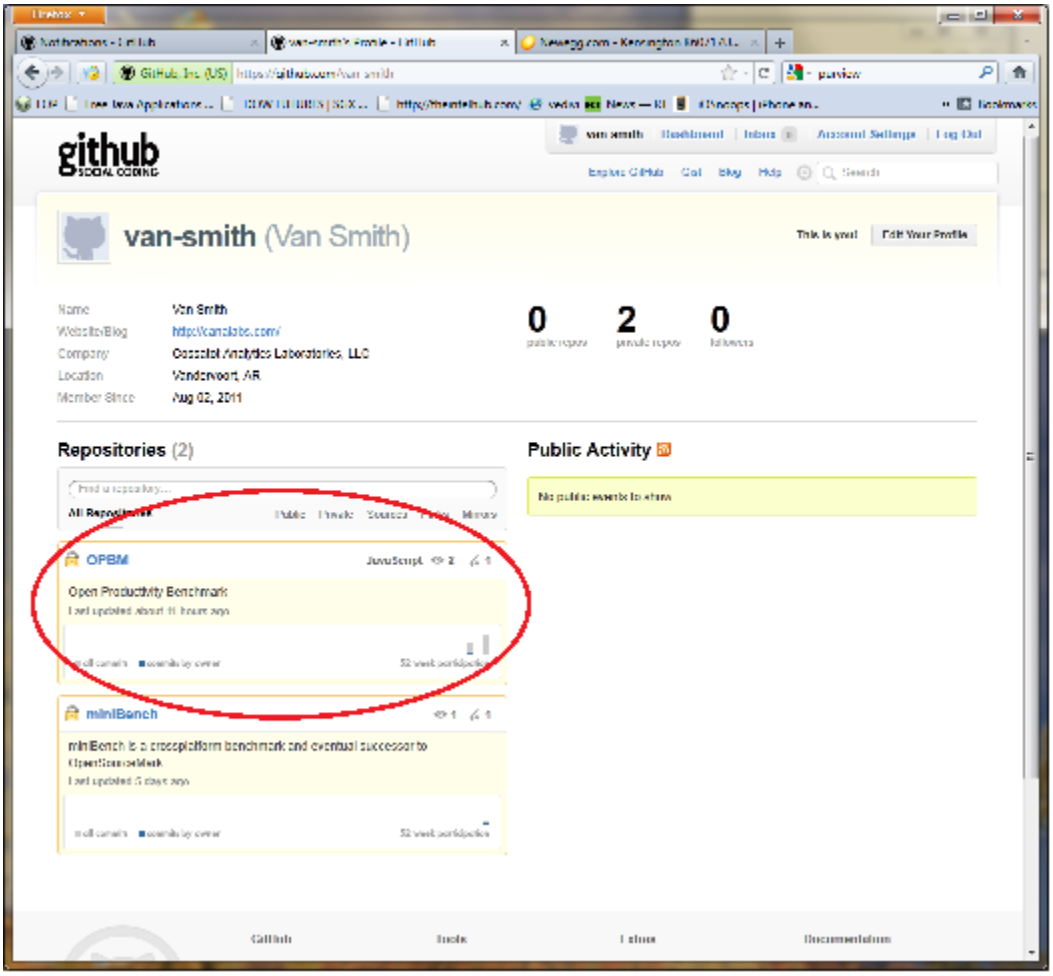
6. At the prompt, type in a passphrase or press <enter> . You will need to enter your passphrase often for Git transactions, but you can avoid this by setting up a blank passphrase; do this only if you are sure your computer is secure.
7. Open File Explorer. Go to `c:\users\"your name"\.ssh`
8. Right-click on `id-rsa.pub` and open with a text editor.
9. Select the text and copy it.

10. From your GitHub account, click on **Account Settings**. From the menu, click on **SSH Public Keys**

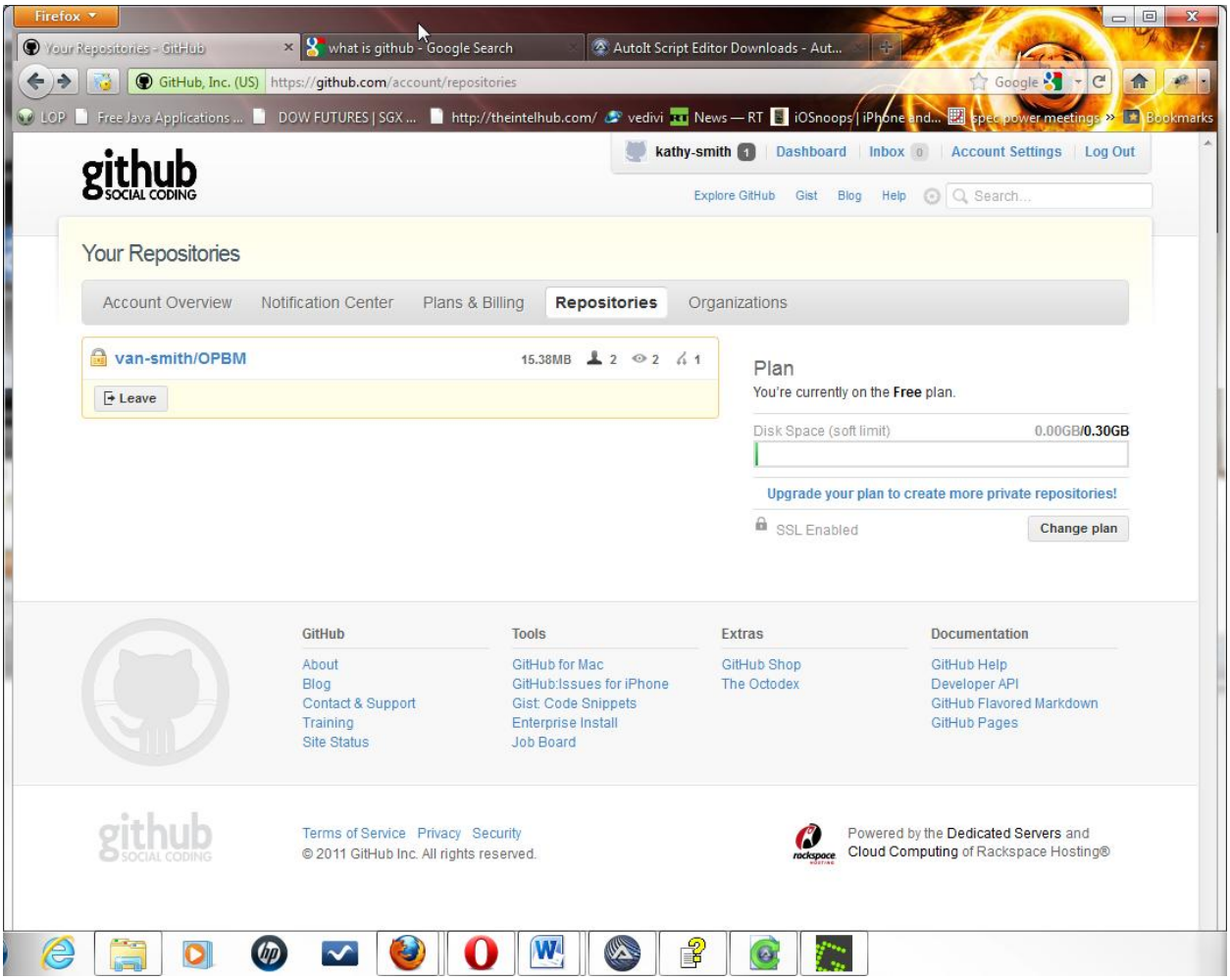


11. Click **Add another public key**.
12. Paste the public key into the **Key** text box. Type "OPBM" or anything you want into the **Title** text box.
13. Your GitHub account should now be ready to work with the OPBM repository pending that I have added you as a collaborator.

Your repositories will look like this on your GitHub home screen.



After I have added you as an OPBM collaborator, you might not see the OPBM repository until you navigate to **Account Settings | Repositories**. Your Repository view will look like this:



Copying a repository locally for development

Installing TortoiseGit

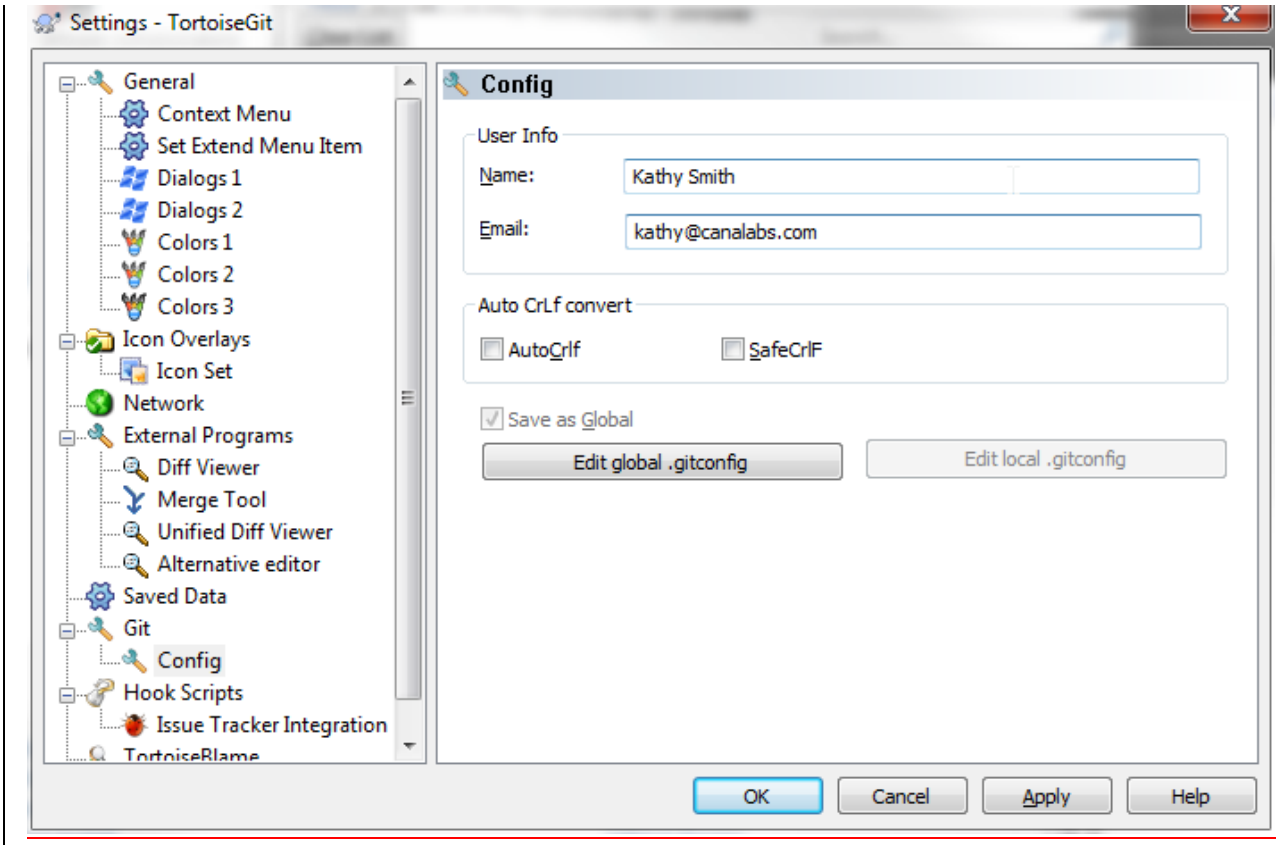
There is a Git version of *TortoiseSvn* called, unsurprisingly, *TortoiseGit* and I encourage you to use it. TortoiseGit allows you to push and pull file changes from a contextual menu inside of Windows File Manager.

However, there are installation issues that can be tricky. Here's how to set up TortoiseGit for your GitHub account.

1. Download **TortoiseGit** from here (be sure to install the correct version for 32-bit or 64-bit Windows):

   http://code.google.com/p/tortoisegit/

2. Install **TortoiseGit**. Use default settings during the installation.
3. After **TortoiseGit** is installed, go to **Start | All programs | TortoiseGit | Settings.**
4. When the application opens, go to **Git | Config** in the tree view on the left of the window. Enter your name and email in the **Config** box like Kathy did below and click **OK**.



Converting your key with PuTTYgen

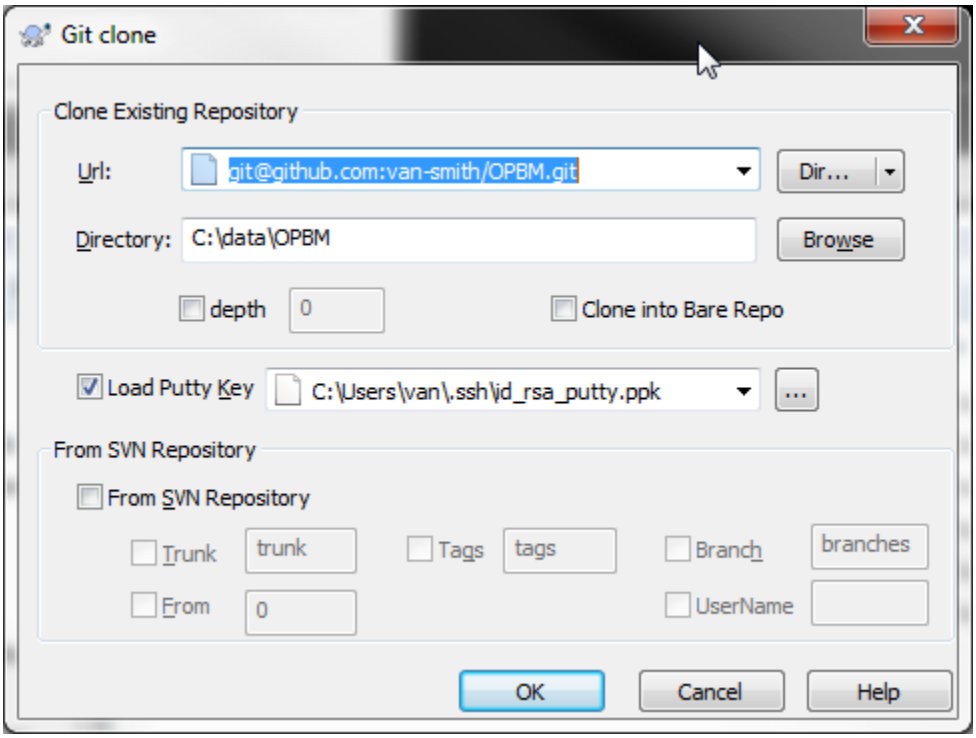
TortiseGit uses **PuTTY** by default which can't read your SSH files. You will need to convert your keys.

1. From the **Start** menu, click on the **TortoiseGit** submenu.
2. Click on **Puttygen**.
3. From Puttygen, click on **Conversions**/**Importkey**. Navigate to your `.ssh` directory.
4. Load the private key file *id_rsa* located in your `.ssh` directory; you will be prompted to enter your passphrase here.
5. Click on the **Save private key** button on the **PuTTY Key Generator** window. Save the private PuTTY key to the same `.ssh` directory. Name the file *id_rsa_putty*. A *.ppk* extension will automatically be added to the file name.
6. Click on **the Save public key** button. Save the public PuTTY key to the same `.ssh` directory. Name the file *id_rsa_pub_putty*. A file type extension will not be generated for this file.
7. Close PuTTY Key Generator.
8. Click on **Apply** in TortoiseGit window.

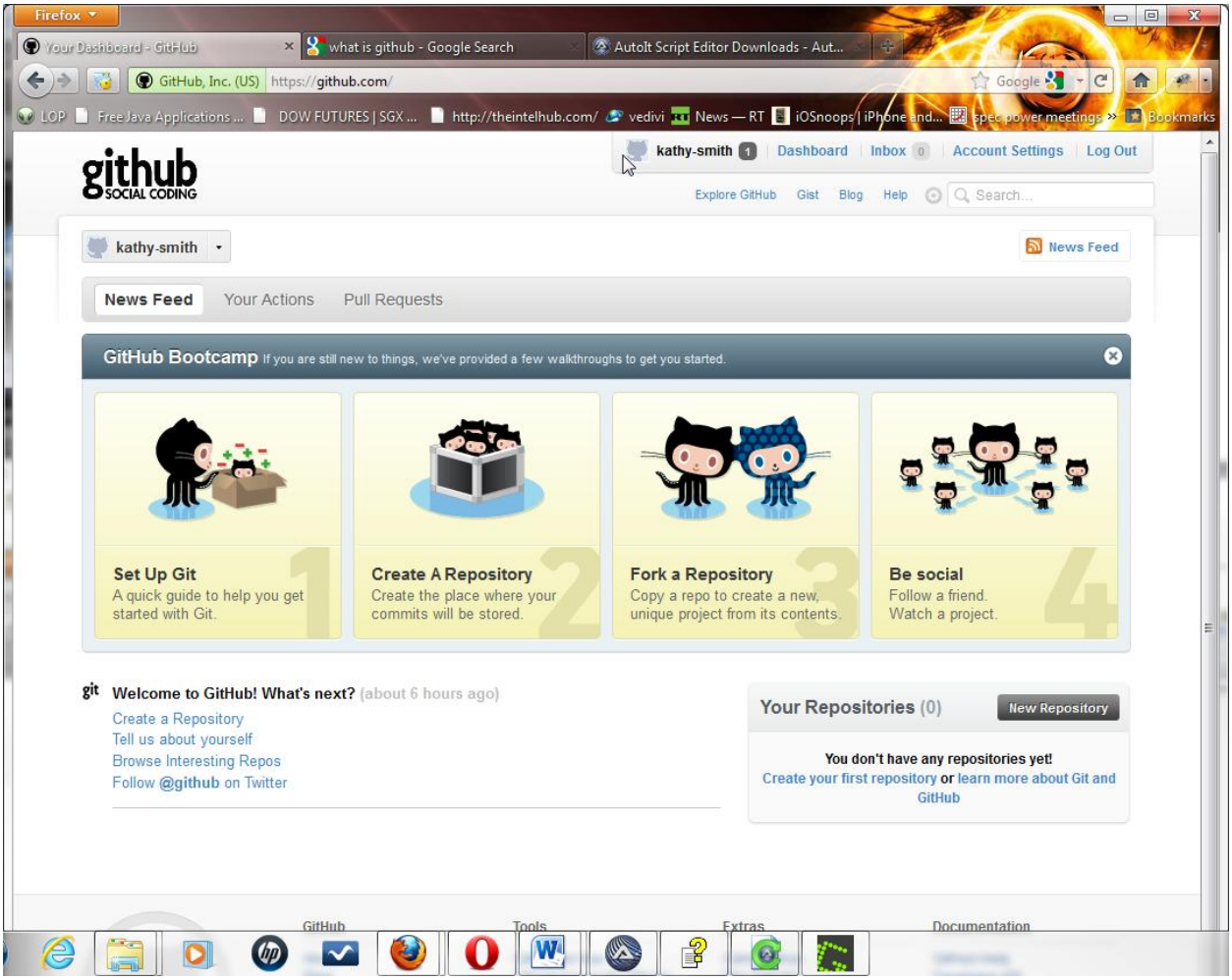
You will use your new PuTTY keys in the next step.

Copy a repository with the Gitclone function in Windows explorer

1. Open Windows File Explorer. Go to the directory where you want to create a copy of the repository.
2. Right-click in the window. A pop-up menu appears.
3. Click on **Git clone**. A **Git clone** window appears.

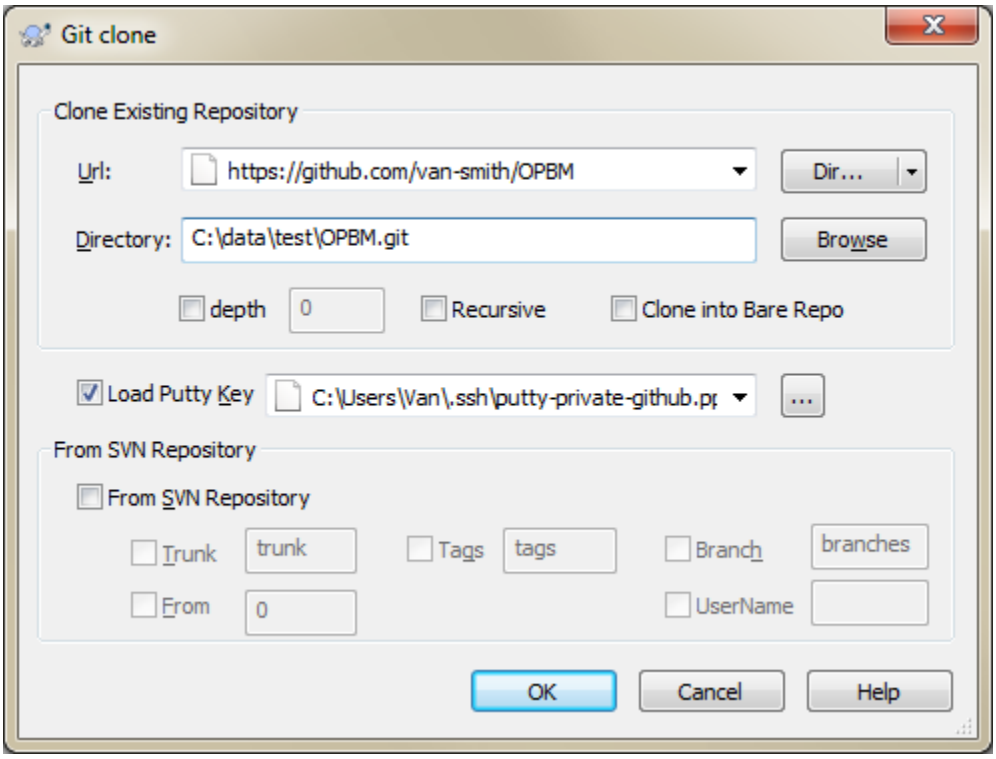


4. Log-on to your **github** account.



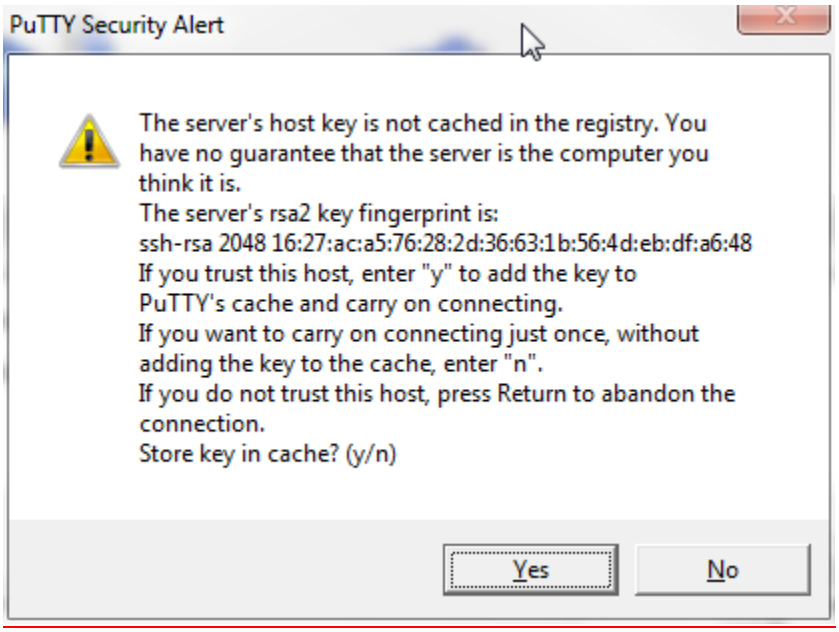
5. Click on **Account Settings | Repositories**. **van-smith/OPBM** repository should appear.
6. Click on **van-smith/OPBM**. The **van-smith/OPBM** repository appears.
7. Copy the SSH path from there or use this:
   `git@github.com:van-smith/OPBM.git`
8. Go back to the **Git clone** window. Paste the SSH path into the textbox labeled **Url**:.
9. To follow convention, add a *.git* extension to the **Directory** name (e.g. change OPBM to OPBM.git).

10. In the **Load Putty Key**, browse to the .ssh directory and add your private Putty key.



11. Click the **OK** button.

The following alert should appear:



After clicking **OK** for the first time, you will see the Alert shown above. Verify that the server's RSA key fingerprint is:

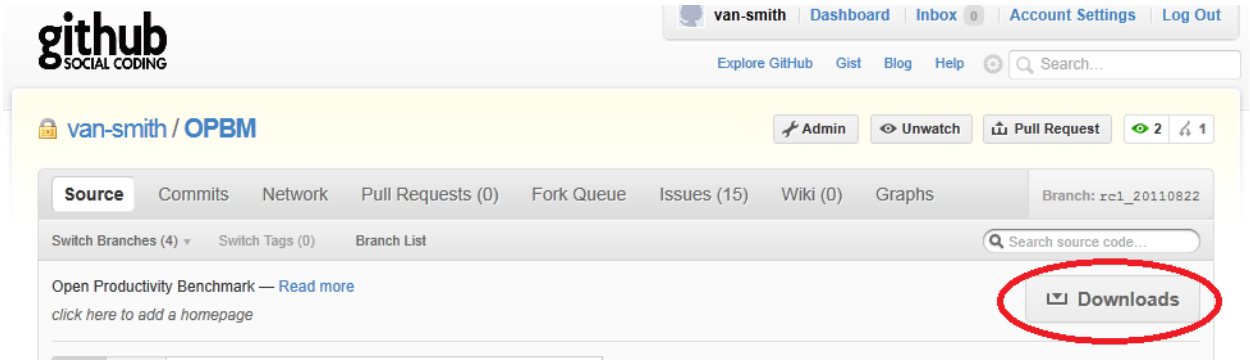
```
16:27:ac:a5:76:28:2d:36:63:1b:56:4d:eb:df:a6:48
```

Then click **Yes.** Follow any additional prompts and the repository default branch (dev) will begin to copy to your selected directory.

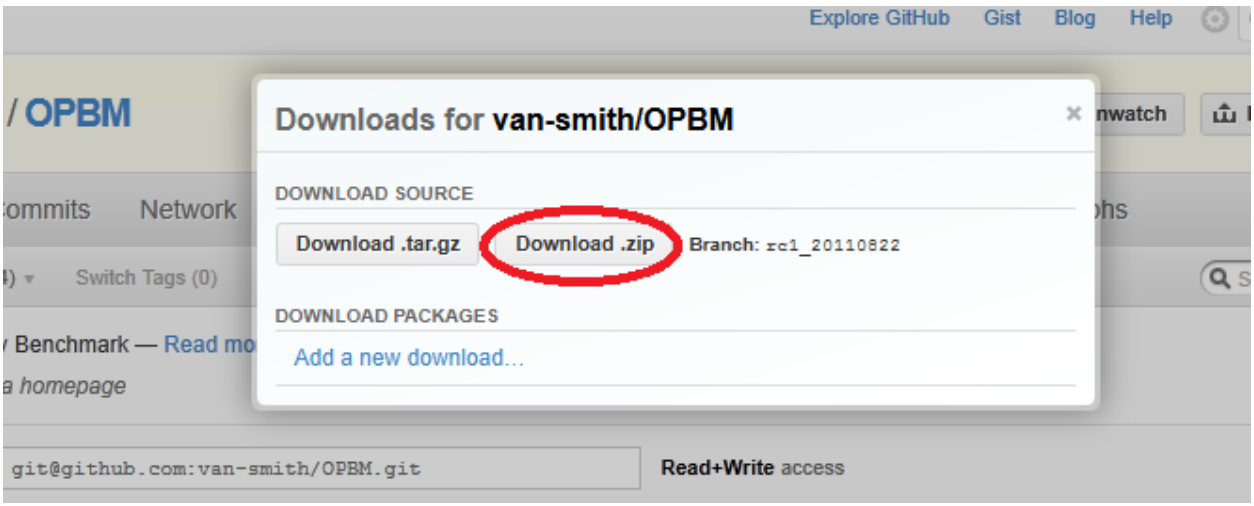
Once the repository exists locally, you can **Push** your changes up to GitHub, or **Pull** the latest changes down to your system. These commands are available from your TortoiseGit contextual menu within Windows File Explorer.

Downloading the latest version of the repository for testing

1. From within the **van-smith/OPBM** repository, click on **Switch Branches** and select **master** to access the most recent stable build.
2. Click on the **Downloads** button to download the latest build in a zip archive.



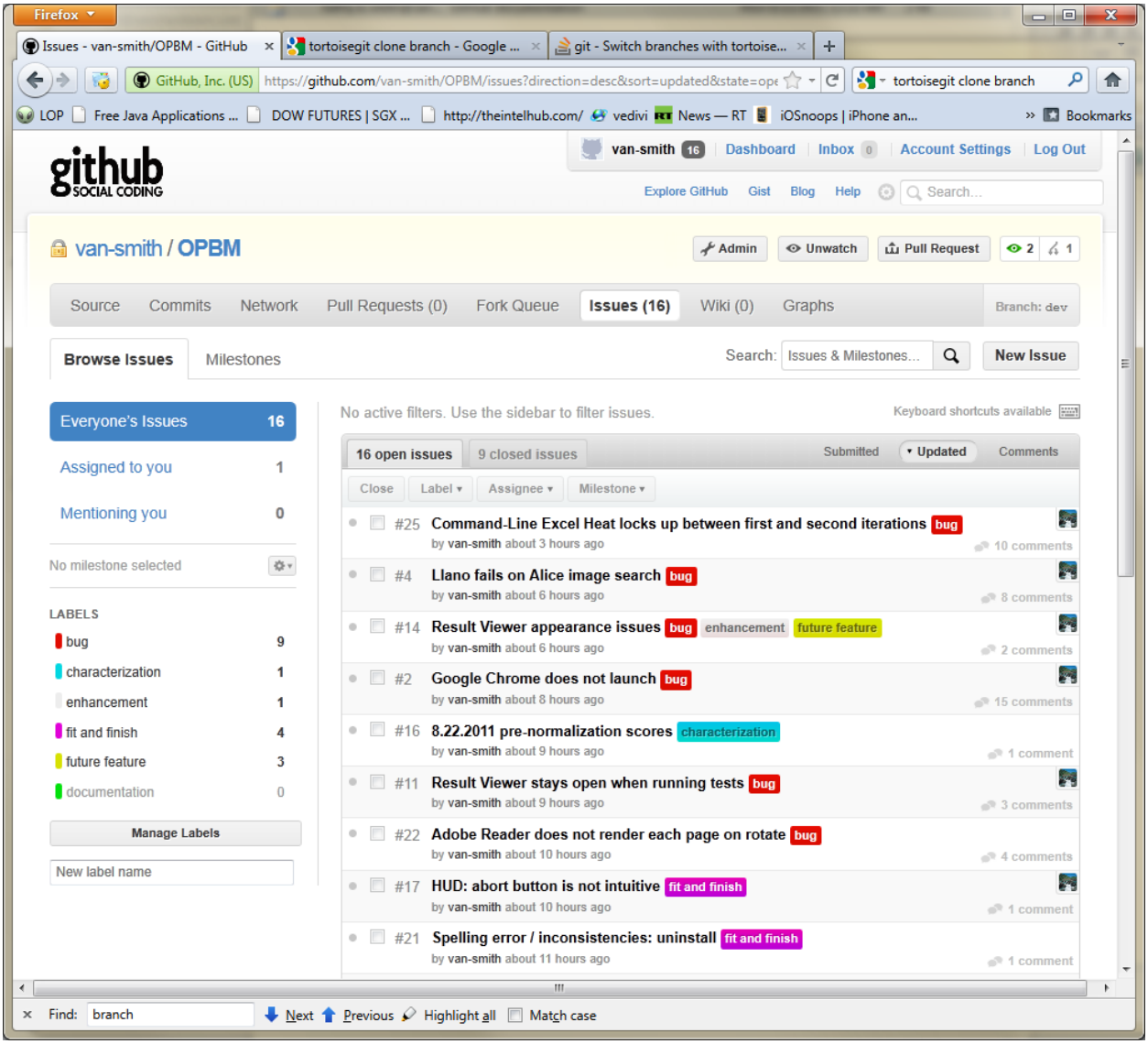
Since this is a Windows application, be sure to download the zip file rather than the .tar.gz file. Note again that you should be using the **Master** branch rather than the branch shown below.



Bug Tracking in GitHub

GitHub has an integrated issue tracker that we are using to track bugs and new features.

1. Click on Issues from the van-smith/OPBM repository. The webpage below appears.



2. To view an issue, click on it. You can sort by the last update or by number of comment. By default, issues are sorted by issue number. There are views for open or closed issues. Lastly, issues can be filtered by labels like **bug**, **documentation**, **enhancement**, etc. Notifications for new comments can also be enabled or disabled. Notifications are automatically created emails sent to you.
3. To add a New Issue, click on the **New Issue** button. Entering a new issue is straightforward as the screenshot below illustrates.