JetBRAINS - TeamCity 7

Requirements

- See for detailed instructions: http://confluence.jetbrains.net/display/TCD7/Supported+Platforms+and+Environments
- Any major operating system
- Tomcat 7 (included with Windows installer if not already installed)
- Java (JRE) 1.6+
- IDE Integration supported for Eclise, but not NetBeans
- JUnit testing is supported

*NOTE: I could not get the OS X install to work properly. It seems the web app hosts on port 9090 instead of 8111 as the instructions say, however the page at 9090 is not right.

Installation

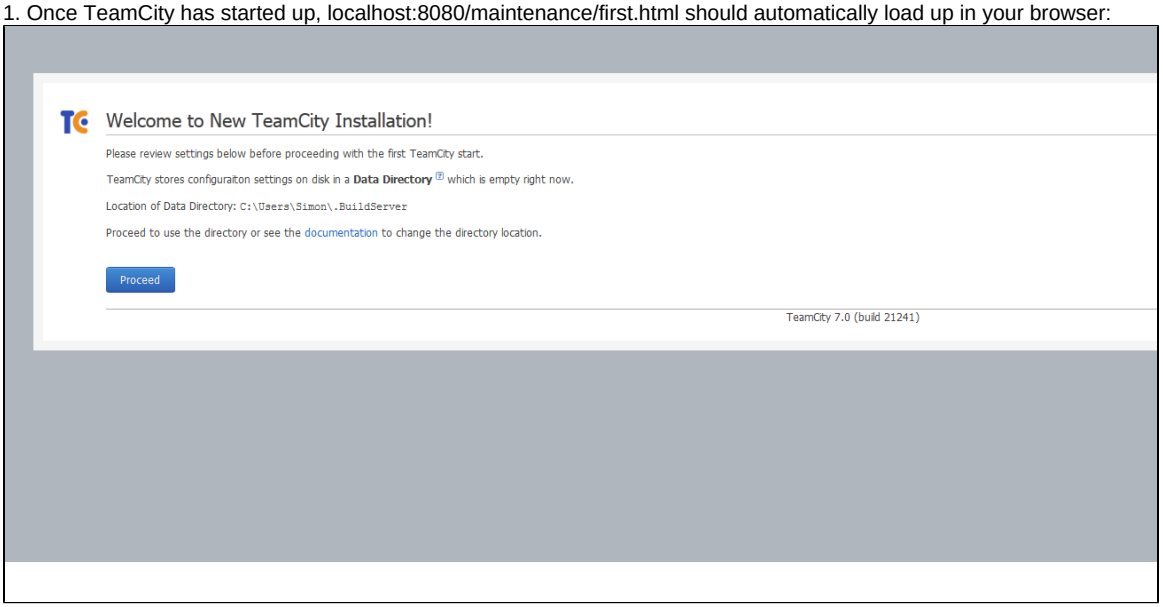
- Download from: http://www.jetbrains.com/teamcity/download/
    - Pick the appropriate version for your operating system
- See this page for detailed information for installing configuring TeamCity: http://confluence.jetbrains.net/display/TCD7/Installing+and+Configuring+the+TeamCity+Server
- OS X install is done at the command line by opening the tar ball. However I could not get the version to work properly, so I tried my Windows machine and had more success.

WINDOWS (7) INSTALLATION:

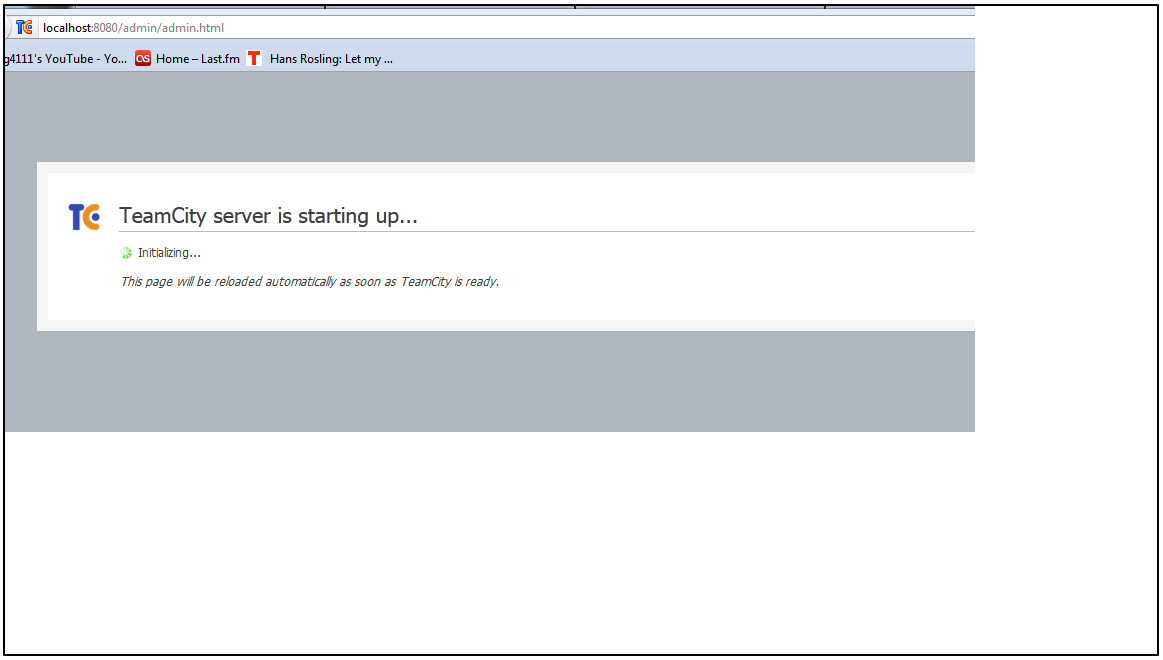
1. Download the windows .exe installer from the above link
2. Run the executable and follow the on screen instructions
3. Choose a port other than 80 as that is used by many other services (I suggest and had success with 8080)
4. Default settings for everything else should work
5. Make careful note of the required write permissions mentioned in the install guide page (should be ok by default if you are an administrator)
6. Check the box that asks to start up TeamCity after closing the installer

Configuration

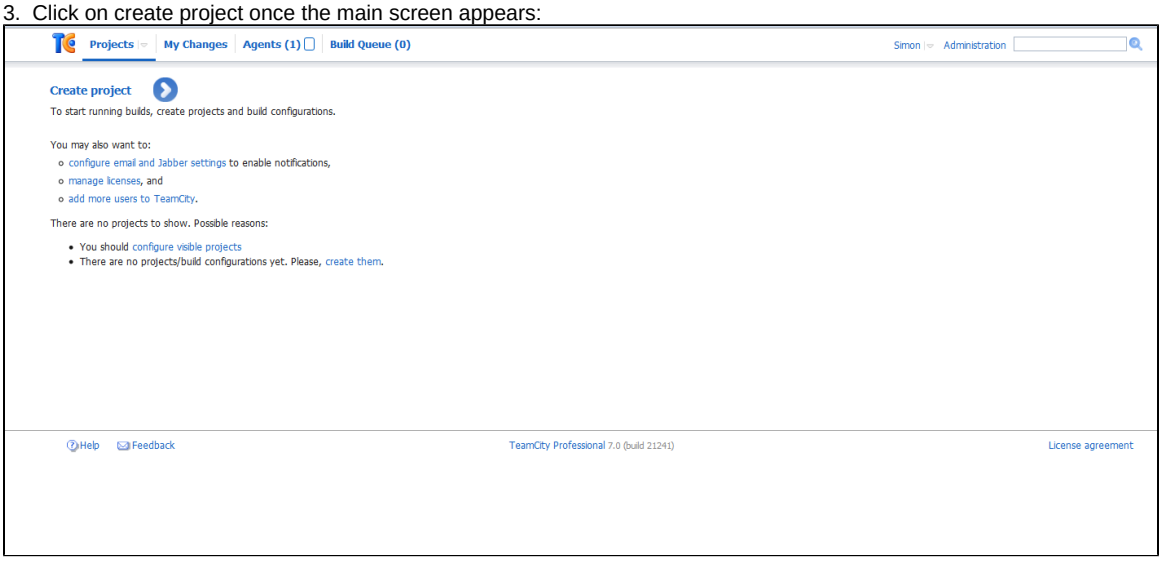
1. Once TeamCity has started up, localhost:8080/maintenance/first.html should automatically load up in your browser:



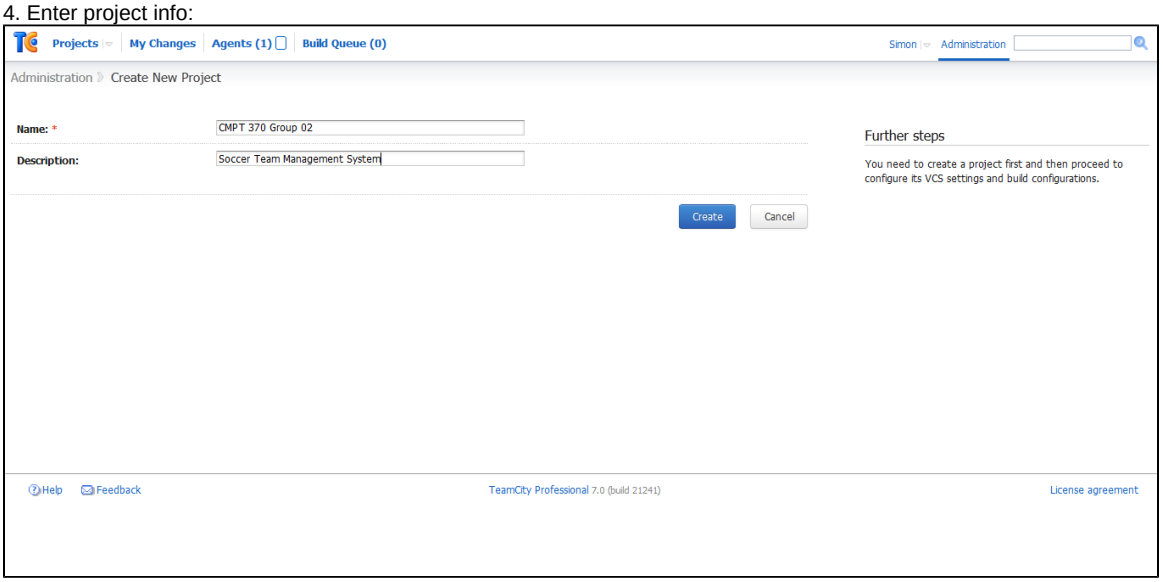
2. TeamCity server starting up:



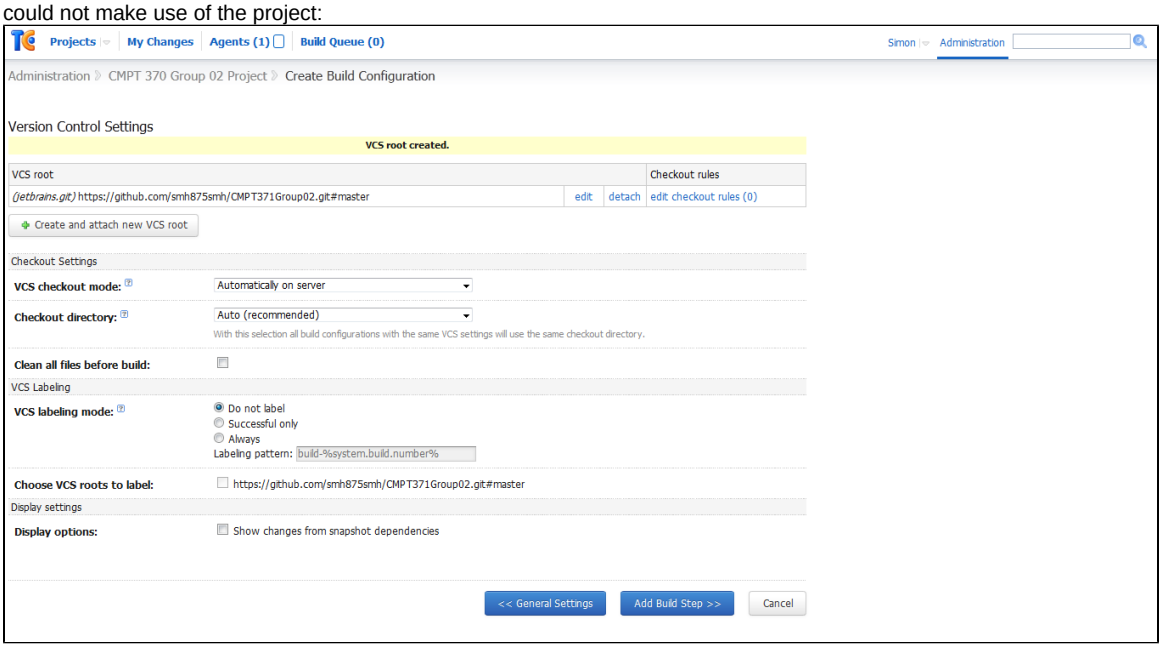
3. Click on create project once the main screen appears:



4. Enter project info:



5. Set up version control system (VCS). There are guides for this if you look around on their site. I used our GitHub for testing. Unfortunately I could not make use of the project:



It is at this point that I ran into a brick wall so to speak. You now have to configure build steps for TeamCity, the options of runners include things such as Maven, .NET process runner and NUnit. See below for a summary of the research on TeamCity 7 and a discussion of possible future work for later milestones.

Conclusion and Possible Future Work

TeamCity provides a platform for developers to achieve continuous integration and keep track of test success, build failures, or dependency issues. It allows for multiple integrations per day, and automatically performs each build and test to allow for quick detection of errors. This in turn means that developers are notified immediately when something has been broken, and the support is there to roll back to a previous version of the build. The overall user interface is fairly well polished, however I would like to see an even more "dumbed-down" version for those that are completely new to Continuous Integration software products. The support for a wide variety of operating systems and many languages and a plethora of features. From what I have read, if a build fails, TeamCity will provide you with a link to open the broken code segment directly in the chosen IDE. There is also detailed reports for each build, including the number of tests passed, failed, or ignored.

As for using our Soccer Management System project with TeamCity, the lack of a plugin for NetBeans greatly increases the difficulty of doing so. I'm assuming that it must be possible to implement it somehow using a variety of other software suites, however, the difficulty greatly increases when trying to create various build steps using runners that we are unfamiliar with. I am considering trying to implement a smaller scale sample project, possibly with the available Eclipse plug-in for a future milestone, or at least looking more into how TeamCity could be incorporated to our existing project. Also, discussion for integration with Maven for using Maven based build configurations is a possibility.

Overall, TeamCity seems like a very comprehensive and complex tool. Unfortunately, it seems understanding it thoroughly will be a task that is outside the available time and scope of this course.