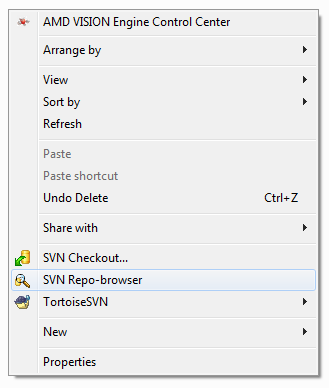
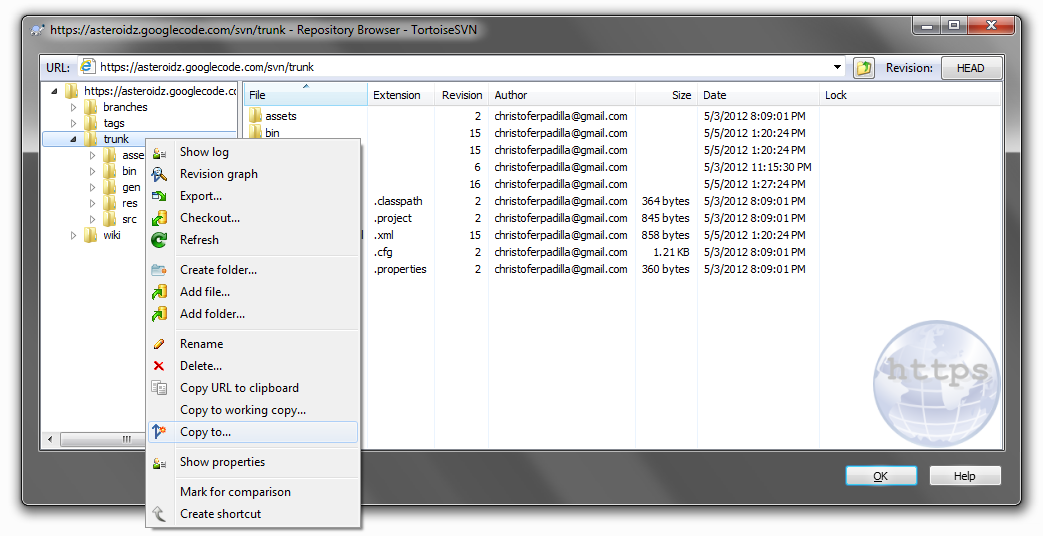
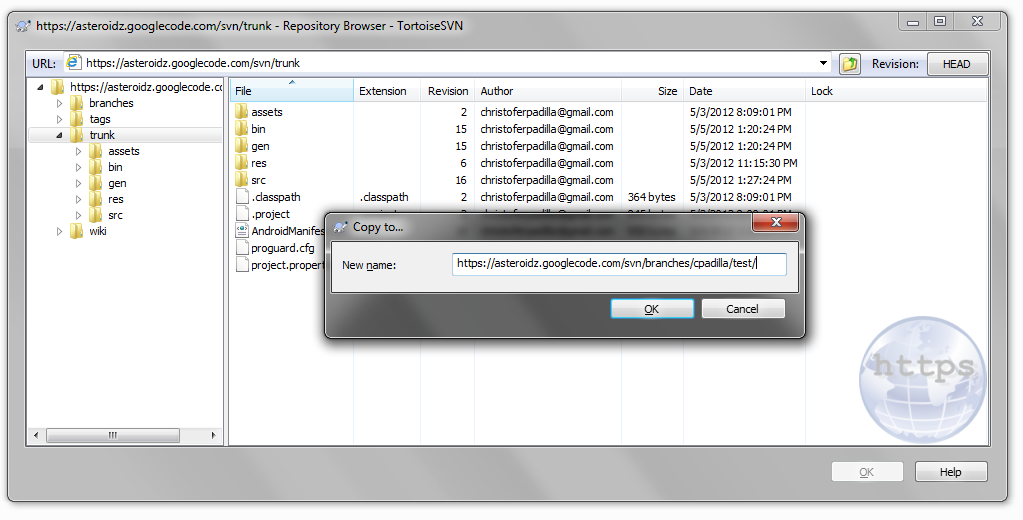
Tortoise SVN Tutorial

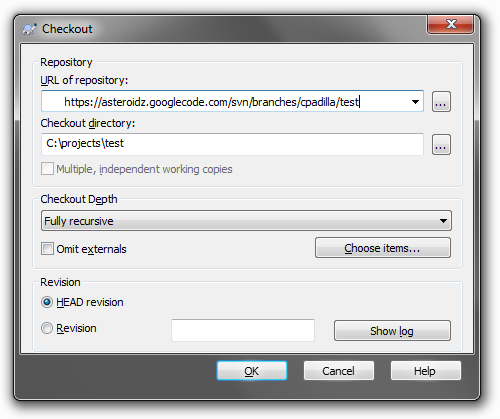
# Creating a branch

When you’re creating a branch, what you’re essentially doing is copying trunk (at whatever revision you want, but usually it will be from latest) into your own directory where you can work on it and commit you’re changes without affecting trunk.

1. Open up a windows explorer (Windows Key + E) and right click on any empty space.
2. You should see some options from Tortoise SVN. Click on “SVN Repo-browser”.
3. A little window pops up asking for the repository URL. The URL for our project is located here: <http://code.google.com/p/asteroidz/source/checkout>. It’ll give you some instructions there, but don’t worry about that. What’s important is that URL. So in the pop up, put in our project URL: **https**://asteroidz.googlecode.com/svn/trunk/
4. Now Tortoise SVN should ask for your username and password. Your username is the email you gave me when I added you as a committer. You should have an account on code.google.com with that email. Check your profile for your Google code password. It’s auto generated so just copy and paste it into the password field.
5. You should now see a little explorer window like the one shown above. This is basically all our files stored on the repository. The trunk folder has all of our main code that we’ll use to run our game. This is what you need if you want to work on any part of the game code. Right click on it and click “Copy to…”.
6. Now enter the URL of the folder you want to copy it to. You should have your own folder set up under /branches, in my case, mine is called cpadilla. I don’t have a folder called test in cpadilla, but if I type in the URL as shown above, it’ll create that folder for me. You can call your folder whatever you want. If you don’t have a folder under branches, you can easily make on by right clicking on the folder and selecting the “Add folder” option. When you make any changes, you’ll also be asked to put in a little log message. You should also put something in, even if it’s something short like, “Adding branch”.

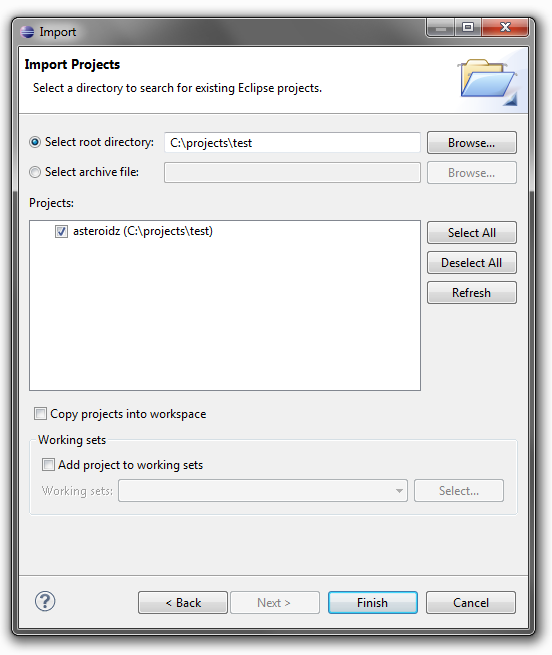
# Checking out a project

Now that you have a copy of the project in your branch, you’ll probably want to start working on it and junk.

1. Open up a windows explorer (Windows Key + E) and right click on any empty space.
2. Click on “SVN Checkout…”
3. A new window should pop up. By default, the URL of the place you want to check out a project form should be trunk. Change the directory to the location of your branch you want to checkout your project from. We’ll use the one we created in the last example. Underneath that is the Checkout directory. This is where the actually project will be saved on your computer. I created a folder called projects under C:\ where I checkout all my subversion projects to. I’d recommend you do this as well, but you can save it wherever you want on your computer.

# Importing a new project into Eclipse

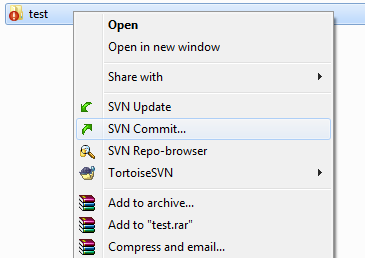
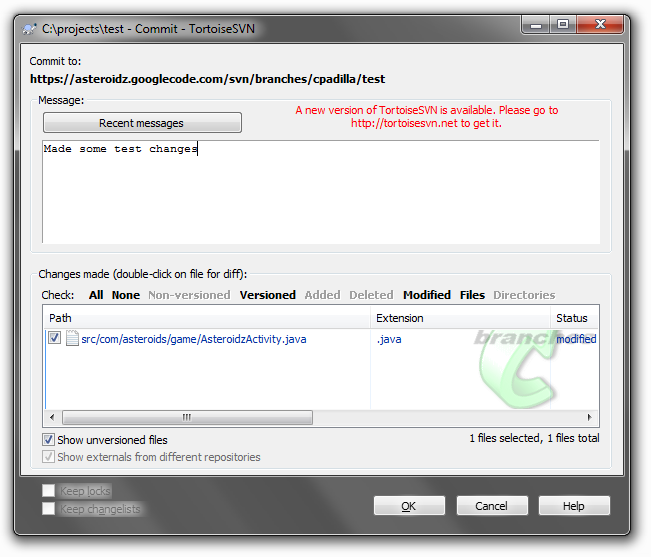
Although your project is stored on your computer, you probably won’t see it right away on eclipse.

1. Open up Eclipse
2. Choose a workspace (wherever. if you have it set manually, that’s great; if you just leave it at it’s default place, that’s great too).
3. Right click on any empty space in the Package Explorer and choose “Import…” or in the menu bar go to File -> Import…
4. Under “Select root directory:” enter in the local address or browse for where your folder for your project is. In the case of the last example, it was saved under C:\projects\test. Click Finish and you should have it in your Package Explorer ready to run!

NOTE: You may occasionally see a little warning sign at the top saying: “Some projects cannot be imported because they already exist in the workspace” That’s probably because you already have the asteroidz project in your Package Explorer from a previous revision. It’s okay to just delete the asteroidz folder in your Package Explorer and then import the new project.

# Committing Changes

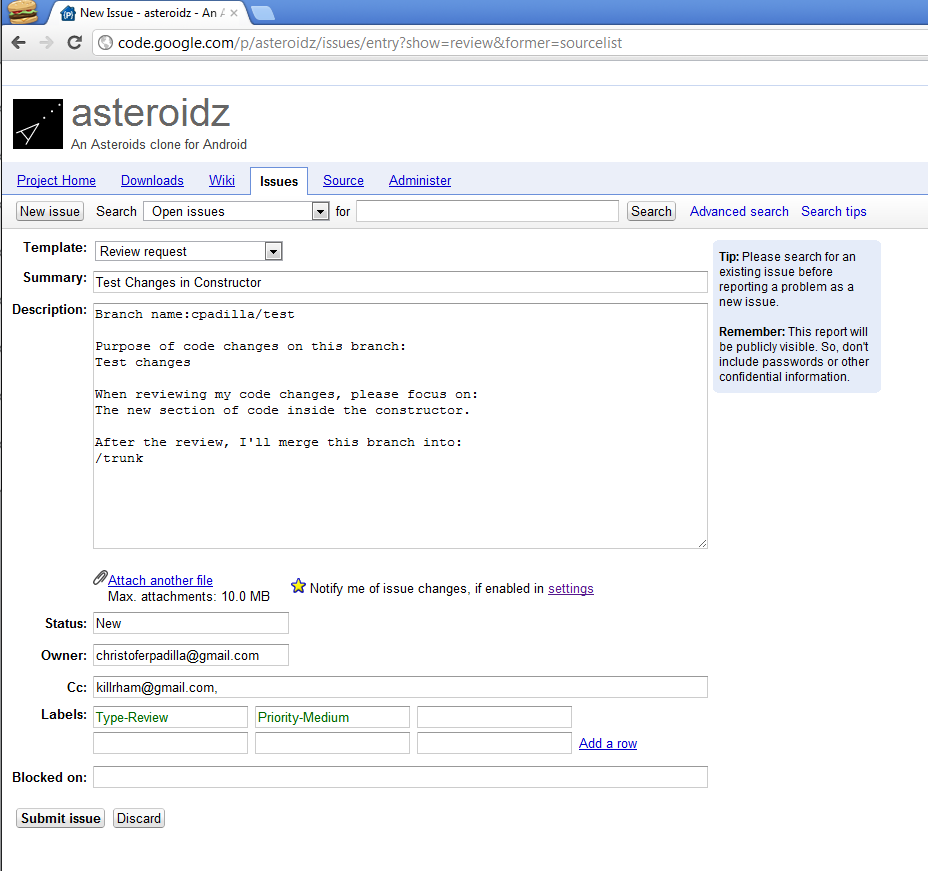
So now that you’ve been working on your project you want to save your work in your branch. You’ll see that after you’ve made some modifications to the code, your folder will have a red exclamation mark, signifying a difference in the project code and the code still saved on the repository.

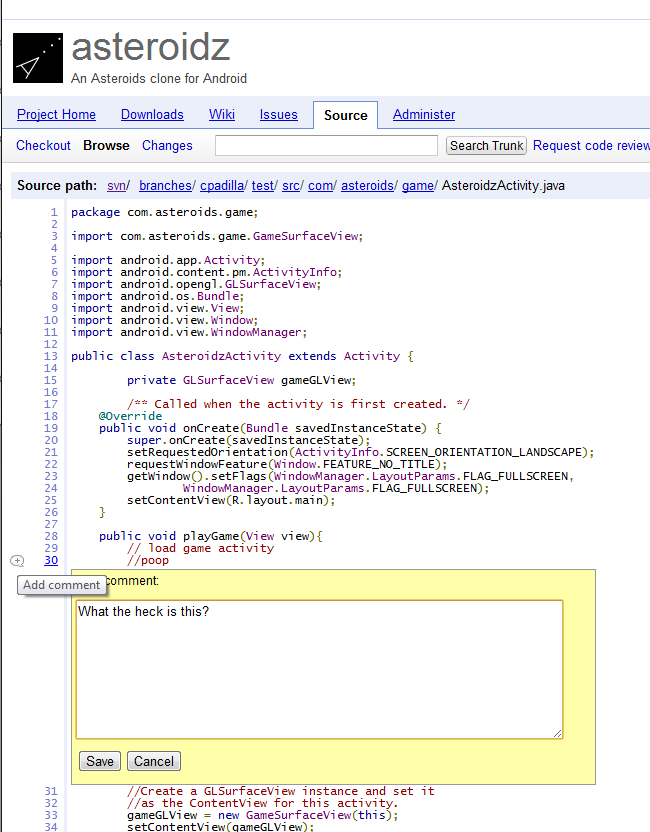
1. Select the folder.
2. Right click on the folder. You should now see some new options. Click on “SVN Commit…”
3. A new window should appear that looks like the one above. Here is where you can see all of the files that you changed as well as a little text field for you to write what you changed and how you changed it (modified, added, deleted). You should always leave a message, even if it’s short. Once you’ve done that, go ahead and click OK.

NOTE: Sometimes you may see things included in the change section that you didn’t mess with. Sometimes it’s stuff that gets changed automatically to reflect your changes, and some stuff is auto generated. I’m not an expert on it all either, so just make a good judgment call on whether to include it or not. Like I think an .apk file is generated if you run it on your phone so that it can be downloaded on it, but you don’t need to commit that.

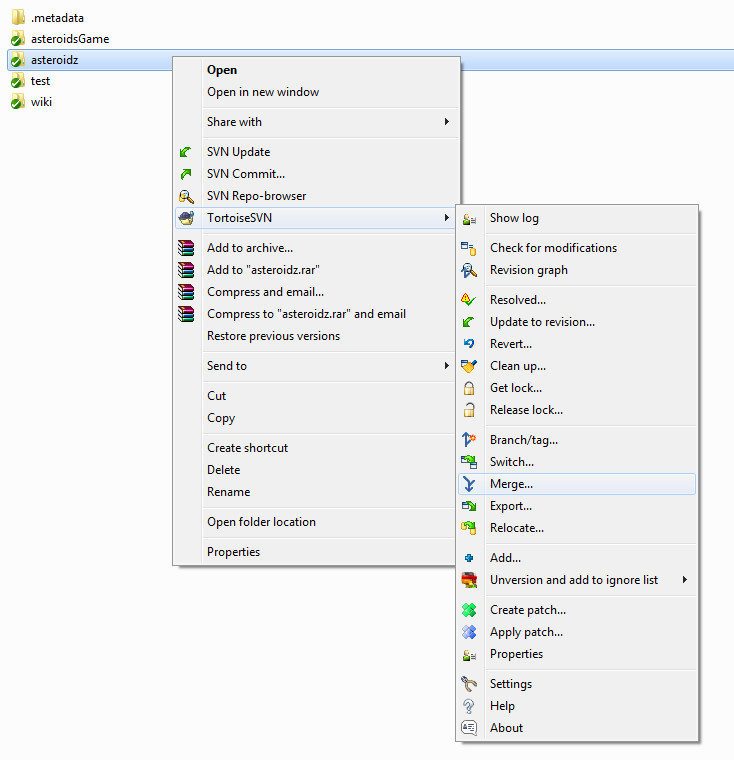
# Submitting changes for code review

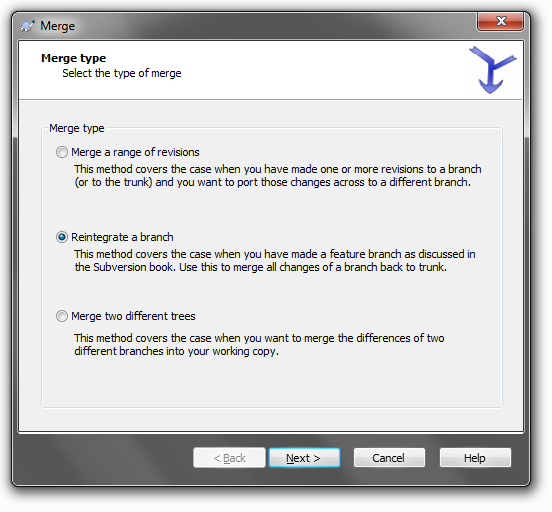
Sometimes you may want to show someone your changes before you commit it to trunk. Here’s how you should go about submitting an issue.

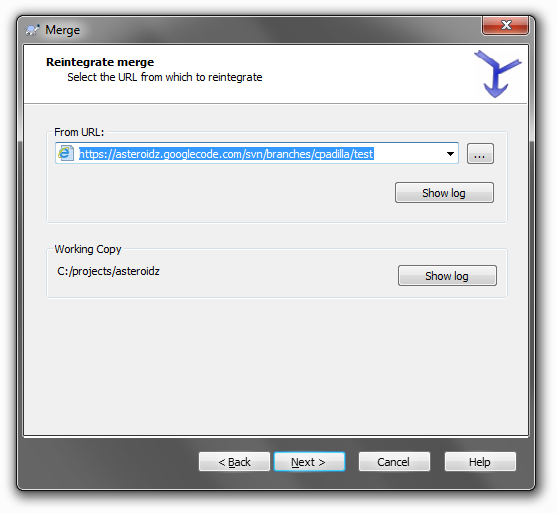


Basically just go to the website and click on the “Issues” tab. I’ve detailed an example above on what a basic code review might look like. You can also expect feedback from others, and you yourself can post comments on changes to code, yours or not. You can do so by going to the “Source” tab and clicking wherever you want to make a comment.

# Committing your changes to trunk

Now that you’re done working with your branch and everything’s good to go, you should commit your changes to trunk. This is how you reintegrate your branch into trunk. Essentially you are pulling a copy of trunk onto your computer, merging your branch with that trunk, then commit that trunk.

1. Check out a new copy of trunk, as explained in “Checking out a project” to your computer.
2. Navigate to wherever you just checked out trunk. (When I checked it out I called it “asteroids”) We’ll now merge your changes from your branch back into trunk.
3. Right click on it and go to TortoiseSVN -> Merge…
4. A new window should appear. Select the second option, “Reintegrate a branch”
5. Click Next.



1. In the “From URL:” field, you should put in the URL of your branch, since that’s the branch we are reintegrating. The Working Copy field should show the directory of your trunk, which is what we’re merging into.
2. Click Next. Click Merge. It should now pop up with a new little window showing all the changes it merges. Let it do its thing. Congratulations! You’ve now merged your branch back into trunk!
3. Your copy of trunk should have a little red exclamation mark on it, signifying a change from what is in the repository. This is good because it means our changes were merged successfully. Now just commit it like normal, and your changes should be in trunk!  
   NOTE: Sometimes you may have some conflicting changes after you’re merge. You’ll know this because it tells you this along with which files are conflicted. If this is the case, your folder should have a little warning sign on it.

You’ll have to manually merge your changes for those files. Once you’ve done that, just right click on it, click TortoiseSVN -> Resolved… and it’ll assume you manually fixed it. Then commit like normal. If this happens though, you should probably let someone know, in case you might have accidentally made some kind of mistake merging.