Tutorial 0: SVN

ECSE 321 – Intro to Software Engineering
Electrical and Computer Engineering
McGill University

Winter 2009

Introduction

In the field of software development, the Concurrent Versions System (CVS), also known as the Concurrent Versioning System, is a free software revision control system. Version control system software keeps track of all work and all changes in a set of files, and allows several developers (potentially widely separated in space and/or time) to collaborate [1].

Subversion (SVN) is a version control system initiated in 2000 by CollabNet Inc. It is used to maintain current and historical versions of files such as source code, web pages, and documentation. Its goal is to be a mostly-compatible successor to the widely used Concurrent Versions System (CVS) [2].

This little tutorial will help your team to set up a code repository for your term project and teach you the basics of SVN logic.

What SVN allows, is to hold a central database of code and other files, so that they can be easily shared amongst members of the group. SVN logic functions as a tree, where the main folder is called the trunk and all subsequent subfolders are referred to as branches. Here is a step by step rundown of what to do to set up your repository. The logic behind it will be explained as we go.

Step 1: Desktop Setup

Create a new folder, wherever you want on your computer. This will be the folder housing all your uploaded files. Give it a meaningful name, as you'll be stuck with it once you've uploaded it to the servers.

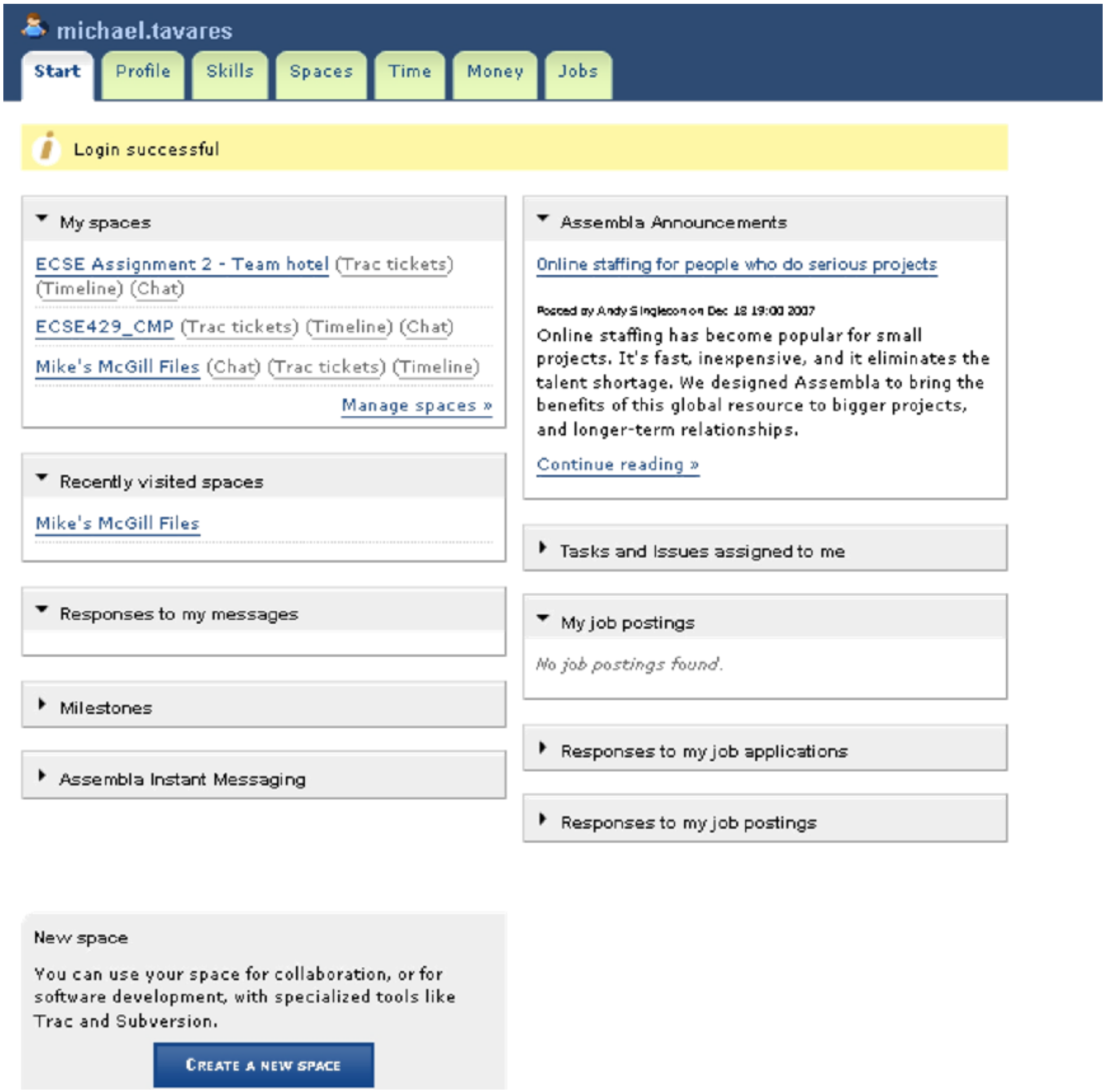
For this course you are required to use Assembla for hosting your code. In the past the school would provide servers, but that has come and gone, and a reliable solution has found for you. Using this solution, you will include whichever TA is in charge of your team to be a part of your Assembla project, this way we can monitor your progress and get a hold of your files without you having to worry about submissions.

Step 2: Registering

Go to http://www.assembla.com and register an account by clicking the *Register* link at the top of the page. Fill in the username field like so: firstname.lastname (this way we know who did what). Feel free to include personal information or not.

Step 3: Creating a Project Space (only 1 team member does this)

When you are done registering, and are logged into the site, you will be brought to the start page.



Login successful

My spaces

ECSE Assignment 2 - Team hotel (Trac tickets) (Timeline) (Chat)

ECSE429_CMP (Trac tickets) (Timeline) (Chat)

Mike's McGill Files (Chat) (Trac tickets) (Timeline)

Manage spaces »

Recently visited spaces

Mike's McGill Files

Responses to my messages

Milestones

Assembla Instant Messaging

Assembla Announcements

Online staffing for people who do serious projects

Posted by Andy Singleton on Dec 18 19:00 2007

Online staffing has become popular for small projects. It's fast, inexpensive, and it eliminates the talent shortage. We designed Assembla to bring the benefits of this global resource to bigger projects, and longer-term relationships.

Continue reading »

Tasks and Issues assigned to me

My job postings

No job postings found.

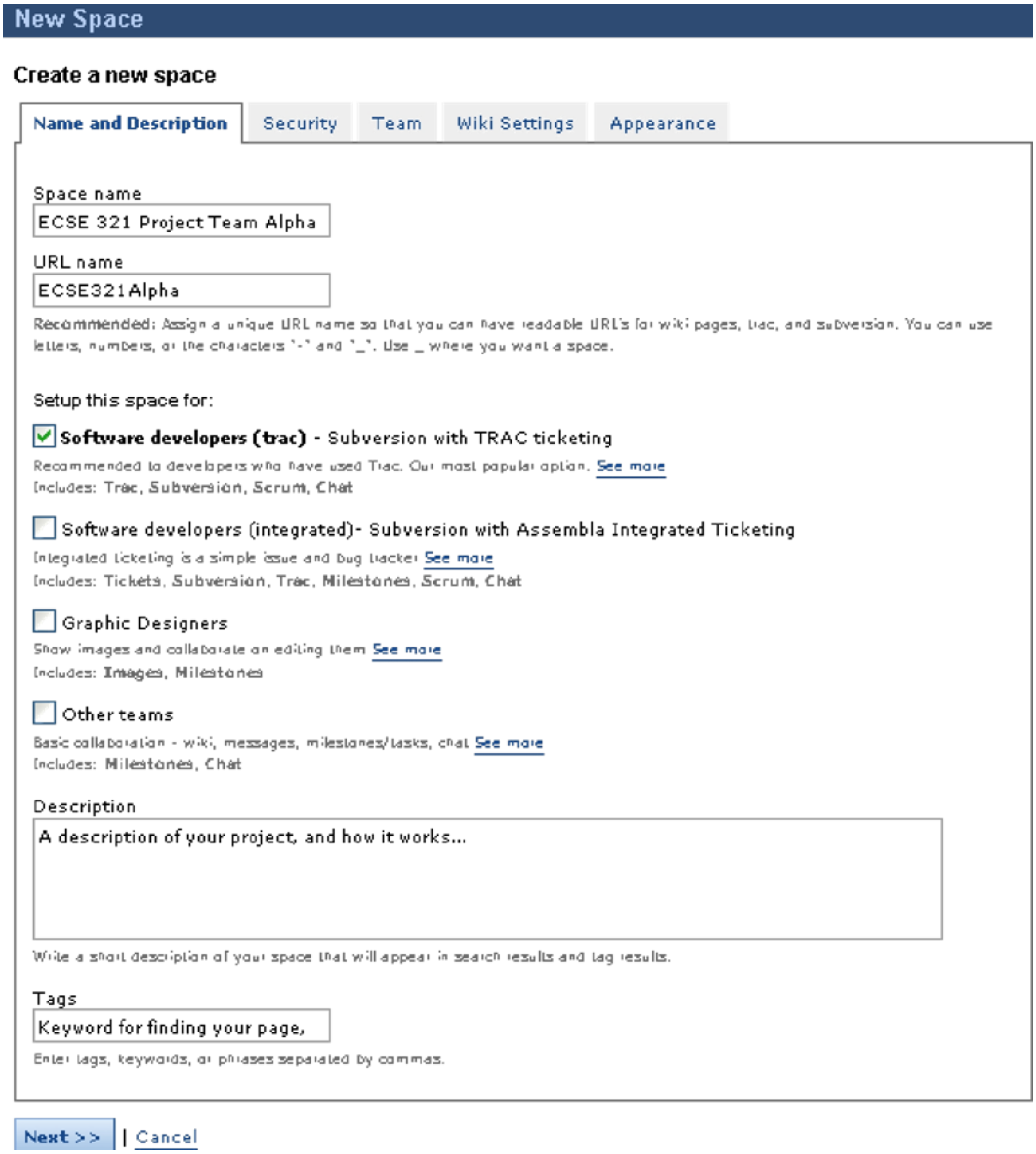
Responses to my job applications

Responses to my job postings

New space

You can use your space for collaboration, or for software development, with specialized tools like Trac and Subversion.

Click the *Create a New Space* button at the bottom of the screen. You will be taken to this page:



New Space

Create a new space

Space name

ECSE 321 Project Team Alpha

URL name

ECSE321Alpha

Recommended: Assign a unique URL name so that you can have readable URL's for wiki pages, trac, and subversion. You can use letters, numbers, or the characters '-' and '_'. Use _ where you want a space.

Setup this space for:

☑ **Software developers (trac)** - Subversion with TRAC ticketing

Recommended to developers who have used Trac. Our most popular option. See more

Includes: Trac, Subversion, Scrum, Chat

□ Software developers (integrated)- Subversion with Assembla Integrated Ticketing

Integrated ticketing is a simple issue and bug tracker See more

Includes: Tickets, Subversion, Trac, Milestones, Scrum, Chat

□ Graphic Designers

Show images and collaborate on editing them See more

Includes: Images, Milestones

□ Other teams

Basic collaboration - wiki, messages, milestones/tasks, chat See more

Includes: Milestones, Chat

Description

A description of your project, and how it works...

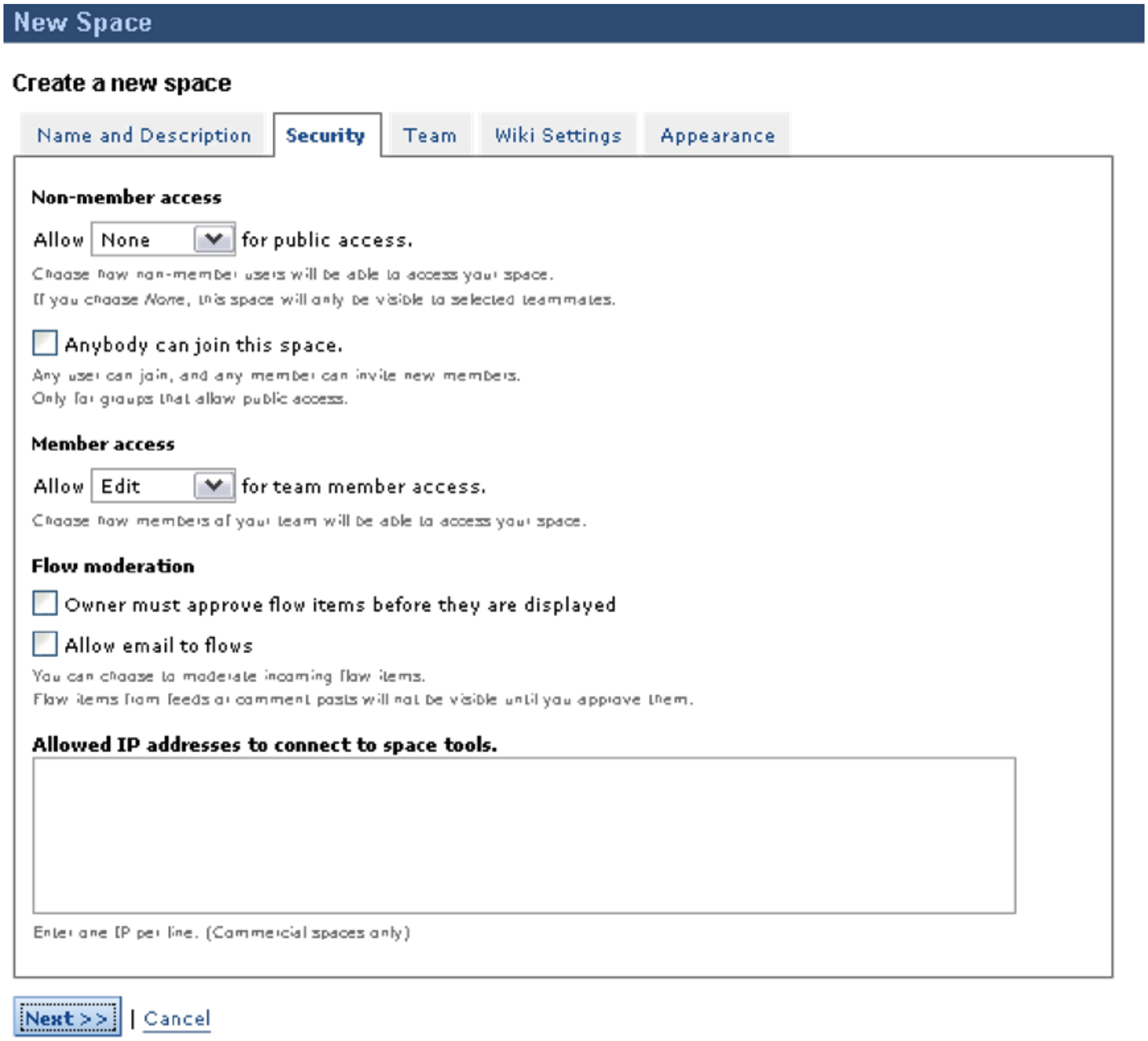
Write a short description of your space that will appear in search results and tag results.

Tags

Keyword for finding your page,

Enter tags, keywords, or phrases separated by commas.

Fill in the fields as you wish, for description and keywords, but make sure you check the Trac box. The next step pertains to security.



Create a new space

Non-member access

Allow None for public access.

Choose how non-member users will be able to access your space.
If you choose *None*, this space will only be visible to selected teammates.

□ Anybody can join this space.

Any user can join, and any member can invite new members.
Only for groups that allow public access.

Member access

Allow Edit for team member access.

Choose how members of your team will be able to access your space.

Flow moderation

□ Owner must approve flow items before they are displayed

□ Allow email to flows

You can choose to moderate incoming flow items.
Flow items from feeds or comment posts will not be visible until you approve them.

Allowed IP addresses to connect to space tools.

Enter one IP per line. (Commercial spaces only)

Leave the default settings which will only allow for team members access the files and only the person who created the project can invite more team members. The next step is for inviting team members, simply enter the email your team registered to the site, you can add more in later easily. The Wiki step, keep the WYSIWYG setting (default), and finally the Appearance step, be creative. Your project space should now be created

Step 4: Accessing your Project Space

When you go back to your start page, the space you just created (or were invited to, and accepted) should appear in Space section. Select it and jot down the Subversion URL for this space, you will need it soon.

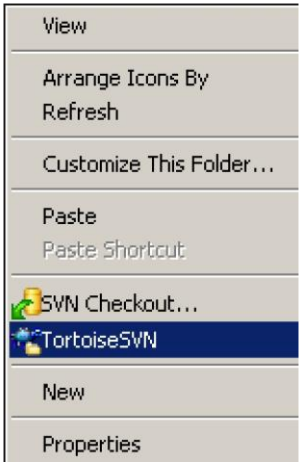
Step 5: Installing Subversion (SVN)

Download and install Subversion
http://subversion.tigris.org/servlets/NewsItemView?newsItemID=1941

Step 6: Installing TortoiseSVN (Windows)

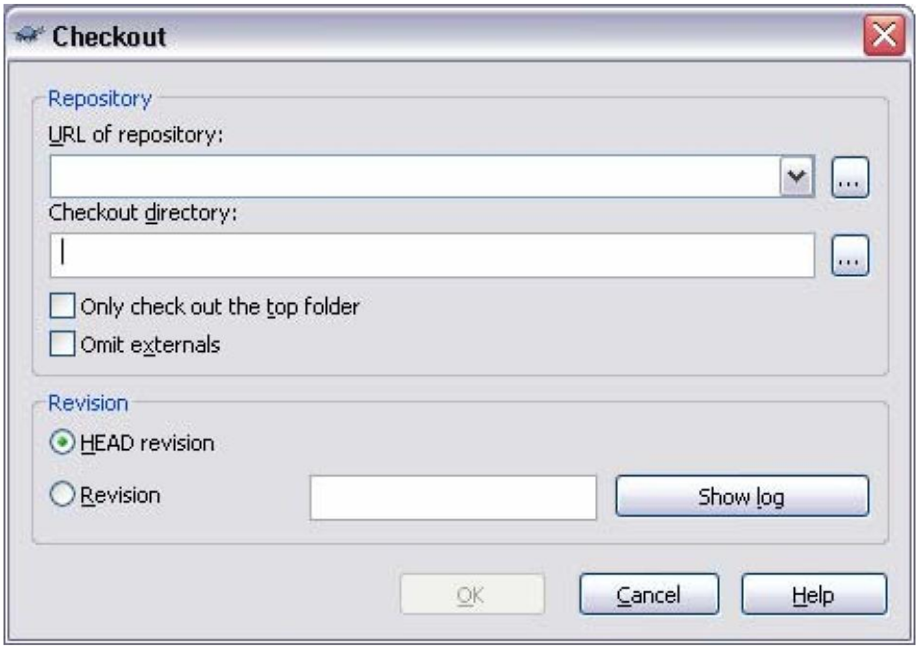
Download and install TortoiseSVN (already installed on all lab computers) http://tortoisesvn.net/downloads

From now on when you right-click on a folder you should see these options:



Step 7: Checking out your Folder

Go to the folder created in Step 1, and right-click on it. Select SVN Checkout. You will get this window:



□ Only check out the top folder

□ Omit externals

● HEAD revision

○ Revision

Step 8: Completing checkout

In “URL of Repository” enter the URL found in step 4, and in checkout directory, there should already be the path of the folder you clicked on in step 7. Click ok. The folder will be synced with the repository and all the necessary files for uploading will be added to the folder. After pressing OK, you will be prompted for a username/password, use your Assembla information. I suggest check the box that will remember this information to save time.

SVN Commands

Here are a list commands that can be used in SVN.

Commit Right-click folder → SVN Commit

This command recursively sends your changes to the SVN server. It will commit changed files, added files, and deleted files. Note that you can commit a change to an individual file or changes to files in a specific directory path by adding the name of the file/directory to the end of the command. Make sure to put some text in the log window, so that everyone knows what has been changed.

Update Right-click folder → SVN Udate

This command syncs your local folder with the server. If you have made local changes, it will try and merge any changes on the server with your changes *on your machine*.

Resolve Conflicts Right-click file → TortoiseSVN → Resolve Conflicts

Should any conflicts arise during updating, you must resolve them. From there, you can see a side-by-side view of both file versions, and choose to either use one or the other for every error. Once all errors are attended to, the file is resolved.

Revert Right-click file/folder → TortoiseSVN → Revert

Should a commit be done and the change is no longer valid or accepted, it is possible to revert to an older version of a file. The repository keeps track of all changes to files and keeps a copy of all versions.

There are numerous other options available, though most of them do not need to be used in the context of this course.

Reference

[1] http://en.wikipedia.org/wiki/Concurrent_Versions_System.
[2] http://en.wikipedia.org/wiki/Subversion_(software)