Version Control for IBM i

BECAUSE THIS TUTORIAL IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE TUTORIAL, TO THE EXTENT PERMITTED BY APPLICABLE LAW. THE TUTORIAL IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY OF THE TUTORIAL IS WITH YOU. SHOULD THE TUTORIAL PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

Outside the IBM i world it is best practice to use version control systems to handle the integrity of the sources of a software project. There are open source VCS like Subversion, Git and Mercurial as well as commercial ones like ClearCase. All those VCS can be used for Java, PHP, .NET and other projects because they provide connectors for the IDE`s used for such projects.

But what about Version Control for my RPG and COBOL projects on IBM i? My strongest suggestion is to use a change management system like CMOne of Task Force IT-Consulting GmbH. All your needs will be satisfied with such a change management system. But those change management systems are not for free. You have to pay a fee and depending on the software vendor this can be a lot of money.

For those ones of you who are not able to spend money for a change management system we have written this step by step tutorial to show you a way to use Version Control for RPG and COBOL projects on IBM i. All you need are the following components.

- **WDSCi 7.0** or **RDP 8.0** or **RDP 8.5** or **RDi 9.0**. This is our IDE we are using for our work on RPG and COBOL projects on IBM i. We will use the “i Projects” perspective of those IDE`s.
- **Subversion server**. This is the VCS which holds the repositories with our sources.
- **Subclipse** as **Subversion client**. This is the connector for our IDE to connect to the Subversion server.
- The open source plugin **iSVN** of Task Force IT-Consulting GmbH. This plugin provides additional functions for the “i Projects” perspective you will find very useful.

So let`s start with the subversion server. In general you can use any Subversion server implementation on any platform. If you want to use a Subversion server on a Windows platform we recommend uberSVN of WANdisco. The base product is for free. It`s extensible via plugin`s but this plugin`s are with costs. But the features of the base product are enough for most of the companies.

If your shop has no Windows or Linux server to host the subversion repositories you can also use your IBM i server. Yes you have heard right. There is one open source Subversion server available written in Java. It`s SCM-Manager of TRIOLOGY GmbH. They are also from Germany like us. SCM-Manager comes as WAR file which means that you can run SCM-Manager in an application server like Websphere or Tomcat. Since Tomcat is also a java application Tomcat runs also on IBM i. If you decide to use SCM-Manager on IBM i as Subversion server there`s another tutorial on our homepage which will guide you through the steps of installing Tomcat and SCM-Manager on IBM i.

If you use uberSVN or SCM-Manager or another Subversion server you should now create a repository for your RPG or COBOL project. How you have to do this depends on your Subversion server.

If not happened it`s now time to install WDSCi 7.0 or RDP 8.0 or RDP 8.5 or RDi 9.0. These are the IDE`s which will be supported by our free iSVN plugin. I think there`s no need to give any additional instructions on how to install those IDE`s.

Next you have to install the necessary plugin`s in your IDE. First if you run Windows Vista, Windows 7 or Windows 8 it`s important to start your IDE with "Run as administrator".

To download and install Subclipse follow the instructions below.

- Go to http://subclipse.tigris.org/.
  - Click "Download and Install".
  - Copy the Eclipse update site URL of the latest version to your clipboard (STRG-C).

The instructions below are for RDP 8.0, RDP 8.5 and RDi 9.0. We will not provide additional information on how to install the plugin in WDSCi 7.0.

- Go to your IDE.
  - Select menu item „Help/Install New Software". The window "Available Software" appears.
  - Click „Add ". The window "Add Repository" appears.
  - In field "Name" enter "Subclipse" and in field "Location" past the copied URL (STRG-V).
  - Click „OK". The Subclipse update site appears in window "Available Software".
  - Expand all items in window "Available Software".
  - Click "Select All".
  - Deselect "Subclipse Integration for Mylyn".

  - Click "Next". The window "Install Details" appears.
  - Click "Next". The windows "Review Licenses" appears.
  - Click "I accept the terms of license agreements".
  - Click „Finish". The window "Installing Software" appears. Don`t cancel the process.
  - The window "Security Warning" appears. Click "OK".
  - The window „Software Updates" with the message „You will need to restart …" appears. Click „Restart Now".

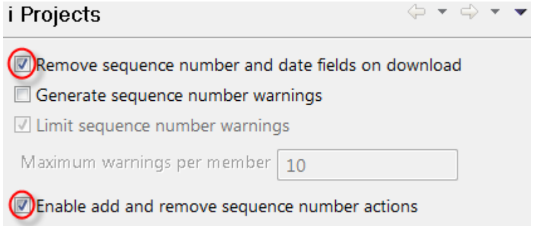
To download and install iSVN follow the instructions below.

- Go to http://www.taskforce-it.de/.
  - Click "Downloads".
  - Download the Zip file "iSVN (Update Site)"
  - Unzip the file.

The instructions below are for RDP 8.0, RDP 8.5 and RDi 9.0. We will not provide additional information on how to install the plugin in WDSCi 7.0.

- Go to your IDE.
  - Select menu item „Help/Install New Software". The window "Available Software" appears.
  - Click „Add ". The window "Add Repository" appears.
  - Click "Local". The window "File Chooser" appears.
  - Select the downloaded folder „iSVN (Update Site)" in the target directory.
  - Click „OK". The iSVN update site appears in window "Available Software".
  - Click "Select All".
  - Click "Next". The window "Install Details" appears.
  - Click "Next". The windows "Review Licenses" appears.
  - Click "I accept the terms of license agreements".
  - Click „Finish". The window "Installing Software" appears. Don`t cancel the process.
  - The window "Security Warning" appears. Click "OK".
  - The window „Software Updates" with the message „You will need to restart …" appears. Click „Restart Now".

After everything has installed we have to do some configurations. Click "Window/Preferences/i Projects". Tick "Remove sequence number and date fields on download". Also tick "Enable add and remove sequence number actions". Click "Apply". Click "OK".



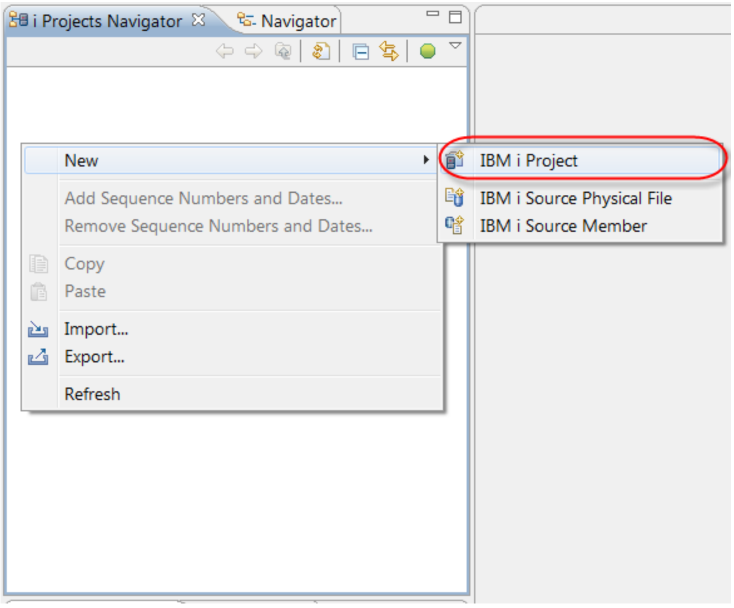
i Projects

☑ Remove sequence number and date fields on download
☐ Generate sequence number warnings
☑ Limit sequence number warnings
Maximum warnings per member 10
☑ Enable add and remove sequence number actions

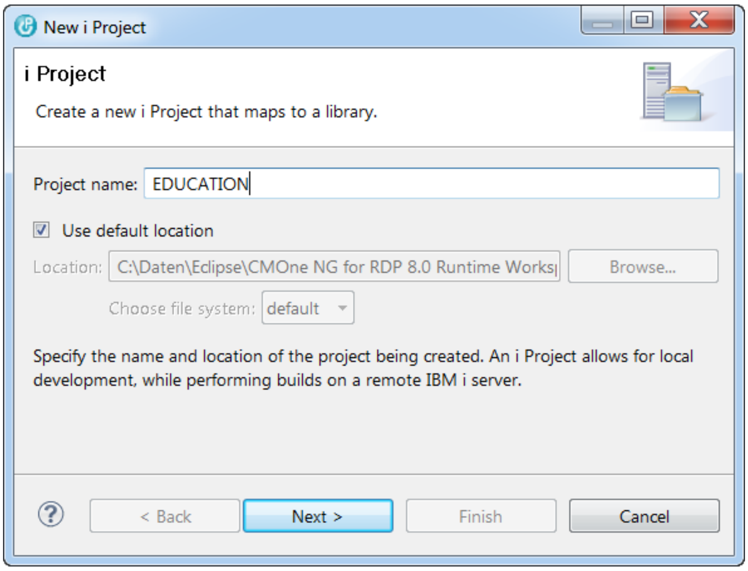
On positions 1 to 12 of IBM i source files the sequence number and the date of the last change will be stored. This is IBM i specific. Source files on other platforms don`t share this. Subversion has problems to handle this. By enabling the options above the source files will be stored in Subversion without sequence number and without date. **This means for you, that you cannot longer search the source files for a change date.** In my opinion this is no problem because Subversion provides me all possibilities to investigate changed statements. But everyone has to decide on his own if this is a problem or not.

Open the perspective "i Projects" by selecting "Window/Open Perspective/Other/i Projects".

Create a new project in "i Projects Navigator" by opening the context menu and selecting "New/IBM i Project".



Enter a project name and click "Next".



Project name: EDUCATION

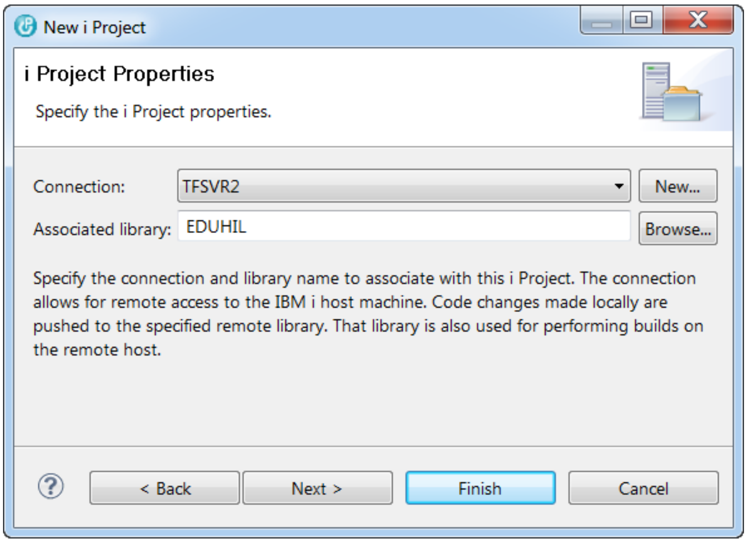
☑ Use default location

Location: C:\Daten\Eclipse\CMOne NG for RDP 8.0 Runtime Works|

Choose file system: default

Specify the name and location of the project being created. An i Project allows for local development, while performing builds on a remote IBM i server.

Select a connection. Specify a working library. This should be a new library and only you should work with this library. This library will contain all the changes you will make. Click “Finish”.

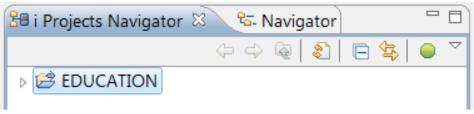


Connection: TFSVR2

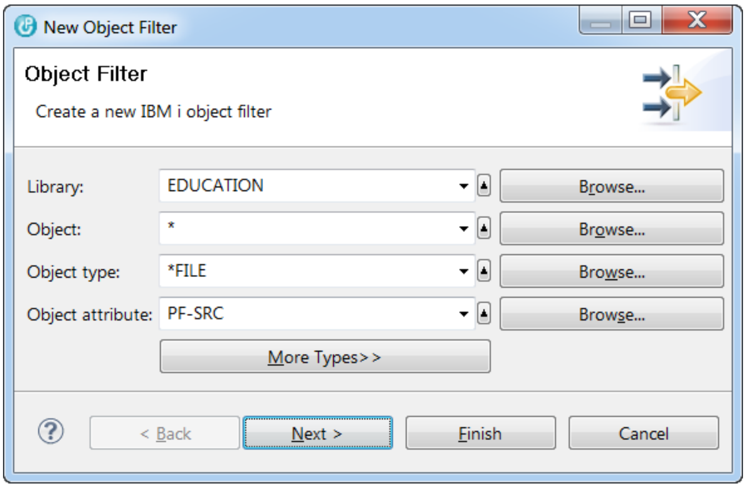
Associated library: EDUHIL

Specify the connection and library name to associate with this i Project. The connection allows for remote access to the IBM i host machine. Code changes made locally are pushed to the specified remote library. That library is also used for performing builds on the remote host.

The project has been created.



Now we have to import the source files of our application. Go to the Remote Systems Explorer and create an object filter for the library which contains your source files.



New Object Filter

Object Filter

Create a new IBM i object filter

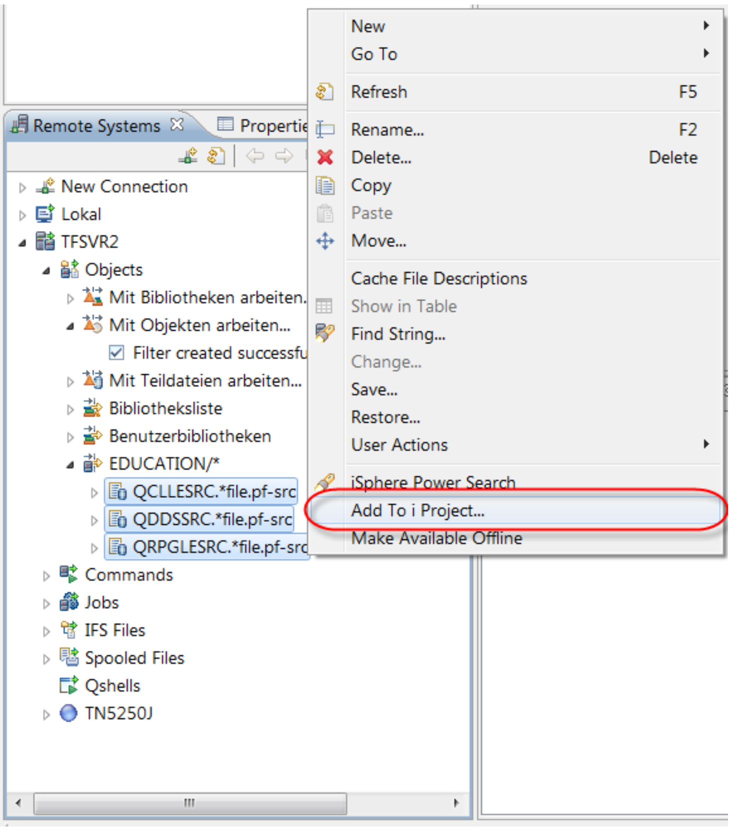
Library: EDUCATION

Object: *

Object type: *FILE

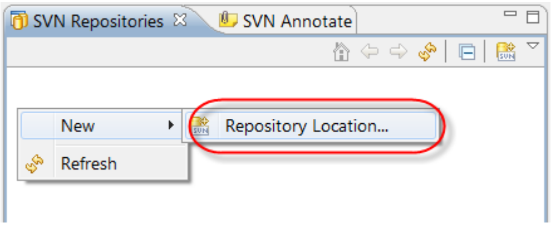
Object attribute: PF-SRC

Expand the object filter and select all source files. Open the context menu and select “Add To i Project”.

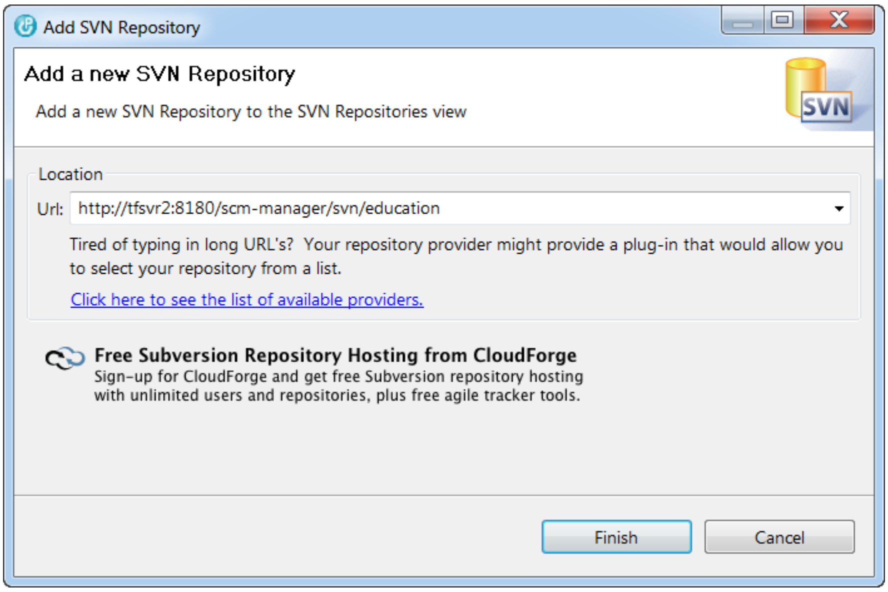


Now the sources have been imported to “i Projects”. After this we have to connect to the Subversion server. Open the perspective “SVN Repository Exploring” by selecting “Window/Open Perspective/Other/SVN Repository Exploring”.

Create a new repository location in “SVN Repositories” by opening the context menu and selecting “New/Repository Location”.



Enter the URL of the repository you have created several steps above. The Subversion server you have chosen will show you the URL needed to connect to the repository. Click “Finish”.



Add a new SVN Repository

Add a new SVN Repository to the SVN Repositories view

Location

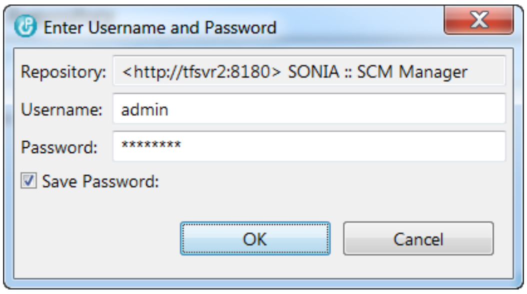
Url: http://tfsvr2:8180/scm-manager/svn/education

Tired of typing in long URL's? Your repository provider might provide a plug-in that would allow you to select your repository from a list.

Click here to see the list of available providers.

Free Subversion Repository Hosting from CloudForge
Sign-up for CloudForge and get free Subversion repository hosting with unlimited users and repositories, plus free agile tracker tools.

The Subversion server requests you for a user and a password. Enter the values and tick “Save Password”. Click “OK”.



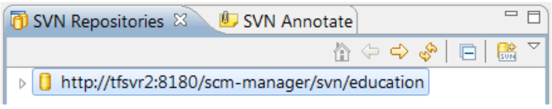
Repository: <http://tfsvr2:8180> SONIA :: SCM Manager

Username: admin

Password: ********

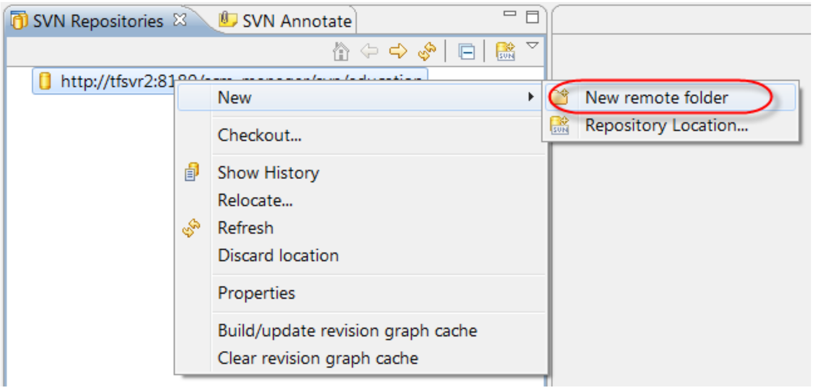
☑ Save Password:

If everything was fine a new repository location has been created.

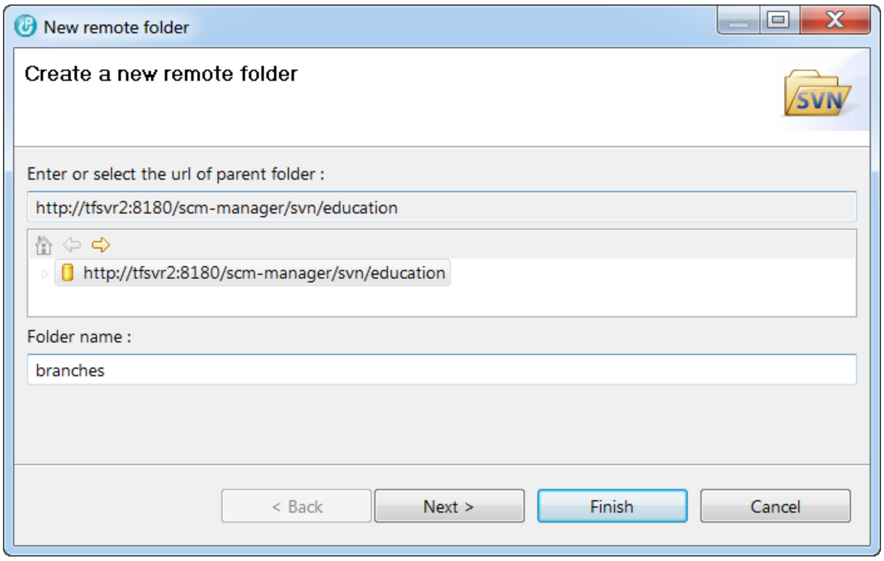


http://tfsvr2:8180/scm-manager/svn/education

Some Subversion servers create the directories branches, tags and trunk within the repository immediately after creation of a repository. Expand the repository location and if you don`t see the directories branches, tags and trunk we have to create them. Open the context menu for the repository location and select "New/New remote folder".



Enter folder "branches" and press "Finish". Repeat it with the folders "tags" and "trunk".


New remote folder

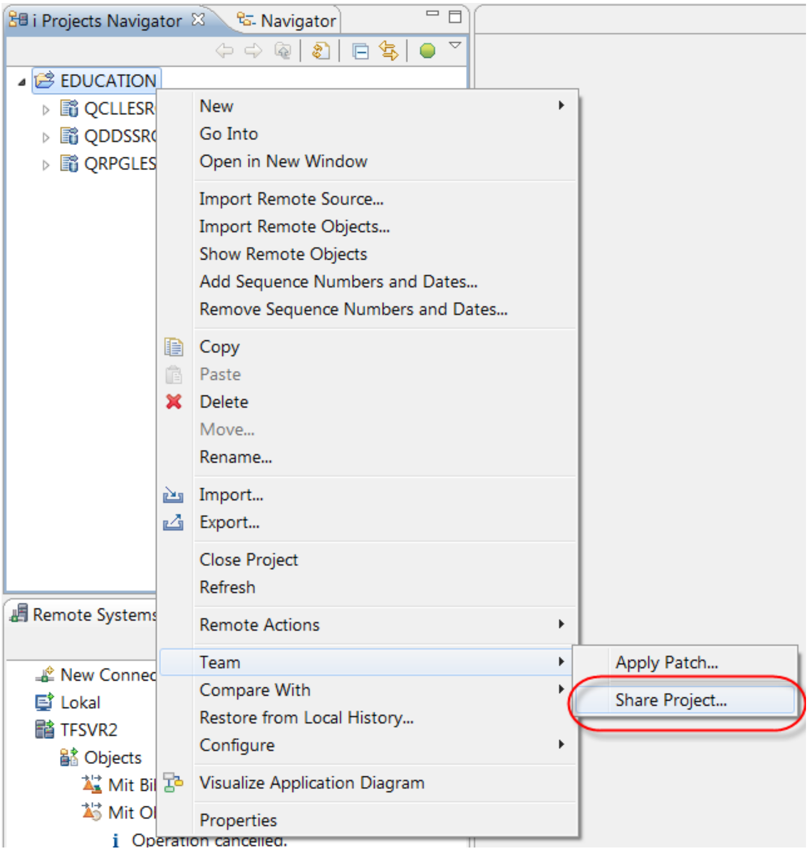
Create a new remote folder

Enter or select the url of parent folder : http://tfsvr2:8180/scm-manager/svn/education

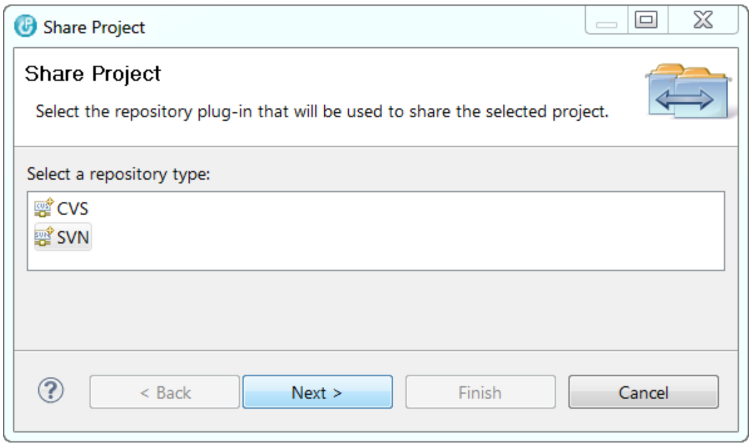
http://tfsvr2:8180/scm-manager/svn/education

Folder name : branches

The next step is to export the sources to the Subversion server. Go back to the perspective "i Projects" and open the context menu for your project and select "Team/Share Project".



Select “SVN” and click “Next”.



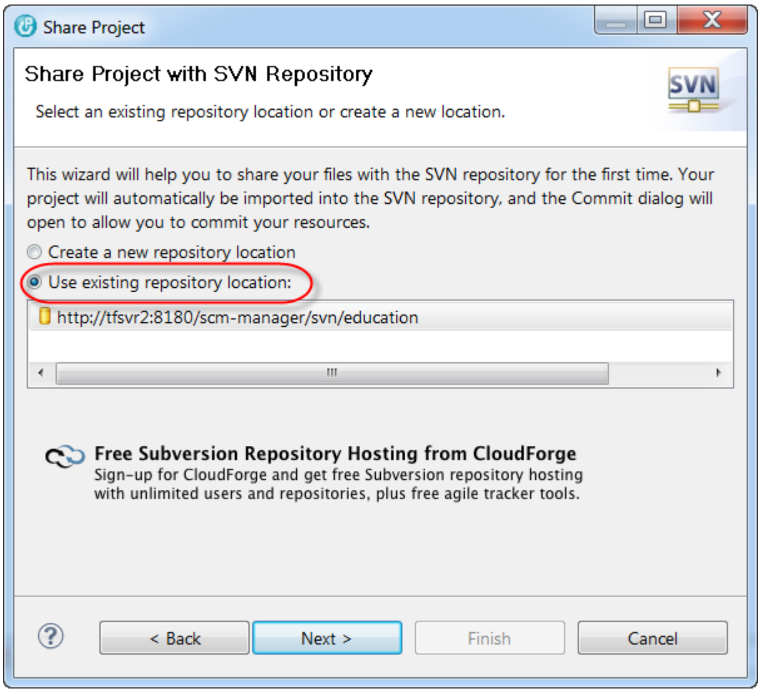
Share Project

Select the repository plug-in that will be used to share the selected project.

Select a repository type:

CVS
SVN

Select “Use existing repository location” and select the repository you want your project to export to. Click “Next”.



Share Project

Share Project with SVN Repository

Select an existing repository location or create a new location.

This wizard will help you to share your files with the SVN repository for the first time. Your project will automatically be imported into the SVN repository, and the Commit dialog will open to allow you to commit your resources.

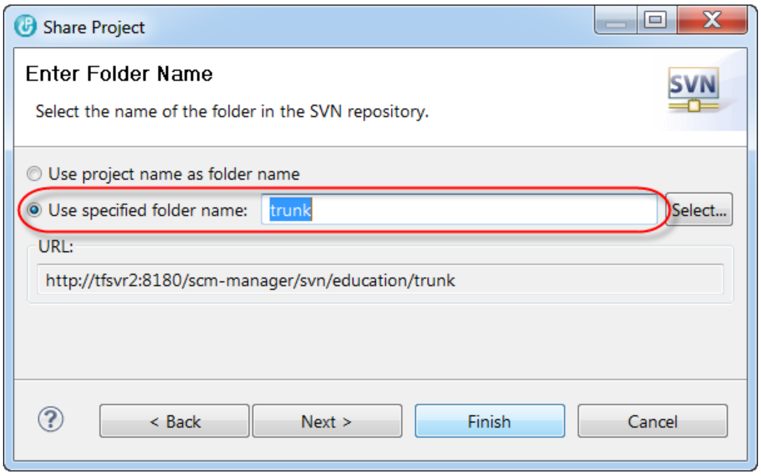
○ Create a new repository location

● Use existing repository location:

http://tfsvr2:8180/scm-manager/svn/education

Free Subversion Repository Hosting from CloudForge
Sign-up for CloudForge and get free Subversion repository hosting with unlimited users and repositories, plus free agile tracker tools.

Select “Use specified folder name” and enter “trunk”. Click “Finish”.



Share Project

Enter Folder Name

Select the name of the folder in the SVN repository.

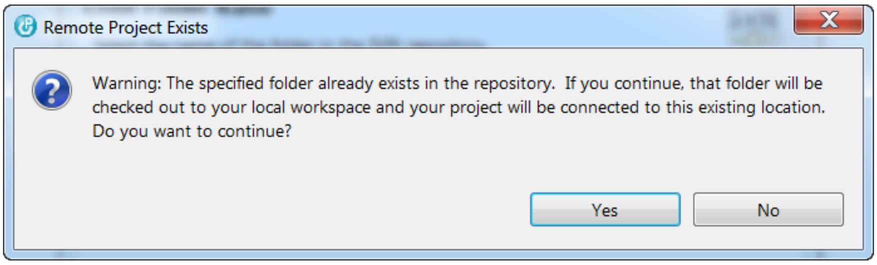
○ Use project name as folder name

● Use specified folder name: trunk

URL:

http://tfsvr2:8180/scm-manager/svn/education/trunk

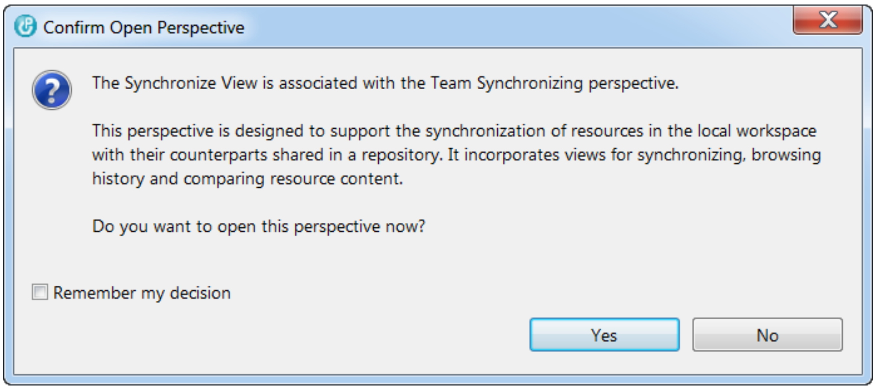
The following window appears. Click “Yes”.



Remote Project Exists

Warning: The specified folder already exists in the repository. If you continue, that folder will be checked out to your local workspace and your project will be connected to this existing location. Do you want to continue?

One further window appears. Click “Yes”.



Confirm Open Perspective

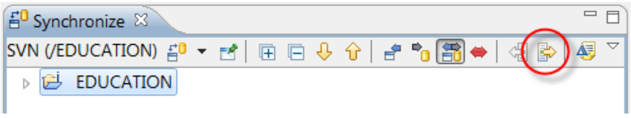
The Synchronize View is associated with the Team Synchronizing perspective.

This perspective is designed to support the synchronization of resources in the local workspace with their counterparts shared in a repository. It incorporates views for synchronizing, browsing history and comparing resource content.

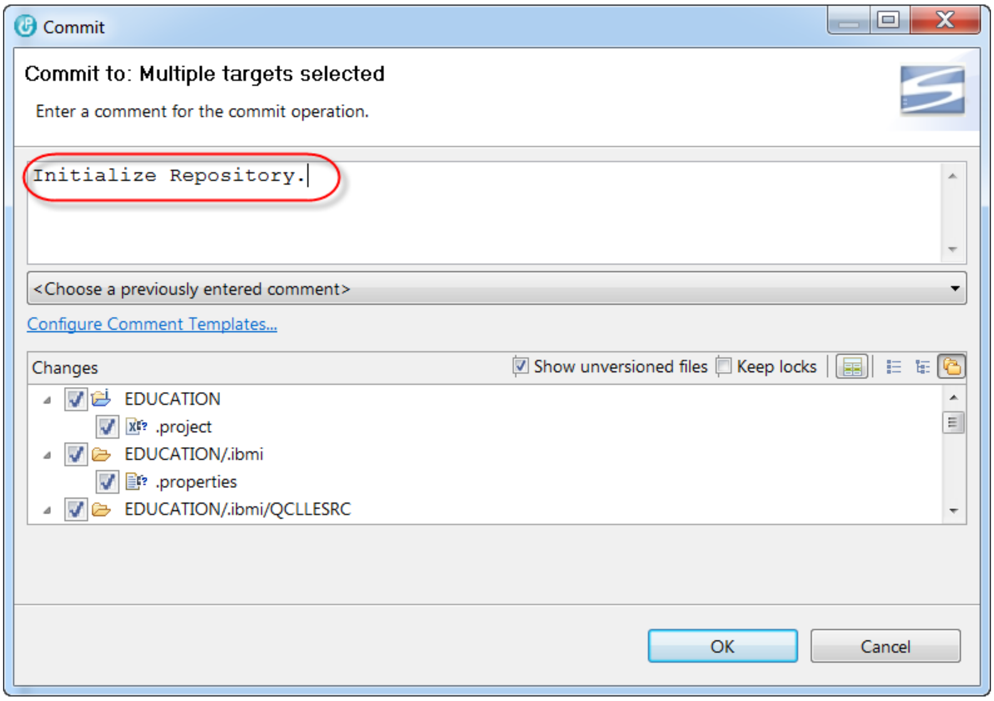
Do you want to open this perspective now?

□ Remember my decision

We are now in perspective "Team Synchronization". This view shows you the differences between your local workspace and the remote repository. Currently your locale workspace contains all the sources and the remote repository is empty. We will now export all of our sources to the repository. Click on the item "Commit all outgoing changes".



A windows pops up. Here you can enter a text for the commit operation. Click "OK".



Commit

Commit to: Multiple targets selected

Enter a comment for the commit operation.

```
Initialize Repository.
```

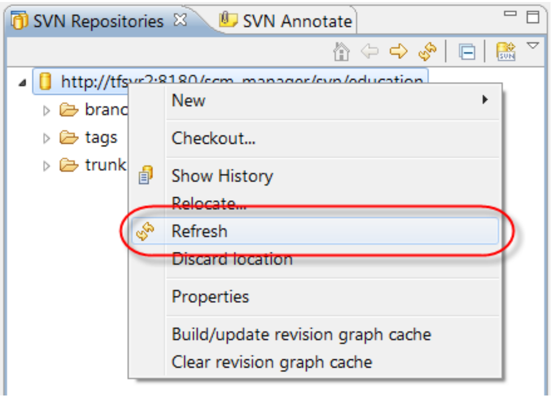
<Choose a previously entered comment>

Configure Comment Templates...

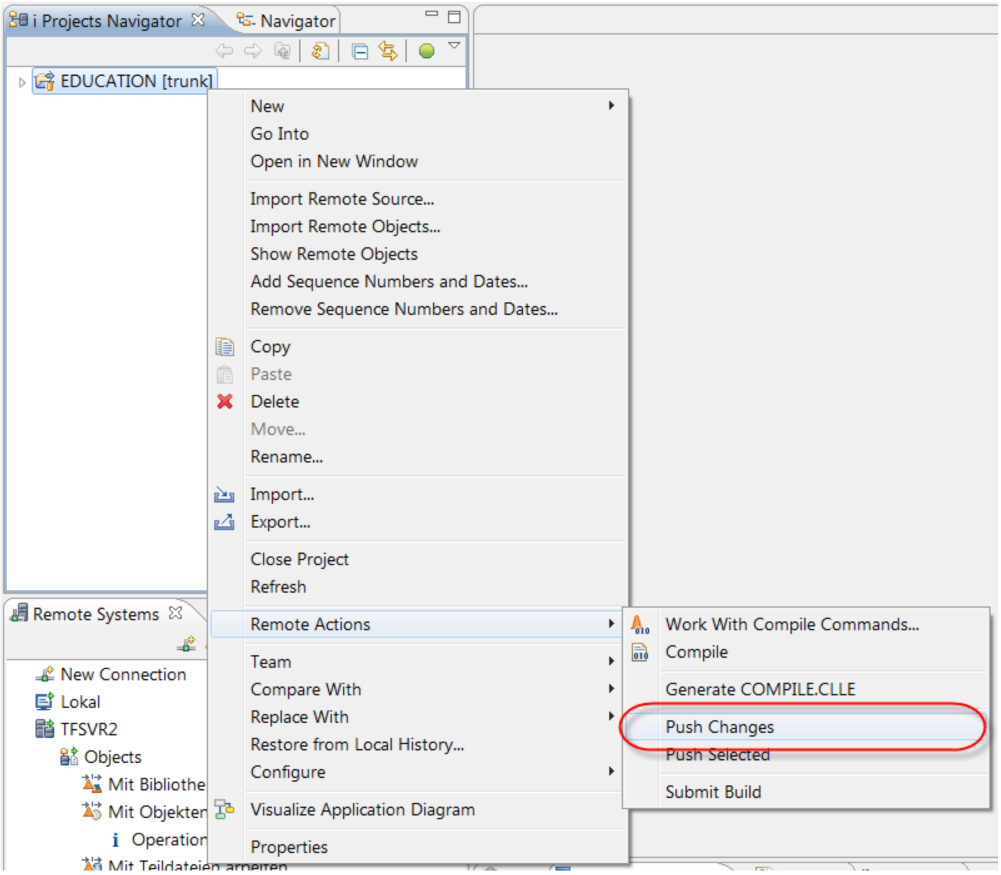
Changes ☑ Show unversioned files □ Keep locks

- ☑ EDUCATION
  - ☑ .project
- ☑ EDUCATION/.ibmi
  - ☑ .properties
- ☑ EDUCATION/.ibmi/QCLLESRC

Now your source files will be transferred to the repository. After this your locale working copy and the repository are in sync. Go back to the perspective "SVN Repository Exploring". Open the context menu for the repository location and select "Refresh".



Expand the directory “trunk” and you will see all of your sources. Go back to perspective “i Projects”. Open the context menu for your project and select “Remote Actions/Push Changes”.



All the sources will be transferred now to the host to the library you have assigned to the project.

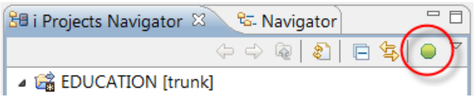
Congratulations. All the configuration and initialization stuff has done. Now we can start with our daily work.

Expand your project so you can see the members in your source files. The icon for each member shows this . This icon means that there are no changes made on the member in your local workspace.

Next we want to change a member. Prior you change a member open the context menu for the member and select “Team/Update to HEAD”. Subclipse looks now in the repository if there is a more current version of your member than in your local workspace. This can occur if a college of you has worked on the same member and has committed his changes to the repository. If Subclipse find a more current version it will be downloaded from the repository and applied to your local workspace. Updating to HEAD avoid you later from merging the changes between the different versions of the same member.

Open the LPEX-Editor for a member and add a new statement. After this save the changes. Now the icon for the member shows this . This icon means that there are changes made on the member in your local workspace.

In “i Projects Navigator” you see all members of your project. This can be confusing. Most of the time i want to see just the members which have changed. Here comes the iSVN plugin. Click on the green sphere in “i Projects Navigator” and you will see only changed members. Click on the green sphere again and you will see all members.



Open the context menu for the member and select “Compare with/Base Revision”. You will now see the two versions of your member with the changes.

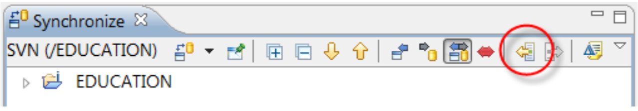
Let`s assume we want to reject the changes on the member we have made. Open the context menu for the member and select “Team/Revert”. All your changes will be rejected and the member is clean again.

Make another change to the member. Open the context menu for the member and select “Remote Actions/Compile”. The member will be compiled. When open the context menu for the project there are the menu items “Generate COMPILE.CLLE” and “Submit Build” in menu “Remote Actions”. Don`t use this menu items. The project contains now all your sources and not only those ones who have changed. Using this menu items means that all the sources of your application will be compiled.

Now let`s assume we are done with our work. We have to commit the changes to the repository. If i have ten changed members but i only want to commit one then i select only this one. If i want to commit five of ten changed members then i select only these five members. If i want to commit all changed members then is select the project. After selecting the specific items i open the context menu and select "Team/Commit". A window appears and you can enter a text to describe your changes.

Select the member you have changed. Open the context menu and select "Team/Show History". All the changes on this member will be shown. Double click an entry will show you the content of the member the member had the specified time.

Assuming multiple persons are working on the same project. That`s normal. The changes other persons have made and committed to the repository are not immediately applied to your local working copy. Therefore you have to synchronize your local working copy with the repository. You should do this one time per day. Open the context menu of your project and select "Team/Synchronize with Repository". A window "Confirm Open Perspective" appears. Click "Yes". Click on the icon "Update All Incoming Changes". A window appears. Click "OK".



Now your local working copy is up to date. The next what you have to do is to push the changes downloaded from the repository to the library assigned to your project. Open the context menu for the project again and select "Remote Actions/Push Changes".

In this tutorial you have done the initializing of the repository. If you have colleges which want to work on the same project there is no need for them to do all the steps described above too. Here is the description what they have to do.

Open perspective "SVN Repository Exploring". Create a repository location as described above. Expand the repository location. Open the context menu for directory "trunk" and select "Checkout". A window appears "Checkout from SVN". Click "Finish". Go back to perspective "i Projects". Open the context menu for the checked out project and select "Properties/i Project". Assign a valid connection and enter a library for your work. Click "Apply". Click "OK". Open the context menu for the project again and select "Remote Actions/Push Changes". The members will be transferred to the specified library. Now he can do his work as described above.

I have described the most important functions of Subclipse and Subversion. There are a lot of more functions especially handling conflicts when several people working on the same members. There are a lot of really good books where you can learn all aspects of Subversion so i don`t have to tell you this.

I would like to finish now and I hope you will find this tutorial useful for your daily work.

Have a great day.

Frank Hildebrandt