

# DAVE™ Usage with SVN

Presentation and Tutorial v 2.0

May, 2014

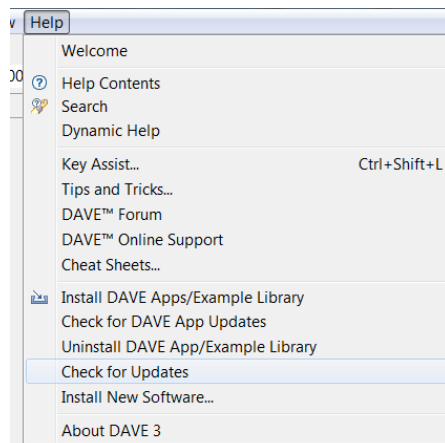


# Required DAVE™ Version

- Required DAVE version: v 3.1.6 or higher (recommend to use the most latest version, as of Feb 28, 2014, v 3.1.10)
- Required Eclipse plug-in update:

Installed Software		Installation History	Features	Plug-ins	Configur
Name	Version				
> C/C++ Development Tools	8.0.2.201202111925				
> DAVE3	1.0.0.201203191804				
DAVE3 CE	1.0.52.201305241711				
DAVE3 IDE	1.0.40.201304121532				
DAVE3 Library Manager	1.0.40.201304121532				
DAVE3 MBS XMC4000	1.0.40.201304121532				
DAVE3 Product Branding	1.0.52.201305241711				

- To make sure that you have the latest plug-in update follow these instructions:



Press: ->Help ->Check for Updates

# Table of contents

■ General Considerations

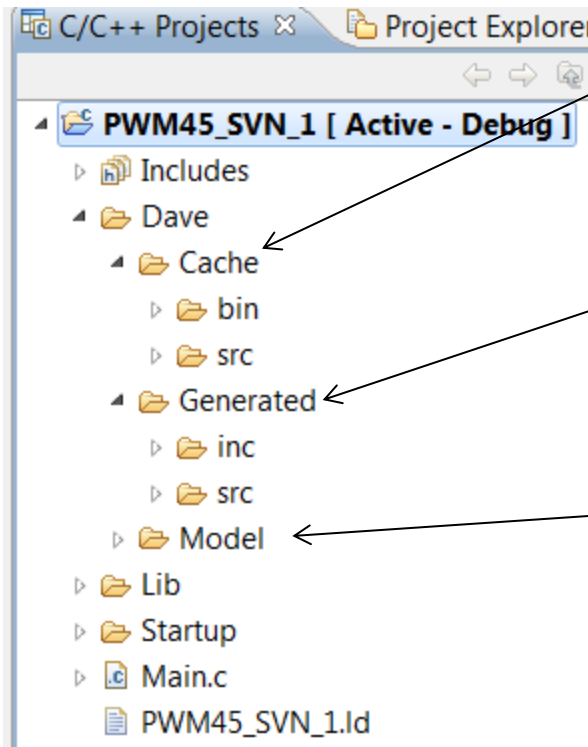
■ Eclipse Subversive Overview

■ Eclipse Subversive Installation in DAVE

■ DAVE CE Example Project usage with SVN

# General Information How to Use DAVE Generated Sources in a Version Control SW

Initial folder structure when creating a new DAVE CE project,  
adding some DAVE Apps and generate code



Contains cached code template class files (intermediate files of a code generation process) to improve operation speed

Contains the generated sources and header files

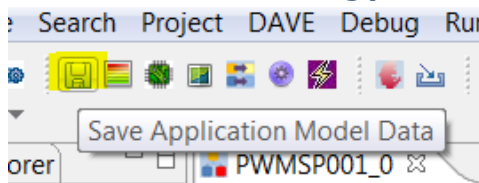
Model Folder contains the model of each DAVE App (static coded, templates, UI models, configurations) that has been added to the project with the current configuration settings

Other files are: startup files, linker script stubs for the c library and a simple main.c

Folder with build artifacts added after first build

# Version Control of the DAVE Generated Code

- DAVE will generate the code from the files stored in the model folder by utilizing the device model of the chosen device that is in the local library store located in the user folder
- Therefore it is only required to put all files of the Dave model folder under version control
- Putting the Dave Generated folder and the Dave Cache folder under version control is not required
  - Recommendation: only the Dave Model folder and may be the Dave Generated folder should be put under version control  
cache folders and .codecache file should be assigned as svn ignore before committing)



- Before committing the Dave Model folder into a repository the application model has to be saved (all cached memory information to be saved in App instance file)
- After updating the Dave Model folder from the repository the code needs to be generated (assuming Dave Generated folder is not under version control)

# What changes the files in the Dave Model folder

Project changes	File changes in the Dave Model folder
A new DAVE App is added to the project	New files and folders are added
A new instance of an existing DAVE App is added	A new App instance file (<app_instance>.app) will be created that contains configuration data
Configuration changes	The content and date of the related App instance file will be changed and the date of all other instance files will also change (same content)
Code generation	No content change in the model folder but date of App instance files will change
Deletion of an App instance	The related App instance file will be deleted; date of remaining App instance files will change
Opening DAVE	No change; date of all App instance files will change
Close DAVE	No content change if no previous configurations changes already saved, but date of all App instance files will change. If previous configuration changes are not saved, will be saved now -> content of related App instance file will change
Switch active project	Same as close DAVE for the current active project, same as open DAVE for the new active project

- To avoid version changes based on changed date information a version control system that verifies the content is required, like SVN

# What changes the files in the Dave Generated folder

Project changes	File changes in the Dave Generated folder
A new DAVE App is added to the project	No change
A new instance of an existing DAVE App is added	No change
Configuration changes	No change
Code generation	All exiting files will be deleted and new files will be created. If there was no model change before the content did not change but the date
Deletion of an App instance	No change
Opening DAVE	No change
Close DAVE	No change
Switch active project	No change

# Merging and Conflicts

- It is not allowed to let SVN merge model files
- In case a second user wants to commit a changed model folder that has been differently changed by another user previously, svn will offer a merge functionality, this merge functionality should not be executed.
- Second user should first update the working copy then make the required changes in the model file and then commit



# Table of contents

■ General Considerations

■ Eclipse Subversive Overview

■ Eclipse Subversive Installation in DAVE

■ DAVE CE Example Project usage with SVN

# Eclipse Subversive

- Subversive plug-in provides access to Subversion (SVN) repositories from the DAVE v 3 (Eclipse) workbench
  - <http://www.eclipse.org/subversive/index.php>
- **Subversive Features**
  - **Full-Scale SVN Client**
    - Subversive is designed to be used as a full-featured SVN client, so you can update, commit, merge changes, work with SVN properties, view change history and perform other operations with SVN directly from the Eclipse environment.
  - **Advanced SVN Features**
    - Subversive includes several features that extend functionality of the standard SVN client. In particular, Subversive can show the SVN repository content grouped by the logical structures of trunk, branch and tag and display changes on a visual revisions graph.
  - **Seamless Integration with Eclipse**
    - Subversive is an official Eclipse project and an integral part of Eclipse Simultaneous releases. The project follows all Eclipse guidelines and requirements to deliver a quality SVN team provider plug-in similar to CVS and Git implementations.
  - **Support of the Latest SVN Versions**
    - Subversive evolves together with the Subversion project to provide Eclipse users with the features that appeared in new versions of the SVN implementation. You can use the new SVN functionality in Eclipse by installing the Early Access version of Subversive

# Table of contents

■ General Considerations

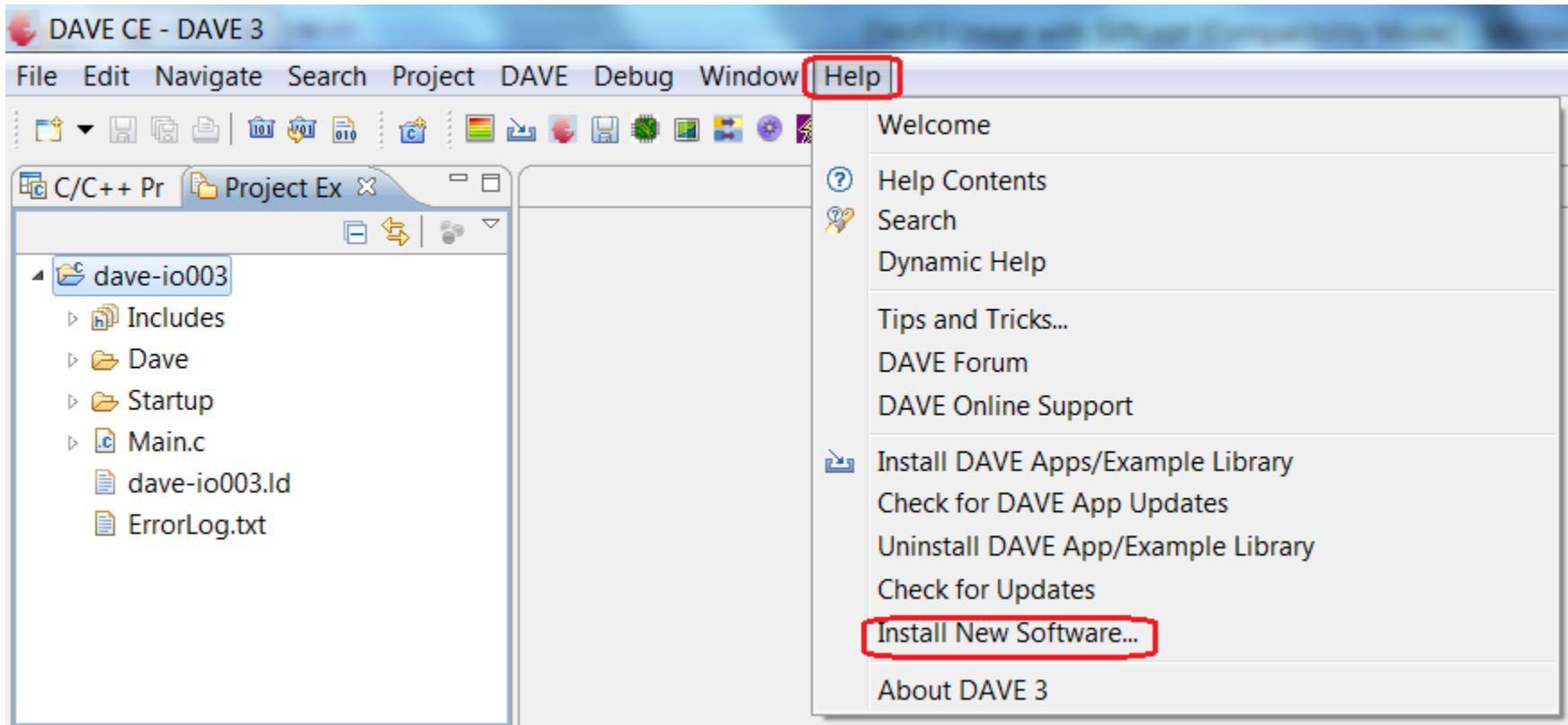
■ Eclipse Subversive Overview

■ Eclipse Subversive Installation in DAVE

■ DAVE CE Example Project usage with SVN

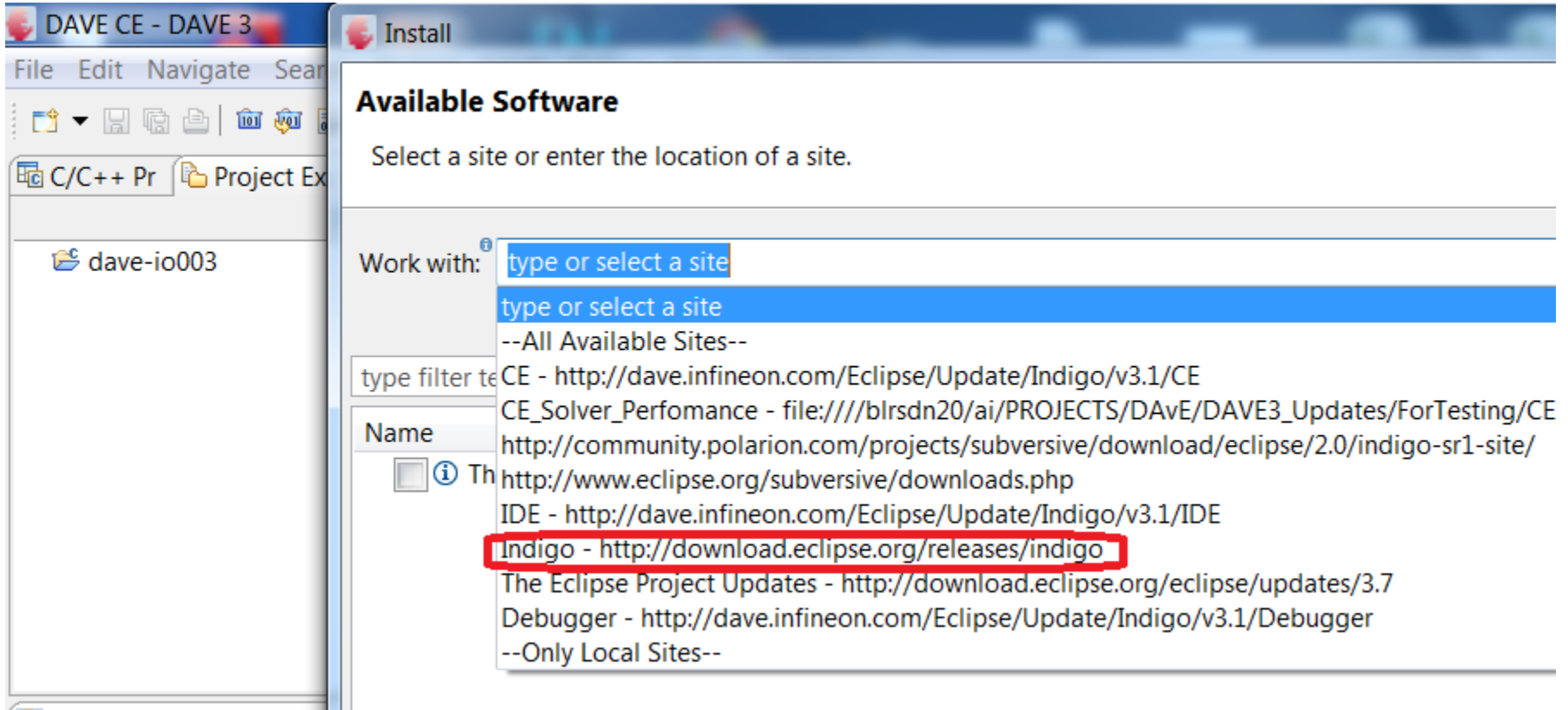
# Eclipse Subversive Installation Instructions (I)

- Step 1: Start DAVE and select menu item *'Help > Install New Software...'*



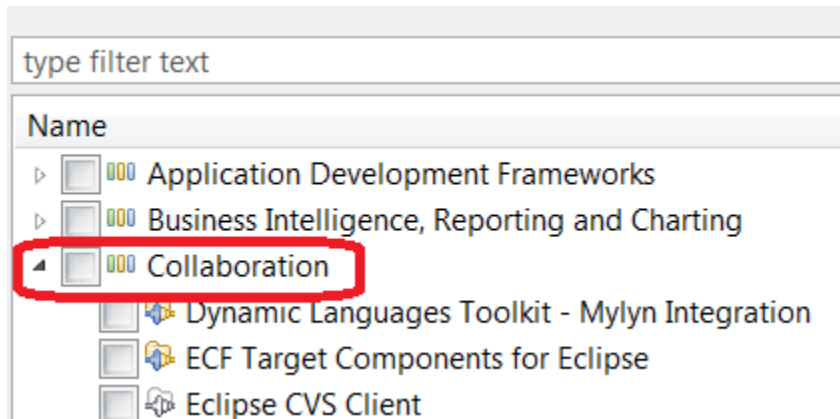
# Eclipse Subversive Installation Instructions (II)

- Step 2: Select Eclipse Indigo link as per below

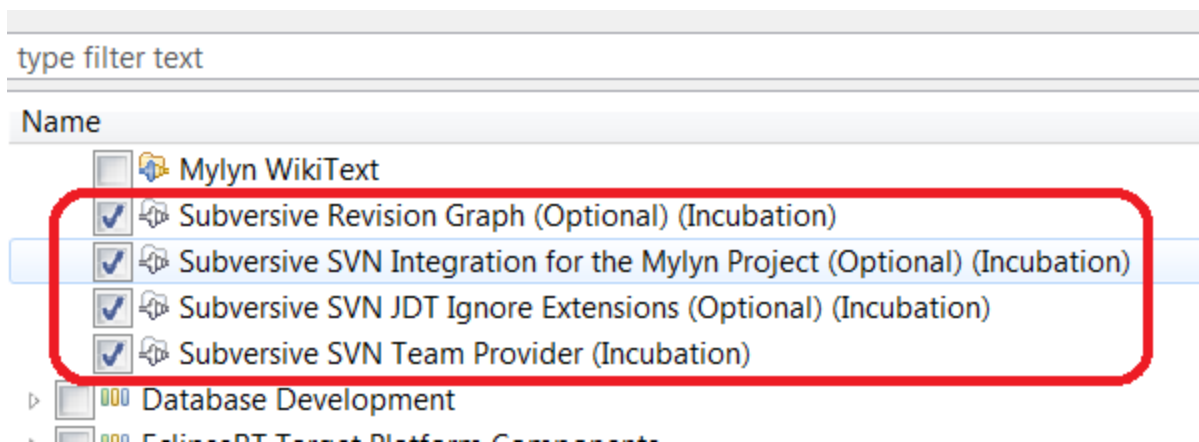


# Eclipse Subversive Installation Instructions (III)

- Step 3: Expand the Collaboration option



- Step 4: Select the Subversive Features & then click “Next”

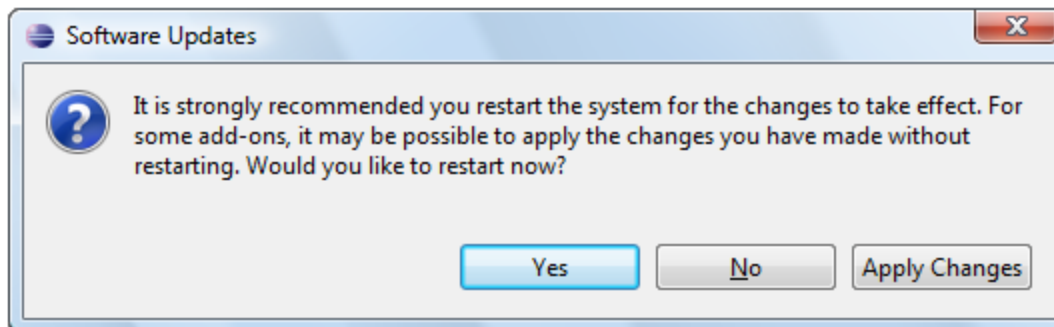


# Eclipse Subversive Installation Instructions (IV)

- Step 5: The update manager calculates dependencies and offers you a list of features to install. Select the needed ones and click the 'Next >' button

Name	Version
<input checked="" type="checkbox"/> Subversive Revision Graph (Optional) (Incubation)	0.7.9.I20110602-1700
<input checked="" type="checkbox"/> Subversive SVN Integration for the Mylyn Project (Optional) (Inc...	0.7.9.I20110602-1700
<input checked="" type="checkbox"/> Subversive SVN JDT Ignore Extensions (Optional) (Incubation)	0.7.9.I20110602-1700
<input checked="" type="checkbox"/> Subversive SVN Team Provider (Incubation)	0.7.9.I20110819-1700

- Step 6: Accept terms of license agreement and click the 'Finish' button in order to start the download of selected features.
- Step 7: To apply installation changes and restart Eclipse click on the 'Yes' button.



# Eclipse Subversive Installation Instructions (V)

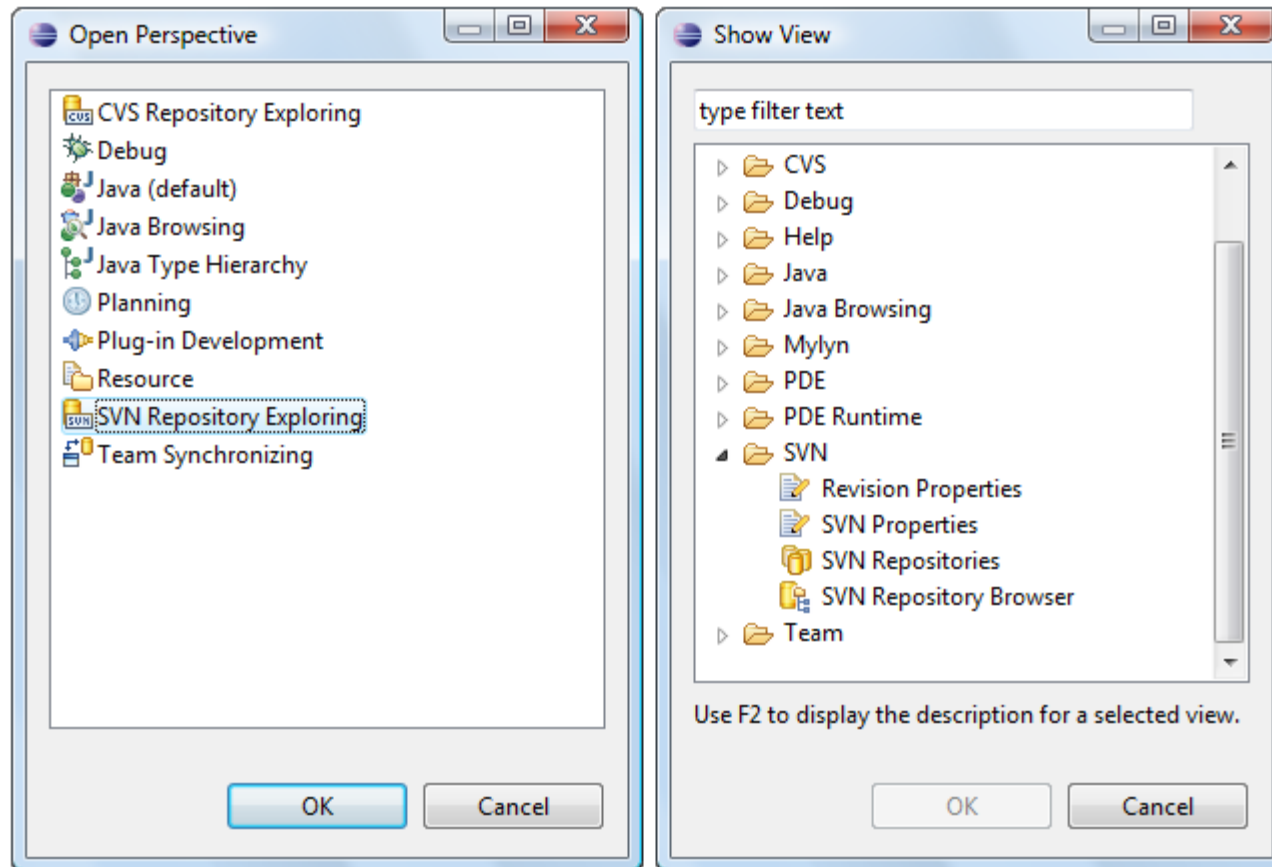
- Step 8: After Eclipse restart you'll see connectors discovery dialog which will allow you to install Subversive Connectors without registering connectors update site manually. Install the appropriate connector as per your subversion repository





# Eclipse Subversive Installation Instructions (VI)

- Step 9: You can find Subversive perspective and views in correspondent dialogs, activated by menu items *'Window > Open Perspective > Other...'* and *'Window > Show View > Other...'*.



# Team Support with SVN

- Various Features of Eclipse Plug-in Usage with SVN are provided below

<http://www.eclipse.org/subversive/documentation/teamSupport.php>

- SVN Exploring, Locating..etc

# Team Support with SVN

- Various Features of Eclipse Plug-in Usage with SVN are provided below

<http://www.eclipse.org/subversive/documentation/teamSupport.php>

- SVN Exploring, Locating..etc

# Table of contents

■ General Considerations

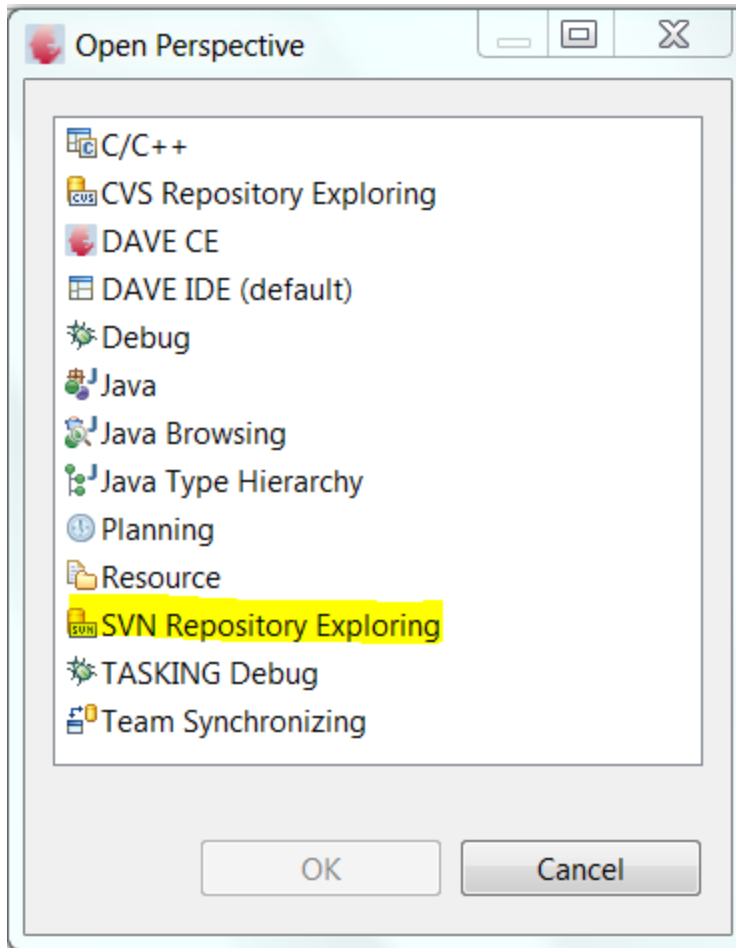
■ Eclipse Subversive Overview

■ Eclipse Subversive Installation in DAVE

■ DAVE CE Example Project usage with SVN

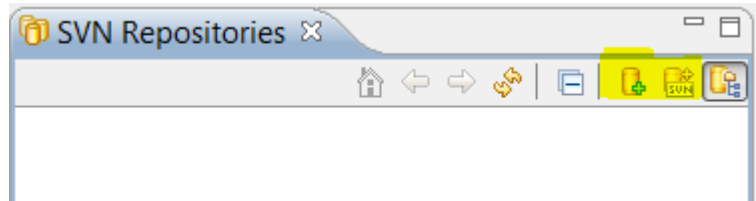
# Open the SVN related Perspectives

- Step1: Select the "SVN Repository Exploring" Perspective



# Connect to an Existing repository or Create a new One

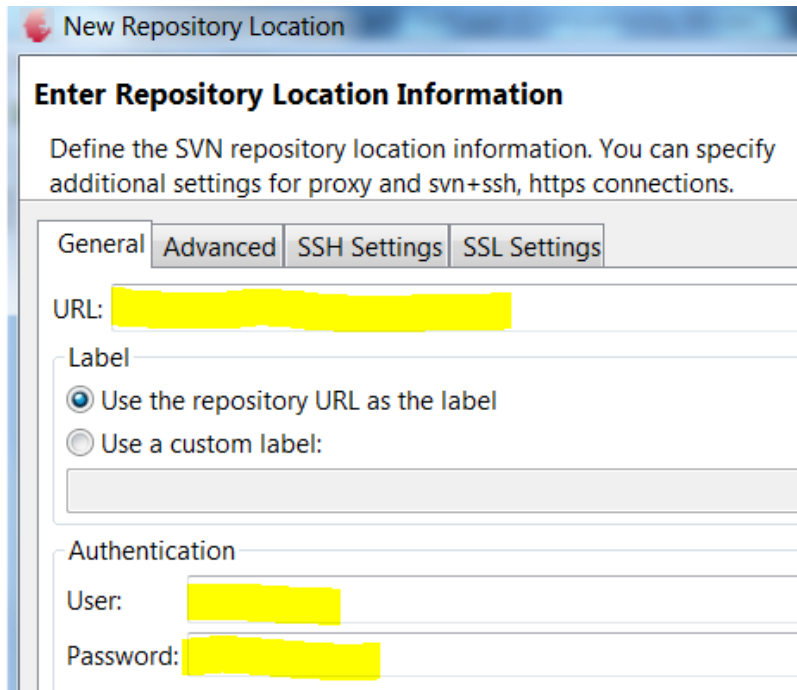
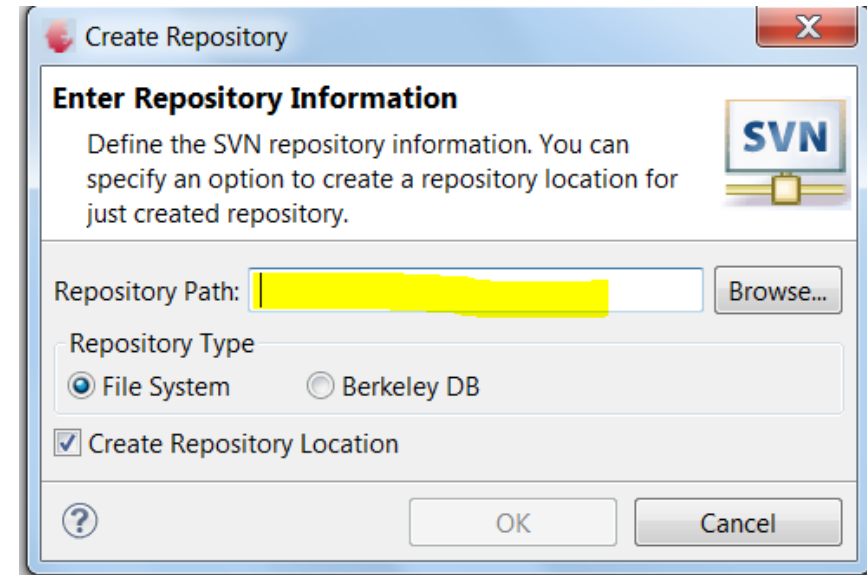
- Step2: Select SVN Repository Location or Create New Repository



## Existing SVN Repository

or

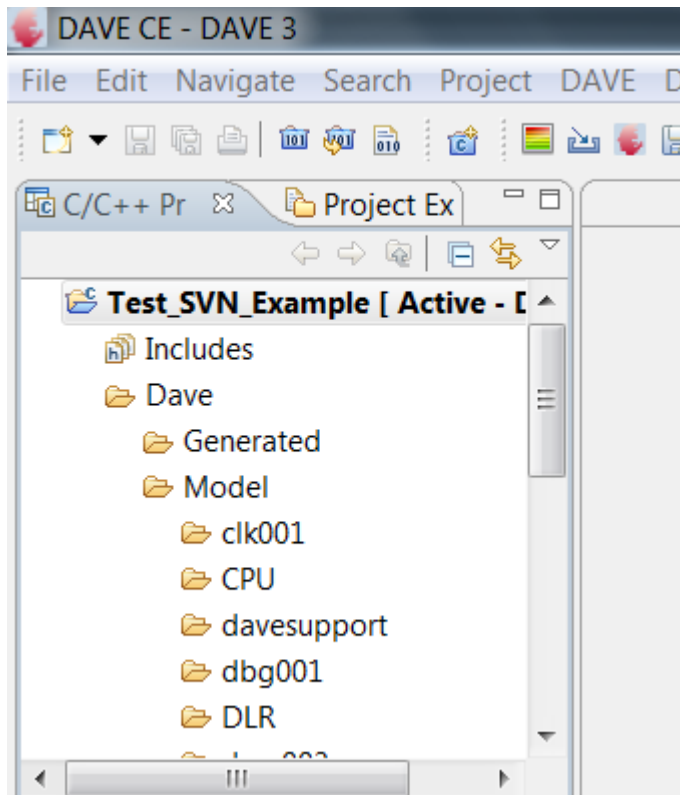
## New SVN Repository

A screenshot of the 'New Repository Location' dialog box. The title bar says 'New Repository Location'. The main section is titled 'Enter Repository Location Information' and contains the text: 'Define the SVN repository location information. You can specify additional settings for proxy and svn+ssh, https connections.' Below this are four tabs: 'General', 'Advanced', 'SSH Settings', and 'SSL Settings'. The 'General' tab is selected. It contains a 'URL:' field with a yellow highlight, a 'Label' section with two radio buttons ('Use the repository URL as the label' is selected), and an 'Authentication' section with 'User:' and 'Password:' fields, both with yellow highlights.A screenshot of the 'Create Repository' dialog box. The title bar says 'Create Repository'. The main section is titled 'Enter Repository Information' and contains the text: 'Define the SVN repository information. You can specify an option to create a repository location for just created repository.' Below this is a 'Repository Path:' field with a yellow highlight and a 'Browse...' button. There is a 'Repository Type' section with two radio buttons ('File System' is selected) and a 'Create Repository Location' checkbox which is checked. At the bottom are 'OK' and 'Cancel' buttons.

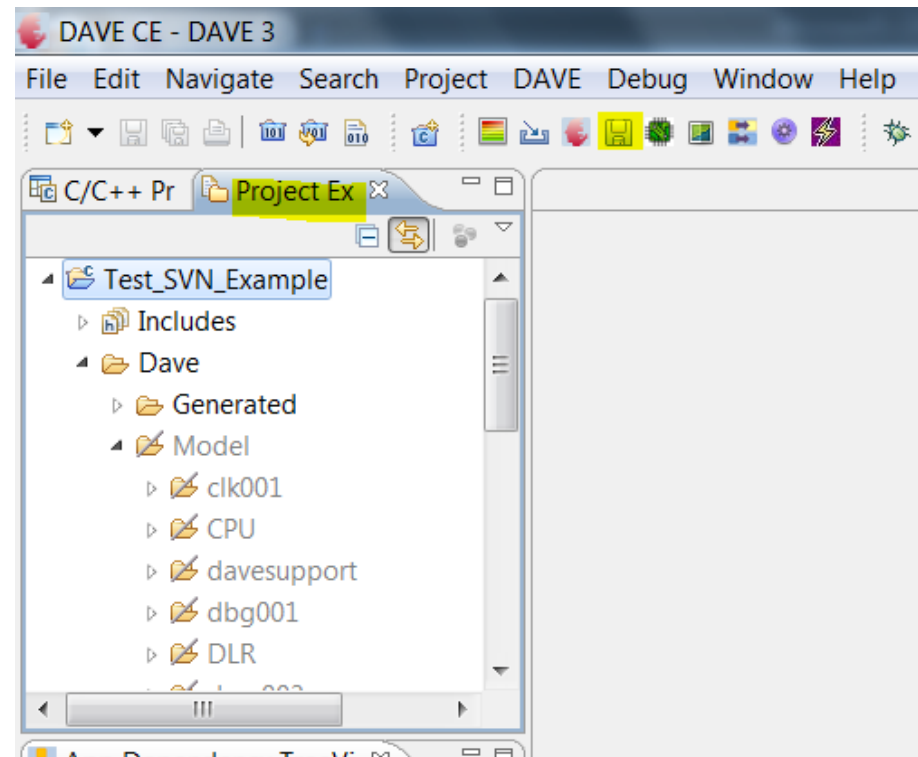
For test purposes a repository on a local drive can be created; Browse to the desired location

# Project to be imported to a SVN Repository

Step3: Select Active Project to be source controlled with SVN

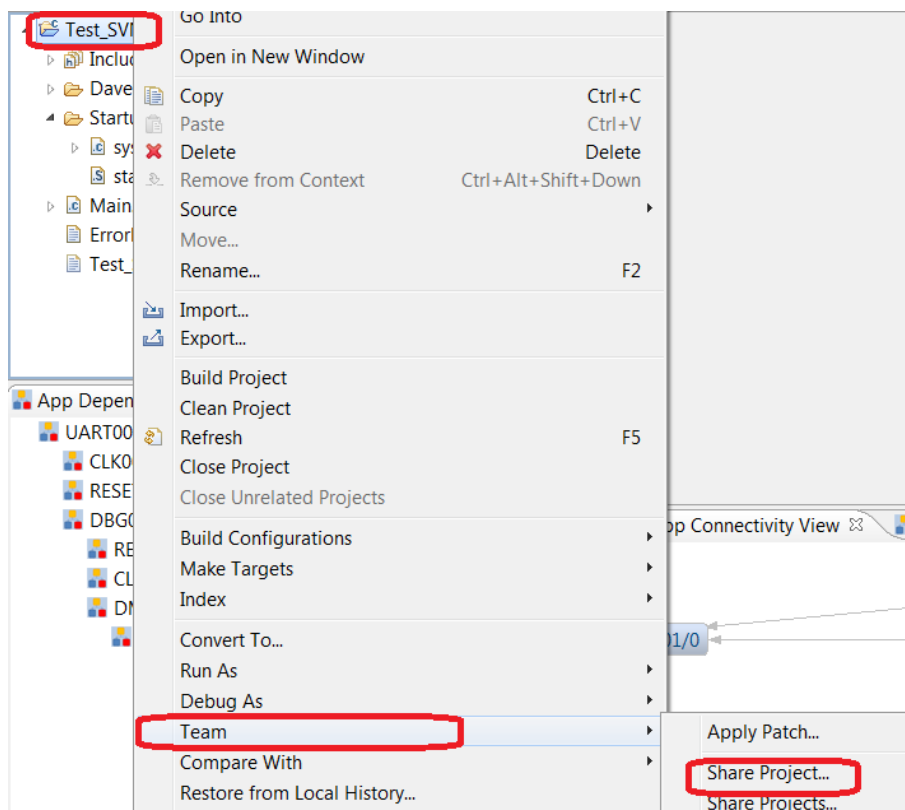


Step4: Save Application Model and Switch to Project Explorer Tab



# Import the project to a SVN Repository

Step5: Before the first import to the repository DAVE should not generate code or redundant folders (DAVE Generated, DAVE Cache folder, artifact folder,...), if already done, theses folders may be deleted.

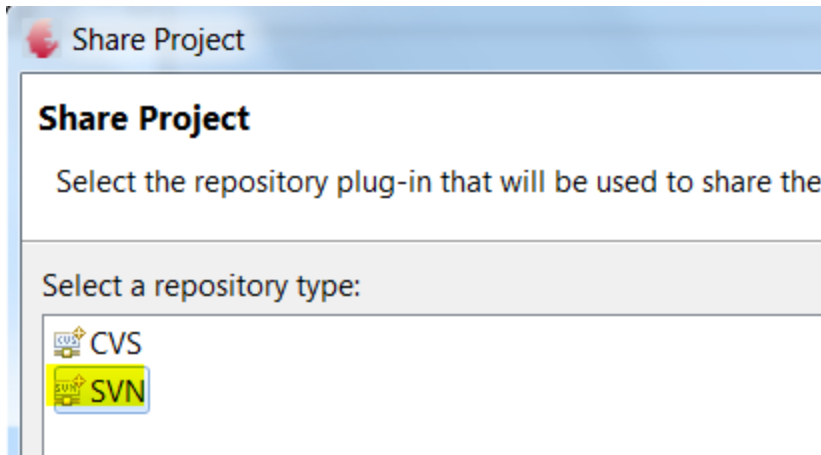


If the DAVE Generated folder should be also put under version control then the code has to be generated and the folders that should not be put under version control should be deleted.

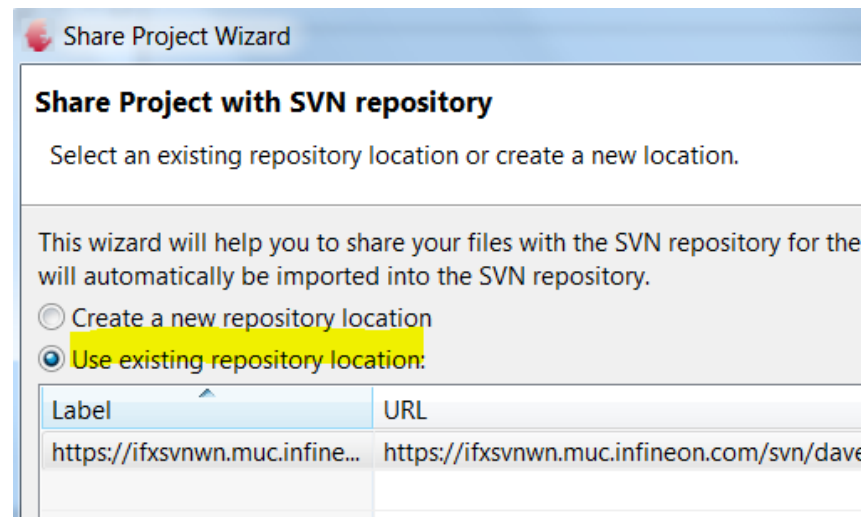


# Import the project to a SVN Repository

Step6: Right Click Project, Select Team & Share Project

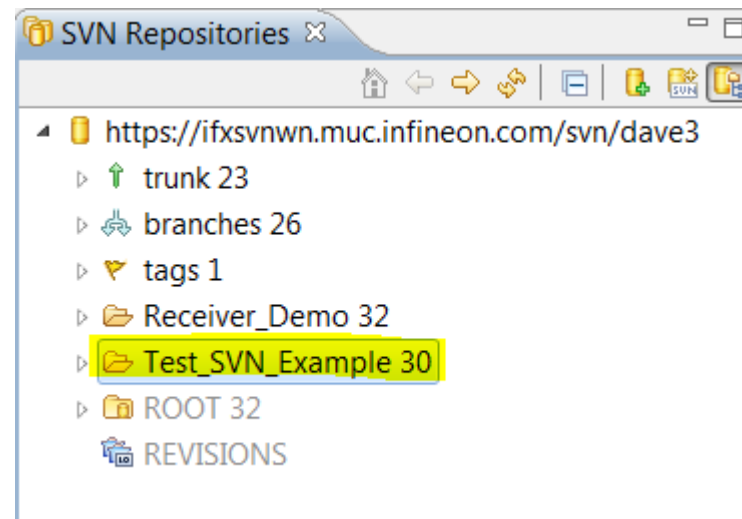
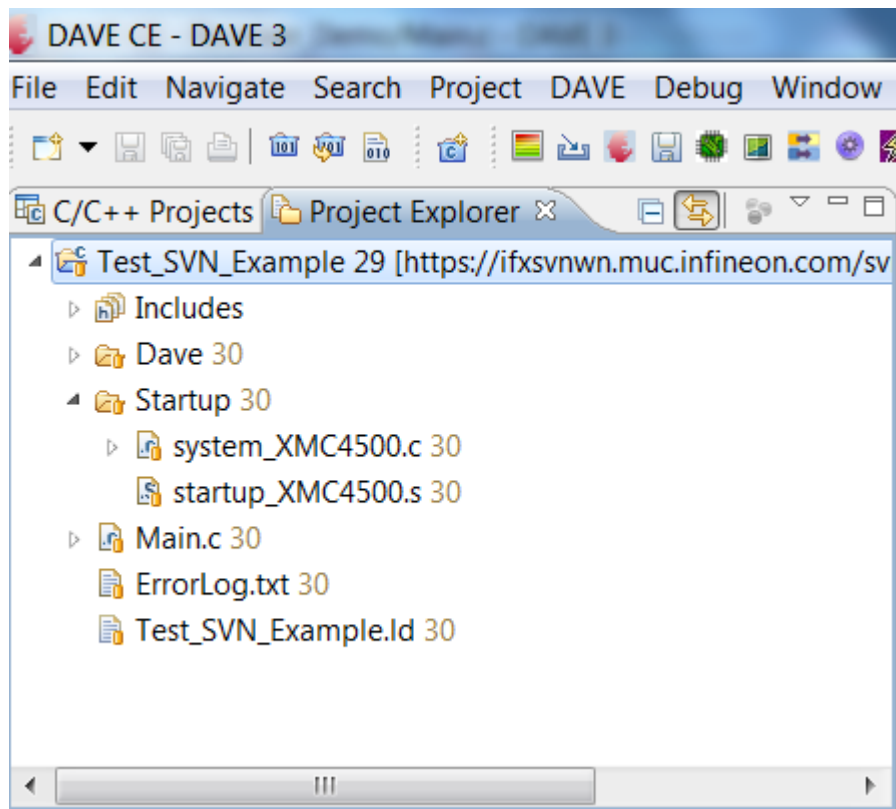


Step7: Select SVN repository & Click "Finish"



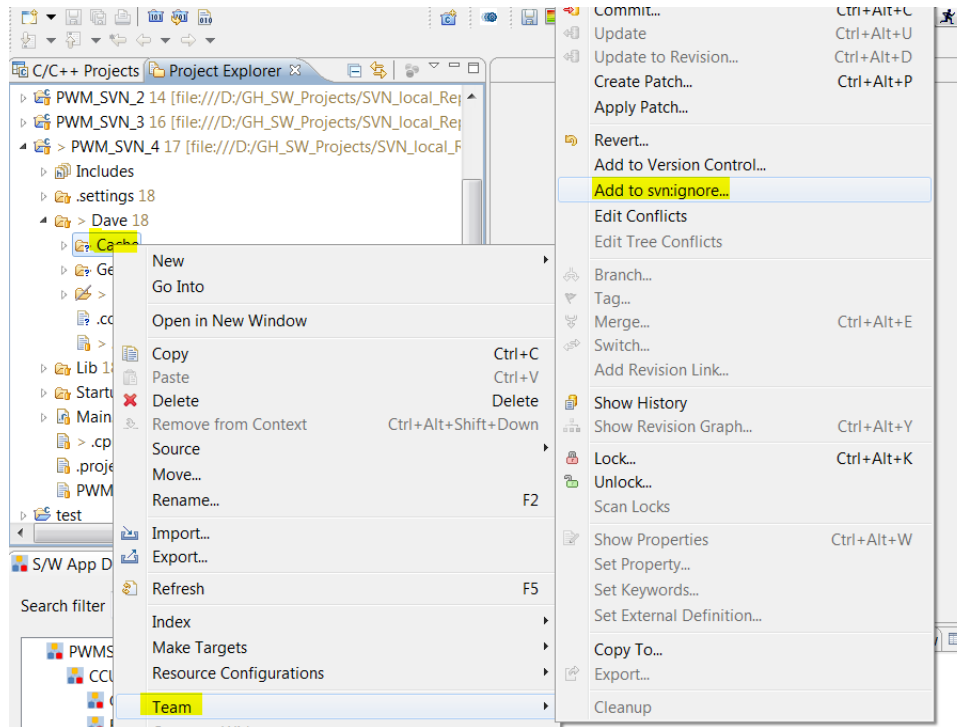
# Import the project to a SVN Repository

Step8: The project is added (imported) to SVN Repository Now!!



# Assign folders that should not be under version control as “svn ignore”

- The “svn ignore” attribute can only be assigned to files and folder if the project is added (imported) in a SVN repository.
- First the folders that should be marked as “svn ignore” has to be created by pressing code generation and build (if the artifact folder should also be ignored)



Step 9: Each folder that should be marked as svn ignore should be handled like shown left:

- Right click on the folder
- Select Team
- Select Add to svn ignore

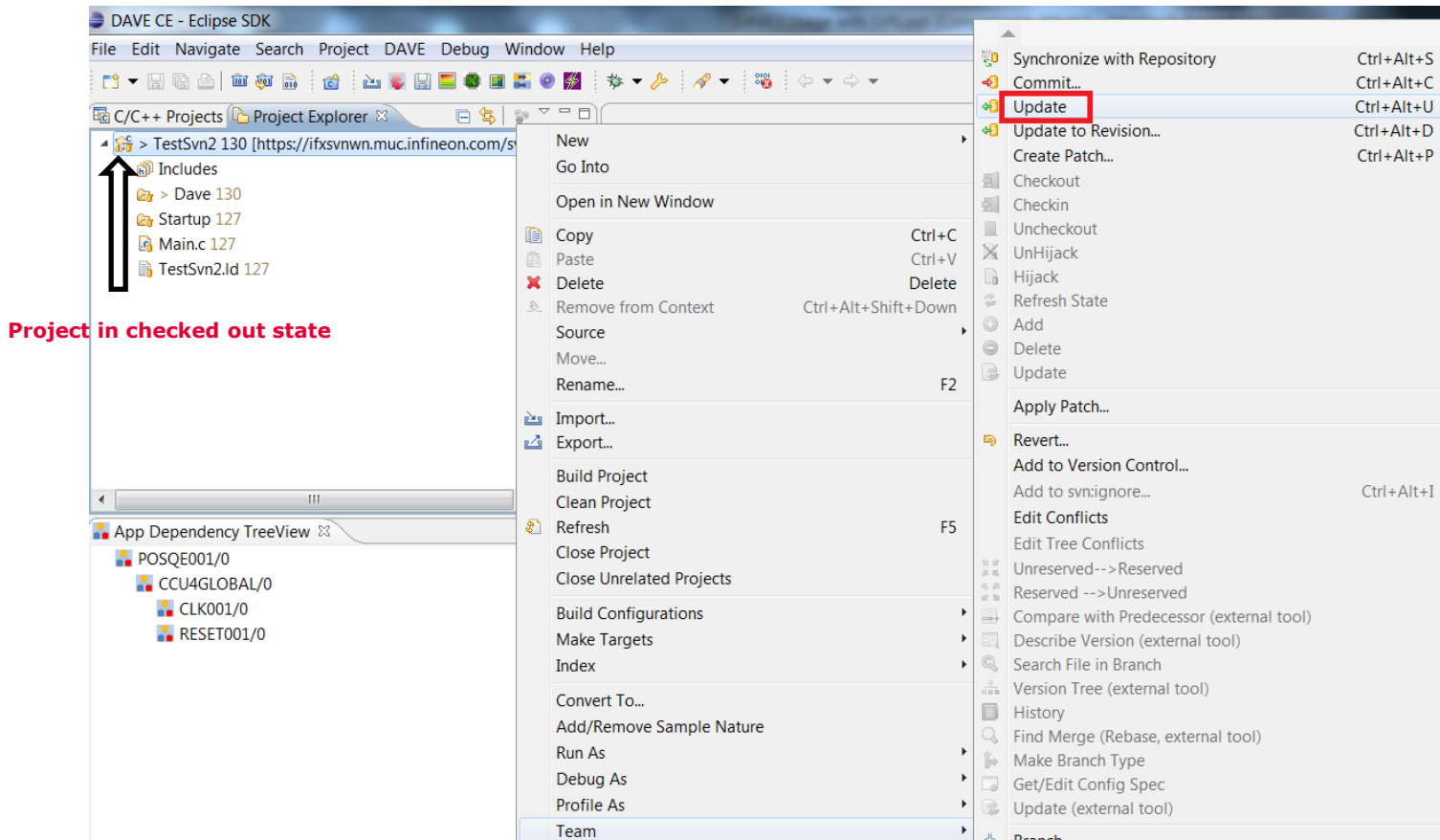
We recommend to do this for the Cache folder and file: .codecache in the Dave folder and also the artifact folder

# Rules to work with SVN

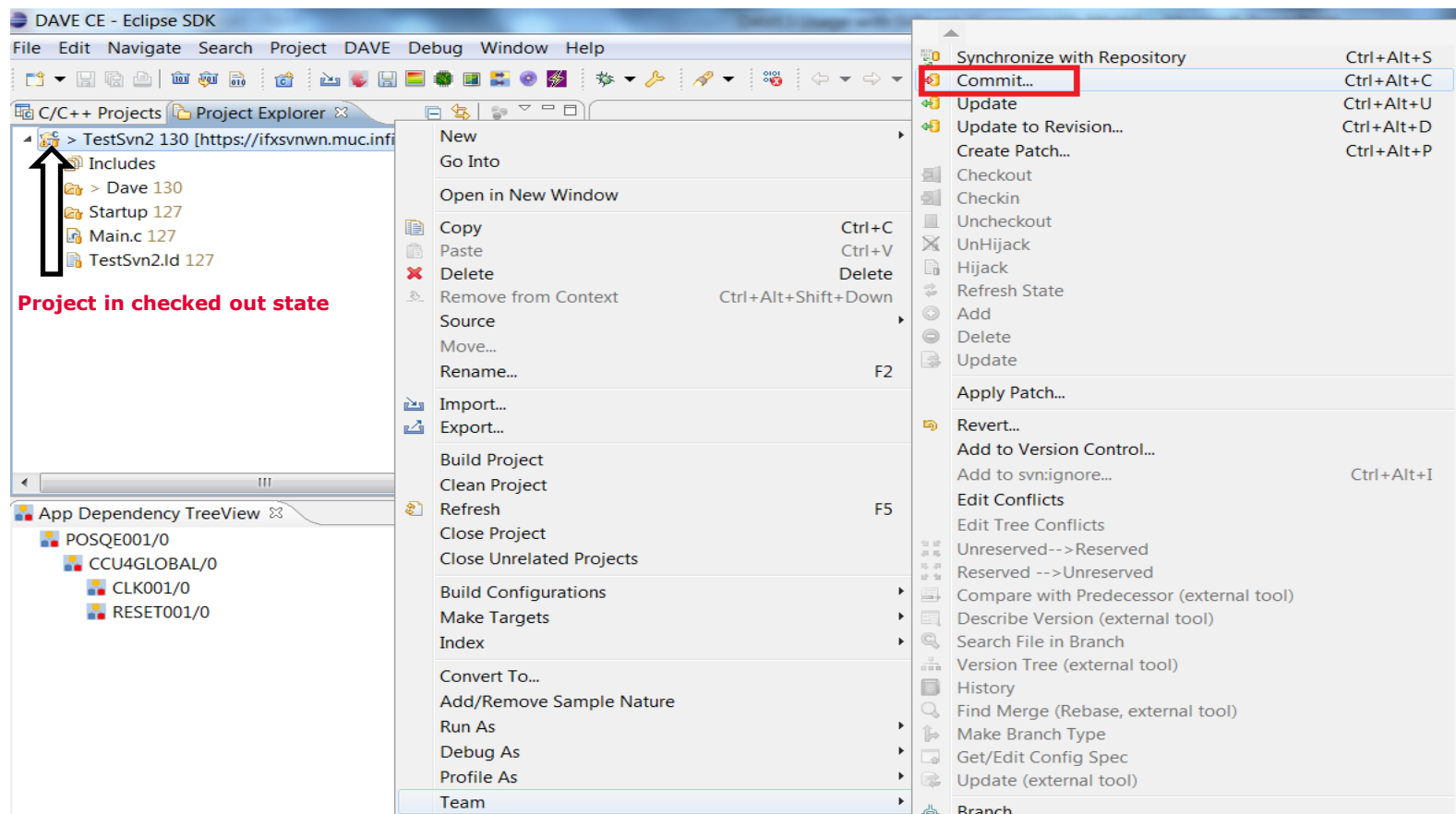
- Step10: Before doing any project modifications make sure that you are working on the most current project version by update your local copy from SVN repository using command SVN->Update
- Rules to make sure that changes are correctly committed to the SVN repository
  - 1) Update local copy from SVN: SVN->Update
  - 2) Make the required changes in the project
  - 3) Commit your changes to SVN using SVN->Commit
  - 4) Check if project is still in check-out mode [using the project icons in the project explorer, see following pages]
  - 5) If yes again start from steps 3
  - 6) Refer attached SVN FAQ for more info



# Step 1: Update local copy from SVN



# Step 3: Commit your changes to SVN



# Alternatives to Subversive

- Instead of Subversive also the eclipses plug-in Subclipse can be used
  - Subclipse is usually more up to date in following the latest updates of svn
- Instead of an eclipse plug-in, tortoises might be used
  - Client and repo browser that is integrated in the windows file explorer
  - After the initial project import all svn commands are available via the windows project explorer

# Summary

- The DAVE generated code can be put under version control as any other project files
- To avoid redundancies only the Dave Model folder should be put under version control
  - The Dave Cache folder should be marked as ignored by the version control system
  - The Dave Generated folder should also be marked as ignored
    - To avoid regeneration it could also be kept under version control
- The version control systems should verify the content of the files and not the date of the last change
  - SVN can be recommended
  - CVS can not be recommended





# ENERGY EFFICIENCY MOBILITY SECURITY

Innovative semiconductor solutions for energy efficiency, mobility and security.

