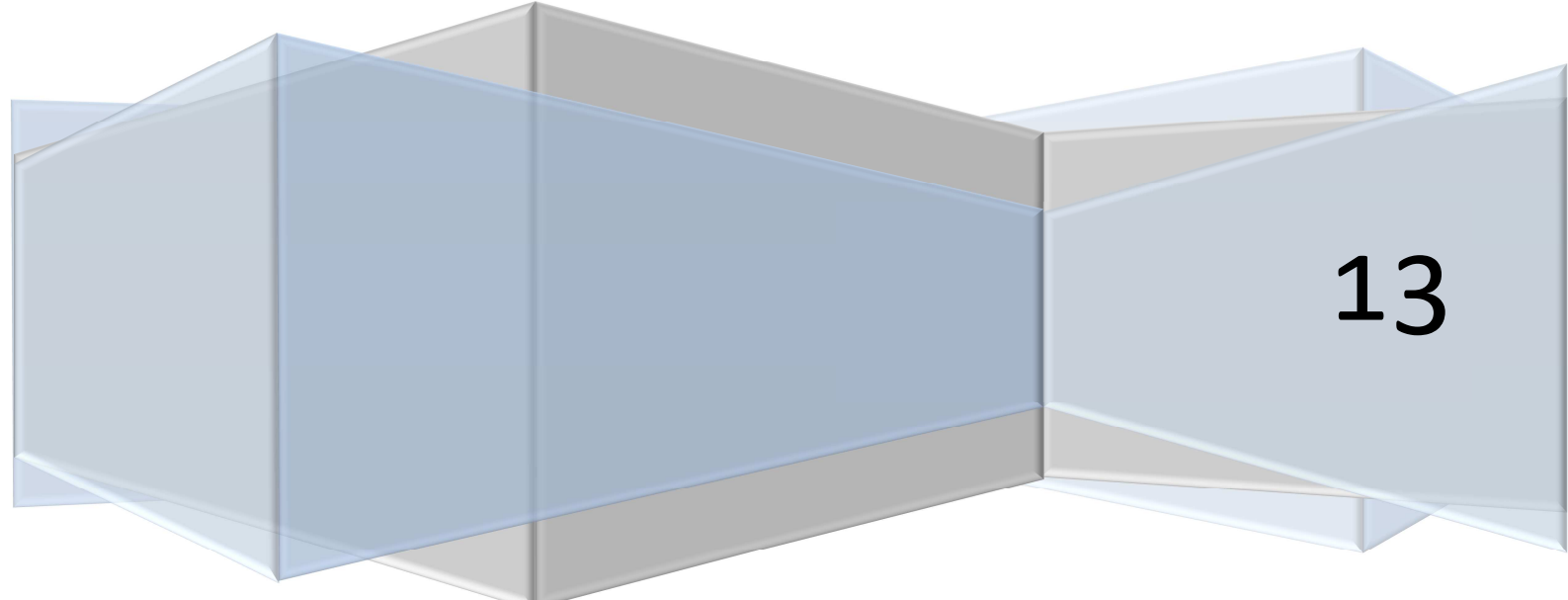


Source Control with Domino Designer

Overview & getting started

Patrick Kwinten



13

Contents

Introduction.....	3
Document version.....	3
Used Sources	3
Presentation /Blog	3
Video	3
Source Control.....	4
What is source control?	4
Central vs. Distributed repository.....	5
Central (client/server)	5
Distributed.....	5
Terminology (Distributed)	5
Source Control in DDE.....	6
Classic way.....	6
Disadvantages:.....	6
From DDE 8.5.3	6
Configure DDE for source control	8
Enable source control for NSF	8
Install EGit to DDE	12
Install CVS	22
Install Mercurial in DDE.....	34
Actions in DDE with source control	43
Synching.....	43
Committing (Mercurial).....	46
Reverting (Mercurial)	50
Rollback	52
Backout.....	52
Stripping	52
Branching (Mercurial).....	52
Branching (CVS).....	54
Switching	57
Merging (Mercurial)	59

Merging (CVS) 64

Tagging (Mercurial) 65

Tagging (VCS) 67

Cloning (Mercurial) 68

Pulling (Mercurial)..... 72

Get data out of source control (CVS) 75

Get data out of source control (eGIT) 79

Best practices..... 80

Tips for DDE 80

Steps (hand-on) 81

Appendix A – List of Terms..... 81

Introduction

This document describes source control and how to use it with Domino Designer. On a high level source control platforms are described. On a low level setup and usage with Domino Designer are explained.

After reading this document, you as a developer should decide if source control can be beneficial in your projects. This document might give you guidance in taking the first steps.

Note that in the 'Lotus Domino Designer User Guide', section 'Administering applications' you can find a good introduction about source control enablement.

Document version

Date	Version	Author	Comments
2013-09-04	0.1	Patrick Kwinten	Initial Draft
2013-09-12	0.2	Patrick Kwinten	Corrections

Used Sources

Besides the help topics from the Designer User Guide other sources have been used to write this summary document.

Presentation /Blog

Title	Author
AD102 : Source Control For The IBM Lotus® Domino® Developer	Declan Sciolla-Lynch
Integrate Domino Designer with CVS Source Control	Frank van der Linden
Version Control for Notes developers	Martin Jinoch
Introduction to source control with Domino Designer 8.5.3 and Git	Per Henrik Lausten
Getting your Git projects back in Domino Designer	Mark Leusink

Video

Title	Author
Part 1 : Installing Mercurial Source Control in Domino Designer 9	Russell Maher
Part 2 : Basic Repo Usage: Committing and History	Russell Maher
Part 3 : Connecting To A Remote Mercurial Repository	Russell Maher
Part 4 : Introduction to Source Control Branching	Russell Maher
Part 5 : Mercurial Branching and Merging	Russell Maher
Part 6 : Using Source Control Tags	Russell Maher
Part 7 : Pulling Changesets from a Remote Repository	Russell Maher

Part 8: Using Revert and Changing Repo NSF Files	Russell Maher
Part 9: Our Mercurial Source Control Strategy	Russell Maher
NotesIn9 041: XPages, Designer – SOURCE CONTROL!!	David Leedy
NotesIn9 044: Using Source Control Part 2	David Leedy

Source Control

What is source control?

Wikipedia defines source control as "Revision control (also known as version control, source control or (source) code management (SCM)) is the management of multiple revisions of the same unit of information."

In other words: source control allows distributed work in teams of any size, at different locations, while avoiding conflicts in source code changes.

Purposes of source control:

- Ability to capture snapshots of your application as it is developed.
- Ability to merge code from those snapshots.
- Ability to track changes to individual elements across snapshots.

Other advantages source control:

- More efficient code review.
- Better linkage with issue and bug tracking management tools.
- Better data mining of code.

Source Code Management requirements:

- Intuitive:
 - A good GUI.
 - Good visualization.
- Robust.
- Affordable.
- Distributed with branching capability.
- Native windows application.
- Command line capable.
- Strong community support.

Central vs. Distributed repository

Central (client/server)

- Uses a central repository on a server.
- Requires an active network connection to this server.
- Higher chance of getting change conflicts when committing.
- Branching and merging are often more complex.
- Central repository always reflects the latest version.
- Developers need to serialize their work.

System examples

- Subversion (SVN), CVS, IBM Rational ClearCase, IBM Rational Team Concert, Microsoft Team Foundation Server.

Distributed

- Each developer has a working copy of the repository locally on their machine (peer-to-peer approach).
- Committing, branching and merging is all done locally (developers can work offline).
- Can push and pull changes from any other local or remote repository.
- No central repository means no definitive 'latest version':
 - You can create a server based repository that each person pulls/pushes with.
 - Increases the possibility of merging conflicts.
- Developers can work independent of each other and merge their work later.

System examples

- GIT, Bazaar, Mercurial.

Terminology (Distributed)

- Commit (save current state).
- Diff (show the differences between commits (versions)).
- Revert (undo (any) commit).
- Branch (make a new “copy” of the whole app).
- Switch.
- Merge (apply changes from one branch to another).
- Branch.
- Tag.
- Clone.
- Pull.
- Push.

Source Control in DDE

Classic way

Name ▲	Alias
Another Copy Of Response	
Copy Of Response	
Response	resp
Response.orig	

Disadvantages:

- No easy way to get back to previous state unless we are keeping:
 - Backup versions of design elements.
 - Backup copies of the whole template.
- We don't know what the last change was. We need to maintain this information in a separate change log.

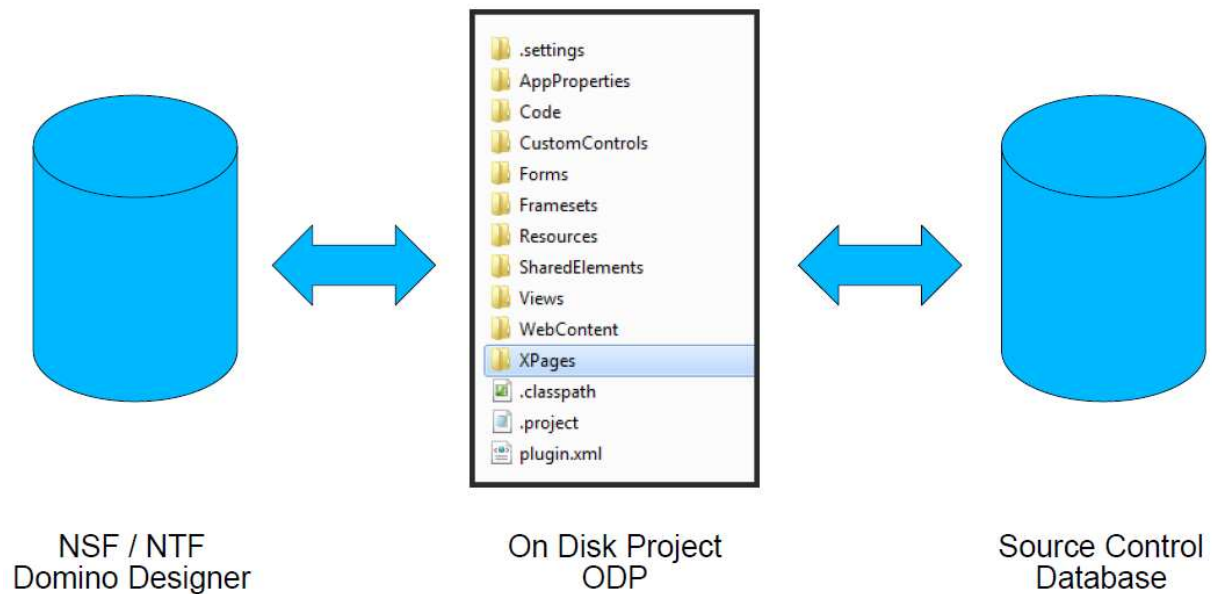
From DDE 8.5.3

From 8.5.3 source control enablement it is built-in to DDE. It lets you store NSF based design elements into a source control repository and perform common source control operations using Designer. The feature does this by letting you create an association between an **on-disk project** managed by the source control and the virtual NSF based project that Designer tracks.

- Version control systems need real files to act on, not virtual file system.
- Some design elements have to be converted to DXL.

The feature lets you integrate an SVN source control client with Designer as seamlessly as possible and to let Designer stay “agnostic” to each source control system as much as possible.

The figure below visualizes how changes in a Notes application (NSF) are synchronized (automatic by default) with a SCM system via an on-disk Eclipse project and vice versa.



NSF	On-disk project	Source Control system
This is where you (typically) do all your development.	Basically a local backup of the database; not associated to any particular source control tool.	

Note: It is the on-disk project which is under source control management.

Managing source code requires a tool:

- Install in Domino Designer (as an Eclipse plug-in install).
 - Examples: Mercurial for Eclipse, SVN for Eclipse, Egit.
- Install outside of Domino Designer.
 - Examples: TortoiseCVS, TortoiseSVN, TortoiseGit, CVS.
- Install on Mac if you run Windows in a VM.
 - Examples: Sourcetree, Tower.

Why do you want source control in DDE?

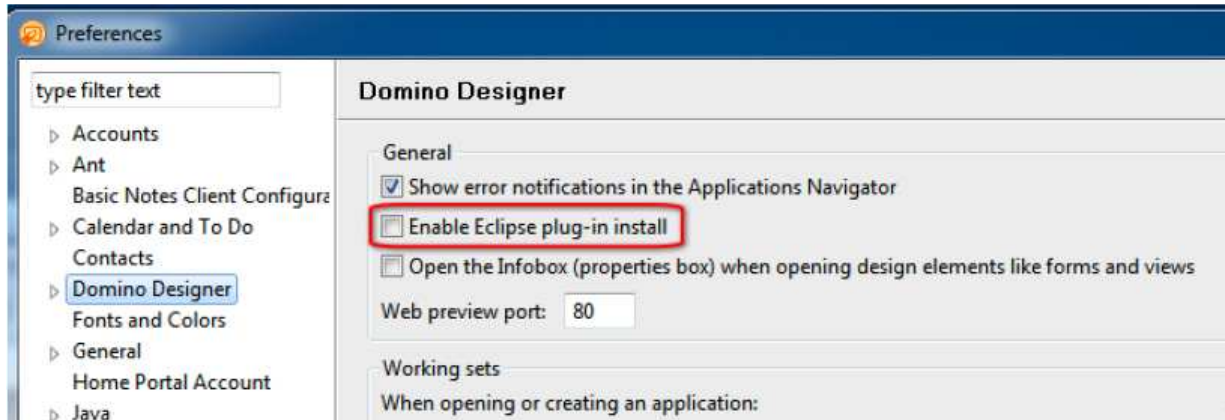
- XPages development requires a more structured approach to release control than classic Notes.
- Move towards a more modular approach to applications.
- More often multi-developers way of working.

Configure DDE for source control

In order to be able to use source control in DDE you need to install plugins. You will need to be able to install from an update site.

- Go to the 'File → Application → Install' menu.

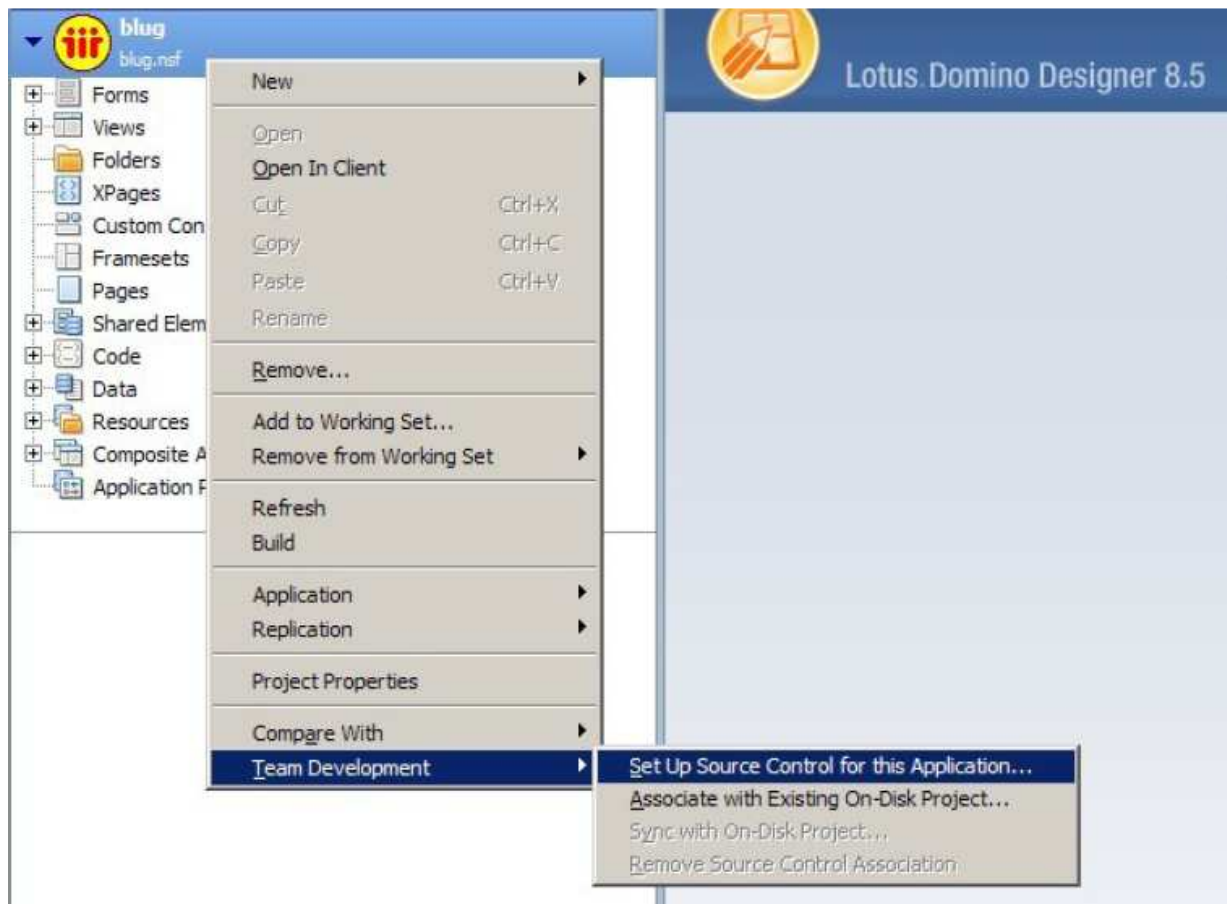
If you don't see this menu option you need to enable it in your DDE preferences.



- EGit for DDE based on current EGit for Eclipse 3.6 and higher.
- Subversion Update Site for DDE.
- Mercurial; choose 1.6.0 version of Mercurial Eclipse plugin and 1.4.3 version of Mercurial Binaries.

Enable source control for NSF

- In DDE select an application and select 'Team Development → Set Up Source Control for this Application...'



- o Select 'Next':

Set Up This Application for Source Control

Get Started with Source Control

To use source control with a Notes application, you must create an on-disk project. The on-disk project is a physical copy of the application (NSF).

1. Create an on-disk project. Domino Designer uses a virtual file system to represent a Notes application as a project in Eclipse. To make the application available to source control, a physical version of the virtual project must be created on the disk. This physical on-disk project also gives the source control system access to individual design elements.



The NSF and on-disk project are automatically kept in sync by default

The on-disk project syncs with the source control repository

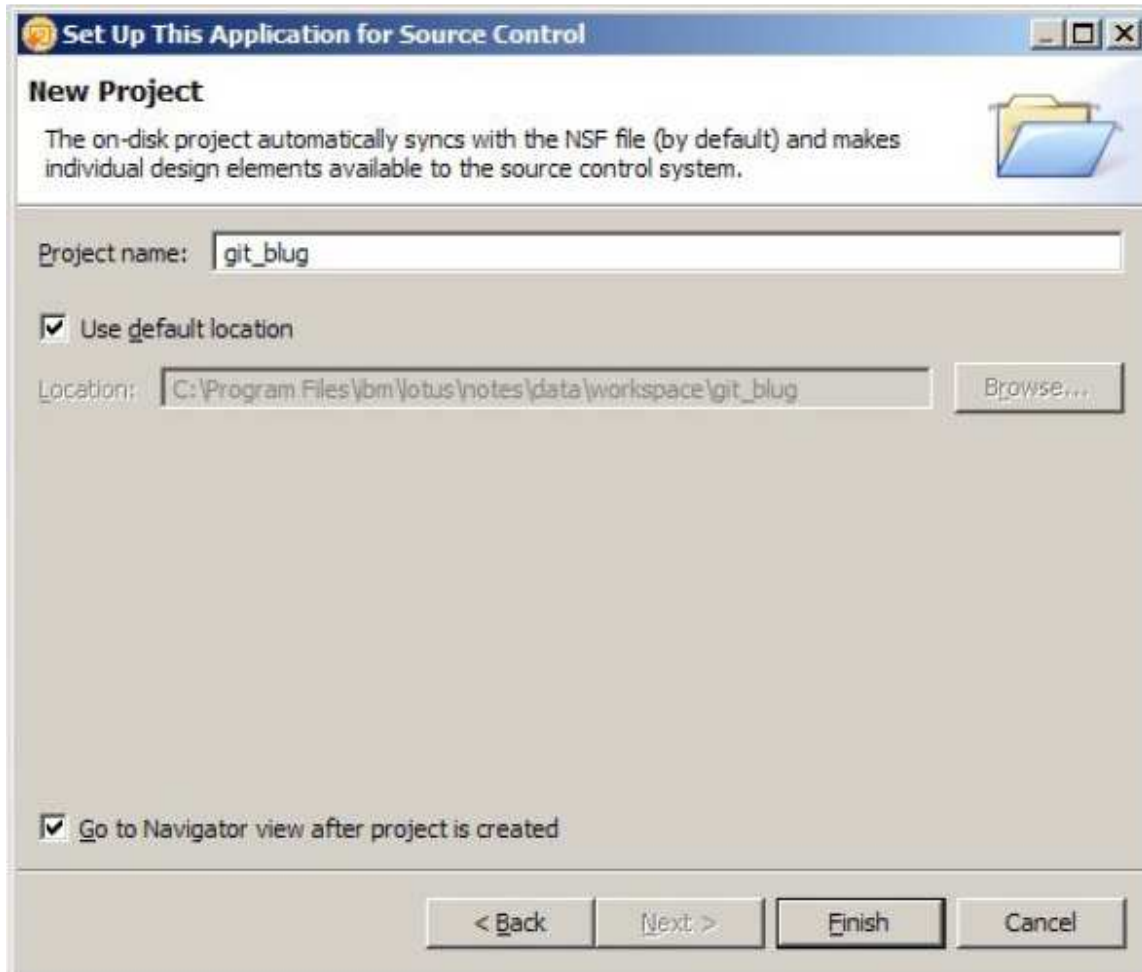
2. Use an Eclipse view, such as the Navigator or Package Explorer, to see the on-disk project and perform source control operations on it.

Click Next to create the on-disk project.

Skip this Helper panel in the future

< Back **Next >** Finish Cancel

- o Enter a name for the project:

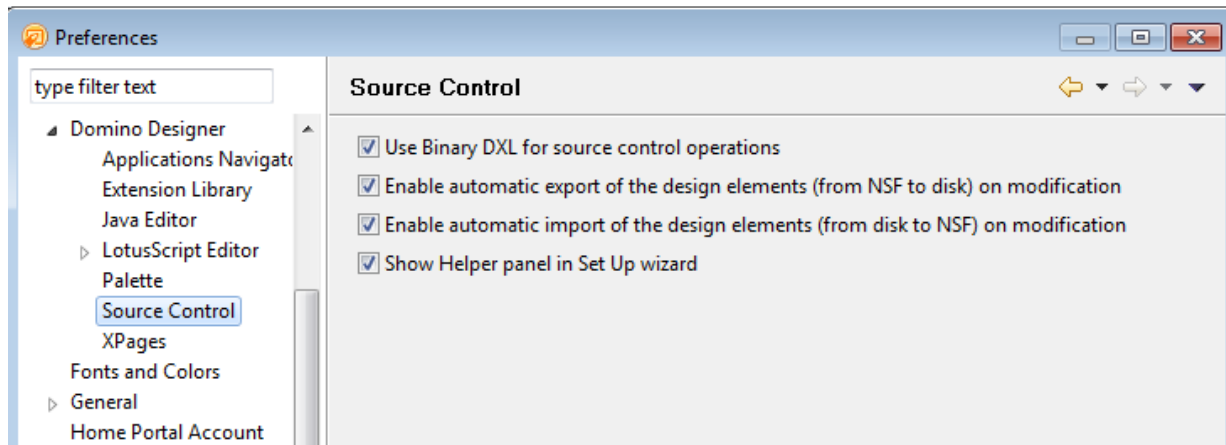


Note: Never, ever, ever create this to an existing folder. It deletes all content in that folder before exporting the NSF. Simply just accept the default option.

- Select 'Finish'. This creates an on-disk Eclipse project of the source code in the application (NSF).

Automatic import/export

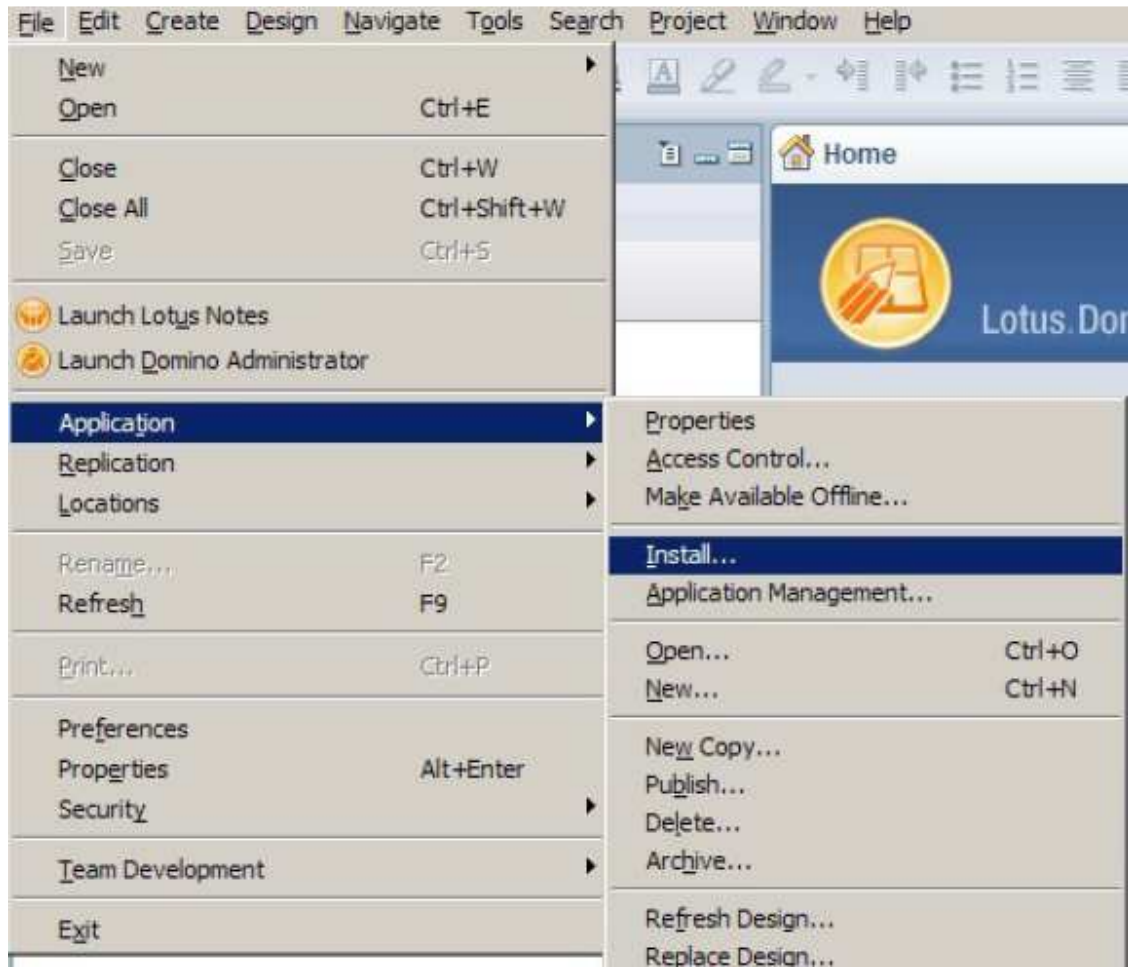
In Domino Designer you can disable automatic import / export to the on disk project, under 'Domino Designer → Source Control'. Here are two checkboxes for import and export.



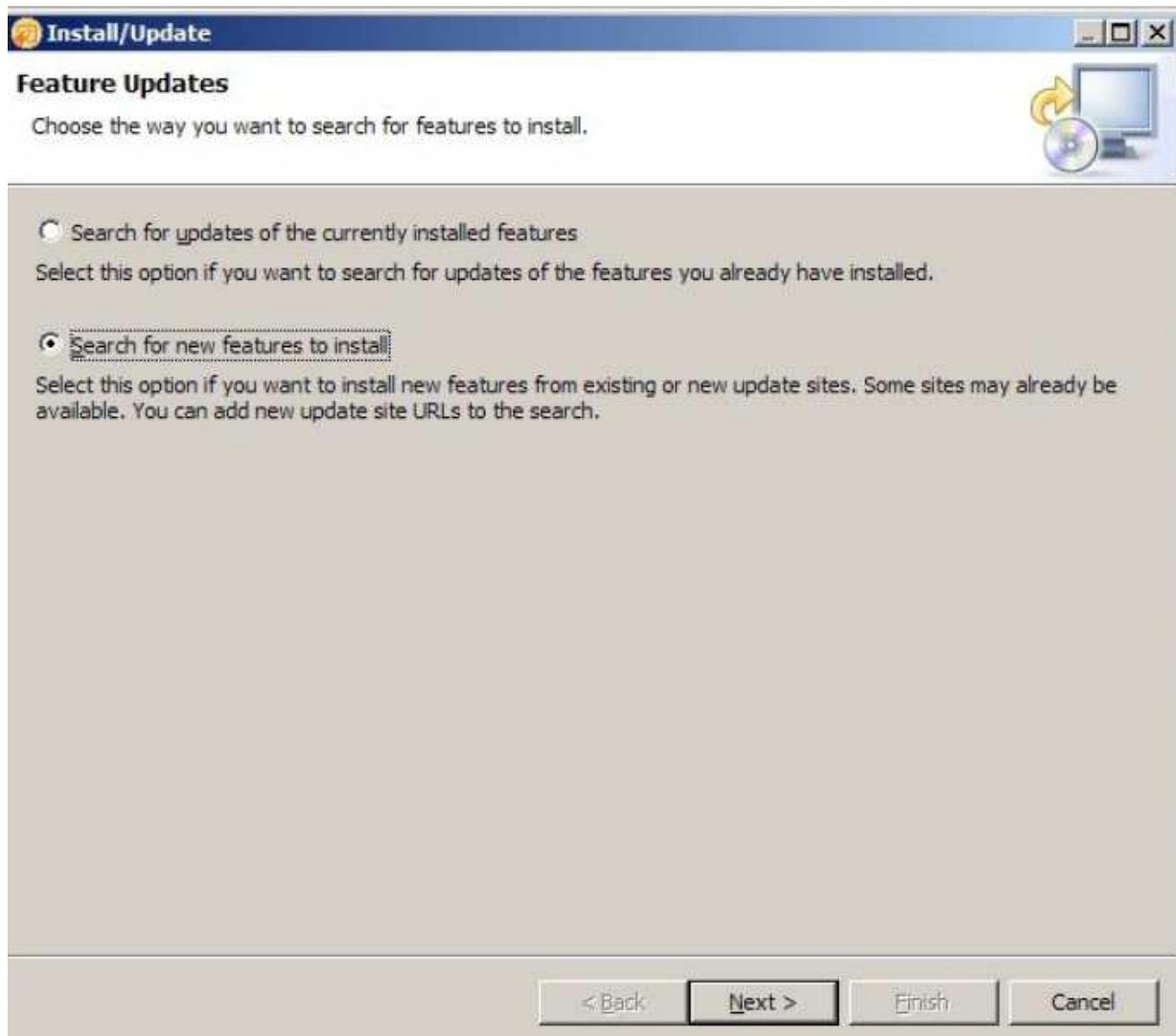
The best way to think of this is that the on disk project is effectively a local replica of the database. If those preferences are checked, then it's like after making any change to each design element, replicating to your local replica and waiting until the replication has completed before doing anything. It will have minimal impact if you're working on local NSFs only. But if you're working with server-based NSFs, you don't need it, so disable it. All you have to do is get into the habit of manually syncing, by using 'Team Development → Sync with On Disk Project...'.

Install EGit to DDE

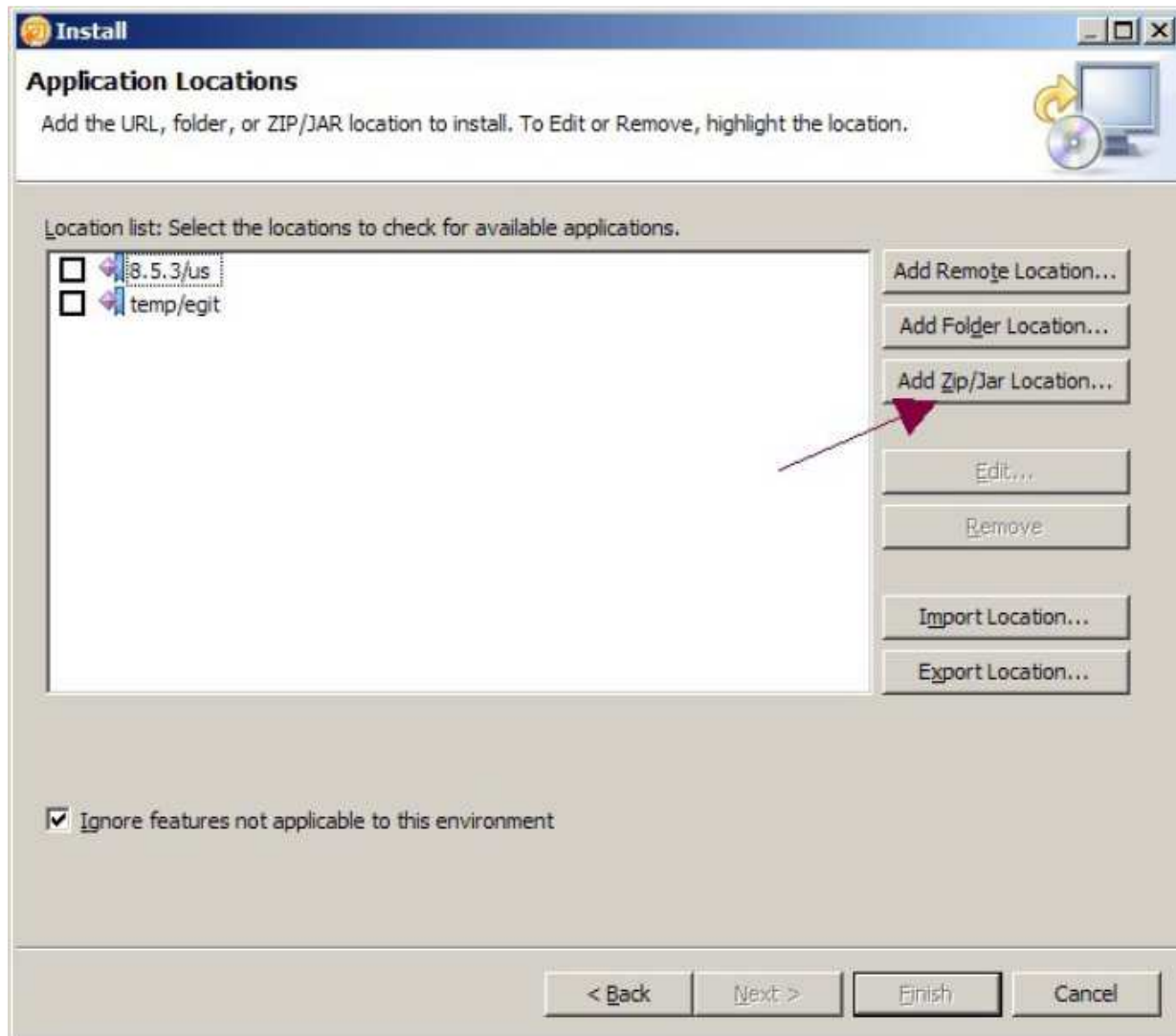
- Download the EGit for IBM Domino Designer plugin at <https://www.atlassian.com/software/sourcetree/overview>. Information about the plugin project you can read here: <http://www.openntf.org/internal/home.nsf/project.xsp?action=openDocument&name=EGit%20for%20IBM%20Domino%20Designer>.
- In DDE select 'File → Application → Install...':



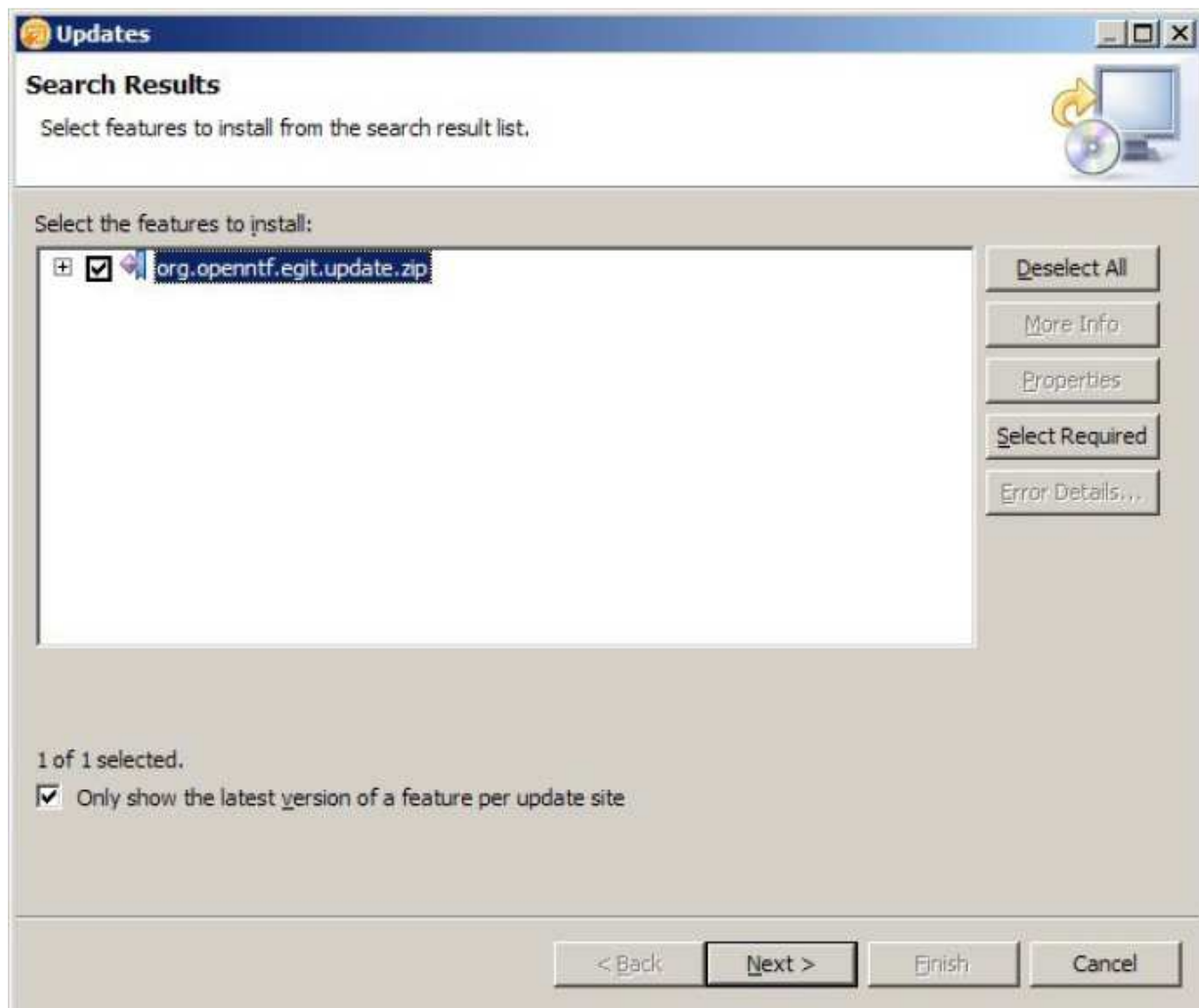
- Choose 'Search for new features to install':



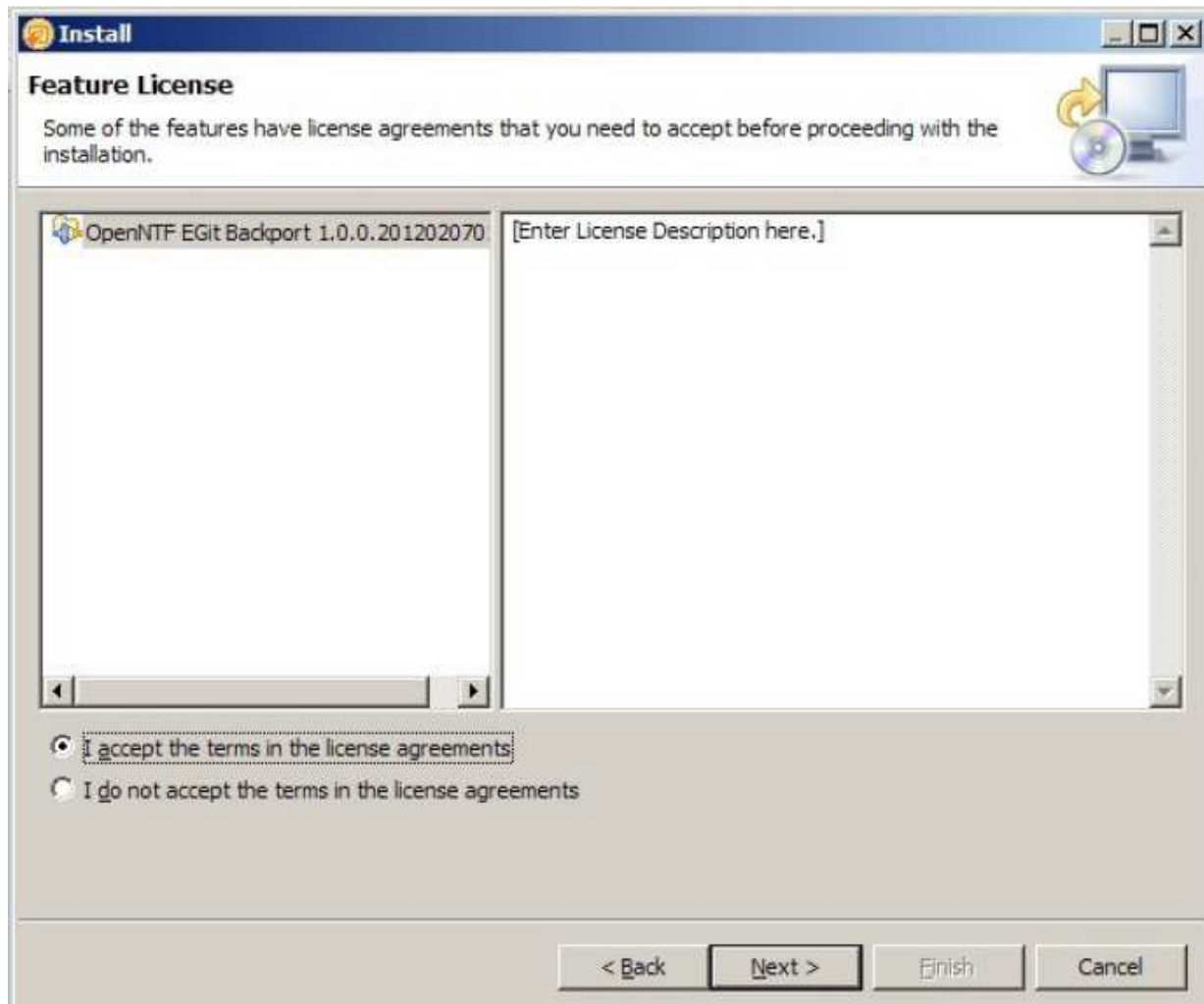
- Point it to the downloaded zip file:



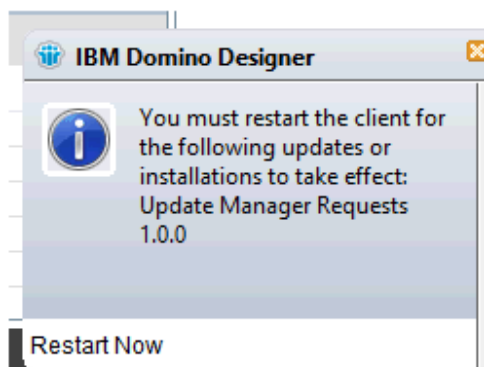
- o Check 'org.openntf.egit.update.zip' and press 'Next':

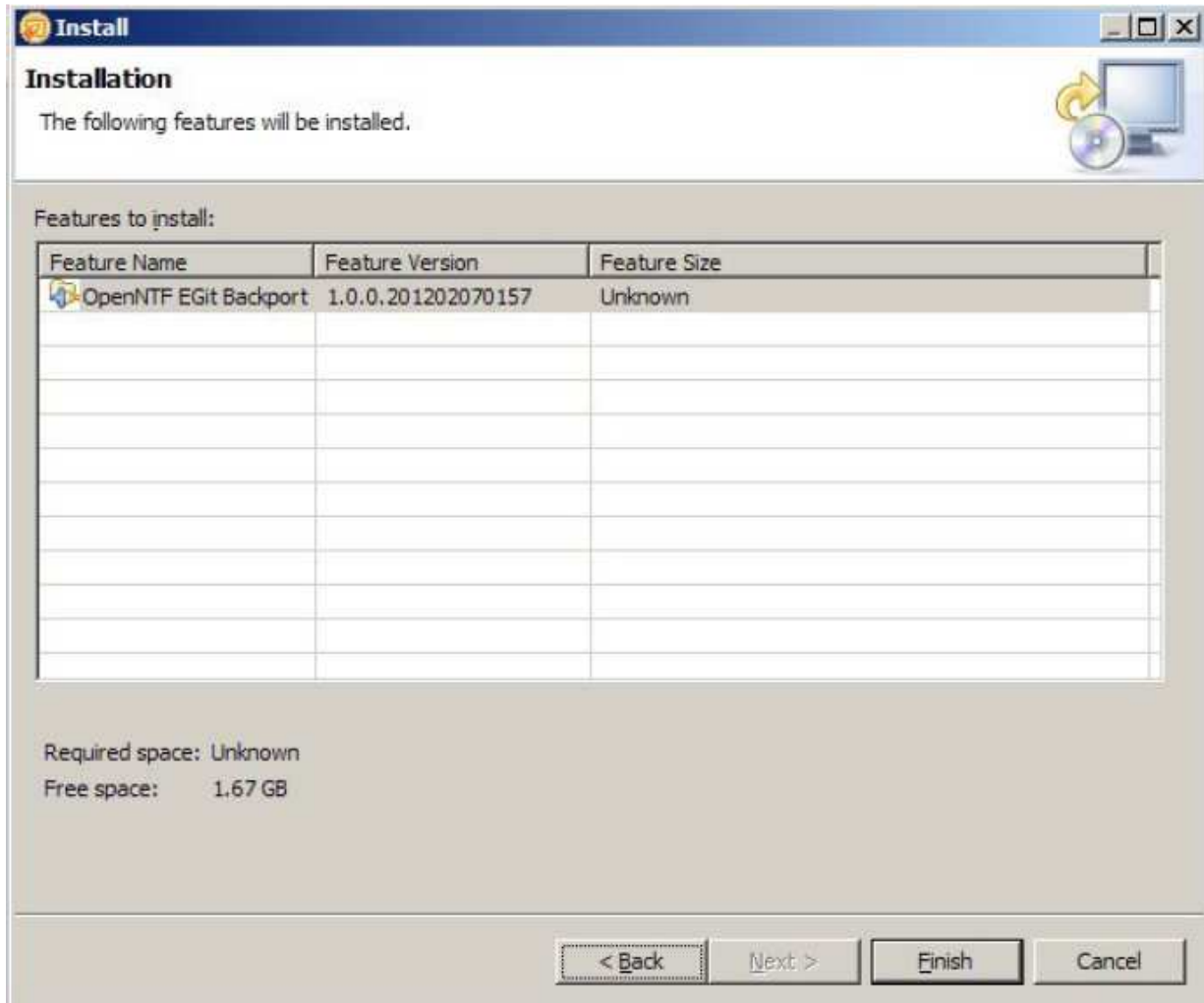


- o Accept the term in the license agreements:



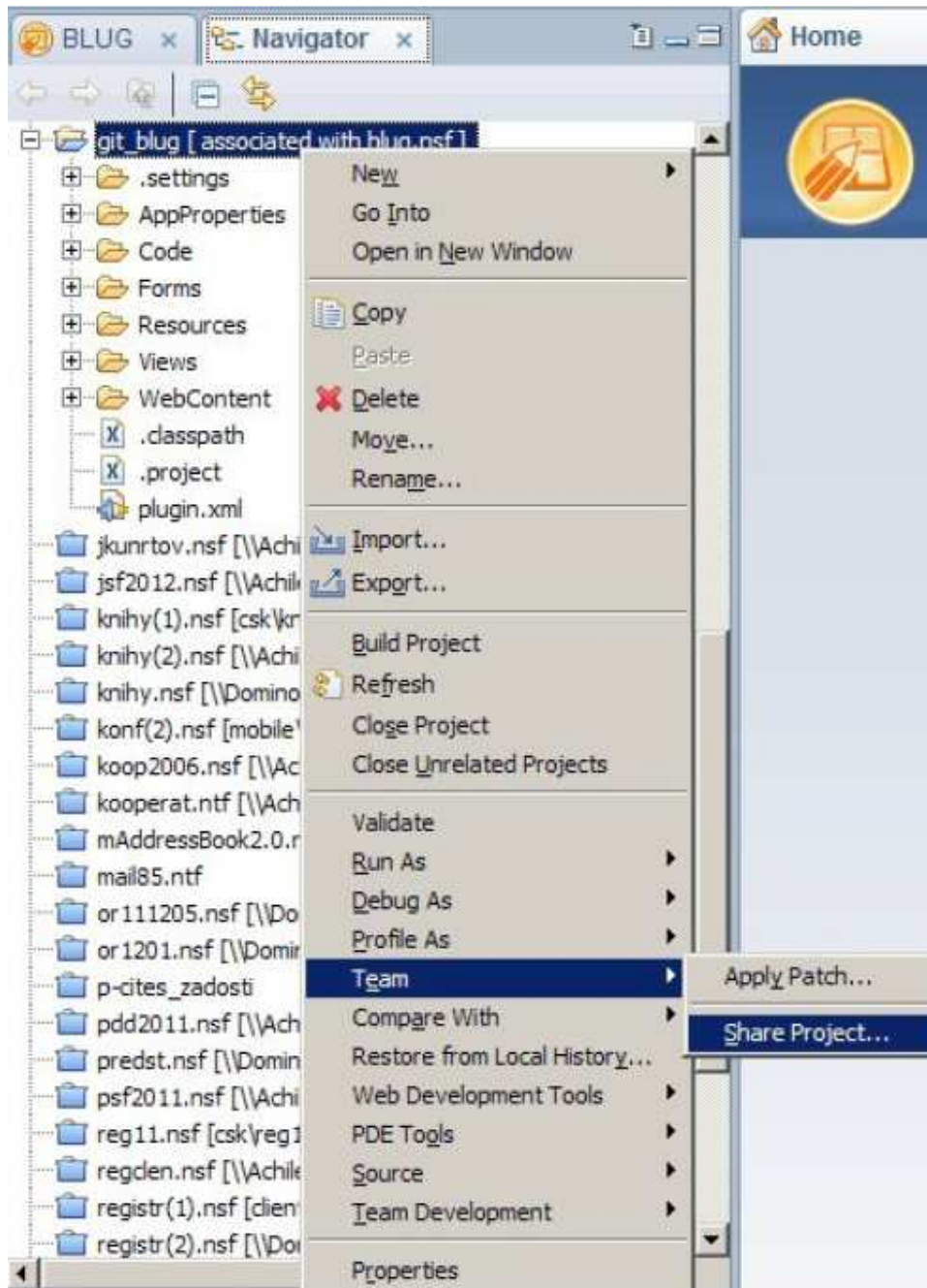
- Press 'Finish'. Keep accepting (there is one more dialog asking for permission to install plugin) and then restart Designer.



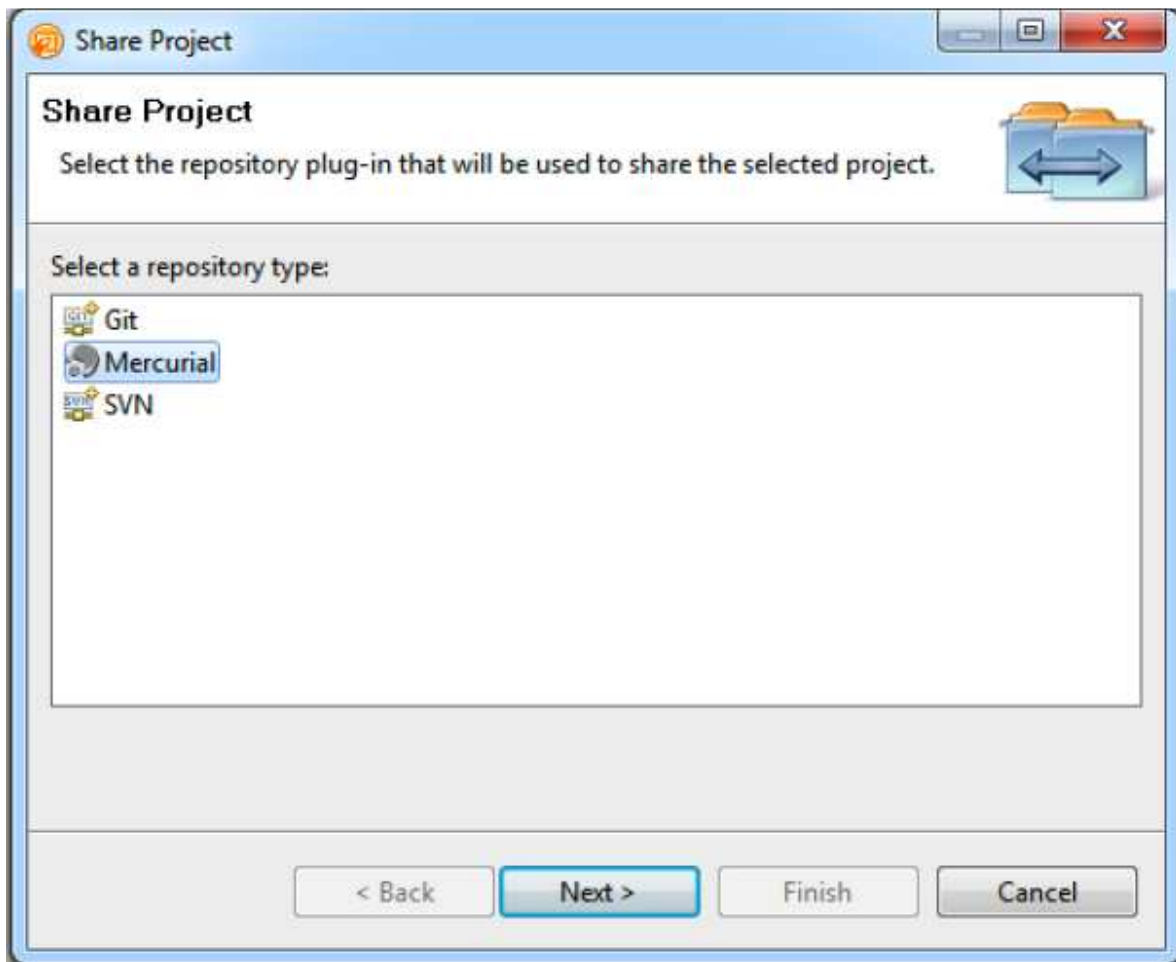


Set up Git for the on-disk project

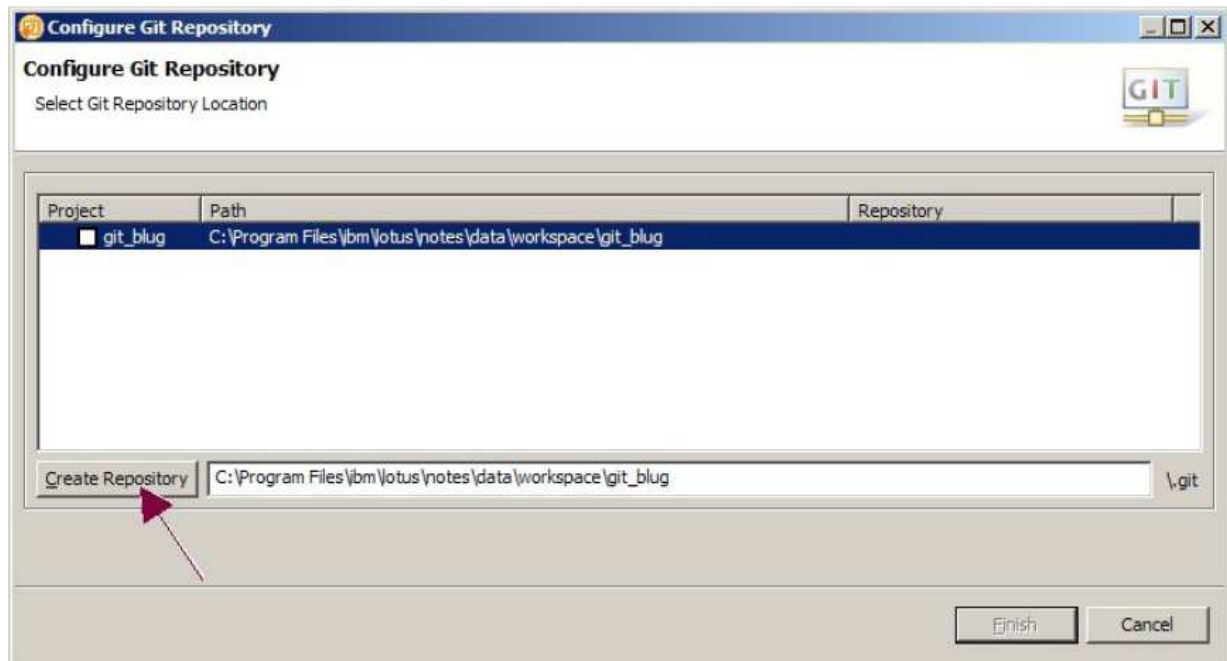
- o Select the project in the Navigator Eclipse view perspective.
- o Choose 'Team → Share Project...':



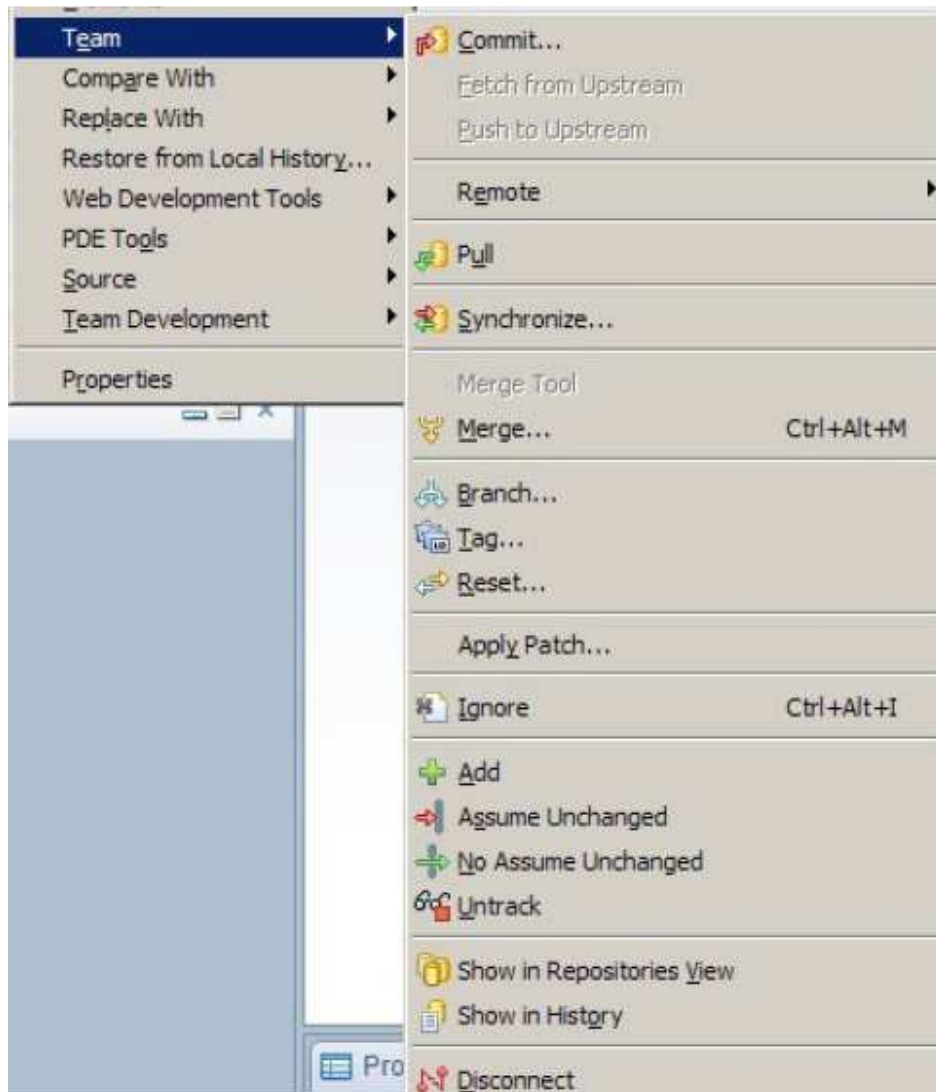
- o Select Git as repository:



- o Configure a repository for GIT:



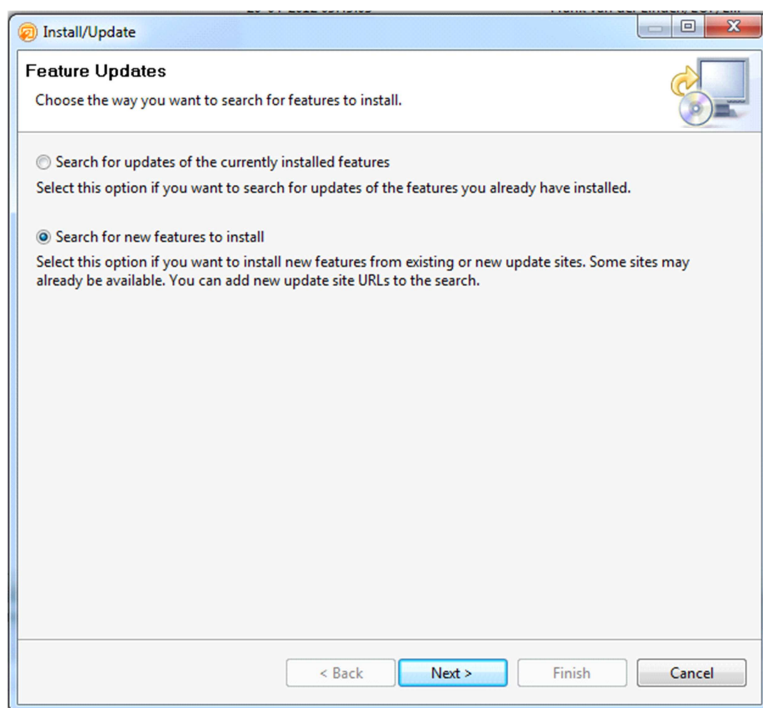
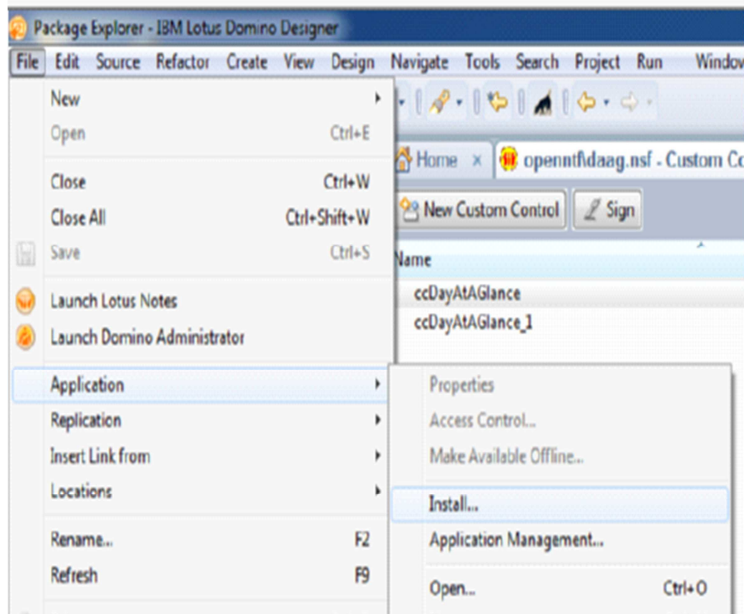
- o Start using Git directly from DDE:



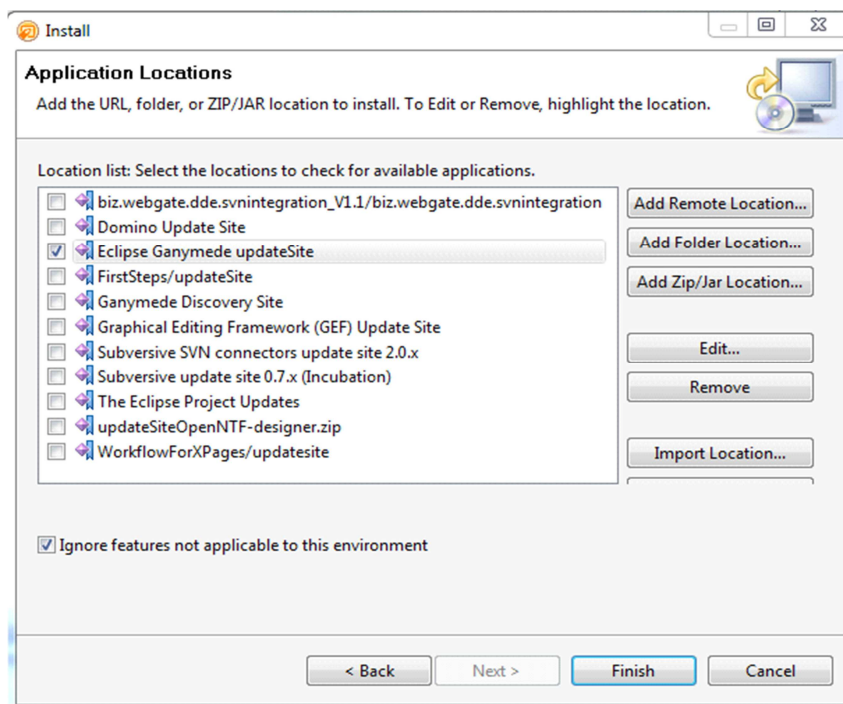
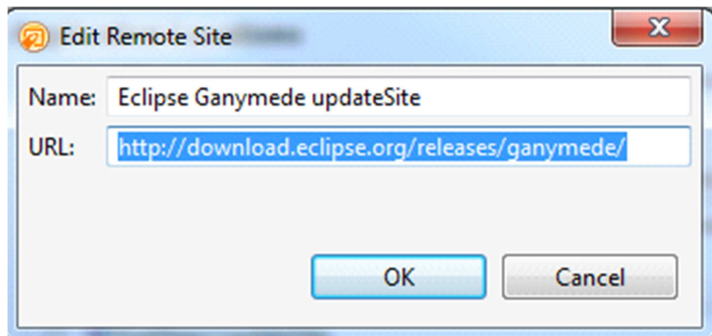
- Alternatively you can use command line in Git Bash (<http://jinoch.cz/using-git-domino-designer-hardcore-way>).

Install CVS

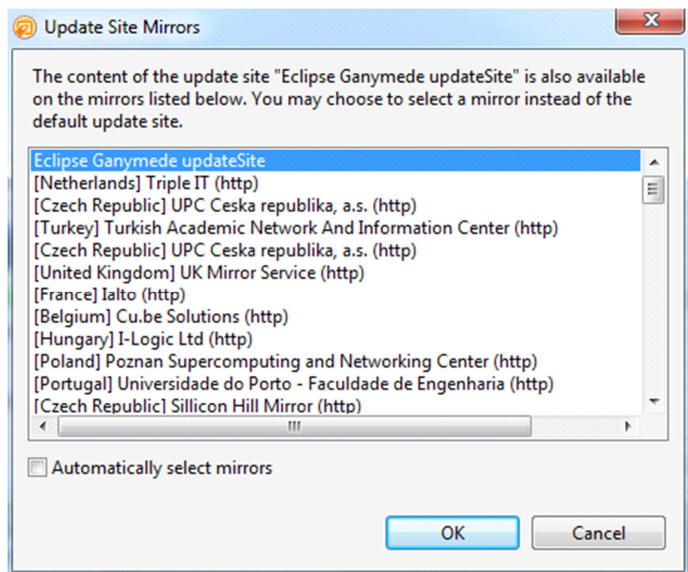
- In DDE select 'File → Application → Install...':



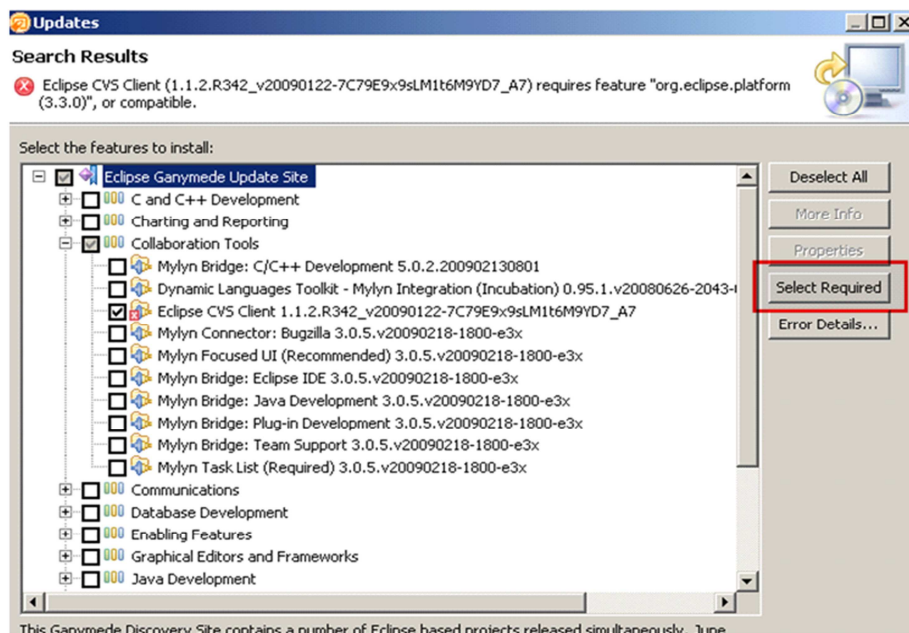
- Specify the Ganymede update site.
- Add as Remote location:
 - URL is <http://download.eclipse.org/releases/ganymede/>



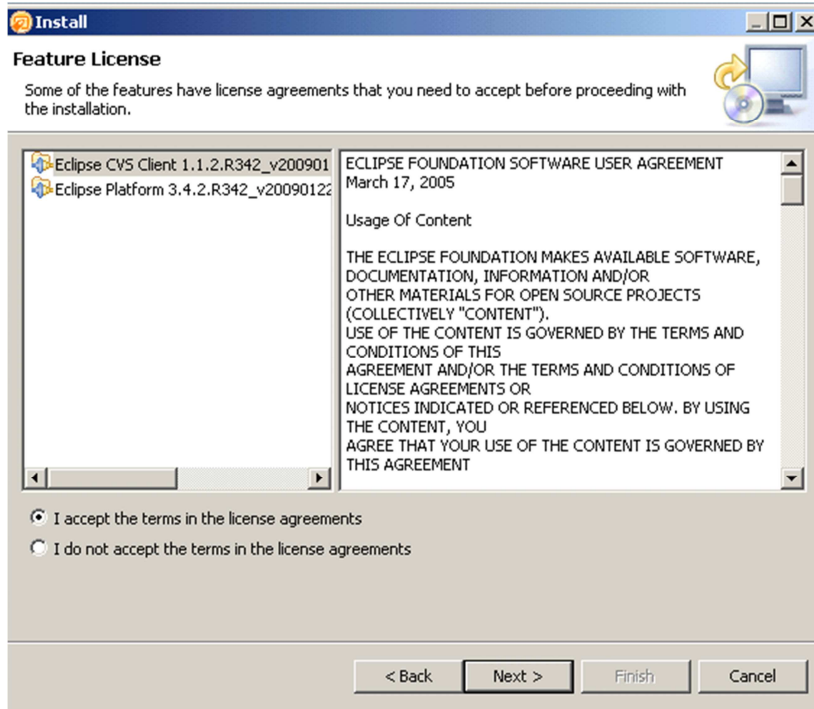
- Click 'Finish' to start searching for software to install.
- Select the required mirror site or leave it as is:



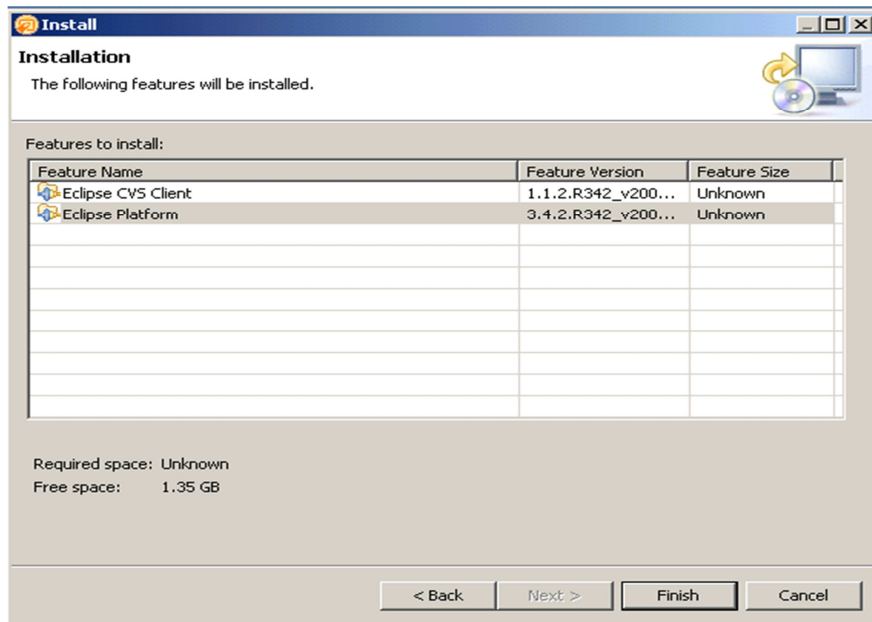
- Click 'OK'.
- Select under Collaboration tools the CVS connectors.
- To get all the depending connectors, click the button 'Select Required'.



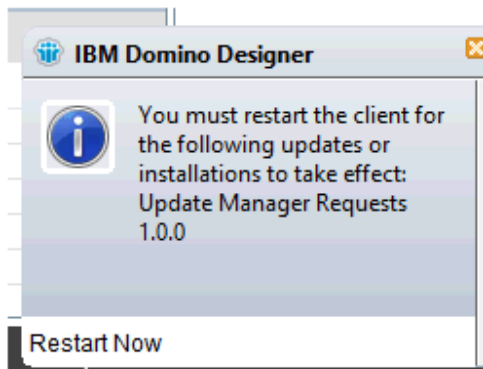
- Click 'Next'.
- You need to accept the license of the connectors:



- Click 'Next'.
- You get an overview of all the plugins to be installed:

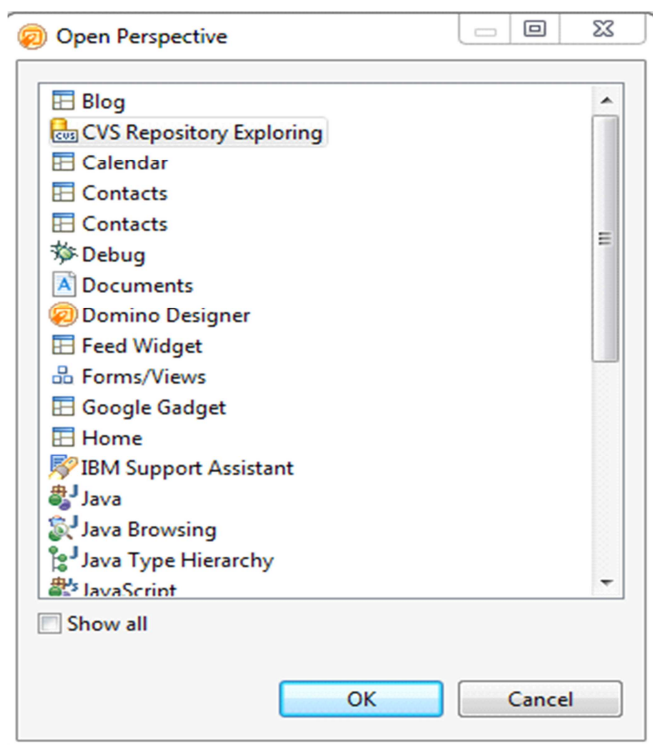


- Click 'Finish'.
- The connectors are being installed, this can take some time.
- Restart Designer.



Set up CVS

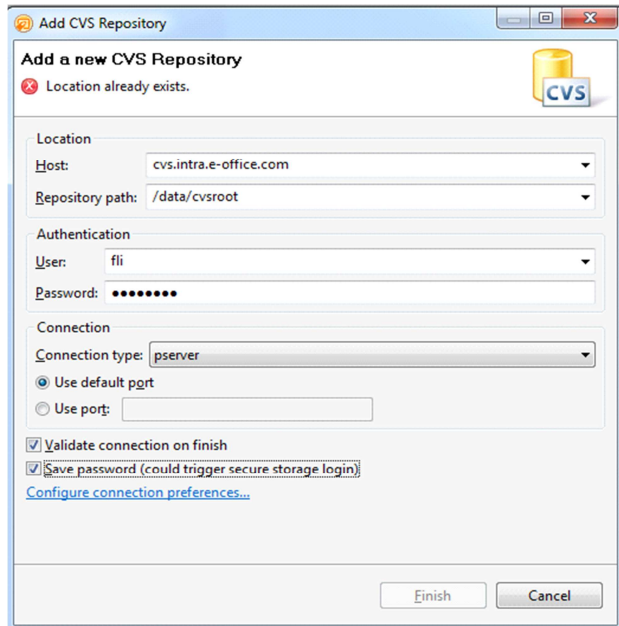
- o In DDE open the CVS Repository Exploring perspective. The perspective can be found via 'Window → Open Perspective → Other':



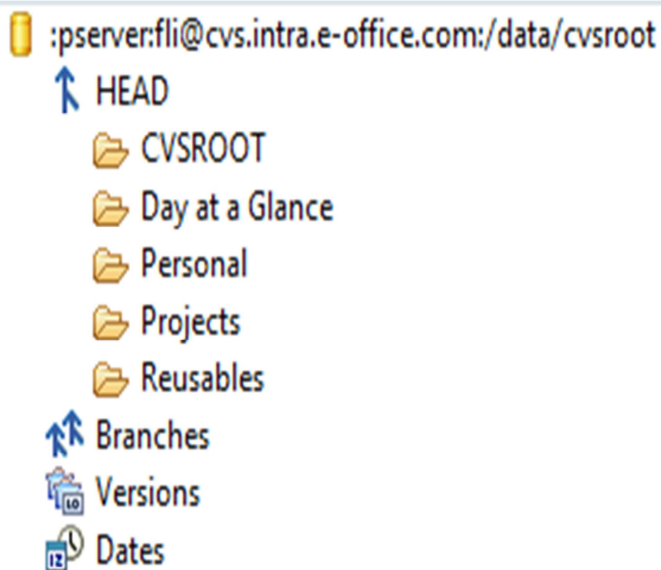
- o Click the CVS icon on top of perspective:



- o In the popup add the host and repository path of source control (CVS):

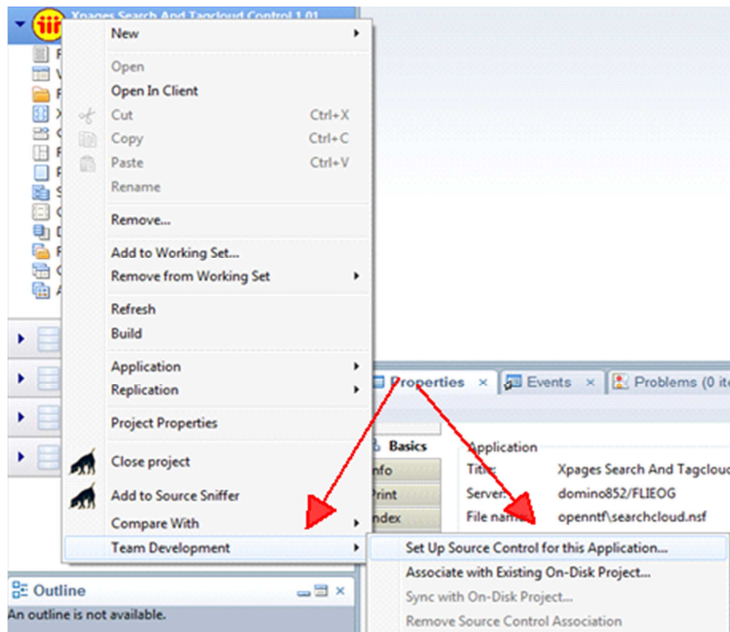


- If Authentication is required, provide User and Password.
- Click 'Finish'.
- If the connection is correct, the root of the source control system will be displayed:

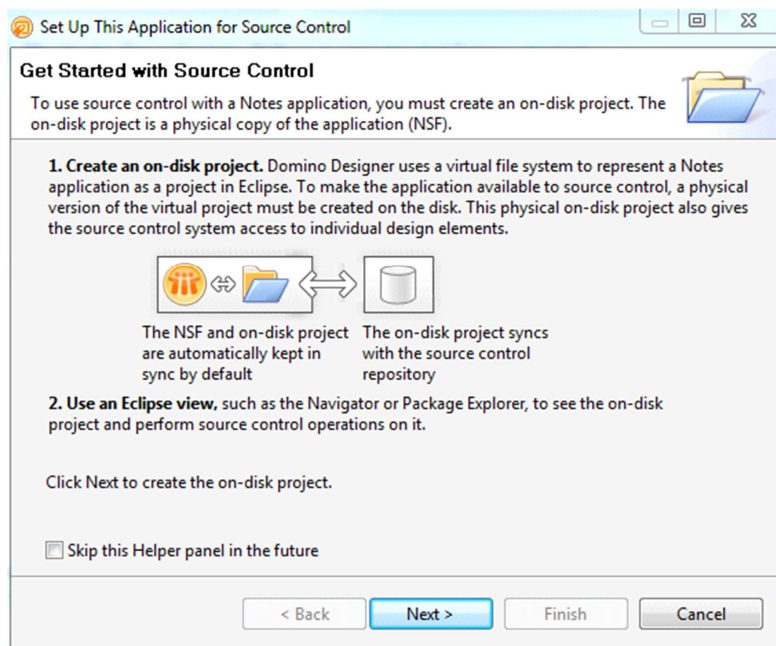


Set up CVS for the on-disk project

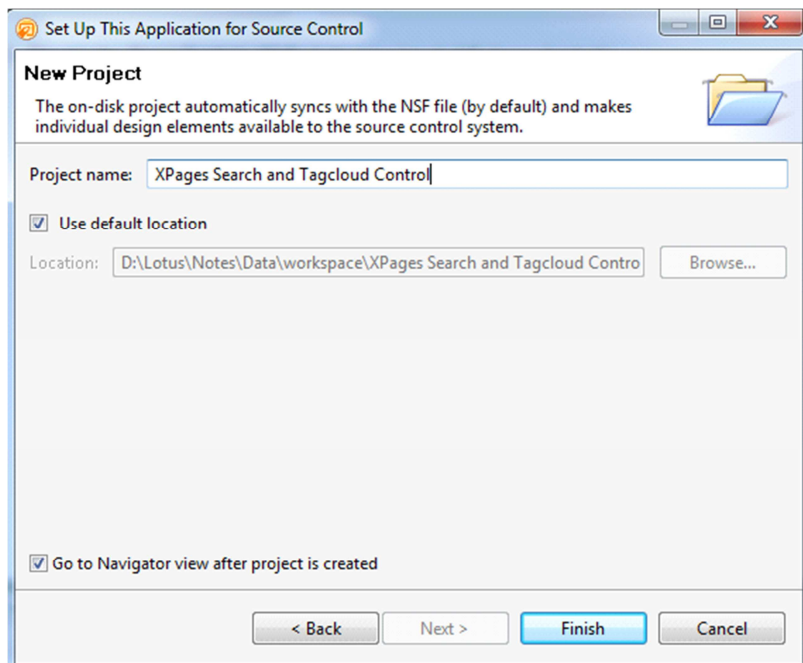
- Create an on-disk project.
 - Right click on the project name in Domino Designer Perspective.
 - Select 'Team Development → Set Up Source Control for this Application':



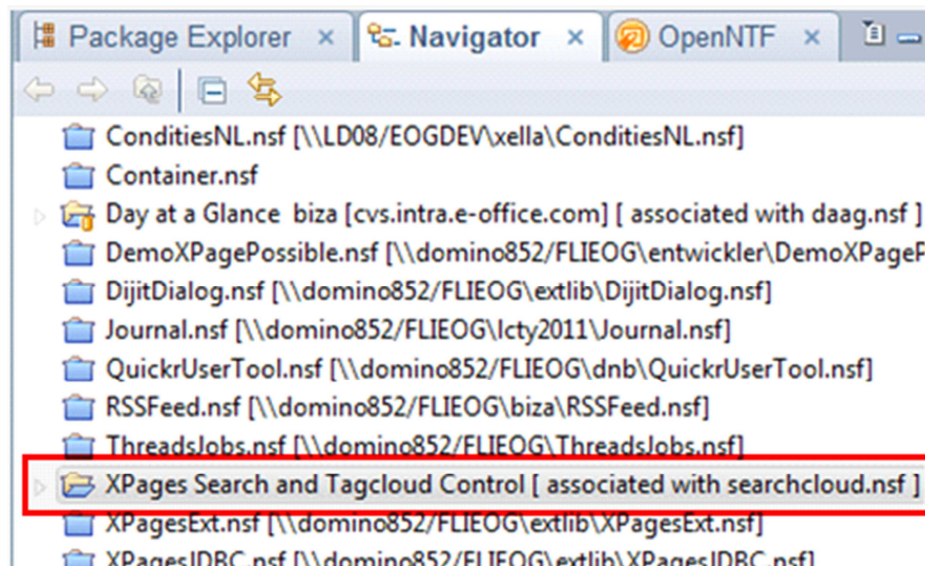
- You will get a Helper panel, with some information about the setup.
- If you don't want this in the future, tick the checkbox at the bottom:



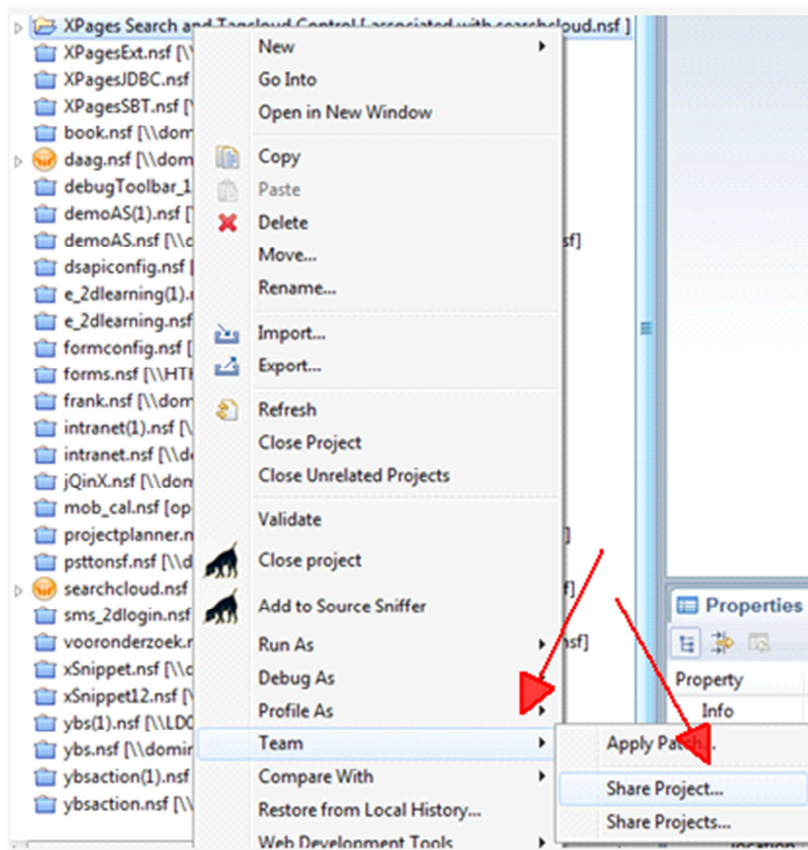
- Fill in the project name:



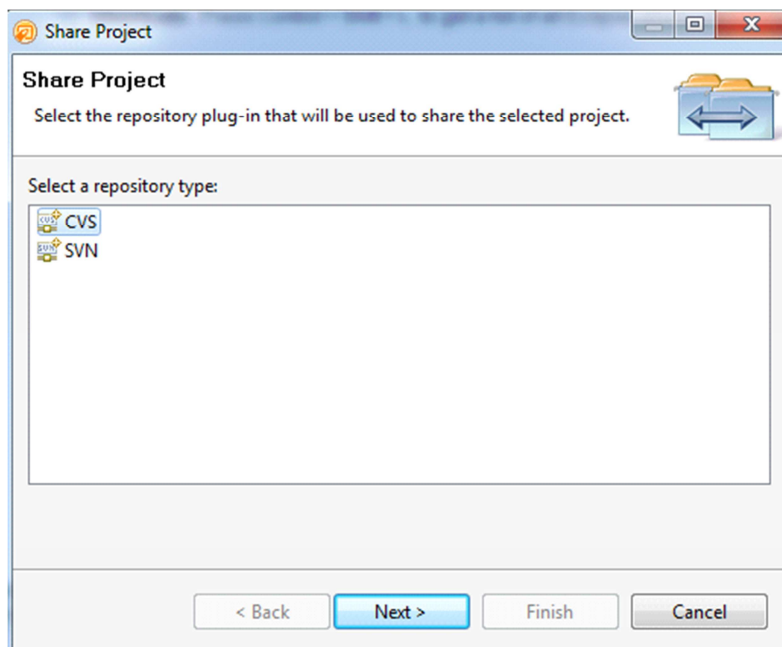
- Use the default location in the data\workspace directory or specify a custom location to store the on-disk project.
- Open the Navigator view via 'Window → Show Eclipse Views'.
- There is the new on-disk project:



- Next share the on-disk project with source control.
- Right mouse click and select 'Team → Share Project':

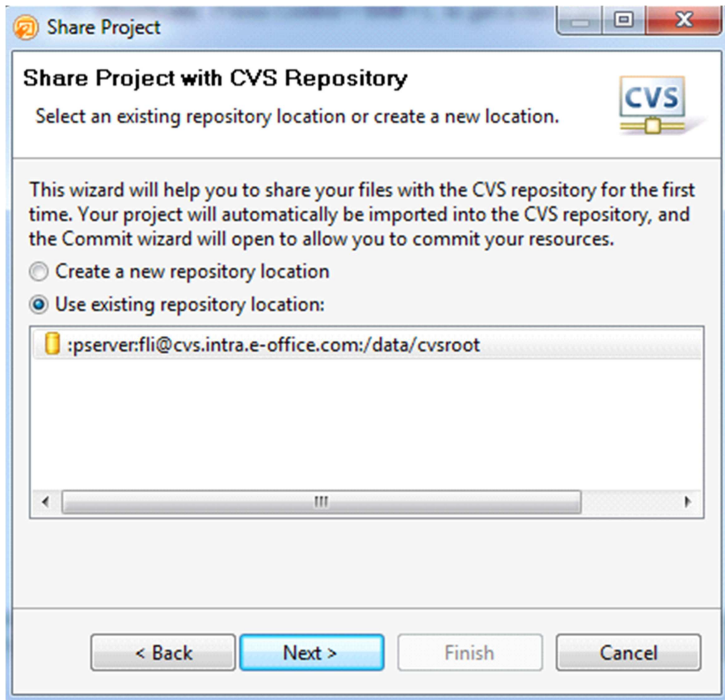


- Select the Repository type, select 'CVS':

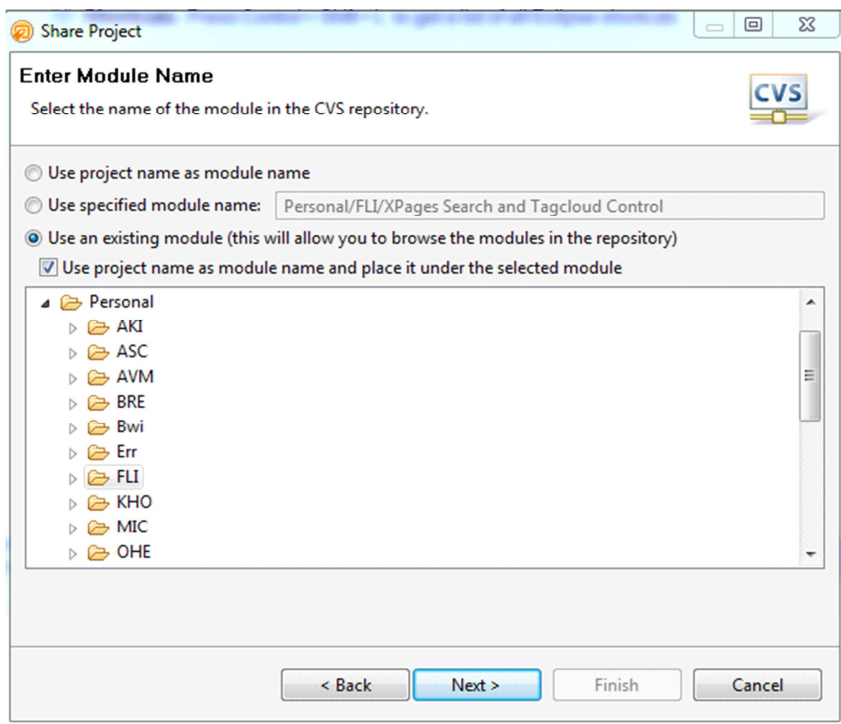


- Click 'Next'.

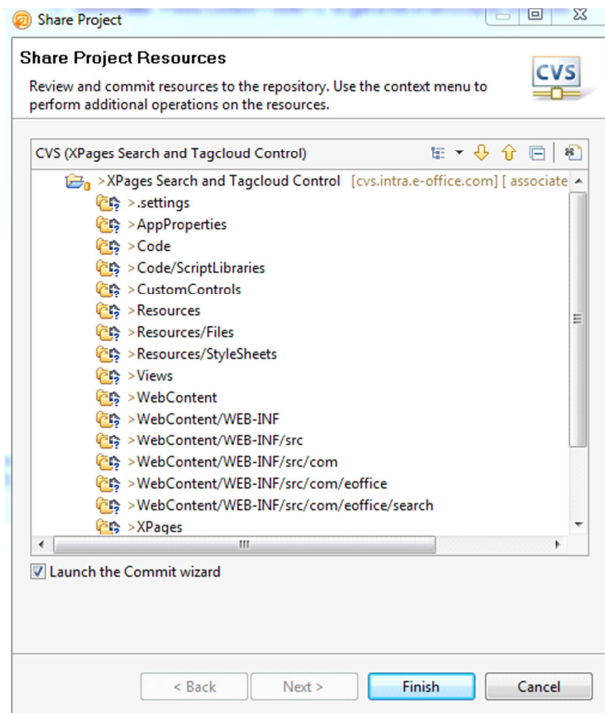
- Use the newly created repository location or create a new one:



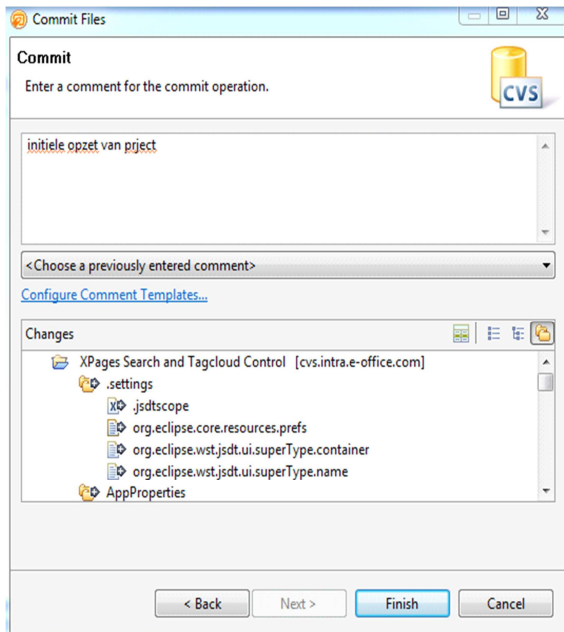
- Click 'Next'.
- Select the location on the source control server:



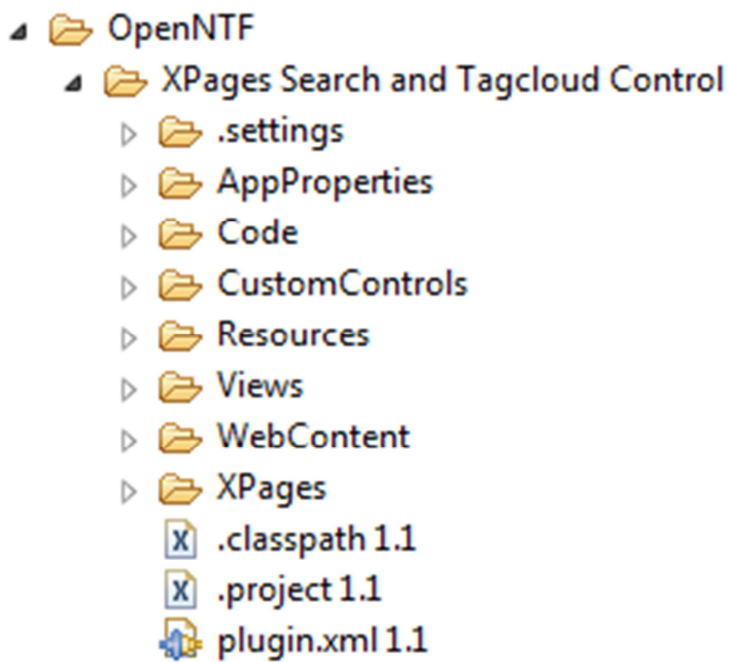
- Keep the checkbox 'Use project name as module name...' selected.
- Click 'Next'.
- An overview of all the design elements that will be committed is presented:



- Leave the 'Launch the Commit wizard' checkbox selected.
- Click 'Next'.
- Commit the files.
- Enter comment for the Commit operation.
 - This should be a meaning full description of this commit operation.



- Click 'Finish' and the commit operation will start directly.
- In the CVS Repository exploring perspective you can see if everything went well:



Install Mercurial in DDE

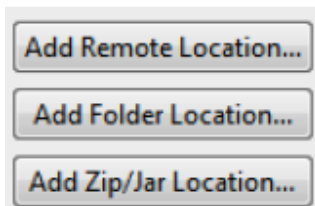
Preparation

- Visit site: <http://mercurial.selenic.com/downloads/>
 - Use version 1.6.0 MercurialEclipse.

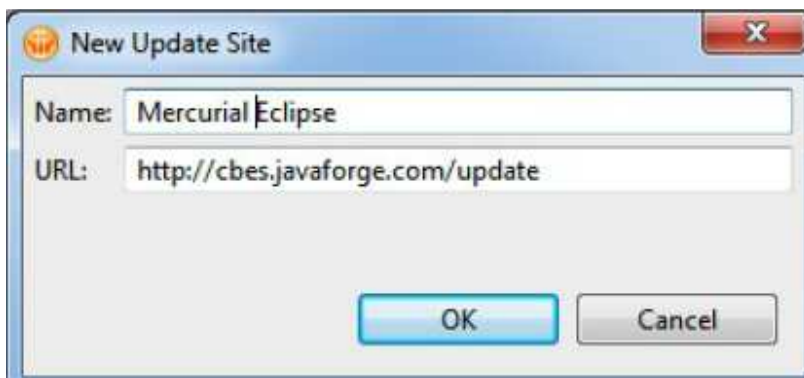
- Use version 1.4.3 Mercurial Binaries.

Installation

- In DDE select the option to install new features:



- Then add a new update site:



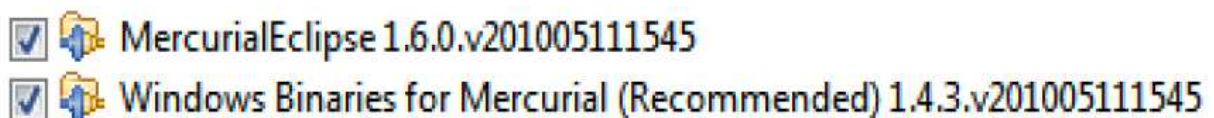
- Complete the details for your selected SCM package.
- Click 'Finish' to start searching for software to install.

Mercurial Eclipse requires an older version

- De-select the following option:

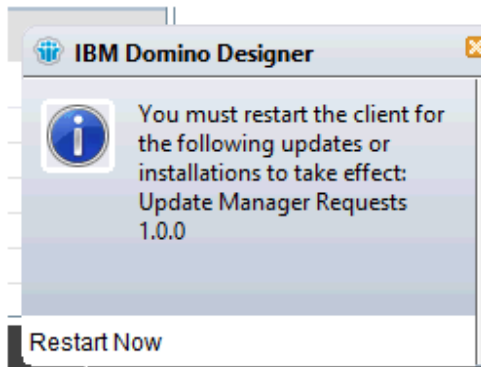


- The older version can now be selected:

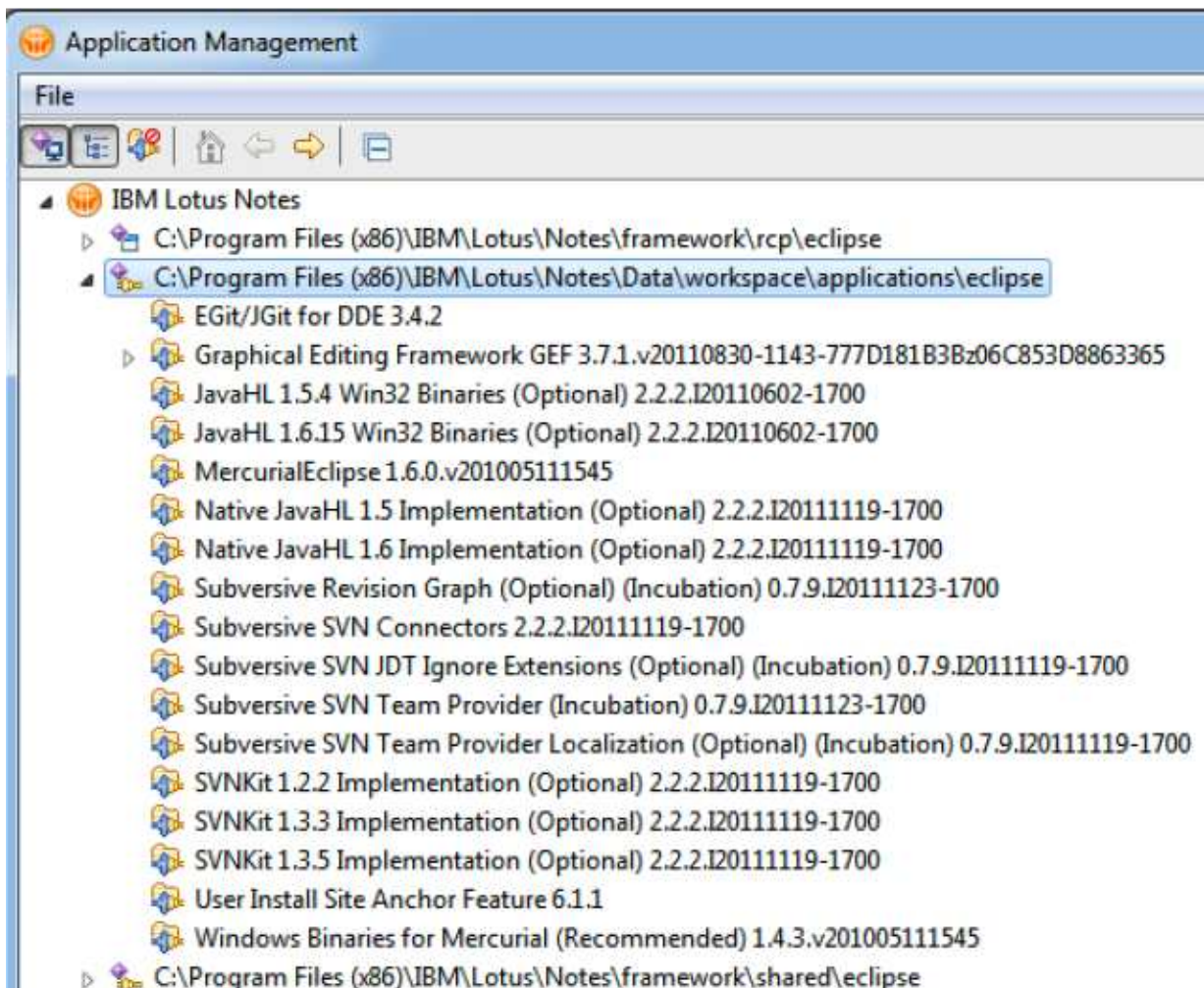


- Once selected click 'Next' and follow the prompts to install the update site.

- o Restart Designer client when prompted to activate the new plugins.

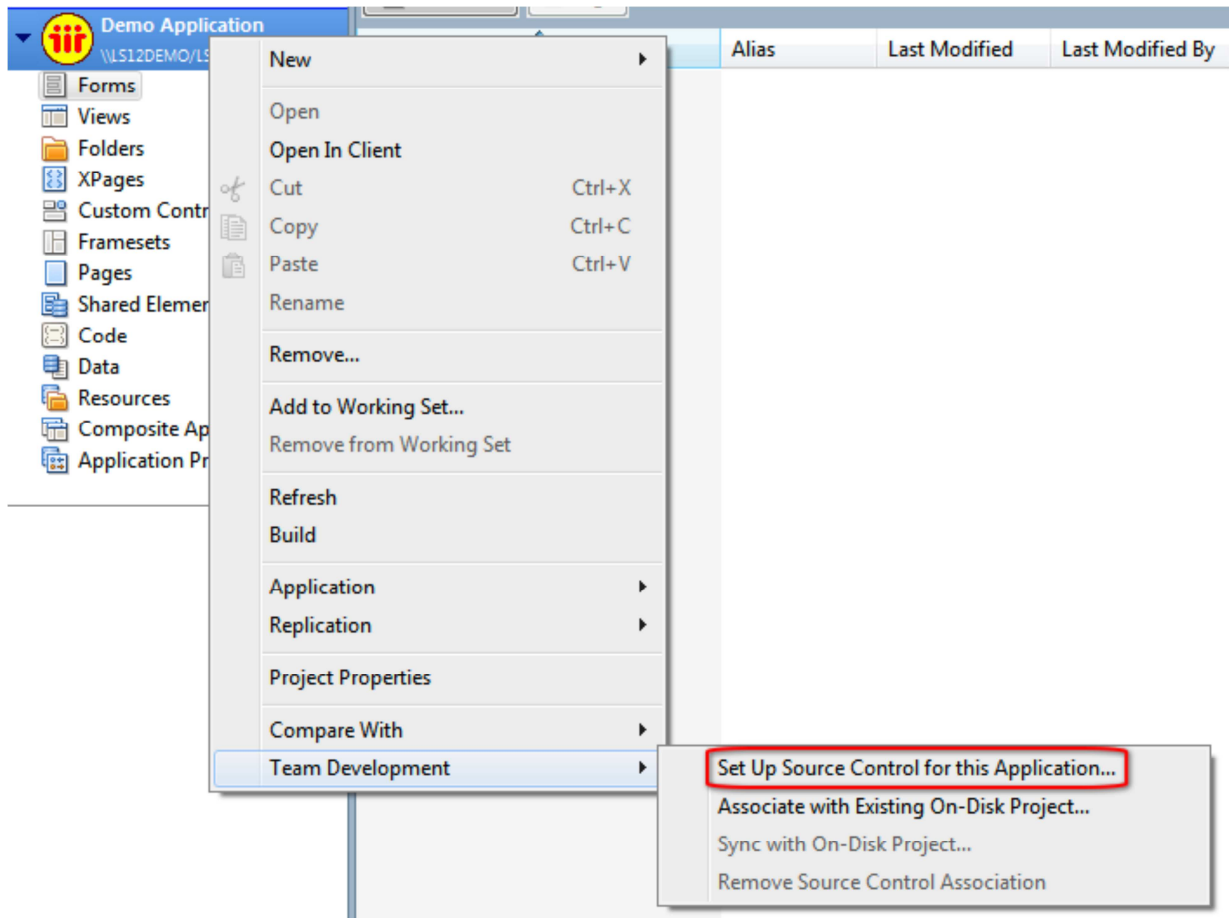


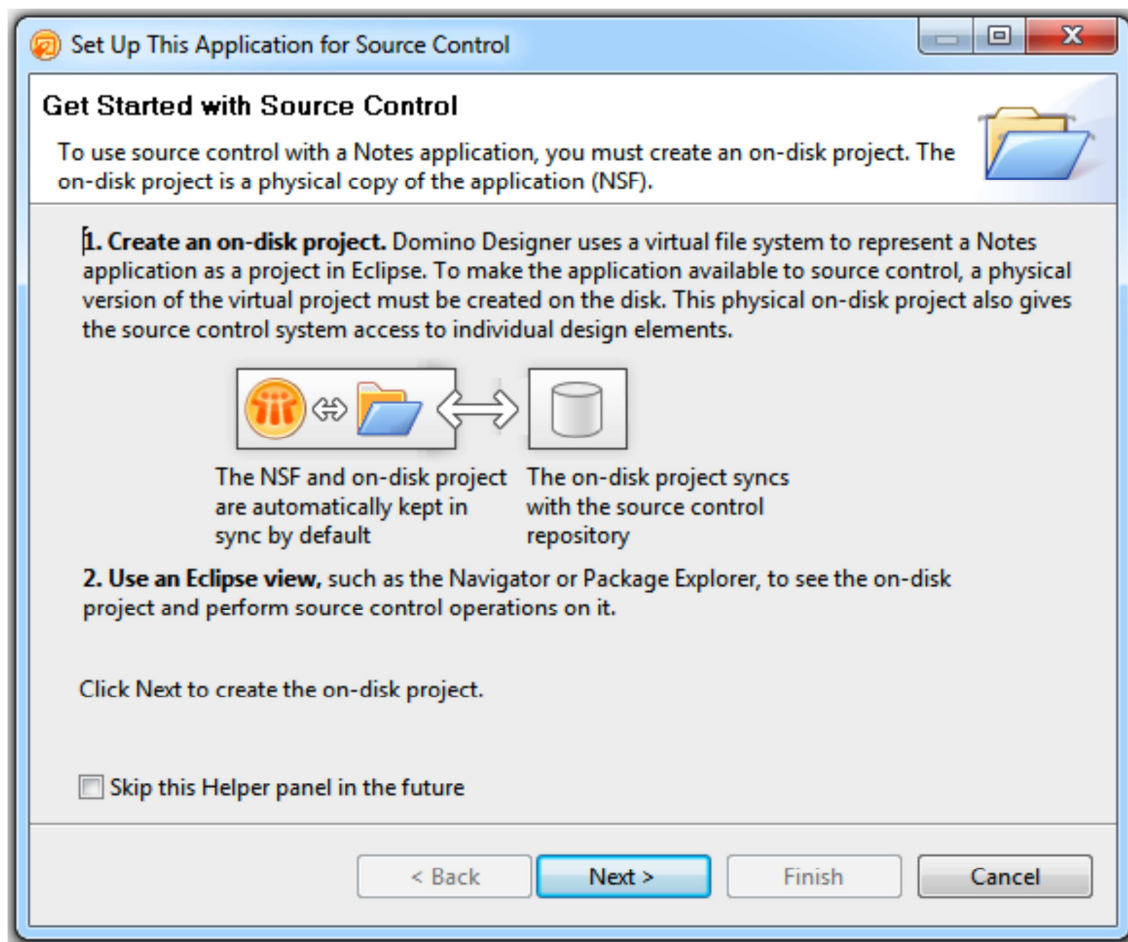
- o You can verify that the plugins are installed via 'File → Application → Application Management...':

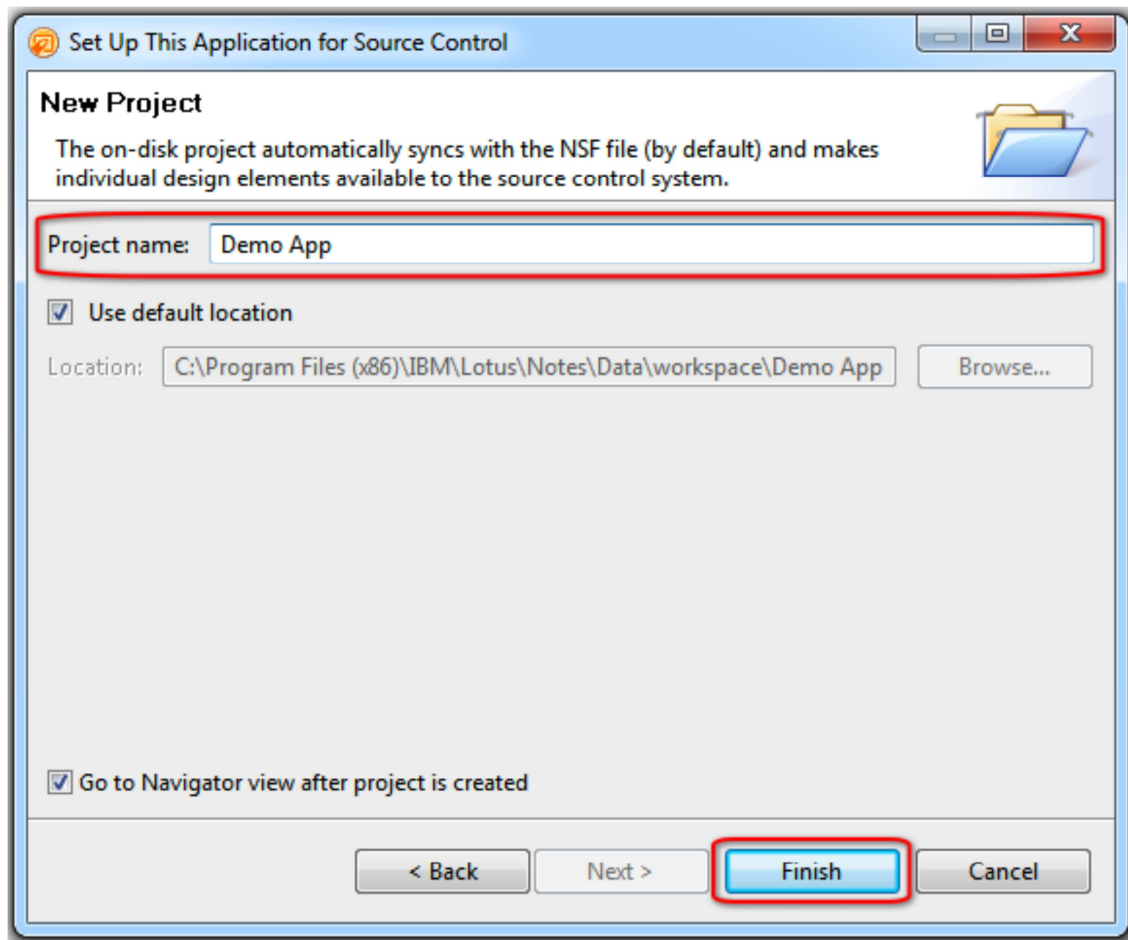


Set up Mercurial for on-disk project

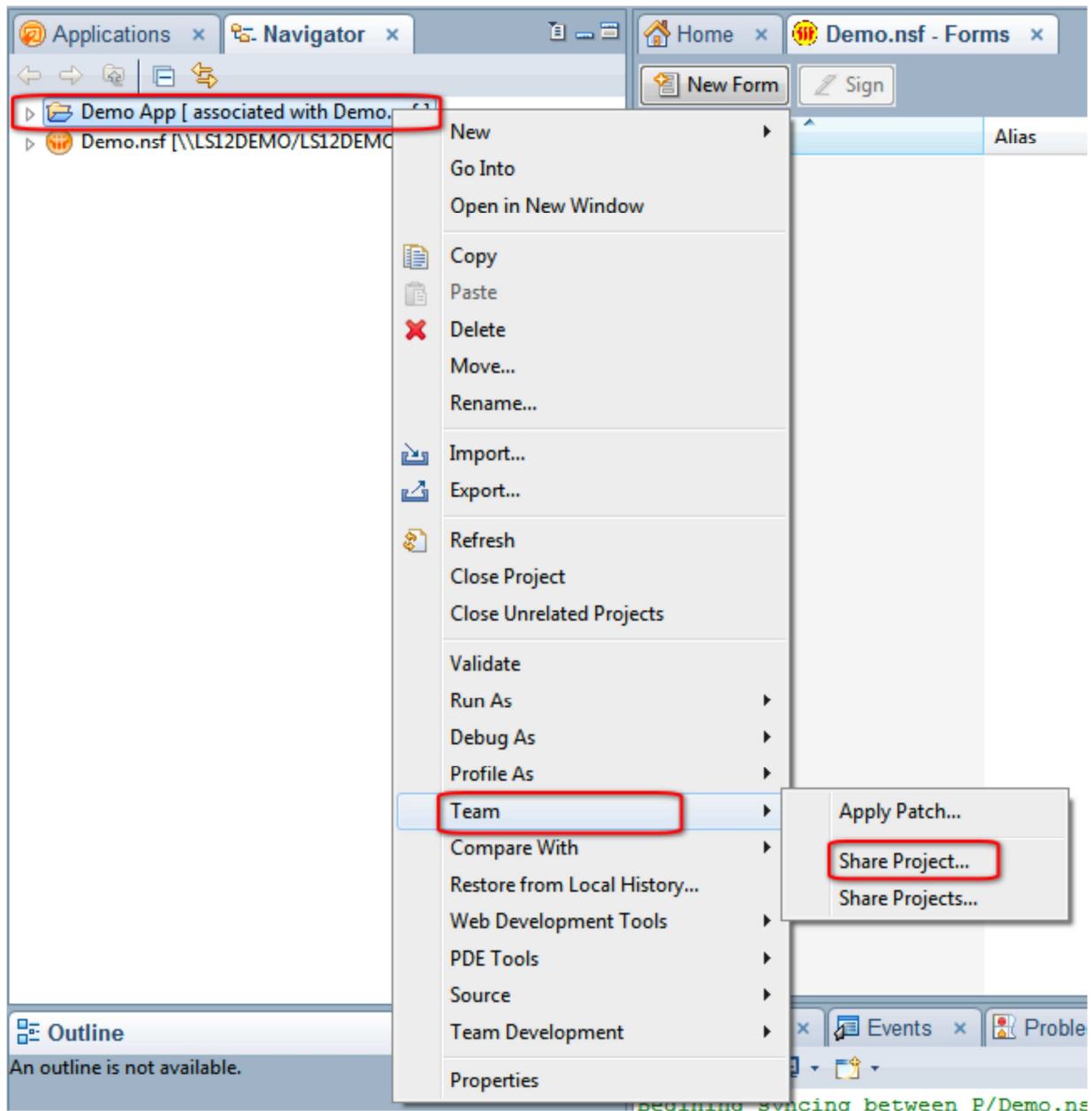
- o Select the project in the Navigator Eclipse view perspective.

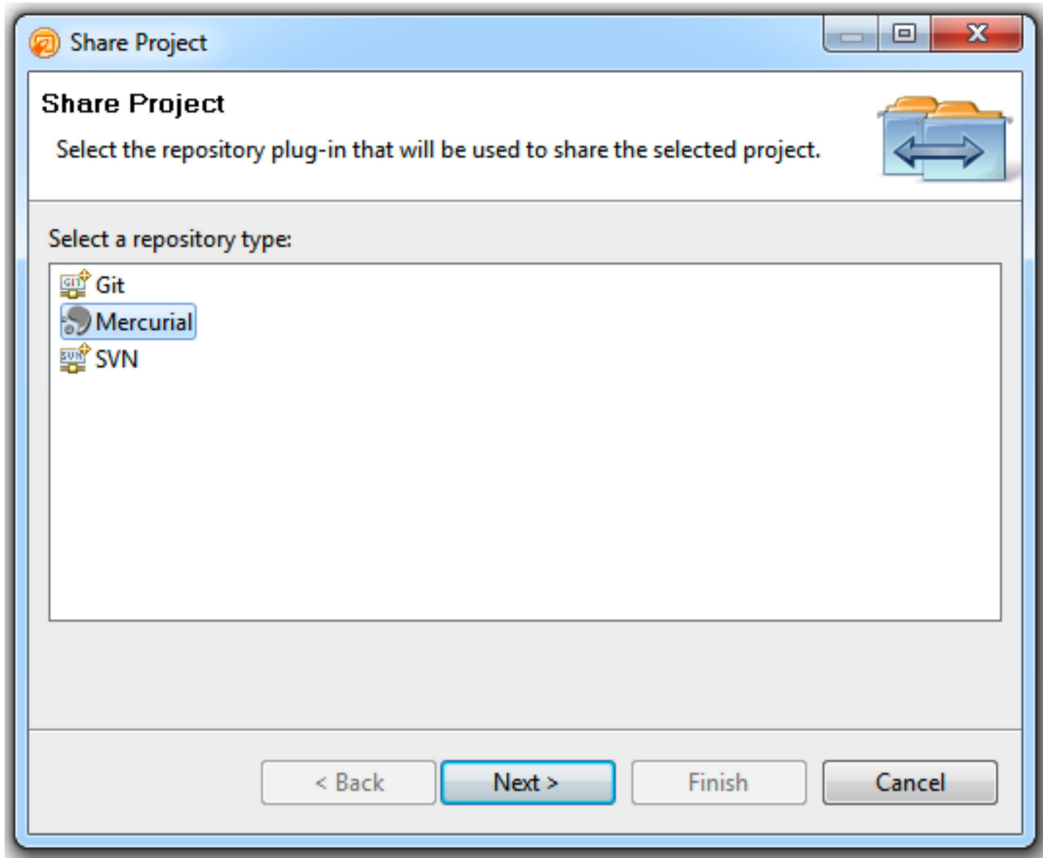






- o Choose 'Team → Share Project...':






Share Project Wizard

Enter Repository Location Information

Define the SVN repository location information. You can specify additional settings for proxy and svn+ssh, https connections.



General | **Advanced** | SSH Settings | SSL Settings

URL:

Label

Use the repository URL as the label

Use a custom label:

Authentication

User:

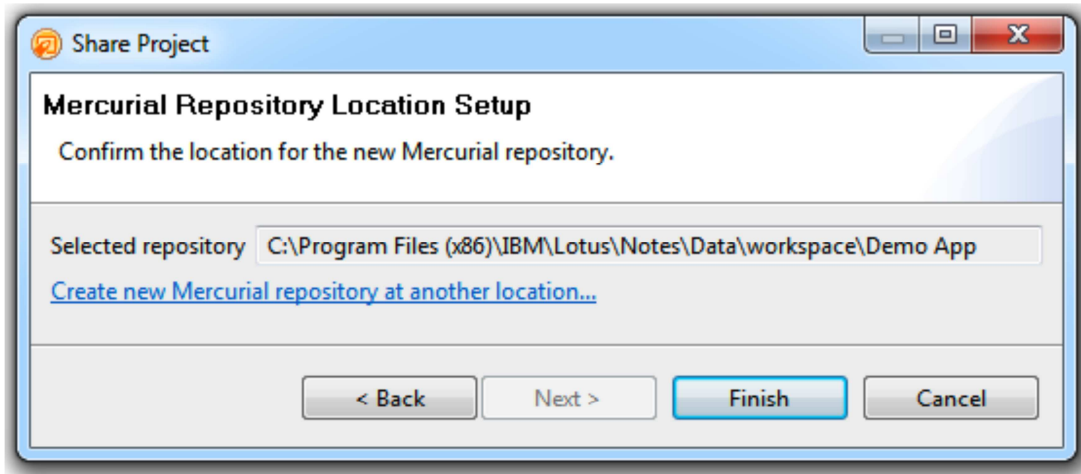
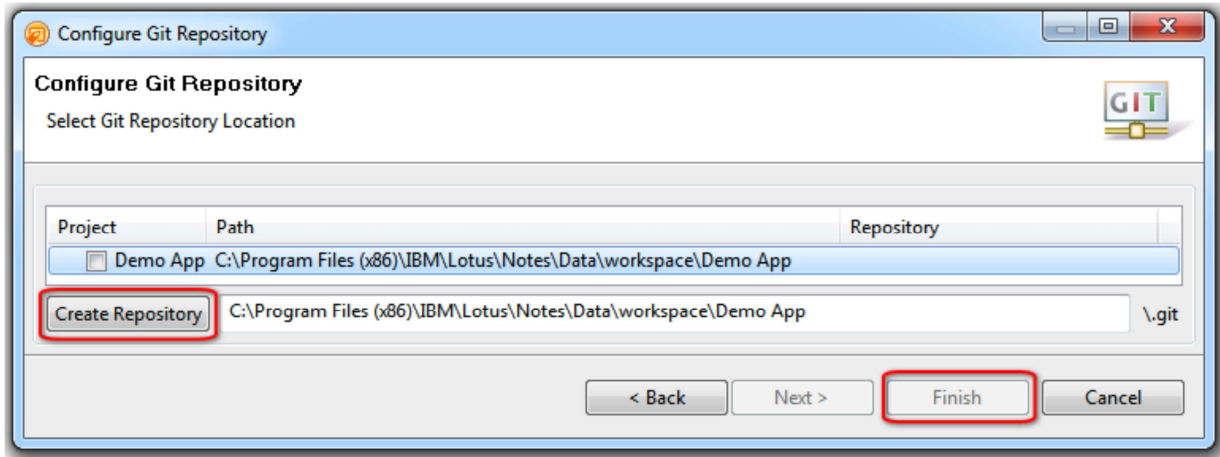
Password:

Save authentication (could trigger secure storage login)

To manage your security data, please see ["Secure Storage"](#)

Show Credentials For:

Validate Repository Location on finish



Actions in DDE with source control

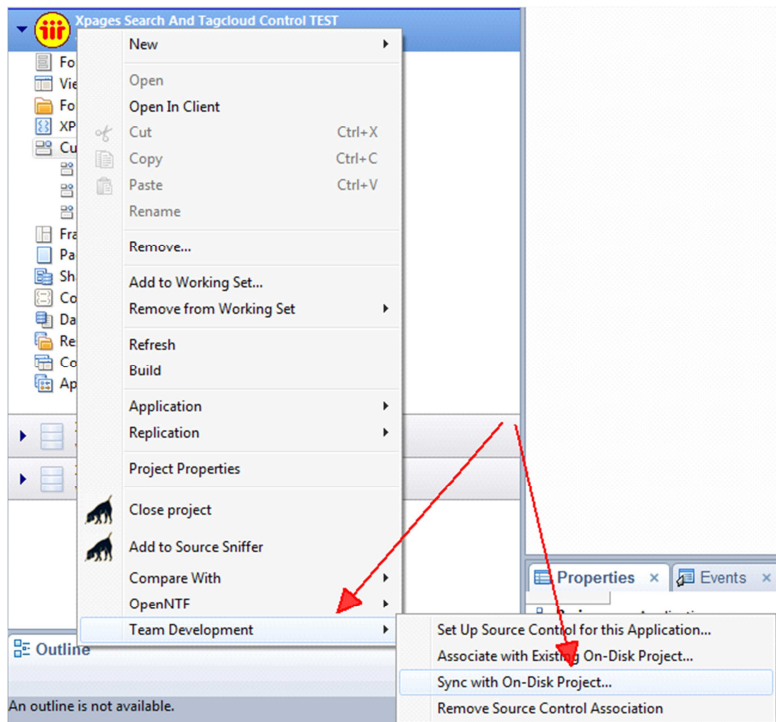
Syncing

Before you can commit changes to the source control you need to sync the changes in the NSF with the on-disk project.

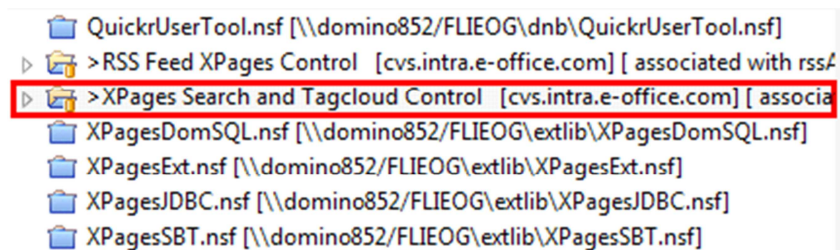
- o Make changes to the database:

Name	Last Modified	Last Modified By			
anotherCustomControl	04-06-2012 13:5...	Frank van der Linden/EOF/E...			
ccSearch	22-04-2012 12:3...	Frank van der Linden/EOF/E...			
newCustomControl	22-04-2012 12:3...	Frank van der Linden/EOF/E...			

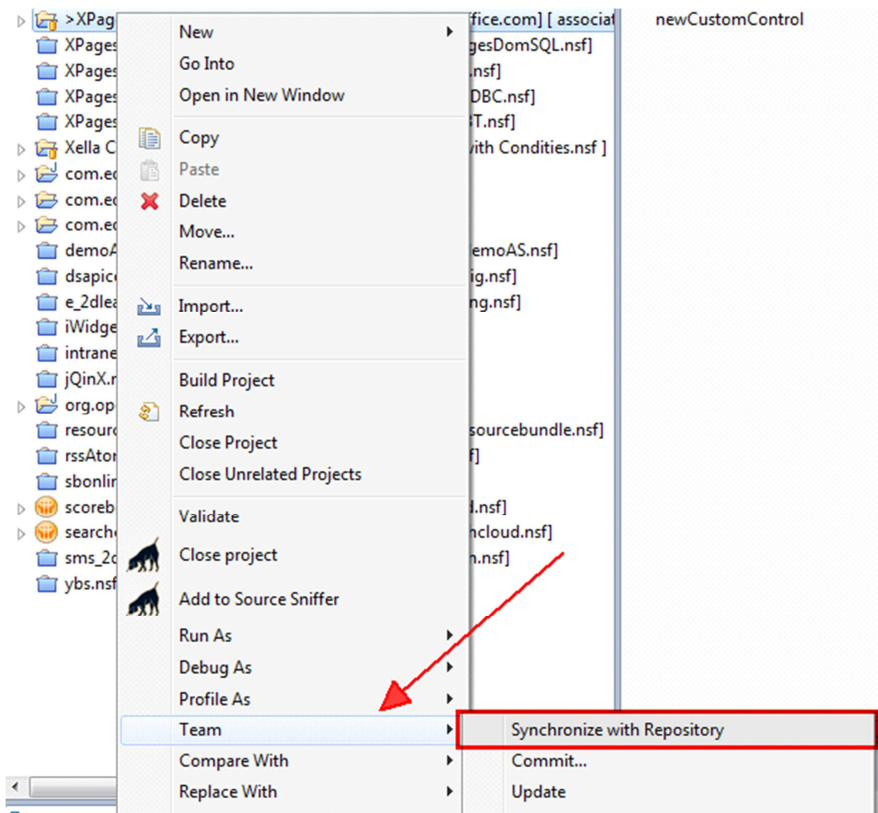
- o Right mouse click.
- o Select 'Team Development → Sync with On-Disk project':



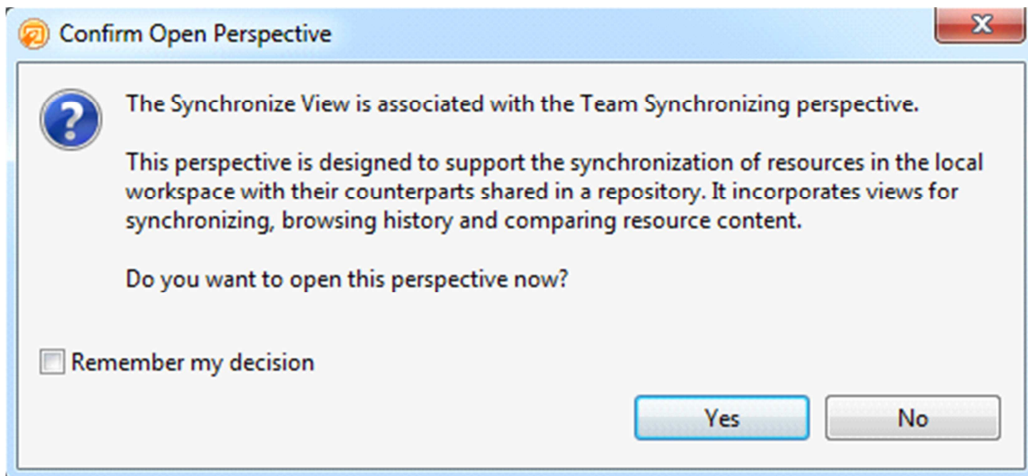
- Open the Navigator view via 'Window → Show Eclipse Views'.
- The arrow before the on-disk Project indicates there are changes to commit:



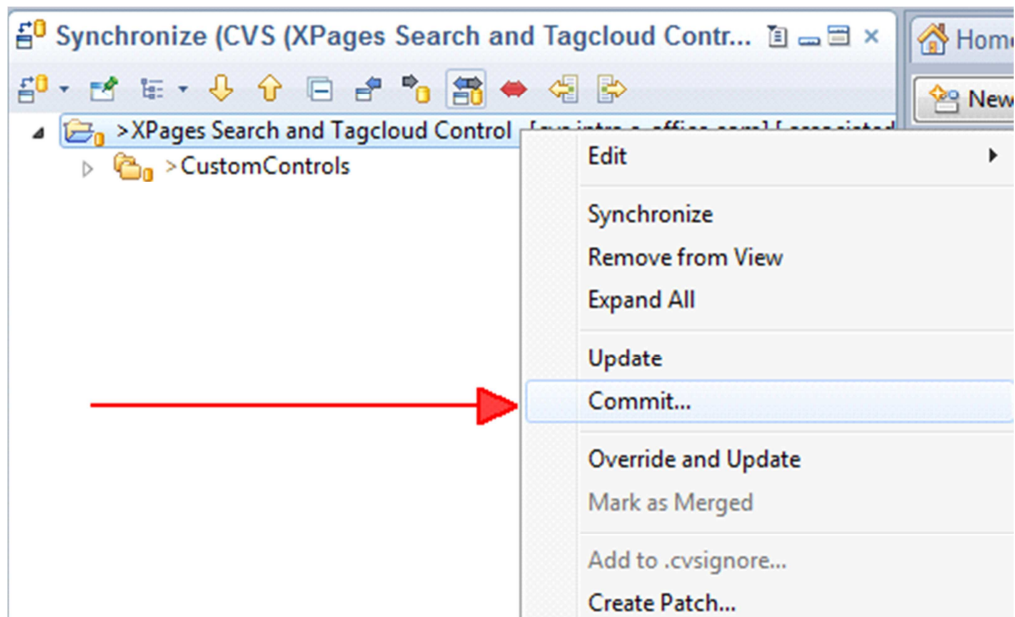
- Right mouse click on project.
- Select 'Team → Synchronize with Repository':



- o A warning of switching perspectives will be presented:

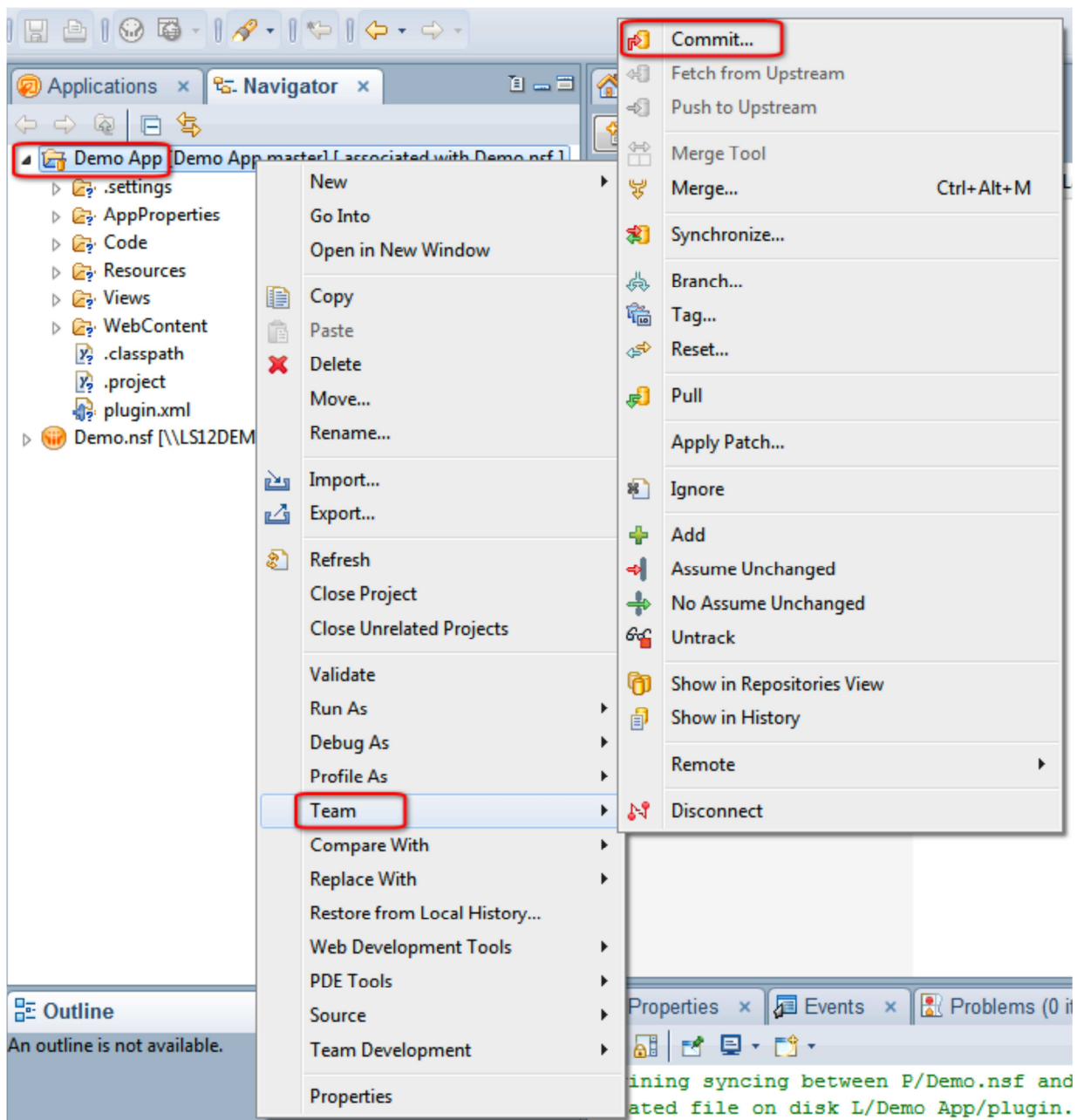


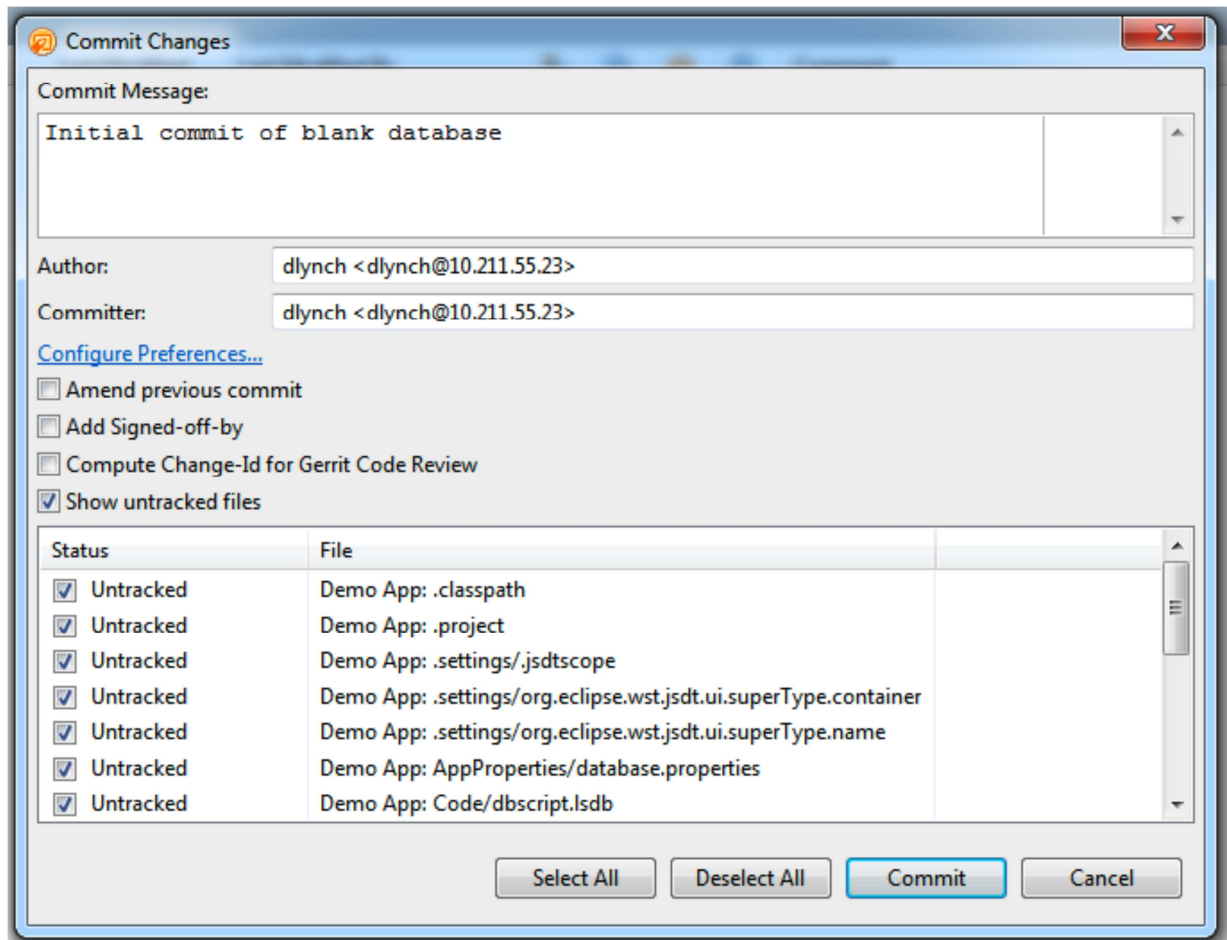
- o If don't want this warning in the future, select the checkbox 'Remember my decision'.
- o The Team Synchronizing perspective opens.
- o It gives an overview of all the changes.
- o Right click on project name and select 'Commit':

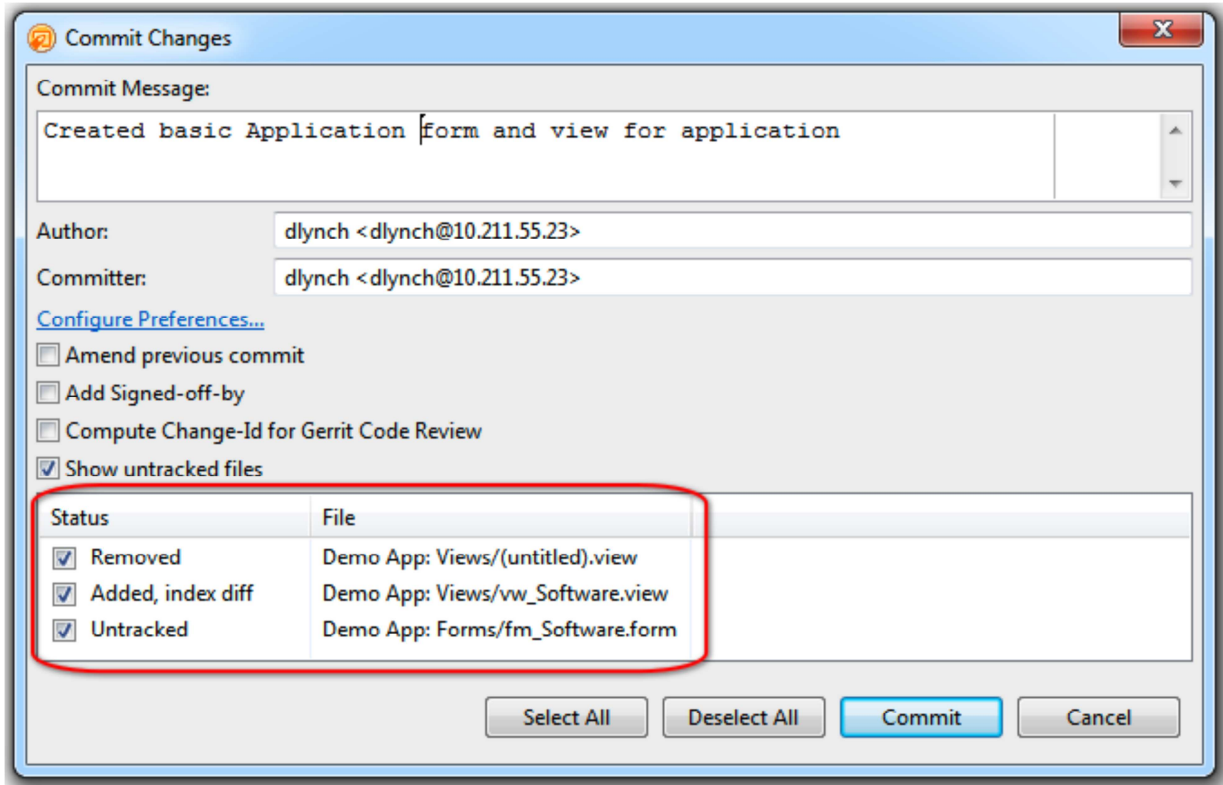


Committing (Mercurial)

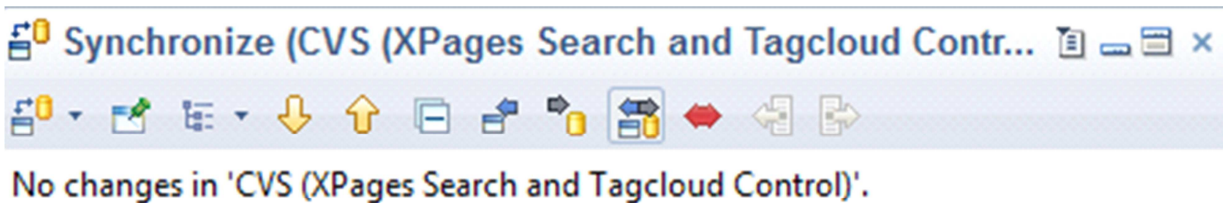
- Before making any changes you should do an initial commit.
- Then you should commit changes every time you make a change and have tested it.
- A change can consist of one or more design elements.
- You should always enter in a good description of the change.







After the Commit operation, the changes have disappeared from the perspective:



As you commit changes a revision history of the application is built up:

Graph	Changeset	Tag	Branch	User	Date	Summary
	4:0e40ae905499	tip		dlynch	2011-12-10 23:30	Added new XPages for the software upload and software download features
	3:ea4b5882da74			dlynch	2011-12-10 23:26	Created first page of application and configured the ExtLib Application Layout Control
	2:7f6bd4922ce6			dlynch	2011-12-10 23:26	Added theme for xPages extending oneUI V2.1
	1:14c2352c65b9			dlynch	2011-12-10 23:25	Created initial form and view for application
	0:feded3625306			dlynch	2011-12-10 23:25	Initial Commit Of Blank Database

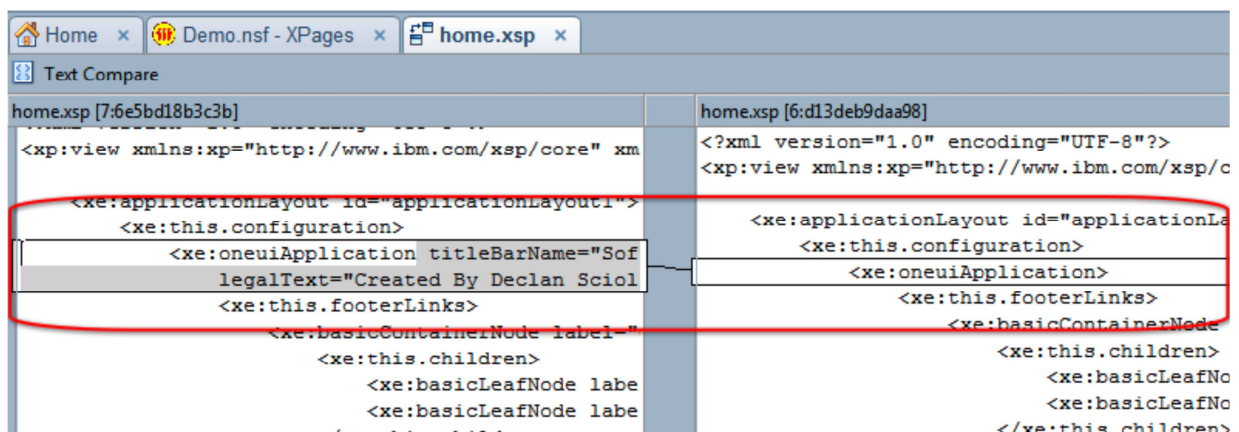
You can use the 'History' panel to see this by using the 'Team → Show In History' menu.

The revision history also shows what design elements changed for the selected Changeset:

Graph	Changeset	Tag	Branch	User	Date	Su
●	7:6e5bd18b3c3b	tip		dlynch	2011-12-10 23:36	M
●	6:d13deb9daa98			dlynch	2011-12-10 23:35	Ac
●	5:b243234363d3			dlynch	2011-12-10 23:35	Re
●	4:0e40ae905499			dlynch	2011-12-10 23:30	Ac
●	3:ea4b5882da74			dlynch	2011-12-10 23:26	Cn
●	2:7f6bd4922ce6			dlynch	2011-12-10 23:26	Ac
●	1:14c2352c65b9			dlynch	2011-12-10 23:25	Cn
●	0:5c1e13625706			dlynch	2011-12-10 23:25	Ini

XPages\home.xsp

Double clicking on a design element in the history will show you the differences:



Handy when you want to remind yourself of exactly what changed at a later date.

Reverting (Mercurial)

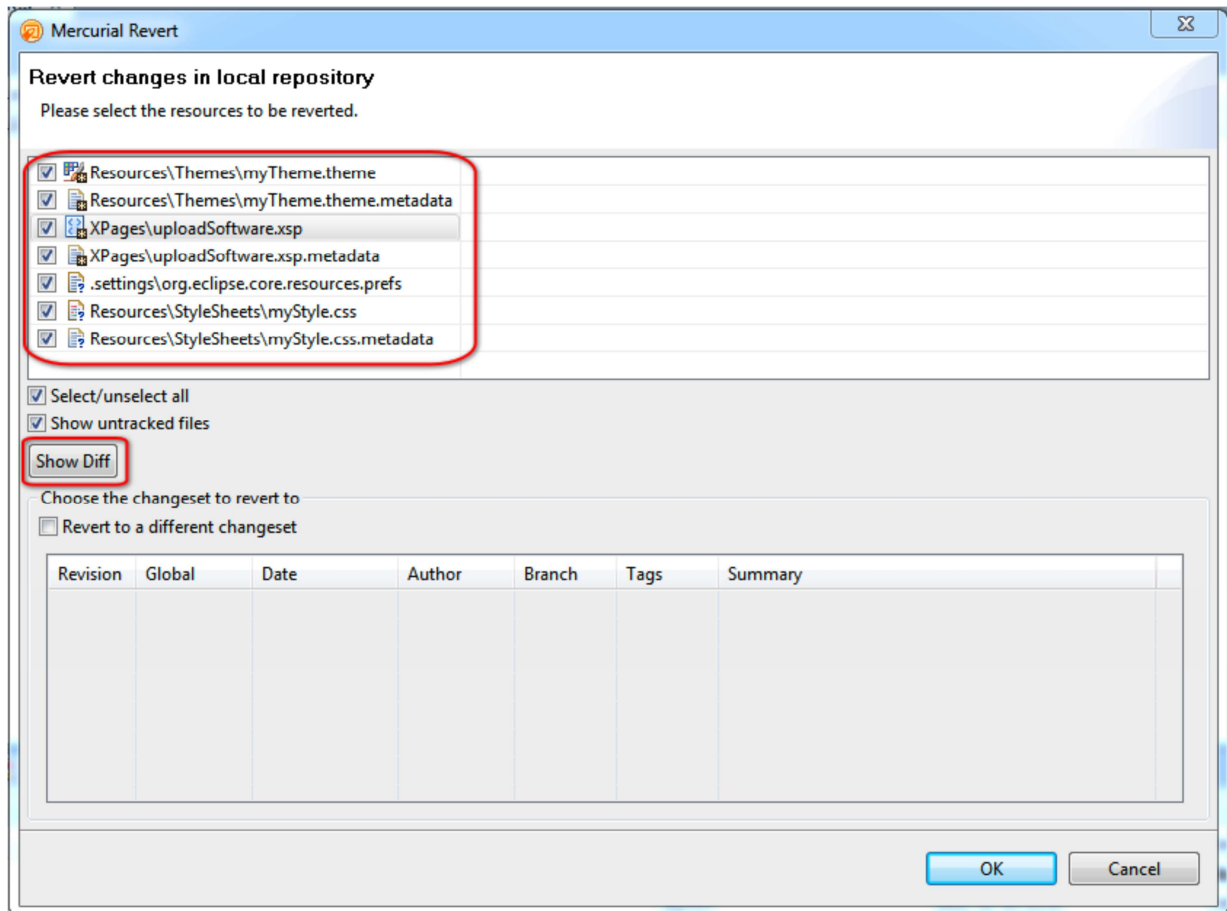
Think of it as an Undo for multiple changes.

Two types of reverting:

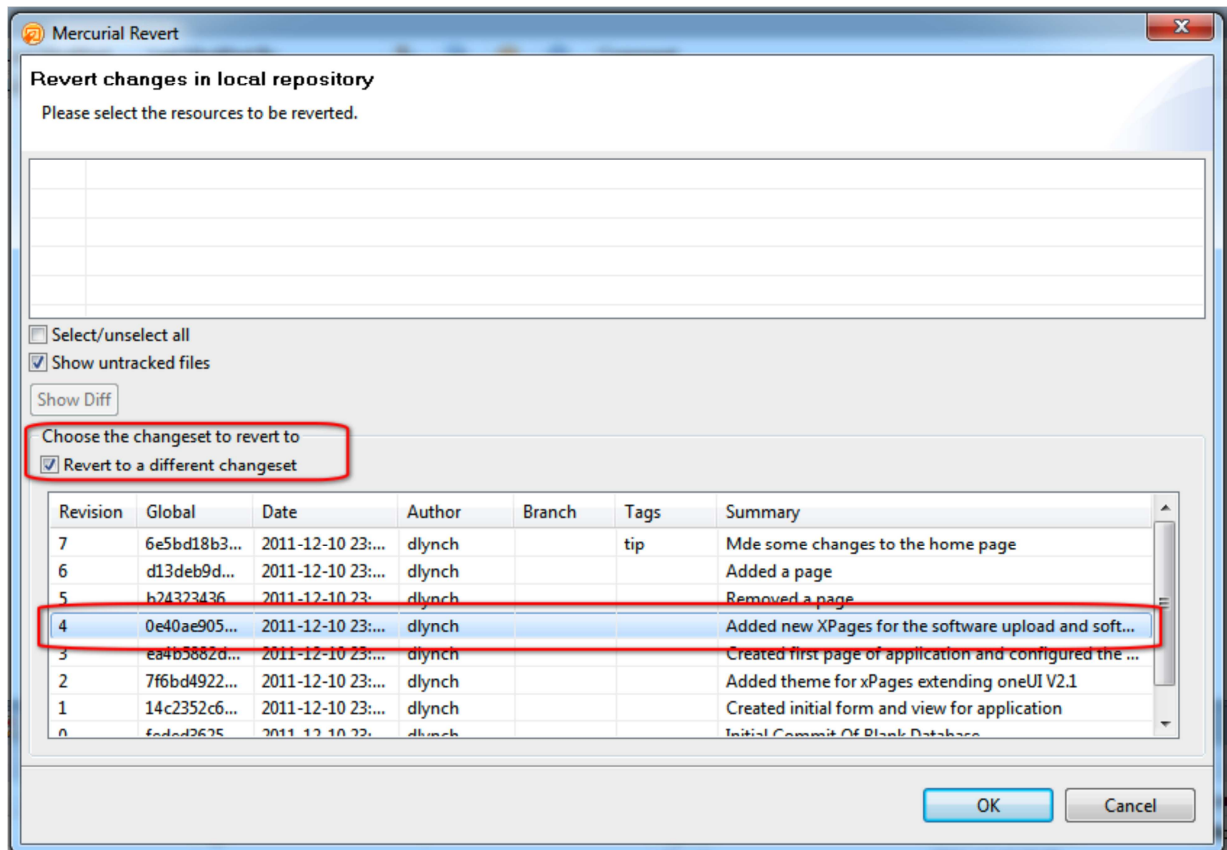
- Reverting changes before they are committed.
- Reverting to an older Changeset.

Select 'Team → Revert' to start the process.

If there are uncommitted changes they will be listed:



If you have not made changes since last commit then you need to select an older revision:



Rollback

Undoes the last action you took.

Backout

- Remove a single Changeset from the middle of the repository.
- May result in a conflict that will need to be resolved.

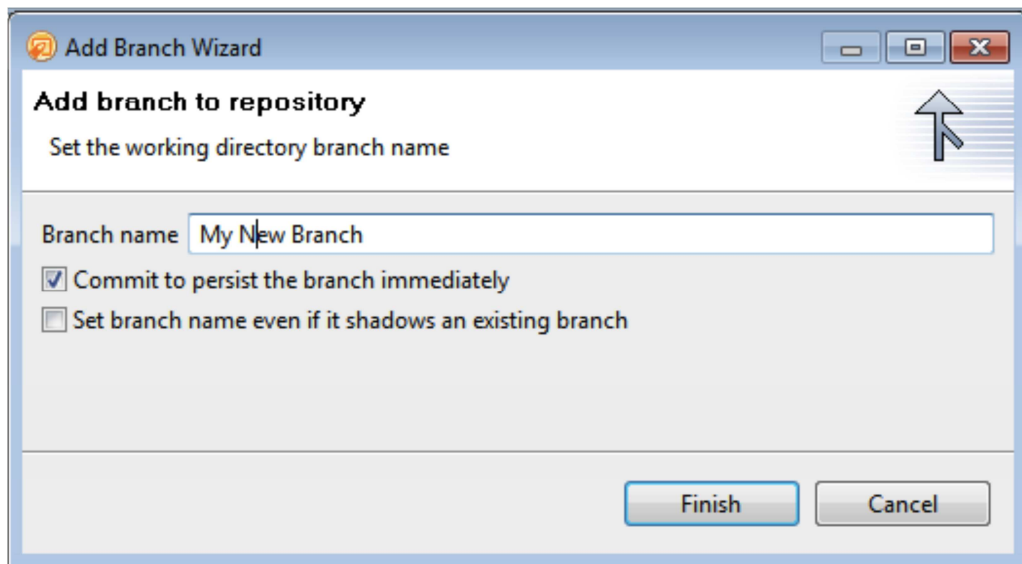
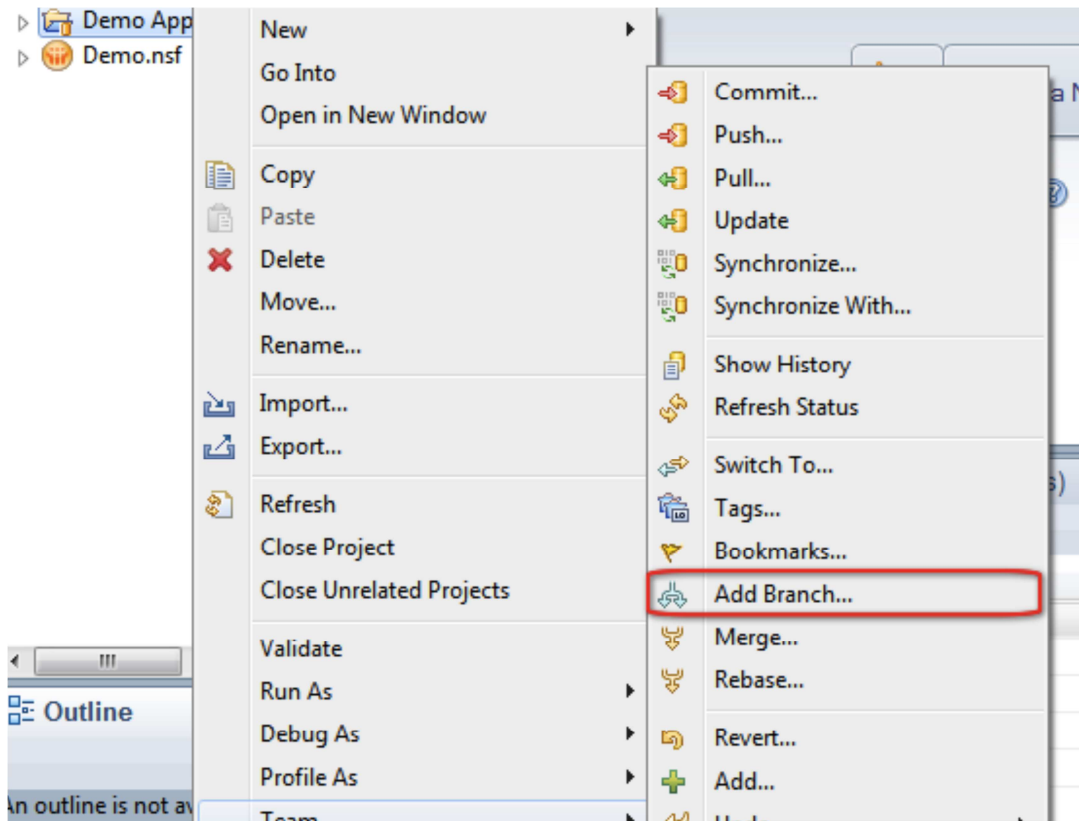
Stripping

- Removes all Changesets back to the selected Changeset.
- Destroys the true history of the repository.
- Stripped Changesets cannot be recovered.

Branching (Mercurial)

- Branching allows you to have multiple versions of an application within the same code base (e.g. customer specific).
- A great way to try out new code without affecting other branches.
 - A developer may create an 'experimental' branch to try out a new way of doing something.
- Or create a new branch for each new version of the application.
 - Some developers could be working on maintaining the V1 branch.
 - Some developers could be working on new features for the V2 branch.

- Branches can also be further branched as required.



As you commit changes to the branch the history graph will show the new branch:

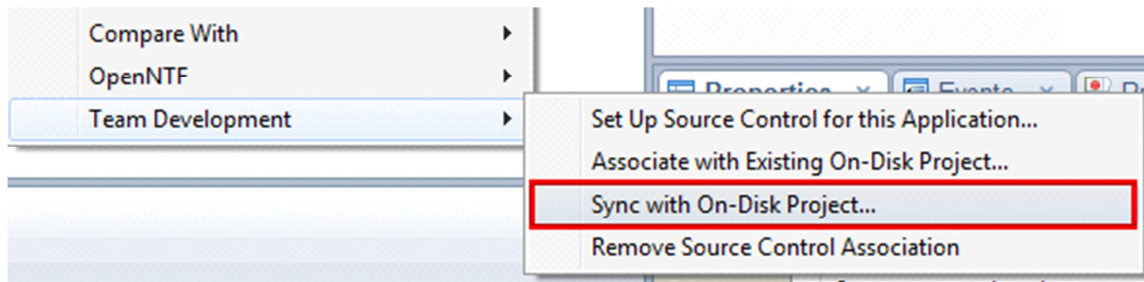
Graph	Changeset	Tag	Branch	User	Date	Summary
	14:e328f2ac4494	tip	Mobile Experiments	Administrator	2011-12-11 22:21	added a page to mobile branch
	13:ae3b06ba527c		Version 2	Administrator	2011-12-11 22:20	Added a page to the v2 Branch
	12:c715ab147410			Administrator	2011-12-11 22:19	Added a page to default branch
	11:225712744b5e		mobile iPad support	Administrator	2011-12-11 22:17	added blank iPad pages
	10:24c7b45017e4		mobile iPad support	Administrator	2011-12-11 22:16	Starting 'mobile iPad support' branch
	9:e1003725f823			Administrator	2011-12-11 22:09	Added password info
	8:6399677e6777		Mobile Experiments	Administrator	2011-12-11 22:07	Added mobile about page
	7:56d02772c62a		Mobile Experiments	Administrator	2011-12-11 22:07	Added mobile serial and help pages
	6:9ebf58f0ccfd		Mobile Experiments	Administrator	2011-12-11 22:06	Starting 'Mobile Experiments' branch
	5:235a69d316a1			Administrator	2011-12-11 22:05	Added the serial number pages
	4:d35c526984d8		Version 2	Administrator	2011-12-11 22:03	Added standard Help and About pages
	3:342cff6ea5fc		Version 2	Administrator	2011-12-11 22:02	Starting 'Version 2' branch
	2:45f4af06e302			Administrator	2011-12-11 21:59	Updated license information in the application foo...
	1:4365d3e41074			Administrator	2011-12-11 21:57	Added main application homepage
	0:f60c1076bc6f			Administrator	2011-12-11 21:54	Initial commit of blank database

Branching (CVS)

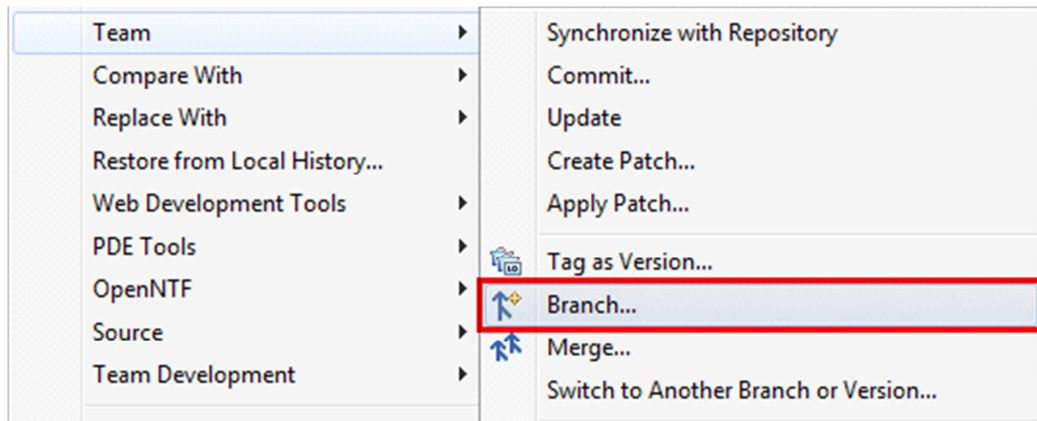
- Create a branch of the project.
 - For example with customer specific design elements:

Name	Last Modified	Last Modified By					Co
newBranch	04-06-2012 15:0...	Frank van der Linden/EOF/E...					
search	04-06-2012 14:5...	Frank van der Linden/EOF/E...					

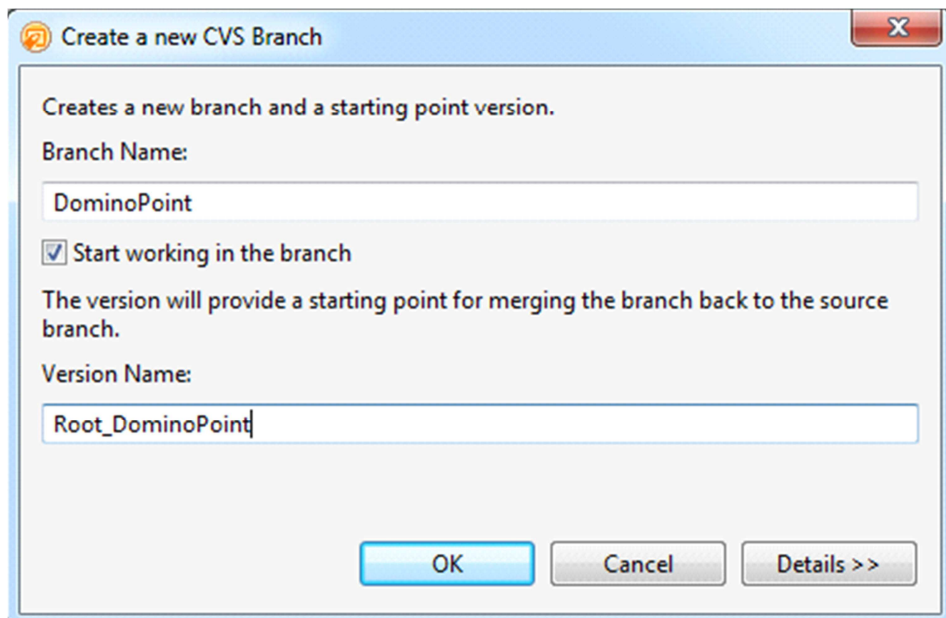
- Sync database with On-Disk Project:



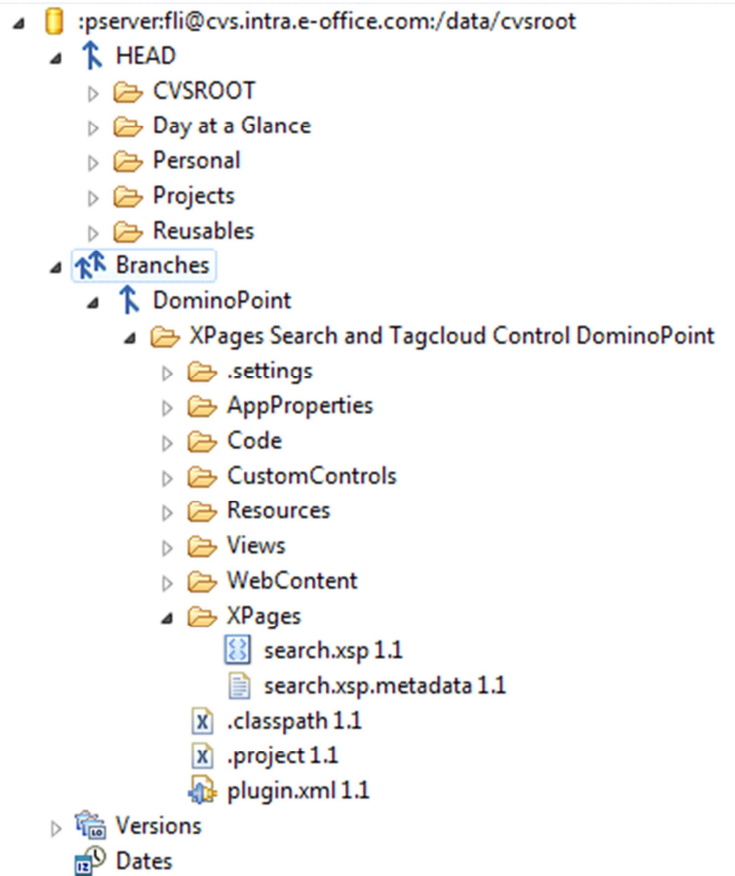
- Switch to the Navigator perspective.
- Right click project.
- Select 'Team → Branch...':



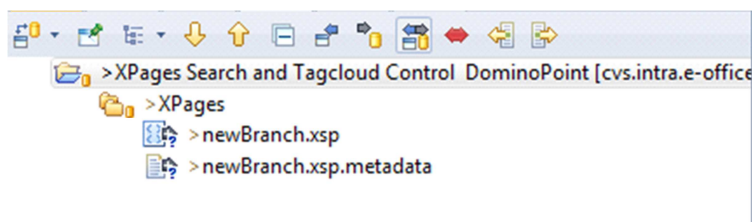
- Fill in a Branch Name.
- And a Version Name:



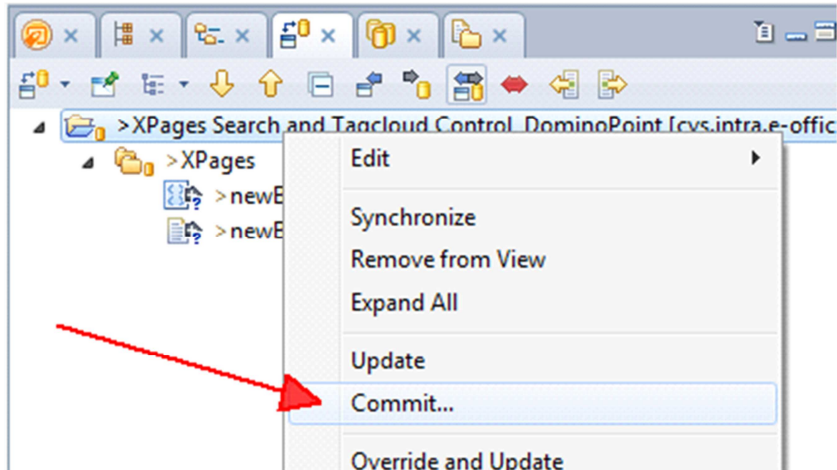
- Click 'OK'.
- In the CVS Repository perspective the branch has been created:



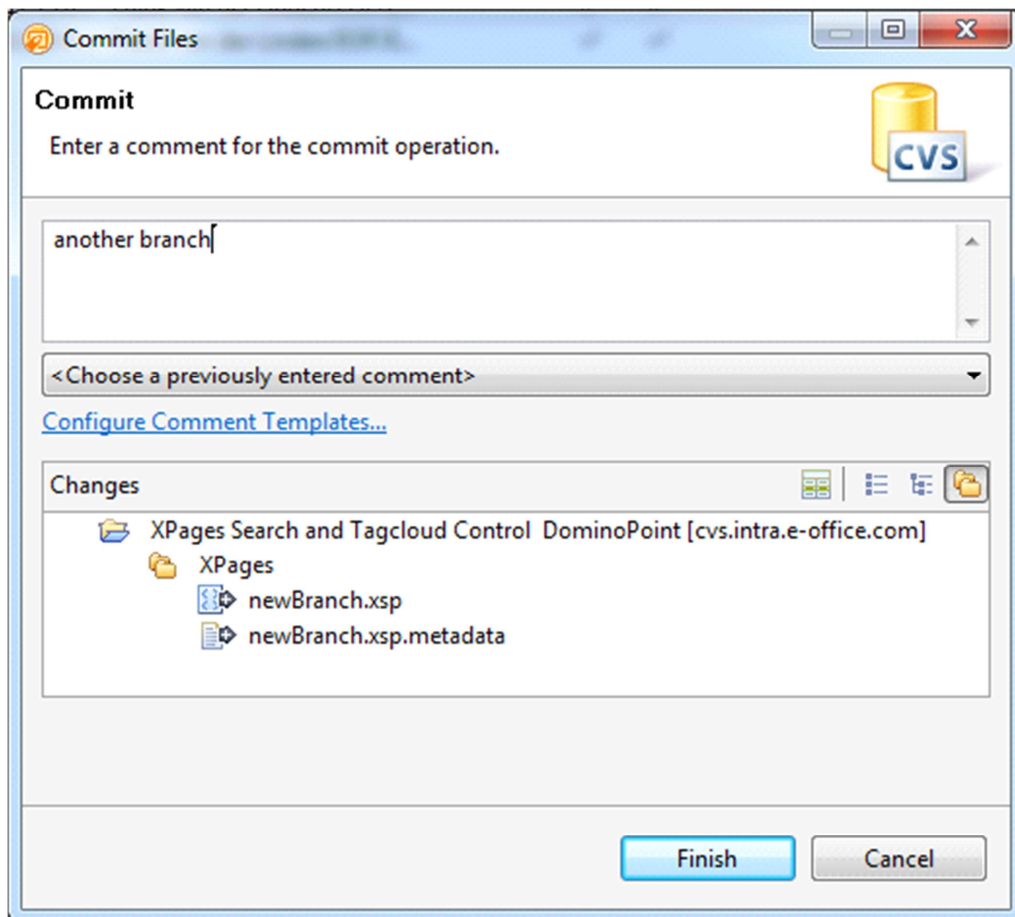
- The new changes are not present yet.
- Switch to the Team Synchronizing perspective:



- Commit the new changes to the branch. Right click the project name:



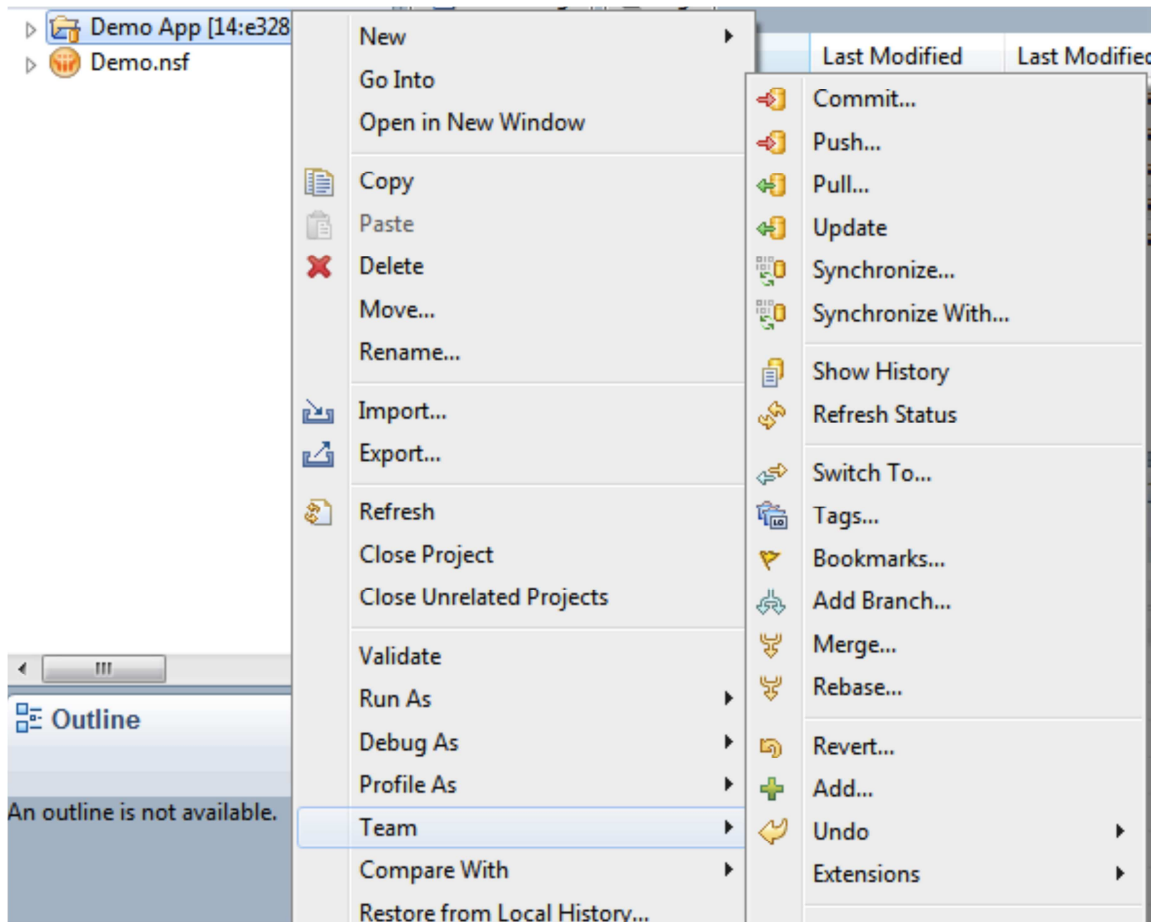
- o Comment the commit operation:



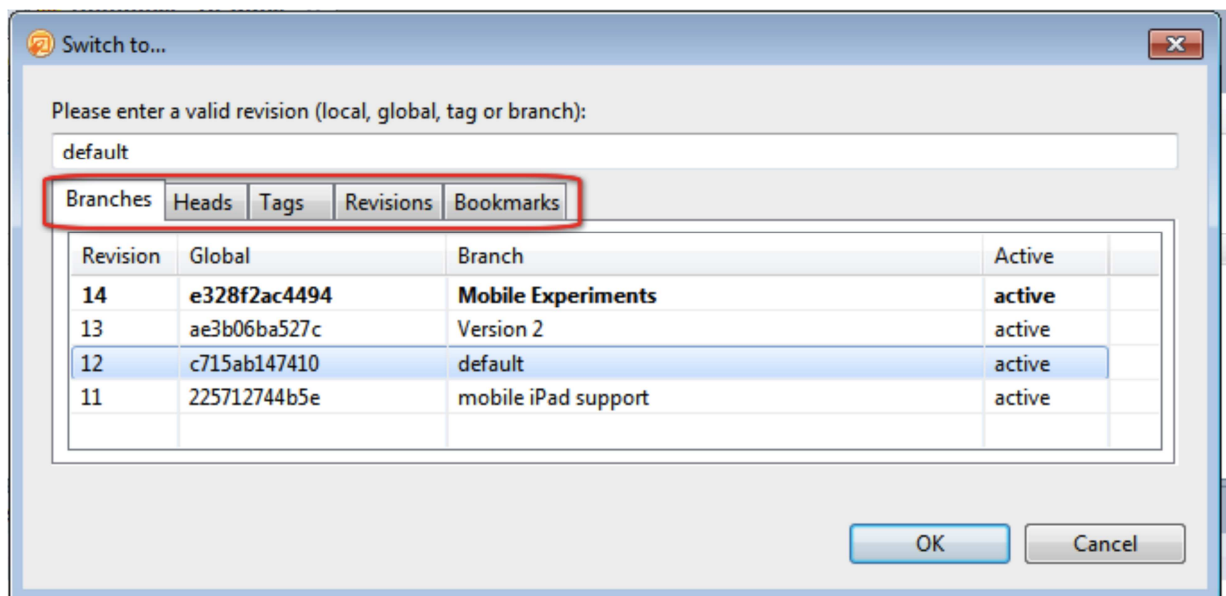
- o Click 'Finish'.

Switching

Use the 'Switch To' menu to change between branches:



Select where you want to switch to, it doesn't have to be a branch:



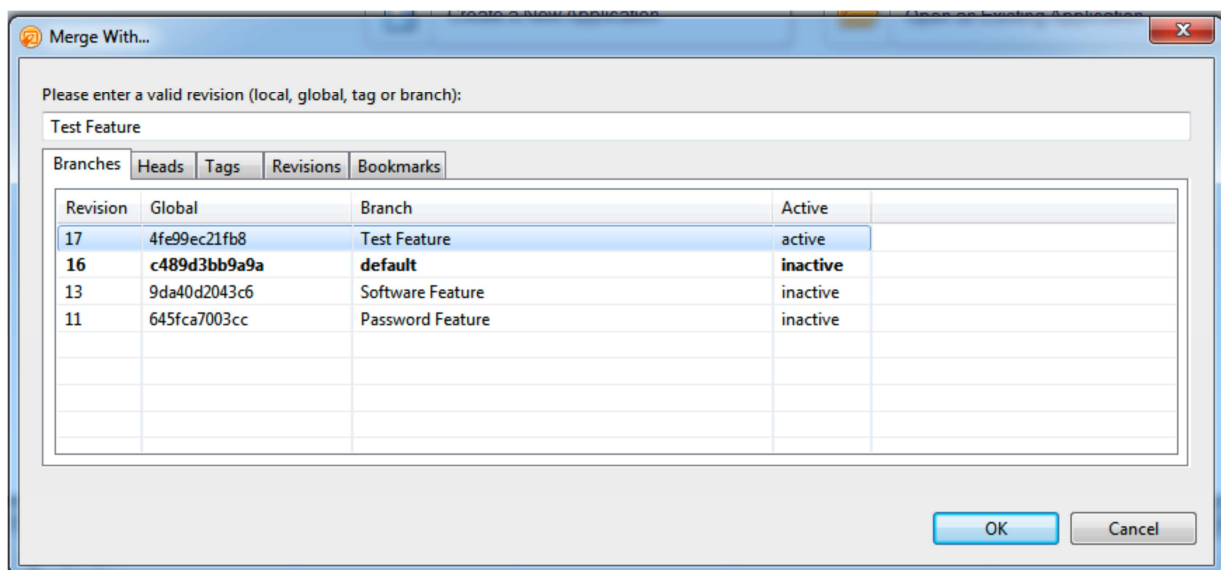
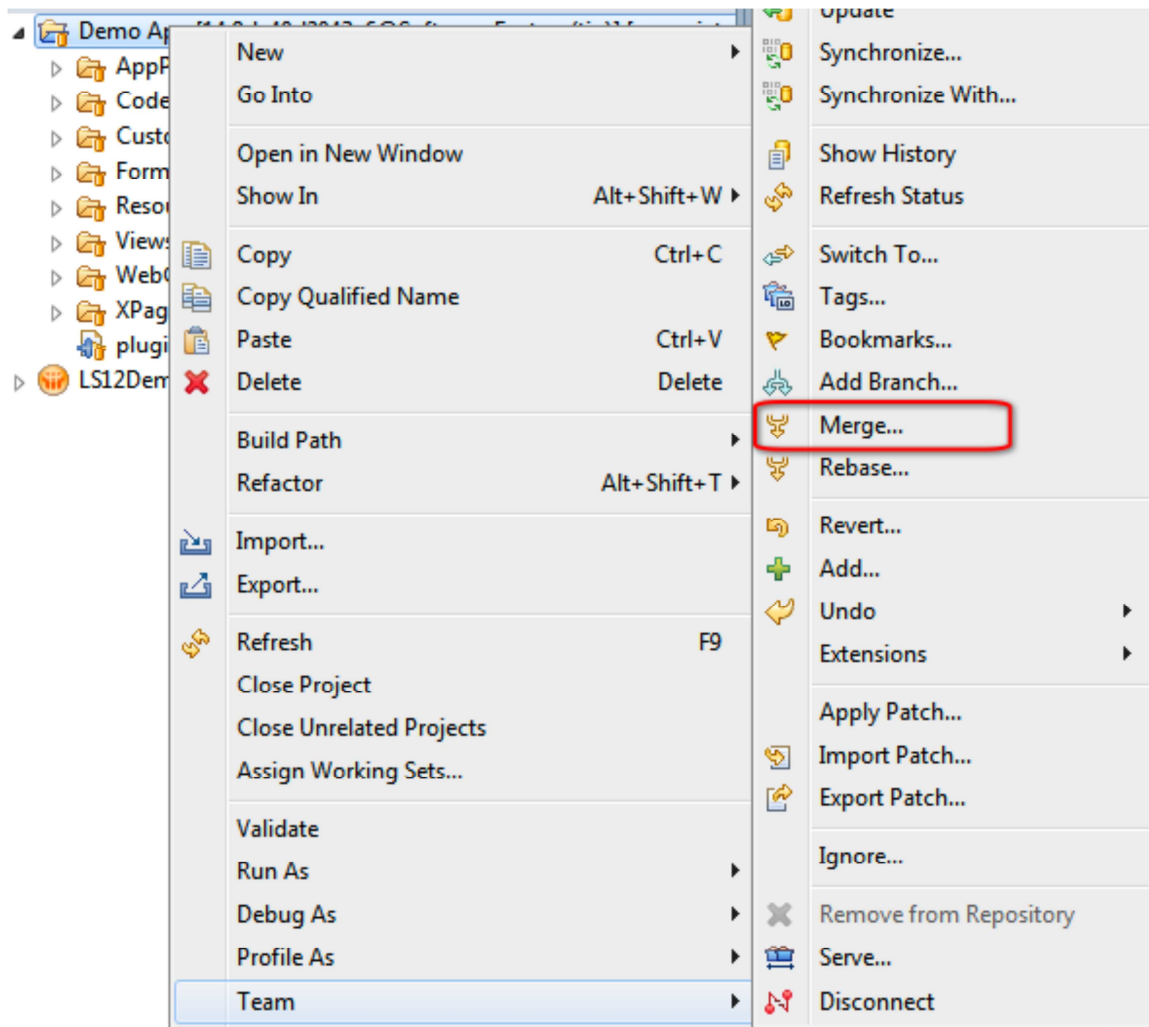
After switching, the repository will be updated to reflect that revision.

Then the 'NSF ↔ On Disk Project' sync will kick in and the NSF will be updated to reflect the on disk version.

If there are a lot of differences in the revision you have switched to this process can take a few moments.

Merging (Mercurial)

- Taking the changes made in one branch and applying them to a different branch.
 - Bug fixes in the V1 branch can be merged into the V2 branch.
 - The experimental branch can be merged into active development branch.
- Merging branches is often easy but edge cases can be tricky.
 - A design element deleted in one branch but changed in another branch.
 - A design element changed in the same place in both branches.
- Not 100% automatic, the developer needs to be involved.
 - Merge conflicts need to be resolved before the merge can complete.
 - Binary files like images can't be merged so a winner needs to be selected.
 - Developer should fully test the resulting code before committing the final merged results.
 - Developer can reverse the merge by not committing it to the repository.



Status	File
Unresolved	XPages/home.xsp

Applications x Package Explorer x

Demo App [16:ab7928672ab4@default(tip) MERGING!

- AppProperties
- Code
- CustomControls
- Forms
- Resources
- Views
- WebContent
- XPages
 - home.xsp
 - home.xsp.metadata
 - home.xsp.orig
 - passwords.xsp

Home x LS12Demo.nsf - XPages x *home.xsp x

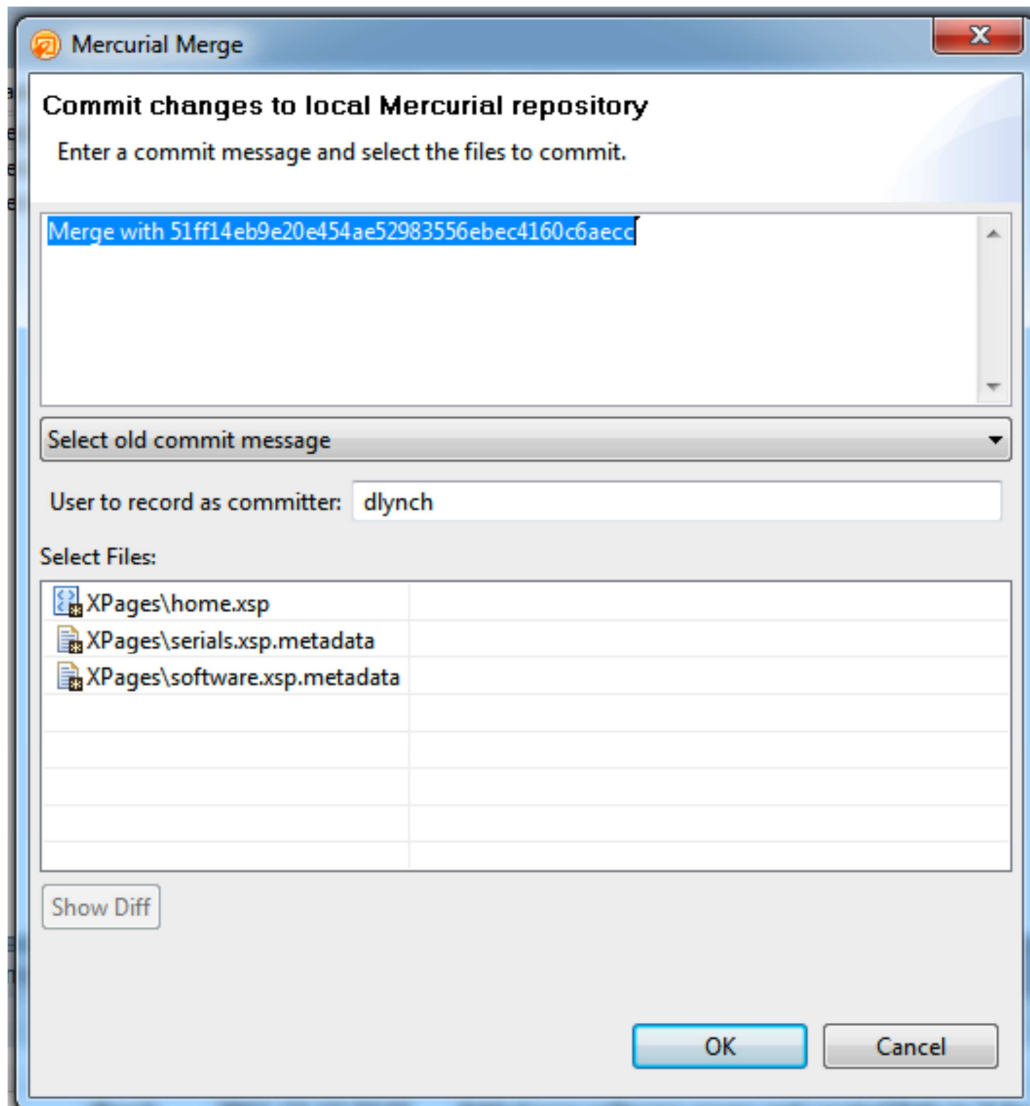
Text Compare

home.xsp	home.xsp [15:f4dda764c]
<pre> </xe:pageTreeNode> <xe:pageTreeNode label=" page="/passwords.xsp </xe:pageTreeNode> </xe:this.titleBarTabs> </xe:oneuiApplication> </xe:this.configuration> </xe:applicationLayout> </pre>	<pre> </xe:pageTreeNode> <xe:pageTreeNode label=" page="/software. </xe:pageTreeNode> </xe:this.titleBarTabs> </xe:oneuiApplication> </xe:this.configuration> </xe:applicationLayout> </pre>

The screenshot displays an IDE window titled 'Text Compare' with two panes comparing XML files. The left pane shows the original file with two red boxes highlighting differences: one around a closing tag and another around a page tree node. The right pane shows the merged file with a blue highlight on a page tree node. Below the editor is a 'Mercurial Merge' window with a table of merge results and a context menu for the 'Unresolved' entry.

Status	File
Unresolved	XPages/home.xsp

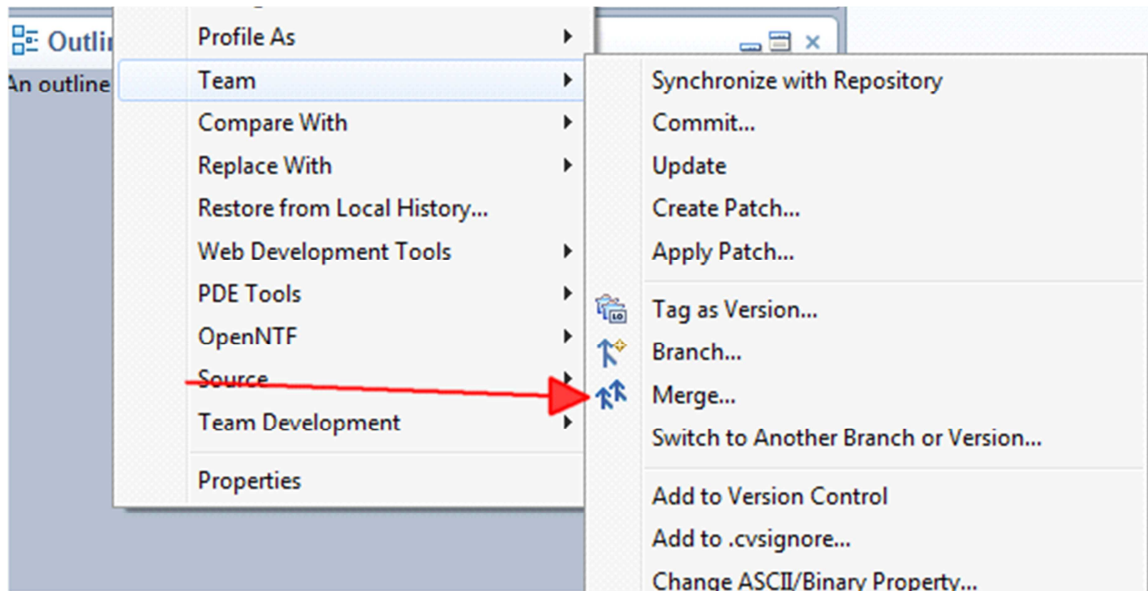
- Open in Merge Editor
- Open in Default Editor
- Show History
- Mark Resolved
- Mark Unresolved
- Abort Merge



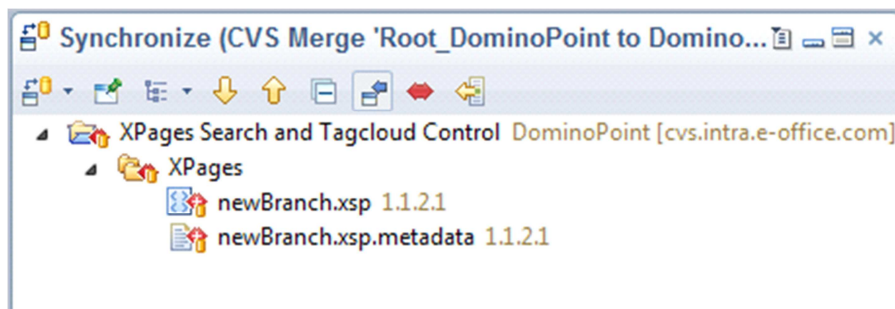
Graph	Changeset	Tag	Branch	User	Date	Summary
	18:54b91139b1c1	tip		dlynch	2011-12-13 22:38	Added tag V2.0.0 for changeset 34b4e43d2551
	17:34b4e43d2551	V2.0.0		dlynch	2011-12-13 22:37	Merge with f4ddaeea764c7a3f3477cb2a063a233228313df0
	16:ab7928672ab4			dlynch	2011-12-13 22:30	Merge with 645fca7003cc0b720f449b5d4a2ea0aefdf6f42
	15:f4ddaeea764c		Software Feature	dlynch	2011-12-13 22:30	Merge with 51ff14eb9e20e454ae52983556ebec4160c6aecc
	14:9da40d2043c6		Software Feature	dlynch	2011-12-13 22:23	Added new software xpage and created links to it from the homepage
	13:9172ee2de750		Software Feature	dlynch	2011-12-13 22:22	Created software form and view
	12:51ff14eb9e20			dlynch	2011-12-13 22:21	Changed application name in homepage
	11:645fca7003cc		Password Feature	dlynch	2011-12-13 22:19	Added password link to titlebar on homepage
	10:48f8b3399d18		Password Feature	dlynch	2011-12-13 22:18	Created password XPage
	9:113e043e67df		Password Feature	dlynch	2011-12-13 22:17	Created Password form and view
	8:a26d5333dc58		Password Feature	dlynch	2011-12-13 22:16	Starting 'Password Feature' branch
	7:284ac298a2e9		Software Feature	dlynch	2011-12-13 22:16	Starting 'Software Feature' branch
	6:a62b1026f882			dlynch	2011-12-13 22:15	Added tag V1.0.0 for changeset f3f2e2444d65
	5:f3f2e2444d65	V1.0.0		dlynch	2011-12-13 22:11	Removed leftover info from Homepage
	4:5be408cb3acf			dlynch	2011-12-13 22:10	Changed the homepage to be an instructions page Added a new Serial Num...
	3:eebf4708ea5c			dlynch	2011-12-13 22:05	Created form and view for Serial Numbers
	2:ffd659b48d51			dlynch	2011-12-13 22:02	Updated home page Added left side menu
	1:3ccd20d2b840			dlynch	2011-12-13 21:53	Created initial homepage Added an application theme extending oneui V2.1 ...
	0:a7b63fbc331d			dlynch	2011-12-13 21:46	Initial Commit

Merging (CVS)

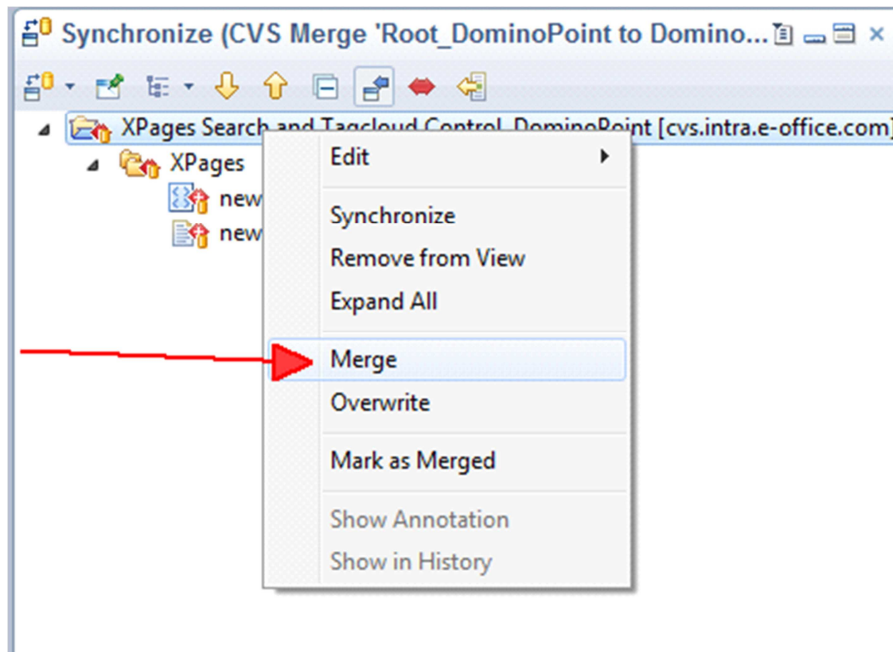
- Branches can also be merged together. Right mouse 'Team → Merge':



- The Synchronize perspective will open:



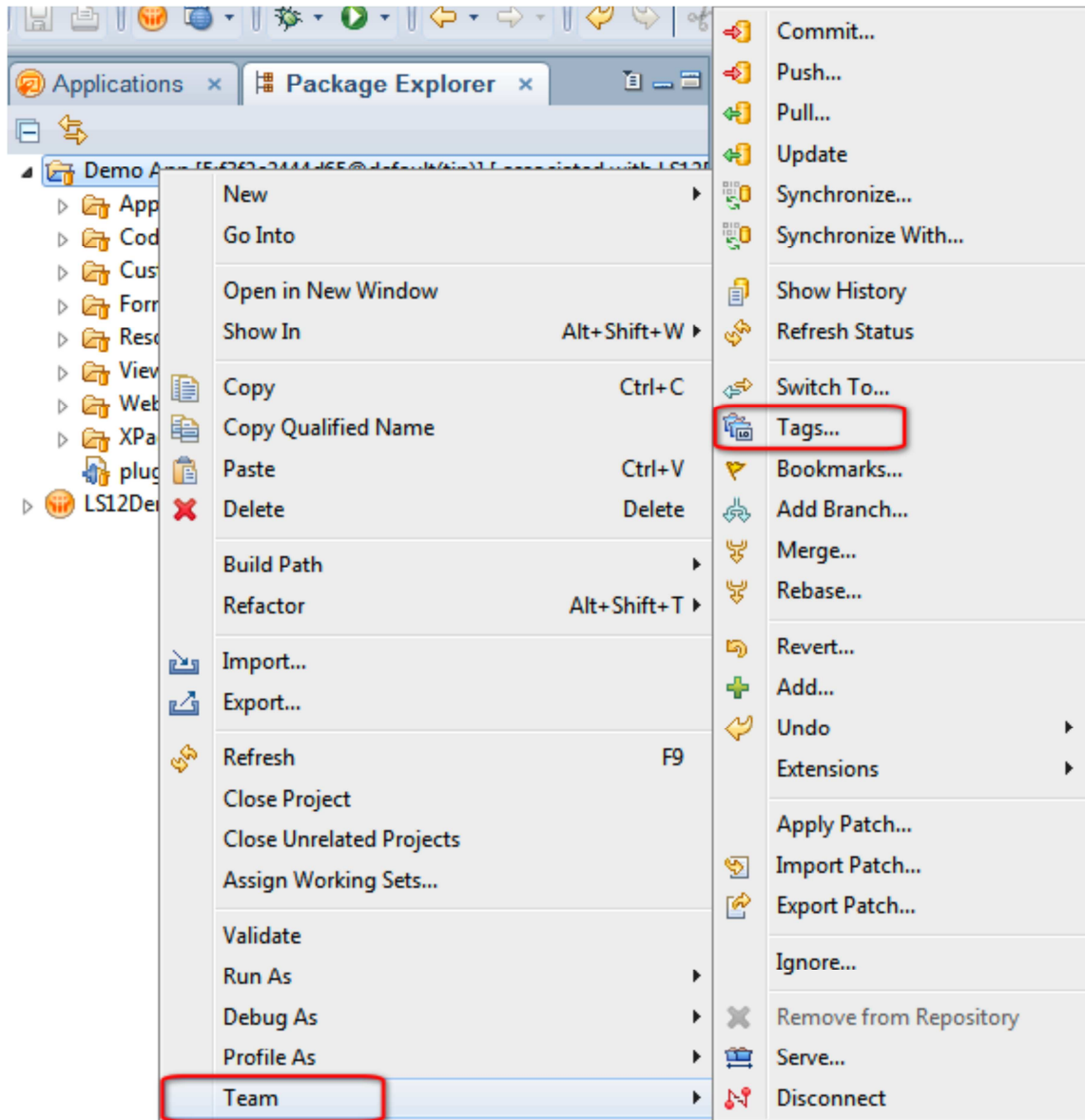
- It shows the modification who will commit to the base.
- Right click 'Merge':

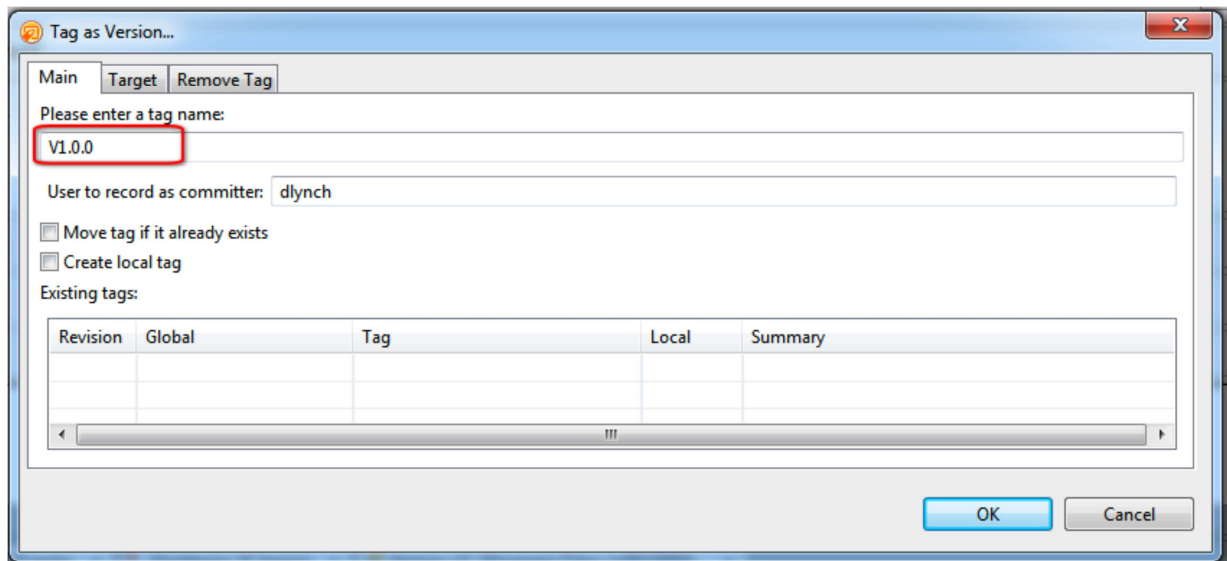


- Two branches will be merged together.

Tagging (Mercurial)

- As the repository history gets bigger it gets harder to find a specific revision.
- Tagging allows you to assign a name to a specific revision.
 - When you make a new release to production tag the revision with the version number.
 - Tag your alpha, beta and Release Candidate revisions.
- Tags don't take up resources so don't hesitate to use them.
- Tags can be reassigned but this is a bad practice.
 - Some source control systems allow you to move a tag to a different revision.
 - Destroys the true history of the repository.

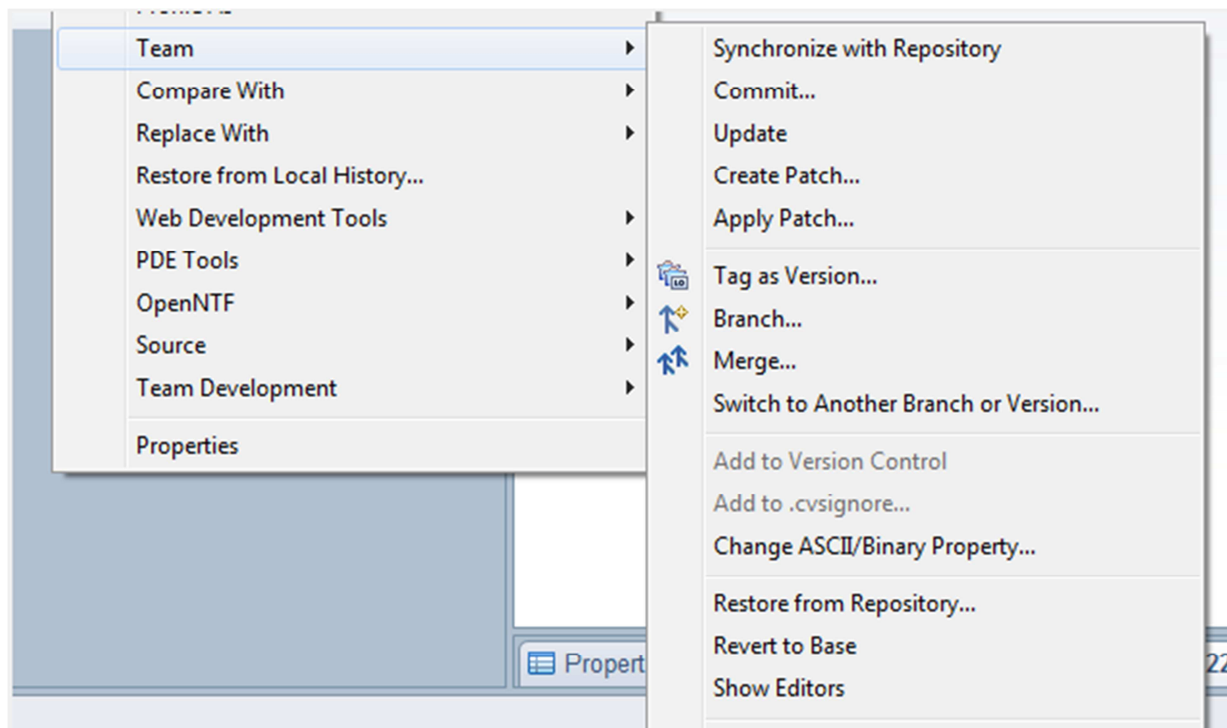




Graph	Changeset	Tag	Branch	User	Date	Summary
	14:9da40d2043c6	tip	Software Feature	dlynch	2011-12-13 22:23	Added new software xpage and created links to it from the homepage
	13:9172ee2de750		Software Feature	dlynch	2011-12-13 22:22	Created software form and view
	12:51ff14eb9e20			dlynch	2011-12-13 22:21	Changed application name in homepage
	11:645fca7003cc		Password Feature	dlynch	2011-12-13 22:19	Added password link to titlebar on homepage
	10:48f8b3399d18		Password Feature	dlynch	2011-12-13 22:18	Created password XPage
	9:113e043e67df		Password Feature	dlynch	2011-12-13 22:17	Created Password form and view
	8:a26d5333dc58		Password Feature	dlynch	2011-12-13 22:16	Starting 'Password Feature' branch
	7:284ac298a2e9		Software Feature	dlynch	2011-12-13 22:16	Starting 'Software Feature' branch
	6:a62b1026f882			dlynch	2011-12-13 22:15	Added tag V1.0.0 for changeset f3f2e2444d65
	5:f3f2e2444d65	V1.0.0		dlynch	2011-12-13 22:11	Removed leftover info from Homepage
	4:5be408cb3acf			dlynch	2011-12-13 22:10	Changed the homepage to be an instructions page Added a new Serial Num...
	3:eefb4708ea5c			dlynch	2011-12-13 22:05	Created form and view for Serial Numbers
	2:ffd659b48d51			dlynch	2011-12-13 22:02	Updated home page Added left side menu
	1:3ccd20d2b840			dlynch	2011-12-13 21:53	Created initial homepage Added an application theme extending oneui V2.1 ...
	0:a7b63fbc331d			dlynch	2011-12-13 21:46	Initial Commit

Tagging (VCS)

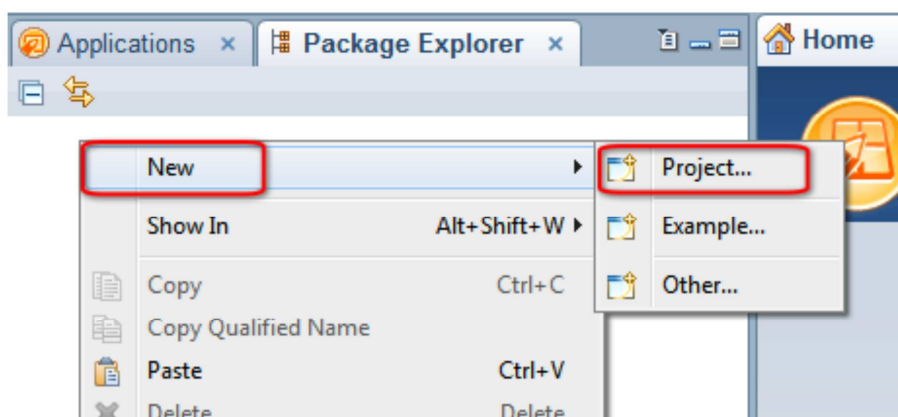
- Tag as Version.
- Can switch to a other version or branch.



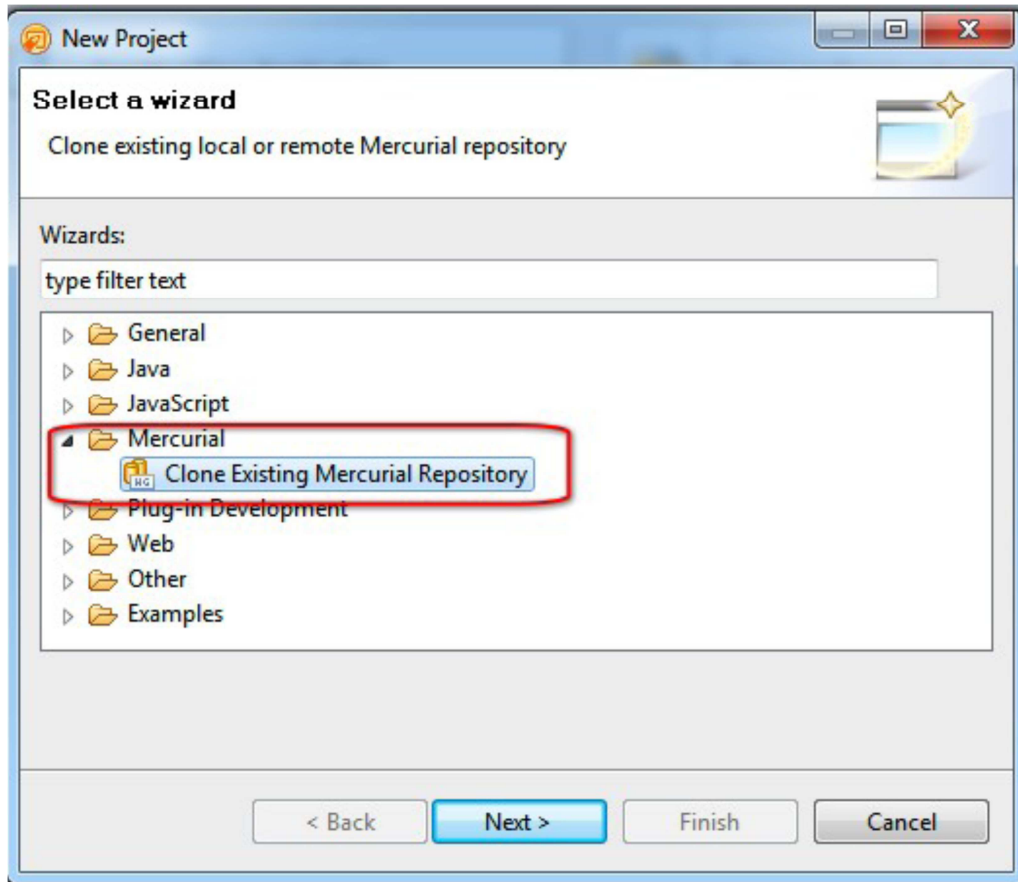
Cloning (Mercurial)

- Cloning a repository creates a copy of a remote repository on your local machine.
 - Normally done when a new developer joins a team.
 - The clone remembers where it originally came from so you can later pull and push changes.
- Cloning just creates the 'on disk' version of the application.
 - The developer can then create a real NSF/NTF to work in.
- The initial clone process can be slow for a large project.

Start the process in the Package Explorer by creating a new project:



Select your Source Control system from the list of wizards:



Mercurial clone repository wizard

Clone repository

Ready to start clone

Repository location
 URL Local...

Authentication
 Username
 Password

Clone destination
 Checkout as a project(s) in the workspace
 Parent directory Browse...
 Clone directory name

Options
 Abort clone when a timeout occurs
 Use pull protocol to copy metadata
 Use uncompressed transfer (fast over LAN)
 Last revision in the cloned repo

< Back Next > Finish Cancel

Mercurial clone repository wizard

Select working directory revision

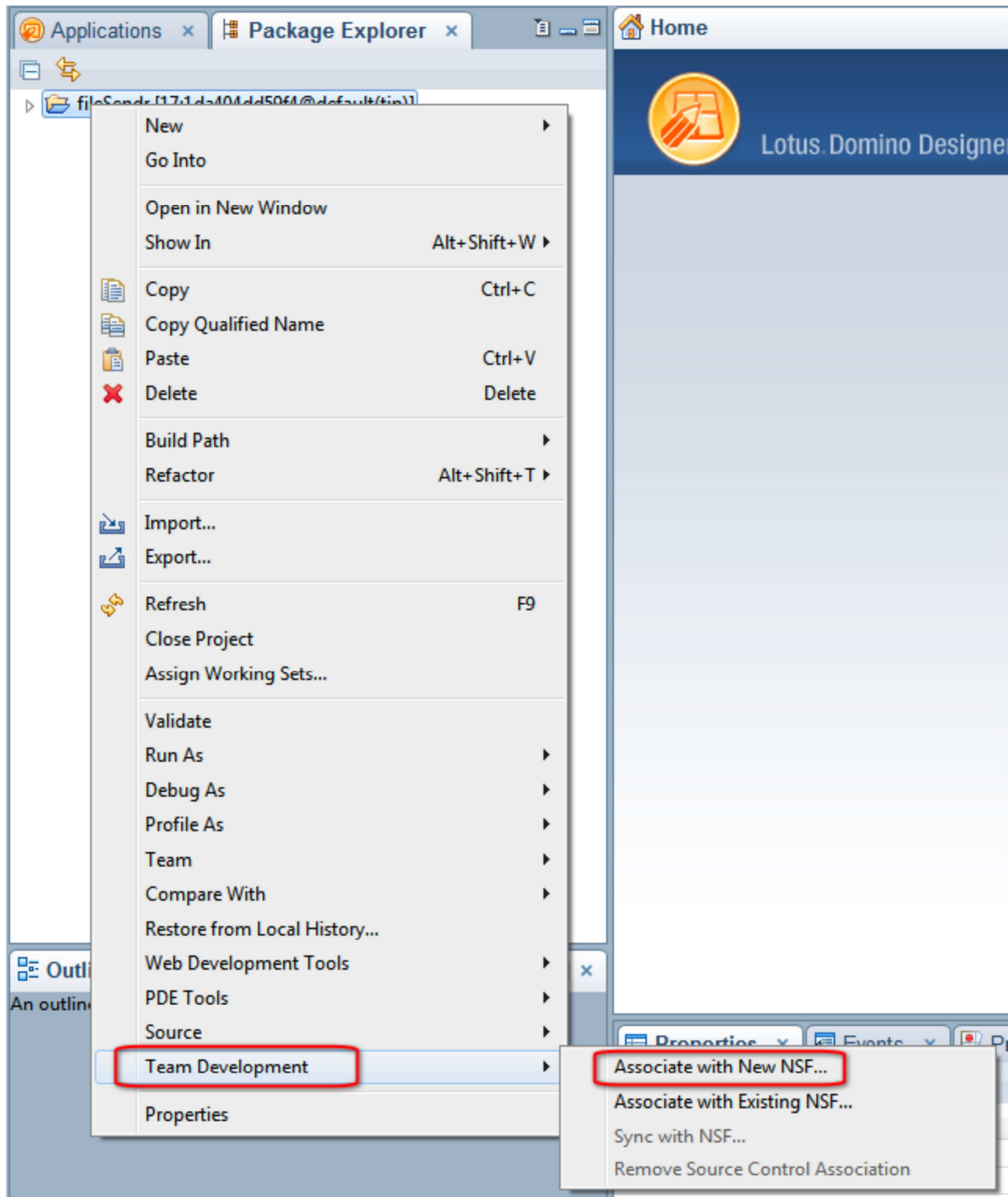
Select the revision to which the working directory will be updated after clone.
 In case nothing is selected, the working directory will be updated to the latest available changeset.

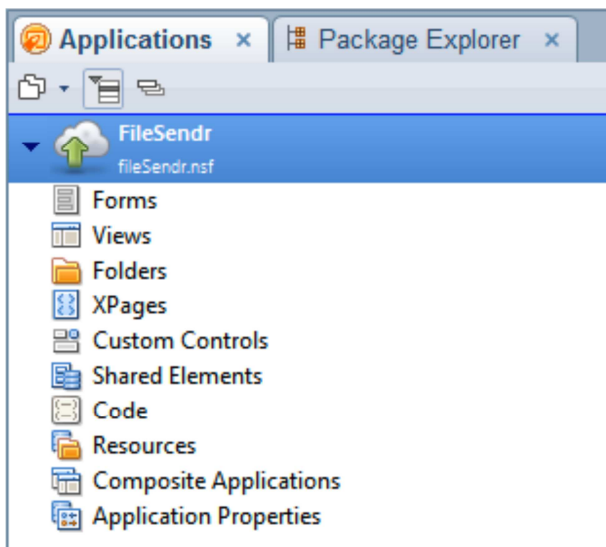
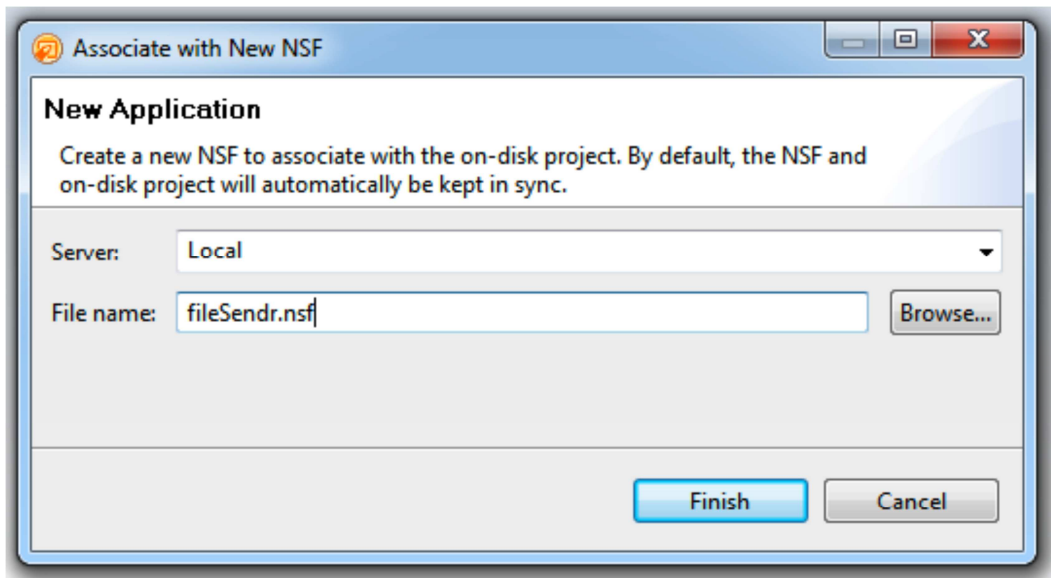
Please enter a valid revision (local, global, tag or branch):

Branches Heads Tags Revisions Bookmarks

Revision	Global	Date	Author	Branch	Tags	Summary
17	1da404dd5...	2011-07-17 16:...	Declan Sci...		tip	Minor text changes
0	c9685fc1df...	2011-01-24 15:...	Declan Sci...			Initial Commit

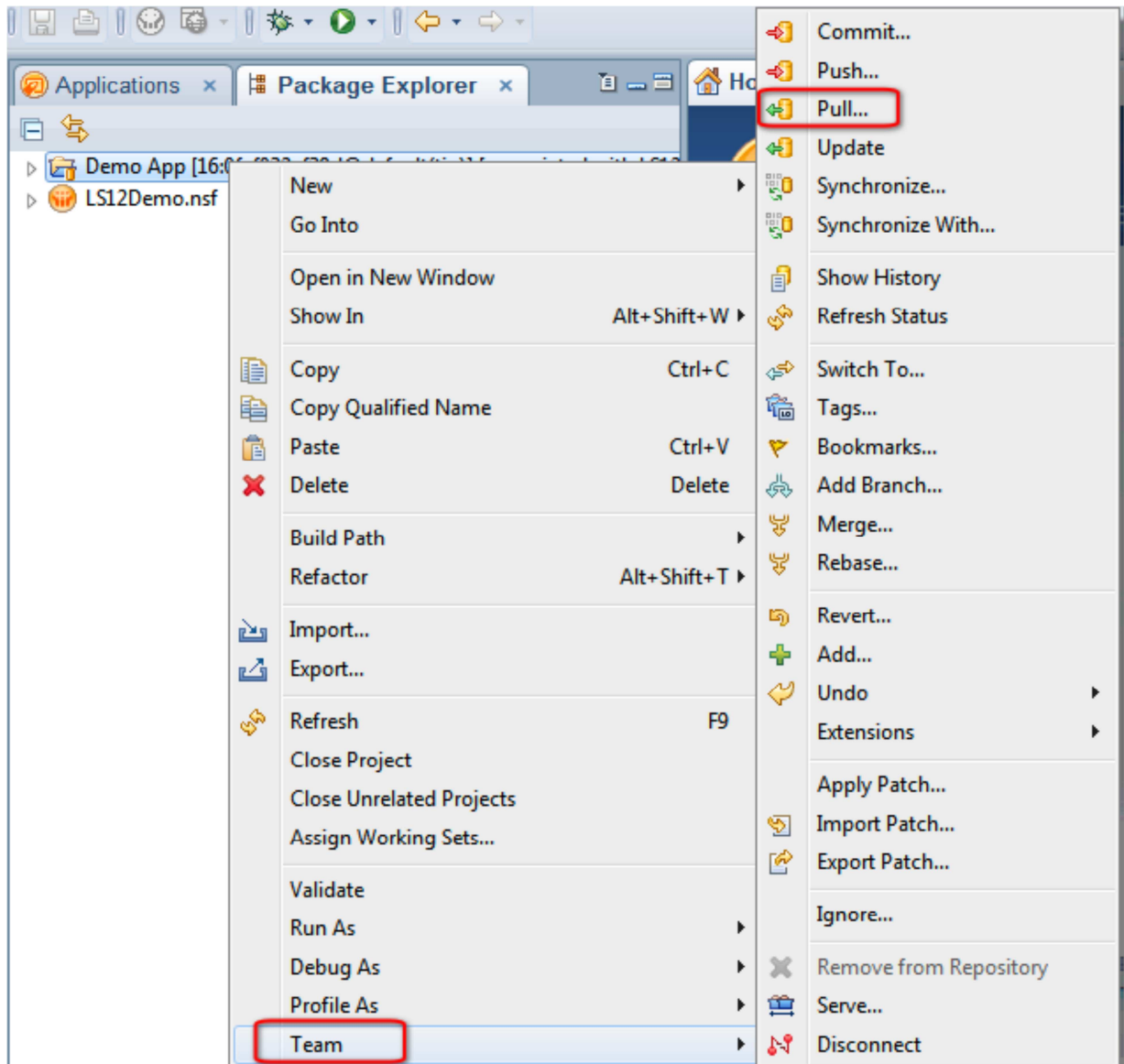
< Back Next > Finish Cancel

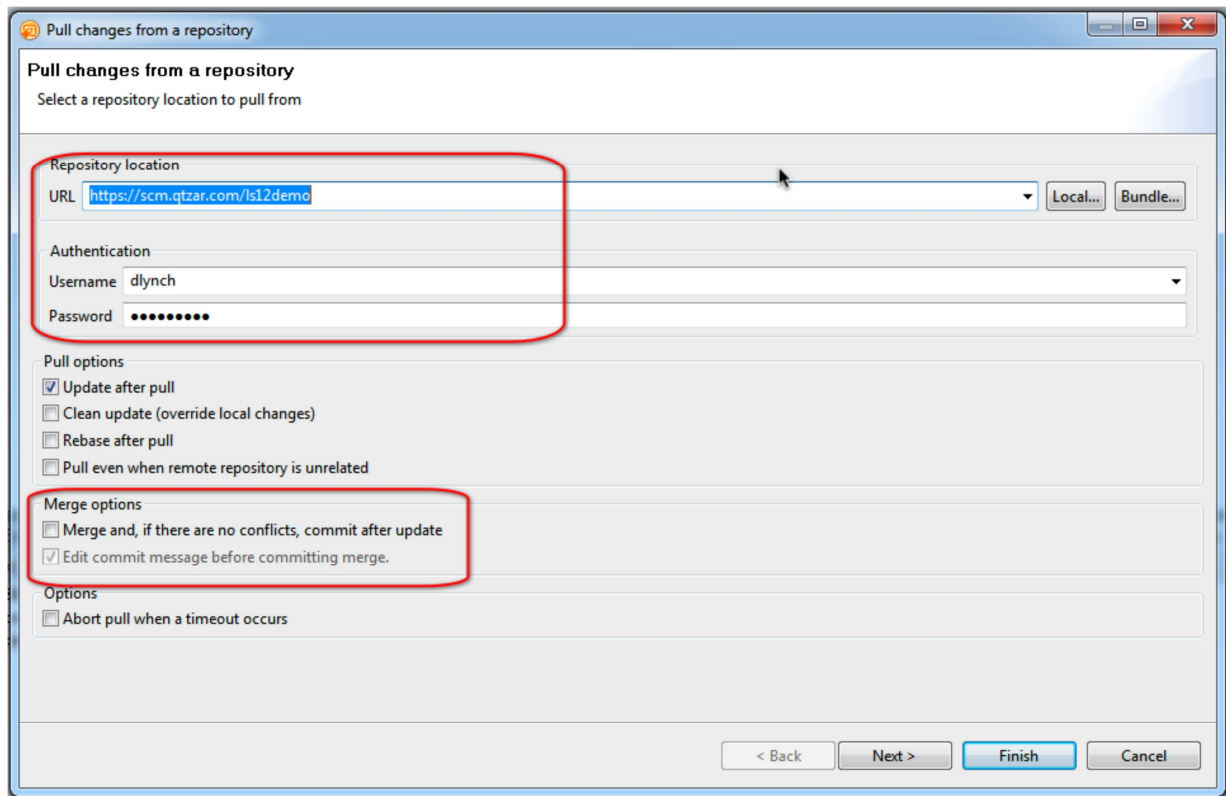




Pulling (Mercurial)

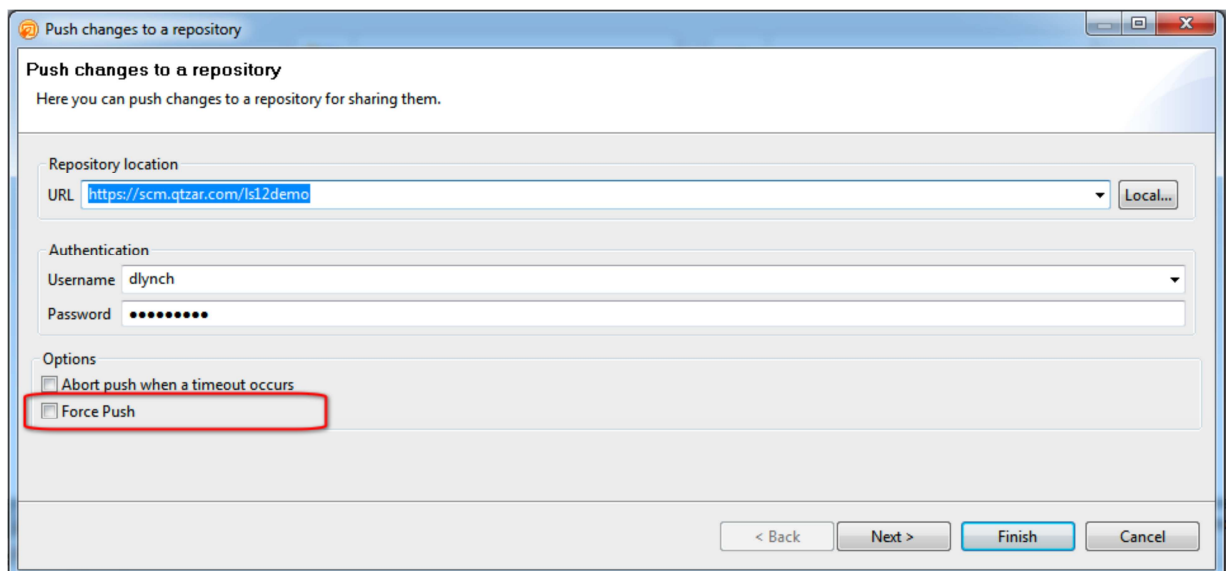
- Pulling updates your repository with Changesets from a remote repository.
- Only Changesets you are missing are transferred.
- As a general rule you should always commit your own changes before pulling in additional Changesets.
- If you pull Changesets into the same branch your currently working in then you should do a merging pull.



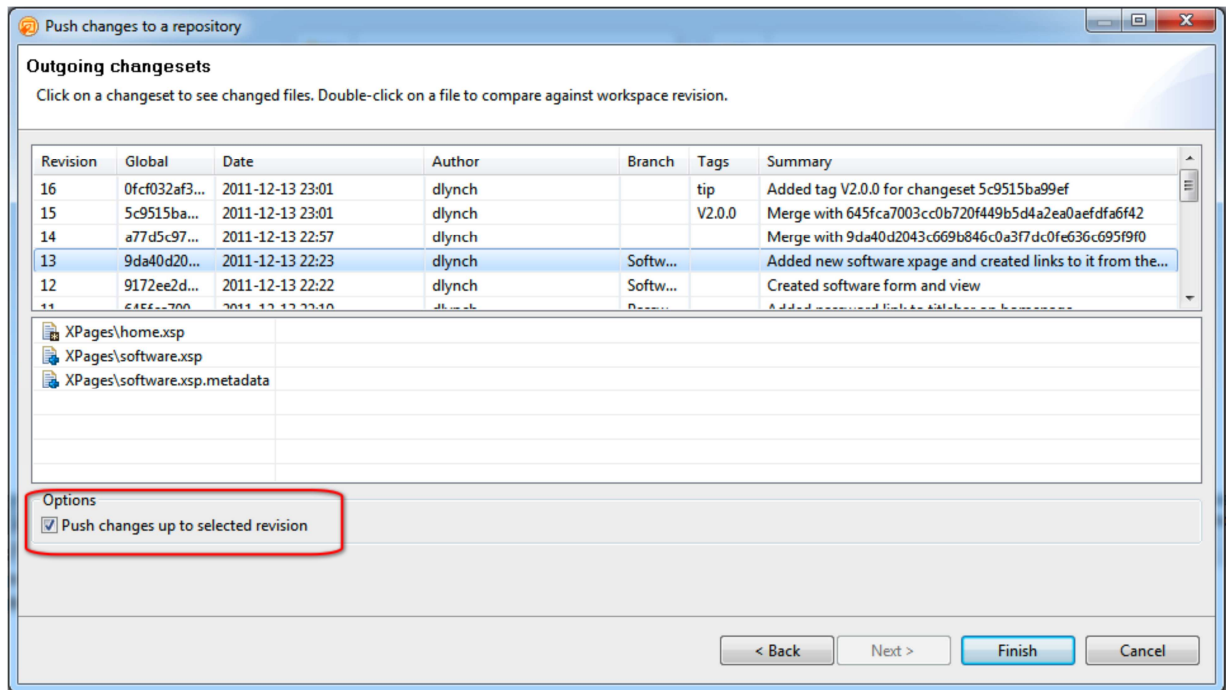


Pushing:

- Pushing will send your changesets to a remote repository.
- If your repository has new branches you may have to select the 'Force' option.

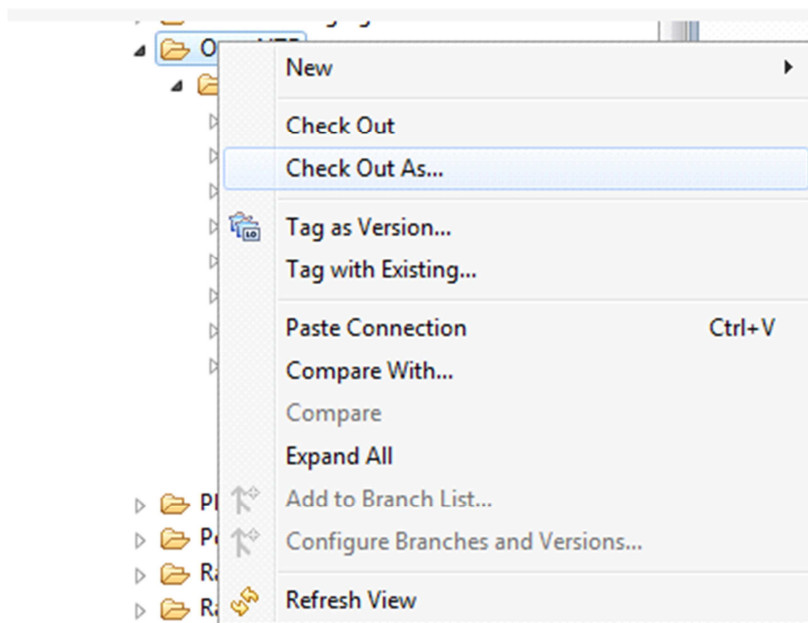


You can also elect to only push up to a certain revision, later revisions will not be pushed to the remote repository:

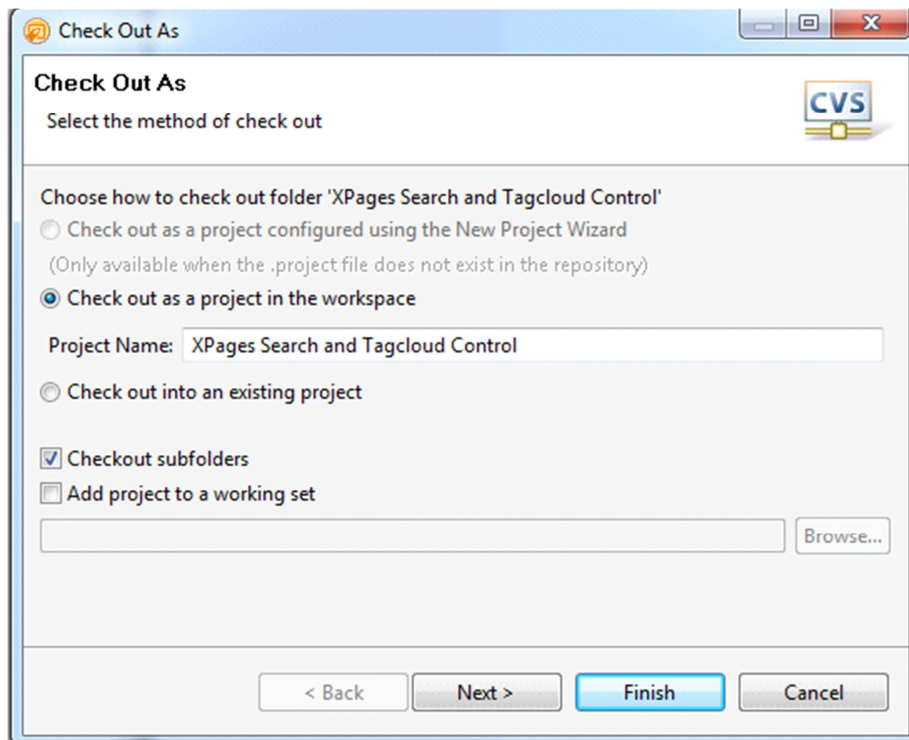


Get data out of source control (CVS)

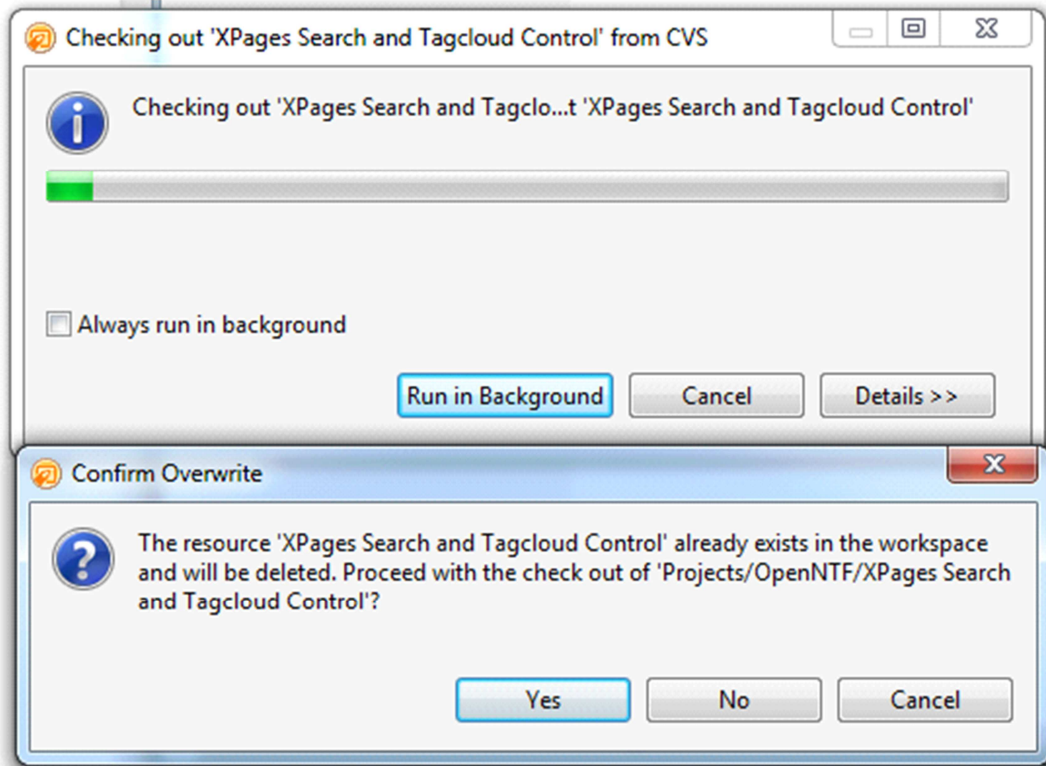
- Open the CVS Repository perspective.
- Right click the project.
- Select 'Check Out As...':



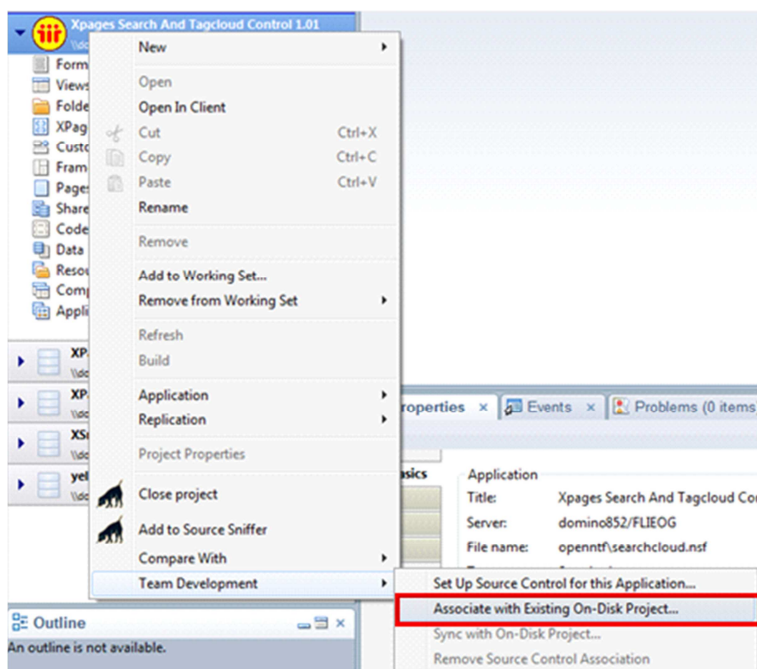
- It will be associated to on on-disk Project.
- If there is no on-disk Project you can also choose the New Project wizard:



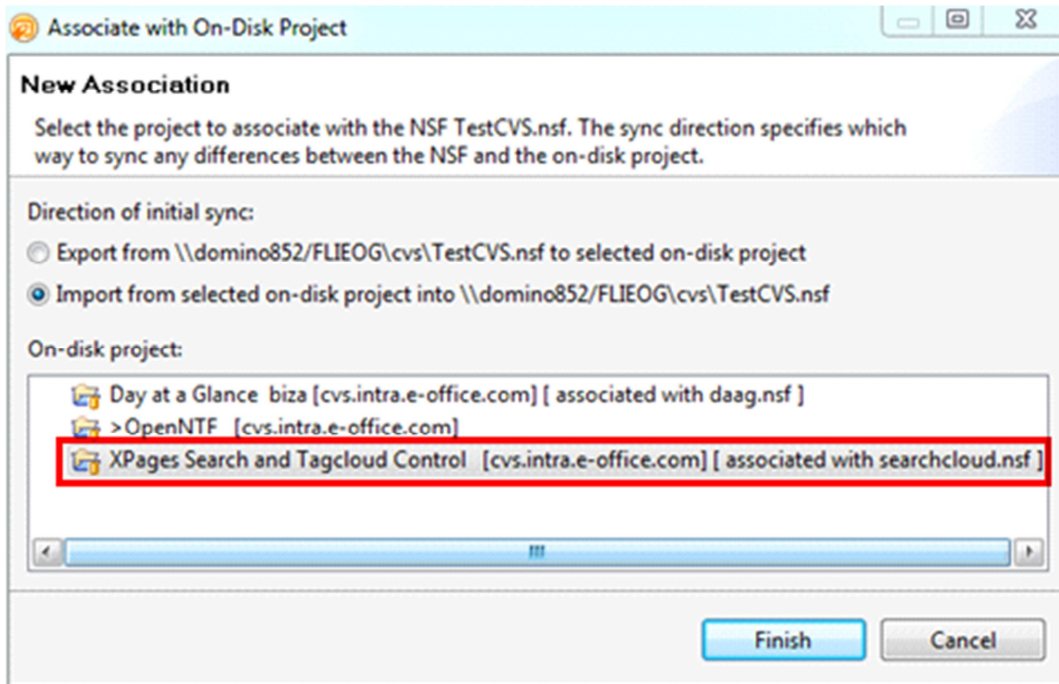
- Click 'Finish'.
- In case an on-disk project already exists, there will be a warning about overwriting the on-disk project:



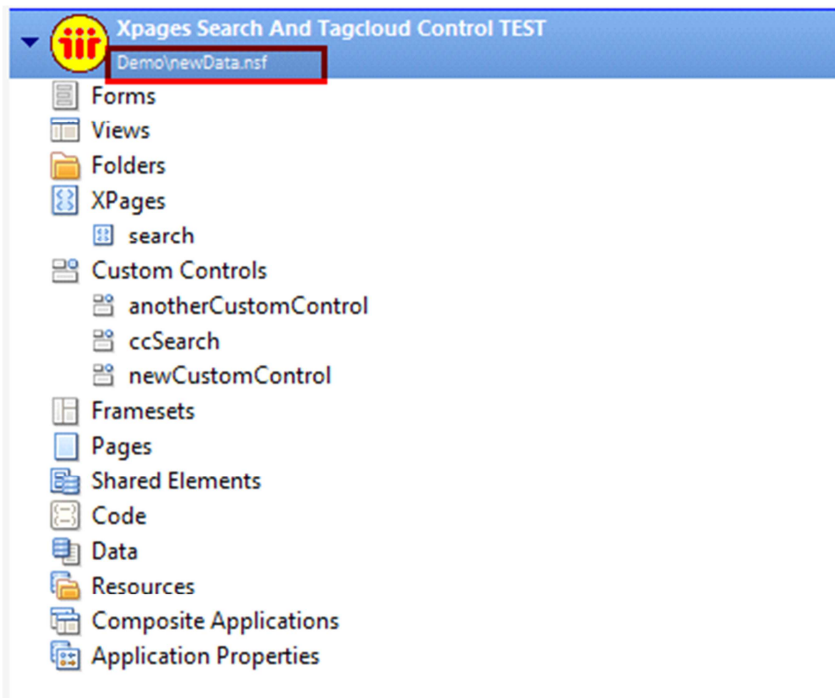
- Choose 'Yes' or 'No'.
- Switch back to the Domino Designer perspective.
- Right click on the application.
- Select 'Team Development → Associate with Existing On-Disk Project':



- o Select the on-disk project to associate the database with:



- o Click 'Finish'.
- o All the design elements are loaded in a new application:



Get data out of source control (eGIT)

In case you need to re-add your on disk projects to and restore the connection with the NSF's below is described is how.

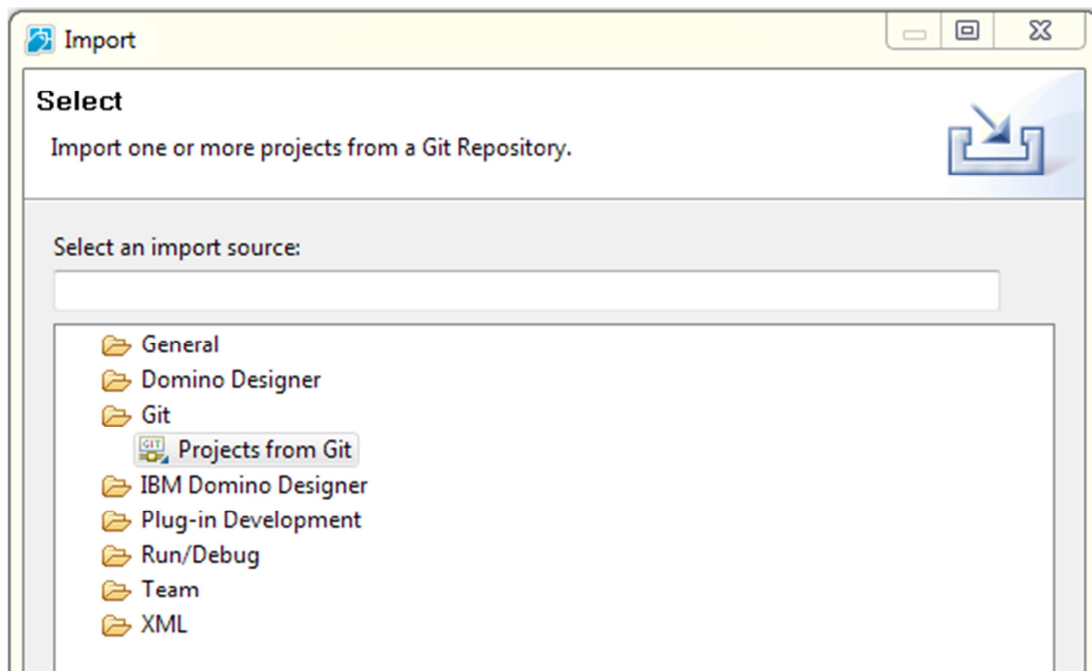
- If needed re-install Designer.
- If needed re-install the eGit plugin.
 - By default, eGit stores the GIT repositories on Windows in a directory called "git" in the users' home folder (c:\users\<>your name>)*. An uninstallation of Designer will not remove that folder, so your on-disk projects are still there.

Note: Add this folder to your backup!

- Add your databases to Designer and organize those using working sets.

You then need to get the Git projects back in Designer:

- Go to the Package Explorer, right click somewhere in it and choose 'Import'.
- Enter "git" in the search dialog and select the 'Projects from Git' option.
- Select the Git repository (folder) that contains your project (or click 'Add' to select the location of the repository on the file system).



- In the next screen, leave the 'Import existing projects' option enabled and select the 'Working Directory' folder.
- Click 'Next' and Designer will show you the on-disk projects it contains.

- Select the ones you would like to import. I would suggest selecting the working set you chose for your database, but for some reason I wasn't able to select it, so I added the Git project to the working set afterwards using the method described here.
- Click 'Finish'.

You're now ready to restore the link from your databases/ applications with the on-disk project:

- Right-click on a database and select 'Team Development > Associate with Existing On-Disk Project'.
 - If you have the latest design sitting in your database, choose the 'Export from database.nsf to selected on-disk project' option.
 - If you want the on-disk project to be leading, choose the 'Import ...' option.
- Select the on-disk project you just imported and hit 'Finish'.
- Done. Now back to work!

Note: It looks like there's a bug the eGit version used in Designer that causes eGit's repository views to sometimes be completely blank. No fear: click the menu icon next to the view and select 'Refresh'.

Note: If you create an on-disk project for a database, all files are stored (by default) in <notes data>\workspace\<project name>. If you then add that local on-disk project to eGit using Team > Share project, the entire folder is moved to the default eGit folder at c:\users\<your name>\git\<new repo name>.

Best practices

- Disciplined approach.
- Commit related changes.
- Commit often.
- Don't commit half-done work. Only commit REAL changed.
- Test code before you commit.
- Write good commit messages.
- Version control is not a backup system (so don't use it for auto-committing on a schedule).
- Use branches.
- Agree on a workflow.

Tips for DDE

1. Always use your own COPY of the database to develop in (using a replica or shared DB causes issues).
 - a. Using a replica or shared DB causes issues
 - b. Your design changes are your own till you pull/merge in other changes.
2. If possible run a local development server and work from that.
 - a. Really speeds up Domino Designer.
3. (Never work directly against your production environment).

Steps (hand-on)

TBD

- Create application
- Create on-disk project
- Share on-disk project (initial commit)
- Committing additional changes
- Editing source code in the on-disk project (outside of DDE)
- Adding source code to the on-disk project (outside of DDE)

Appendix A – List of Terms

DDE	Domino Designer on Eclipse
NSF	Notes Structured File, physical file where design and document notes are stored.
SCM	Source Code Management