

Source Management Tool Client Environment Operation

Summary

The e- government standard framework uses the open- source Subversion(SVN) as the source management tool for the development environment. Similar to other source management tools, there are various clients that provide GUI so that the user can use it on their PCs in addition to the direct use through the system console.

Various Subversion clients were reviewed and tested through the e- government standard framework construction project and as a result the Subversive Eclipse IDE plug- in which showed better functions and more stable operation than other SVN clients has been included and provided in the development environment of the e- government standard framework.

As of May 2009, the installation version of Subversive plug- in provided as the e- government standard framework development environment is as follows.

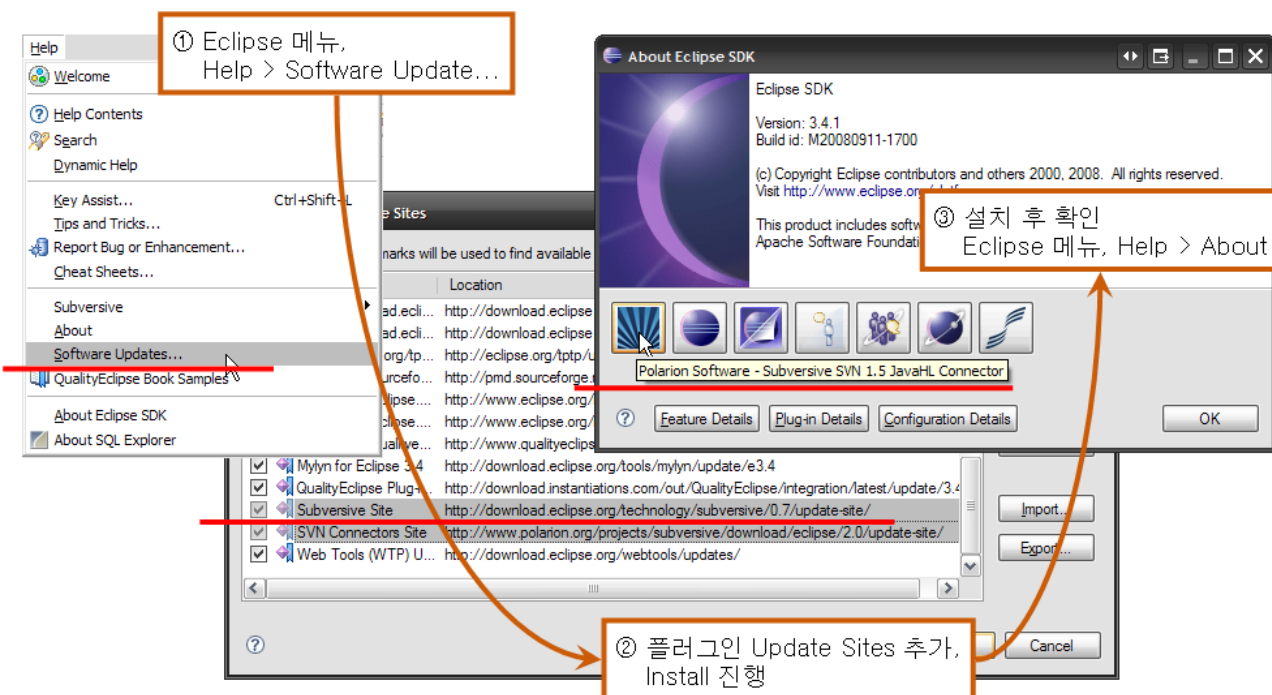
- Polaron Subversive SVN Connectors 2.1.0

This Subversive version may be revised when distributed in the future, due to the improvement efforts of e- government standard framework development environment.

Installation

The version check and update for Subversive included in the e- government standard framework development environment is carried out in the following sequence.

- Check the Subversive Eclipse plug- in version
 1. In the Eclipse IDE menu bar, select **Help > About**.
 2. Put the mouse pointer on  icon to check the version.
- Subversive Eclipse plug- in update
 1. In the Eclipse IDE menu bar, select **Help > Software Update....**
 2. On the popup window, select **Available Software** tab.
 3. Find Subversive Site in the list and check the checkbox.
 4. Click **Install...** button on the right top to carry out the update.



Main Concept

Using the Subversive Eclipse IDE plug-in is much similar to using the CVS client provided in Eclipse IDE by default. Its usage and concept are as follows.

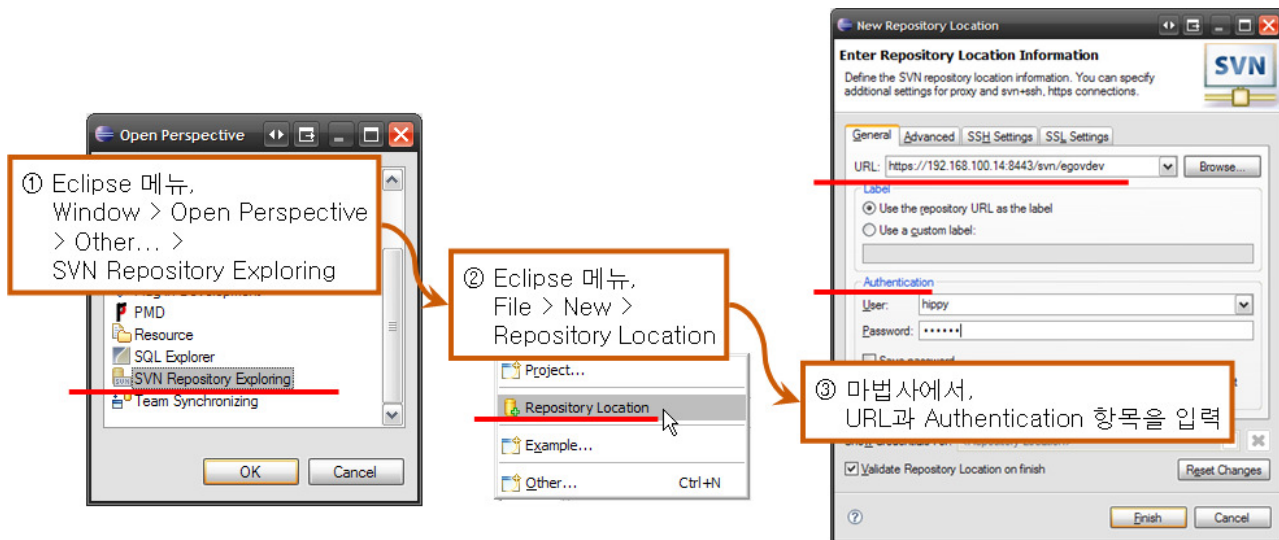
Repository

SVN's repository is characterized by the followings.

- All project program sources and their revisions are stored.
- Access by multiple people via a network
- SVN uses Berkeley DB(Default) or file system(Optional) as a repository

Creating (connecting) the repository information in the user's PC

1. For the convenience, convert Eclipse Perspective to SVN exploration.
 - In Eclipse IDE menu, select **Window > Open Perspective > Other... > SVN Repository Exploring**
2. In Eclipse IDE menu, select **File > New > Repository Location**.
3. Enter URL for repository access and authentication information to create repository information.
 - In the first tap, in '**URL**', enter the project SVN repository URL
 - Under Authentication, in **User** and **Password**, enter the user's SVN repository access ID and Password
 - Click **Finish** button to finish the procedure.

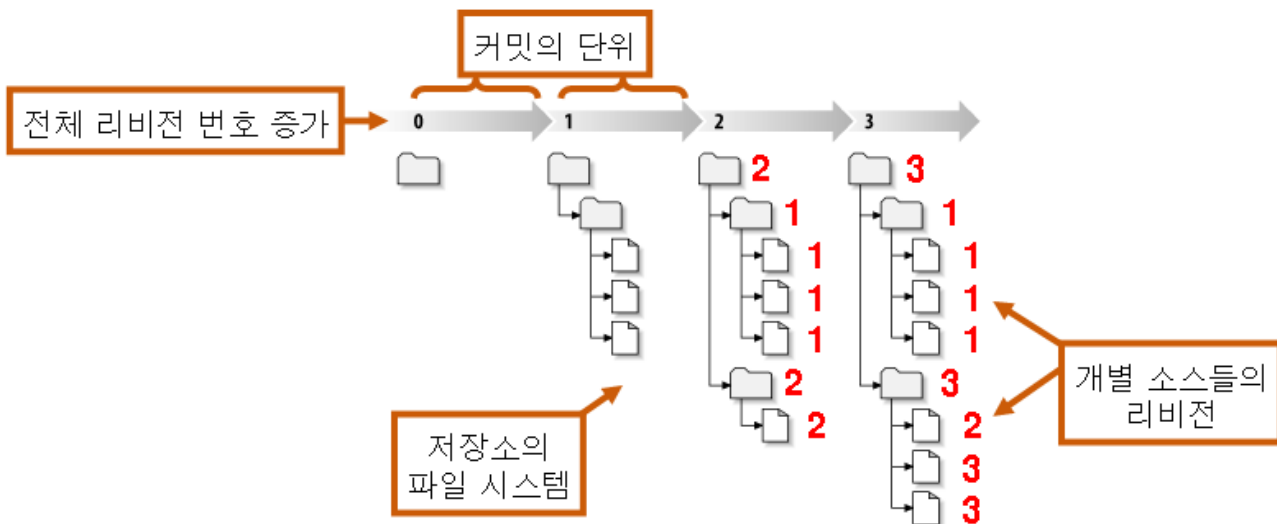


Revisions

The revisions created by Subversion through the source revision management procedure are characterized by the followings.

- Modify and committing source files will increase the number based on a certain rule.
- These are regarded as file versions of the ones stored in the repository.
- For SVN, revisions are numbered for files, but for revision occurrence unit.
- The project progress can be understood from the revisions.

You can understand how revision is numbered in Subversion in the following example of file system structure change.



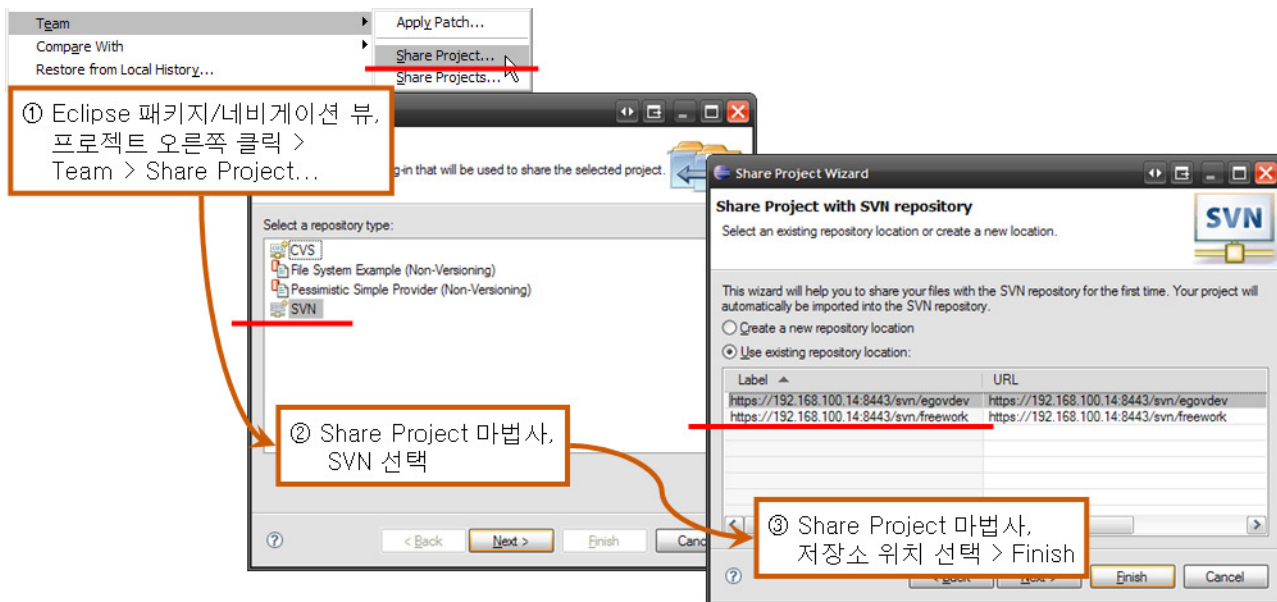
Source Sharing

You can share the first project/source or new project/source with other users as in the following.

Project Sharing

This is a procedure to add a new project to the repository.

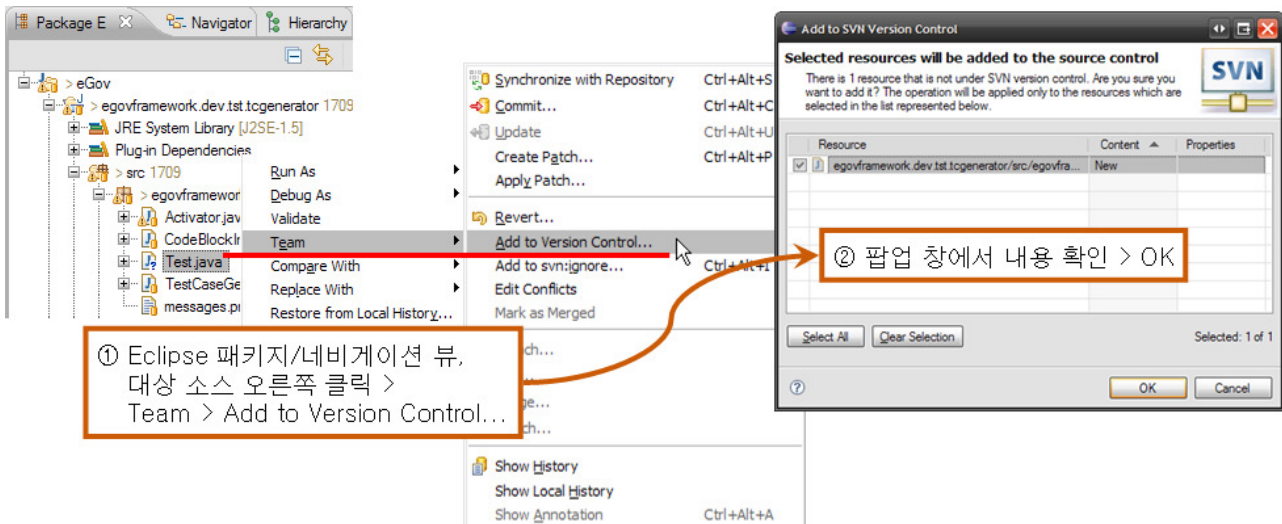
1. In Eclipse IDE package or the navigation view, select the project you want to add to the repository.
2. Right- click the project and then select **Team > Share Project...**
3. The Share Project wizard appears in a popup window. In the list, select **SVN** and click **Next** button.
4. Select '**Use existing repository location**' radio button.
5. In the list at the bottom, select the repository URL that was previously created (connected) and click **Finish** button.



Source Unit Sharing

This is a procedure to add new sources created in a shared project to the repository..

1. In Eclipse IDE package or navigation view, select the source to share (add to the repository).
2. Right click the mouse and then select **Team > Add to Version Control...**
3. A popup window 'Add to SVN Version Control' appears. Check the source to add/share to the repository and click **OK** button.



① Eclipse 패키지/네비게이션 뷰, 대상 소스 오른쪽 클릭 > Team > Add to Version Control...

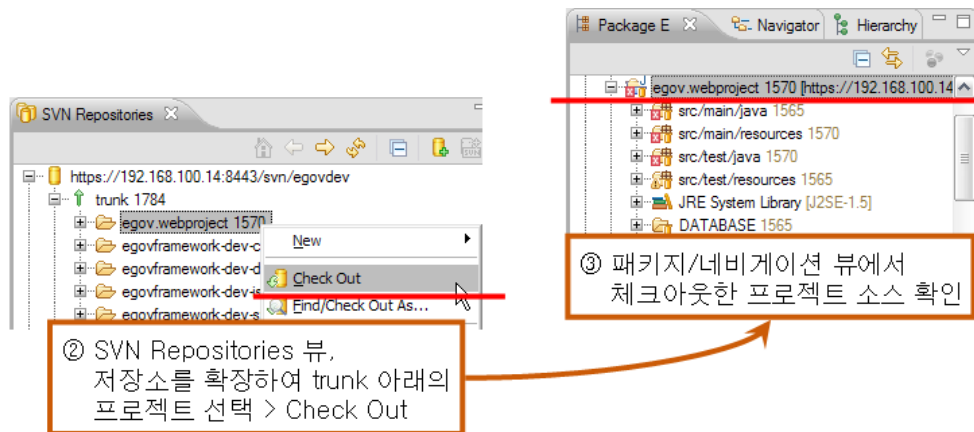
② 팝업 창에서 내용 확인 > OK

Checkout

- To receive the final revisions of all sources from the repository.
- According to the repository authentication policy, whether to enter ID and Password will be decided.

In general, this is carried out once at the beginning of project development, and sometimes when to receive the whole sources during development. Carry out in the following sequence.

1. For the convenience, convert Eclipse Perspective to SVN exploration.
 - In Eclipse IDE menu, select **Window > Open Perspective > Other... > SVN Repository Exploring**.
2. In SVN Repositories view, expand the repository used the project and, under the trunk, select a desired project.
3. Right- click the project and then select **Check Out**.
4. In the package view or navigation view, check the project sources checked out from the repository to the user's local.



② SVN Repositories 뷰, 저장소를 확장하여 trunk 아래의 프로젝트 선택 > Check Out

③ 패키지/네비게이션 뷰에서 체크아웃한 프로젝트 소스 확인

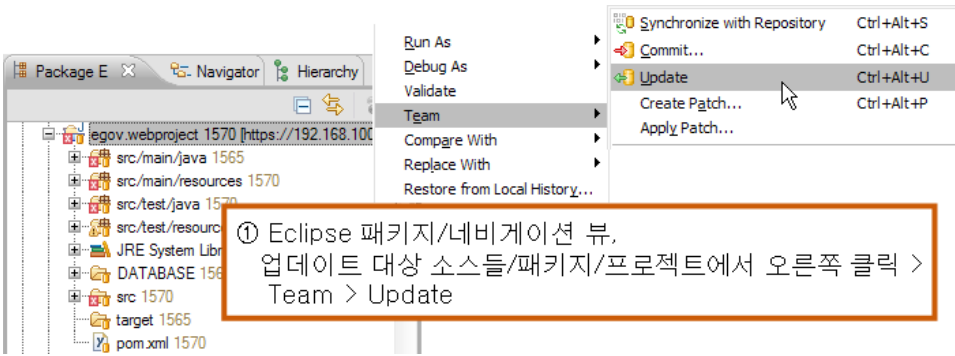
① Eclipse 메뉴, Window > Open Perspective > Other... > SVN Repository Exploring

Update

- After checking out, check the source changes made by others
- Compare the sources in the repository with the local copy and get the latest version of changed sources.
- The latest version of changed sources is integrated with the local copy

Carry out the update as in the following.

1. In Eclipse IDE package view and or navigation view, select the project to update.
2. Right click the project and then select **Team > Update**.

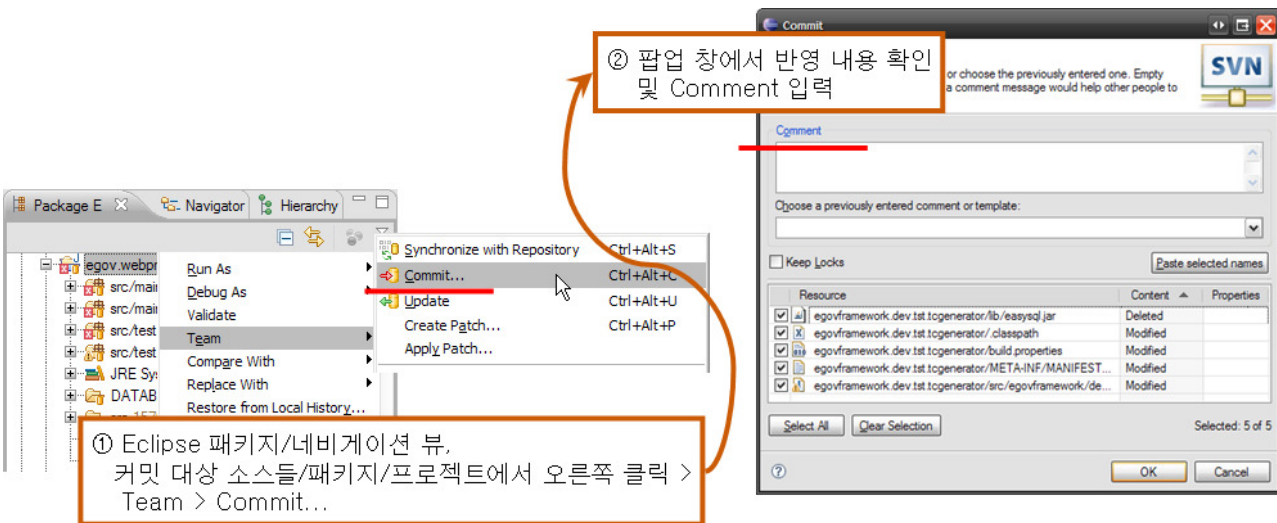


Commit

- To modify, add files to, or delete a source checked out to the local and then store it in the repository for update
- When commit is done, the whole revision increases by 1(for CVS, the revision increases for each modified file)

Carry out the commit as in the following.

1. In Eclipse IDE package view and or navigation view, select sources or a package or project to commit.
2. Right- click them (it) and then select **Team > Commit...**
3. In the popup window, check the results. Enter your comment and click **OK** button.



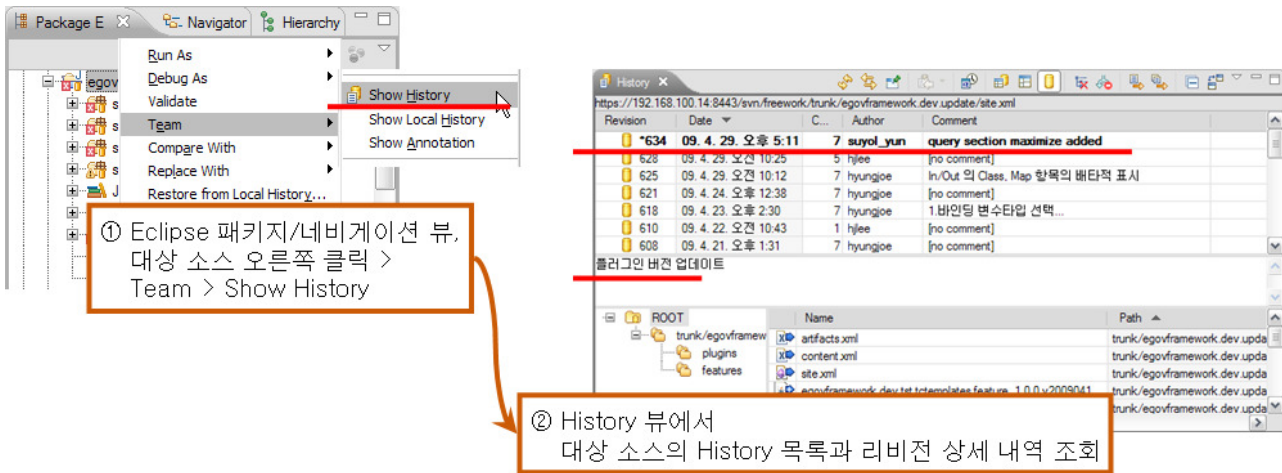
History

- Details of all previous revisions related with certain sources managed by SVN.

History Inquiry

You can make an inquiry through '**Team**'s '**History**' view, the base category of Eclipse IDE, in the following sequence.

1. In Eclipse IDE package view and or navigation view, select a source to check the history for.
2. Right click it and then select **Team > Show History**.
3. In History view, check the retrieved history details.

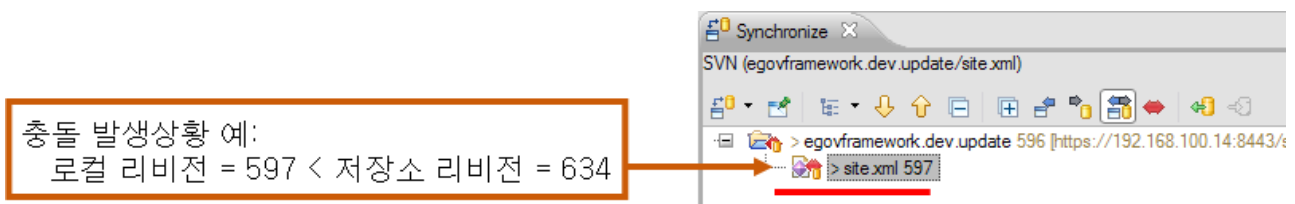


Conflict

- When committing a modified source to the repository, a conflict occurs if the revision in the repository is higher (already modified and committed by another).

The following figure is an example of conflict found in Synchronize view:

As shown in the figure, when the local revision of a source (site.xml in the figure) is 597 but the revision in the repository is higher, a conflict occurs. In other words, while a user (User A) is modifying a source whose revision is 597, which was received from the repository, if another user (User B) changes and commits the source (revision increases) and then User A tries to commit the revision 597, a conflict occurs.

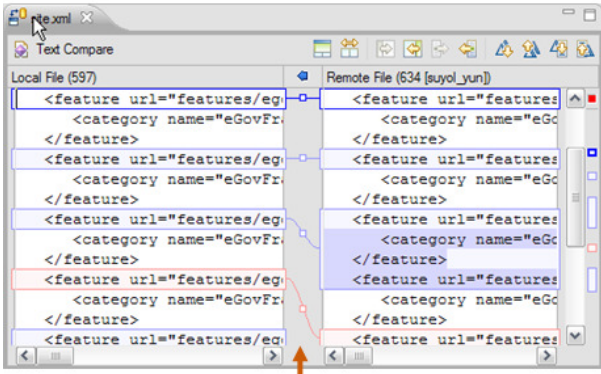


Troubleshooting

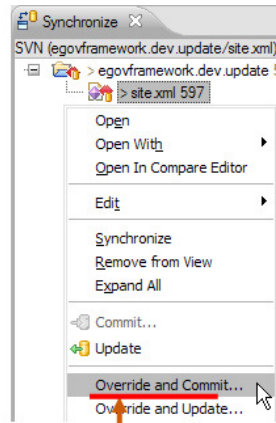
You can troubleshoot conflict by doing the followings:

- postpone: Continue to modify source codes without immediate incorporation. Retry when modification is completed(no troubleshooting)
- diff: Analyze the conflict. Identify the difference of sources between local and repository
- edit: Open and modify the local file
- mine- full: Ignore the revision in the repository and commit the local one.
- theirs- full: Ignore the modifications of the local file and use the revision in the repository.

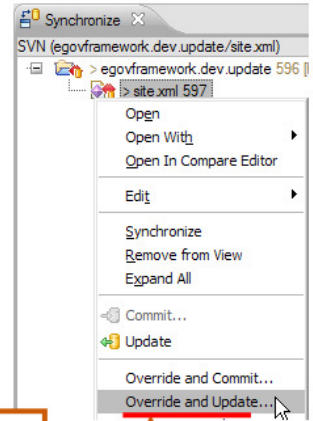
The following figures are the sample of those troubleshooting.



diff/edit:
로컬과 저장소의 소스를
모두 열어 비교/편집



mine-full:
충돌이 발생한 파일에서 오른쪽 클릭 >
Override and Commit... 선택



theirs-full:
충돌이 발생한 파일에서 오른쪽 클릭 >
Override and Update... 선택

Reference

- Subversion open source project site: <http://subversion.tigris.org/>
- Subversion online help: <http://svnbook.red-bean.com/>