CM & RE Tool

T3

Marvin Habermann

오희수

정훈섭

진경훈

1. Configuration Management Tool
2. Requirement Engineering Tool

1. Configuration Management Tool

1.1 What is CM & SCM?

1.2 Why we need it?

1.3 SCM process

1.4 SCM tools & Subclipse

1.5 Benefit of SCM

2. Requirement Engineering Tool

1. Configuration Management Tool

2. Requirement Engineering Tool

2.1 What is Requirement?

2.2 Requirement Management
Requirement Engineering

2.3 Why we need it

2.4 RE proccess

2.5 RE tools & JFeature

2.6 Benefit of RE

CM tool

What is CM & SCM?

Configuration is the functional and physical characteristics of hardware and software as set forth in technical documentation or achieved in a product

What is CM & SCM?

Configuration management is unique identification, controlled storage, change control and status reporting of selected intermediate work products, product components and products during the life of a system.

What is CM & SCM?

Software Configuration management is the task of tracking and controlling changes in the software throughout its life cycle.

Why we need it?

Lack of Visibility

Difficulty of Control

Difficult to track

Inconsistency

SCM process

Identify Configuration items

Control Configuration items

Perform Configuration audits

Establish CM Records

SCM process

Identify Configuration items

Control Configuration items

Perform Configuration audits

Establish CM Records

：Start of the SCM.

：Identify system components which should be managed in SCM process.

SCM process

Identify Configuration items

Control Configuration items

Perform Configuration audits

Establish CM Records

：review the changes and control them to apply baseline.

SCM process

Identify Configuration items

Control Configuration items

Perform Configuration audits

Establish CM Records

：check the integrity whether changes are applied properly

SCM process

Identify Configuration items

Control Configuration items

Perform Configuration audits

Establish CM Records

：record all changes during software life cycle in documents.

Terms used in SCM

Version is initial release or re-release of software configuration item, associated with a complete compilation or recompilation of the computer software configuration item

Revision indicate particular points in the s/w development cycle when there are changes.

Terms used in SCM

Baseline is a specification or product that has been formally reviewed an agreed upon, that thereafter serves as the basis for further development, and that can be changed only through formal change control procedures.

SCM tools

Subversion

Subclipse

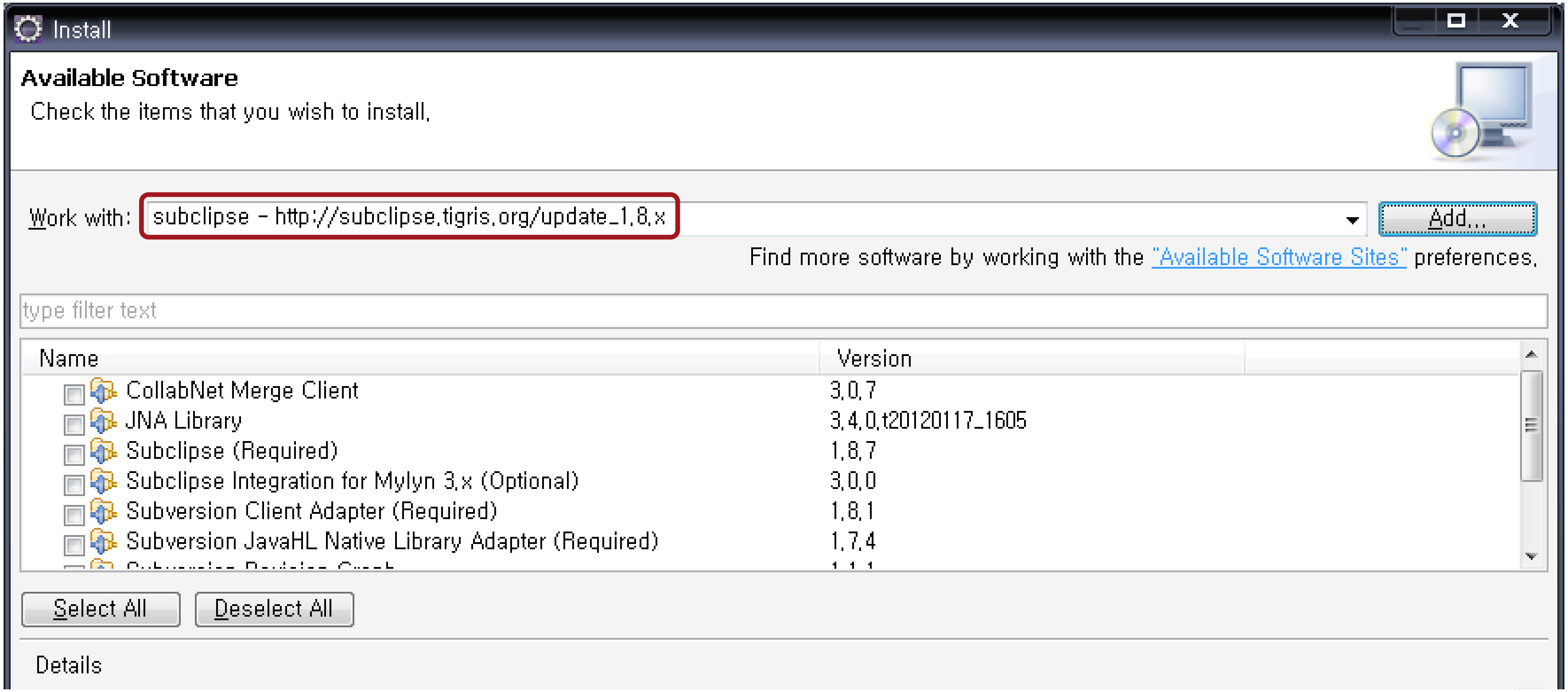
TortoiseSVN / RabbitCVS

Github

Redmine

Trac

Subclipse



Available Software

Check the items that you wish to install.

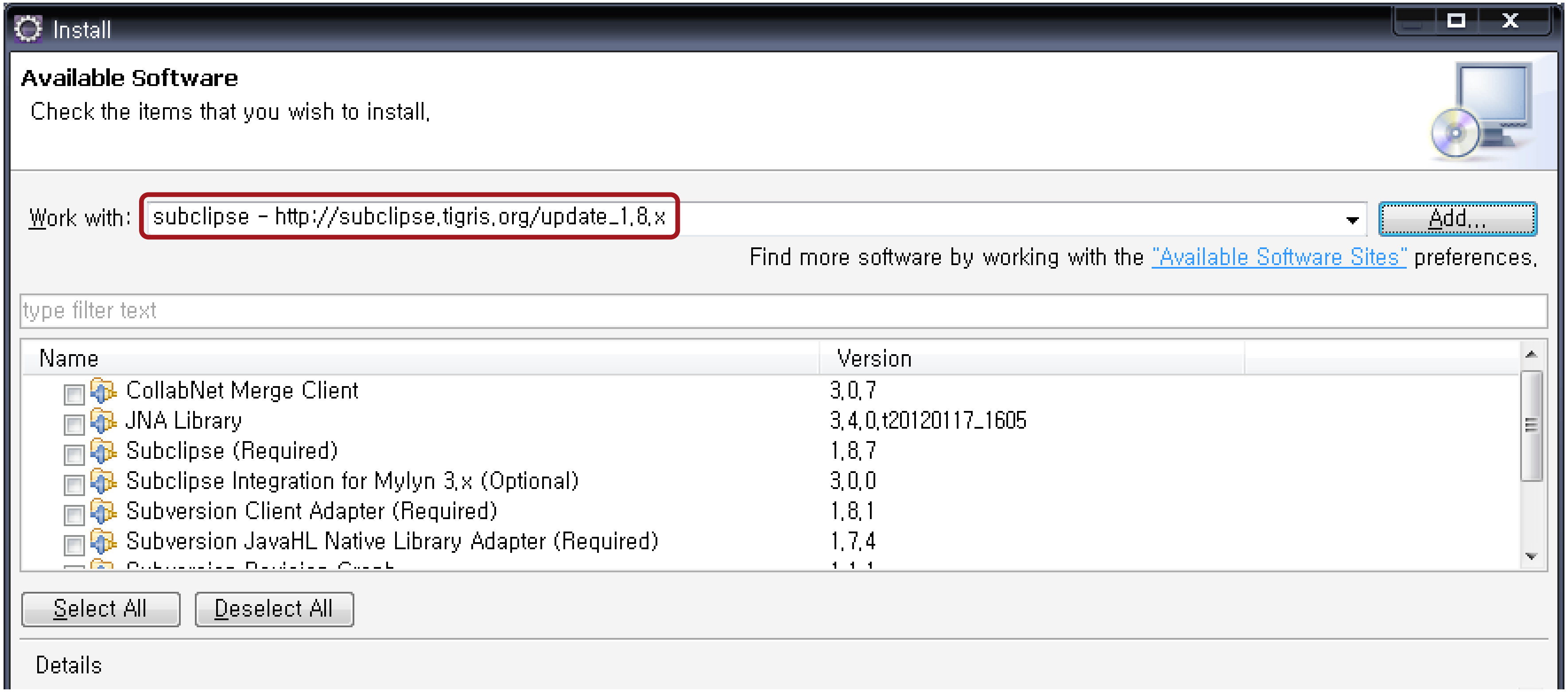
Work with: subclipse - http://subclipse.tigris.org/update_1.8.x

Find more software by working with the "Available Software Sites" preferences.

type filter text

	Name	Version
□	CollabNet Merge Client	3.0.7
□	JNA Library	3.4.0.t20120117_1605
□	Subclipse (Required)	1.8.7
□	Subclipse Integration for Mylyn 3.x (Optional)	3.0.0
□	Subversion Client Adapter (Required)	1.8.1
□	Subversion JavaHL Native Library Adapter (Required)	1.7.4
□	Subversion Revision Graph	1.1.1

Subclipse



Available Software

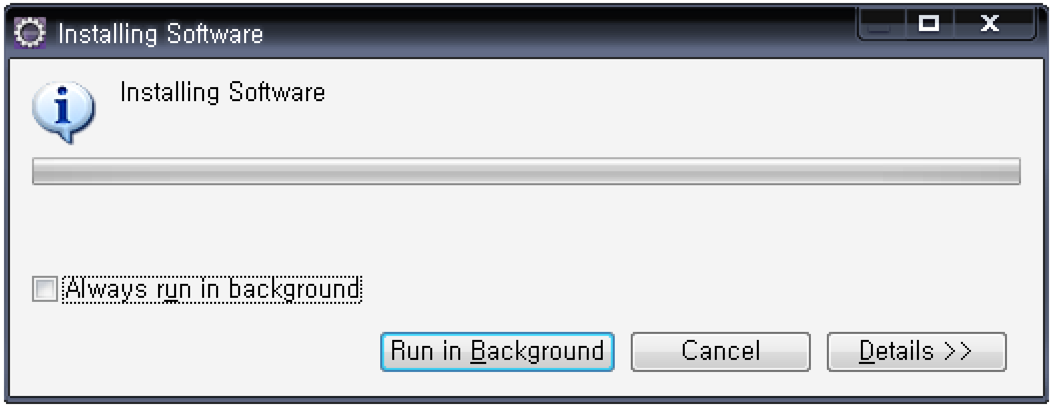
Check the items that you wish to install.

Work with: subclipse - http://subclipse.tigris.org/update_1.8.x

Find more software by working with the "Available Software Sites" preferences.

Name	Version
□ CollabNet Merge Client	3.0.7
□ JNA Library	3.4.0.t20120117_1605
□ Subclipse (Required)	1.8.7
□ Subclipse Integration for Mylyn 3.x (Optional)	3.0.0
□ Subversion Client Adapter (Required)	1.8.1
□ Subversion JavaHL Native Library Adapter (Required)	1.7.4
□ Subversion Revision Graph	1.1.1

Subclipse

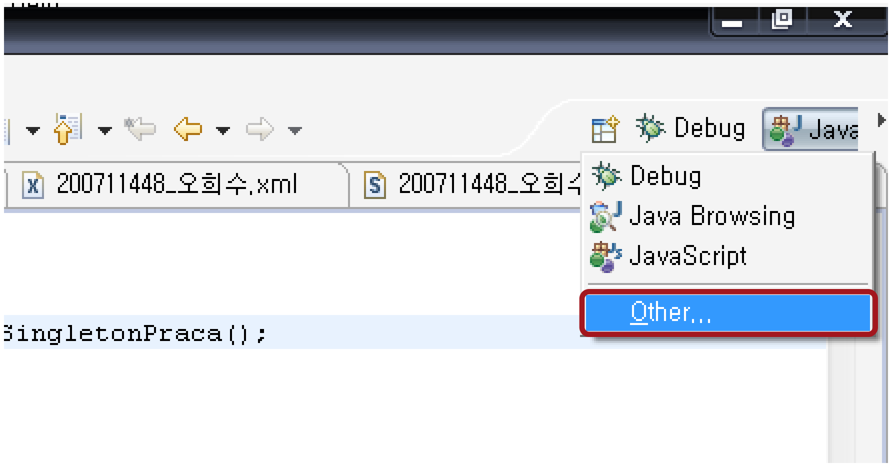


Installing Software

Installing Software

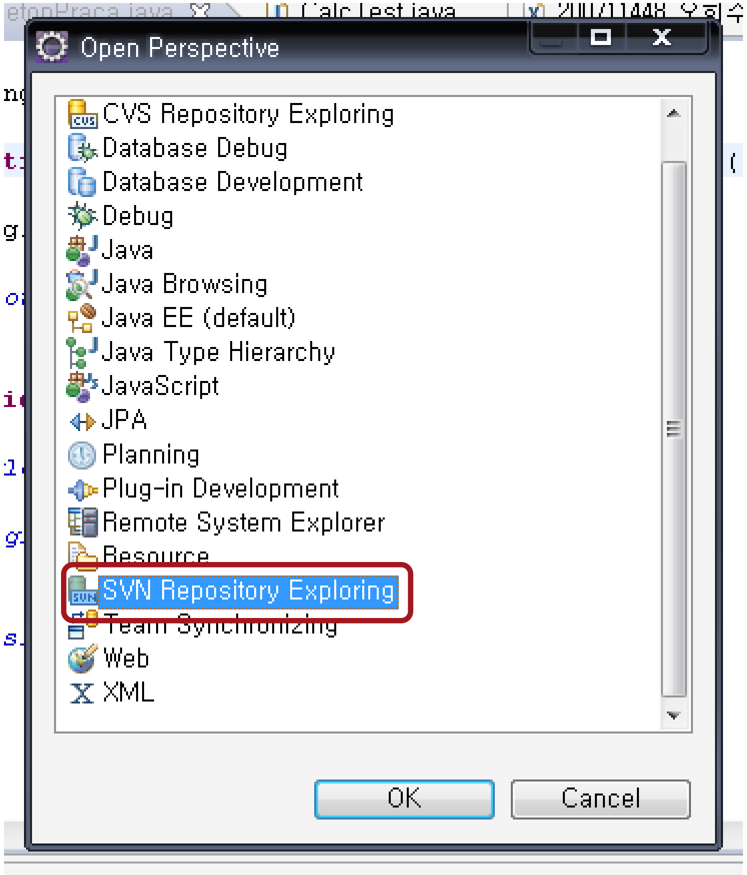
□ Always run in background

Subclipse



```
SingletonPraca();
```

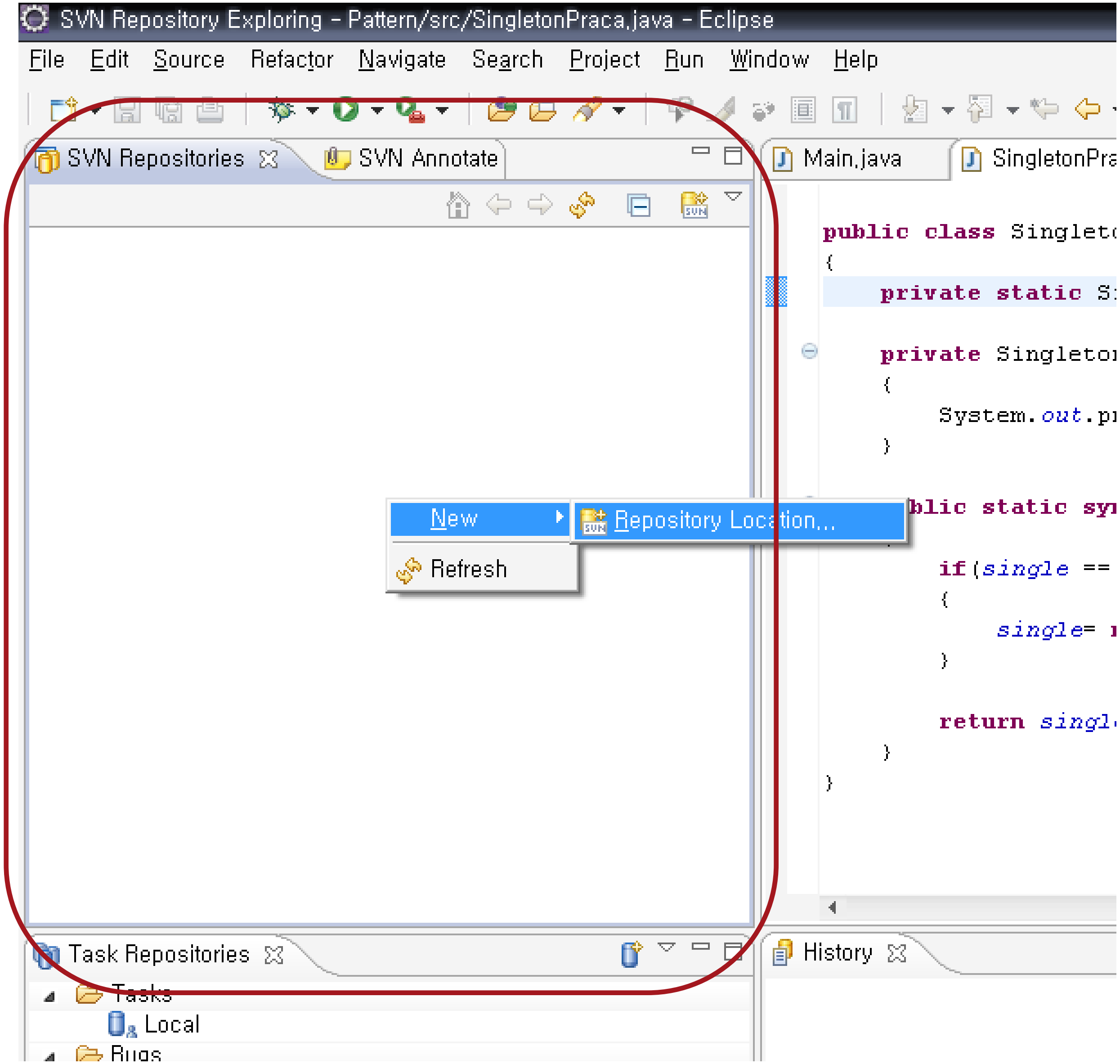
Subclipse



Open Perspective

CVS Repository Exploring
Database Debug
Database Development
Debug
Java
Java Browsing
Java EE (default)
Java Type Hierarchy
JavaScript
JPA
Planning
Plug-in Development
Remote System Explorer
Resource
SVN Repository Exploring
Team Synchronizing
Web
XML

Subclipse



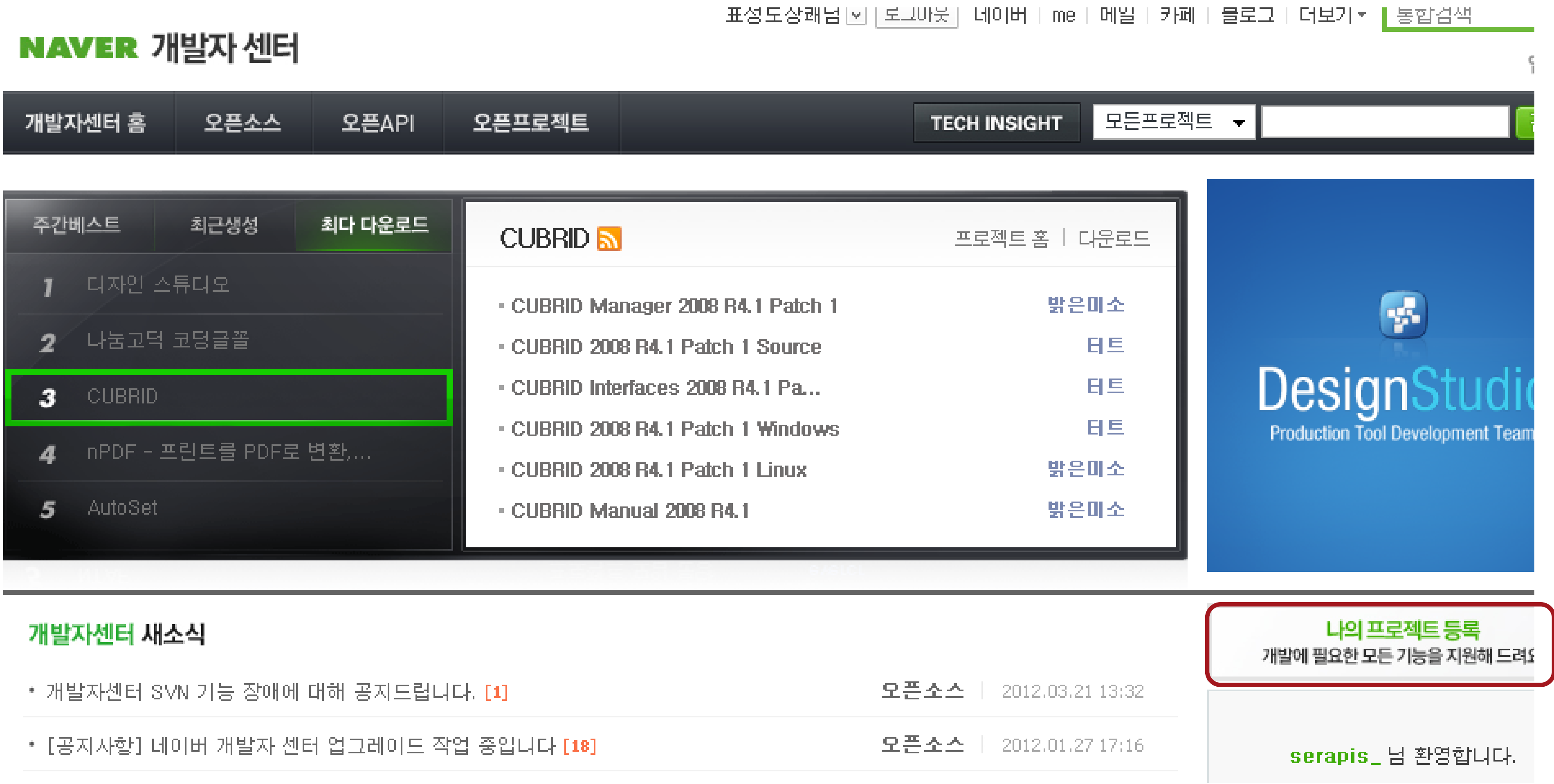
```
public class Singlet
{
    private static S

    private Singleto
    {
        System.out.p
    }

   blic static sy

        if(single ==
        {
            single= 
        }

        return singl
    }
}
```

Subclipse

Need to set up the server

Should get license(Enterprise Edition)

Subclipse



1 디자인 스튜디오
2 나눔고딕 코딩글꼴
3 CUBRID
4 nPDF - 프린트를 PDF로 변환,...
5 AutoSet

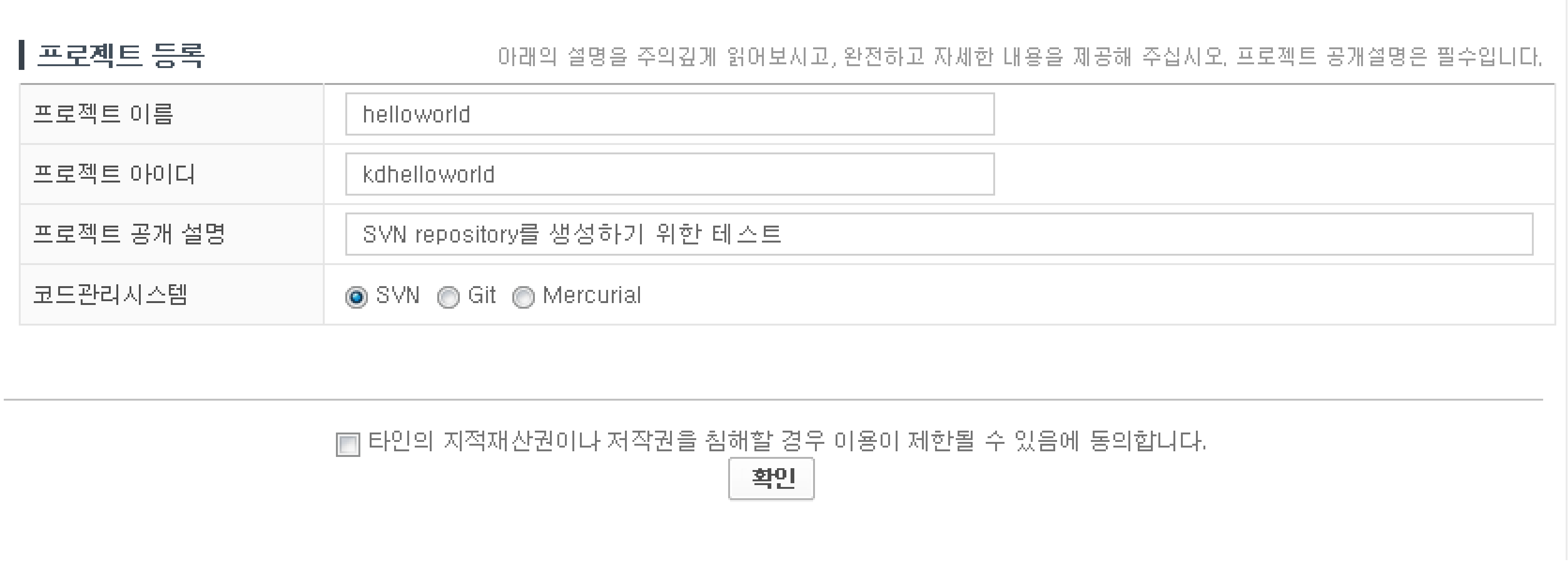
CUBRID Manager 2008 R4.1 Patch 1	밝은미소
CUBRID 2008 R4.1 Patch 1 Source	티트
CUBRID Interfaces 2008 R4.1 Pa...	티트
CUBRID 2008 R4.1 Patch 1 Windows	티트
CUBRID 2008 R4.1 Patch 1 Linux	밝은미소
CUBRID Manual 2008 R4.1	밝은미소

개발자센터 새소식

개발자센터 SVN 기능 장애에 대해 공지드립니다. [1]	오픈소스	2012.03.21 13:32
[공지사항] 네이버 개발자 센터 업그레이드 작업 중입니다 [18]	오픈소스	2012.01.27 17:16

serapis_ 님 환영합니다.

Subclipse



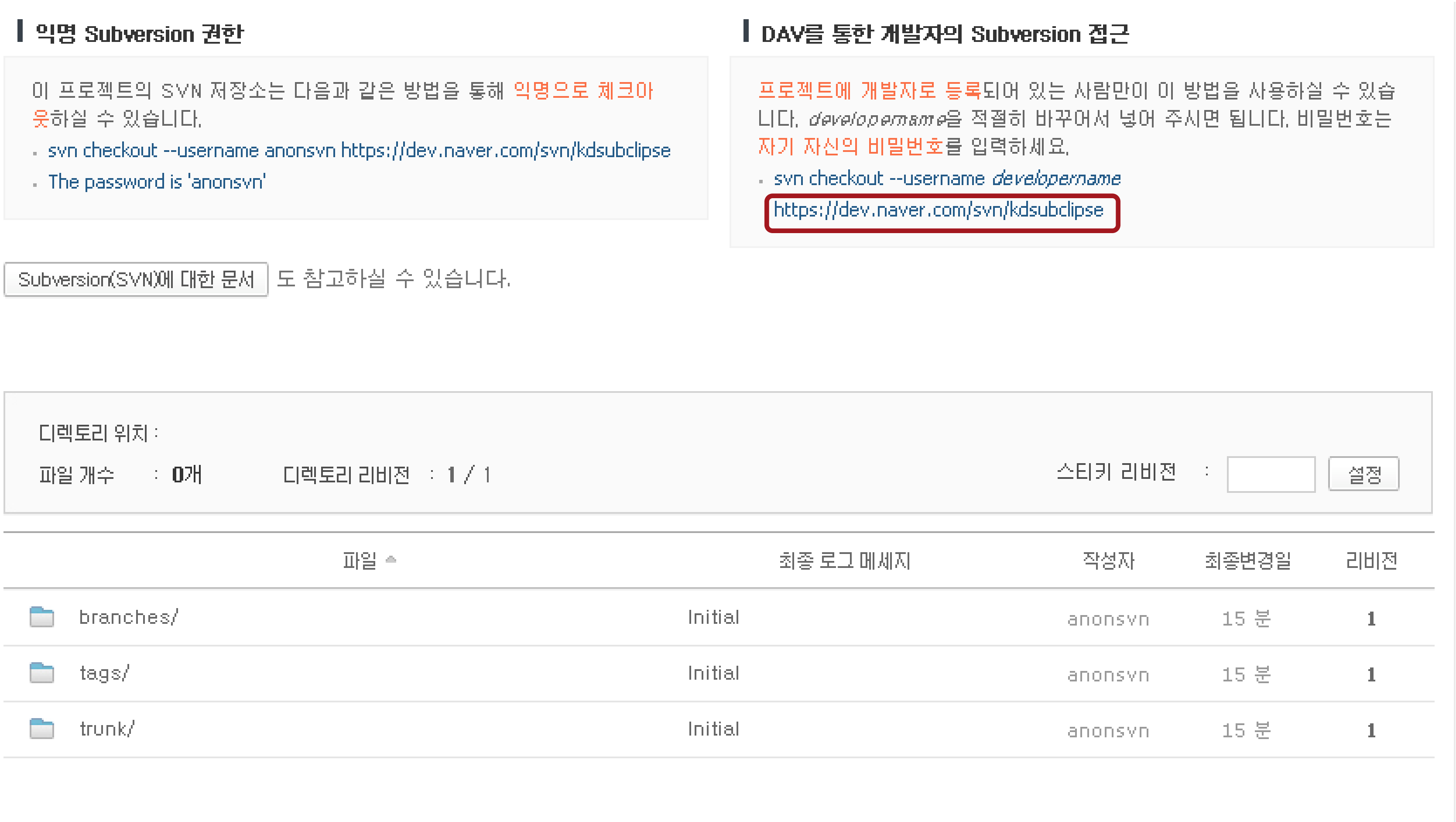
프로젝트 등록

아래의 설명을 주의깊게 읽어보시고, 완전하고 자세한 내용을 제공해 주십시오. 프로젝트 공개설명은 필수입니다.

프로젝트 이름	helloworld
프로젝트 아이디	kdhelloworld
프로젝트 공개 설명	SVN repository를 생성하기 위한 테스트
코드관리시스템	● SVN ○ Git ○ Mercurial

□ 타인의 지적재산권이나 저작권을 침해할 경우 이용이 제한될 수 있음에 동의합니다.

Subclipse



익명 Subversion 권한

이 프로젝트의 SVN 저장소는 다음과 같은 방법을 통해 익명으로 체크아웃하실 수 있습니다.

- svn checkout --username anonsvn https://dev.naver.com/svn/kdsubclipse
- The password is 'anonsvn'

DAV를 통한 개발자의 Subversion 접근

프로젝트에 개발자로 등록되어 있는 사람만이 이 방법을 사용하실 수 있습니다. *developername*을 적절히 바꾸어서 넣어 주시면 됩니다. 비밀번호는 자기 자신의 비밀번호를 입력하세요.

- svn checkout --username *developername* https://dev.naver.com/svn/kdsubclipse

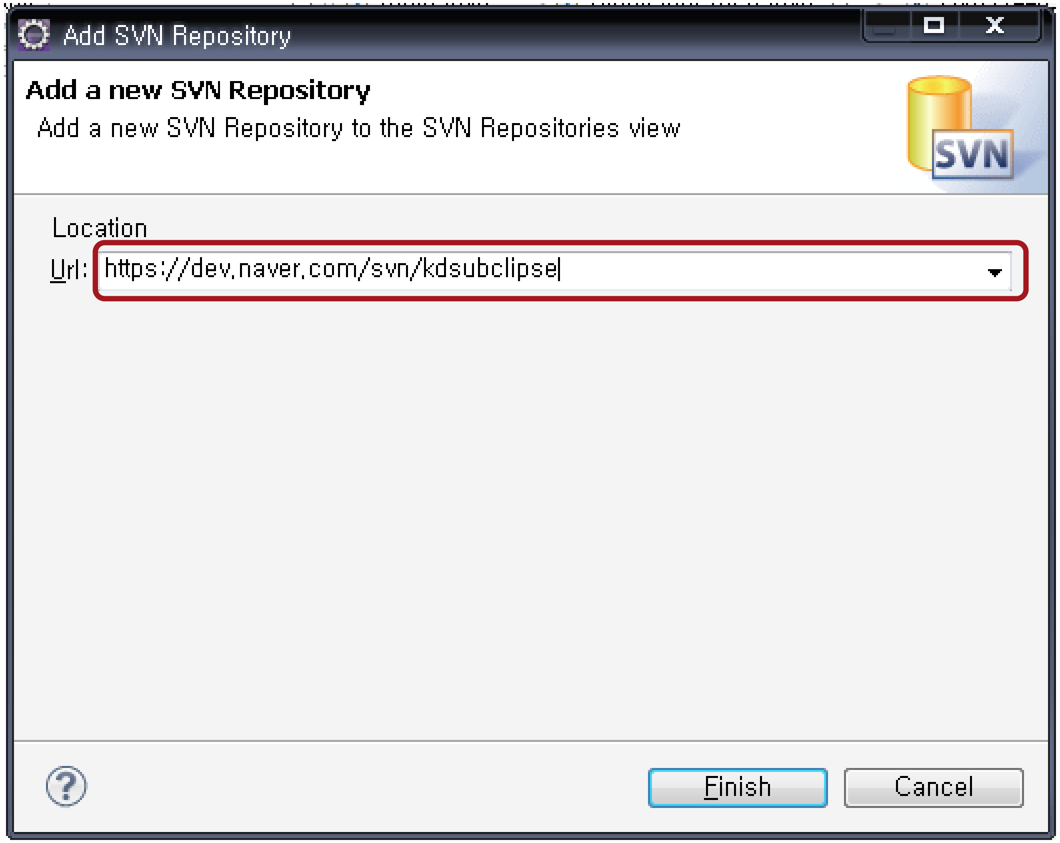
Subversion(SVN)에 대한 문서 도 참고하실 수 있습니다.

디렉토리 위치 :

파일 개수 : 0개 디렉토리 리비전 : 1 / 1 스티키 리비전 : 설정

파일 ▲	최종 로그 메세지	작성자	최종변경일	리비전
branches/	Initial	anonsvn	15 분	1
tags/	Initial	anonsvn	15 분	1
trunk/	Initial	anonsvn	15 분	1

Subclipse



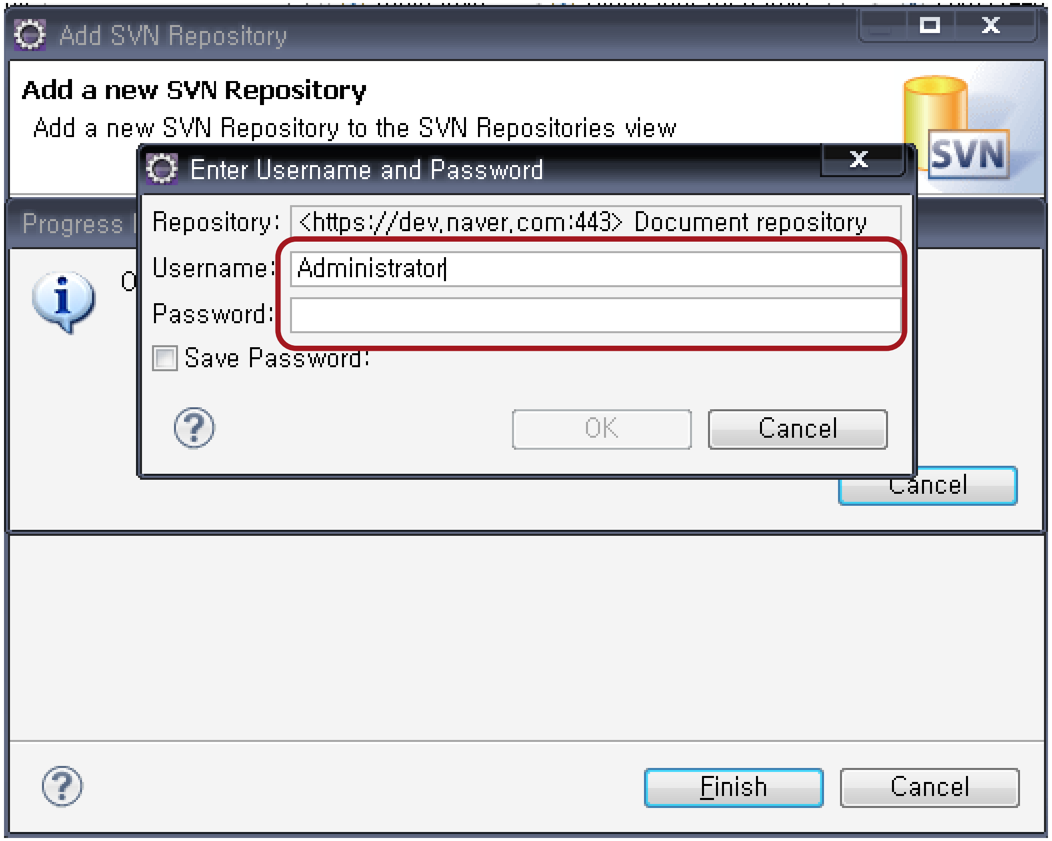
Add a new SVN Repository

Add a new SVN Repository to the SVN Repositories view

Location

Url: https://dev.naver.com/svn/kdsubclipse

Subclipse



Add SVN Repository

Add a new SVN Repository

Add a new SVN Repository to the SVN Repositories view

Enter Username and Password

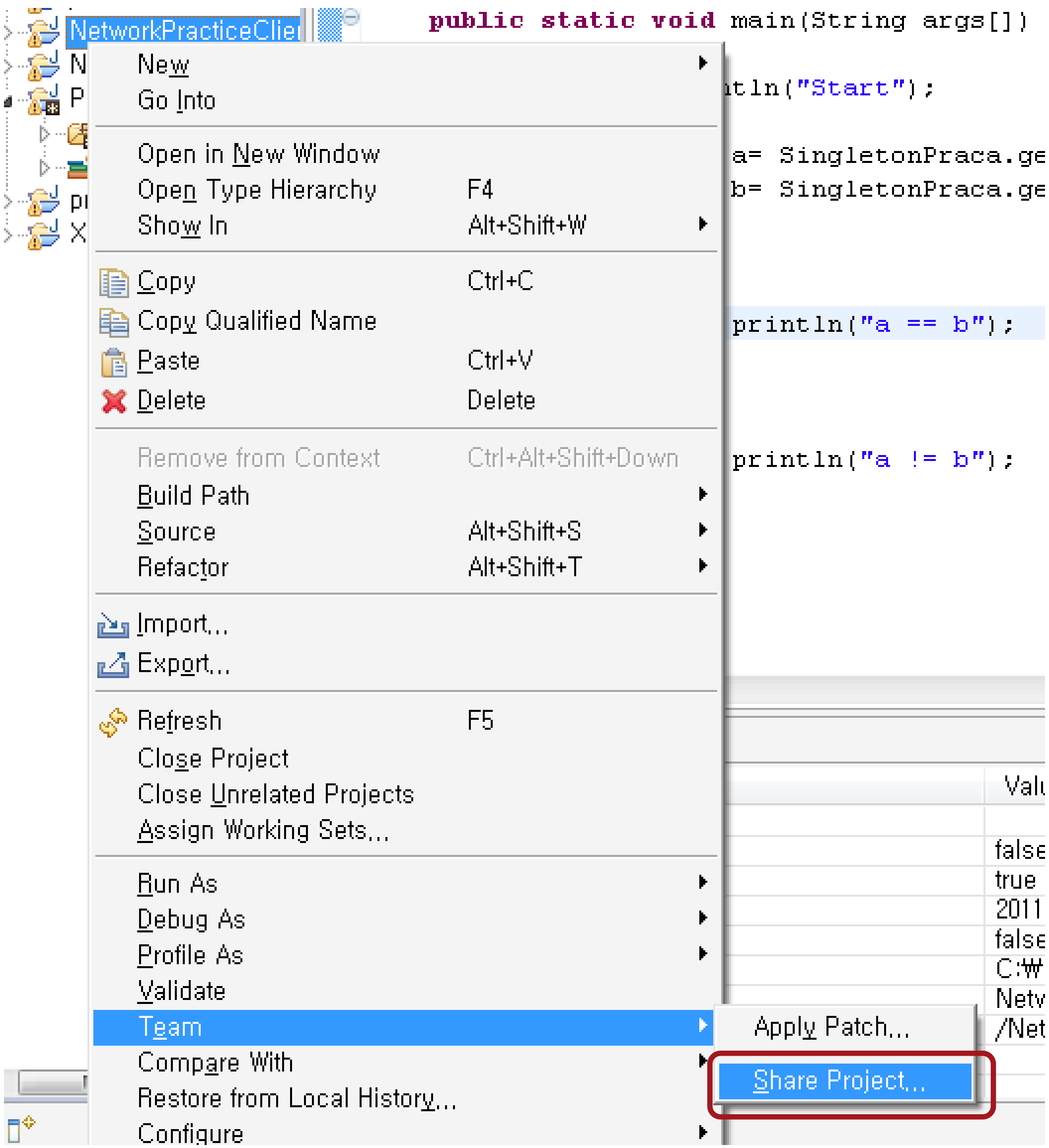
Repository: <https://dev.naver.com:443> Document repository

Username: Administrator

Password:

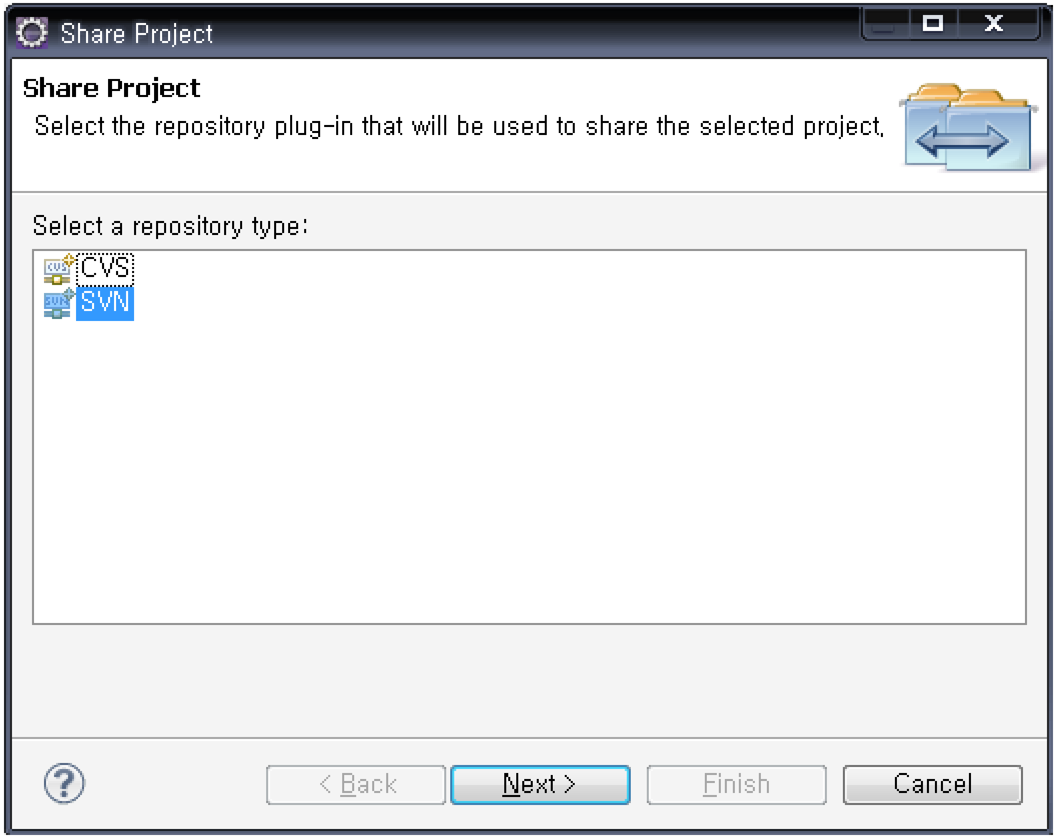
□ Save Password:

Subclipse



```
public static void main(String args[])
tln("Start");
a= SingletonPraca.ge
b= SingletonPraca.ge
println("a == b");
println("a != b");
```

Subclipse



Share Project

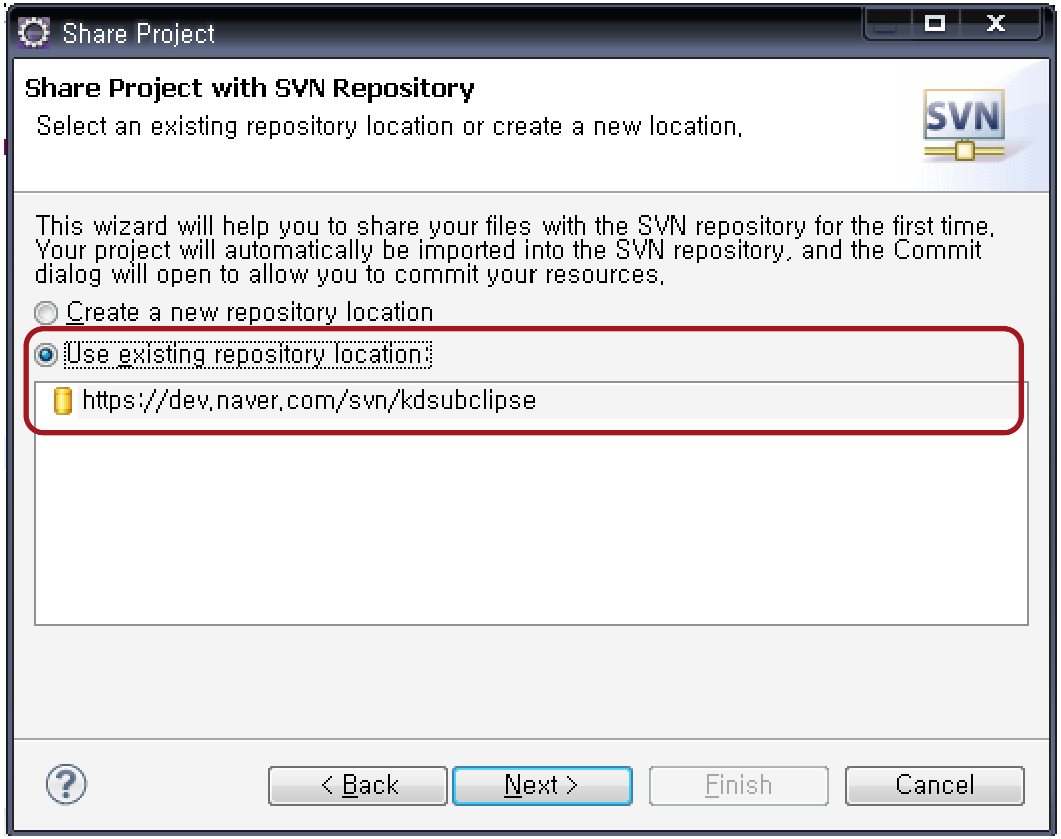
Share Project

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

Select a repository type:

- CVS
- SVN

Subclipse



Share Project with SVN Repository

Select an existing repository location or create a new location.

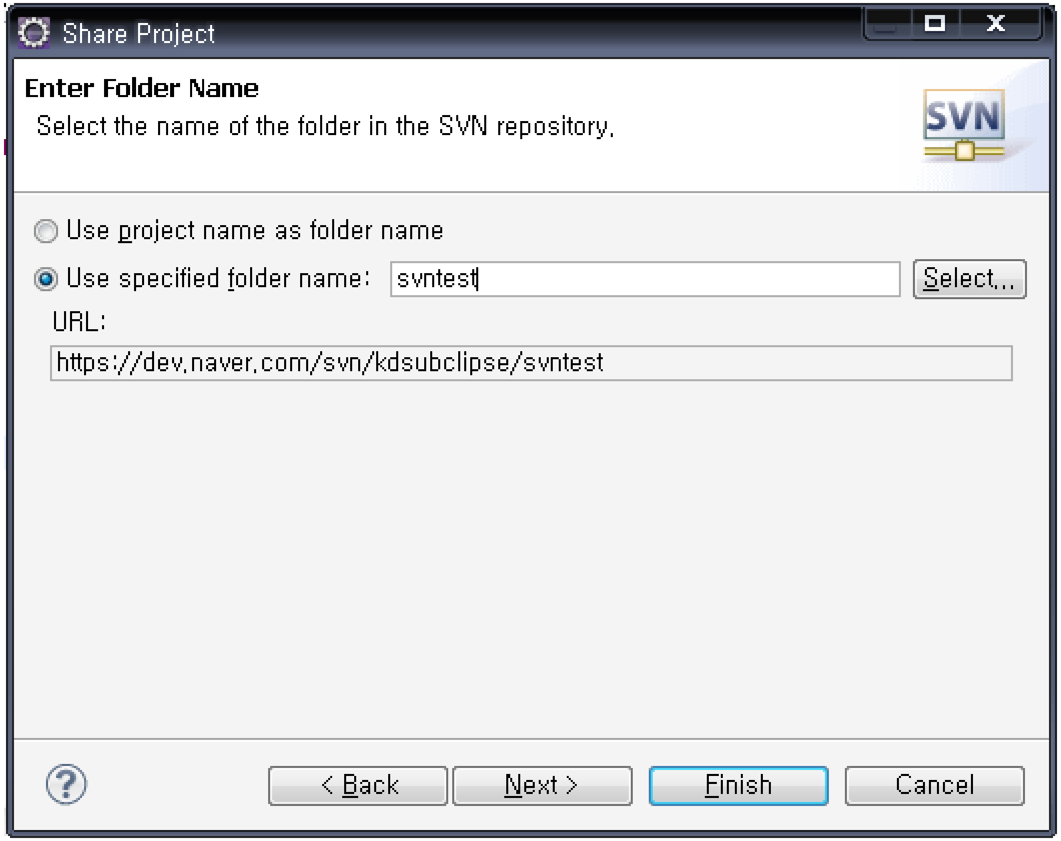
This wizard will help you to share your files with the SVN repository for the first time. Your project will automatically be imported into the SVN repository, and the Commit dialog will open to allow you to commit your resources.

○ Create a new repository location

● Use existing repository location:

https://dev.naver.com/svn/kdsubclipse

Subclipse



Share Project

Enter Folder Name

Select the name of the folder in the SVN repository.

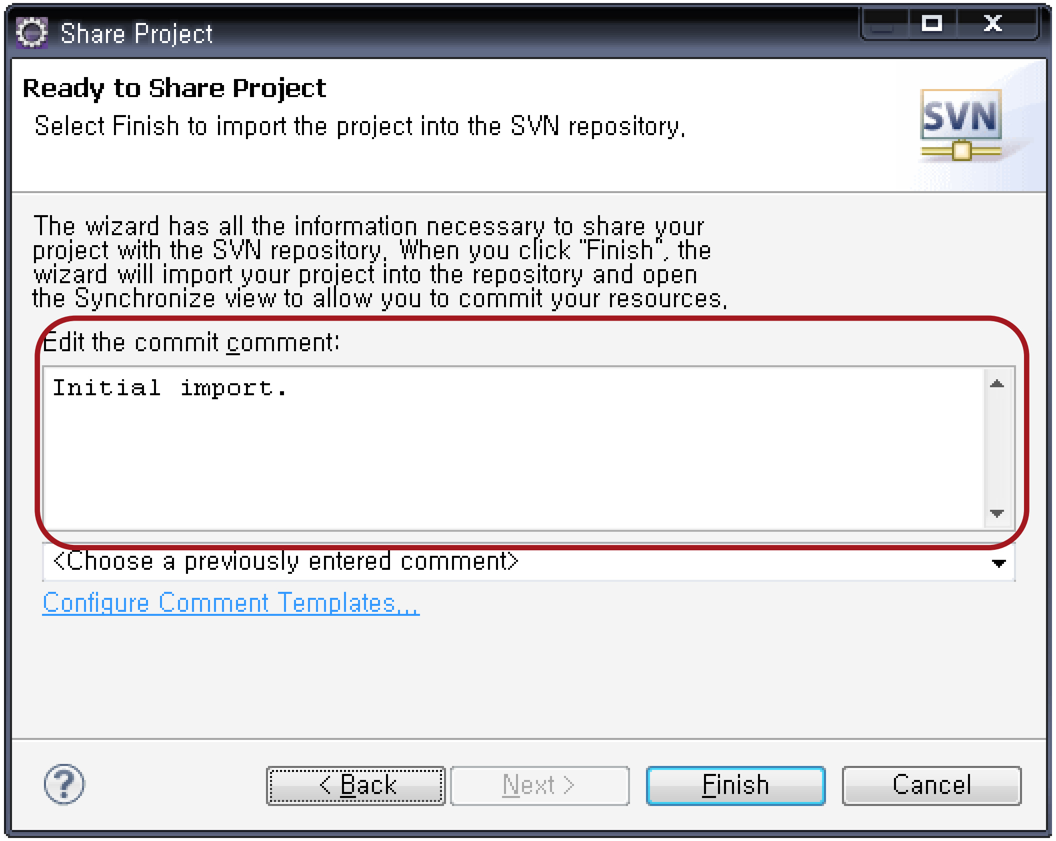
○ Use project name as folder name

● Use specified folder name: svntest

URL:

https://dev.naver.com/svn/kdsubclipse/svntest

Subclipse



Ready to Share Project

Select Finish to import the project into the SVN repository.

The wizard has all the information necessary to share your project with the SVN repository. When you click "Finish", the wizard will import your project into the repository and open the Synchronize view to allow you to commit your resources.

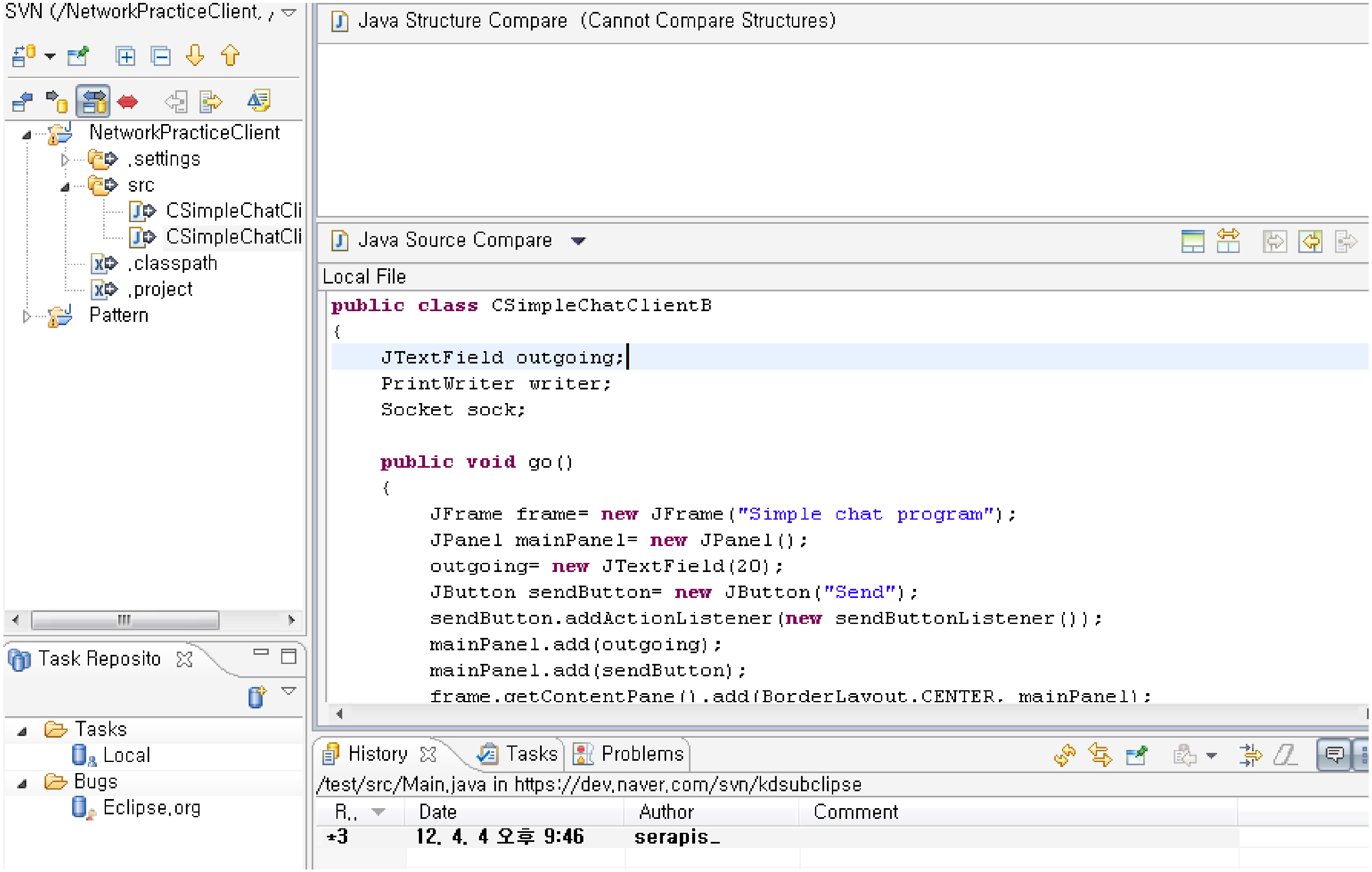
Edit the commit comment:

```
Initial import.
```

<Choose a previously entered comment>

Configure Comment Templates...

Subclipse



```
public class CSimpleChatClientB
{
    JTextField outgoing;
    PrintWriter writer;
    Socket sock;

    public void go()
    {
        JFrame frame= new JFrame("Simple chat program");
        JPanel mainPanel= new JPanel();
        outgoing= new JTextField(20);
        JButton sendButton= new JButton("Send");
        sendButton.addActionListener(new sendButtonListener());
        mainPanel.add(outgoing);
        mainPanel.add(sendButton);
        frame.getContentPane().add(BorderLayout.CENTER, mainPanel);
```

/test/src/Main.java in https://dev.naver.com/svn/kdsubclipse

R..	Date	Author	Comment
*3	12. 4. 4 오후 9:46	serapis_	

Subclipse

Commit

Update

Check Out

Compare

Subclipse

Commit

Update

Check Out

Compare

Apply the changes of the files to the repository

Subclipse

Commit

Update

Check Out

Compare

Get latest files from the repository

Subclipse

Commit

Update

Check Out

Compare

Get files from the repository at initial time

Subclipse

Commit

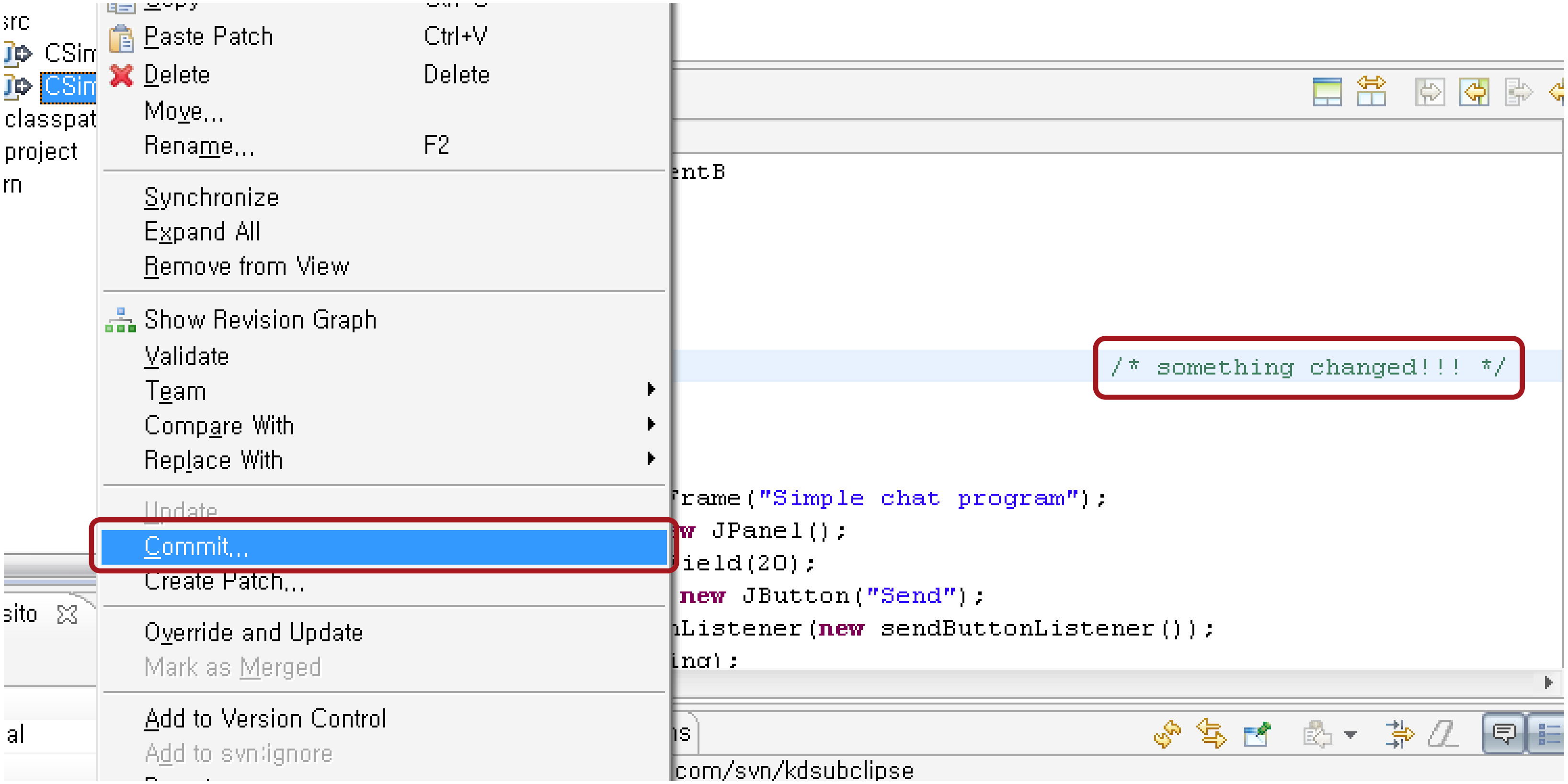
Update

Check Out

Compare

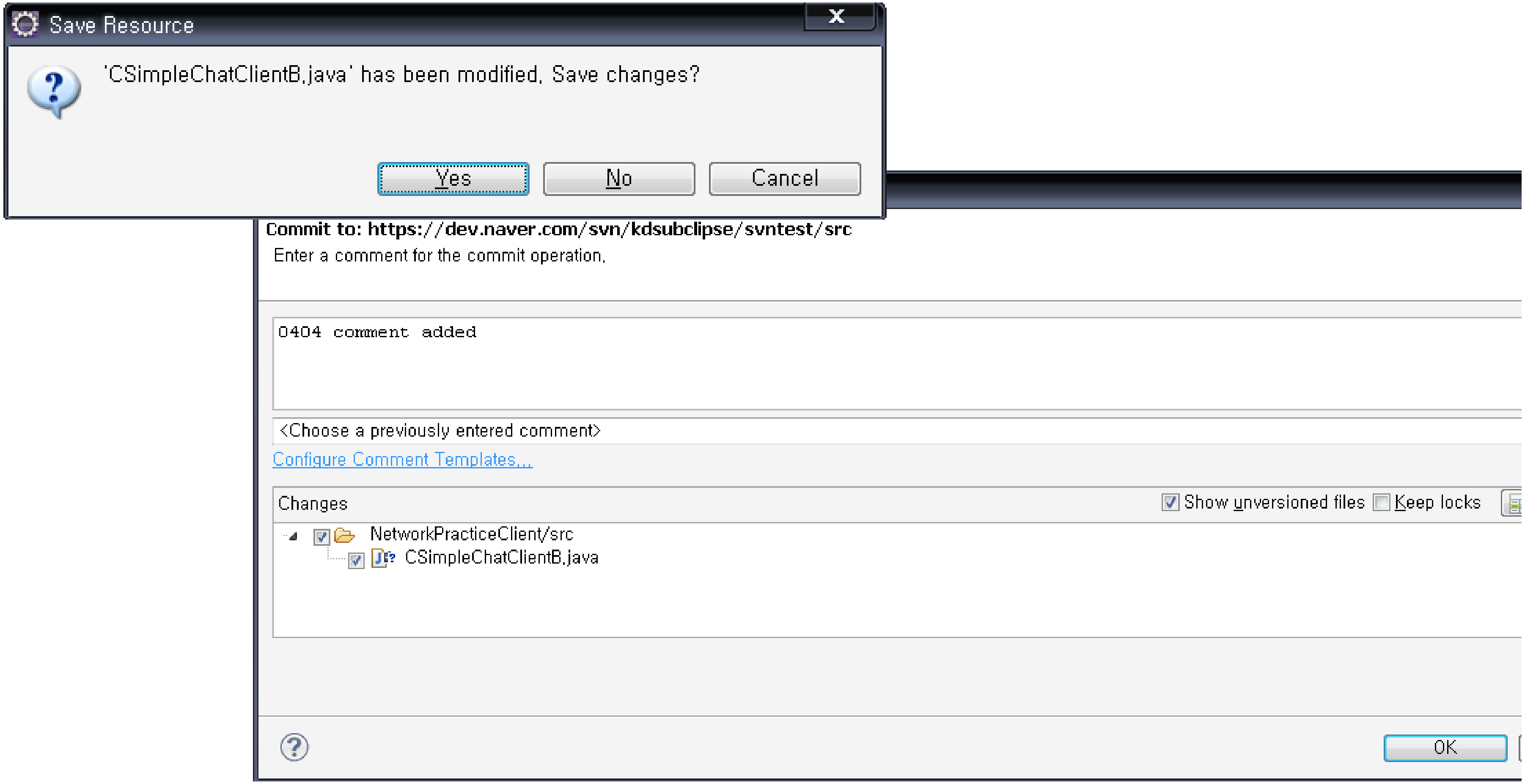
Compare with last files

Subclipse



```
entB

/* something changed!!! */

rame("Simple chat program");
w JPanel();
ield(20);
 new JButton("Send");
nListener(new sendButtonListener());
ing);
```

Subclipse



'CSimpleChatClientB.java' has been modified. Save changes?

Commit to: https://dev.naver.com/svn/kdsubclipse/svntest/src

Enter a comment for the commit operation.

```
0404 comment added
```

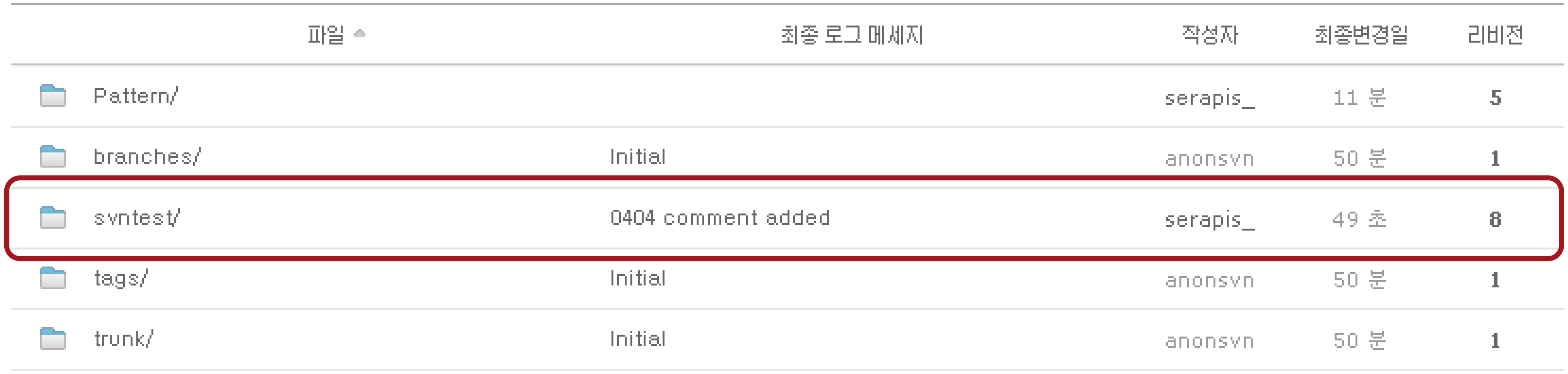
<Choose a previously entered comment>

Configure Comment Templates...

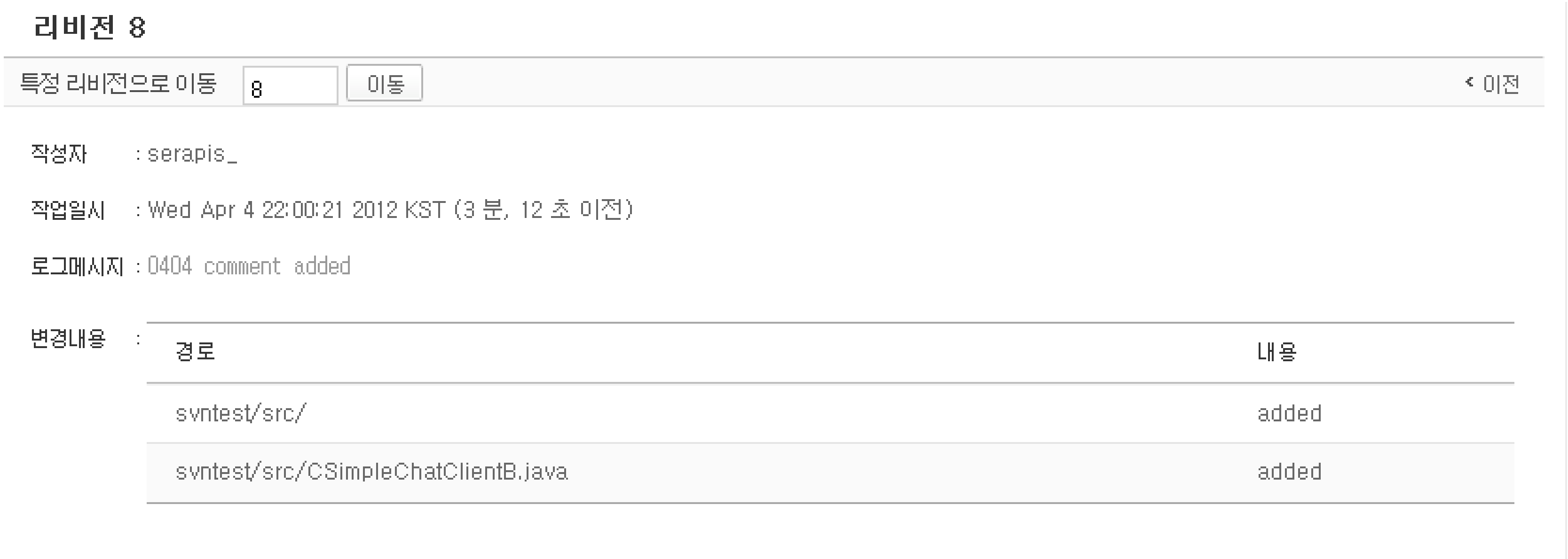
Changes ☑ Show unversioned files □ Keep locks

- ☑ NetworkPracticeClient/src
  - ☑ CSimpleChatClientB.java

Subclipse



파일 ▲	최종 로그 메세지	작성자	최종변경일	리비전
Pattern/		serapis_	11 분	5
branches/	Initial	anonsvn	50 분	1
svntest/	0404 comment added	serapis_	49 초	8
tags/	Initial	anonsvn	50 분	1
trunk/	Initial	anonsvn	50 분	1



리비전 8

특정 리비전으로 이동 8 이동 ‹ 이전

작성자 : serapis_

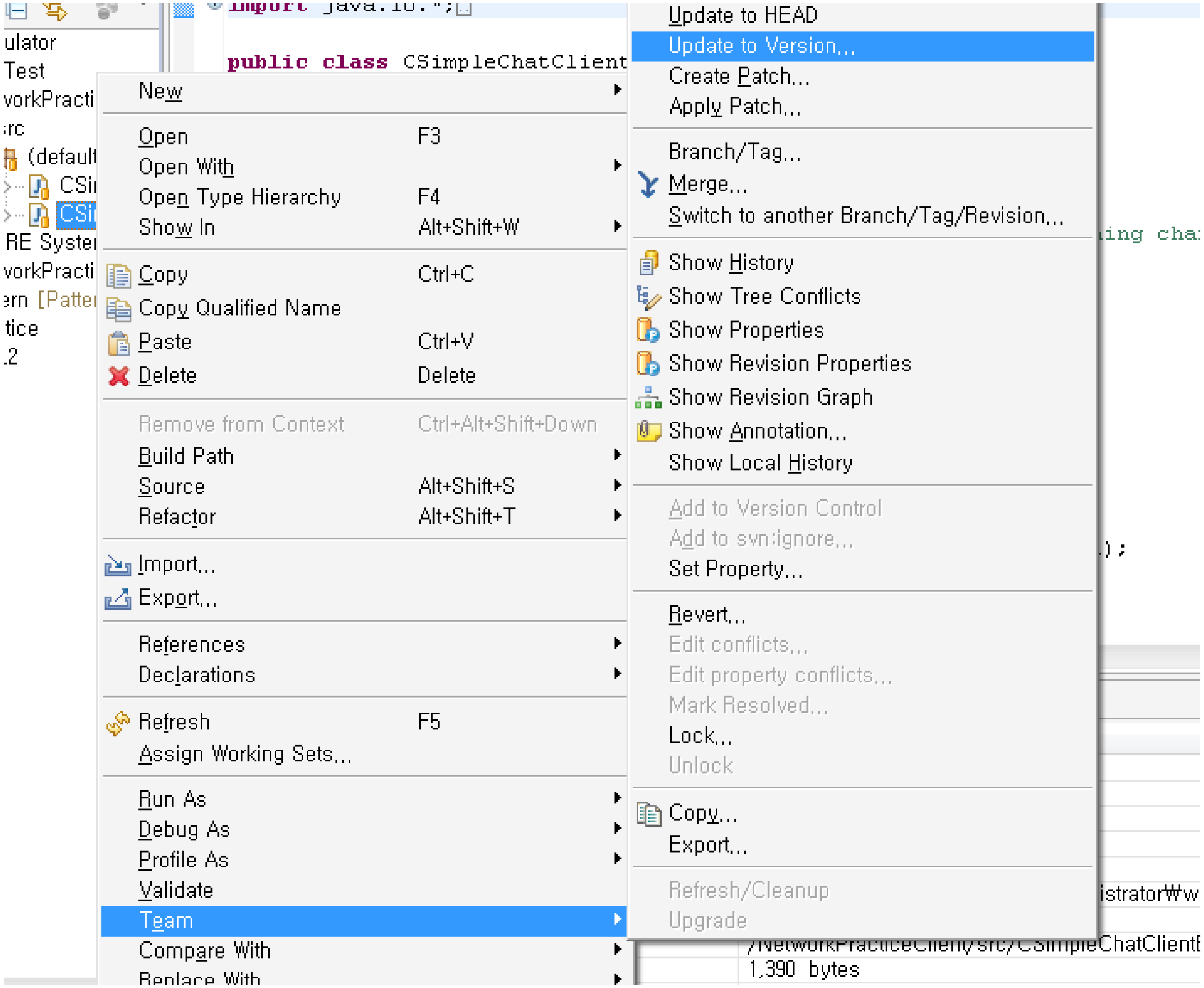
작업일시 : Wed Apr 4 22:00:21 2012 KST (3 분, 12 초 이전)

로그메시지 : 0404 comment added

변경내용 :

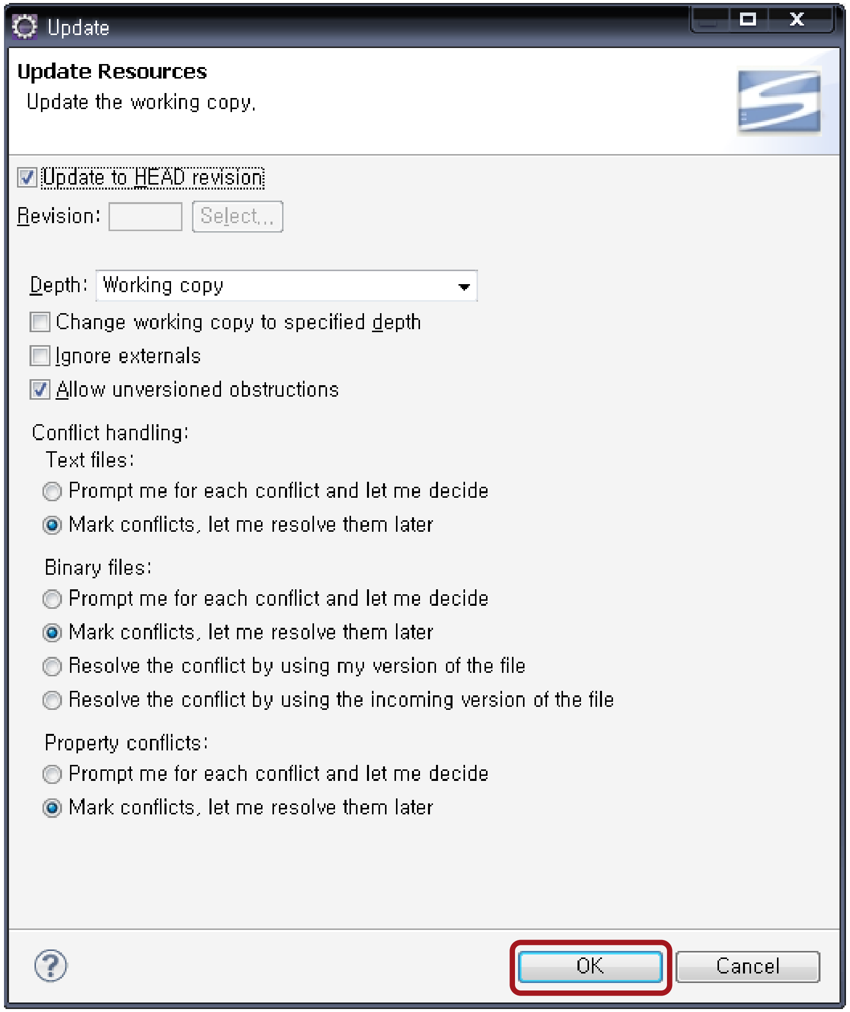
경로	내용
svntest/src/	added
svntest/src/CSimpleChatClientB.java	added

Subclipse



```
import java.io.*;
public class CSimpleChatClient
```

Subclipse



Update Resources

Update the working copy.

☑ Update to HEAD revision

Revision: Select...

Depth: Working copy

□ Change working copy to specified depth

□ Ignore externals

☑ Allow unversioned obstructions

Conflict handling:

Text files:

○ Prompt me for each conflict and let me decide

● Mark conflicts, let me resolve them later

Binary files:

○ Prompt me for each conflict and let me decide

● Mark conflicts, let me resolve them later

○ Resolve the conflict by using my version of the file

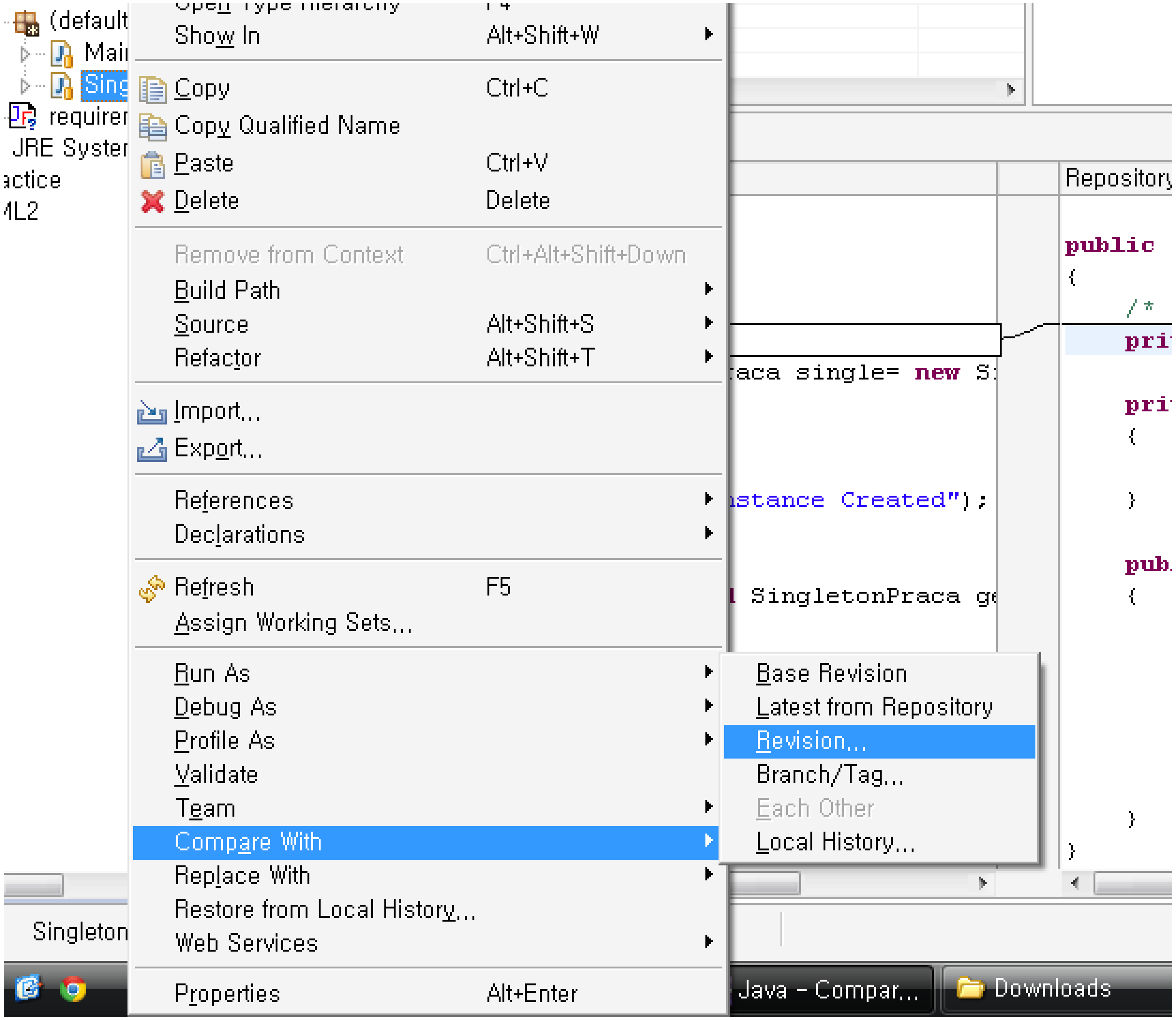
○ Resolve the conflict by using the incoming version of the file

Property conflicts:

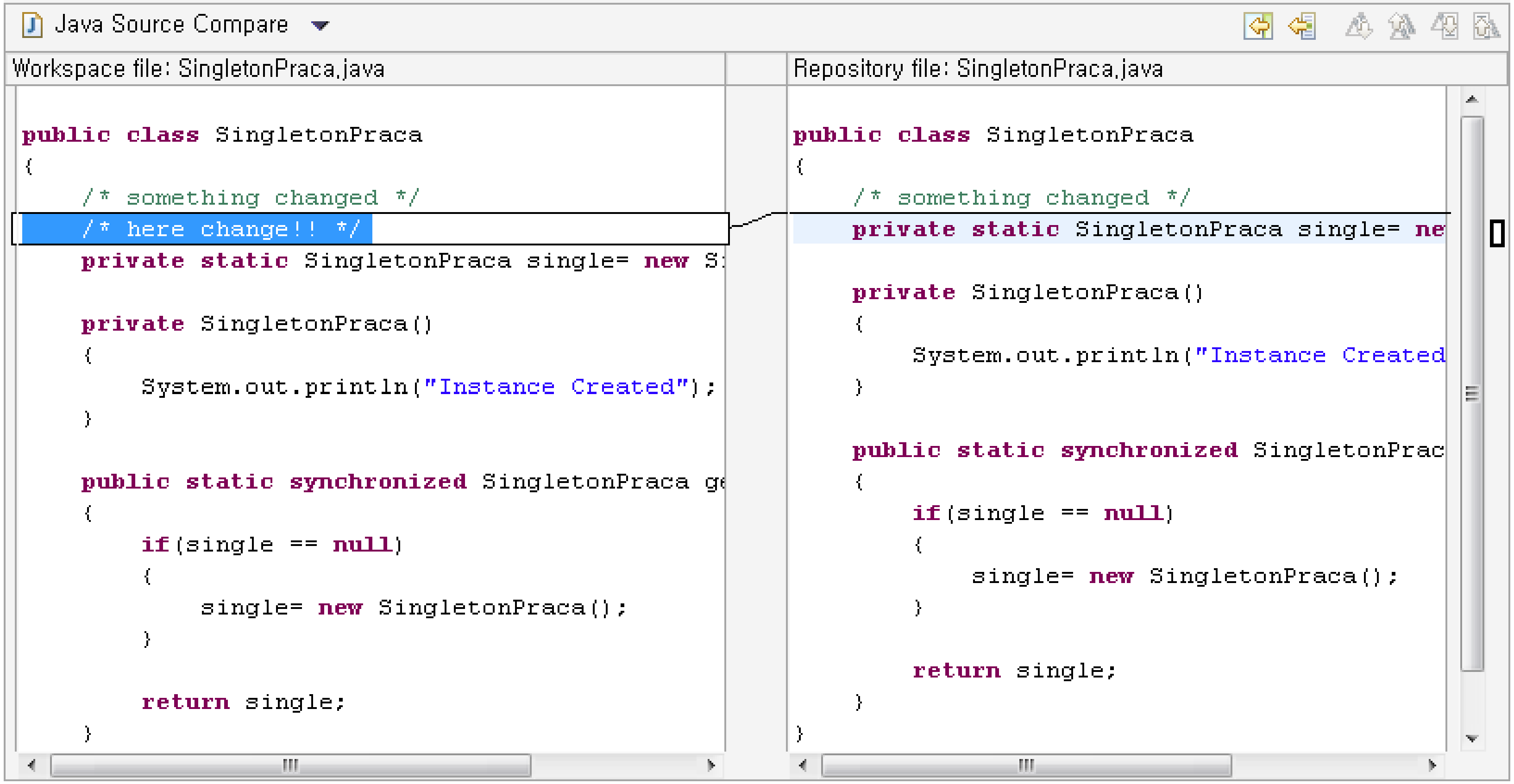
○ Prompt me for each conflict and let me decide

● Mark conflicts, let me resolve them later

Subclipse


(default
Mai
Sing
requirer
JRE Syster
actice
ML2
Singleton
Open Type Hierarchy F4
Show In Alt+Shift+W
Copy Ctrl+C
Copy Qualified Name
Paste Ctrl+V
Delete Delete
Remove from Context Ctrl+Alt+Shift+Down
Build Path
Source Alt+Shift+S
Refactor Alt+Shift+T
Import...
Export...
References
Declarations
Refresh F5
Assign Working Sets...
Run As
Debug As
Profile As
Validate
Team
Compare With
Replace With
Restore from Local History...
Web Services
Properties Alt+Enter
Base Revision
Latest from Repository
Revision...
Branch/Tag...
Each Other
Local History...
Repository
public
{
/*
pri
pri
{
}
pub
{
}
}
aca single= new S
nstance Created");
l SingletonPraca ge
Java - Compar...
Downloads

Subclipse



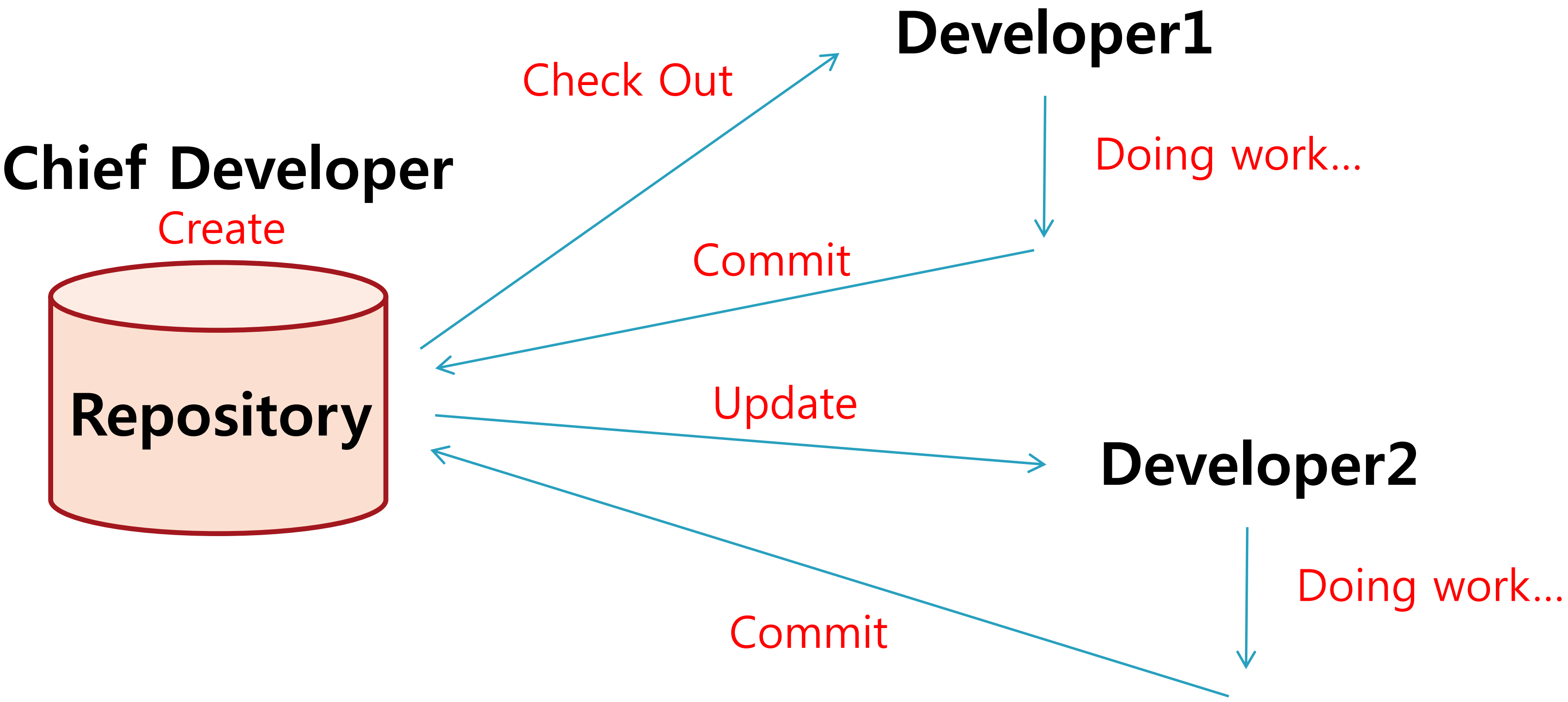
Workspace file: SingletonPraca.java

```
public class SingletonPraca
{
    /* something changed */
    /* here change!! */
    private static SingletonPraca single= new S
    
    private SingletonPraca()
    {
        System.out.println("Instance Created");
    }
    
    public static synchronized SingletonPraca ge
    {
        if(single == null)
        {
            single= new SingletonPraca();
        }
        
        return single;
    }
```

Repository file: SingletonPraca.java

```
public class SingletonPraca
{
    /* something changed */
    private static SingletonPraca single= ne
    
    private SingletonPraca()
    {
        System.out.println("Instance Created
    }
    
    public static synchronized SingletonPrac
    {
        if(single == null)
        {
            single= new SingletonPraca();
        }
        
        return single;
    }
}
```

Subclipse applies in project



Benefit of SCM

already corrected bugs are avoided

better reputation of company

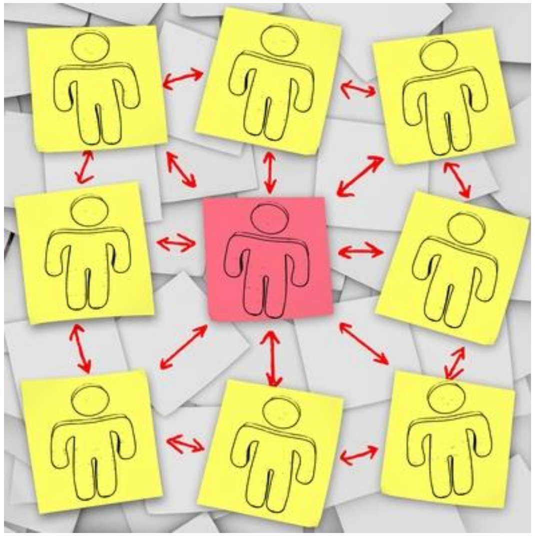
Concurrent development is simplified

RE tool

What is Requirement?

Define the **What** of a software

What the s/w must do **for its stakeholders**



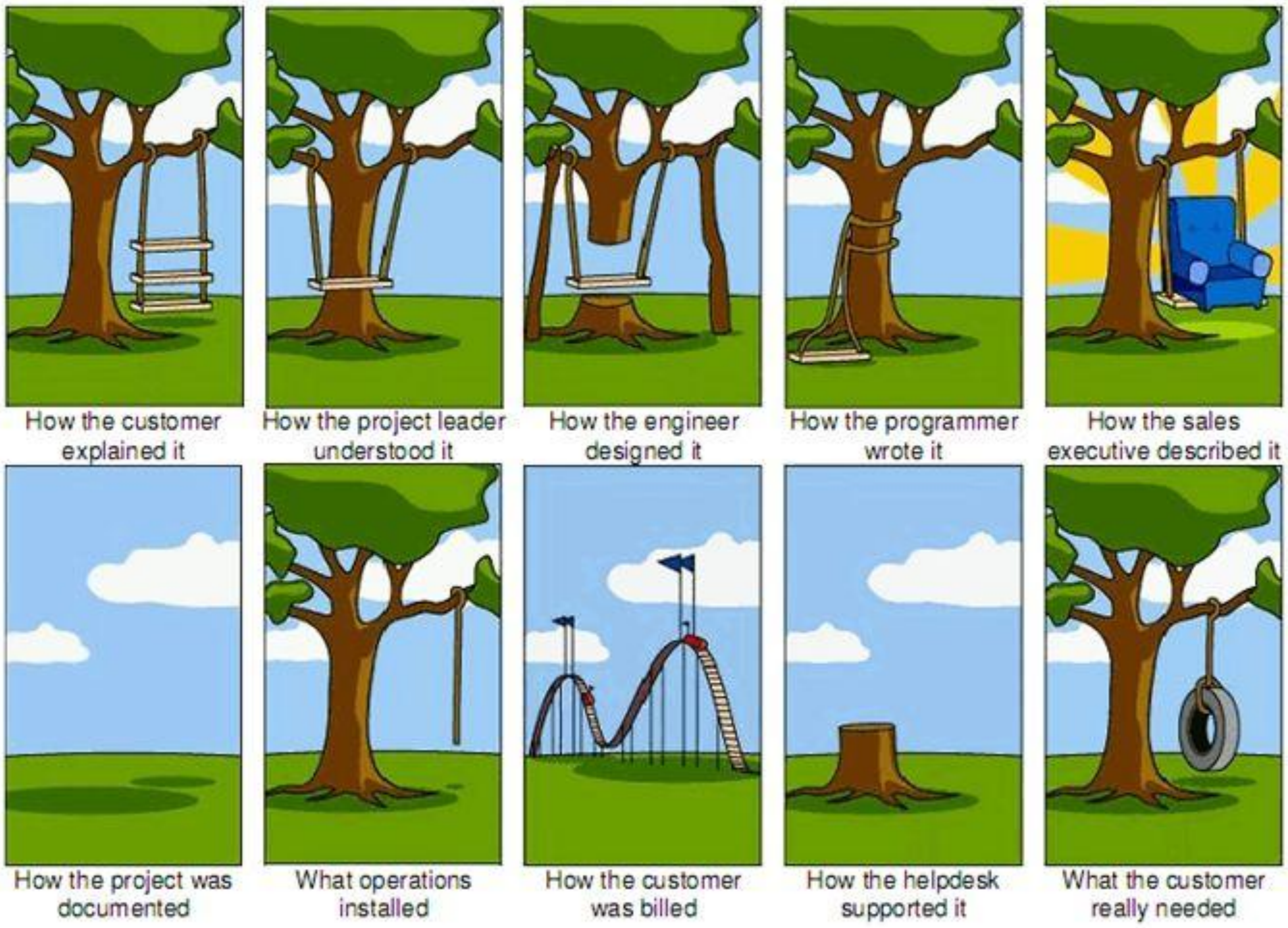
Requirement Engineering

Requirement Engineering is the process of establishing the services that the **customer requires from a system**

concerned with the **real-world goals** and **constraints** on software systems.

Also concerned with **evolution over time**

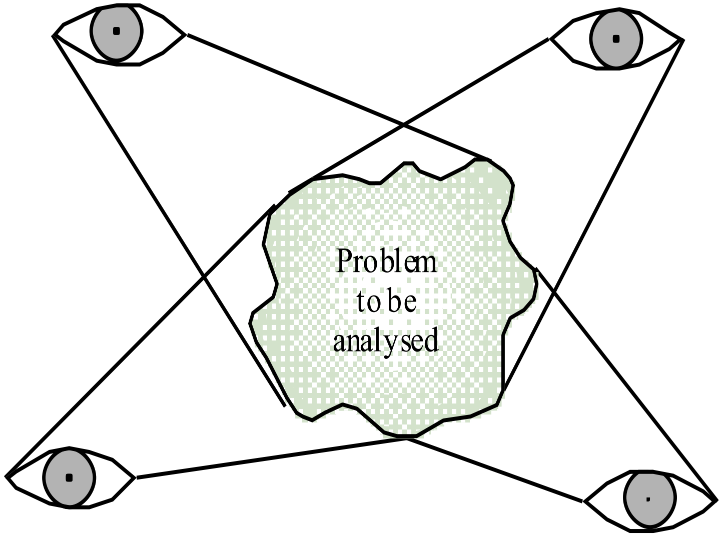
Why we need it?


How the customer explained it
How the project leader understood it
How the engineer designed it
How the programmer wrote it
How the sales executive described it
How the project was documented
What operations installed
How the customer was billed
How the helpdesk supported it
What the customer really needed

Why we need it?

Multiple view points

Evolution



RE process

Feasibility study

Requirements analysis

Requirements definition

Requirements validation

RE process

Feasibility study

Requirements analysis

Requirements definition

Requirements validation

：Find out if the current user needs be satisfied given the available technology and budget

RE process

Feasibility study

Requirements analysis

Requirements definition

Requirements validation

：Find out what system stakeholders require from the system

RE process

Feasibility study

Requirements analysis

Requirements definition

Requirements validation

：Define the requirements in a form understandable to the customer

RE process

Feasibility study

Requirements analysis

Requirements definition

Requirements validation

：validate that defined requirements are satisfied user’s needs.

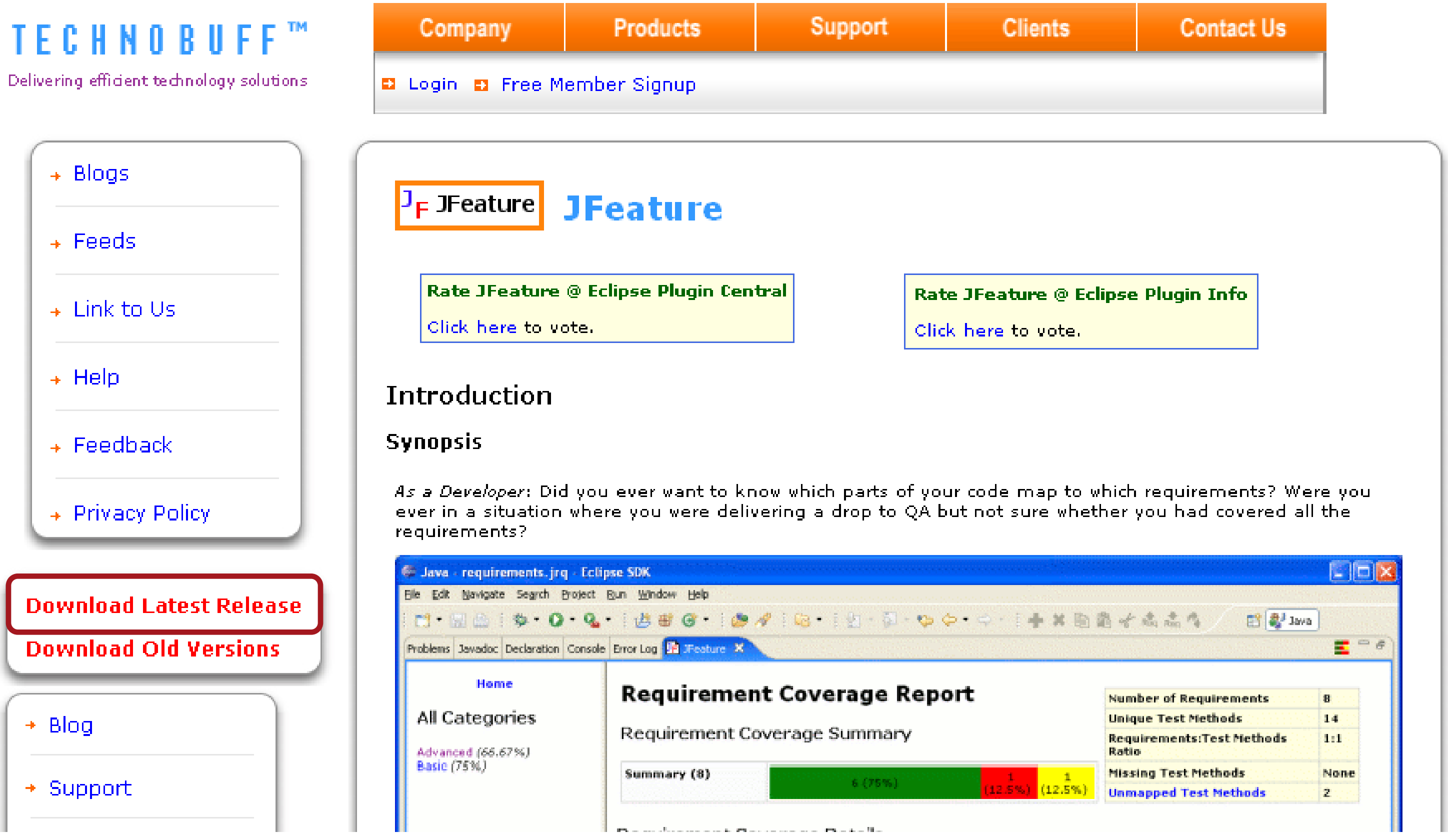
RE tools

JFeature

OSRMT

JFeature

www.technobuff.net



JFeature

Rate JFeature @ Eclipse Plugin Central
Click here to vote.

Rate JFeature @ Eclipse Plugin Info
Click here to vote.

Introduction

Synopsis

As a Developer: Did you ever want to know which parts of your code map to which requirements? Were you ever in a situation where you were delivering a drop to QA but not sure whether you had covered all the requirements?

Java - requirements.jrq - Eclipse SDK
File Edit Navigate Search Project Run Window Help
Java
Problems Javadoc Declaration Console Error Log JFeature
Home
All Categories
Advanced (66.67%)
Basic (75%)

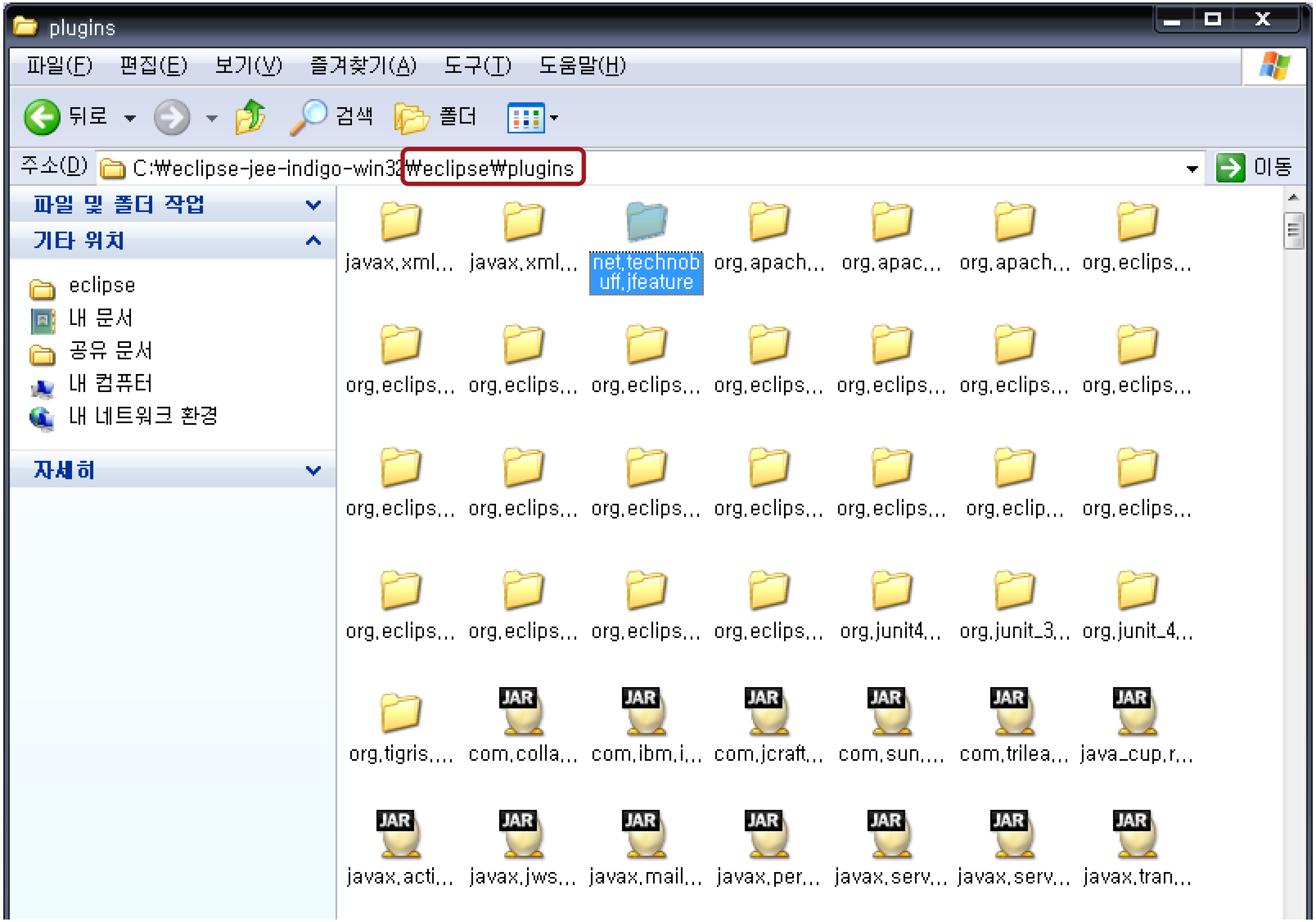
Requirement Coverage Report

Requirement Coverage Summary

Summary (8)	6 (75%)	1 (12.5%)	1 (12.5%)

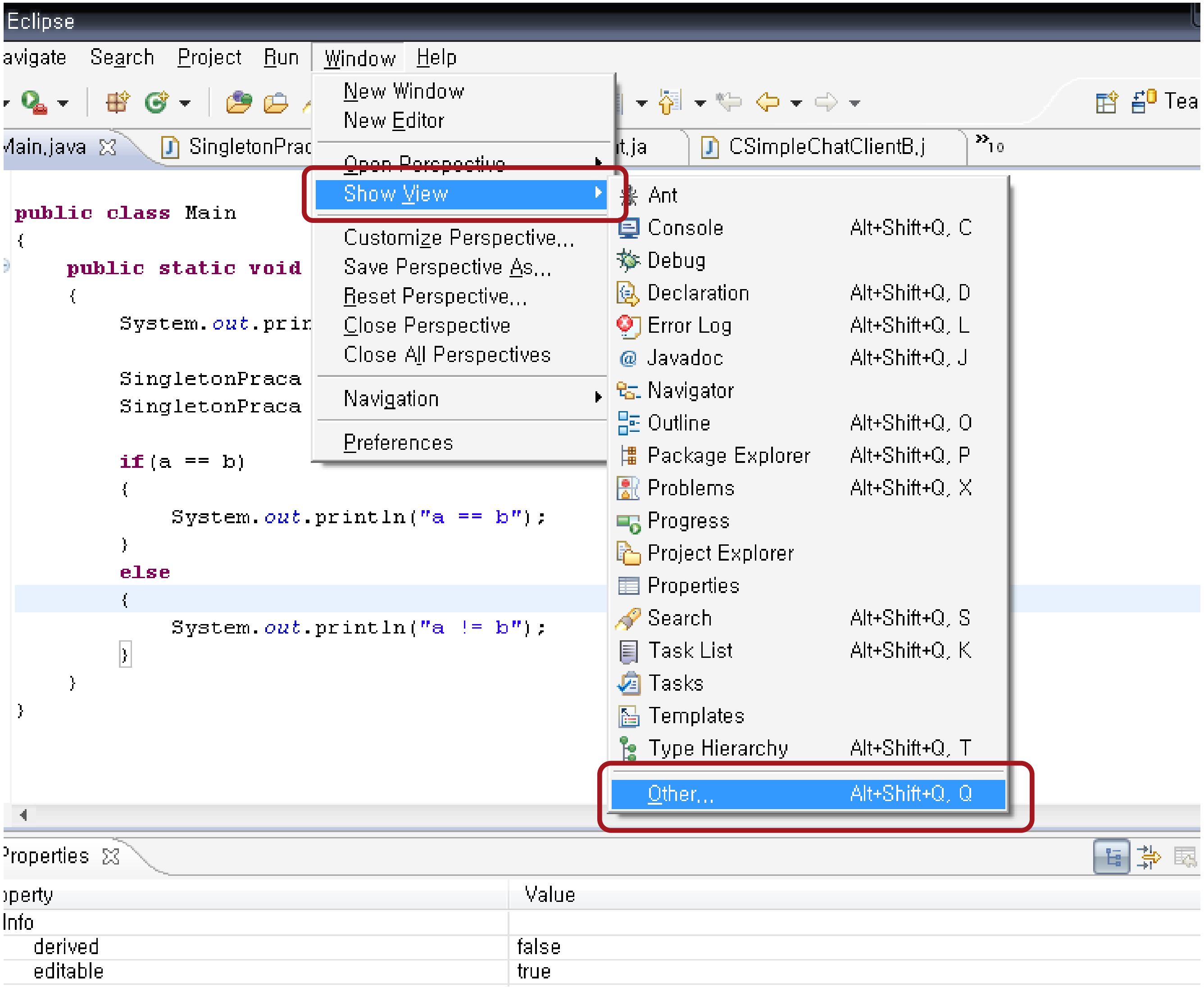
Number of Requirements	8
Unique Test Methods	14
Requirements:Test Methods Ratio	1:1
Missing Test Methods	None
Unmapped Test Methods	2

JFeature



주소(D) C:₩eclipse-jee-indigo-win32₩eclipse₩plugins

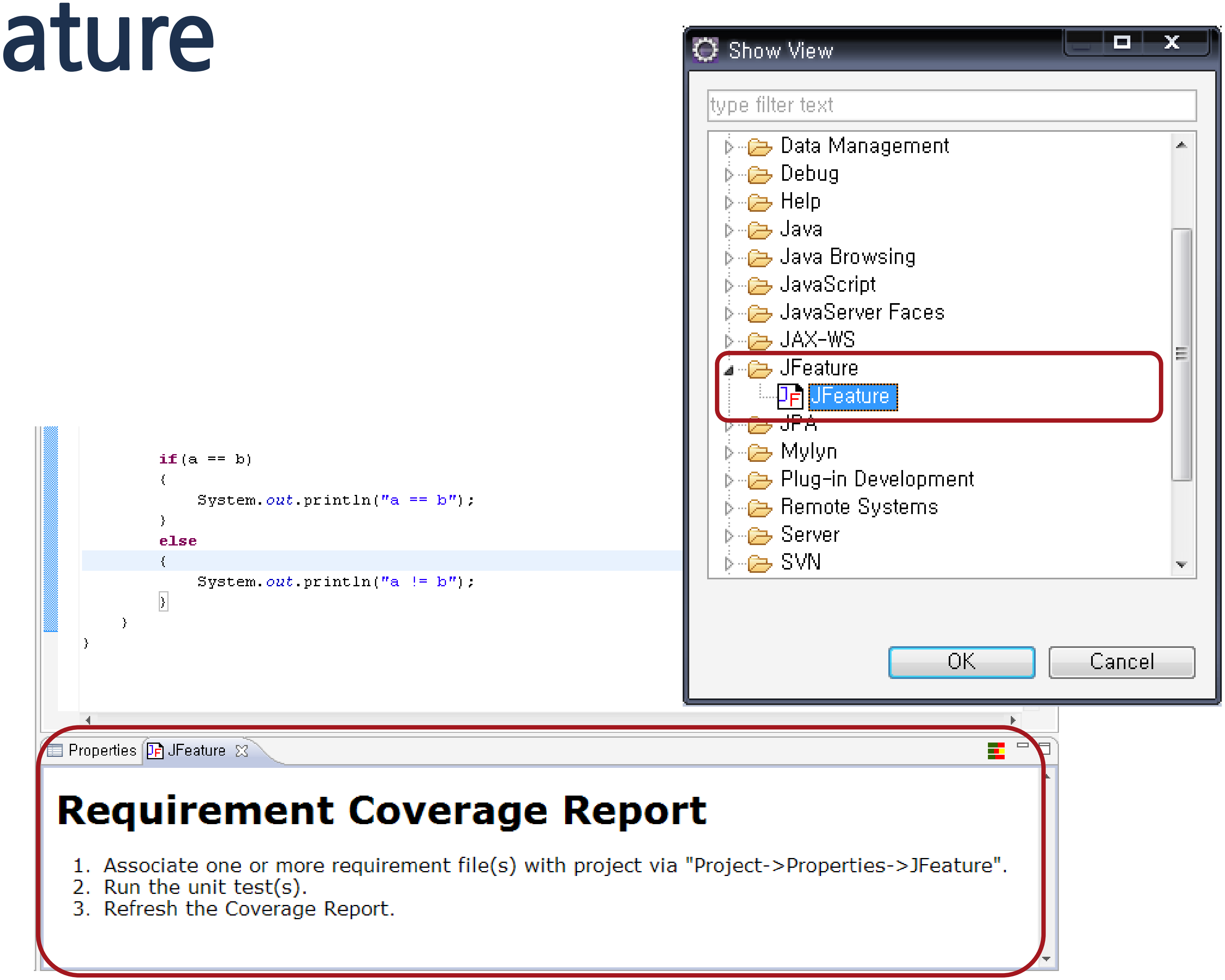
JFeature



```
public class Main
{
    public static void
    {
        System.out.prir

        SingletonPraca
        SingletonPraca

        if (a == b)
        {
            System.out.println("a == b");
        }
        else
        {
            System.out.println("a != b");
        }
    }
}
```

operty	Value
Info	
derived	false
editable	true

JFeature



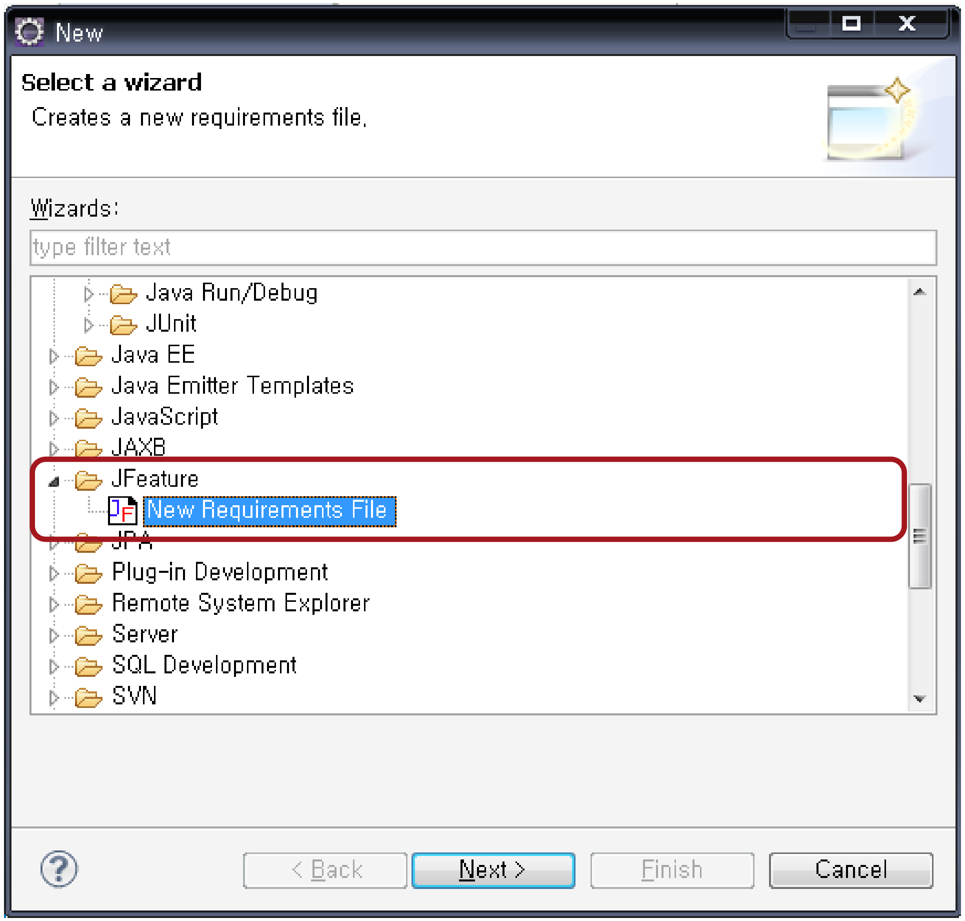
```
        if(a == b)
        {
            System.out.println("a == b");
        }
        else
        {
            System.out.println("a != b");
        }
    }
}
```

Requirement Coverage Report

1. Associate one or more requirement file(s) with project via "Project->Properties->JFeature".
2. Run the unit test(s).
3. Refresh the Coverage Report.

JFeature

File -> New -> Other



New

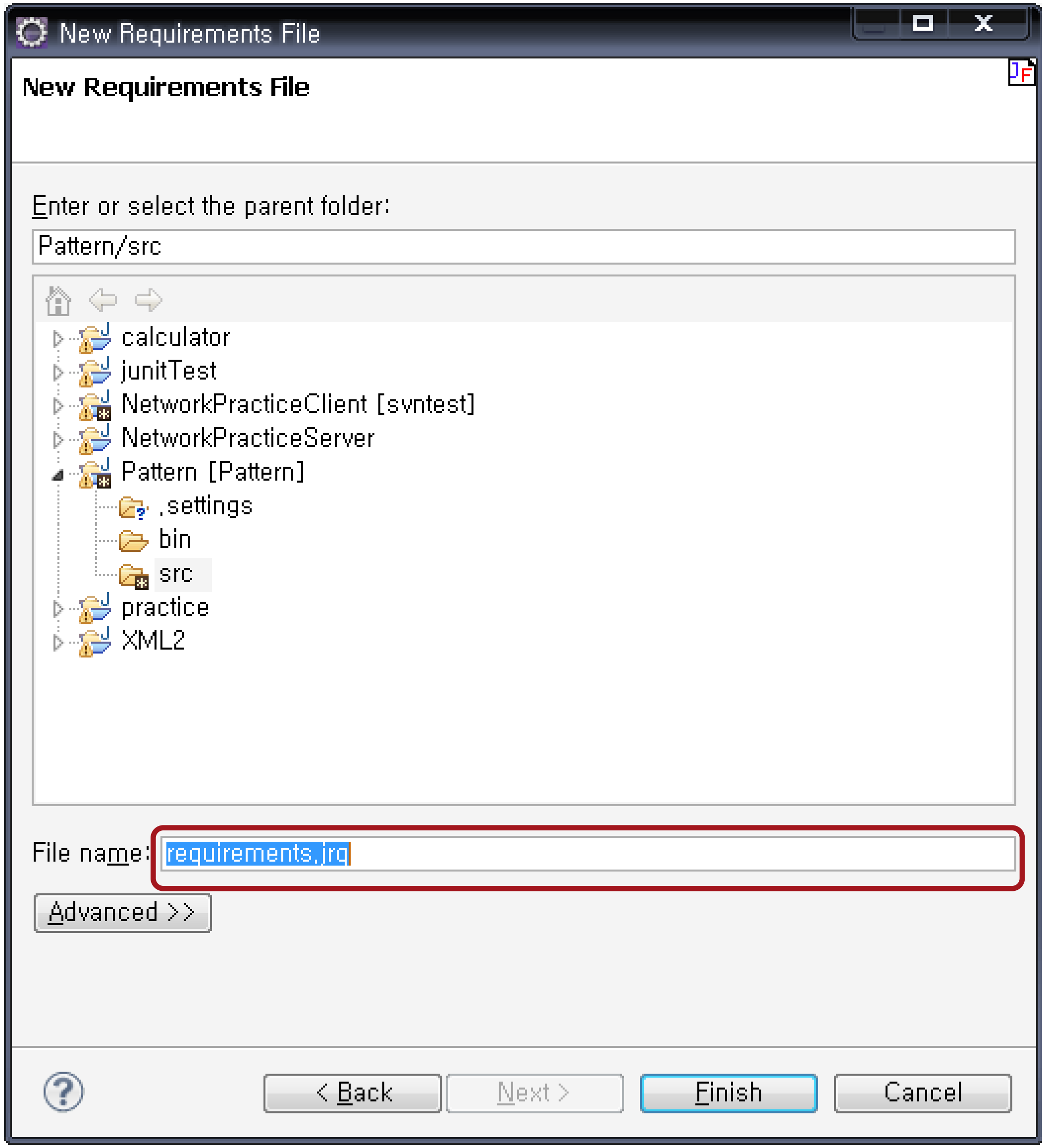
Select a wizard

Creates a new requirements file.

Wizards:

type filter text

JFeature

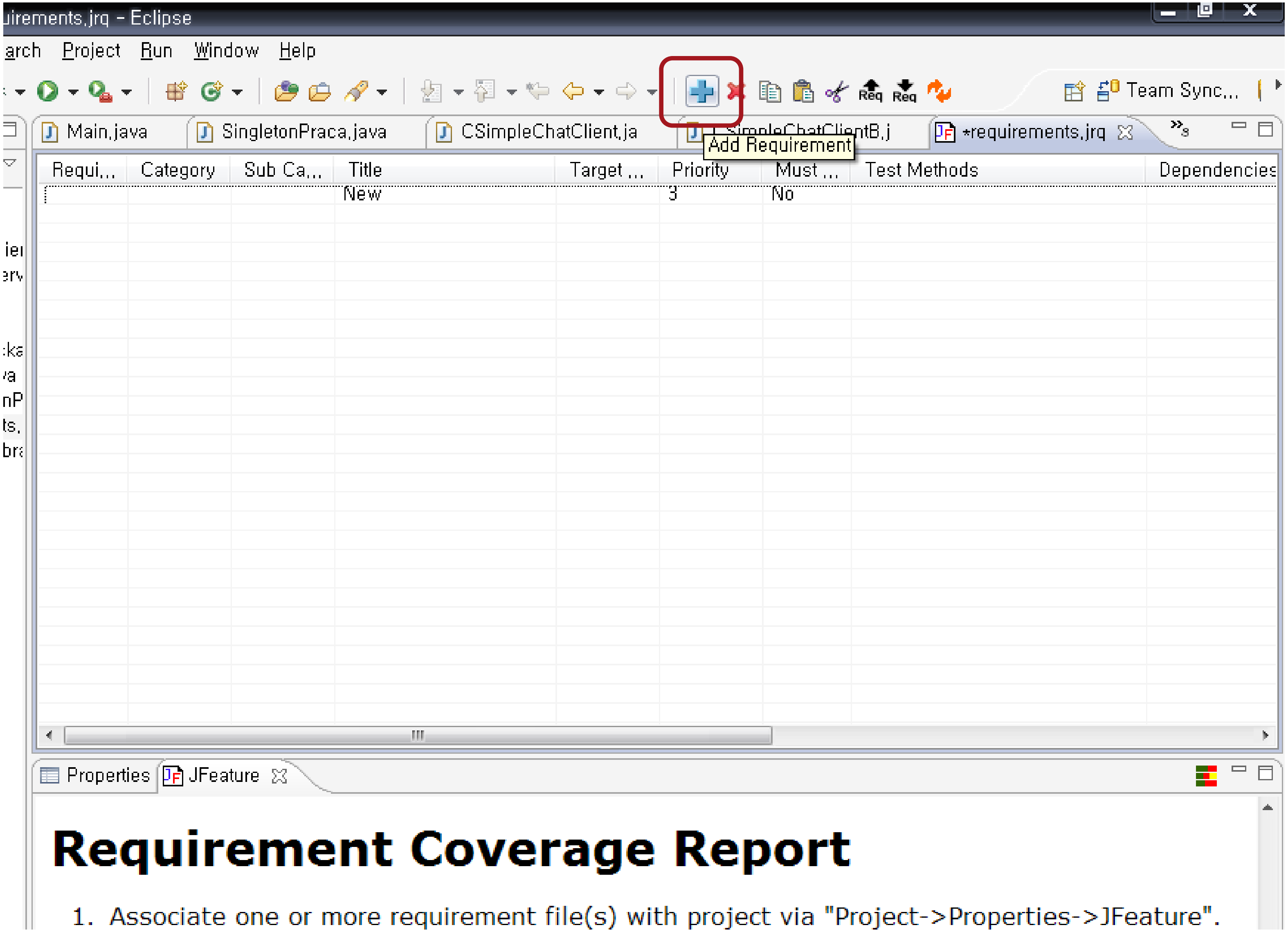


New Requirements File

Enter or select the parent folder: Pattern/src

File name: requirements.jrq

JFeature

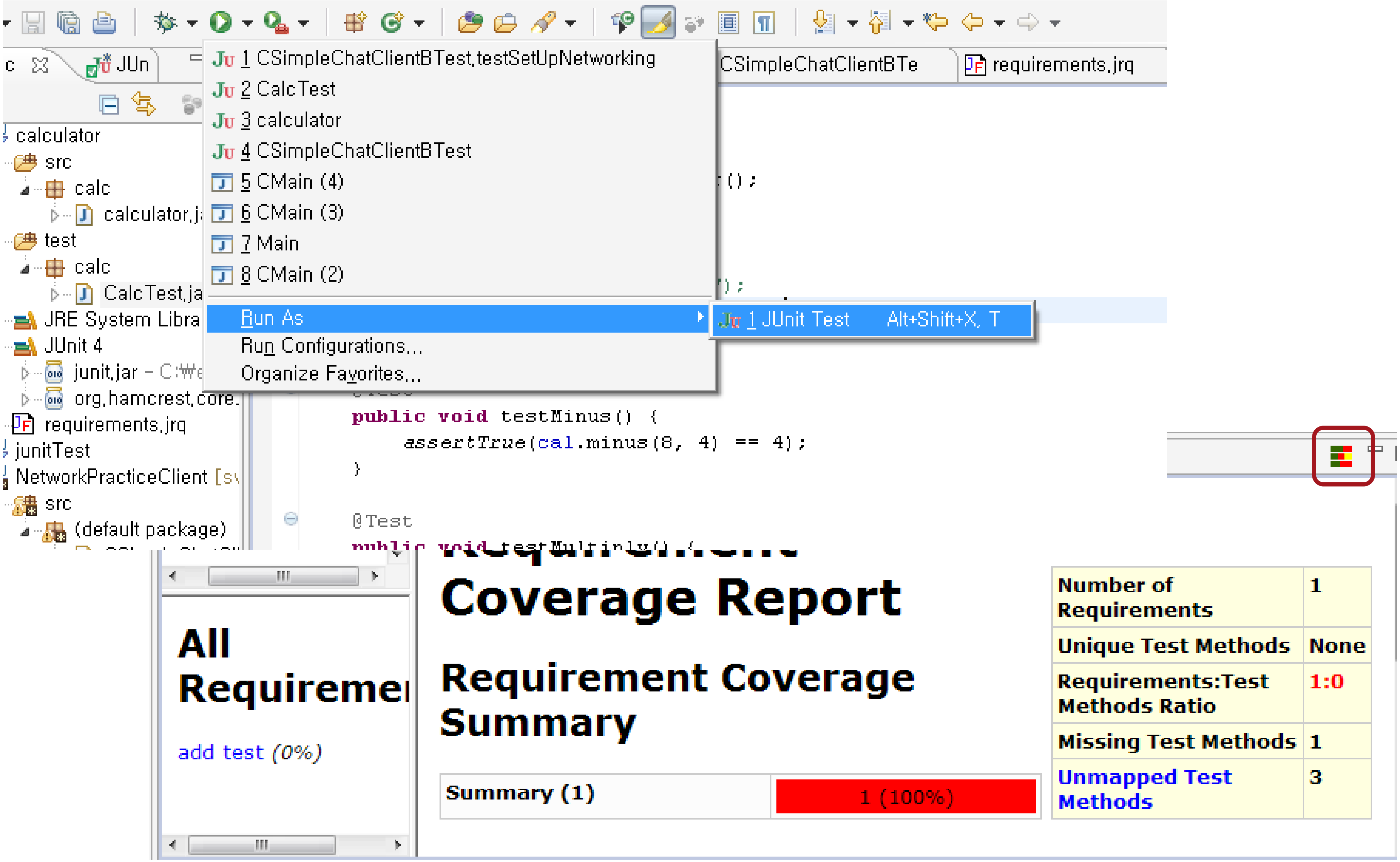


Requi...	Category	Sub Ca...	Title	Target ...	Priority	Must ...	Test Methods	Dependencies
			New		3	No		

Requirement Coverage Report

1. Associate one or more requirement file(s) with project via "Project->Properties->JFeature".

JFeature



```
public void testMinus() {
    assertTrue(cal.minus(8, 4) == 4);
}

@Test
```

All Requireme

add test *(0%)*

Coverage Report

Requirement Coverage Summary

Summary (1)	1 (100%)

Number of Requirements	1
Unique Test Methods	None
Requirements:Test Methods Ratio	1:0
Missing Test Methods	1
Unmapped Test Methods	3

Benefit of RE

Give whole view of the entire requirements

Able to notice requirement changes

Can check requirements are applied

Easy to track flows of the requirement

Thank you