

Công Cụ & Phương Pháp Thiết Kế - Quản Lý (Phần Mềm)

TRAN KIM SANH
Instructor of DTU

Email: trankimsanh@dtu.edu.vn
Tel: 0987 409 464

Using Your Configuration Management Processes Part 1

Contents

■ Java Development Tools:

- ❑ Eclipse
- ❑ SVN
- ❑ Ant
- ❑ JUnit
- ❑ CruiseControl (not used in this course)



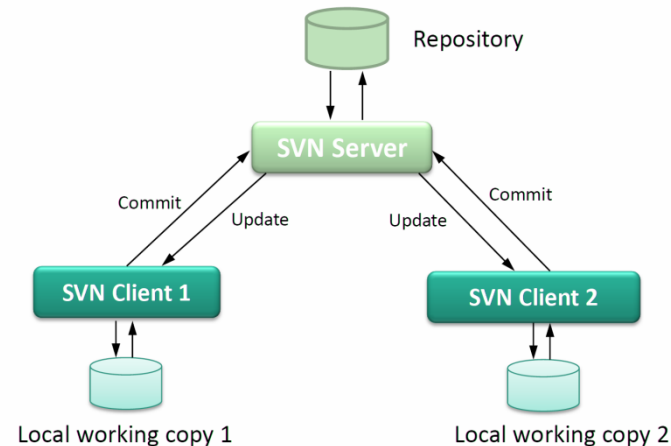
Question

- **The version of the software is 1.01.0012. The number 0012 means.... ?**
 - A. the Major Release Number
 - B. the Minor (Maintenance) Release Number
 - C. the Build Number
 - D. the revision

Question

- **For Eclipse to work with SVN, which tool must you add to Eclipse?**

A. subversive
B. Ant
C. Maker
D. SVN client

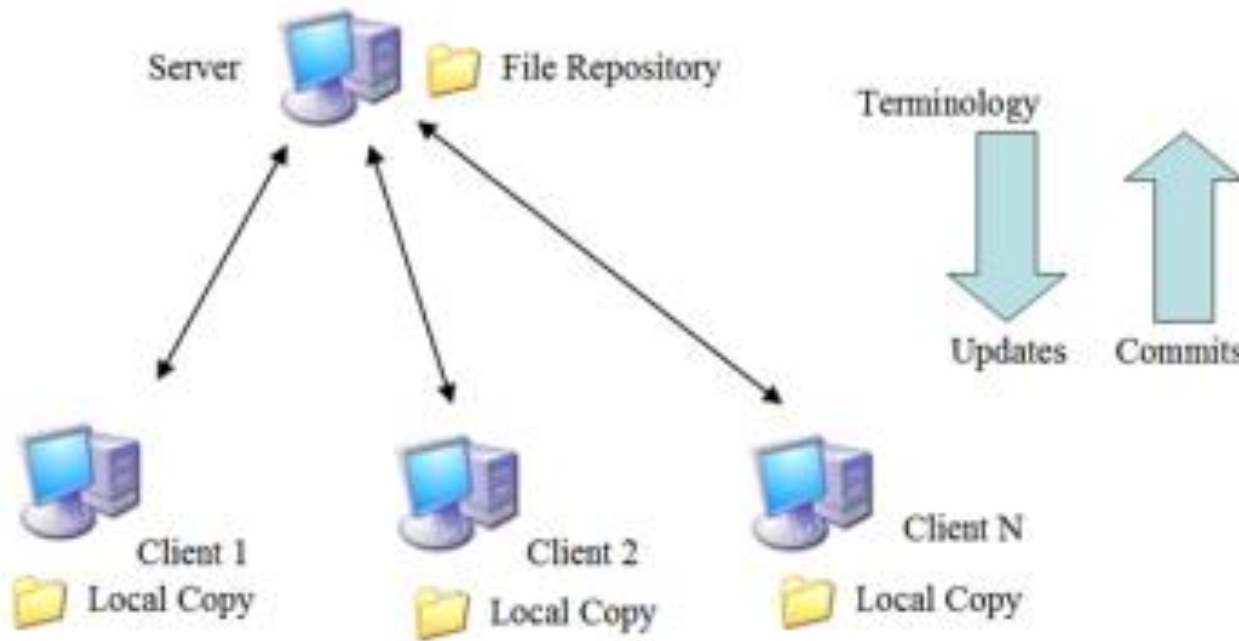


How the Tools Work Together

- Eclipse is an interactive development environment (IDE) used by programmers to edit and debug source code, initiate build and test processes and display results
- SVN is a software configuration management system that stores (keeps revisions of) your project's source files as they are updated by programmers using the Eclipse (or another) IDE

How the Tools Work Together

- Ant is a program that automatically builds your Java project from source files in your SVN repository and creates the project distribution packages



Eclipse

- Eclipse is the most popular open source development environment for the Java programming
 - www.Eclipse.org
- Use Eclipse to write, compile and debug your source code
- Prerequisite: You must have a current version of the Java Development Kit (JDK) installed on your PC or laptop
 - java.sun.com/javase/downloads/index.jsp



Simple Demo Program

- Tells us how much money is in our account
 - 1 class (account)
 - ✓ 3 Fields
 - Account Number
 - Balance
 - Name
 - ✓ 4 Methods
 - Get Account Number
 - Set Balance
 - Get Balance
 - Get Name
 - 1 test case

Account

Account Number

Balance

Name

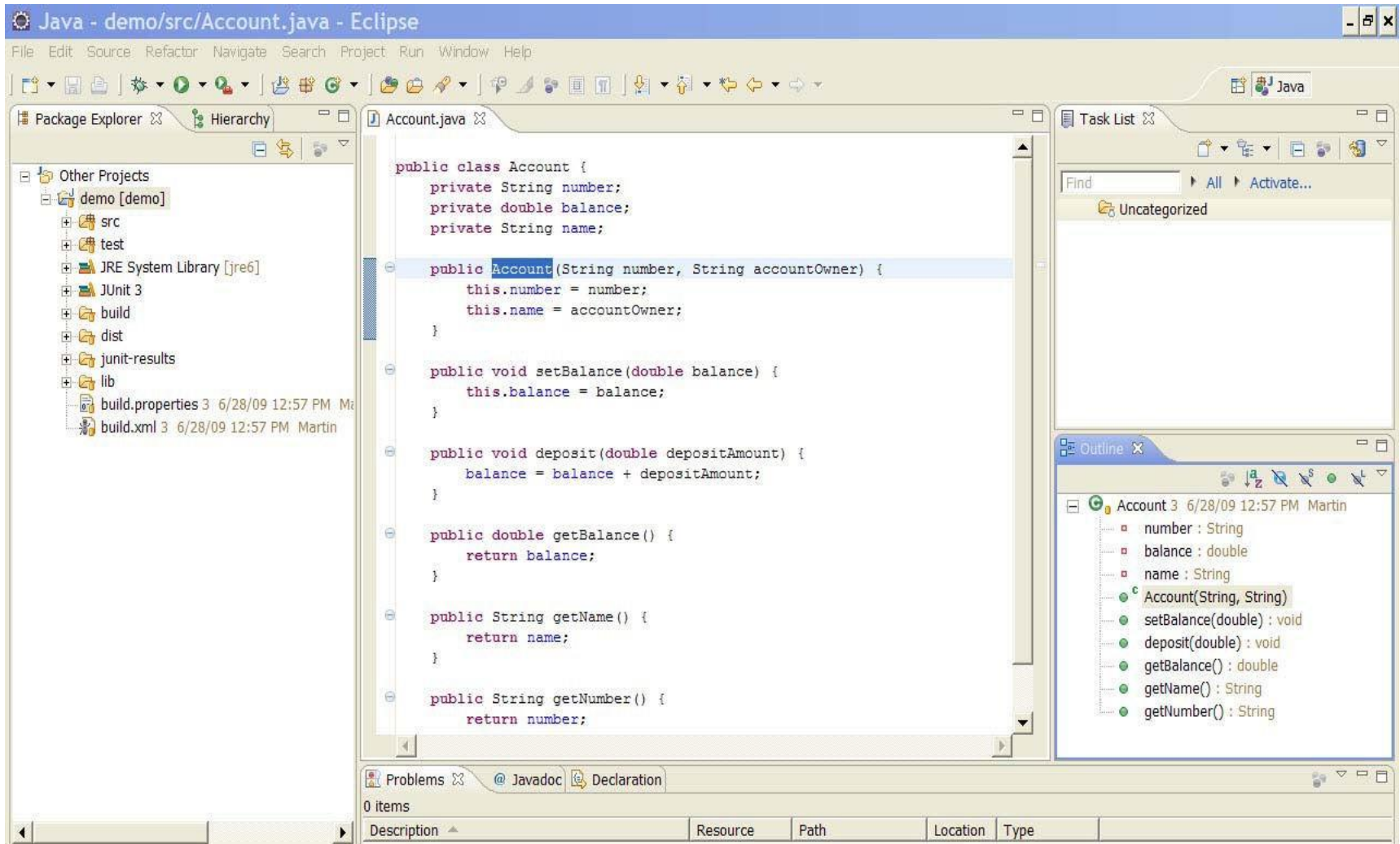
Get Account Number()

Set Balance()

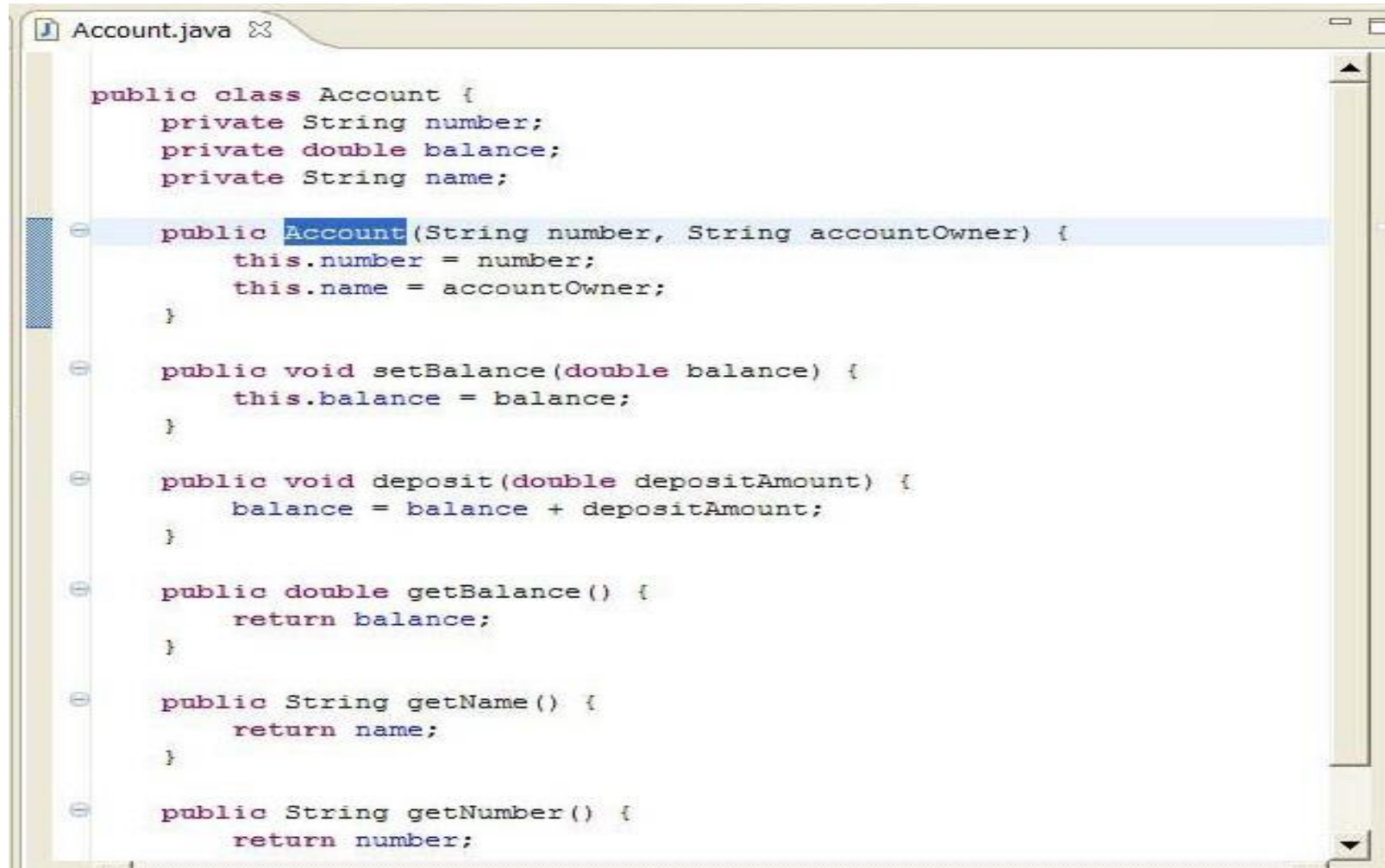
Get Balance()

Get Name()

Eclipse – Demo Project



Eclipse – Account Class



```
Account.java

public class Account {
    private String number;
    private double balance;
    private String name;

    public Account(String number, String accountOwner) {
        this.number = number;
        this.name = accountOwner;
    }

    public void setBalance(double balance) {
        this.balance = balance;
    }

    public void deposit(double depositAmount) {
        balance = balance + depositAmount;
    }

    public double getBalance() {
        return balance;
    }

    public String getName() {
        return name;
    }

    public String getNumber() {
        return number;
    }
}
```

SVN

- Subversion is used to maintain current and historical versions of source code, web pages, documentation and other files
- subversion.tigris.org
- SVN integrates with Eclipse through a plug-in called "subclipse"

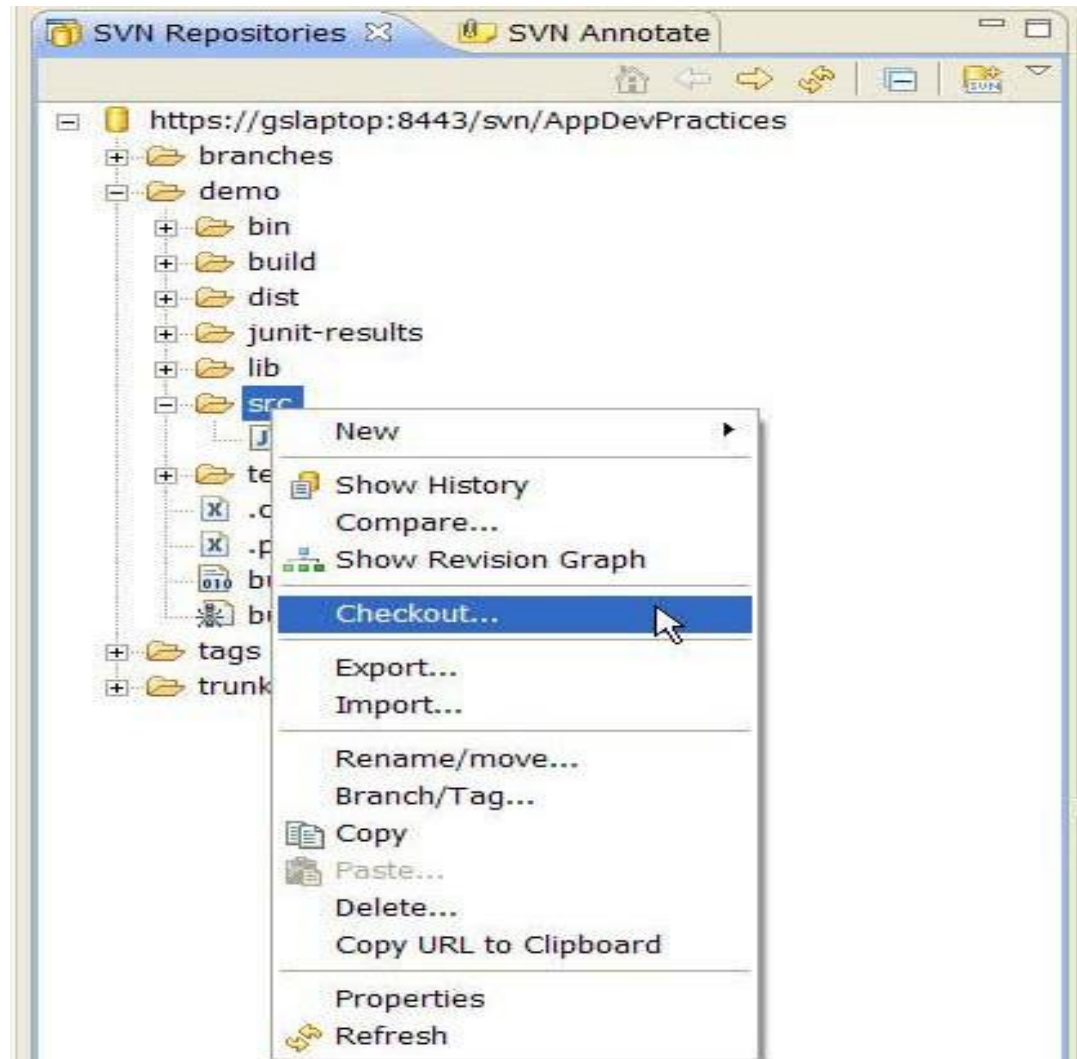


SVN Working Copy

- Your working copy is your own private work area containing a collection of files
 - You can edit these files however you wish, and if they're source code files, you can compile them
 - You can have multiple working copies of the same project
- Populate your working copy with
 - checkout: will populate your working copy with latest code in repository
 - update: will populate your working copy with latest code in repository, but won't overwrite your local changes

Eclipse – SVN checkout

■ Checkout

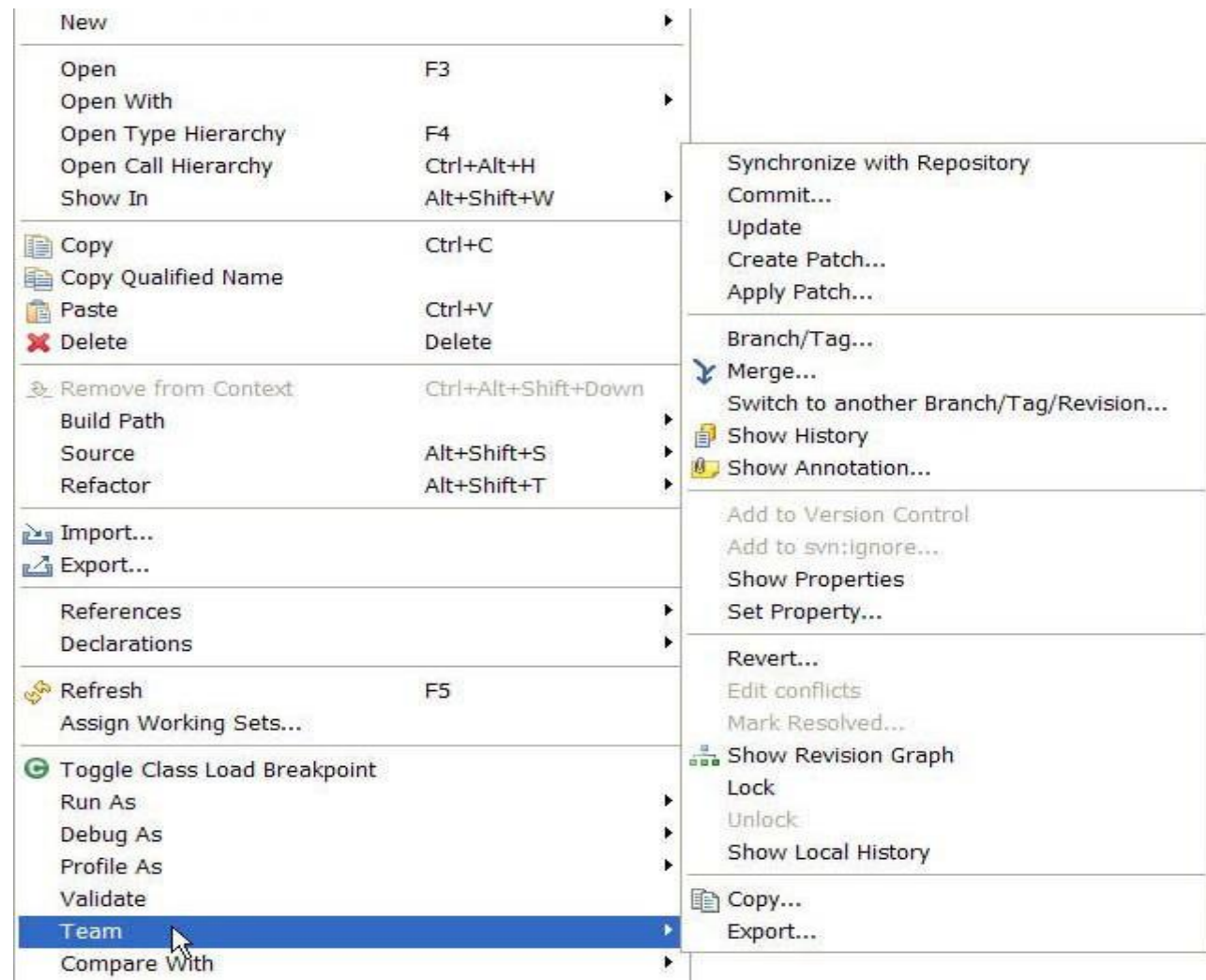


SVN Working Cycle

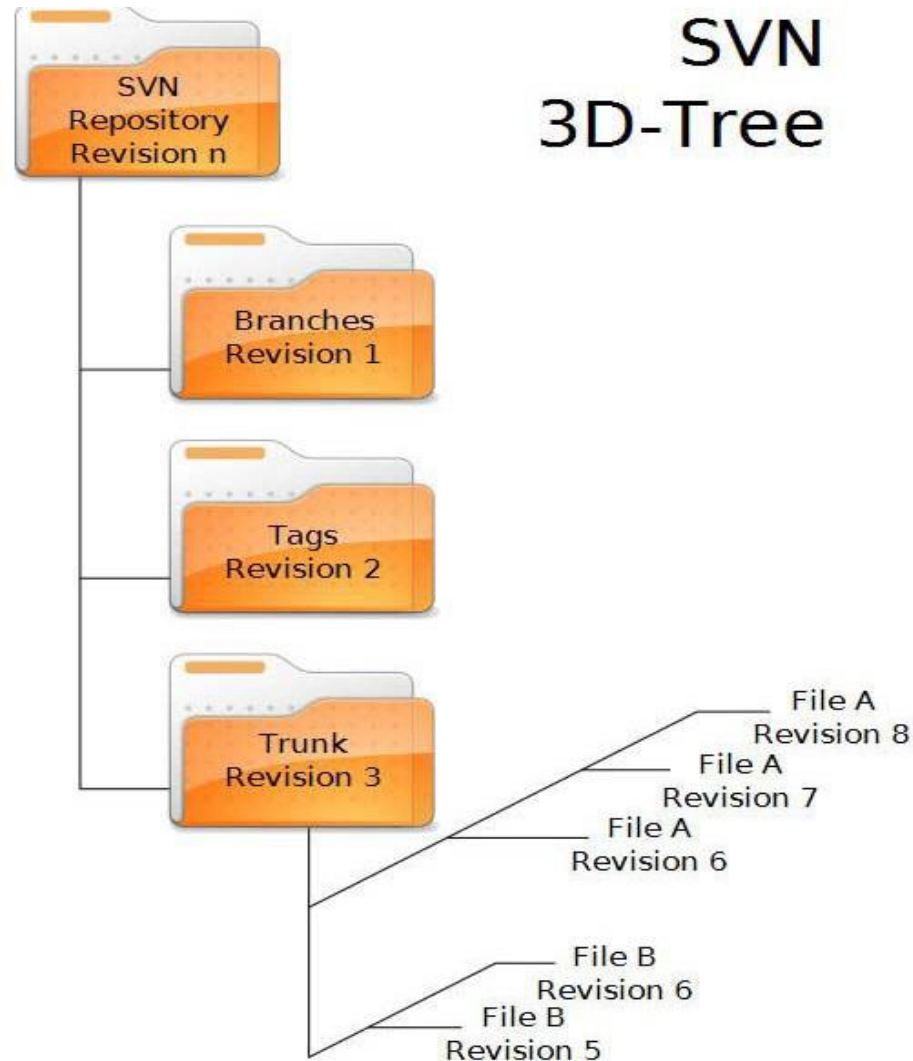
- Checkout of files from repository into working copy
- Local edit of file
- Local unit and integration testing
- Update (to get changed files from repository)
- Local unit and integration testing
- Commit file back to repository
 - Commit will update the repository with your changes

Eclipse – SVN update / commit

■ Checkin



SVN Tree Structure



Building Your Java Product

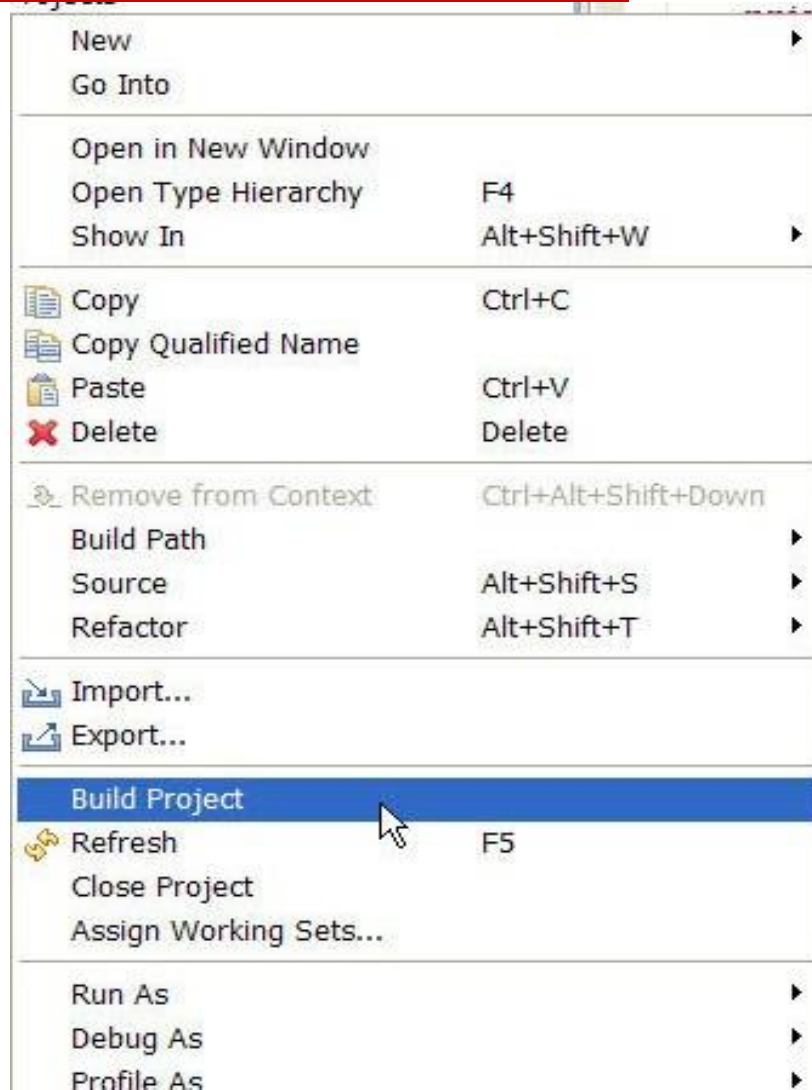
- Simple Java products
 - Simply compile source files
- Complex Java-based projects
 - Retrieve up-to-date source files from source control
 - Manage dependencies not automatically handled by the Java compiler
 - Clean target folders
 - Bundle classes and deliver to appropriate locations (as JAR/WAR files)
- Do you really want to do this manually?

Ant

- Ant is an open source build tool that is the standard for building, packaging and installing Java applications
- Ant uses XML based build files (which makes them much easier to read and modify than “make” files)
- Ant integrates with most Java IDEs
 - Is part of the standard Eclipse download
 - ant.apache.org/



Ant Inside Eclipse



Build Your Java Product with Make

- Make used to be the standard, but ...
 - Makefiles have their own language syntax, requiring highly trained authors
 - Make lacks platform-independence, requiring multiple versions of the same makefile (one for each target platform) to be maintained and distributed
- The time and maintenance required to use Make is too high for complex Java-based projects

Group discussion

- Find 3 revision management tools
- Find 3 website support revision management
- (4 students – 10 minutes)



Summary - 1

- Build, compile and debug software
 - Eclipse
- Build, compile and debug software where source code is stored in a shared repository
 - Eclipse
 - SVN
- Build, compile and debug software where source code is stored in a shared repository and builds are automated
 - Eclipse
 - SVN
 - Ant

Summary - 2

- Different tools for different environments and development languages
 - Eclipse (NetBeans, ...)
 - SVN (CVS, ...)
 - Ant (Make, NAnt, Maven, Phing, Rake, ...)
- Take advantage of the many good open source tools to increase:
 - Your productivity
 - The quality of your code



Video link

- <https://www.youtube.com/watch?v=5ul6VZ7mP10&list=PLnSIu0V9A1pVqs4IwKJ09jWgR7gxoxmfx>
- <https://www.youtube.com/watch?v=wu7p-wqy2D0&list=PLnSIu0V9A1pVqs4IwKJ09jWgR7gxoxmfx&index=5>

References

- <https://subversion.apache.org/docs/>