Introduction to Eclipse
Ed Gehringer
Using (with permission) slides developed by—
Dwight Deugo (dwight@espirity.com)
Nesa Matic (nesa@espirity.com)
Sreekanth Konireddygarri (IBM Corp.)

Overview of Eclipse Lectures
1. Overview Lecture 1
2. Installing and Running
3. Building and Running Java Classes
4. Refactoring
5. Debugging
6. Testing with JUnit
7. Version Control with CVS
Module Road Map

1. Overview
   - Background
   - Architecture
   - Components
   - Usage scenarios
2. Installing and Running
3. Building and Running Java Classes
4. Refactoring
5. Debugging
6. Testing with JUnit
7. Version Control with CVS

Overview » What is Eclipse?

- Eclipse is an open-source project
  - http://www.eclipse.org
  - Consortium of companies, including IBM
  - Launched in November 2001
  - Designed to help developers with specific development tasks
- Consists of four separate projects:
  - Eclipse Project
  - Eclipse Tools Project
  - Eclipse Technology Project
  - Eclipse Web Tools Platform Project
Overview » IBM’s IDE History

1994
VisualAge for Smalltalk

1996
VisualAge for Java

VisualAge Micro Edition

2001
Eclipse Project

Overview » The Eclipse Platform Motivation

- Application development tools construction support
- Independent tool vendors support
- GUI and non-GUI application development support
- Numerous content types support
  - Java, HTML, C, XML, ...
- Easy integration of tools
- Use of Java language for writing the tools
- Multiple operating systems support
Overview » Plug-in Architecture

Eclipse Platform
- Workbench
- Workspace
- Help
- Team
- Tool (plug-in)
- Tool (plug-in)
- Tool (plug-in)
- Tool (plug-in)
- Platform Runtime

Overview » Workbench

- Represents the desktop development environment
  - It contains set of tools for resource management
  - It provides common way of navigating through the resources
- Multiple workbenches can be opened at the same time
Overview » Workspace

- Represents users data
- It is a set of user-defined resources
  - Files
    - Contain arbitrary number of bytes
  - Folders
    - Contain other folders or files
  - Projects
    - Collections of files and folders

Overview » Help

- Used for creating and publishing documentation
- There are two different documentation styles:
  - Help style documentation is published in the user guide
  - API documentation is published in the programmer guide
- Help content is in HTML format
- Help navigation is in XML format
Overview » Team

- Provides support for:
  - Versioning
  - Configuration management
  - Integration with team repository
- Allows team repository provider to hook into the environment
  - Team repository providers specify how to intervene with resources
- Has optimistic and pessimistic locking support

Overview » How is Eclipse Used?

- As an IDE - Integrated Development Environment
  - Supports the manipulation of various content types
  - Used for writing code
- As a product base
  - Supported through plug-in architecture and customizations
Overview » Eclipse as an IDE

- Java Development Tooling (JDT) is used for building Java code
- Provides set of workbench plug-ins for manipulating Java code
  - Java projects, packages, classes, methods, ....
- Java compiler is built in
  - Used for compiling Java code
  - Creates errors (special markers of code) if compilation fails

Overview » Eclipse as a Product Base

- Eclipse can be used as a Java product base
- Its flexible architecture used as a product framework
  - Reuse plug-in architecture
  - Create new plug-ins
  - Customize the environment
Module Road Map

1. Overview
2. Installing and Running Eclipse
   - Where to get Eclipse?
   - What is the support for Eclipse?
   - Installing Eclipse
   - Running Eclipse
3. Building and Running Java Classes
4. Refactoring
5. Debugging
6. Testing with JUnit
7. Version Control with CVS

Installing and Running Eclipse » Getting Eclipse
Installing and Running Eclipse » http://www.eclipse.org

- Main point for finding Eclipse resources
  - Downloads
  - Articles
  - Discussion groups
  - Bugs
- Contains various resources for using Eclipse
- Contains references to other Eclipse related sites

Installing and Running Eclipse »
Downloading Eclipse’s Install Zip File

- Click on the Download from the main page on http://www.eclipse.org

---

**Eclipse Europe Packages**

- **Eclipse IDE for Java Developers**: Windows (78 MB)
  - The essentials for any Java developer, including a Java IDE, C/C++ editor, HTML, SQL and more.

- **Eclipse IDE for Java EE Developers**: Windows (12 MB)
  - Tools for Java developers creating EJ and Web applications, including a Java IDE, tools for Eclipse, JEE, SQL and more.

- **Eclipse IDE for C/C++ Developers**: Windows (82 MB)
  - An IDE for C/C++ developers.

- **Eclipse for RCP/Plug-in Developers**: Windows (53 MB)
  - A complete set of tools for developers who need to create Eclipse plug-ins or Rich Client Applications. It includes a complete IDE, developer tools and source code.

- **Eclipse Classic**: Windows (140 MB)
  - The classic Eclipse, download the Eclipse Platform, Java Development Tools, and Plug-in Development Environment, including source code and documentation.
Installing and Running Eclipse » Installing Eclipse

- Unzip the downloaded file to the directory of your choice

![Unzipped Eclipse Directory]

Installing and Running Eclipse » eclipse.exe

- Eclipse is run by double-clicking on the eclipse.exe file
  - The file is located in the Eclipse installation directory
  - If there is no path set for javaw.exe program, the following dialog will come up

![Eclipse Dialog]

v1.6 08/02/2006
Installing and Running Eclipse » Missing a Java VM?

- Download the latest Java Runtime Environment (JRE), http://java.sun.com/javase/downloads/index.jsp

Installing and Running Eclipse » Running Eclipse

- When Eclipse is run, a Workbench opens
Installing and Running Eclipse » Default Workspace

- The default workspace is created when Eclipse runs for the first time
  - The workspace is located under \workspace\ directory of Eclipse installation directory
  - For example c:\eclipse\workspace
- The workspace contains user defined data – projects and resources such as folders and files

Installing and Running Eclipse » Running Different Workspace …

- It is possible to run workspace other than default
  - -data argument must be used with eclipse.exe
  - Workspace location must be specified
- Useful for grouping project specific data
- Multiple workspaces can run at the same time
Installing and Running Eclipse » Running Different Workspace ...

- Customize your working directory by creating a shortcut identifying the eclipse.exe and the working directory

Installing and Running Eclipse » Module Summary

- In this module you have learned:
  - What Eclipse is, its background and components
  - How Eclipse is used
  - How to download, install and run Eclipse
  - How to create and run multiple workspaces with Eclipse
Module Road Map

1. Overview
2. Installing and Running
3. Building and Running Java Classes
   - Developing Java applications
   - Projects, packages, classes
   - Browsing Java code
   - Searching Java code
   - Organizing Java code
   - Using Code Assist
   - Running Java applications
   - Scrapbook
4. Refactoring
5. Debugging
6. Testing with JUnit
7. Version Control with CVS

Java Development Tooling - JDT

- Eclipse’s Java Development Environment is often referred to as JDT – Java Development Tooling
  - Using the JDT you can do following with the Java programs:
    - Write
    - Compile
    - Test
    - Debug
When developing Java code commonly used perspectives are:

- **Java Perspective**
  - Designed for working with Java projects

- **Java Browsing Perspective**
  - Designed for browsing structure of Java projects

- **Java Type Hierarchy Perspective**
  - Designed for exploring type hierarchy

- **Debug Perspective**
  - Designed for debugging Java programs

---

**Java Perspective**

- **Contains:**
  - Editor area
  - Package Explorer View
  - Hierarchy View
  - Outline View
  - Problems View
Building and Running Java Classes »
Java Browsing Perspective

- Contains:
  - Editor area
  - Projects View
  - Packages View
  - Types View
  - Members View

Building and Running Java Classes »
Java Type Hierarchy Perspective

- Contains editor area and Hierarchy View
Building and Running Java Classes » New Project Preferences

- You can set global preferences for a project
- Select Window » Preferences to get Preferences View
- Good idea to separate your Java files into source and compiled directories (src and bin)
- This action only needs to be done once
- Done for all subsequent projects

---

Building and Running Java Classes » Creating Java Project

- Project used to organize resources (source, class, icons) for a project
- To create Java project:
  - Select File » New Project… from the menu
  - The New Project wizard comes up
  - Select Java Project
  - Click Next
Building and Running Java Classes » New Project Wizard

- Specify Project Name
- Click Next

Building and Running Java Classes » Java Settings

- Specify Java settings
  - Output folder (where compiled files will be stored)
  - Click on Libraries tab to specify dependencies such as
    - External jar files project depends on
    - Classes from other projects that are referenced in the project
- Click Finish
Building and Running Java Classes » Project Properties

- You can change the Java build path at any time
  - Choose Properties from the context menu on the project

```
Properties for HelloWorldProject

Java Build Path
- External Tool Build
- Build Path
- Source Location
- Java Classpath
- Project References

Build Path:
- HelloWorldProject

Default output folder:
- HelloWorldProject

Button:
- OK
- Cancel
```

Building and Running Java Classes » Creating Packages

- Package contains Java class files
- To create a package for a project:
  - Select the project in the Package Explorer
  - Choose New » Package from the context menu
  - New Java Package window appears
  - Specify package name
  - Click Finish

```
New Java Package

Java Package:
Create a Java package.

Created/folders corresponding to packages:
Source Folder:
HelloWorldProject
```
To create a class in a package:
- Select the package in the Package Explorer
- Choose New » Class from the context menu
- The Class wizard comes up
- Specify class details (Name, Modifiers & Superclass)
- Click Finish

Exercise 1

- Create a new Java application Project titled EgApp.
- Create a new Package csc517 in the Project EgApp.
- Create a new class titled MainClass with the main() method in the Package csc517.