

Developing Java in Eclipse: Part I of III

SHARE Orlando, February 2008

Oliver Fenton

Java Technology Center, IBM Hursley Labs,
Winchester, UK

Session: 8358



Trademarks



IBM, AIX, CICS, DB2, IMS, z/OS, OS/390, S/390, System/390, VisualAge, WebSphere Application Server, WebSphere Studio, z/VM - are trademarks or registered trademarks of the IBM Corporation

Sun, Sun Microsystems, JavaSoft, Java, JavaBeans, JDK, Java 2 Micro Edition, J2ME, Java 2 Standard Edition, J2SE, Java 2 Enterprise Edition, J2EE - are trademarks or registered trademarks of Sun Microsystems Inc.

Agenda

❖ Introduction

- Eclipse Overview and History
- Eclipse Architecture and plug-ins
- Perspectives
- Projects

❖ The Java Perspective

❖ Eclipse Java Projects

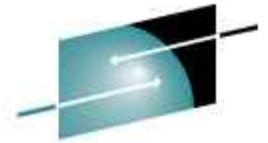
❖ Exercise

❖ Further Help and Additional Resources

Eclipse Overview and History

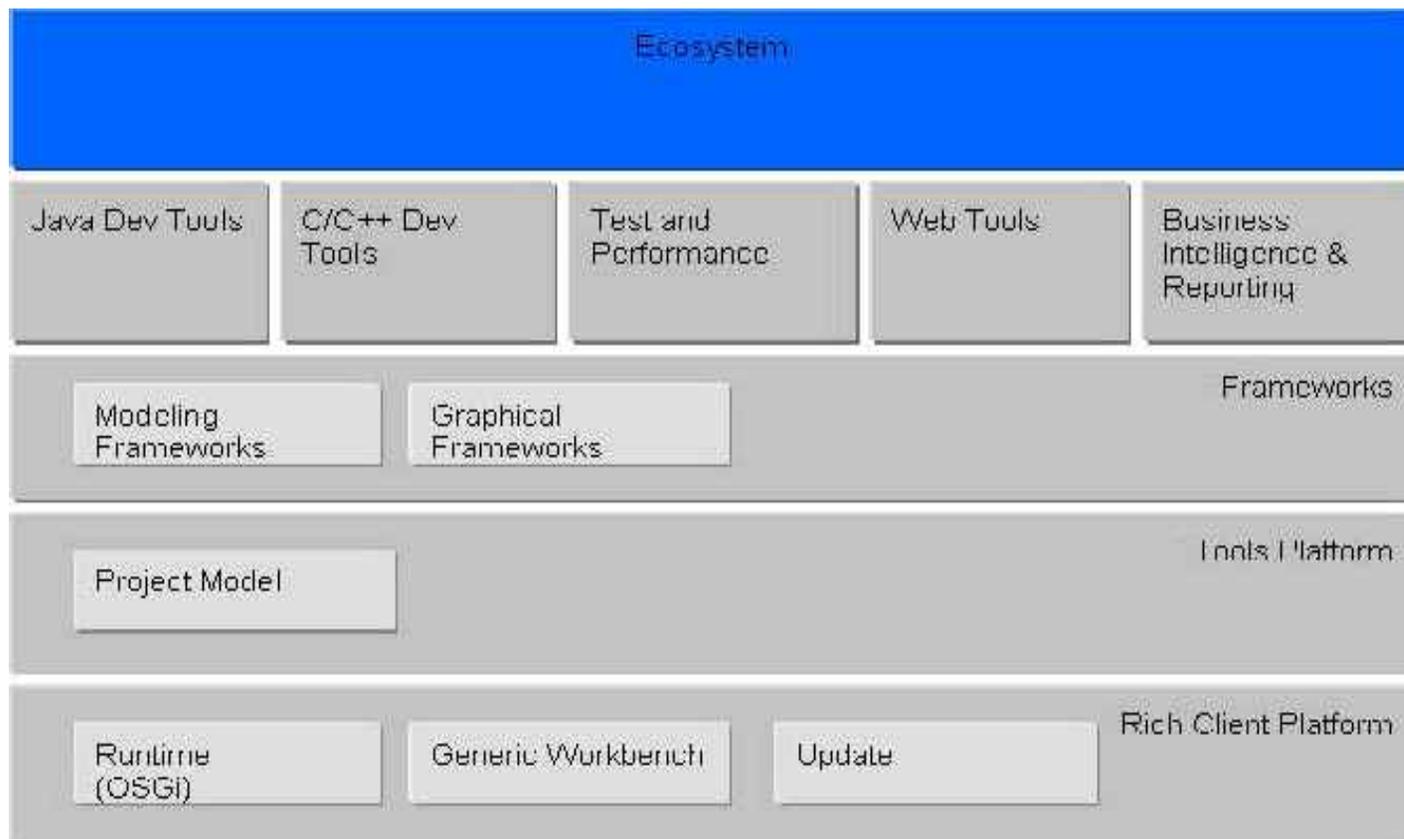


- ❖ Eclipse is an extensible Integrated Development Environment (IDE) based on a plug-in architecture
- ❖ Eclipse is open source, released under a common public license (CPL)
- ❖ Originally developed by IBM, version 1.0 being released in November 2001, Eclipse is now under the stewardship of an independent consortium including: IBM, SAP, Sybase, Fujitsu, HP, BEA...
- ❖ Available as a free download from www.eclipse.org

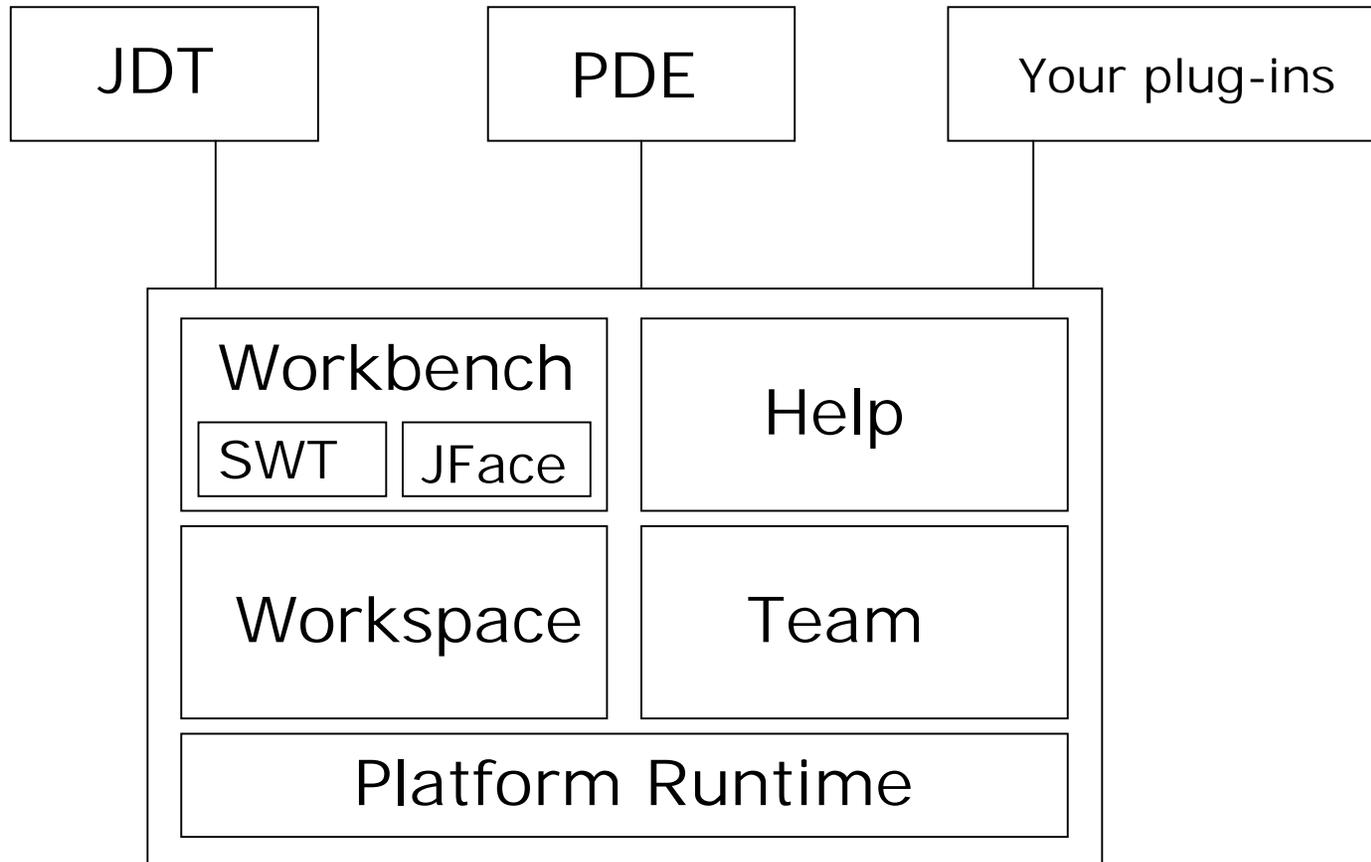


SHARE
Technology • Connections • Results

Uses of Eclipse



Eclipse Architecture



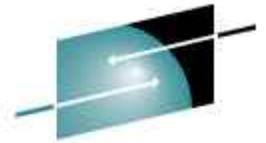
Plug In's

- ❖ Eclipse is a framework which enables plug-in operability and interoperability
- ❖ Click on 'Help>About Eclipse SDK' then 'Plug-in Details' to see currently active Plug-ins



- ❖ Eclipse plug-ins are available from a number of sources including www.eclipseplugincentral.com

Perspectives



SHARE
Technology • Connections • Results

- ❖ Eclipse perspective is a pre-selected set of views arranged in a predetermined way.
- ❖ Included with Eclipse are several pre-customized perspectives including:
 - Java
 - Debug – session 8355 (Remote z/OS Debugging)
 - Testing – session 8384 (JUnit & Test Driven Development)
 - Plug-in Development
 - Team Synchronizing

Projects



- ❖ Eclipse projects contain folders and files like in a hierarchical file system
- ❖ An Eclipse project is a collection of resources (including code) that combine to make an application

Agenda

❖ Introduction

- Eclipse Overview and History
- Eclipse Architecture and plug-ins
- Perspectives
- Projects

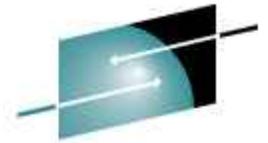
❖ The Java Perspective

❖ Eclipse Java Projects

❖ Exercise

❖ Further Help and Additional Resources

Perspectives and Eclipse 1



SHARE
Technology • Connections • Results

The screenshot shows the Eclipse IDE interface. The title bar reads "Resource - RunIt.java - Eclipse SDK". The menu bar includes File, Edit, Source, Refactor, Navigate, Search, Project, Run, Window, and Help. The toolbar contains various icons for file operations and development. The Navigator view on the left shows a project named "Lab" with a folder "eclipseLab" containing files "MemGrab.class", "MemGrab.java", "RunIt.class", "RunIt.java", ".classpath", and ".project". The Outline view below it shows the package structure: "eclipseLab", "import declarations", "eclipseLab.MemGrab", and "RunIt" with a sub-entry "main(String[])". The main editor window displays the source code for "RunIt.java":

```
/**
 *
 */
package eclipseLab;

import eclipseLab.MemGrab;

public class RunIt {

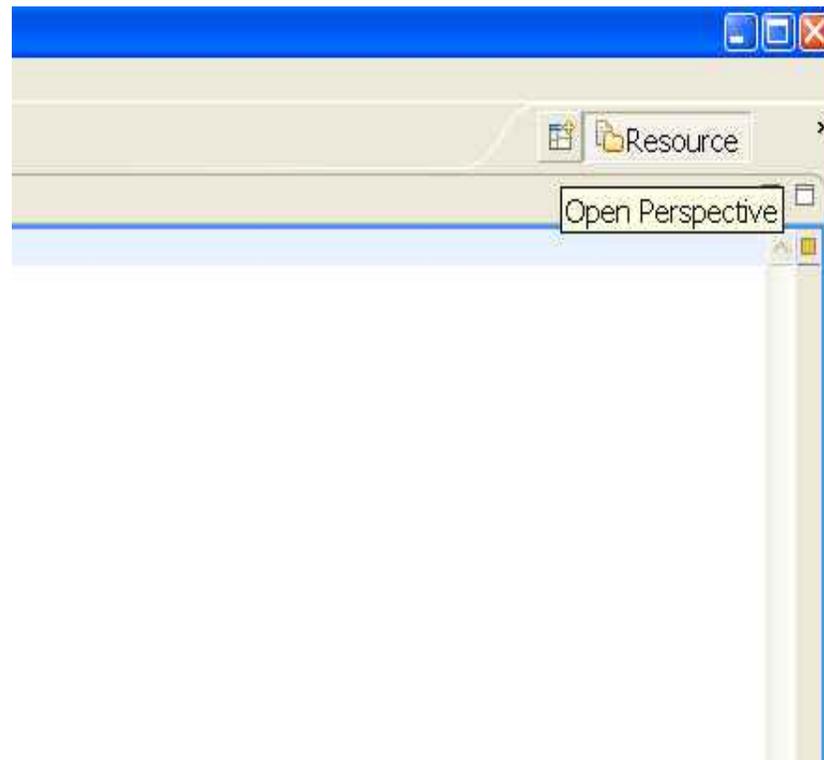
    /**
     * @param args
     */
    public static void main(String[] args) {
        int i=0;
        System.out.println("Lab1");
        while (i++ < 1000) {
            MemGrab aGrab = new MemGrab();
            // TODO demonstrate task
        }
    }
}
```

The Tasks view at the bottom right shows one item:

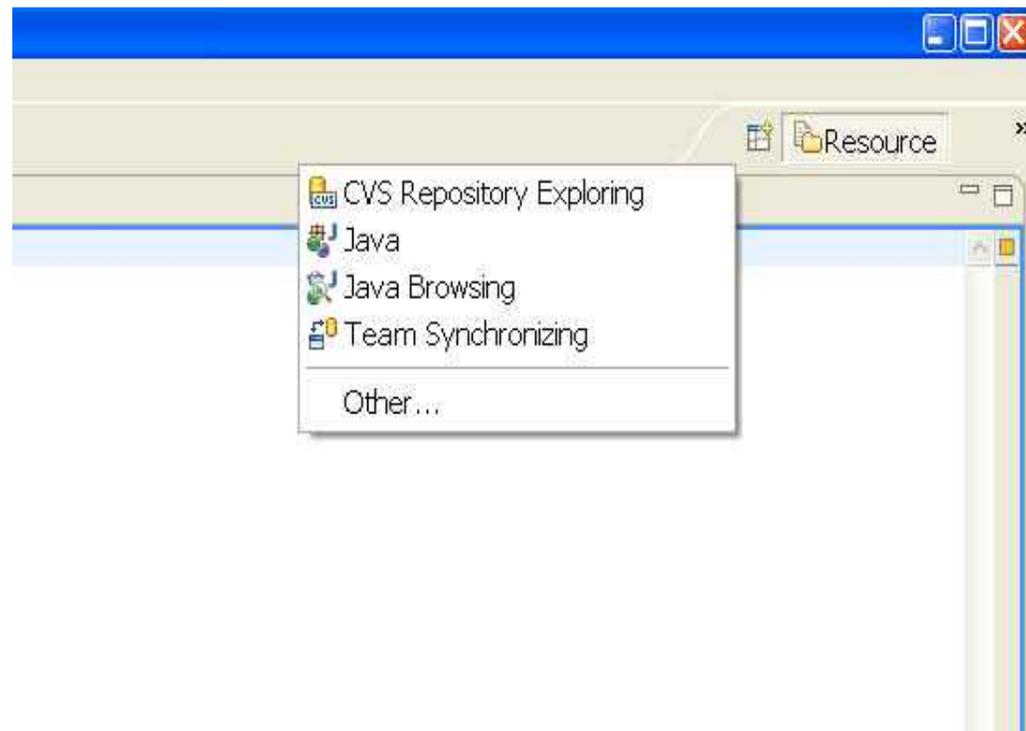
✓ !	Description	Resource	In Folder	Loca...
<input type="checkbox"/>	TODO demonstrate task	RunIt.j...	Lab/eclipseLab	line 20

The status bar at the bottom indicates "Writable", "Smart Insert", and "1 : 1".

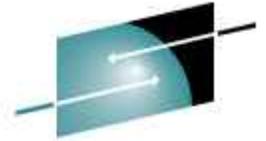
Perspectives and Eclipse 2



Perspectives and Eclipse 3



Java Perspective



SHARE
Technology • Connections • Results

The screenshot displays the Eclipse IDE in the Java perspective. The main editor window shows the source code for `RunIt.java` in the `eclipseLab` package. The code includes an import for `MemGrab` and a `main` method that prints "Lab1" and creates a `MemGrab` instance in a loop. The Package Explorer on the left shows the project structure with `RunIt.java` selected. The Outline view on the right shows the class structure. The Problems view at the bottom shows two warnings: "The local variable buf is ne..." at line 15 and "The local variable aGrab is..." at line 19.

```
package eclipseLab;

import eclipseLab.MemGrab;

public class RunIt {

    /**
     * @param args
     */
    public static void main(String[] args) {
        int i=0;
        System.out.println("Lab1");
        while (i++ < 1000) {
            MemGrab aGrab = new MemGrab();
            // TODO demonstrate task
        }
    }
}
```

Description	Resource	In Folder	Loca...
⚠ The local variable buf is ne...	MemGr...	Lab/eclipseLab	line 15
⚠ The local variable aGrab is...	RunIt.j...	Lab/eclipseLab	line 19

Agenda

❖ Introduction

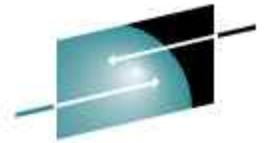
- Eclipse Overview and History
- Eclipse Architecture and plug-ins
- Perspectives
- Projects

❖ The Java Perspective

❖ Eclipse Java Projects

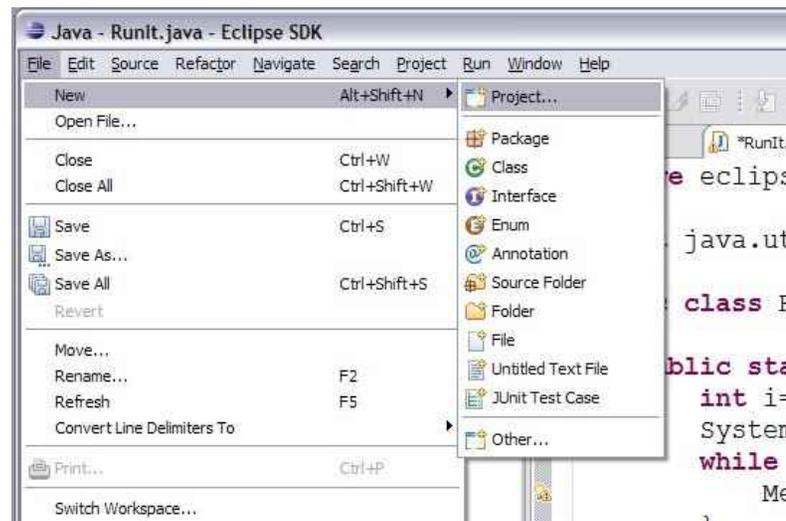
❖ Exercise

❖ Further Help and Additional Resources



Java Projects

- ❖ An Eclipse project is a collection of code and resources that combine to make an application
- ❖ Each project within a workspace can have different settings



Creating a New Java Project 1



Creating a New Java Project 2



New Java Project

Create a Java project
Create a Java project in the workspace or in an external location.

Project name:

Contents

Create new project in workspace
 Create project from existing source

Directory:

JRE

Use default JRE (Currently 'j9vmwi3223-20070426-sdk') [Configure JREs...](#)
 Use a project specific JRE:

Project layout

Use project folder as root for sources and class files
 Create separate source and output folders [Configure default...](#)

Creating a New Java Project 3



Exercise



- ❖ Creating a project, package and class
 - Example program is CountPrinter.java
- ❖ Code assist, identifying errors, code completion
- ❖ Running code

Some Cool Features of Eclipse

- ❖ Automatic building
- ❖ Classpath resolution
- ❖ Real-time error checking
- ❖ Code assist
- ❖ Quick-Fix
- ❖ Refactoring
- ❖ Declarations & References

Review

❖ Introduction

- Eclipse Overview and History
- Eclipse Architecture and plug-ins
- Perspectives
- Projects

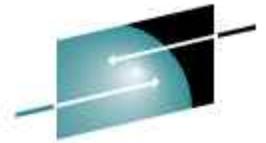
❖ The Java Perspective

❖ Eclipse Java Projects

❖ Exercise

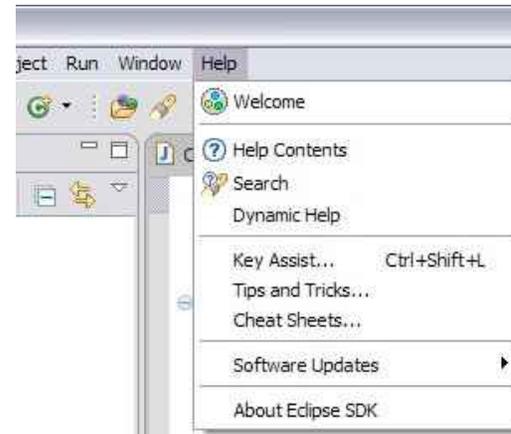
❖ Further Help and Additional Resources

Further Help and Additional Resources



SHARE
Technology • Connections • Results

❖ Eclipse in-built help



- ❖ www.eclipse.org
- ❖ www.eclipseplugincentral.com
- ❖ www.google.com