References

The following slides are extracted from

- a courseware which is part of the Open Source "ECESIS" project
  - The courseware is produced by, and is copyrighted by Espirity, Inc. and CMA
  - To get the whole course material go to
    - http://www.eclipse.org/ecesis/
- the Eclipse Project FAQ page
- Basic tutorial by Adam Kiezun
    - http://www.eclipse.org/

Module Road Map

- **Overview**
  - **Background**
  - **Architecture**
  - **Components**
  - **Usage scenarios**
- Installing and Running
- Workbench
- Building and Running Java Classes

What is Eclipse?

- Eclipse is an open source project
  - http://www.eclipse.org
  - Consortium of companies
    including Borland, IBM, Rational, Red Hat
  - Launched in November 2001
  - Designed to help developers with specific development tasks
- Consists of three separate projects:
  - Eclipse Project
  - Eclipse Tools Project
  - Eclipse Technology Project

The Eclipse project

- The Eclipse Project is an open source software development project dedicated to providing a robust platform for the development of highly integrated tools.
- It is composed of three subprojects:
  - Platform,
  - JDT - Java development tools,
  - PDE - Plug-in development environment.

The Eclipse Tools Project

- The mission of Eclipse Tools Project is to foster the creation of a wide variety of tools for the Eclipse Platform.
  - The Tools Project provides point of coordination for open source tool developers in order
    - to minimize overlap and duplication,
    - ensure maximum sharing and creation of common components,
    - and promote seamless interoperability between diverse types of tools.
  - The Tools project is composed of subprojects which are proposed, selected and developed by the community of tool developers and the Tools Project PMC.

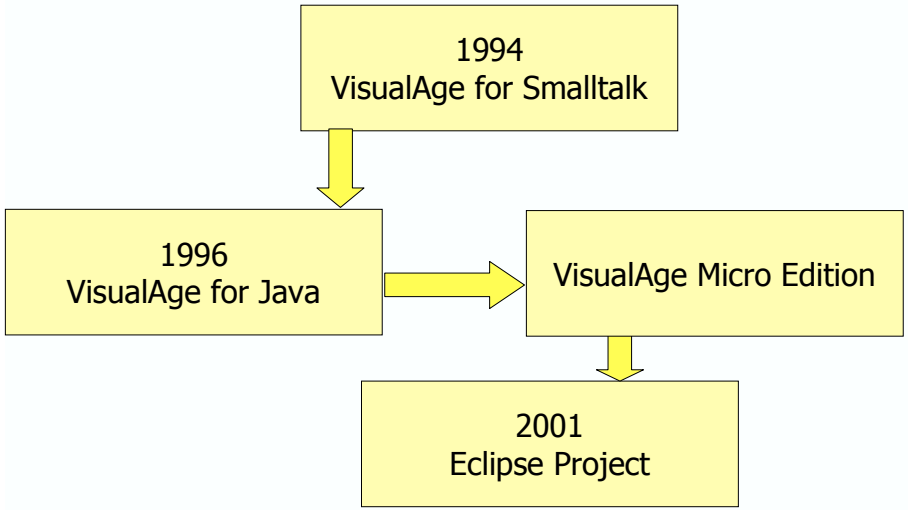
The Eclipse Technology Project

- The mission of the Eclipse Technology Project is to provide new channels for open source developers, researchers, academics and educators to participate in the on-going evolution of Eclipse.
- It is organized as three related project streams
  - Research: explore research issues in Eclipse-relevant domains such as programming languages and development environments;
  - Incubators: small, informally structured projects which add new capabilities to the Eclipse software base;
  - Education: projects focus on the development of educational materials, teaching aids and courseware.

The Eclipse SDK

- The Eclipse SDK (software development kit) is the consolidation of the components produced by the three Eclipse Project subprojects
  - Platform,
  - JDT - Java development tools,
  - PDE - Plug-in development environment

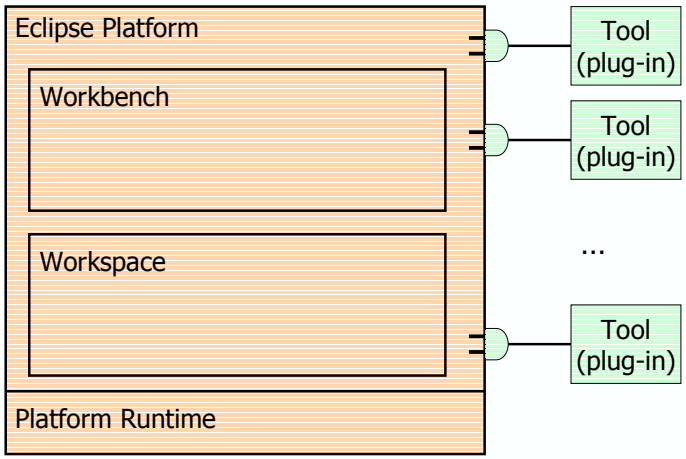
IBM's IDE History



The Eclipse Platform Motivation

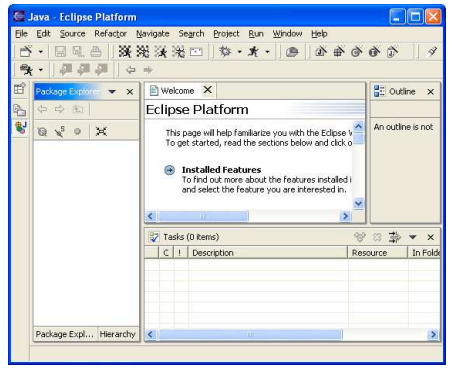
- Application development tools construction support
- GUI and non-GUI application development support
- Numerous content types support
  - Java, HTML, C, C++, XML, ...
- Easy integration of tools
- Use of Java language for writing the tools
- Multiple operating systems support

Plug-in Architecture



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



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

How is Eclipse Used?

- As an IDE - Integrated Development Environment
  - Used for writing code
- As a product base
  - Supported through plug-in architecture and customizations

Eclipse as an IDE

- Java Development Tool (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

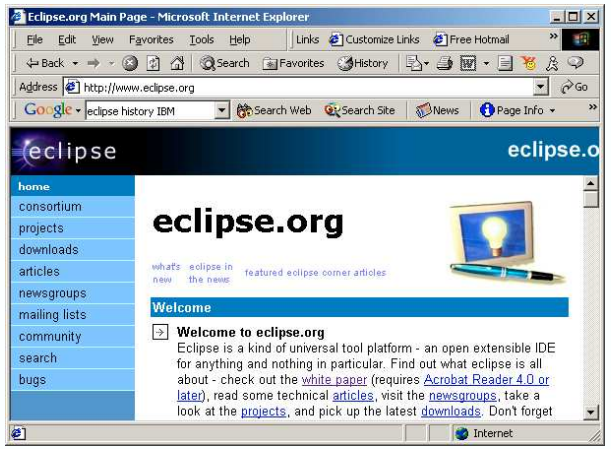
Eclipse as a Product Base

- 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

- Overview
- **Installing and Running Eclipse**
  - **Where to get Eclipse?**
  - **What is the support for Eclipse?**
  - **Installing Eclipse**
  - **Running Eclipse**
- Workbench
- Building and Running Java Classes

Getting Eclipse



http://www.eclipse.org

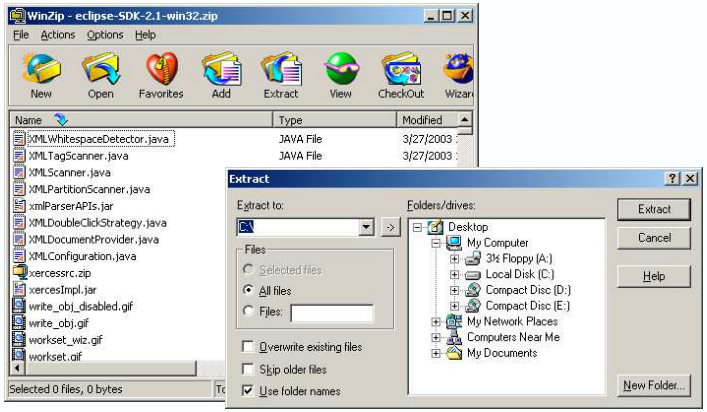
- Main point for finding Eclipse resources
  - Downloads
  - Articles
  - Discussion groups
  - Bugs
- Contains various resources for using Eclipse
- Contains references to other Eclipse related sites

Downloading Eclipse's Install Zip File

- Click on the Download from the main page on http://www.eclipse.org
  - Choose the closest site from which to download (geographical sites hosting the download)
  - Choose the build for download (usually the latest build)
  - Choose the platform for download and type of download (http or ftp)
  - Specify where to save the download locally

Installing the Eclipse

- Unzip the downloaded file to the directory of your choice



eclipse.exe

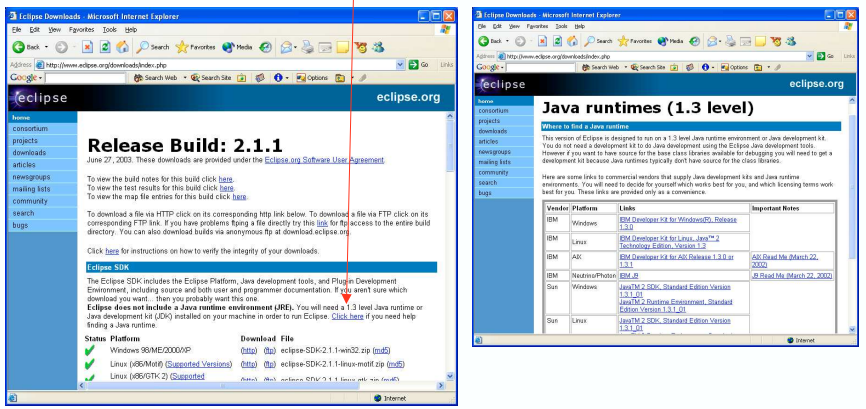
- Eclipse is run by double-clicking on the eclipse.exe file
  - The file is located in the Eclipse installation directory
- If there is no path set for javaw.exe program, the following dialog will come up

Missing a Java VM? …

- Download the latest Java Runtime Environment (JRE), e.g., v1.4.1_02 from http://java.sun.com/j2se/
- Click on the downloaded EXE file to install
- When given the option by the installer, identify a directory of your choice for the install files
- Update the PATH environment variable to include the JRE's bin directory

… Missing a Java VM?

- Other VMs are available at the Eclipse download web site



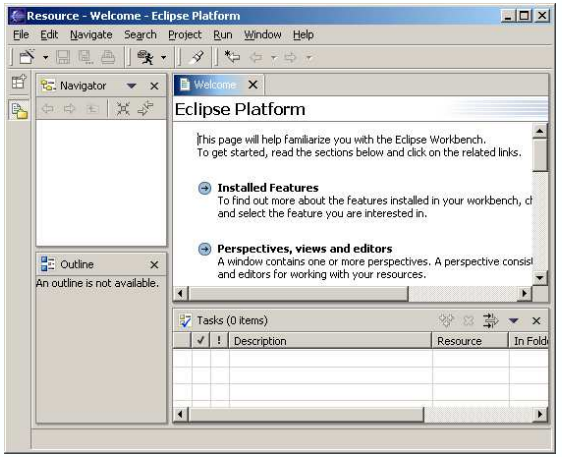
Specifying JVM

- There are two ways of specifying JVM for Eclipse:
  - By installing JVM under the \jre\ directory off the eclipse install directory
  - By specifying existing JVM in the PATH environment variable

Start -> Control Panel -> System ->Advanced -> Environment Variables

Running Eclipse

- When Eclipse is run, a Workbench opens



Module Road Map

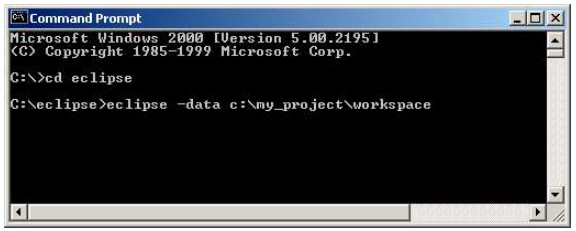
- Overview
- Installing and Running
- **Workbench**
  - **Basics**
  - **Resources**
  - **Components**
    - **Perspectives, Views and Editors**
  - **Specialized Views**
  - **Customization**
- Building and Running Java Classes

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

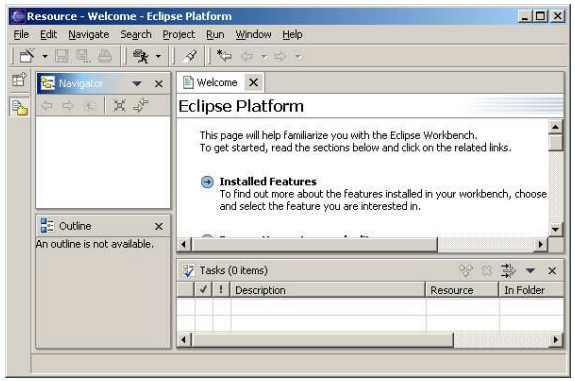
Running Different Workspace

- 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



What is the Workbench?

- Starting point for the development with Eclipse
  - Opens up when Eclipse starts
  - Represents the working environment in Eclipse



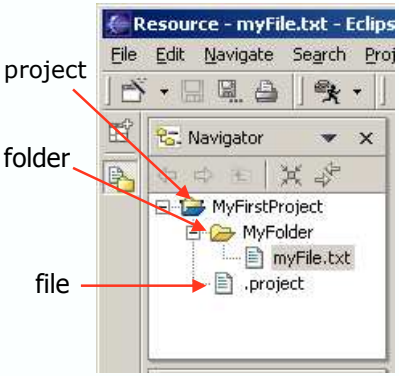
Multiple Workbench Instances

- Instance of the workbench comes up when Eclipse launched
- It is possible to open another instance of the Workbench
  - Window → New Window
  - This opens up a new Workbench window
  - This is important when developing Eclipse plug-ins and you want to test them within the environment
    - When testing plug-ins new instance of Workbench opens with developed plug-ins

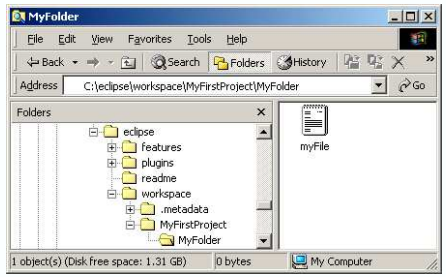
Resources in a Workbench

- When working with Eclipse, you work with its resources
- Resources are organized as a file/directory structure in the Workbench
  - They correspond to the actual files and directories in the Workspace
  - There are three different levels on resources:
    - Projects
    - Folders
    - Files

Organizing Resources



Workbench

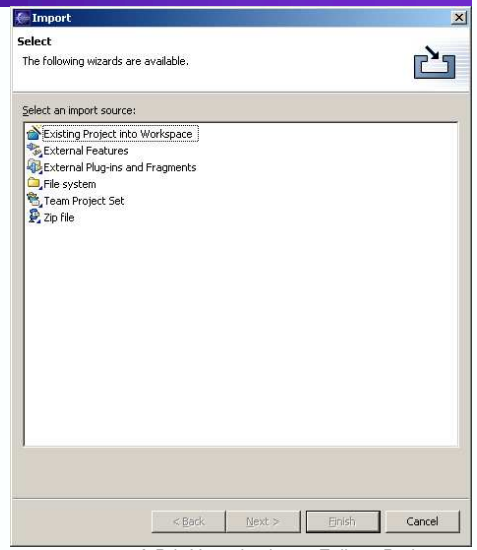


Workspace

- It is possible to drag and drop resources directly between Workbench and the directory structure

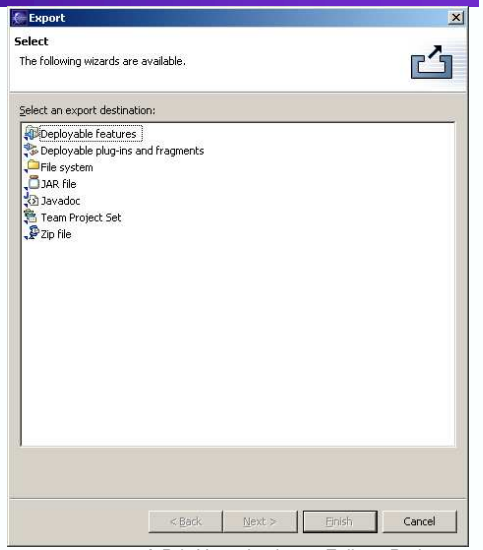
Importing Resources

- Available through menu option File → Import...
- Different source of resources can be imported:
  - Existing projects
  - Files (directory structure)
  - Zip files



Exporting Resources

- Available through menu option File → Export...
- Different source of resources can be exported:
  - File (directory structure)
  - Zip files
  - Jar files



Refreshing Workbench

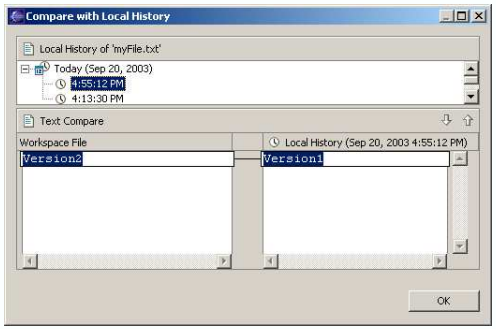
- Used for refreshing resources that change in the Workspace directly
- For example, if file is added to the directory structure in the Workspace:
  - Select the project
  - Choose Refresh from the context menu
    - This will bring the added resource into the Workbench

Resource History

- All resource versions are stored in local history
- Each resource version is identified by a time stamp
- This allows you to compare different versions of the resource
- There are two pop-up menu options available for accessing local history (the resource is selected):
  - Compare With → Local History…
  - Replace With → Local History…
  - Both options bring up the same window

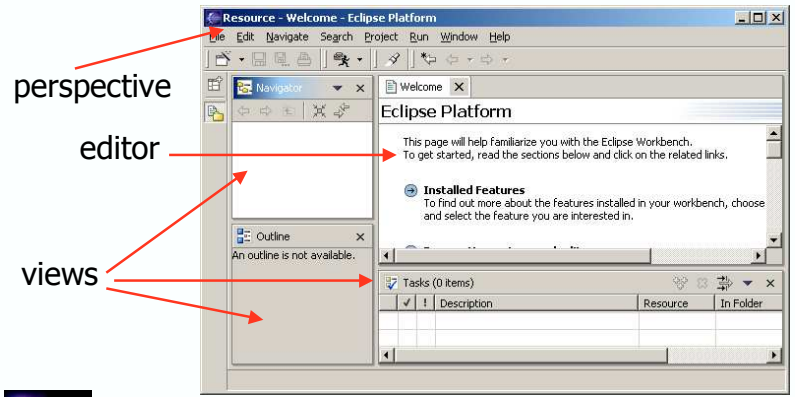
Comparing Files

- Allows you to view differences between different versions
  - You can move between differences
  - Differences are selected



Workbench Components

- Workbench contain perspectives
- Perspective contain views and editors

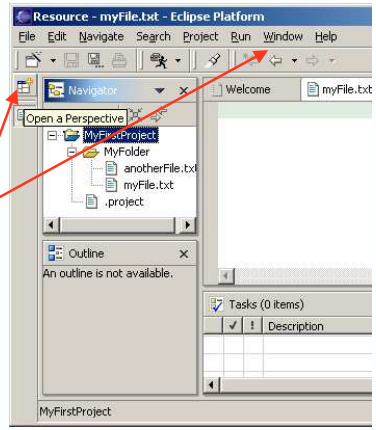


Perspectives

- Perspective defines initial layout of the views in the Workbench
- They are task oriented, i.e. they contain specific views for doing certain tasks:
  - Java Perspective for manipulating Java code
  - Resource Perspective for manipulating resources
  - Debug Perspective for debugging applications
- One Workbench window contains many perspectives

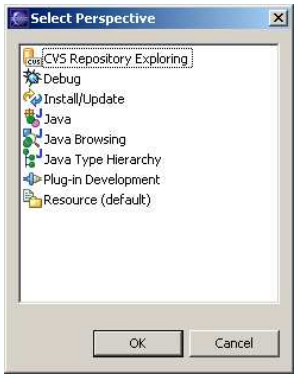
Opening Perspective

- It is possible to open a perspective:
  - In a same Workbench window
  - In a new Workbench window
- Perspective can be open by:
  - Choosing Window → Open Perspective… menu option
  - Clicking on a perspective shortcut button



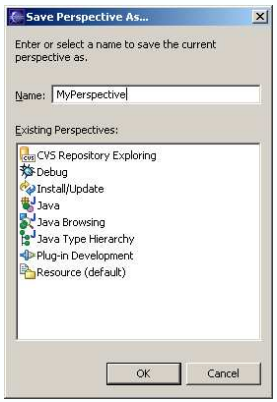
Available Perspectives

- By default, the following perspectives are available in the Workbench:



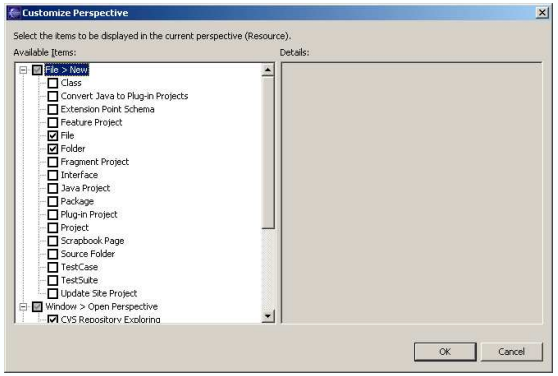
Saving Perspective

- Arrangement of views and editors can be modified and saved for perspectives
  - Choose Window → Save Perspective as … to save perspective
  - Perspective can be saved under a new (use-defined perspectives) or an existing name
- You can also reset perspective to original arrangement
  - Choose Window → Reset Perspective…



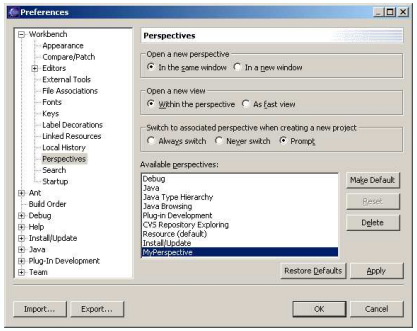
Customizing Perspectives

- Available for current perspective
  - Allows customization such as menu items and views



Deleting Perspectives

- Only user-defined perspectives can be deleted
- To delete perspective:
  - Choose Window → Preference
  - Extend Workbench category
  - Select Perspectives
  - Select perspective for deleting
  - Click Delete

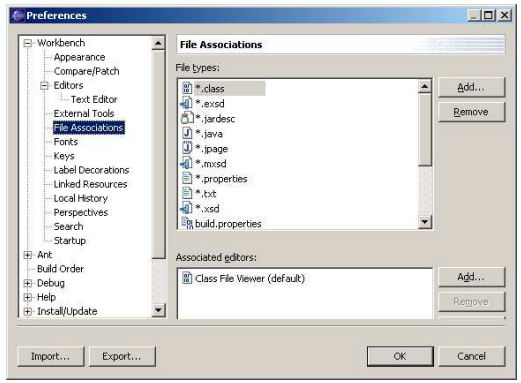


Editors

- An editor for a resource opens when you double-click on a resource
  - Editor type depends on a type of the resource, for example .txt files are open with the text editor
  - When an editor opens on a resource, it stays open across different perspectives
    - When perspectives change editor stays open
  - Active editor contains menus and toolbars specific to that editor
  - When you change a resource, an asterisk on the editor's title bar indicates unsaved changes

Editors and File Types

- It is possible to associate an editor with a file type
  - Choose Window → Preferences
  - Select File Associations
  - Select file type
  - Click Add to associate it with an editor



Views…

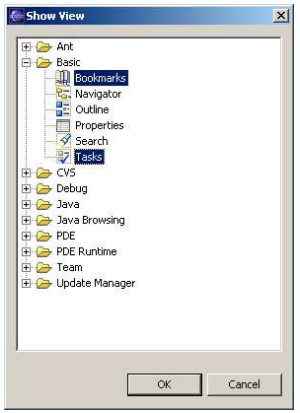
- The main purpose of a view is:
  - To support editors
  - To provide alternative presentation and navigation in the Workbench
- Views can have their own menus and toolbars
  - Items available in menus and toolbars are available only in that view

...Views

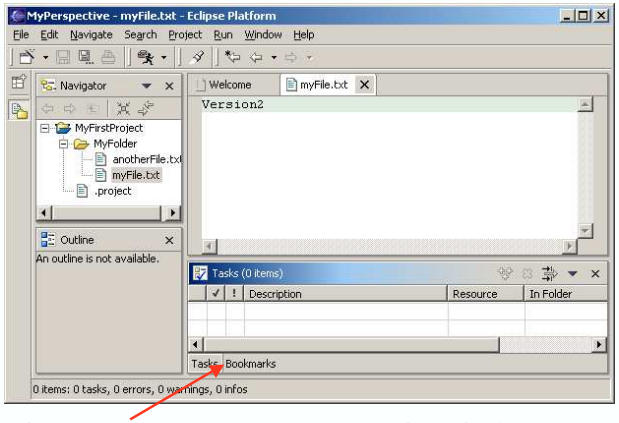
- Views can:
  - Appear on their own
  - Appear stacked with other views
- Layout of the views can be changed by clicking on the title bar and moving views
  - Single views can be moved together with other views
  - Stacked views can be moved to be single views

Adding Views to Perspective

- To add a view to the current perspective:
  - Choose Window → Open Perspective → Other…
  - The Show View window comes up
  - Select window to be shown
  - Click OK



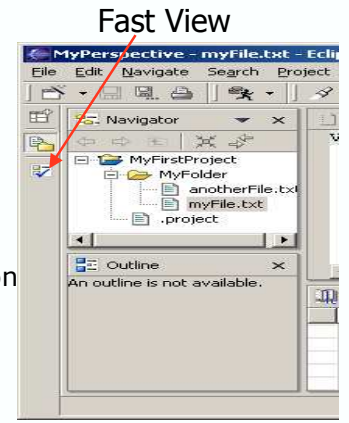
Stacked Views



- Stacked views appear in a notebook form
  - Each view is a notebook page

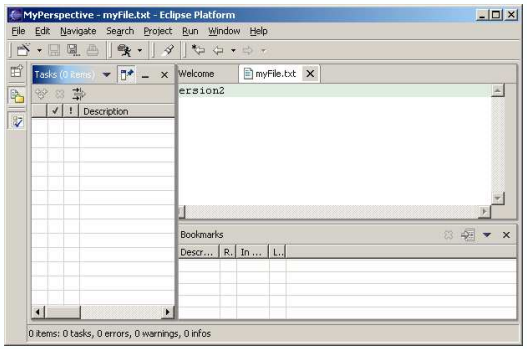
Fast Views...

- Hidden views that can be quickly opened and closed
  - They take up space in the Workbench
- Fast views can be created by:
  - Dragging an open view to the shortcut bar
  - Selecting Fast View from the view's menu
- Fast View is activated by clicking on its Fast View pop-up menu option



...Fast Views

- Click on the Fast View opens the view in the current perspective
- Click outside of the view makes it hidden again

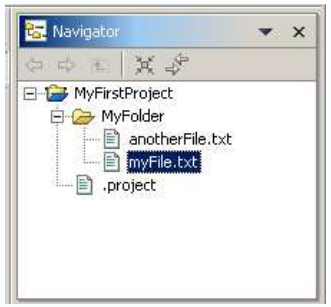


Specialized Views

- Some of the most commonly used views in Eclipse are:
  - Navigator View
  - Bookmark View
  - Task View
  - Search View
- These views display specific information

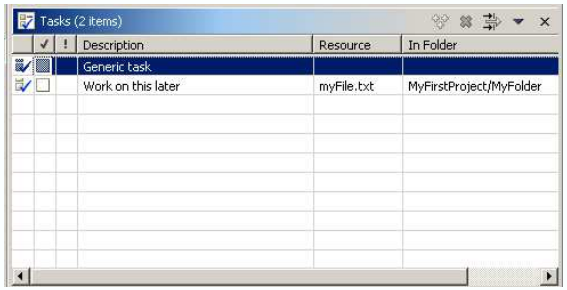
Navigator View

- Used for displaying resources in the Workbench
  - Provides hierarchical view of the resources
- Also used for manipulating resources
  - Editing
  - Import/Export
  - Refactoring



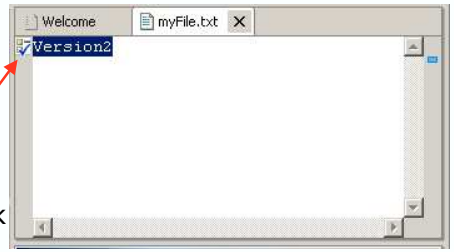
Tasks View

- Displays all tasks
  - Tasks associated with the resources
  - Generic tasks
  - Compiler errors that occur during the development



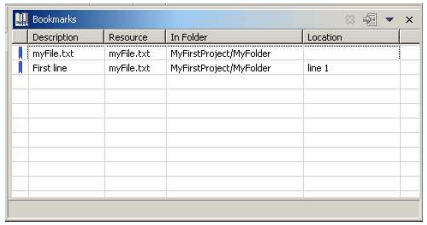
Adding Tasks

- Tasks usually indicate action for the resource
  - Associated with resources
    - With lines in a file
  - Generic
- To add a task:
  - Pop-up the menu on a line in the editor
  - Choose Add Task...
- To delete task:
  - Pop-up the menu on a task
  - Choose Delete Task...



Bookmarks View

- To add a bookmark on a line in a file:
  - Pop-up the menu on a line in the editor
  - Choose Add Bookmark...
- Bookmarks View displays all bookmarks available
  - Allows manipulation of the bookmarks

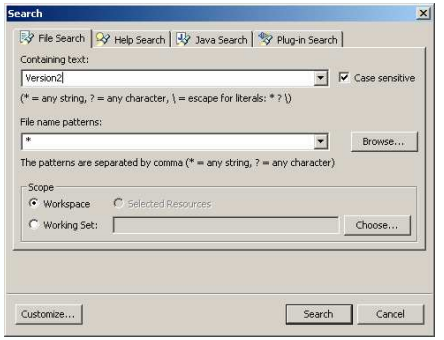


Adding Bookmarks

- Bookmarks are used as markers for:
  - Resources
  - Lines in a file
- To add a bookmark on a resource:
  - Pop-up menu on a resource in the Navigator view
  - Choose Add Bookmark...
  - Specify bookmark name
  - Click OK

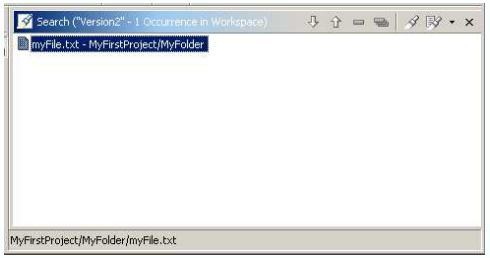
Searching

- Eclipse comes with an extensive Search mechanism
  - You can search for entities such as files, text, and Java classes
- To start search:
  - Select Search → Search... from the menu
  - Choose search criteria
  - Click Search



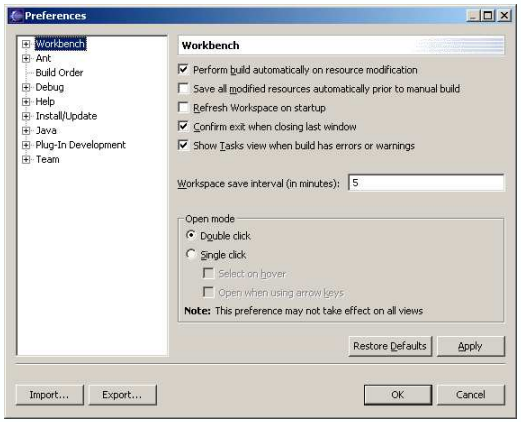
Search Results

- Displayed in a Search View
- Double-click on an item in a view opens an editor on that item
- Search Views allows for sorting and next search



Preferences Dialog

- Used for customization in the Eclipse
  - For setting user preferences
  - Opens by selecting Window → Preferences from the menu



Module Road Map

- Overview
- Installing and Running
- Workbench
- **Building and Running Java Classes**
  - **Developing Java applications**
  - **Browsing Java code**
  - **Searching Java code**
  - **Using Code Assist**
  - **Running Java applications**
  - **Scrapbook**

Java Development Tooling - JDT

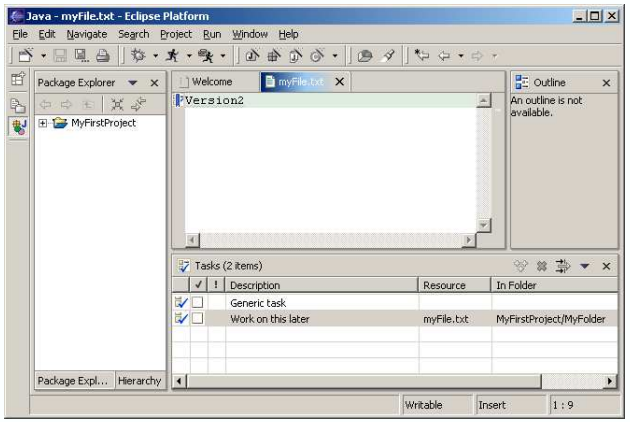
- Eclipse's Java Development Environment is often referred to as JDT – Java Development Tooling
  - Using the JDT you can do following with the Java programs:
    - Write
    - Compile
    - Test
    - Debug

Perspectives

- 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

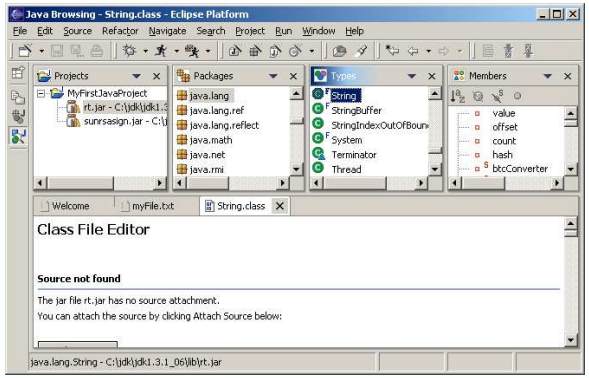
Java Perspective

- Contains:
  - Editor area
  - Package Explorer View
  - Hierarchy View
  - Outline View
  - Tasks View



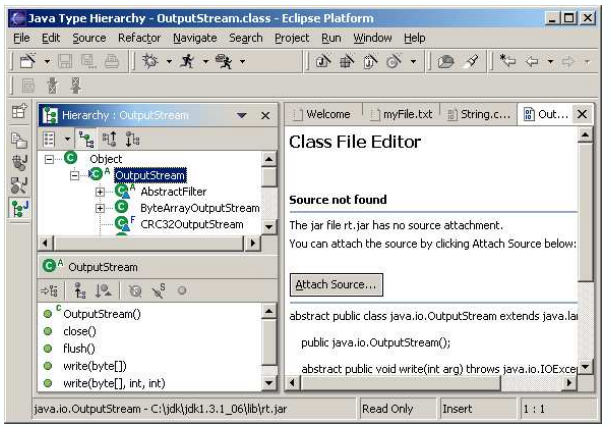
Java Browsing Perspective

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



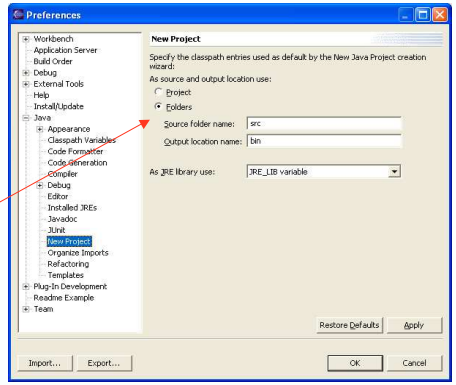
Java Type Hierarchy Perspective

- Contains editor area and Hierarchy View



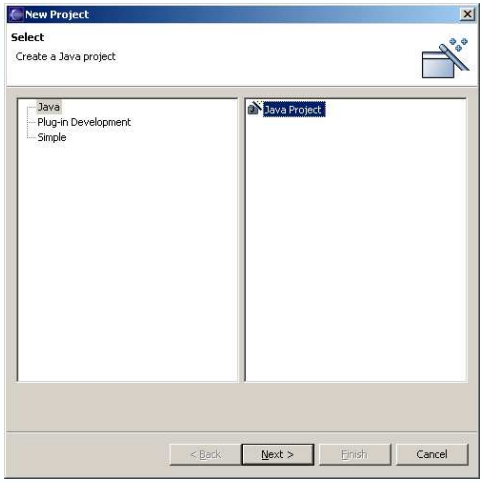
New Project Preferences

- 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



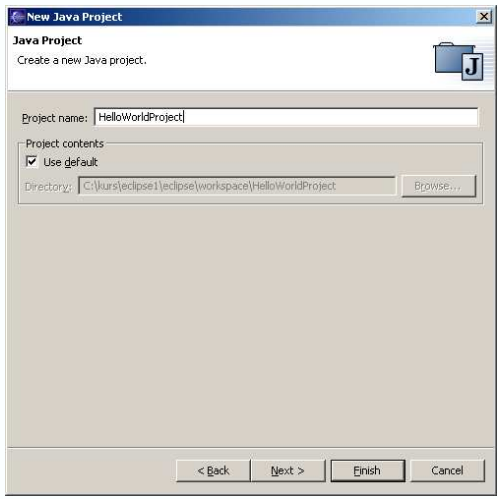
Creating Java Project

- Project used to organize resources (source, class, icons) for a project
- To create Java project:
  - Select Window → New → Project... from the menu
  - The New Project wizard comes up
  - Select Java → Java Project
  - Click Next



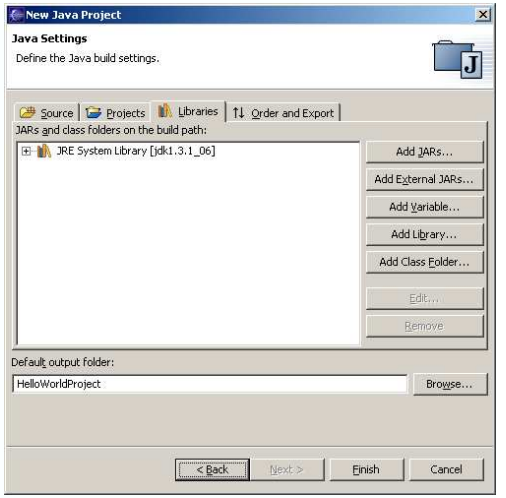
New Project Wizard

- Specify Project Name
- Click Next



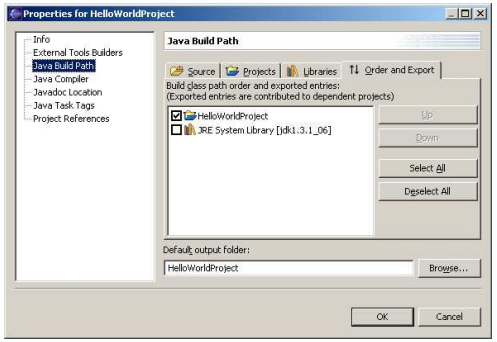
Java Settings

- Specify Java settings
  - Output folder (where compiled files will be stored)
  - External jar files project depends on
  - Classes from other projects that are referenced in the project
- Click Finish



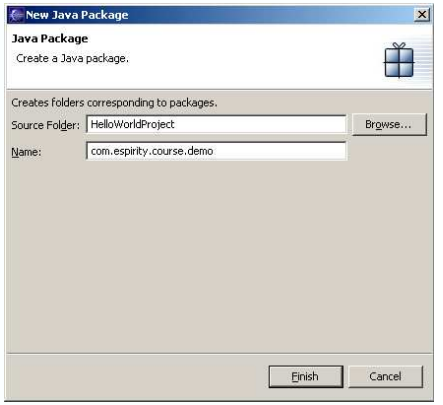
Project Properties

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



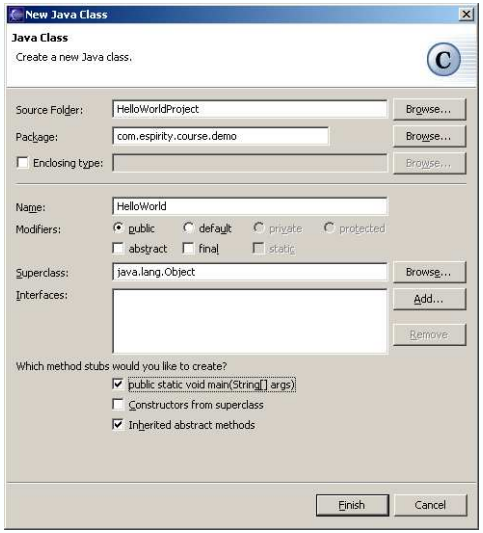
Creating Packages

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



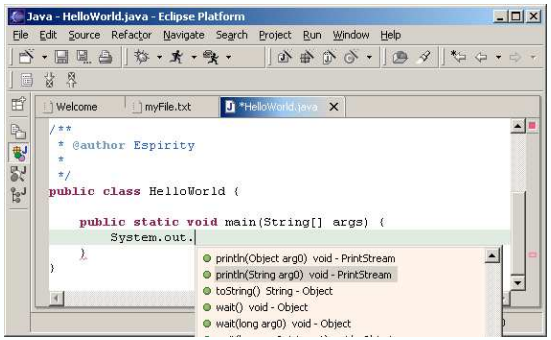
Creating Classes

- To create a class in a package:
  - Select the package
  - Choose New → Class from the context menu
  - The Class wizard comes up
  - Specify class details
  - Click Finish



Using Code Assist

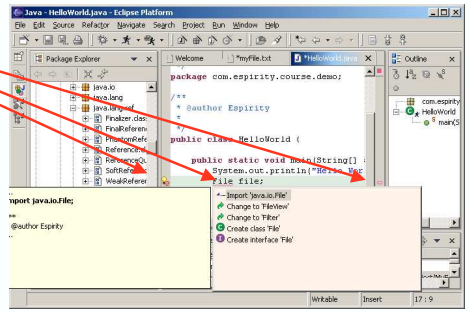
- When activated, code assist opens a list of available code completions
- Code Assist activates by Crtl+Space
  - Activates automatically when a message needs to be sent to an object (after the dot is typed)



Using Quick Fix

- Useful if Java compiler shows errors
  - Gives options for fixing the errors
  - Activated through Edit → Quick Fix menu option

Error indication



Searching for Java Classes

- When developing Java application a good search mechanism is very important
  - You often search for class, method declarations, and references
  - It is important to be able to find things quickly
- Eclipse Java Search mechanism is very extensive
- It allows you to search for:
  - Types, methods, constructors, packages, and fields
  - Declarations, Implementers, References
  - In the scope of Workspace, Working Set, or Selected Resources

Organizing Java Code

- Eclipse comes with extensive support for organizing and refactoring Java code
- It is possible to:
  - Generate getters and setters for the fields
  - Organize missing import statements
  - Move fields, methods, classes
  - Rename methods, classes, packages

Generating Getters and Setters

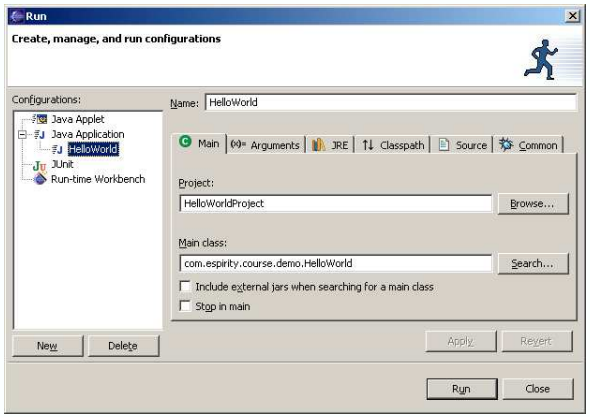
- Available for creating get and set methods on the fields
  - It supports encapsulation
  - Activated by choosing Source → Generate Getter and Setter from the editor's context menu

Refactoring

- Available from the Refactor context menu option in the editor
- Helps with refactoring Java code
- Allows for:
  - Renaming
    - Methods, fields, packages, projects, parameters, or local variables
  - Changing of method signature
  - Pull up a field or method (into super class)
  - Push down a field or method (into sub class)
  - Encapsulate field (generate getter and setter)

Running Java Classes

- To Run Java application
  - Choose Run → Run... from the menu



Console View

- Represents standard Java console
- Opens by default when standard Java output is used
  - Can also be open from Window → Show View menu

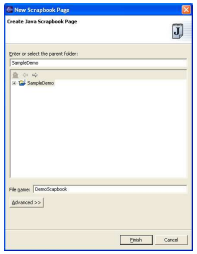
```
System.out.println("Hello World");
```

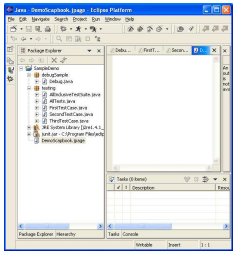
Scrapbook...

- Allows for writing and executing of Java code
  - Very useful for quick test of Java code that you write
- The Java code in the Scrapbook can be:
  - Displayed as a string when evaluated
  - Inspected when evaluated
    - Opens an Inspector view where you can see returning object from evaluation and all containing fields
  - Executed

…Scrapbook…

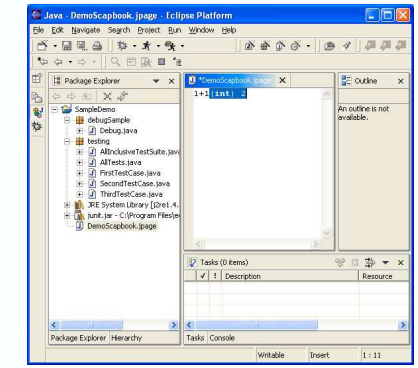
- It is created by selecting a project and choosing New → Scrapbook Page from the Package Explorer's context menu and then entering the name of the page
- Your scrapbook page will become a resource in your project





… Scrapbook…

- To open the scrapbook page just click on the resource
- It opens up like a Java source file editor
- Type Java code and select the context menu to Display or Inspect



…Scrapbook

- Class names must be fully qualified in what you type
- Set imports to make life easier
- Think of your scrapbook as a page that Eclipse will take the source you type, wrap it in a class with your source in the main menu, then compile and execute