Chapter 2

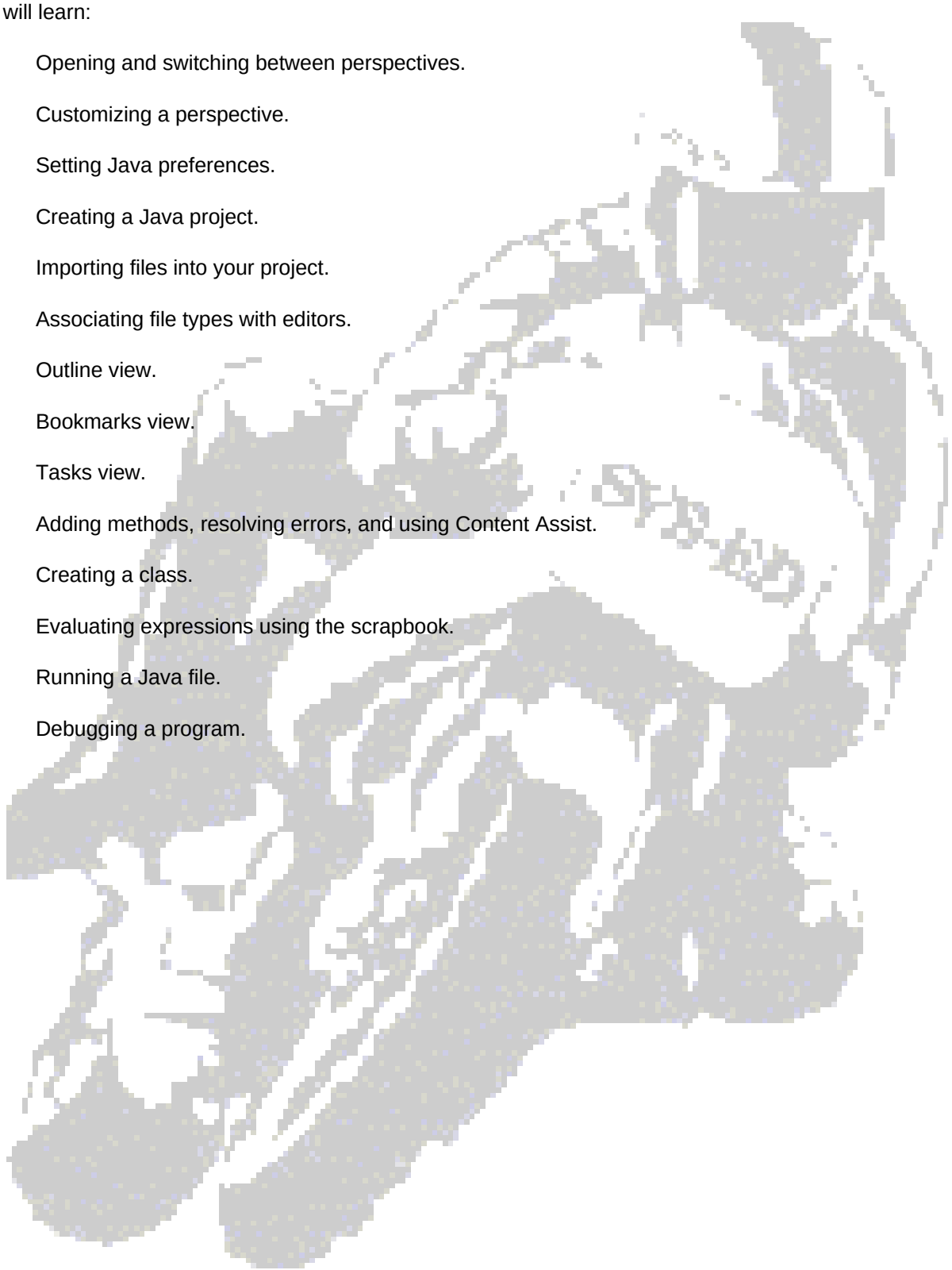
WORKBENCH

**SYS-ED/
Computer
Education
Techniques, Inc.**

Objectives

You will learn:

- Opening and switching between perspectives.
- Customizing a perspective.
- Setting Java preferences.
- Creating a Java project.
- Importing files into your project.
- Associating file types with editors.
- Outline view.
- Bookmarks view.
- Tasks view.
- Adding methods, resolving errors, and using Content Assist.
- Creating a class.
- Evaluating expressions using the scrapbook.
- Running a Java file.
- Debugging a program.



1 Workspace

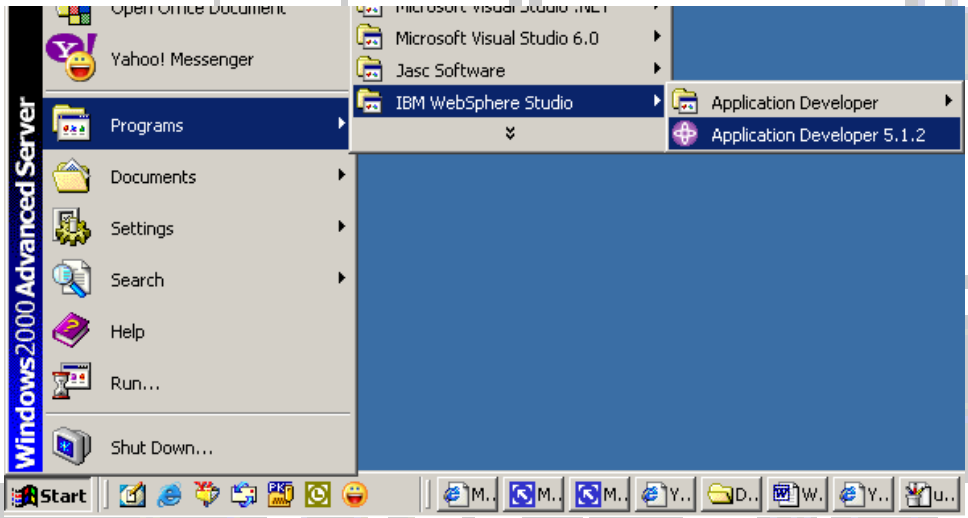
WebSphere Studio stores metadata about the work in a workspace. By default, new application resources are also stored in the workspace.

In order to work on different applications or different releases of the same application, different workspaces can be maintained. The required workspace can be specified when WebSphere Studio is started.

1.1 Starting Workbench in Windows

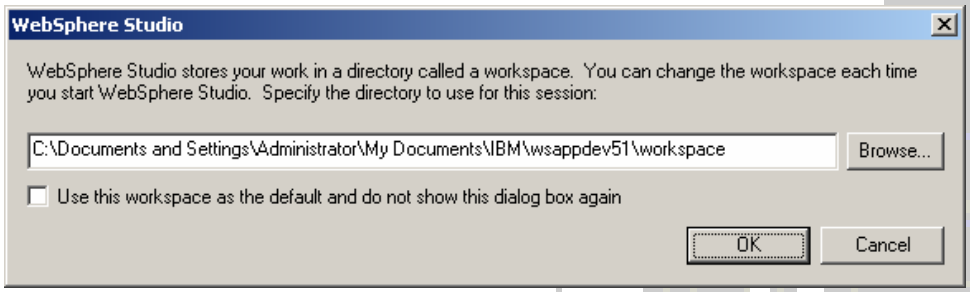
To start the workbench on Windows and create a new workspace:

1. Select Start…Programs…IBM WebSphere Studio.



2. Click Application Developer 5.x.x.

The first time WebSphere Studio is started, a dialog box opens with the default workspace directory already prefilled. By default, the work is stored in a directory called workspace, located in the My Documents\IBM\wsad directory.



3. In the directory field, enter the name of the required workspace. This will be the new default workspace directory.

4. Click OK. Because the new workspace directory does not exist, a new one with the supplied name will be created and opened.

1.2 Starting Workbench in Linux

To start the workbench on Linux and create a new workspace:

1. Open a command prompt.

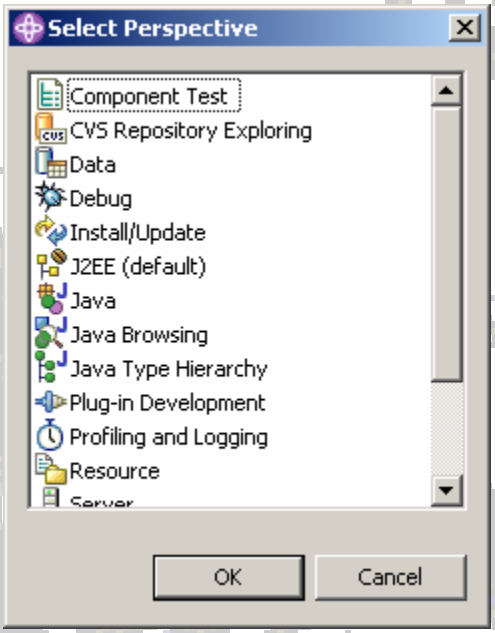
2. Type wsappdev and press Enter. The first time WebSphere Studio is started, a dialog box opens with the default workspace directory already prefilled. By default, the work is stored in a directory called workspace, located in the $HOME/IBM/wsad directory.

2 Opening and Switching between Perspectives

Perspective is a group of views that show various aspects of the resources in the workbench. The workbench user can switch perspectives, depending on the task at hand, and customize the layout of views and editors within the perspective.

To open the perspectives:

1. Select Window…Open Perspective…Other**.**



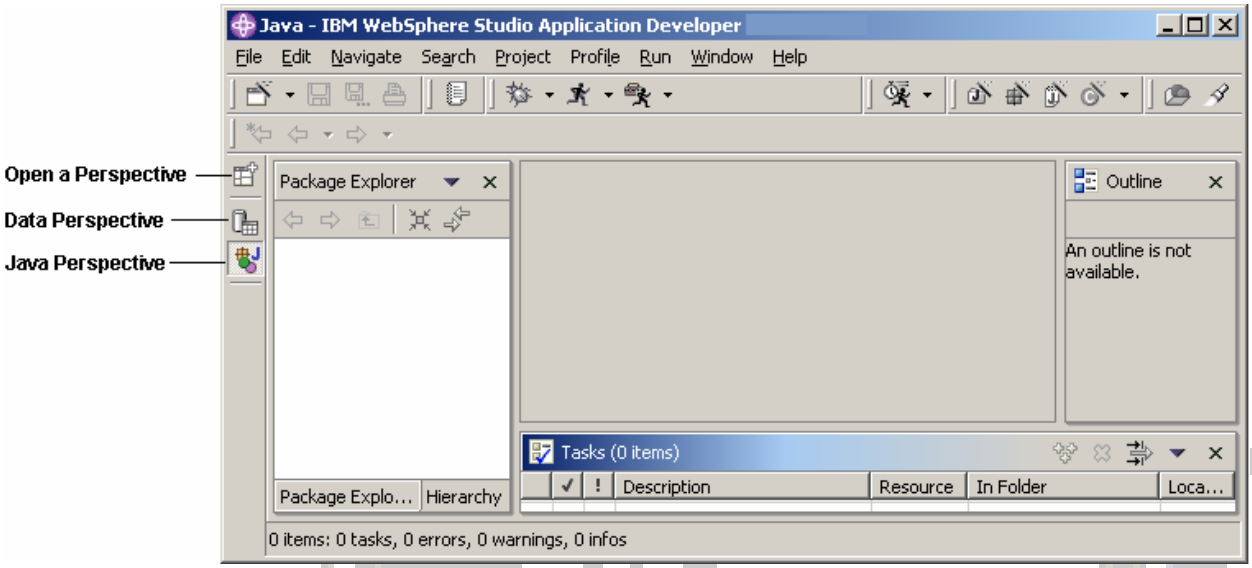
2. In the Select Perspective box, select Resource.

3. Select Window…Open Perspective…Java. This perspective contains several views: Packages, Hierarchy, Outline, Tasks, Search and Console.

4. Select Window…Open Perspective…Other. Switch to each of the other perspectives.

There is a vertical toolbar at the left side of the workbench; it is called the shortcut bar.

The shortcut bar contains icons that can be used to switch between the perspectives.



Click the appropriate button on the shortcut bar to switch to a perspective.

3 Perspective - Customization

A perspective can be customized by adding, closing, moving, and resizing views.

1. Click on the Resource Perspective button on the shortcut bar.

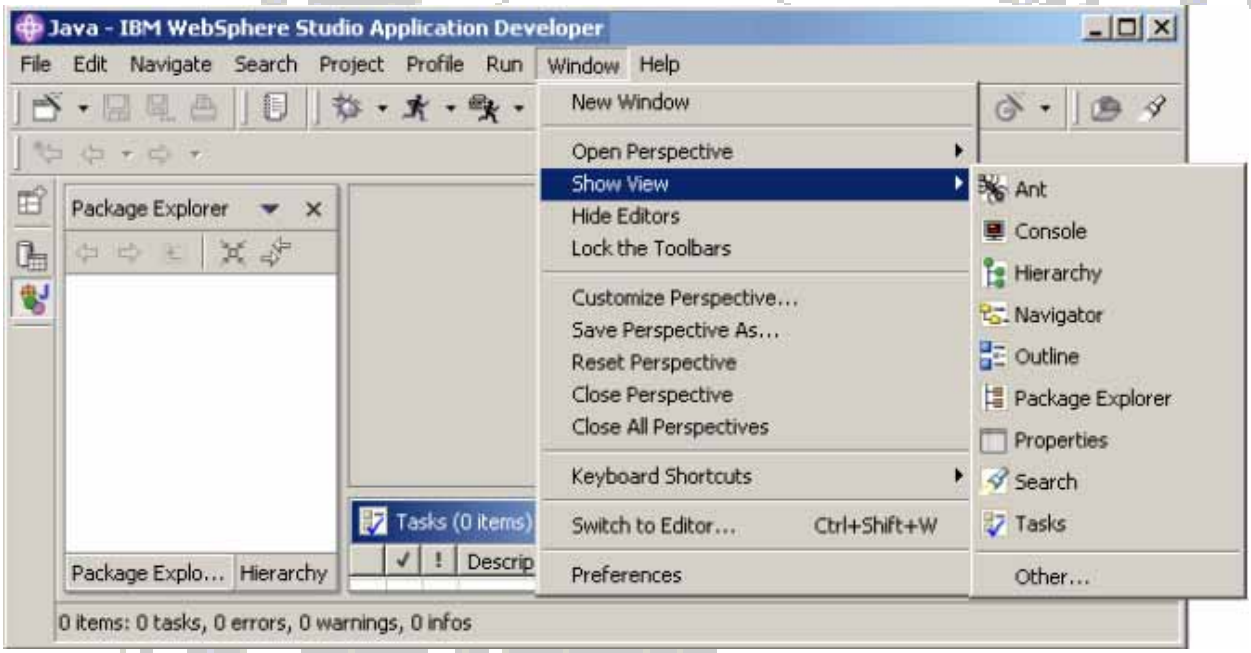
2. To resize the Navigator view, select the bottom border and drag the mouse pointer.

3. A view can be dropped on top of another view.

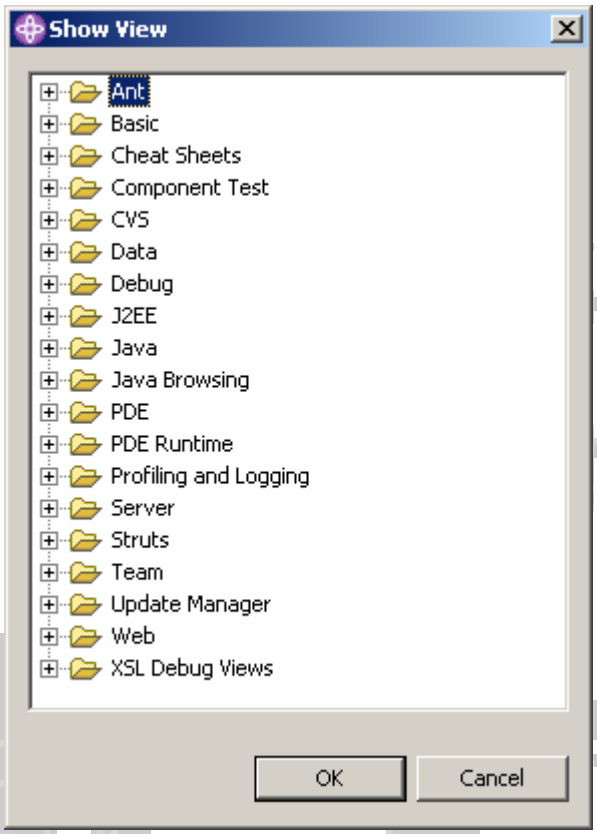
4. To remove the Navigator view from the Resource perspective, click the Close icon.

5. To open the Navigator view by click Window, Show View and Navigator.

6. New views can be added to the perspective. To add a view, click Window, Show View and Select the view.



7. To add more views, click Window, Show View and Other.



3.1 Default Perspective - Restoring

1. Click Window...Reset Perspective to reset the layout back to its original state.

   The current perspective can then reset.

2. Click OK.

   The views appear in their original places, and any new views that have been added.

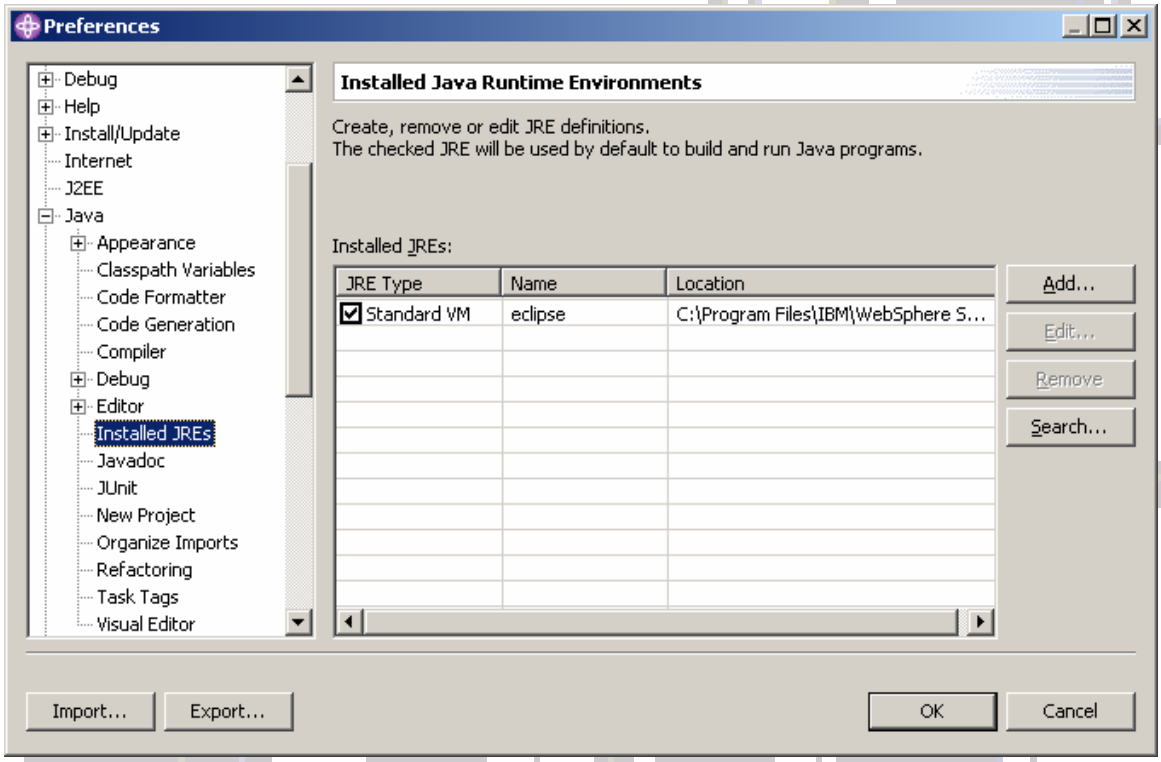
4 Java Preferences

To set user preferences, press Window and Preferences.

1. Press Window, Preferences Java and Installed JREs.

   Ensure the Standard VM check box is selected.

2. In the following screen capture, this is the standard Java virtual machine that is included with the workbench.



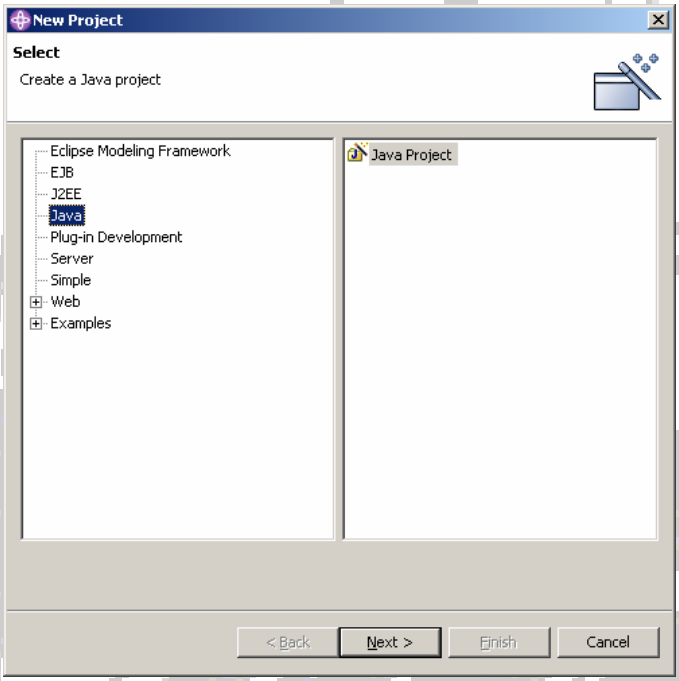
5 Java Project - Creation

All of the files and folders of an application are organized in projects.

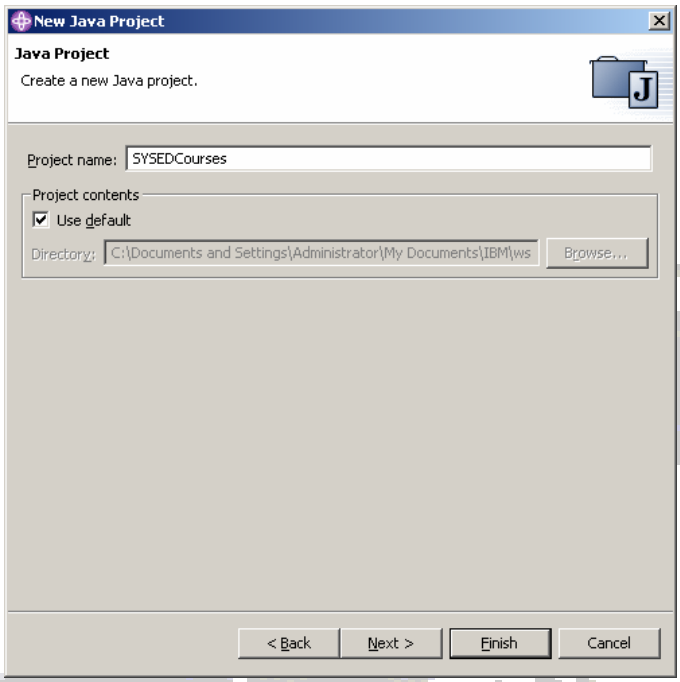
All of the elements of a project are known as resources.

A file cannot exist outside a project. A project must be created before work can be started on a single resource or file.

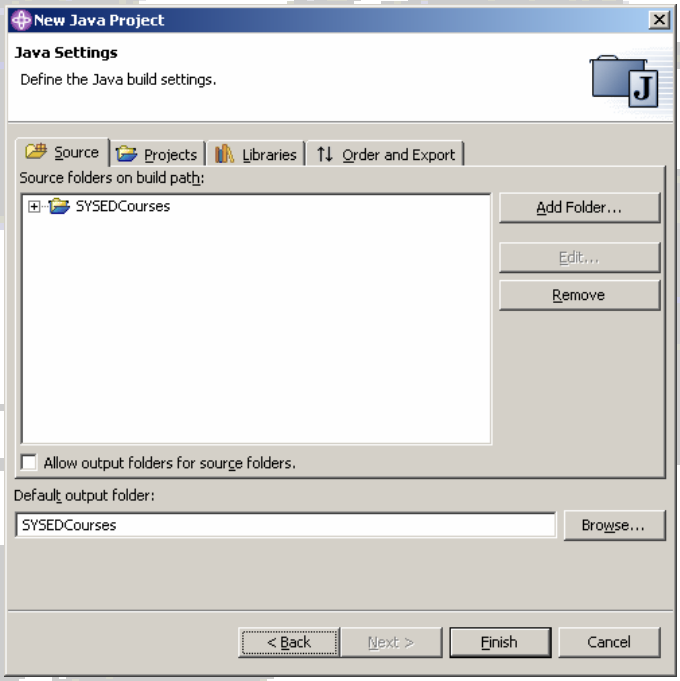
1. Press File, New and Project.



2. In the left pane, click Java. In the right pane, click Java Project.



3. Type in the name into Project Name. Press Finish.

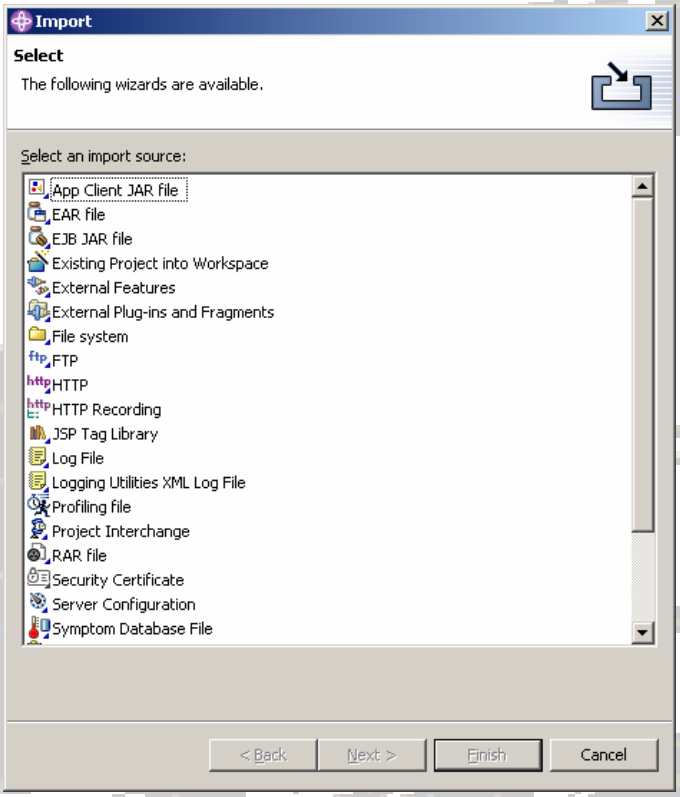


6 Importing Files into a Project

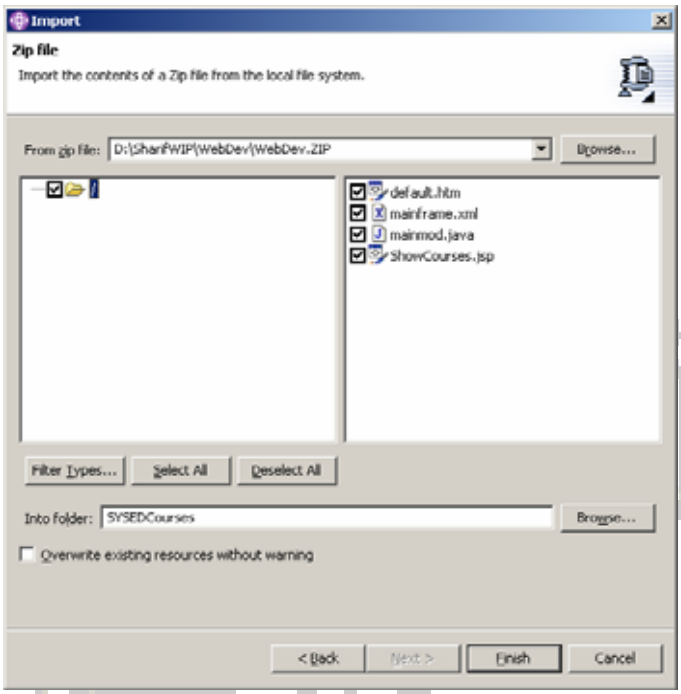
To import source code from a file system into the workbench:

1. Select the Package Explorer view. Select the Java project.

2. Press File and Import.



3. From the Import wizard, select Zip file, then click Next.



4. Press Browse.

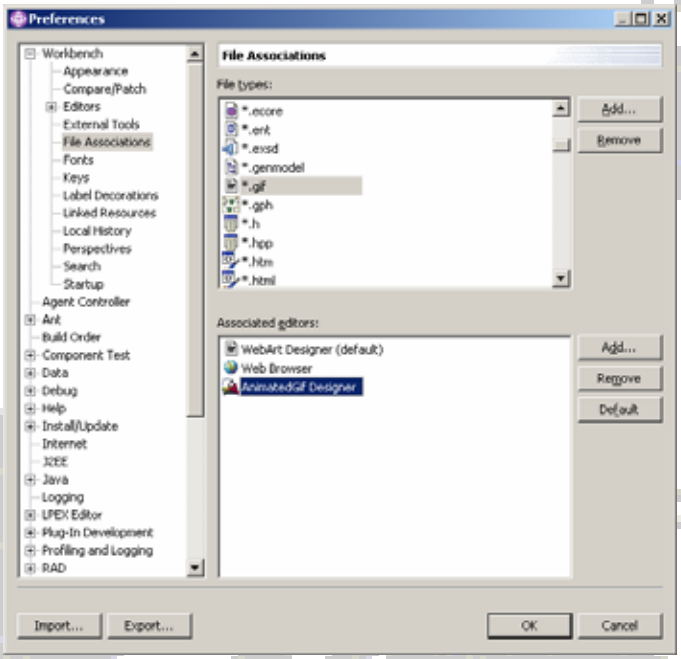
5. Press Select Types**.** Various file types can be selected and deselected for importing.

6. Make certain that the project appears in the Folder field. Then click Finish.

7 Associating File Types with Editors

1. Locate a .gif file and double click on it. The file should open in an associated editor. The default editor for the .gif filetype is WebArt Designer. Close the editor.

2. Press Window, Preferences, Workbench, File Editors. In this window, the file type can be associated with a different editor.



3. Select *.gif in the File Type pane.

4. Select AnimatedGif Designer in the Associated Editors pane.

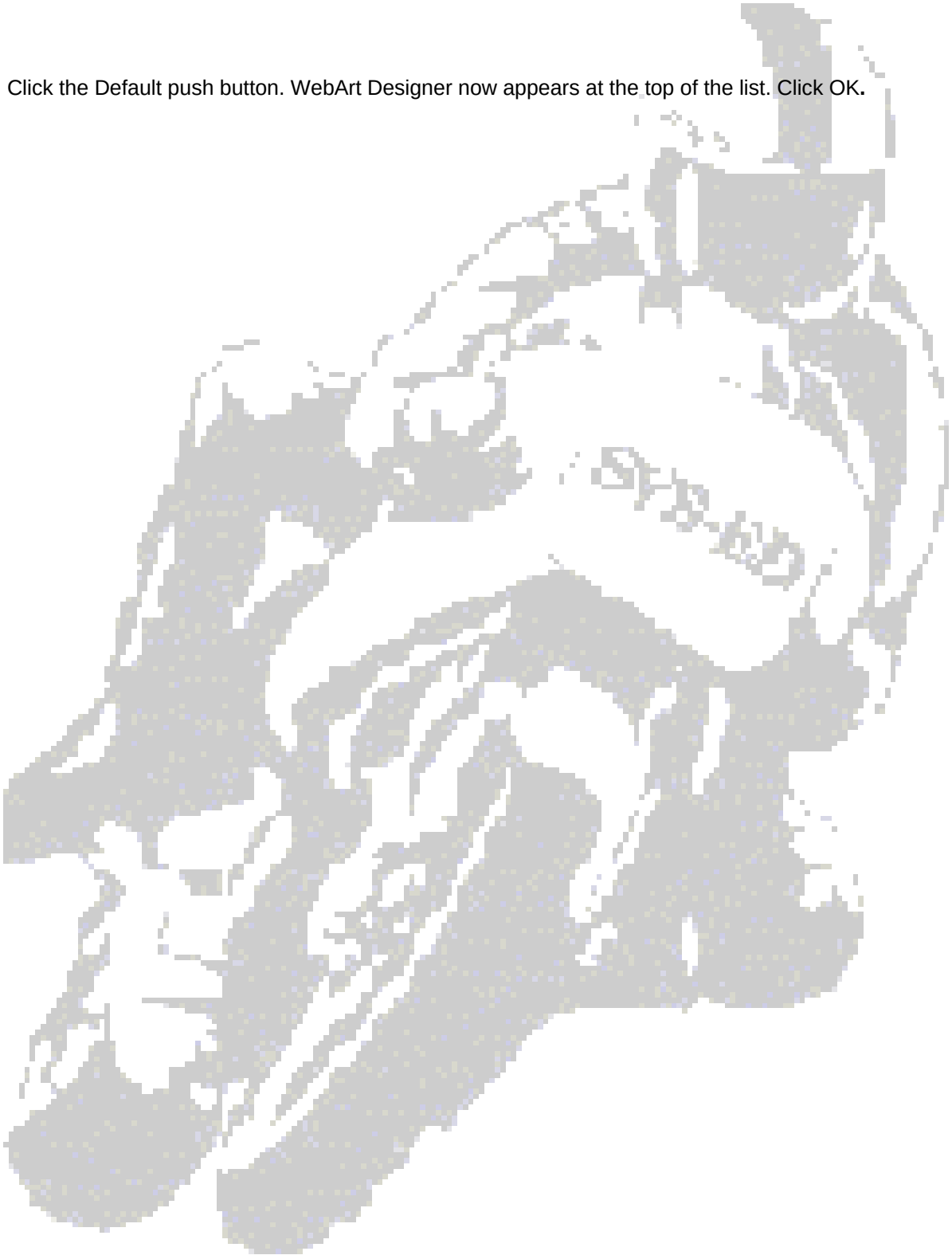
5. Click the Default push button. AnimatedGif Designer now appears at the top of the list. Click OK**.**

6. Click Window...Preferences...Workbench...File Editors to make WebArt Designer the default editor again.

7. Select *.gif in the File Type pane.

8. Select WebArt Designer in the Associated Editors pane.

9. Click the Default push button. WebArt Designer now appears at the top of the list. Click OK**.**



8 Outline View

The Outline view shows a structured outline of the file that currently is being edited, such as the methods and declarations in a Java file, or tags in an HTML file.

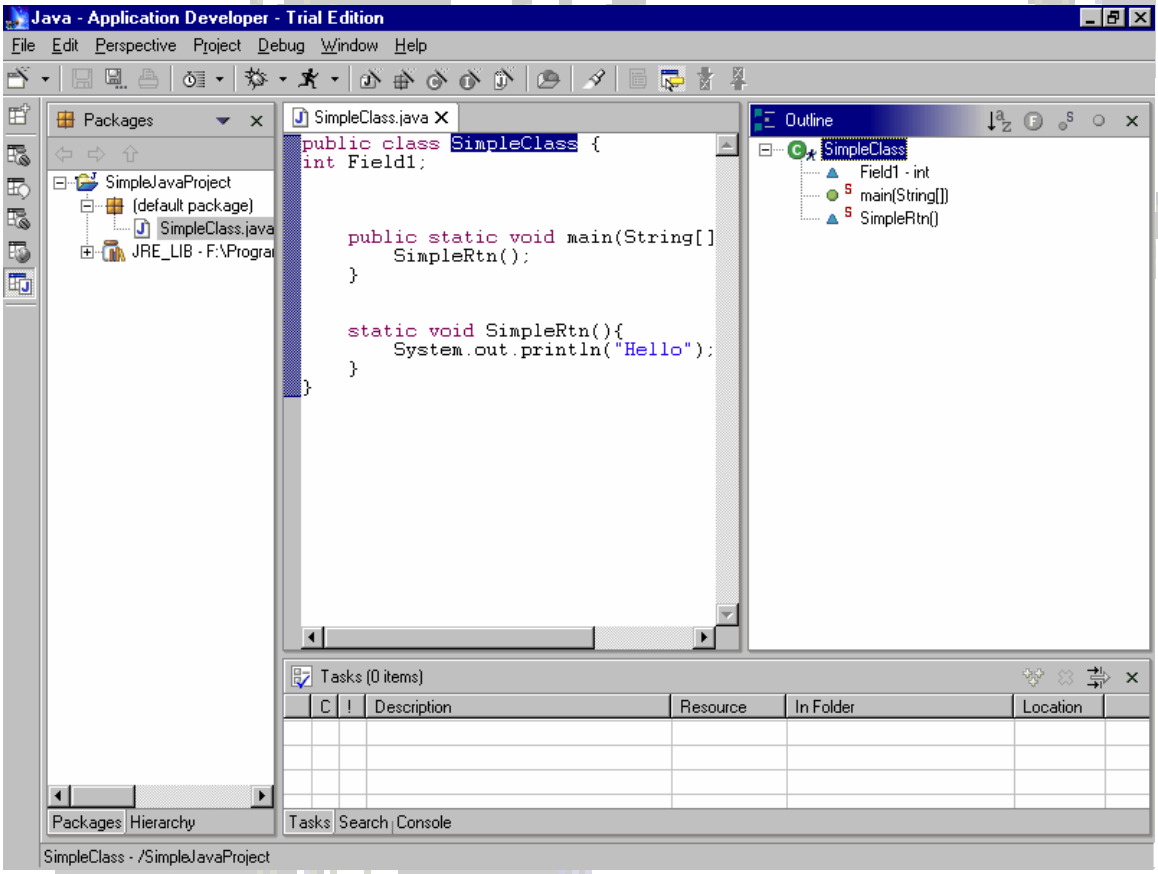
To view an outline of a file:

1. Go to the shortcut toolbar and press the Resource perspective  .

2. Expand a project in the Navigator view.

3. Double-click a java program. The file opens in the Java editor.

4. Open the Outline view by pressing Perspective, Show View and Outline.



9 Bookmarks View

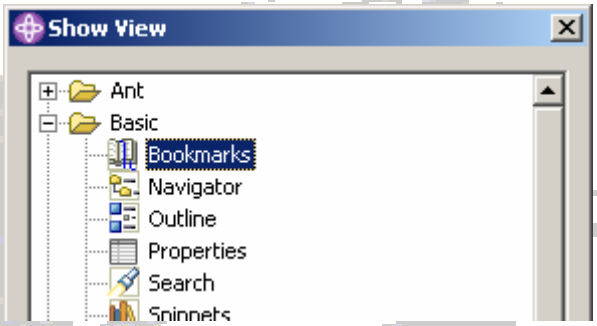
Sections of code that are frequently used or edited can be set for bookmarks.

To create a bookmark in a test file:

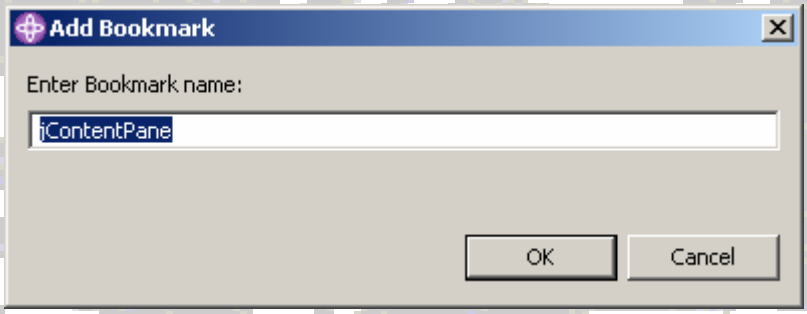
1. Expand a project.

2. Double-click .java. This will display a source editor.

3. Press Window, Show View, Others, Basic, Bookmarks. The Bookmarks view opens.



4. Select Java code and click Edit...Add bookmark.

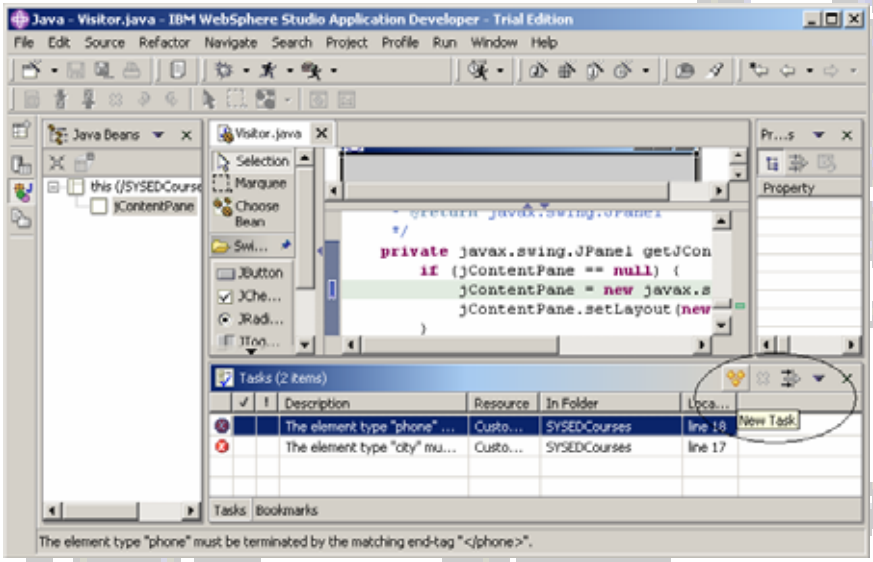


10 Tasks View

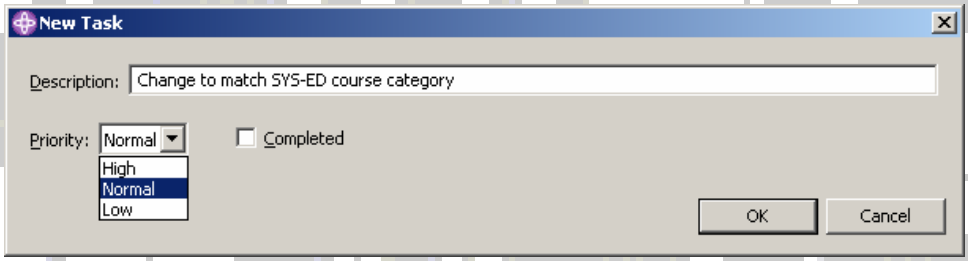
The Tasks view can be used for keeping track of tasks that have to be performed. This window is also used to view problems that the workbench logs automatically as applications are being built.

Perform the following steps in order to create a task:

1. In the Tasks view, click the New Task icon on the Tasks view toolbar.



2. In the Description field, type a description of the task.



3. In the Priority column ! , select Low from the drop-down list. The blue arrow indicates that the task is a low priority one.

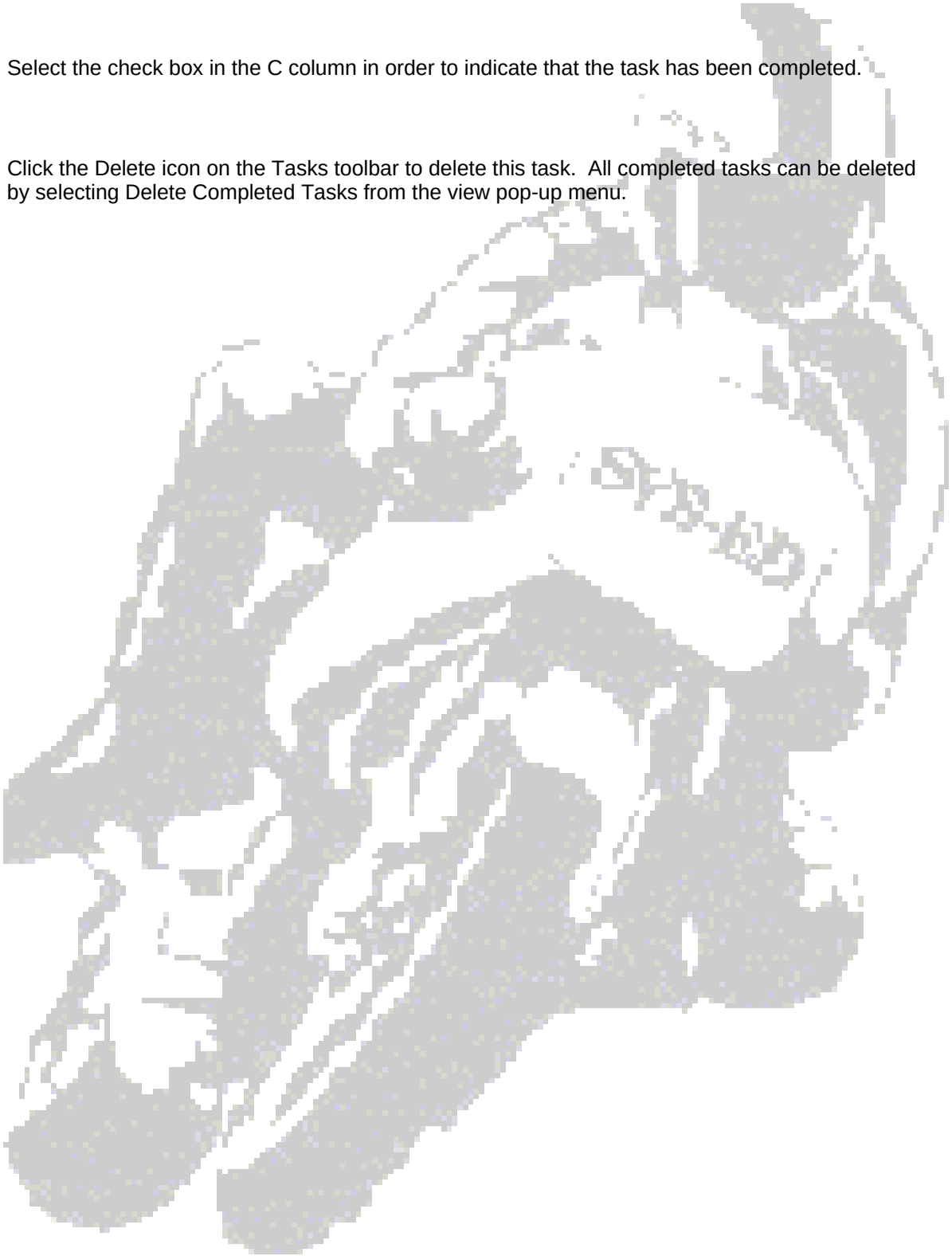
4. From the drop-down list, select Normal. The Priority column is now blank to indicate that this task has normal priority.

Now change the priority to High. An exclamation mark appears ! to indicate this is a high priority task.

The Resource field lists the file that the task is associated with. It is a read-only field.

5. Select the check box in the C column in order to indicate that the task has been completed.

6. Click the Delete icon on the Tasks toolbar to delete this task. All completed tasks can be deleted by selecting Delete Completed Tasks from the view pop-up menu.



11 Adding Methods, Resolving Errors, and Using Content Assist

1. Switch to the Java perspective by clicking Window...Open Perspective...Java.

2. Open a package. Open a .java file.

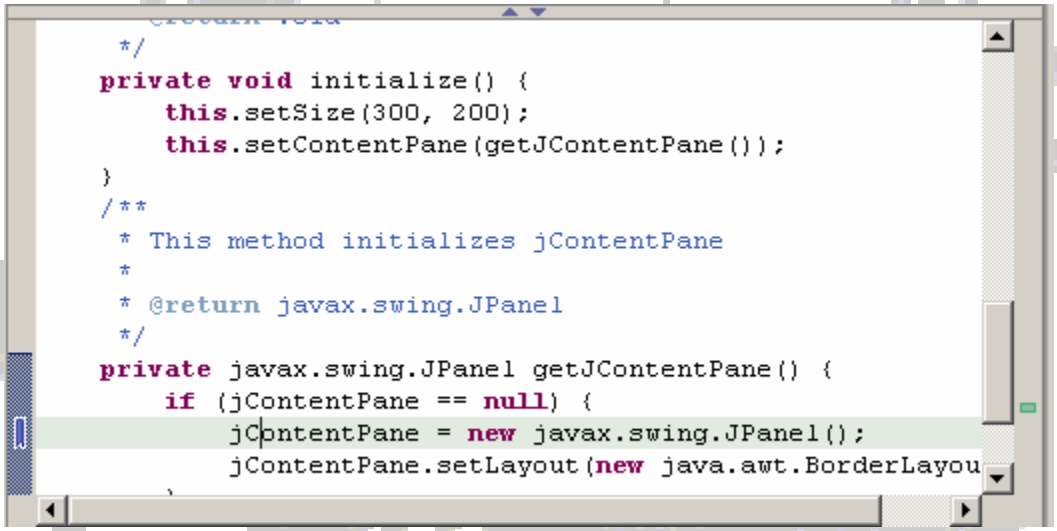
   The editor has the intelligence to differentiate Java tokens with color coding. Regular comments are green, Javadoc comments are cyan, keywords and built-in types are purple, and strings are blue.

3. In the editor, type in a method.

   Right after the method name has been typed in the editor, the new method appears in the Outline view.

4. Click Save to save the file.

   WSAD performs an incremental build that compiles the modified resource. Look in the Tasks view for any errors.



```
     */
    private void initialize() {
        this.setSize(300, 200);
        this.setContentPane(getJContentPane());
    }
    /**
     * This method initializes jContentPane
     *
     * @return javax.swing.JPanel
     */
    private javax.swing.JPanel getJContentPane() {
        if (jContentPane == null) {
            jContentPane = new javax.swing.JPanel();
            jContentPane.setLayout(new java.awt.BorderLayou
```

5. Fix the error and save the file again. The error should disappear.

   Content assist can now be invoked by selecting a statement and pressing Ctrl and then pressing the space bar.

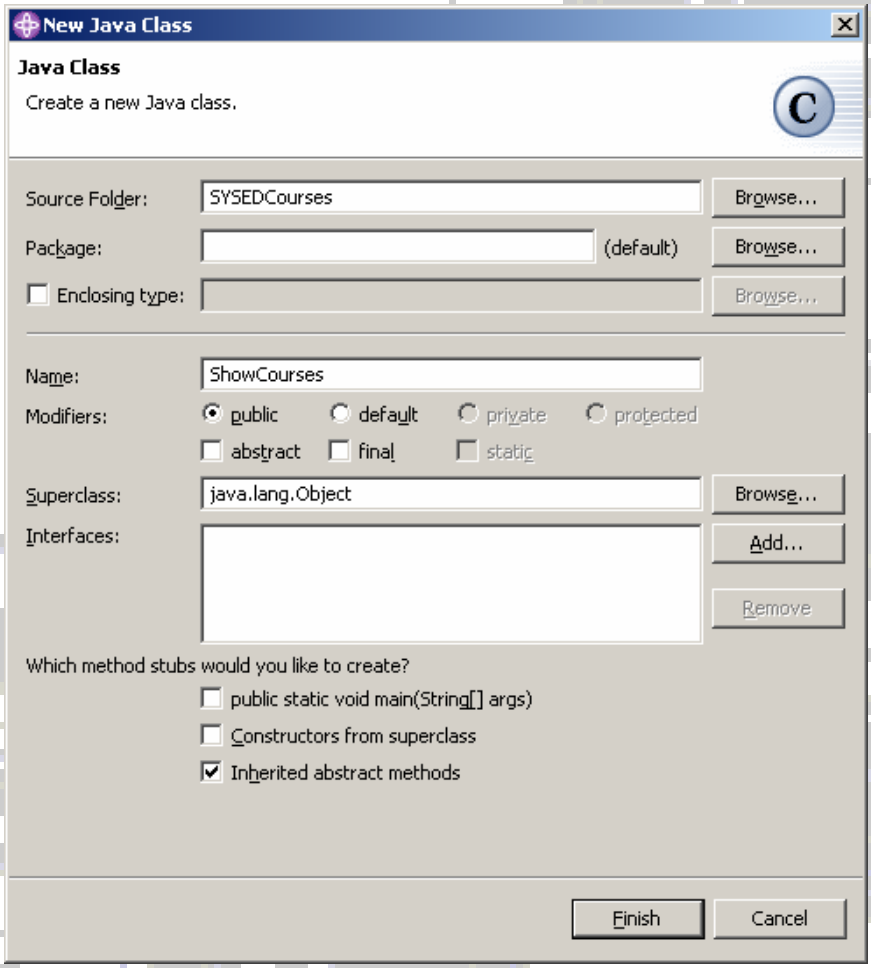
12 Class Creation

1. Select a Package Explorer view.

2. Click the Create a Java Class icon  .

   The New Java Class wizard opens.

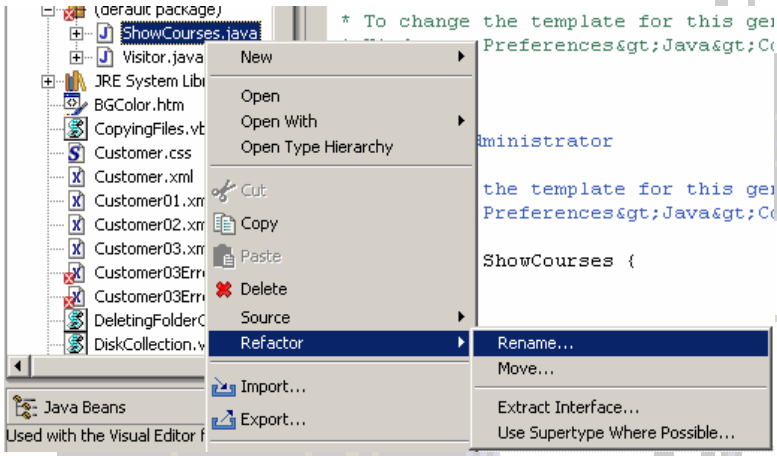
3. The Folder and Package fields are primed with the names of the folder and package that has been selected.



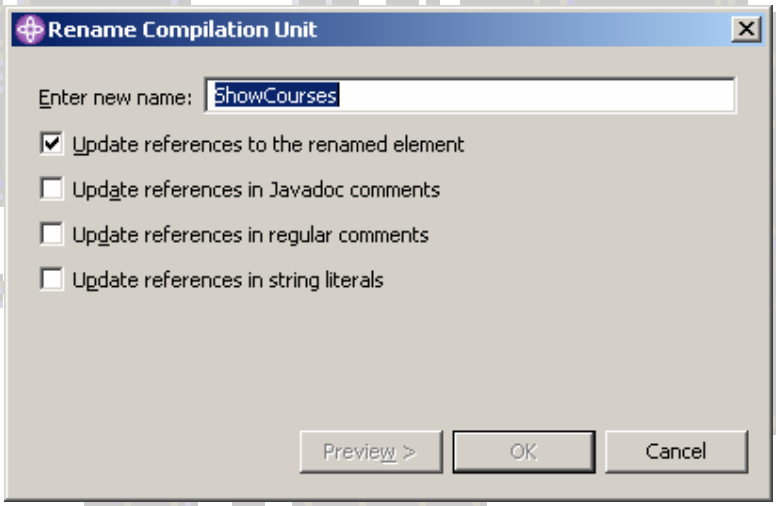
12.1 Class Renaming

1. Select the Package Explorer view. Select a .java from a package.

2. Right click on the program. Select Refactor…Rename on the pop-up menu.



3. The Rename Compilation dialog box opens.



4. WSAD reviews the changes and presents a preview of the changes that would take place if the resource has been renamed. This technique is called refactoring.

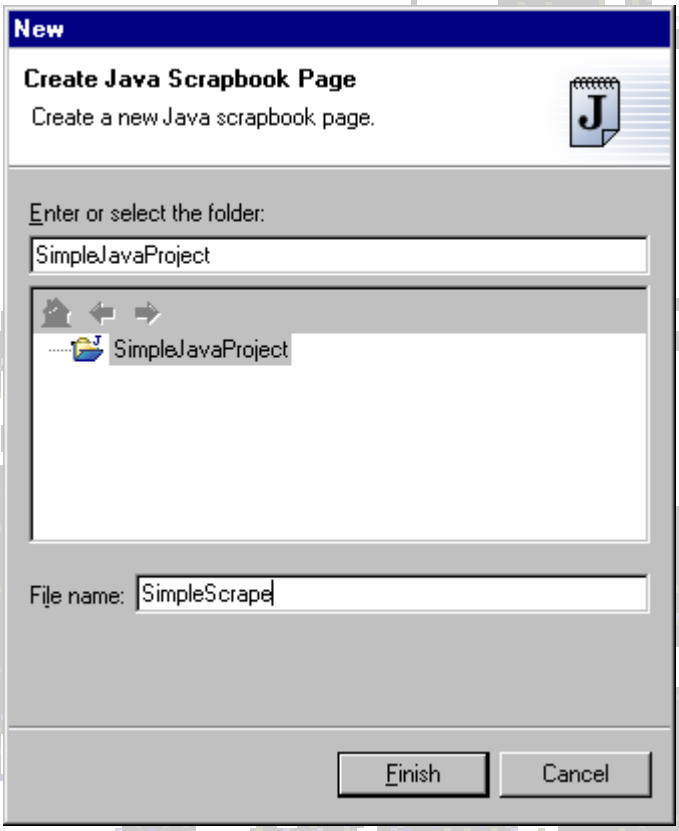
13 Scrapbook - Evaluating Expressions

WSAD has a facility to experiment with Java code fragments. It is called the Scrapbook.

Java expressions can be typed in to be run, inspected, and displayed. The Scrapbook can be invoked under the debugger during a break.

To evaluate a code snippet:

1. Select the Java perspective. Select the Create a Scrapbook Page icon in the workbench toolbar.



2. Select a project from the pane.

3. Enter the File name.

4. The jpage extension is added automatically.

5. Click Finish.

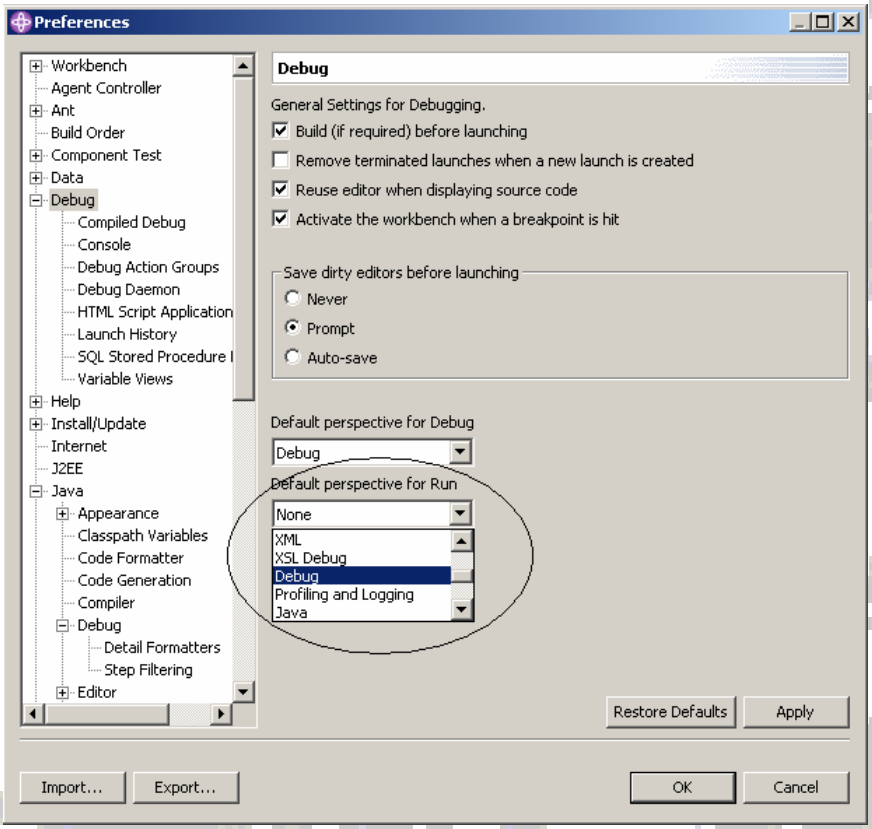
6. In the editor, type in a java statement that returns values. Select the line that just has been typed in and select Inspect from the pop-up menu.

The result of the inspection appears in the Inspector view; it will be opened if it is not already visible.

14 Running a Java File

To prepare the workbench for launching:

1. Select Window…Preferences…Debug.



2. In the Default perspective for Run list, select Debug, and then click OK.

To launch the Java file:

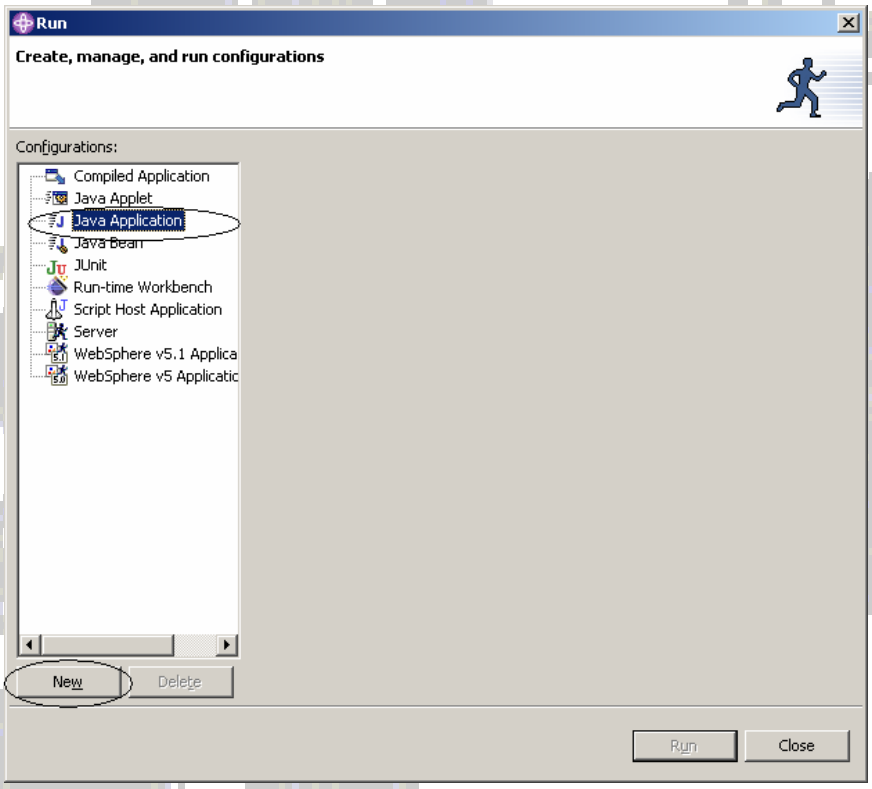
1. In the Package Explorer view, select the project, expand the package that contains the java file double-click it.

2. Look at the class in the Outline view. Verify that this class has an icon indicating that the class can be run.

3. In the Package Explorer view, select the package.

4. Click the Run icon on the main toolbar. The Launch Configurations page opens where you will create a launch configuration.

5. Select the Java Application and then click New. The right-side of the Launch Configuration page displays the options to be defined.

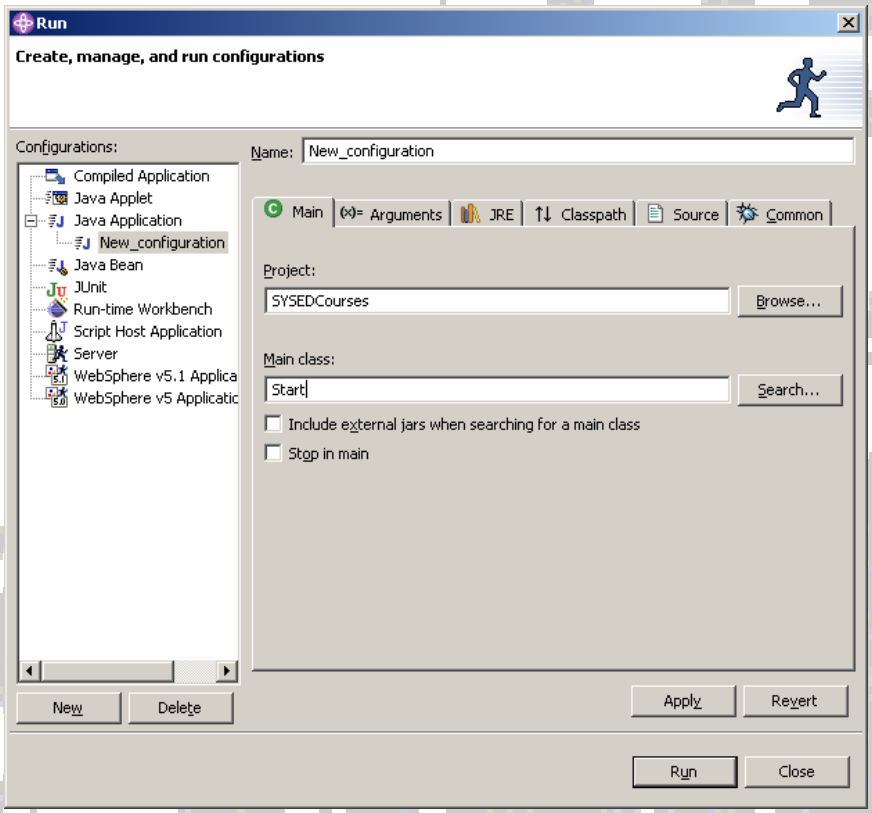


6. Enter a name In the Name field.

7. In the Main view, enter project name in the Project field.

8. Click Search beside the Main class field.

9. In the Choose Main Type dialog box, select the main type and click OK. The name of the class appears in the Main class field.



The file can now be launched in the project.

To launch a Java file:

1. In the Package Explorer view, open a .java.

2. Look at the class in the Outline view. Verify that this class has an icon indicating that the class can be run  .

3. Click the Run icon  on the main toolbar.

15 Debugging a Program

15.1 Setting Breakpoints and Launching a Program

1. Switch to the Java perspective. Open a .java file.

2. Move to a Java statement.

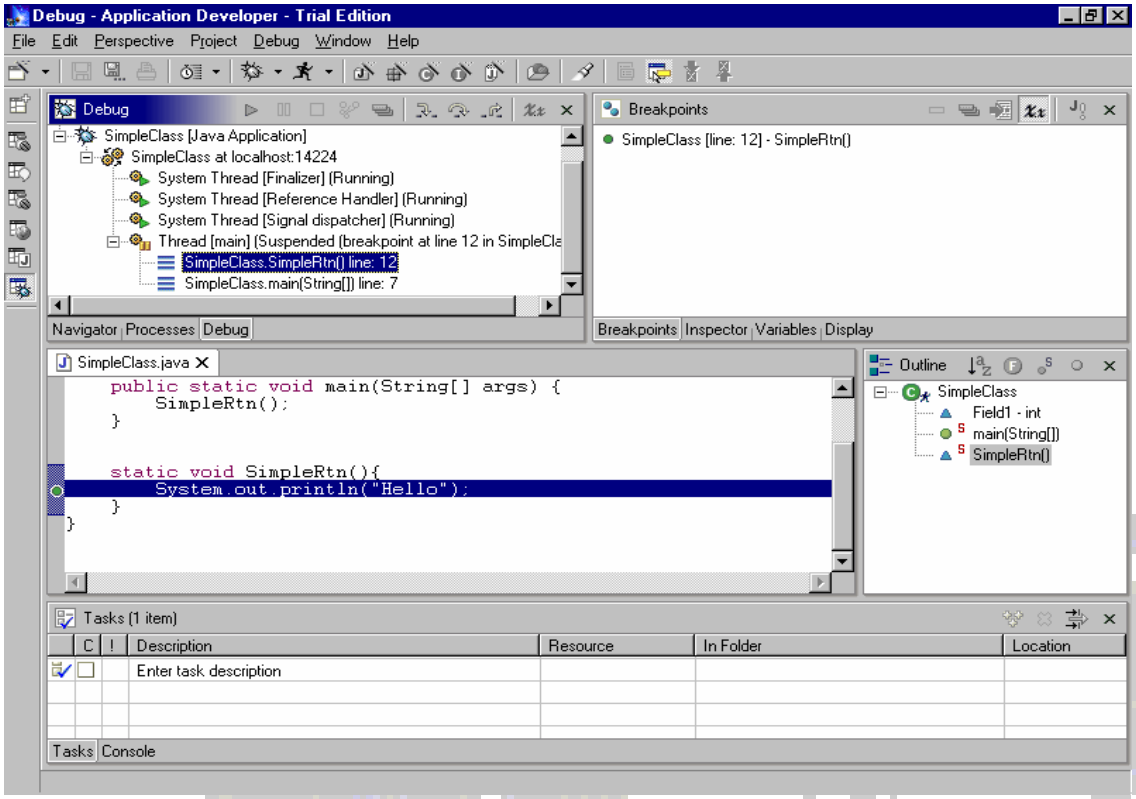
3. Click Add...Breakpoint on the pop-up menu. A breakpoint symbol appears in the left margin.

4. Click the Debug icon  on the main toolbar. The Java Application dialog box opens. Select the class and click Finish.

15.2 Debugging Steps

1. The workbench switches to the Debug perspective as soon as the breakpoint is hit. The breakpoint symbol is now green, indicating that it is verified.

   The stack frame appears in the Debug view. It shows the list of running threads.



2. The variables are listed in the Variables view. As an example of how to inspect a variable in a program, select a variable and click Inspect on the pop-up menu. The expression is displayed in the Inspector view.

3. The values of the variables can be changed in the Variables view. Expand and select variable. On the pop-up menu, click Change Variable Value.

4. To add an exception, click the Add Java Exception Breakpoint icon on the Breakpoints toolbar. The Add Exception dialog box opens. Select an exception from the list and click OK. The exception appears in the Breakpoint view.

5. On the Debug view toolbar, click the Step Over icon to step over the highlighted line of code and continue at the next line in the same method. Notice that the variable values change in the Inspector view.

6. Click the Run to Return icon  to step out of the current method.

7. Click the Resume icon  to continue execution of the program.

16 Local History

Every time an editable file is saved in the workbench, the workbench updates the local history of that file and logs the changes made to it.

The local history of a file can be accessed and reverted to a previously saved copy of the file, as long as the previous state is sufficiently recent in the save history.

1. Create a new file named sysedFile.txt.

2. Modify the file and save the file.

3. Repeat this by entering a new line and saving it again.

4. Add a third line and save it again.

5. From the resource's context menu in the Navigator view, select Replace With...Local History. The Replace from Local History dialog opens and shows the previous local history of the file.