Chapter

4

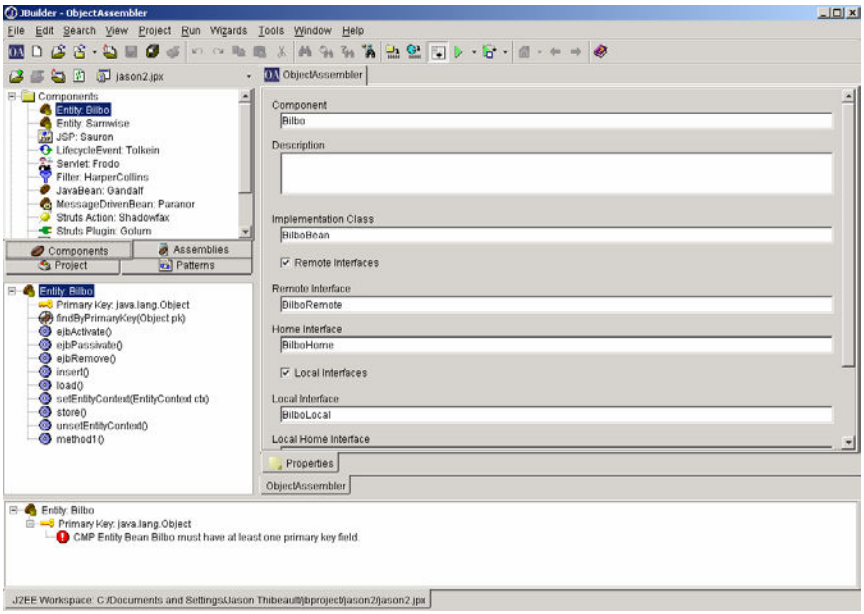
Working with ObjectAssembler™

Using ObjectAssembler™ is as easy as using your existing IDE.

IDE Integration

Without undermining your projects, or requiring you to learn an entirely new environment, ObjectAssembler™ integrates into the existing IDE menu-structure, toolbars and windows.

Workspaces and Views



An example Workspace—Component.

ObjectAssembler™ is broken into four central Workspaces

- Files—this Workspace is from the perspective of the physical files (i.e., java classes, HTML files, etc.) within your project

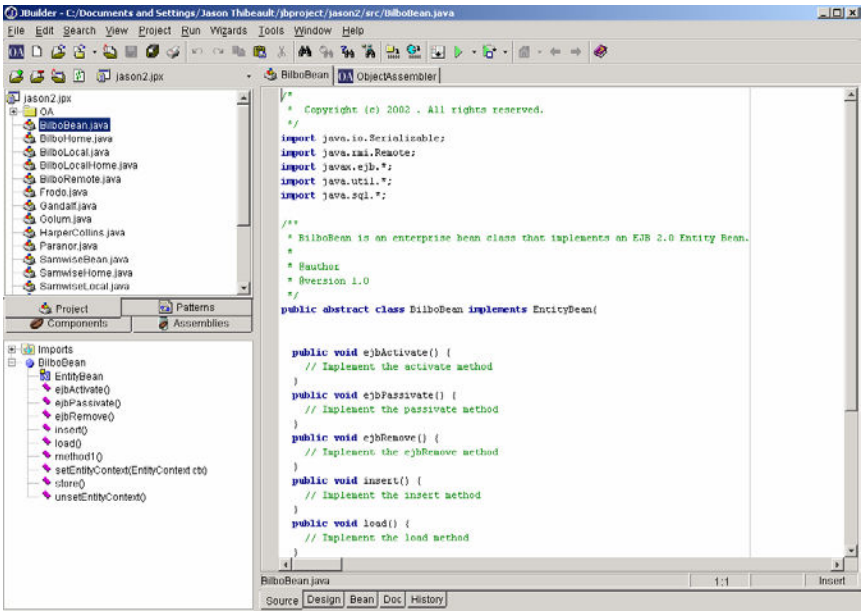
- Components—this Workspace is from the perspective of individual components (i.e., EJBs, Servlets, etc.)

- Assemblies—this Workspace is from the perspective of the application in which the components are placed (i.e., WAR, EAR, etc.)

- Patterns—this Workspace is from the perspective of the application patterns and strategies of which the components are a part[3].

As a developer using ObjectAssembler™, you can move freely between these Workspaces through the convenient 4-buttons/tabs in the IDE. Through the Intellisynch™ technology of ObjectAssembler™, changes made in one Workspace (i.e., to a component in the Assembly Workspace) are reflected in other Workspaces as well (i.e., the same component in the Component Workspace).

Files



The Files Workspace.

The following provides an overview of each View within the Files Workspace.

Project View

The Project View provides you an overview of

- the physical java files that comprise your project
- the descriptor files for the components (i.e., the java files) within your project

You can click on a project file or descriptor file to access the source code. For example

- If you click on a java file in the Project View

[3] It is not necessary, when building applications in ObjectAssembler™ 2.0, to utilize patterns. If you choose to build a pattern and apply it within your application, components will be mapped to roles within the patterns. These components and roles will then show up in the Component Workspace.

- A visual representation of that file (with an expandable tree-structure) will appear in the Detail View. This tree structure will show
  - Imported classes
  - The class of the java file (i.e., EntityBean)
  - The specific methods.
- The source code will appear in the Editor Window

Detail View

The Detail View provides you an overview of a java file in an expandable tree structure that includes[4]

- Imported classes
- The class of the java file (i.e., EntityBean)
- The specific methods.

If you make changes to a java file's source code, ObjectAssembler™'s InStep Validator™ will automatically check and validate your code. If there are errors, a new element will appear at the top of the Detail View. When expanded, this Errors folder provides detailed information on any errors discovered. Click on an error to highlight the specific code.

Editor View

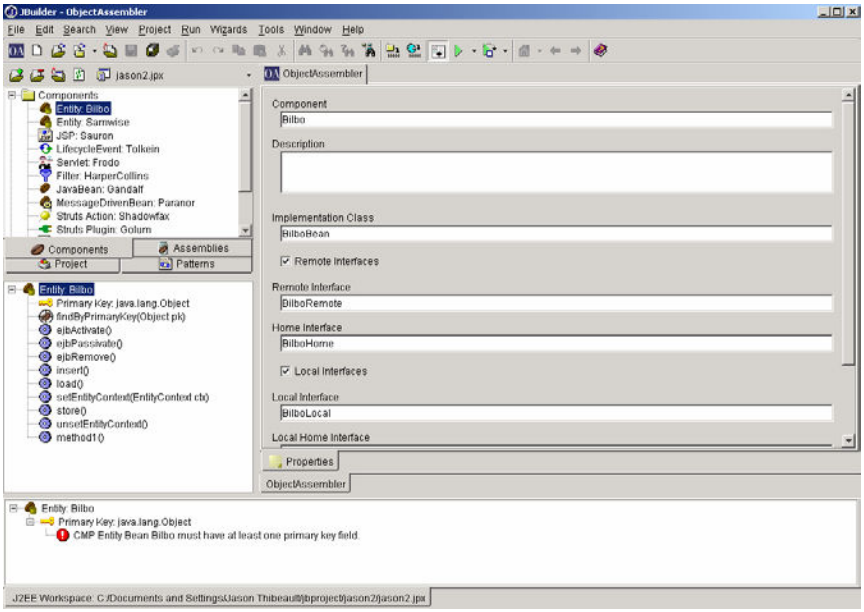
The Editor View allows you to make source code changes to your application (both to Java and Markup files).

Message View

The Message View is not available in the Files workspace, because ObjectAssembler™ uses the host IDE's source code editor and validator.

[4] When a java file is displayed in the Project View in the Files Workspace, the source code is automatically loaded into the Editor View.

Components



The Components Workspace.

The following provides an overview of each View within the Components Workspace.

Project View

The Project View provides you an overview of

- the components—those Beans, servlets, or JSP you have created or imported into your project
- mapped strategies—strategies from a pattern that are mapped to components
- connectors—any connectors you have created to define relationships between components.

You can click on a component or connector to set some of its source-code properties and XML descriptor properties

- For example, if you click on a component in the Project View
  - A visual representation of that component (with an expandable tree-structure) will appear in the Detail View. This tree structure will show all of the properties for that component as well as connectors and mapped strategies

If you make changes to a component's properties, the java source code is immediately modified through the IntelliSynch™ engine. As a result of those source-code changes, ObjectAssembler™'s InStep Validator™ will automatically check and validate your code. If there are errors, a message will appear in the Message View notifying you of such.

Detail View

The Detail View provides you an overview of a component, mapped strategy, or connector in an expandable tree structure that includes the various properties (represented as different icons[5]) of that object. Selecting a property will bring up it's properties window in the Editor View.

If you make changes to a component's properties, the java source code is immediately modified through the IntelliSynch™ engine. As a result of those source-code changes, ObjectAssembler™'s InStep Validator™ will automatically check and validate your code. If there are errors, a message will appear in the Message View notifying you of such.

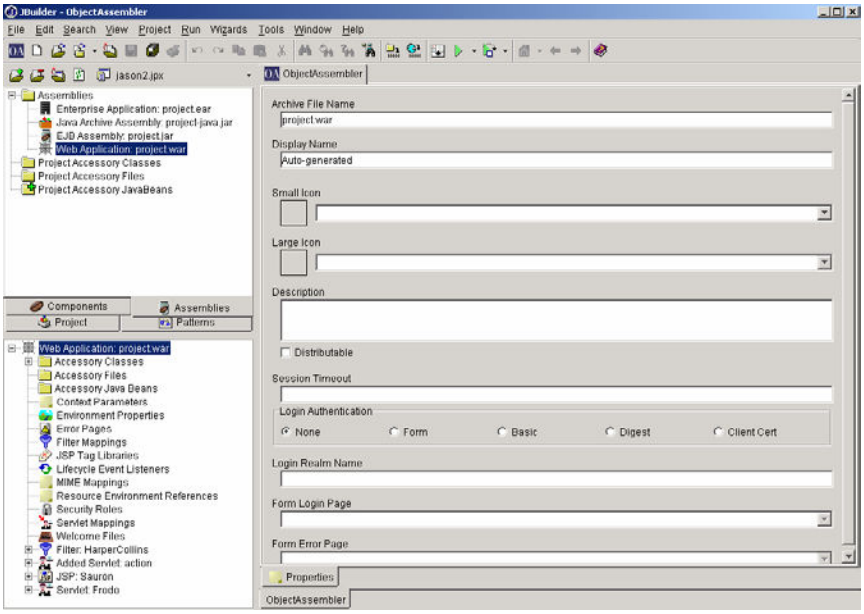
Editor View

The Editor View allows you to make changes to component, mapped strategy, or connector properties in source code and XML descriptor files through form-field, text-entry input.

Message View

The Message View displays any error or warning messages about the currently selected component. To see errors and warnings for multiple components, connectors, or mapped pattern strategies at once, select the associated folder that contains them.

Assemblies



The Assemblies Workspace.
The following provides an overview of each View within the Assemblies Workspace.

[5] See the Icon Table in Appendix B for a complete list of icons and where they are located in ObjectAssembler™ 2.0

Project View

The Project View provides you an icon for each type of application

- Enterprise Application (EAR)
- Java Archive Assembly (JAR)
- Enterprise JavaBean Assembly (EJB)
- Web Application Assembly (WAR)

The Project View also provides a list of all Project Accessory Classes, Project Accessory Files, and Project Accessory JavaBeans.

You can click on an application type to access all objects within the application. For example

- If you click on an application type
  - A visual representation of that application (with an expandable tree-structure) will appear in the Detail View. This tree structure will show all of the objects in that application. See Appendix C for a list of objects in each type of application.
- If you click the Project Accessory Classes folder
  - An expanded tree-view of the project classes in the Detail View. All classes included in the project that are not associated with a component created with ObjectAssembler™ will show up here.
- If you click the Project Accessory Files folder
  - A list of all Accessory Files in the Detail View
- If you click the Project Accessory JavaBeans folder
  - All JavaBean components created with ObjectAssembler™ in the current project

Detail View

The Detail View provides you a tree-view overview of all objects within an application with each object represented as a different icon[6]. Selecting an object will bring up it's properties window in the Editor View.

If you make changes to a component's properties, the java source code is immediately modified through the IntelliSynch™ engine. As a result of those source-code changes, ObjectAssembler™'s InStep Validator™ will automatically check and validate your code. If there are errors, a message will appear in the Message View notifying you of such.

[6] See the Icon Table in Appendix B for a complete list of icons and where they are located in ObjectAssembler™ 2.0

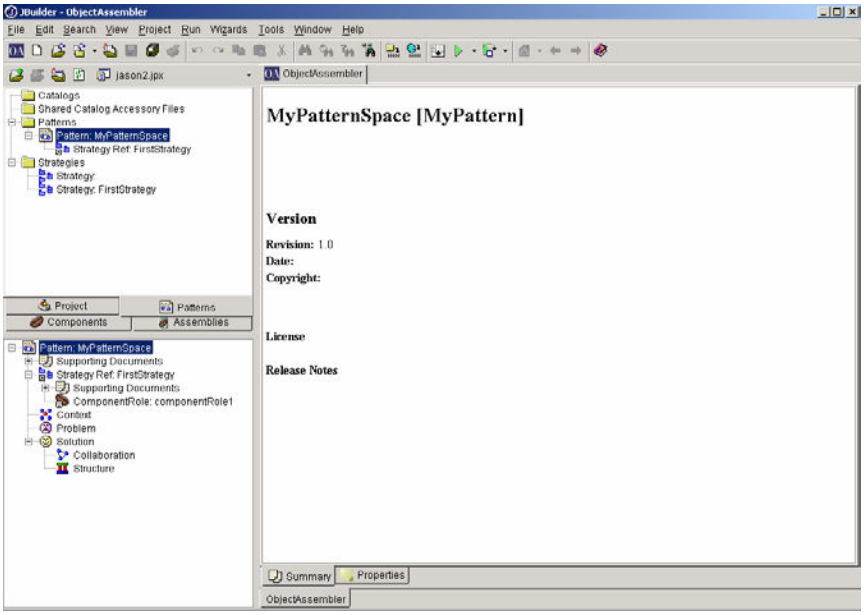
Editor View

The Editor View allows you to make changes to an application object in source code and XML descriptor files through form-field, text-entry input.

Message View

The Message View displays any error or warning messages about the currently selected assembly. To see errors and warnings for multiple assemblies at once, select the **Assemblies** folder in the Project View.

Patterns



The Patterns Workspace.

The following provides an overview of each View within the Patterns Workspace.

Project View

The Project View provides you an overview of the

- Catalogs
- Patterns
- Strategies

You can click on a catalog, pattern, or strategy to access the objects within that element. For example

- If you click on a Pattern in the Project View
  - A visual representation of that Pattern (with an expandable tree-structure) will appear in the Detail View. This tree structure will show all of the objects in a Pattern that include

- Detail Elements
- Context
- Problem
- Solution

Detail View

The Detail View provides you an overview of a catalog, pattern, or strategy in an expandable tree structure that includes all elements. If you click on an element, you can modify its properties[7] in the Editor View.

Editor View

The Editor View allows you to make property changes to pattern, catalog, or strategy elements through form-based, text-input fields.

Message View

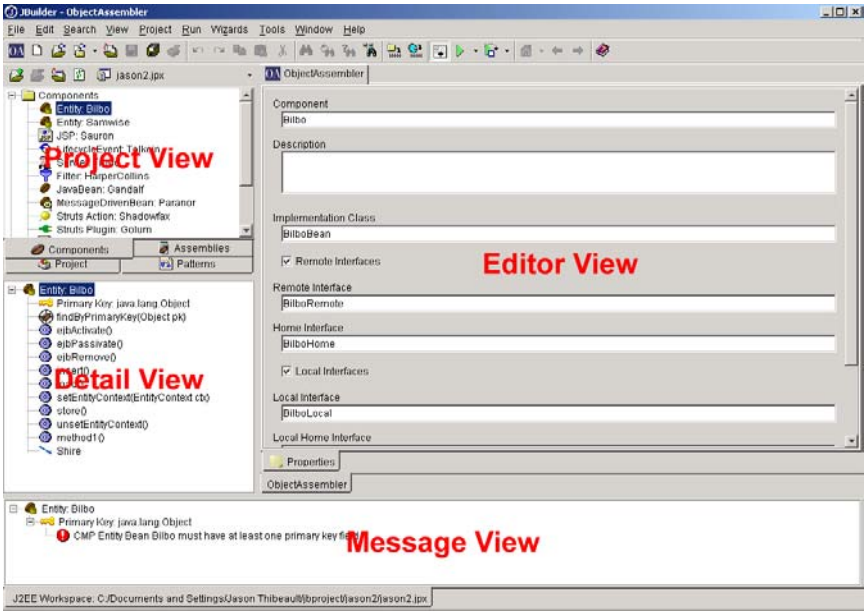
The Message View displays any error and warning messages about the currently selected catalog, pattern, or strategy. To see errors and warnings for multiple catalogs, patterns, or strategies at once, select the associated folder that contains them.

[7] Properties for Pattern elements (either catalogs, patterns, or strategies) are XML descriptor file information.

Views

Quick Tips:

Here's how it works:
1. Click on an object in the Project window
2. Expand that object in the Detail window
3. Click on one of the Object's properties and edit it in the Editor window
4. See real-time validation of the changes to the properties in the Message window



The Workspace Views.

Each Workspace in ObjectAssembler™ is broken into four main "views"

- Project—this view provides you a tree-based overview of all the objects within your project (i.e., components, assemblies, patterns, etc.).
- Detail—this view provides you an expanded view of an object when you click on it in the Project View. The selected object is displayed in the Detail window using a tree-based overview.
- Editor—this view provides you the ability to edit or view the specific properties and associated source code of a project object from the Detail window.
- Message—this view is where you see real-time validation of the changes to object properties using the InStep Validator™.

Project View

The Project View of each Workspace provides a high-level visual representation of the objects within the project. For example

- Files Workspace—the Project View provides a list of physical java files, markup documents, etc. within the project. This view is accessed in Netbeans/Forte by selecting the **Project** tab.
- Components Workspace—the Project View provides a visual list of the components, connectors (relationships between components), and pattern strategies to which components are mapped within the project.
- Assemblies Workspace—the Project View displays each type of Assembly (EAR, Java Jar, WAR – including Struts configurations, and EJB Jar) as well as Accessory Classes, Files, and JavaBeans.

- Pattern Workspace—the Project View displays Catalogs, Patterns, and Strategies within the application.

Detail View

The Detail View of each Workspace[8] provides detailed information about project objects. Every element of a project object is accessible here. For example

- Files Workspace—the Detail View provides a tree-view of each object's elements. This tree-like view includes
  - Imported classes
  - The class of the java file (i.e., EntityBean)
  - The specific methods
- Components Workspace—the Detail View provides a tree-view of the currently selected component's elements. This tree-like view includes elements such as methods, attributes, tags, etc.
- Assemblies Workspace—the Detail View provides a tree-view of the entire selected application assembly including
  - Accessory files, classes, and JavaBeans
  - Components you have designated as part of the application
  - Application properties (i.e., XML descriptor information)
- Pattern Workspace—the Detail View provides a tree-view of the specific PCML object—Catalog, Pattern, or Strategy—with corresponding XML descriptor file elements such as
  - Nested catalogs, patterns, and strategies
  - Artifacts
  - Authors

Editor View

The Editor View of each Workspace provides modification and text-entry capability. For example

- Files Workspace—the Editor View provides access to the source code by clicking on a project object from the Project View. Clicking on a specific method or class in the Detail View highlights that specific code within the Editor View. Any changes made to the source code are immediately evaluated by the InStep Validator™.
- Components Workspace—the Editor View provides form-field entry access to component elements displayed in the Detail View tree.

[8] The Detail View does not exist in the Files workspace of Netbeans/Forte, because all objects and elements are shown in the Project View.

- Assemblies Workspace— the Editor View provides form-field entry access to XML descriptor information for component and application elements displayed in the Detail View tree.

- Pattern Workspace— the Editor View provides form-field entry access to XML descriptor information for catalog, pattern, or strategy elements displayed in the Detail View tree.

Message View

The Message View works the same for each Workspace. It provides passive error and warning message alerts when there are specification violations or questionable coding practices. This view will only appear if there is a message to display.

ObjectAssembler™ Menu System

Because ObjectAssembler™ integrates with your existing IDE, there are not specific menus to learn for using the software. To access specific ObjectAssembler™ functions through your IDE, look for the "OA" icon in the menu system.

For example, in JBuilder, the "OA" icon can be found in the following menus

- **File >>** New J2EE Component
- **File >>** Import J2EE Components
- **Project >>** Generate Archive
- **Project >>** Deploy
- **Project >>** ObjectAssembler Project Options
- **Project >>** ObjectAssembler Environment Options
- **Help >>** About ObjectAssembler
- **Help >>** ObjectAssembler Help

Project Menus

The ObjectAssembler™ Project Menus allow you to generate and deploy your java applications.

The table below represents the ObjectAssembler™ Project Menu options and a brief explanation of their function.

Project Menu	Explanation
Generate Archive >> Generate EAR	Generate the EAR based upon the ObjectAssembler Project Options
Generate Archive >> Generate JAR	Generate the JAR based upon the ObjectAssembler Project Options
Generate Archive >> Generate WAR	Generate the WAR based upon the ObjectAssembler Project Options

Generate Archive >> Generate JAVA JAR	Generate the JAVA JAR based upon the ObjectAssembler Project Options
Deploy >> HPAS >> Deploy EAR	Deploy the assembly via the Hewlett Packard Application Server bypassing RadPak.
Deploy >> HPAS >> Deploy EJB JAR	Deploy the assembly via the Hewlett Packard Application Server bypassing RadPak.
Deploy >> HPAS >> Deploy WAR	Deploy the assembly via the Hewlett Packard Application Server bypassing RadPak.
Deploy >> HPAS >> Generate HPAS EAR	Generates JAR specific to HP Application Server
Deploy >> HPAS >> Generate HPAS EJB JAR	Generates JAR specific to HP Application Server
Deploy >> HPAS >> Launch RadPak with EAR	Open your EAR in HP RadPak toolset for editing or deployment
Deploy >> HPAS >> Launch RadPak with EJB JAR	Open your EAR in HP RadPak toolset for editing or deployment
Deploy >> HPAS >> Launch RadPak with WAR	Open your EAR in HP RadPak toolset for editing or deployment

Context Sensitive Menus

In addition to the sprinkling of the "OA" icon throughout the IDE menus, ObjectAssembler™ also has several context sensitive menus that can be accessed through a "second-button" click on

- Project Files (in the Project View of all Workspaces) such as
- Applications
- Patterns, Catalogs, and Strategies
- Components
- Components in the Detail view

These context-sensitive menus allow you to

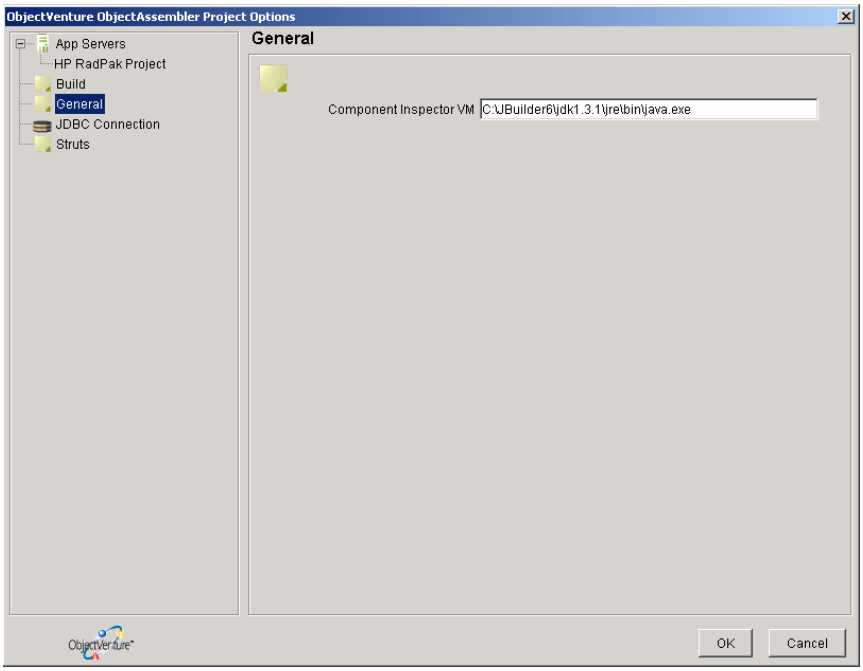
- Add component properties or elements
- Delete the clicked-upon element
- View source code

Setting ObjectAssembler™ Options

ObjectAssembler™ allows you to set two types of options

- Project Options—options that are specific just to this project (such as generated file locations)
- Environment Options—your working preferences for ObjectAssembler™.

Project Options



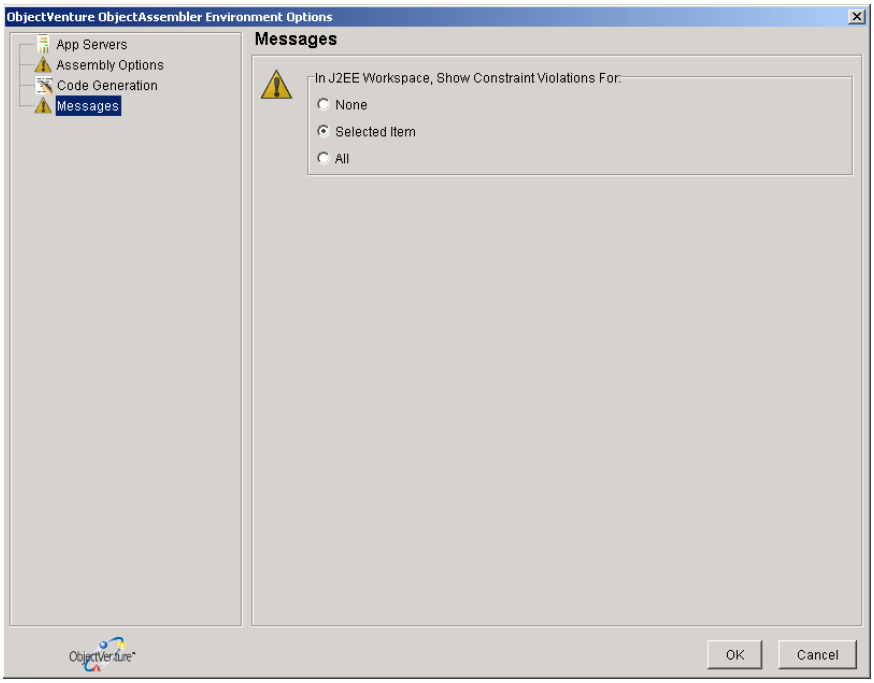
The Project Options Window.

For each ObjectAssembler™ project, you can set the following options for each project

- App Server Configuration—you can set specific configuration issues for deployment to supported servers[9].
- Build—you can set specific options for the way in which ObjectAssembler™ builds your applications including
  - Ignore warnings or Ignore errors and warnings
  - Skipping project autobuild and archive generation
  - Default filenames for application types such as EJB JARs, EARs, WARs, and Java JARs
- JDBC connection—URL, Driver, Database name, and password
- Struts—enabling struts and or wireless struts
- General—the location (on your harddrive) of the Component Inspector VM.

[9] In this current version of ObjectAssembler™, the only supported application server is HP RadPak.

Environment Options



The Environment Options Window.

For each ObjectAssembler™ project, you can set the following options for every project

- App Servers
- Assembly Options—You can choose whether or not to use an Assembly subdirectory underneath the Project Directory.
- Code Generation—for instantiated Patterns, you can include some versioning, descriptive, and authorial information for the generated code.
- Messages—You can set how sensitive ObjectAssembler™ will be regarding constraint violations in the J2EE workspace.

A Special Note About Entering Data in the Editor View

It is common practice (in many field-entry based editors) to type values into a text-field and then click the “OK” button to commit the changes.

Unfortunately, ObjectAssembler™ does not work this way. In order for you to commit your changes typed into form-fields within the Editor View, you must click on another text-entry box (or press the Tab key) prior to clicking the “OK” button. Doing so will commit your changes to the appropriate XML descriptor file or source code. This is especially important for most tables, where you must select another row before your current changes will be committed.

Chapter

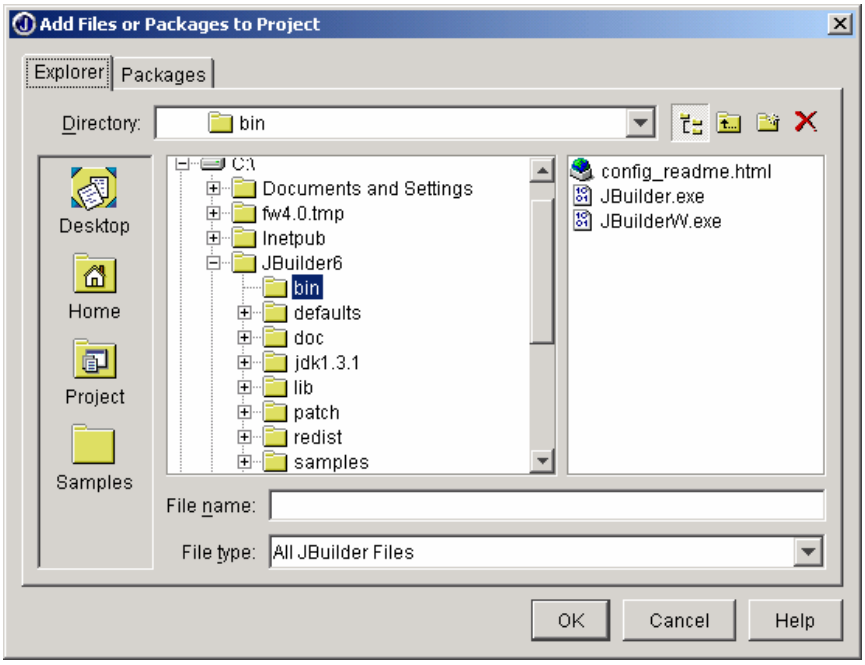
5

Working with Project Files

ObjectAssembler™-specific Files

The project files in your ObjectAssembler™ application are accessible from the Files View. These files are listed in the order you have specified in the host IDE.

Adding and Removing Accessory Files



Adding Accessory Files or Packages Window.

When building Java Applications with IDEs like JBuilder and NetBeans/Forte, you have the ability to add files (i.e., JavaBeans, JSP, Java source code) to your existing project so that you can incorporate it into your application as a component or other object.

In JBuilder, you can add accessory files through one of two means

- Context Sensitive Menus—You can click your secondary mouse button on the project files window to bring up the Project

Workspace, Project View menu. From here, you can select the **Add Files/Packages…** menu option.

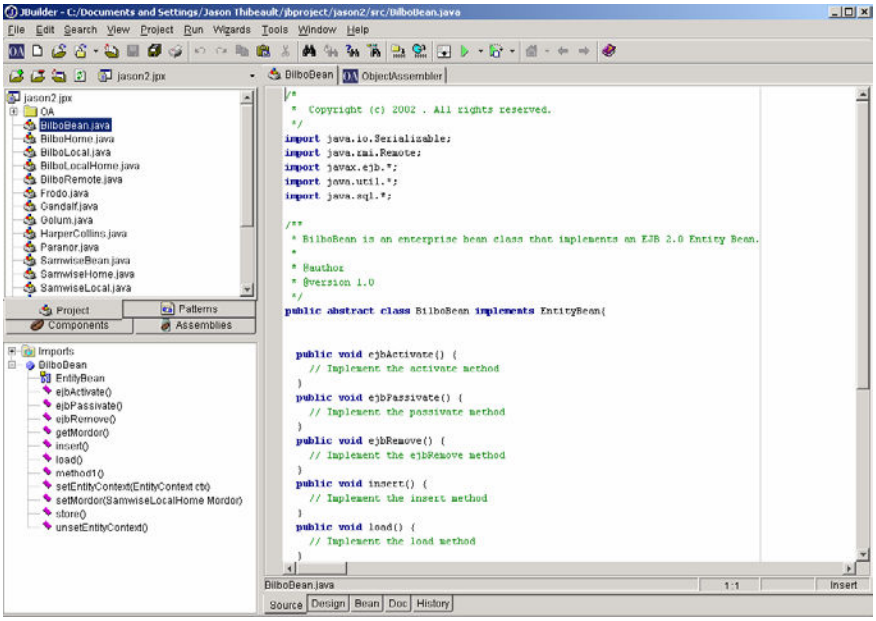
- Choose the **Add Files/Packages…** from the Project Menu option

In Netbeans/Forte, you can add accessory files through one of two means

- You can click your secondary mouse button on the **Project** node in the Project View of the Project Workspace to bring up the context sensitive menu. From here, you can select the **Add New…** or **Add Existing…** menu option.

Removing files is just as easy. Simply click on the file you want to remove (with the secondary mouse button) and select the **Remove From Project "{accessory file name}"** menu option (in JBuilder) or the **Delete** menu option (in Netbeans).

Viewing Source Code



Viewing source code in the Editor View.

You can view source code for any project file. This source code can be

- Java source code (i.e., JavaBeans, Java classes)
- Markup language source code (i.e., JSP, HTML)
- Descriptor information (i.e., XML)

Accessing source code for any type of file project can be accomplished in one of two ways

- Context Sensitive Menus—You can click your secondary mouse button on the project files window to bring up the Project Workspace, Project View menu. From here, you can select the **Open** menu option.

- Double click on the project file

With either method, you will open the source code into the Editor Window (and Detail Window if necessary).