What's new in Borland® JBuilder® 9

The latest enhancements to the leading Java™ development solution

A Borland White Paper

by Java Business Unit, Borland Software Corporation

May 2003

Contents

Introduction

Accelerate your Java™ development with Borland® JBuilder,® the leading, cross-platform environment for building industrial-strength enterprise Java applications. JBuilder® 9 Enterprise simplifies Web and Enterprise JavaBeans™ (EJB™) development with two-way visual designers and rapid deployment to J2EE™ platform application servers. Enhance productivity with EJB to Struts assembly, refactoring, HotSwap debugging, enterprise unit testing, performance tools, and version control integration.

JBuilder 9 contains changes in the following areas:

- Project management
- Build system
- Archive Builder
- Team development
- Web Services
- Productivity enhancements
- Editor enhancements
- Debugger
- Web application development
- EJB Designer
- J2EE servers
- Mobile development

Project management

The following features enhance your ability to manage projects easily.

- Dragging and dropping files and directories between and among projects
- Managing files in the project pane
- Opening TogetherSoft projects in JBuilder

Dragging and dropping between and among projects

Working with projects is easier as you can now drag and drop files and directories while working with them. You can:

- Quickly create a directory view in your project by dragging a directory from outside of JBuilder and dropping it into the project pane.
- Quickly add a file to a directory view by dragging a file from outside of JBuilder and dropping it on the directory view in the project pane.
- Drag and drop files between parent nodes in the project pane.
- Drag and drop a copy of a file or directory node to another project pane in a different AppBrowser.
- Quickly open a file by dragging it from the project pane and dropping it on the content pane.
- Reorder projects within a project group by dragging projects to their new location.

Managing files in the project pane

Now you can not only quickly add or create files and packages from the context menu of the project pane, but you can create a directory or any file type within the Web application root

directory or subdirectory. Right-click on the Web application root directory or subdirectory, and choose from the context menu.

You can select and expand an archive node in the project pane and see the files it contains. By double-clicking a file node in an archive, you open it in the content pane for viewing and editing. The JBuilder viewer used to display the file depends on the file type. For example, if your archive includes a `.gif` file and an HTML file, the View page of the content pane uses JBuilder's image viewer to display the `.gif` file and its browser viewer to display the HTML file.

Opening Borland® Together® projects in JBuilder®

JBuilder can now open Borland® Together® ControlCenter® projects (`.tpr` files) as JBuilder projects in the JBuilder project pane. For information, see "Opening Together ControlCenter projects in JBuilder" in the JBuilder documentation.

Build system

Changes to Borland Make compiler

Borland® Make, the default compiler in the JBuilder integrated development envrionment (IDE), has been re-architected to use the standard `javac` compiler in conjunction with smart dependencies checking. If you're using JBuilder® Developer or Enterprise, you can continue to use the old version of Borland Make (JBuilder 8), which uses smart dependencies checking but doesn't use `javac`. To switch compilers, choose **Project|Project Properties**, click the **Build** tab and the **Java** tab. Choose a compiler from the Compiler drop-down list. Switching compilers is a feature of JBuilder Developer and Enterprise.

The error messages in the structure pane and in the message pane have changed due to the changes to Borland Make. For help on an error message, compile the project, choose an error message in the message pane, and click **F1**.

Apache™ Ant

Ant has been updated to version 1.5.2.

Exporting a project to Ant

This is a feature of JBuilder Developer and Enterprise.

JBuilder supports exporting a JBuilder project to an Ant build file that contains standard Ant tasks. You can then use this Ant build file to build your project independently of JBuilder. This is useful for executing command-line integration builds and for licensing reasons.

The Export To Ant wizard, available on the Wizards menu and on the Build page of the object gallery (**File**|**New**), creates an Export To Ant node in the project pane. Right-click the node and choose **Make** to generate the Ant build file.

It's important to note that JBuilder builds and Ant builds aren't necessarily identical, because some JBuilder custom Ant tasks aren't equivalent to standard Ant tasks. As a result, some JBuilder build tasks aren't supported.

Archive Builder™

This is a feature of JBuilder Developer and Enterprise.

The Archive Builder™ allows more flexible control over what files make up an archive than do previous versions. Following are the main features:

- The ability to specify files to include and exclude.
- The ability to specify exclusion and inclusion filters, where filters allow for the usage of wildcards, and recursion.
- The ability to put files from any location into an archive and specify its target path and name.

Team development

The following version control system (VCS) integrations have been updated:

- Concurrent Versions System (CVS)
- Microsoft® Visual SourceSafe®
- Rational® ClearCase®

The following integration has been added:

- Borland® StarTeam®

CVS integration updates

The following features were added to the CVS integration:

- Automatic checkout support when refactoring: when read-only files are encountered during refactoring, and the files are under version control, the refactoring is stopped, and you are prompted to check out the files.
- Ability to change actions for multiple selected files: the context menu for selected files in the table in the Commit Browser allows you to change actions for multiple files.
- Revert capability: the Revert command in the Team menu discards any changes in the edit buffer, and reverts the active file back to the most recent checked out version.
- Ability to pull a project from CVS by date: the Pull Project From CVS wizard provides the capability to check out the most recent revision of a project, no later than the specified date.

- Automatic refresh of the History view: the History view updates automatically after performing Commit or Checkin operations.

For more information about the CVS integration for JBuilder, see "CVS in JBuilder" in *Team Development using JBuilder* in the JBuilder documentation.

Microsoft® Visual SourceSafe® integration updates

Microsoft® Visual SourceSafe® integration is a feature of JBuilder Developer and Enterprise.

The following features were added to the Visual SourceSafe integration:

- Rename support in Commit Browser: when a file under version control is renamed in the workspace, the Commit Browser provides an option to rename the file in the repository, or revert the file in the workspace back to the original name.
- Automatic checkout support when refactoring: when read-only files are encountered during refactoring, and the files are under version control, the refactoring is stopped, and you are prompted to check out the files.
- Ability to change actions for multiple selected files: the context menu for selected files in the table in the Commit Browser allows you to change actions for multiple files.
- Automatic refresh of the History view: the History view updates automatically after performing Commit or Checkin operations.
- Revert capability: the Revert command in the Team menu discards any changes in the edit buffer, and reverts the active file back to the most recent checked out version.

For more information about the Visual SourceSafe integration for JBuilder, see "Visual SourceSafe in JBuilder" in *Team Development using JBuilder* in the JBuilder documentation.

Rational® ClearCase® integration updates

Rational ClearCase integration is a feature of JBuilder Enterprise.

The following features were added to the ClearCase integration:

- Rename support in Commit Browser: when a file under version control is renamed in the workspace, the Commit Browser provides an option to rename the file in the repository, or revert the file in the workspace back to the original name.
- Automatic checkout support when refactoring: when read-only files are encountered during refactoring, and the files are under version control, the refactoring is stopped, and you are prompted to check out the files.
- Ability to change actions for multiple selected files: the context menu for selected files in the table in the Commit Browser allows you to change actions for multiple files.
- Automatic refresh of the History view: the History view updates automatically after performing Commit or Checkin operations.

For more information about the ClearCase integration for JBuilder, see "Rational ClearCase in JBuilder" in *Team Development using JBuilder* in the JBuilder documentation.

Borland® StarTeam® integration

Borland StarTeam integration is a feature of JBuilder Developer and Enterprise.

The StarTeam integration provides access to the most critical StarTeam features and functions so you can perform version control and configuration management tasks from within JBuilder. The integration connects to the StarTeam Server using the TCP/IP (Sockets) protocol. The integration also provides some JBuilder-specific features to allow you to easily check in and check out JBuilder project source files.

The StarTeam integration incorporates wholly large portions of the StarTeam Cross-Platform Client into the JBuilder development environment, and includes the StarTeam Cross-Platform Client itself. The StarTeam integration for JBuilder supports StarTeam 5.2 and 5.3 Servers.

In addition to all the features provided by StarTeam, some highlights of the integration include:

- Status Browser (**Team|Status Browser**) support: the Status Browser is primarily a viewing tool. It browses the active project, and displays the version control status of each changed file, the source for each available version, and differences between any two versions of a changed file.

- Commit Browser (**Team|Commit Browser**) support: the Commit Browser provides the viewing capabilities of the Status Browser, and provides access to common version control operations for files that have changed. With the Commit Browser, you can set the version control command you want to apply to each changed file, enter comments for individual files and for the whole group, then execute all of the commands with one click. When a file under version control is renamed in the workspace, the Commit Browser provides an option to rename the file in the repository, or revert the file in the workspace back to the original name.

- Placing Project Into StarTeam wizard (**Team|Place Project Into StarTeam Server**): the Placing Project Into StarTeam wizard lets you add a JBuilder project to the StarTeam repository.

- Pull Project From StarTeam wizard (**Team|Pull Project From StarTeam Server**): the Pull Project From StarTeam wizard sets the project's connection to the StarTeam Server, and pulls (checks out) a JBuilder project from the server into your work area. Once you pull a project, you can perform source control activities on the project files, such as checking files in and out.

- Process item shortcuts: process item shortcuts provide quick and comprehensive access to property editors for process items and topics. Process item shortcuts appear as subnodes of the StarTeam Repository node in the project pane. The context menu

for shortcuts lets you open the property editor for the process item in the content pane. This modeless property editor lets you view and modify process item properties while you work on your project. The shortcut context menu also allows you to set or clear the active process item, remove a shortcut, or rename a shortcut. The Create Short Cut button in the toolbar for the StarTeam Repository creates shortcuts for selected process items. This new feature replaces the StarTeam ProcessItem node in the project pane, and the Edit Active Process Item on the Team menu.

Note: The modeless property editor is not supported for alternate property editors (APEs). If your team has set up StarTeam to use alternate property editors (APEs), the shortcut context menu includes an Edit Item command for opening the APE in a modal dialog box.

For more information about the StarTeam integration for JBuilder, see "StarTeam in JBuilder" in *Team Development using JBuilder* in the JBuilder documentation.

Web Services

This is a feature of JBuilder Enterprise.

Apache® Axis toolkit

The Apache™ Axis toolkit has been updated to version 1.1.

SOAP toolkit

Apache SOAP has been updated to version 2.3.

BEA® WebLogic® toolkit

The BEA®WebLogic® toolkit has two new features: the Web Services Console and the Export As An Asynchronous Web Service wizard.

Web Services Console

The Web Services Console provides direct access for testing your deployed service on the WebLogic Web Services Home Page. Configure the project for Web Services, create a Web Service, and deploy the service to the WebLogic Server as you normally would. Then, choose **Tools|Web Services Console** to open the testing page in Internet Explorer.® The Web Services Console lists the deployed services, the exposed methods (operations) in the service, and the Web Services Definition Language (WSDL). You can test the service on this page, view the SOAP request and response messages from a successful execution of an operation, and use the provided example code to invoke the service.

Export As An Asynchronous Web Service wizard

This is a feature of WebLogic Server 8.1.

The Export As An Asynchronous Web Service wizard, a feature of the WebLogic toolkit and WebLogic Server 8.1, can be used to create an asynchronous Web Service. Because the WebLogic toolkit supports Java™ Message Service (JMS) -implemented Web Services using queue as the messaging model, you must configure your WebLogic Server for JMS. Your project must also contain an EJB module and a message-driven bean before you can create the service.

The Export As An Asynchronous Web Service wizard is available on the Web Services tab of the object gallery (**File|New**) and the project pane context menu.

The typical steps involved in developing an asynchronous web service include:

1. Configure a JMS server, a destination queue, and an optional receive queue on WebLogic Server 8.1.

2. Create an EJB module and a message-driven bean, which processes JMS messages from the client.

3. Create a data type, such as a serialized JavaBean, to send between the client and the server.

4. Create a Web Services implementation to send the data to the server and optionally receive return data.

Supported enterprise application servers

The JBuilder Web Services features are supported on the following enterprise application servers:

- BEA® WebLogic® toolkit: WebLogic® Server 7.0 SP2 and 8.1
- Axis and SOAP toolkits:
    - Borland® Enterprise Server 5.1.x, 5.2, and 5.2.1
    - BEA WebLogic Server 7.0 SP2 and 8.1
    - IBM® WebSphere® Application Server 4.0 AES/AE and 5.0
    - Sybase® EAServer 4.1.3 and 4.2

Productivity enhancements

JBuilder offers faster coding and code documentation with the following new coding tools and enhanced productivity features.

- Javadoc
- Refactoring
- ErrorInsight ™

Javadoc

The Javadoc tab of the **Project Properties|Build** page allows you to add custom tags for Javadoc. You can also add heading text for a custom tag. The heading text is rendered in the Javadoc HTML output file as heading-level text.

The Show Console Output option on the Choosing documentation build options page of the Javadoc wizard now displays Javadoc output in the Build tab of the message pane. Javadoc output will be displayed with compiler messages and other output from the build process.

Refactoring

You can now use a new refactoring command, Find Overridden Method, to find the method overridden by the selected method. To find the overridden method, open the source file containing the declared method. Right-click the method in the structure pane or the editor and choose **Find Overridden Method**. JBuilder opens the superclass where the overridden method is declared and positions the cursor on the method declaration. You can travel up the chain of superclasses to find all of the overridden superclass methods.

ErrorInsight™

Code correction is easier than ever with the ErrorInsight tool. ErrorInsight allows you to quickly correct common coding errors with many error correction suggestions and wizards. See ErrorInsight in *Introducing JBuilder* in the JBuilder documentation for more information. You can access ErrorInsight several ways.

- Choose **Edit|ErrorInsight**.
- Right-click the error description adjacent to the **ErrorInsight** icon in the structure pane.
- Click the **ErrorInsight** button in the left gutter of the editor adjacent to the source code error.

Editor enhancements

You now have more choices for IDE customization and new time-saving tools to speed up your coding with the following features.

- Customizing the IDE
- Using auto import
- Editing with Sync Edit
- Adding macros
- Adding common code templates

Customizing the IDE

You can change the look and feel of JBuilder's IDE by choosing **Tools|IDE Options**. JBuilder has added a new Borland option to the look and feel drop-down list, which includes the Windows®, Metal, and CDE/Motif look-and-feel options.

Using auto import

Code templates now include an import statement at the top of the code snippet (**Tools|Editor options|Templates**). When you insert the code template into your source code (**Ctrl+J**) the import statement is stripped from the template and correctly placed at the top of your source code.

Editing with Sync Edit

The Sync Edit tool allows you to edit duplicate identifiers simultaneously in code or code templates.

To invoke Sync Edit from the main menu,

1. Select code or a code template that contains duplicate identifiers.

2. Choose **Edit|Sync Edit**.
   You are now in Sync Edit mode. Your code selection is highlighted and duplicate identifiers are either outlined or underlined in blue.

To invoke Sync Edit from the editor,

- Press **Ctrl+J** and select a template to insert into your code.
  You are now in Sync Edit mode.

- Select the area where you want to add a code template.
  The Sync Edit button displays in the left gutter of the editor. You can click the Sync Edit button to add code templates or to simultaneously edit duplicate identifiers.

Sync Edit correctly inserts the template and automatically indents your code.

Adding macros

You can make your coding even more efficient by adding macros to your code templates. Macros can automate replacing identifier names in the source code when you insert a code template. For example, if you select the `equals` template (includes the `$CurrentClass` macro) to insert into your code, the `$CurrentClass` macro is automatically replaced by the actual class name in your source code.

To add macros to a template,

1. Choose **Tools|Editor Options** and click the **Templates** tab.

2. Select the template from the Templates list from either the Java or HTML page.

3. Place the cursor in the Code text field where you want to insert the macro.

4. Click the **Insert Macro** button on the bottom of the Templates page.
   The Insert Macro dialog box opens.

5. Select the macro from the list and click **OK**.
   The macro is inserted into the code template.

Adding common code templates

You can easily add code templates to your code template list that will display regardless of file type or location within that source file.

1. Choose **Tools**|**Editor Options** and click the **Templates** tab.

2. Click the **Add** button on the Templates page.
   You can choose to add a code template to your Java, HTML, or Common template list. The common templates are always displayed regardless of file type or location within that file.

3. Enter the name and description of the new code template, then press **OK**.

4. Type the code for the new template in the Code text editing area.
   Remember to include the appropriate import statement.

5. Add or edit other templates, then click **OK** when you are finished to close the Editor Options dialog box.

Debugger

The following new features have been added to the Debugger:

Displaying an object as a String

The Show toString() command executes the `toString()` method on the selected object (including the `this` object) and displays the resulting string. For example, if the selected object is `City`, the Show toString() command will display the object value, such as `San Francisco` rather than `com.mycode.City@391`. This command is available when you select an object in the Data watches view and the Threads, call stack, and data view.

Going to a scoped variable watch

You can go to the method in the editor where the variable for the selected scoped variable watch is defined. (A scoped variable watch watches the variable in the scope in which you created the watch. You create scoped variable watches in the Threads, calls stacks, and data view.) To go to the method, right-click a scoped variable watch in the Data Watches view and choose **Go To Watch**.

Web application development

The following web application development features are new:

Access to the JavaServer Pages™ wizard

In addition to the Web page in the object gallery, there's a new way to access the JavaServer Pages™ (JSP™) wizard. You can now access this wizard by right-clicking the Root Directory node of a WebApp in the project pane and selecting **New|JSP** from the context menu.

Directory inclusion rules for WEB-INF

Directory inclusion rules for subdirectories of the `WEB-INF` directory have changed. If Include Regular Content In WEB-INF And Subdirectories is checked on the Directories page of the WebApp Properties dialog box, all the subdirectories of `WEB-INF` are automatically

included in the project. In previous versions, each subdirectory had to be included manually. The UI on the Directories page of the WebApp Properties dialog box for including those subdirectories has been removed.

If `WEB-INF` is treated as a regular directory, the contents of `WEB-INF/classes` and `WEB-INF/lib` are shown, but intended as a read-only view. Options to rename or delete the files under those directories are disabled.

Directory exclusion for a WebApp

A new context menu option for all directories in a WebApp adds an exclusion rule to exclude that directory (and all the directories underneath it). To exclude a directory, right-click it in the project pane and select **Exclude Directory From WebApp** from the context menu. The rule can be removed from the Directories page of the WebApp Properties dialog box.

Servlet wizard

The Generate SHTML File option and the two related options on the Edit Standard Servlet Details page of the Servlet wizard have been replaced by the Generate HTML File With Form option. This option allows you to directly test `doPost()` on a servlet or test different parameters on a `doGet()`.

The Generate HTML File With Form option and the resulting form's ACTION attribute depend on the `doGet()` and `doPost()` method selections:

doGet()	doPost()	Result
Not selected	Not selected	Generate HTML File With Form option disabled
Selected	Not selected	Generated form's ACTION attribute set to GET
Not selected	Selected	Generated form's ACTION attribute set to POST

Selected	Selected	Generated form's ACTION attribute set to POST

The Enter Servlet Request Parameters page has been switched with the Enter WebApp Details page of the wizard. For a standard servlet, the order of the pages on the Servlet wizard is:

1. Choose **Servlet Name And Type**
2. Enter Standard Servlet Details
3. Enter Servlet Request Parameters
4. Enter WebApp Details
5. Define Servlet Configuration

Apache™ Struts 1.1 support

JBuilder can now detect if the `struts-config.xml` file supports Apache™ Struts 1.0 or Struts 1.1. Parameters in the Struts Config Editor are displayed accordingly.

The Properties For Struts dialog box forces JBuilder to reread the `web.xml` file and locate new or changed `struts-config.xml` files. For Struts 1.1, JBuilder then scans those Struts configuration files to locate the Tiles definition file(s). You can also use the Properties For Struts dialog box to add or remove configuration files and reorder them in the `web.xml` file. To display this dialog box, right-click the Struts node in the project pane and choose **Properties**.

Support for adding files or directories to WebApp

You can now directly add a directory, JSP file, HTML file, or any other file type to the WebApp or its sub-directories (including the Root Directory). To do this, go to the project pane. Choose the WebApp or the WebApp directory you want to add to, right-click and

choose **Add To WebApp**. (You can add either files or directories.) In the dialog box, enter the file or directory you want to add and click **OK**.

JSP™ wizard

The Identify JSP Beans page of the wizard has been modified. In addition to adding and removing JSP beans from this page, you can now edit an existing bean's ID and Scope.

EJB™ Designer

The performance of the EJB Designer has been enhanced. Also, the **Tools|IDE Options|EJB Designer** page has new options you can set to quicken the display of the EJB Designer. Click the **Help** button for more information.

A DTO/Session Facade wizard has been added to generate data transfer objects (DTOs) and session facades for your entity beans. For complete information, see "Creating session facades for entity beans" in the JBuilder 9 documentation.

J2EE™ servers

JBuilder now supports these additional servers:

- BEA WebLogic Server™ 8.1.
- IBM® WebSphere® Application Server 5.0.
- Borland® Enterprise Server 5.1.1 - 5.2.1
- Sybase® EAServer 4.2.

Support has also been added for these service packs:

- BEA WebLogic Server™ 6.1, service pack 4
- BEA WebLogic Server™ 7.0, service pack 2

Support for BEA WebLogic Server™ 5.1 and IBM® WebSphere® Server 3.5 has been removed.

Sybase® Enterprise Application Server

If you use the Sybase EAServer, you no longer need to map the fields for container-managed persistance (CMP) EJBs using the Sybase Jaguar Console. The JBuilder OpenTool plugin for Sybase now handles this. Also, for relationship components, you no longer need to update the datasource to be the same as that used by the EJB components involved in the relationship.

Updating projects with the latest server settings

Every time you open a project, JBuilder updates it with the latest server settings for the project's selected server(s). For example, if you have modified server settings for one project and you have others that use the same server configuration, the next time you open those other projects, your modified server settings will be in force automatically. If you want to use a similar but unique server configuration for a particular project, use the **Copy** button, and use the Copy Servers wizard to edit it to your needs.

Copying a server configuration

You can copy an existing server configuration so you can edit it to make changes. For information, see "Creating a duplication configuration to edit" in the JBuilder documentation.

Mobile development

JBuilder mobile development support, formerly called JBuilder MobileSet, has been incorporated into the core JBuilder Developer and Enterprise editions. It is an optional choice

during JBuilder installation. The mobile development feature does not work in JBuilder® Personal.

JBuilder mobile development has the following new features on the Upload page of the Over the Air (OTA) Manager:

- A checkbox that allows you the option of remembering the password. If this checkbox is checked, the password is saved to the disk. Otherwise, it will be removed.

- A checkbox called Specify URL Prefix for JAR URL Property that allows you to override the MIDlet-Jar URL in the JAD file with a specific prefix before it is uploaded to the server. There is also a text field where you can specify a URL prefix. The OTA manager will add to or replace the URL prefix specified in the JAD file with this one when the Override JAR URL Property option is checked.

Conclusion

JBuilder 9 includes a wealth of capabilities to help accellerate your Java development, let you work the way you want to work, and help you build better software faster. For additional information, please visit the Borland Web site or speak with your Borland representative.

Useful links

JBuilder product documentation

http://www.borland.com/jbuilder

JBuilder product documentation

http://info.borland.com/techpubs/jbuilder/jbuilder8/index1280x1024-ent.html

Optimizeit Suite product documentation

http://info.borland.com/techpubs/optimizeit/optimizeit5/index1280x1024.html

JBuilder certification

http://www.borland.com/services/certification/index.html

Training on Borland products

http://www.borland.com/services/training/course_descriptions.html

Borland newsgroups

http://info.borland.com/newsgroups/