Borland®

JBuilder™ 4 Professional Edition

Extending Your Java® Development Capabilities from JBuilder™ 4 Foundation Edition Using JBuilder 4 Professional Edition

by Jason Vokes, Borland

Table of Contents

Introduction

Borland has taken the lead among development tool vendors for many years. Borland® JBuilder,™ the award-winning Java® development suite from Borland has been no exception. With every new release of JBuilder, Borland has set itself ahead by providing new levels of support for the latest Java standards, increased developer productivity, maximum database support, and tightly integrated enterprise technologies such as CORBA® and J2EE.™

Businesses around the world are gaining significant benefits from the use of Java technologies through Borland JBuilder. In fact, developers can benefit from the Borland JBuilder "experience" by downloading the JBuilder Foundation Edition free from the Borland website (http://www.borland.com/jbuilder/foundation/); however, this is rarely the end of the story. New Java technologies are rapidly emerging and, to ensure that developers are able to provide maximum benefit to an organization, they need the correct level of support within the development tools that they use.

In this white paper, the discussion will focus on how developers can build on the experience they have gained using JBuilder 4 Foundation Edition to become a cohesive team of highly productive, commercial Java Internet and database developers by upgrading to JBuilder 4 Professional Edition. It will assume a level of familiarity with JBuilder 4 Foundation Edition.

JBuilder

white paper

Definitions

Acronym	Description
API	Application Programming Interface
CORBA®	Common Object Request Broker Architecture
EJB™	Enterprise JavaBeans®
HTML	Hyper Text Markup Language
IDE	Integrated Development Environment
J2EE™	Java 2 Enterprise Edition
JDBC®	Java Database Connectivity
JDK™	Java Development Kit
JPDA	Java Platform Debugger Architecture
JSP™	Java Server Pages
ORB®	Object Request Broker
RDBMS	Relational Database Management System
SQL	Structured Query Language
WML	Wireless Markup Language
XHTML	Extended Hyper Text Markup Language
XML	Extended Markup Language

Additional info

Borland® JBuilder™
http://www.borland.com/jbuilder/
Borland® Application Server™
http://www.borland.com/appserver/

Overview

Whether you have committed to delivering a commercial application within a certain timeframe and budget, or you simply want to know that you are not reinventing the wheel every time you write a piece of code, it's now time to look beyond JBuilder 4 Foundation Edition to the increased features available in JBuilder 4 Professional Edition, including:

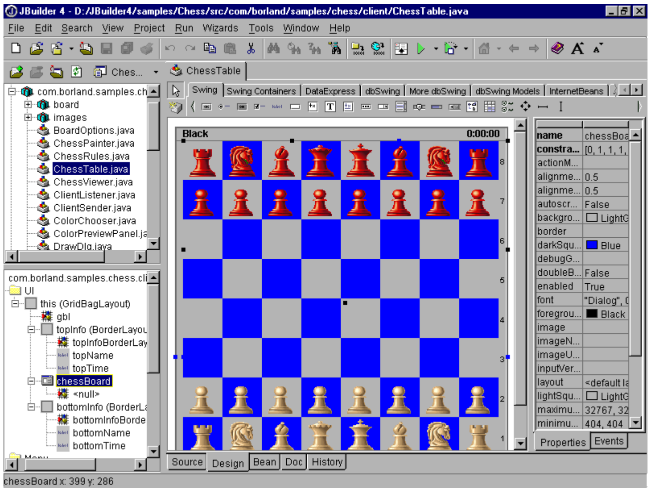
- time-saving wizards that jump-start application development and free developers from mundane coding tasks, enabling them to maintain focus on providing the real solutions.
- advanced debugging features that help developers to resolve problems quickly, enabling the highest levels of productivity to be maintained.
- integrated Web development technologies to speed the coding and debugging of Java Server Pages (JSP™) and Servlets
- advanced and comprehensive support for working with corporate databases
- code history and file system based source versioning, integrated for team development capabilities that will help keep both project development and management staff in sync.

With added functionality such as this, you can be sure that you and your developers are getting maximum benefit from the time that you invest in developing your Java applications.

If you have become familiar with JBuilder Foundation Edition, it's time to extend your development horizons. Anyone that has used JBuilder 4 Foundation Edition will be instantly at home using JBuilder 4 Professional Edition. There are some differences that are immediately apparent-more Wizard features are present and an increased number of components are available. There are also many other, less directly obvious, capabilities.

Integrated Development Environment (IDE)

First, let's take a look at some of the productivity and visual aspects in which JBuilder 4 Professional is more advanced than JBuilder Foundation.



JBuilder 4 - D:/JBuilder4/samples/Chess/src/com/borland/samples/chess/client/ChessTable.java

name	chessBoa
constra...	[0, 1, 1, 1,
actionM...	
alignme...	0.5
alignme...	0.5
autoscr...	False
backgro...	□ LightG
border	
darkSqu...	■ Blue
debugG...	
doubleB...	False
enabled	True
font	"Dialog", (
foregrou...	■ Black
image	
imageN...	
imageU...	
inputVer...	
layout	<default la
lightSqu...	□ LightG
maximu...	32767, 32
minimu...	404, 404

chessBoard x: 399 y: 286

Figure 1: The JBuilder 4 Professional Edition IDE.

As you can see in Figure 1, JBuilder 4 Professional Edition and JBuilder 4 Foundation Edition are initially very similar to look at. There are, however, many additions and improvements in the Professional Edition that empower the developer, as well as making Java application development of any kind much easier.

Advanced Syntax Highlighting

Developing a project using Java technologies often involves multiple file formats, such as Java, HTML, JSP, and XML among others. JBuilder 4 Professional provides Advanced Syntax Highlighting, which recognizes the file type that the developer wants displayed and highlights the varying syntaxes accordingly. JBuilder understands the syntax of file types that are important to Java developers and ensures that developers can focus on the important sections of code, thus saving time in code comprehension and enabling the developer to provide the right solutions in the minimum time.

Multiple Executables per Project

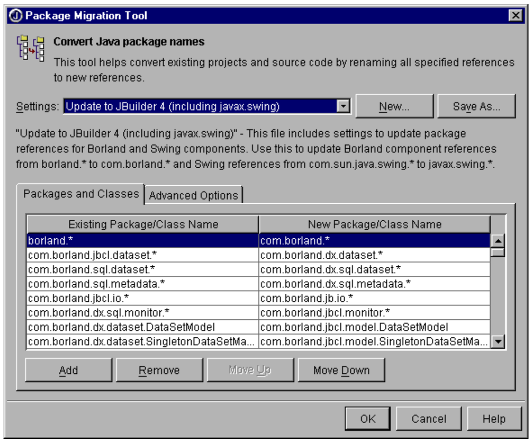
Java projects can quickly get to be complex, with some requiring several applications to work together to offer a solution. This would be the case when any client and server need to be developed and tested. JBuilder 4 Professional Edition enables the developer to include more than one executable in the same project. In this way, a developer can write, run, and debug these applications all within the same project and the same instance of JBuilder. Imagine how much more productive you are going to be when you can simultaneously run and debug multiple applications.

JDK® Switching

One of the first questions when approaching a Java development process is to consider which JDK version your application is going to run against. The Java JDK is a set of classes that provide the building block objects by which we may create applications. There are many different versions of JDKs available and, over a period of time, developers will doubtless be required to either write new applications or to maintain existing code that uses different JDKs. JBuilder 4 Professional allows the developer to choose which JDK they wish to develop against. This means that JBuilder 4 Professional becomes the one-stop-shop for all your Java development and that the reinstallation of various development tools, according to the job-in-hand, becomes a thing of the past. This saves the developer time and inconvenience, not to mention the requirement for multiple development tool licensing per developer seat.

Package Migration

The Package Migration Tool facilitates automated project and source code conversion by undertaking the renaming of all package and class references to new references. This enables developers to rapidly and painlessly update existing code to use new class and package versions as they become available, or to update existing code as part of code maintenance.



Convert Java package names

This tool helps convert existing projects and source code by renaming all specified references to new references.

Settings: Update to JBuilder 4 (including javax.swing)

"Update to JBuilder 4 (including javax.swing)" - This file includes settings to update package references for Borland and Swing components. Use this to update Borland component references from borland.* to com.borland.* and Swing references from com.sun.java.swing.* to javax.swing.*.

Existing Package/Class Name	New Package/Class Name
borland.*	com.borland.*
com.borland.jbcl.dataset.*	com.borland.dx.dataset.*
com.borland.sql.dataset.*	com.borland.dx.sql.dataset.*
com.borland.sql.metadata.*	com.borland.dx.sql.metadata.*
com.borland.jbcl.io.*	com.borland.jb.io.*
com.borland.dx.sql.monitor.*	com.borland.jbcl.monitor.*
com.borland.dx.dataset.DataSetModel	com.borland.jbcl.model.DataSetModel
com.borland.dx.dataset.SingletonDataSetMa...	com.borland.jbcl.model.SingletonDataSetMa...

Figure 2: The Package Migration Tool takes the legwork out of package conversion and updating.

Code Templates Within a Team Environment

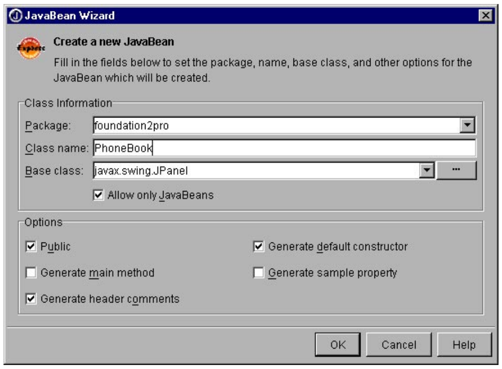
To free developers from mundane coding tasks, JBuilder 4 Professional Edition enables developers to configure their own code templates. By including the Edit, Add, and Delete capability to the code templates that are activated by using CTRL-J from within the editor, developers need only write routine code sections once. Such code sections can be added as a template and then, when needed, be inserted into the code editor within a couple of keystrokes. This frees the developer from the time-consuming task of remembering mundane or lengthy code syntax and prevents needless or repeated reference to coding manuals.

JavaBeans®

Software development using components is now considered vital to the cost-effective development of commercial software. JavaBeans is the name for components that comply with the Java coding conventions. Since its first release, JBuilder has been considered as the development tool for JavaBeans development, and now, with JBuilder 4 Professional Edition, developers get increased support for raising the percentage of actual source code reuse via use of JavaBean component development.

Wizard for JavaBeans

The Wizard for JavaBeans is used to rapidly create and modify JavaBeans components. This saves developers from the task of producing and maintaining the code required to produce Java components that encapsulate coded solutions, which can be unit-tested and then used over and over again.



Create a new JavaBean

Fill in the fields below to set the package, name, base class, and other options for the JavaBean which will be created.

Class Information

Package: foundation2pro

Class name: PhoneBook

Base class: javax.swing.JPanel

☑ Allow only JavaBeans

Options

☑ Public
☑ Generate default constructor
☐ Generate main method
☐ Generate sample property
☑ Generate header comments

Figure 3: The JBuilder Wizard for JavaBeans radically simplifies the coding of JavaBeans components.

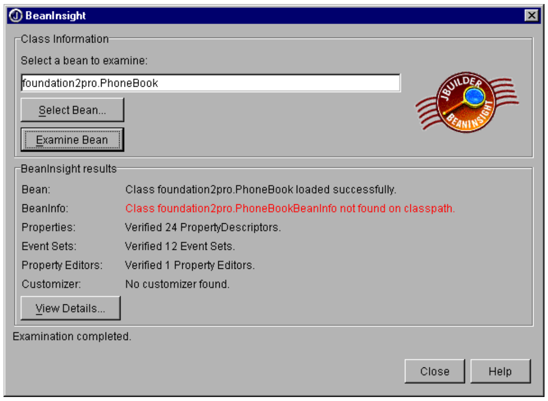
JBuilder 4 Professional Edition also speeds the high-standard production of JavaBean components by providing Visual Bean Designers to easily create properties, events, and BeanInfo.

Source code for 250+ beans

There is no better way to learn from the experts than to be shown the source code they have written. That's exactly what developers can do with JBuilder 4 Professional Edition: the source code is included for the 250+ JavaBeans that are available.

BeanInsight™

There are times when it will not be viable to develop the JavaBean that you require in-house. You may find just the functionality that you need available from a third party, or downloadable from the Internet. In this case, you are making your customer dependent on code from another party. This is not something that you should do lightly or without thoroughly checking out the code that you are including in your project. In order to help you inspect the prospective JavaBean, JBuilder 4 Professional Edition equips developers with BeanInsight. BeanInsight will verify that the JavaBeans are indeed valid and provide information on properties, events, BeanInfo, property editors, and so on. In a matter of seconds, BeanInsight will perform a task that would take a skilled developer a great deal of time and patience.



Select a bean to examine: foundation2pro.PhoneBook

Bean:	Class foundation2pro.PhoneBook loaded successfully.
BeanInfo:	Class foundation2pro.PhoneBookBeanInfo not found on classpath.
Properties:	Verified 24 PropertyDescriptors.
Event Sets:	Verified 12 Event Sets.
Property Editors:	Verified 1 Property Editors.
Customizer:	No customizer found.

Examination completed.

Figure 4: BeanInsight verifies JavaBeans within seconds.

Wizards

One of the most immediately noticeable differences when upgrading from JBuilder 4 Foundation Edition to JBuilder 4 Professional Edition is the increased number of Wizards in the Object Gallery. Developers need to be productive and that's where Wizards can boost developer output and code reliability tremendously. JBuilder 4 Professional Edition adds another ten extra wizards to those available in the JBuilder 4 Foundation Edition.

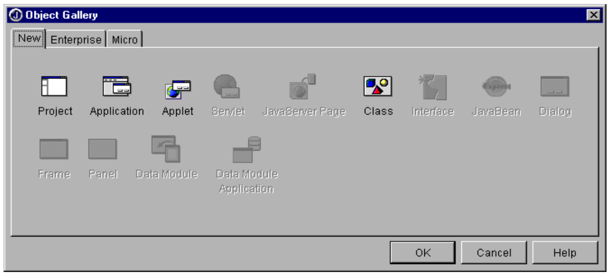


Figure 5: JBuilder 4 Foundation Object Gallery.

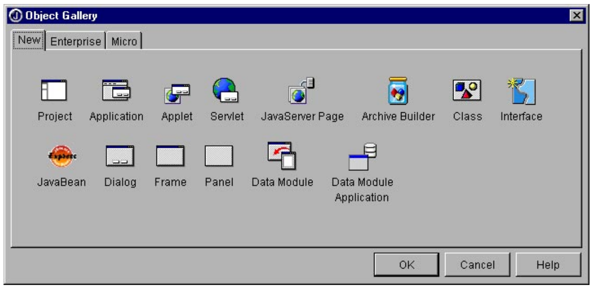
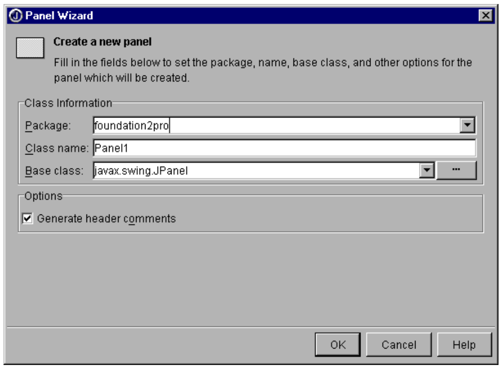


Figure 6: JBuilder 4 Professional Edition Object Gallery.

Some of these Wizards, such as the Wizards for JavaBeans, Servlets, and JSP are covered in other sections of this document, so please refer to the corresponding sections for summaries of those.

Dialog, Frame, and Panel Wizards

Developers not only need to be productive, they also need to be prevented from performing repeated and mundane coding tasks. The requirement for Dialogs, Frames, and Panels in Java applications is frequent and JBuilder 4 Professional Edition provides the Dialog, Frame, and Panel Wizards as shown in Figure 6. These wizards help to ensure that Java application developers can focus on providing the solution for the business and not becoming distracted on fine-grain coding matters.



Create a new panel

Fill in the fields below to set the package, name, base class, and other options for the panel which will be created.

Class Information

Package: foundation2pro

Class name: Panel1

Base class: javax.swing.JPanel

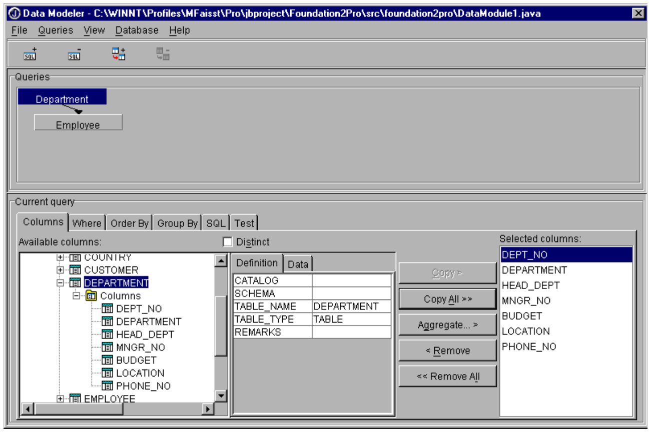
Options

☑ Generate header comments

Figure 7: The Panel Wizard as provided in JBuilder 4 Professional Edition.

DataModule™ Wizard

The DataModule™ Wizard provides the developer with a two-stage process to create DataModules. DataModules simplify your code by providing a centralized container for all the data access components, and help to separate business rules from data access layer logic. The developer can choose to have the DataModeler™ invoked as a second stage to using DataModule Wizard. As can be seen in Figure 8, the DataModeler provides a visual way of creating database views and even master/detail relationships.



Available columns:

- COUNTRY
- CUSTOMER
- DEPARTMENT
  - Columns
    - DEPT_NO
    - DEPARTMENT
    - HEAD_DEPT
    - MNGR_NO
    - BUDGET
    - LOCATION
    - PHONE_NO
- EMPLOYEE

□ Distinct

CATALOG	
SCHEMA	
TABLE_NAME	DEPARTMENT
TABLE_TYPE	TABLE
REMARKS	

Selected columns:

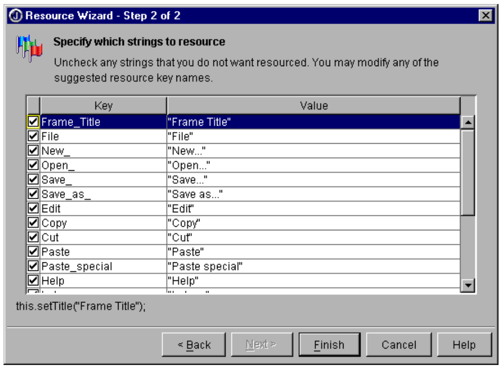
DEPT_NO
DEPARTMENT
HEAD_DEPT
MNGR_NO
BUDGET
LOCATION
PHONE_NO

Figure 8: The DataModule Wizard can be used to start the Data Modeler, shown above, for visual data view creation.

Once created, a DataModule can be used by visual components, such as panels and frames, without the need to re-create data access components. All visual components share a common view of the same data. DataModules can be shared among applications and developers.

Resource String™ Wizard

The Resource String Wizard helps speed the development of international products by making applications and applets easier to translate into other languages. The Resource String Wizard collects hard-coded strings in source code files, from buttons, labels, and so on, in an application or applet and stores them in a specialized file called a ResourceBundle.™ Strings that are taken from the code are automatically replaced with references. The ResourceBundle file can then be deployed with the class files for the application, and the value of the strings can be translated to other languages to internationalize the application without having to modify any of the source code. Figure 9 shows how each string is given a “key” field entry to match the displayed value of the actual string.



Specify which strings to resource

Uncheck any strings that you do not want resourced. You may modify any of the suggested resource key names.

	Key	Value
☑	Frame_Title	"Frame Title"
☑	File	"File"
☑	New_	"New..."
☑	Open_	"Open..."
☑	Save_	"Save..."
☑	Save_as_	"Save as..."
☑	Edit	"Edit"
☑	Copy	"Copy"
☑	Cut	"Cut"
☑	Paste	"Paste"
☑	Paste_special	"Paste special"
☑	Help	"Help"

```
this.setTitle("Frame Title");
```

Figure 9: The Resource String Wizard makes light work of software internationalization.

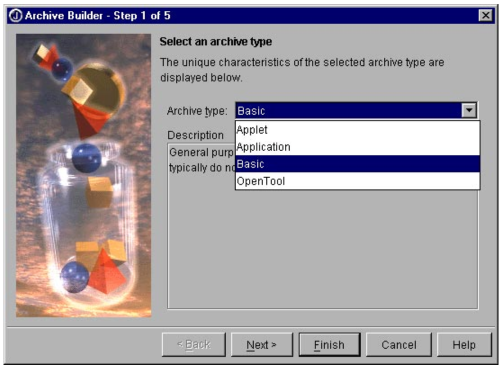
Archive Builder™

All but the simplest of Java applications will consist of more than one Java class file; and deployment of applications involve packaging class, image resource, and other types of files to produce a deployable product. The Archive Builder™ in JBuilder Professional guides the developer through the process of producing JAR or ZIP files in the manner most appropriate for the job at hand.

As can be seen in Figure 10, the Archive Builder offers four archive types for selection: Applet, Application, Basic, and Open Tool.

To illustrate the need for different archive types, we will take a look at how two of these archive types might differ. An Applet archive type would use compression by default, and would also include the necessary library files to support the main applet, whereas an Application archive does not compress by default and does not include dependent libraries as part of the archive. To save the developer from needless keystrokes, a selection can be made so that JBuilder will automatically create the archive file on each recompilation of the project.

The Archive Builder will also guide the developer through the most appropriate use of the manifest file that is included in Java archive files as a means of describing the archive contents.



Select an archive type

The unique characteristics of the selected archive type are displayed below.

Archive type: Basic

Description

General purp
typically do no

Figure 10: The Archive Builder speeds Java application deployment.

Database

One of the main requirements of systems written for use within commercial environments is to read from and write to company databases. This section looks at how Borland JBuilder 4 Professional provides database and database accessing technologies that enable the delivery and collection of data for all aspects of a business.

DataExpress™

Using the JBuilder DataExpress™ components, you can rapidly build all-Java client-server applications, applets, and Servlets for the Internet or intranet. The modular architecture of DataExpress offers many benefits, including support for:

- Network computing
- Mobile computing
- Embedded applications
- Rapid development of user interfaces

Figure 11: The DataExpress Tab enables applications to be easily linked to corporate data resources.

Borland JBuilder 4 Professional Edition includes more than 30 dbSwing™ type JavaBeans available on the component palette. dbSwing components are swing components, provided by Borland, which have the capability of being made data aware through accessing DataExpress datasets. The dbSwing components enable developers to rapidly create Java 2 database applications using these data-aware JavaBeans. The visual dbSwing components are shown in Figure 12.

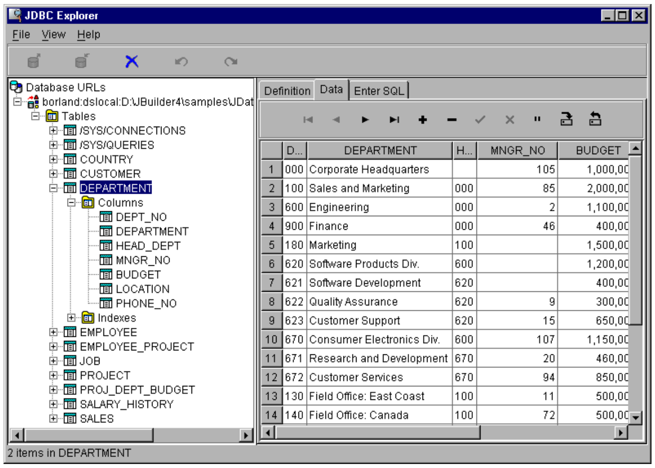
Figure 12: The dbSwing Tab provides rapid development of database-accessing user interface.

SQL Tools

The majority of information held on commercial computer systems is stored using relational databases, with Structured Query Language (SQL) being the language used to access and manipulate the data that is stored. JBuilder 4 Professional provides a comprehensive set of tools for developers to visually create, change, and edit SQL queries as well as view both data and table structure. All JBuilder and Java application data access is via Java Database Connectivity (JDBC®), the primary API for data access in Java applications.

JBuilder Explorer for JDBC

The Explorer for JDBC provided with JBuilder Professional enables developers to create connection URLs to new databases as well as view database contents and schema.



```
Database URLs
  borland:dslocal:D:\JBuilder4\samples\JDat
    Tables
      /SYS/CONNECTIONS
      /SYS/QUERIES
      COUNTRY
      CUSTOMER
      DEPARTMENT
        Columns
          DEPT_NO
          DEPARTMENT
          HEAD_DEPT
          MNGR_NO
          BUDGET
          LOCATION
          PHONE_NO
        Indexes
      EMPLOYEE
      EMPLOYEE_PROJECT
      JOB
      PROJECT
      PROJ_DEPT_BUDGET
      SALARY_HISTORY
      SALES
```

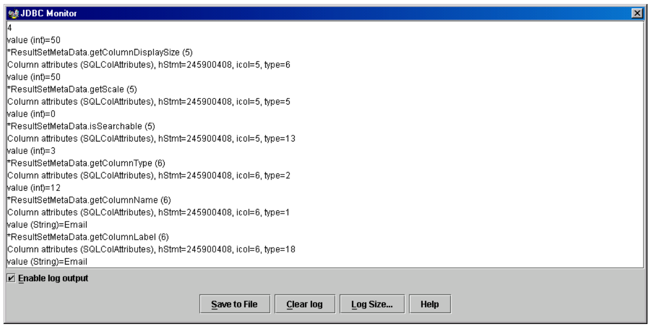
	D...	DEPARTMENT	H...	MNGR_NO	BUDGET
1	000	Corporate Headquarters		105	1,000,00
2	100	Sales and Marketing	000	85	2,000,00
3	600	Engineering	000	2	1,100,00
4	900	Finance	000	46	400,00
5	180	Marketing	100		1,500,00
6	620	Software Products Div.	600		1,200,00
7	621	Software Development	620		400,00
8	622	Quality Assurance	620	9	300,00
9	623	Customer Support	620	15	650,00
10	670	Consumer Electronics Div.	600	107	1,150,00
11	671	Research and Development	670	20	460,00
12	672	Customer Services	670	94	850,00
13	130	Field Office: East Coast	100	11	500,00
14	140	Field Office: Canada	100	72	500,00

2 items in DEPARTMENT

Figure 13: JBuilder Explorer for Pure Java JDBC for viewing schema and creating connection URLs.

JDBC Monitor

JBuilder 4 Professional Edition provides a user interface to enable developers to monitor any JDBC driver while in use within JBuilder. This can be particularly useful for ensuring efficient interaction between database servers and client applications.



```
4
value (int)=50
*ResultSetMetaData.getColumnDisplaySize (5)
Column attributes (SQLColAttributes), hStmt=245900408, icol=5, type=6
value (int)=50
*ResultSetMetaData.getScale (5)
Column attributes (SQLColAttributes), hStmt=245900408, icol=5, type=5
value (int)=0
*ResultSetMetaData.isSearchable (5)
Column attributes (SQLColAttributes), hStmt=245900408, icol=5, type=13
value (int)=3
*ResultSetMetaData.getColumnType (6)
Column attributes (SQLColAttributes), hStmt=245900408, icol=6, type=2
value (int)=12
*ResultSetMetaData.getColumnName (6)
Column attributes (SQLColAttributes), hStmt=245900408, icol=6, type=1
value (String)=Email
*ResultSetMetaData.getColumnLabel (6)
Column attributes (SQLColAttributes), hStmt=245900408, icol=6, type=18
value (String)=Email
```

☑ Enable log output

Figure 14: JDBC Monitor enables the developer to track all interactions between a data client and a database that is being accessed via JDBC.

Comprehensive database capability is imperative for a development tool that is being used in all but the most trivial of applications. JBuilder 4 Professional Edition builds on the JBuilder heritage for providing commercial developers with the ability to access corporate data resources via JDBC, the principal Java database accessing API, and to access and manipulate that data using industry-standard SQL.

JDataStore™

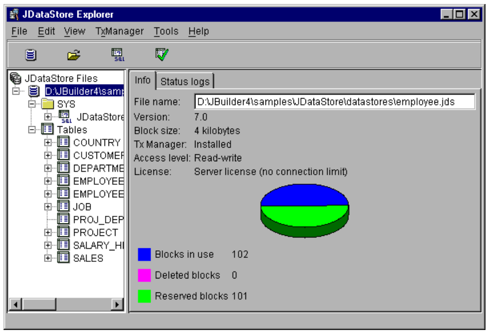
Recognizing the need for storing information in many formats, including objects, graphics, and tables of data, Borland developed JDataStore as a high-performance, small-footprint, 100% Pure Java™ multifaceted data storage solution that is supplied with JBuilder 4 Professional Edition. JDataStore can be described as the following:

- A zero-administration embedded relational database with both JDBC and DataExpress interfaces that supports transactional multi-user access with crash recovery.
- An object store for storing serialized objects, tables, and other file streams.
- A JavaBeans component that can be manipulated with visual bean builder tools like JBuilder.

JDataStore can be used to serialize all of an application's objects and file streams into a single physical file; this can be both convenient and very portable. Should you need to store tables of information or have the requirement to cache query answer-sets, JDataStore offers SQL-92 compliance for interacting with information extracted from an RDBMS.

Using DataExpress datasets, JDataStore can enable mobile and off-line applications. JDataStore asynchronously replicates and caches data from a data source, allows access and updates, and resolves changes back into the data source. The data might come from a database server, a CORBA® application server, SAP, BAAN, or other data source.

As part of the high level of database tool support in JBuilder 4, the Professional Edition provides tools for interacting with JDataStore. JDataStore Explorer provides a user interface for developers to view and manage DataStores. The JDataStore Explorer enables the developer to view many different types of data that have been stored, including text and graphic information. Graphical formats that have viewer support are JPG and GIF.



File name: D:\JBuilder4\samples\JDataStore\datastores\employee.jds
Version: 7.0
Block size: 4 kilobytes
Tx Manager: Installed
Access level: Read-write
License: Server license (no connection limit)

🟦 Blocks in use 102
🟪 Deleted blocks 0
🟩 Reserved blocks 101

Figure 15: JDataStore Explorer enables developers to view and manage DataStores.

Debugging

When developing applications to a deadline, any problem that remains unsolved for more than a short period of time will not only cause great developer frustration, but can soon result in compounded product scheduling problems, release delays, and ultimately loss in revenue. When unexpected behavior or other problems occur, the software developers rely on their development tool to provide an environment where they are able to rapidly and reliably recreate, locate, and resolve problems. There is also the requirement to prove that the work performed has resolved the issue at hand.

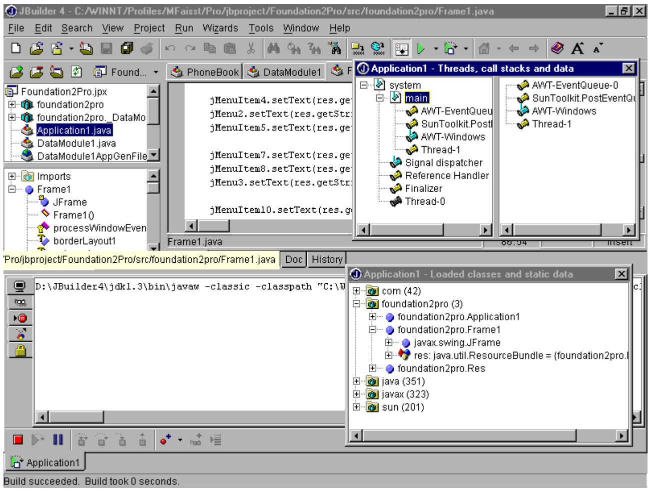
JBuilder 4 Foundation Edition does provide a debugger, with line stepping, conditional breakpoints, and other debugging capabilities. These debugging features are built upon and extended in JBuilder 4 Professional Edition to speed the software development cycle, so progress is uninterrupted and software release is not delayed. JBuilder 4 Professional developers benefit from the inclusion of features such as:

> Multiple JDK version debugging ensures that the same installation of JBuilder can be used to debug products written using different JDK versions. In fact, JBuilder 4 Professional can debug any JDK with Java JPDA debug API support. This greatly extends the flexibility of your development tool investment and developer productivity-you are able to build and maintain all your Java products using a single JBuilder version.

Breakpoint actions provide for programmable control on break events. Using this facility, the developer can select one or more actions to be performed when a breakpoint occurs. The debugger can either stop the program and display a message, log a message in the console, or evaluate an expression and log the results.

Java projects will often require more than one application per project. For this reason, JBuilder 4 Professional allows you to debug multiple processes in multiple debugging sessions. Processes can be in the same JBuilder project, or in different ones. This allows for debugging of both a client and a server process at the same time, in the same JBuilder instance.

Debugger views are displayed at the bottom of the AppBrowser™ in the message pane. Views are selected via tabs on the left side of the debugger. JBuilder Professional Edition allows the developer to configure debugger views as floating windows. Floating windows allow you to see multiple debugger views at the same time, rather than having to switch back and forth between them.



JBuilder 4 - C:/WINNT/Profiles/MFaisst/Pro/jbproject/Foundation2Pro/src/foundation2pro/Frame1.java

```
jMenuItem4.setText(res.ge
jMenu2.setText(res.getStr
jMenuItem5.setText(res.ge

jMenuItem7.setText(res.ge
jMenuItem8.setText(res.ge
jMenu3.setText(res.getStr

jMenuItem10.setText(res.g
```

Application1 - Threads, call stacks and data

```
system
  main
    AWT-EventQueu
    SunToolkit.Postl
    AWT-Windows
    Thread-1
  Signal dispatcher
  Reference Handler
  Finalizer
  Thread-0
```

```
AWT-EventQueue-0
SunToolkit.PostEventQu
AWT-Windows
Thread-1
```

Application1 - Loaded classes and static data

```
com (42)
foundation2pro (3)
  foundation2pro.Application1
  foundation2pro.Frame1
    javax.swing.JFrame
    res: java.util.ResourceBundle = {foundation2pro.I
  foundation2pro.Res
java (351)
javax (323)
sun (201)
```

```
D:\JBuilder4\jdk1.3\bin\javaw -classic -classpath "C:\W
```

Build succeeded. Build took 0 seconds.

Figure 16: JBuilder 4 Professional Edition offers state-of-the-art debugging for commercial developers.

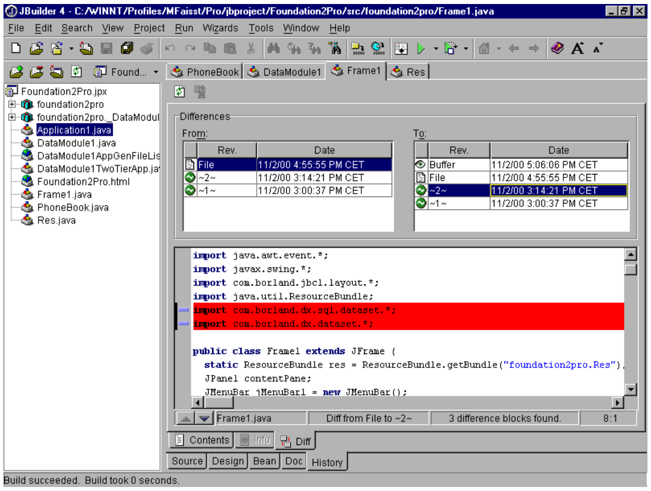
Although basic debugging facilities are provided in JBuilder 4 Foundation Edition, developers need extensive debugging capabilities from a development tool that they are going to spend any significant amount of time using and wherever there is a software delivery deadline to be met. JBuilder 4 Professional Edition supports developers to produce Java applications on time and in budget.

Source File Versioning & Team Development

JBuilder 4 has brought with it new levels of source code control and management. JBuilder 4 Foundation Edition offers history and revision capabilities, but for those that require source code revision comparison capabilities or source code versioning, JBuilder 4 Professional Edition offers the solution.

First, JBuilder 4 Professional provides visual source-level display of differences. The developer can access this by selecting the “History” tab (as shown in Figure 17) for any file residing within the project. Displaying of differences is not only for the comparison of previously saved file versions; comparisons can also be performed on the unsaved file version that JBuilder keeps buffered in memory while the developer is coding. Should you find that you want to revert to a previously stored version, you can do this using the revert button.

JBuilder 4 allows the project to be configured so that backup files are not placed into the same folder as current working copies. This ensures that it’s still easy to share your source.



JBuilder 4 - C:/WINNT/Profiles/MFaisst/Pro/jbproject/Foundation2Pro/src/foundation2pro/Frame1.java

Differences

From:

Rev.	Date
File	11/2/00 4:55:55 PM CET
~2~	11/2/00 3:14:21 PM CET
~1~	11/2/00 3:00:37 PM CET

To:

Rev.	Date
Buffer	11/2/00 5:06:06 PM CET
File	11/2/00 4:55:55 PM CET
~2~	11/2/00 3:14:21 PM CET
~1~	11/2/00 3:00:37 PM CET

```
import java.awt.event.*;
import javax.swing.*;
import com.borland.jbcl.layout.*;
import java.util.ResourceBundle;
import com.borland.dx.sql.dataset.*;
import com.borland.dx.dataset.*;

public class Frame1 extends JFrame {
  static ResourceBundle res = ResourceBundle.getBundle("foundation2pro.Res").
  JPanel contentPane;
  JMenuBar jMenuBar1 = new JMenuBar();
```

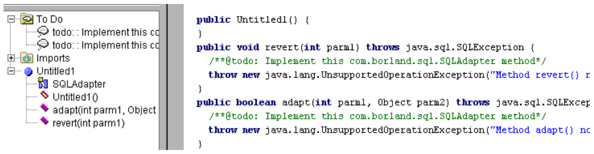
Frame1.java | Diff from File to ~2~ | 3 difference blocks found. | 8:1

Build succeeded. Build took 0 seconds.

Figure 17: JBuilder 4 enables you to track the changes that are made to your code.

To-do List Manager and To-do Source Comment Tags

A neat example of the Borland attention to detail is the addition of the To-do list feature in JBuilder 4 Professional Edition. If you are fed up with leaving sticky notes on the monitors of your colleagues, and even for yourself, then you are going to enjoy using this feature. To-do items can now be shared within a JBuilder project amongst a whole development team. This enables both development and management staff to keep track of the exact status of work. By adding To-do item comments into your Java code, JBuilder will recognize these and then add children to a To-Do node in the structure pane of the AppBrowser. Figure 18 shows how this looks within the IDE.



```
public Untitled1() {
}
public void revert(int parm1) throws java.sql.SQLException {
  /**@todo: Implement this com.borland.sql.SQLAdapter method*/
  throw new java.lang.UnsupportedOperationException("Method revert() r
}
public boolean adapt(int parm1, Object parm2) throws java.sql.SQLExcep
  /**@todo: Implement this com.borland.sql.SQLAdapter method*/
  throw new java.lang.UnsupportedOperationException("Method adapt() nc
}
```

Figure 18: To-do list manager and source comment tags enable project staff to stay posted.

Web Development

Providing the right information to the right person regardless of geography is something that is key to the IT strategy of most organizations. As Java Web technologies have matured, Borland has added these features to JBuilder 4 Professional Edition to the point where JBuilder Professional now offers an admirable arsenal of Internet development capabilities.

With most Java Web development being centered around Java Server Pages and Servlets, JBuilder 4 Professional Edition offers more than just source code creation wizards; it comes with a JSP/Servlet-enabled web server so developers can test their code without leaving the JBuilder IDE.

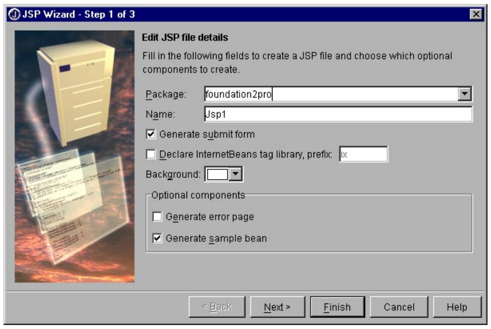
Using JBuilder 4 Professional Edition, developers can rapidly build and debug dynamic Web-based applications using Servlets and JSPs. Wizards provide visual tools to create thin-client front ends to your Pure Java applications that can be accessed by any Web browser.

Web development is a real focus of JBuilder 4 Professional Edition with Internet technologies such as:

- Express for JSP for rapid development of dynamic Java Server Page thin-client services.
- Express for Servlet to speed dynamic servlet thin-client services development.
- InternetBeans™ Express presentation components that enable developers to provide Web-driven Internet applications.
- Web Debug and Web Run for testing your applications within the AppBrowser.

Express for JSP

Java Server Pages (JSP) offer the developer a good mix between design and integration into business systems by using XML-like tags and code written in Java to provide dynamic content in Web pages.



Edit JSP file details

Fill in the following fields to create a JSP file and choose which optional components to create.

Package: foundation2pro

Name: Jsp1

☑ Generate submit form

□ Declare InternetBeans tag library, prefix: ix

Background:

Optional components

□ Generate error page

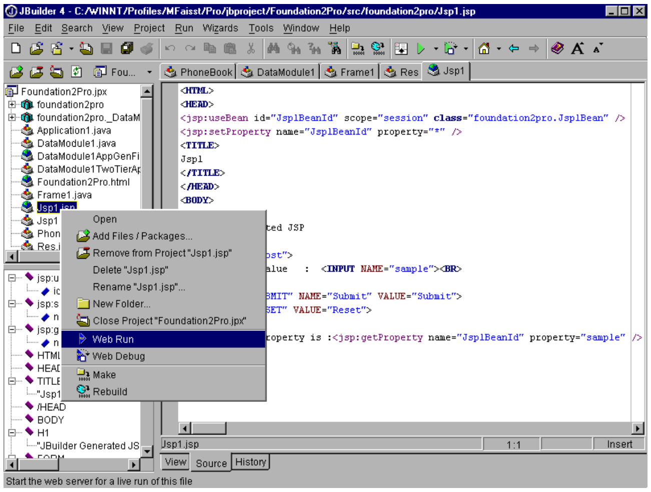
☑ Generate sample bean

Figure 19: The JSP Wizard steps you through the process of creating a Java Server Page.

JBuilder 4 Professional Edition recognizes JSP files and appropriately highlights the syntax of these accordingly. The easiest way to start creating a JSP is by using the Wizard for JSP.

The Wizard for JSP takes the developer through three simple configuration screens so that the name, background color, and other settings can be chosen before creating the JSP. The developer is then free to extend and adapt the JSP accordingly.

Once the JSP has been created, it will need testing and, to do this, the developer will need to see it running. This is made very easy in JBuilder 4 Professional Edition, as all the developer need do is to right-click on the JSP in question and, as shown in Figure 20, select either the Web Run or Web Debug options.

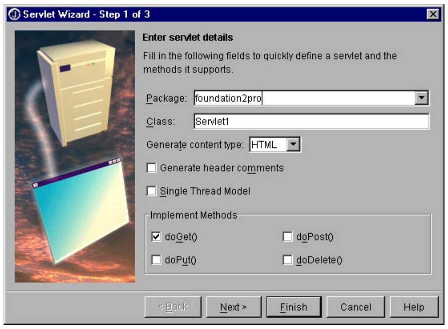


```
<HTML>
<HEAD>
<jsp:useBean id="Jsp1BeanId" scope="session" class="foundation2pro.Jsp1Bean" />
<jsp:setProperty name="Jsp1BeanId" property="*" />
<TITLE>
Jsp1
</TITLE>
</HEAD>
<BODY>

ted JSP

ost">
alue   :  <INPUT NAME="sample"><BR>

BMIT" NAME="Submit" VALUE="Submit">
SET" VALUE="Reset">

roperty is :<jsp:getProperty name="Jsp1BeanId" property="sample" />
```

Figure 20: Running a JSP or servlet in JBuilder 4 Professional Edition is as easy as right-clicking.

Express for Servlet

Servlets are the Java platform technology of choice for extending the functionality of Web servers. In fact, JSPs are compiled down to servlets before they can be executed. So, servlets are very important for providing a common, platform-independent method for building Web-based applications.



Servlet Wizard - Step 1 of 3

Enter servlet details

Fill in the following fields to quickly define a servlet and the methods it supports.

Package: foundation2pro

Class: Servlet1

Generate content type: HTML

☐ Generate header comments

☐ Single Thread Model

Implement Methods

☑ doGet() ☐ doPost()

☐ doPut() ☐ doDelete()

Figure 21: The Wizard for Servlets steps you through the process of creating a Java servlet.

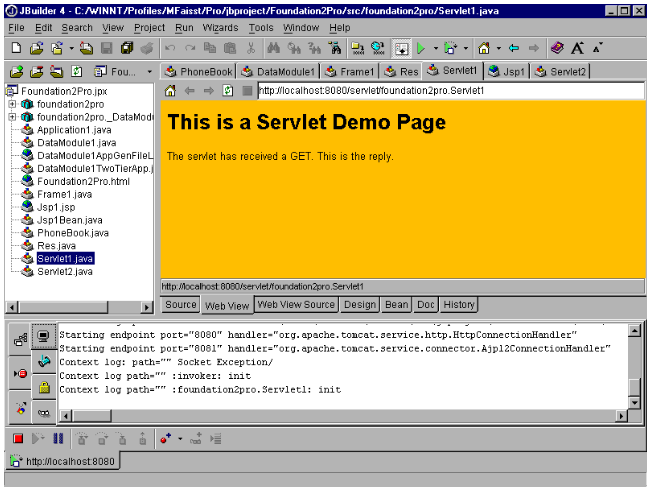
Figure 21 shows the JBuilder 4 Professional Edition Wizard for Servlets. The Wizard provides a fast route to getting a servlet up and running very quickly.

You may have noticed that on the Wizard for Servlets shown in Figure 21 there is a "Generate content type" selection. Using this, the content type that this servlet provides can be selected. Options here are HTML, XML, XHTML, and WML. JBuilder 4 Professional Edition is very much geared to supporting the development of e-business applications through the support of XML as well as supporting information access on the move via the inclusion of WML.

Again, as can be seen in Figure 21, an important point to appreciate about the JBuilder 4 Wizard for Servlets is that it will create all of the HTTP implementation methods that are needed by simply selecting check boxes.

The JBuilder 4 Professional Edition support for servlet development does not stop once the servlet has been created. If an application requires access to dynamic content from a database, JBuilder 4 Professional Edition can help there too. That's where the following section on InternetBeans™ Express comes into play.

As with JSPs, JBuilder 4 Professional Edition will enable the developer to test and debug servlets by simply right-clicking on the servlet within the IDE and selecting either Web Run or Web Debug. If Web Debug is selected and breakpoints have been set within the code, JBuilder will halt at those breakpoints where appropriate. Figure 22 shows a servlet that is running within the JBuilder IDE.


JBuilder 4 - C:/WINNT/Profiles/MFaisst/Pro/jbproject/Foundation2Pro/src/foundation2pro/Servlet1.java
File Edit Search View Project Run Wizards Tools Window Help
Fou...
PhoneBook DataModule1 Frame1 Res Servlet1 Jsp1 Servlet2
Foundation2Pro.jpx
foundation2pro
foundation2pro._DataModu
Application1.java
DataModule1.java
DataModule1AppGenFileL
DataModule1TwoTierApp.j
Foundation2Pro.html
Frame1.java
Jsp1.jsp
Jsp1Bean.java
PhoneBook.java
Res.java
Servlet1.java
Servlet2.java
http://localhost:8080/servlet/foundation2pro.Servlet1
http://localhost:8080/servlet/foundation2pro.Servlet1
Source Web View Web View Source Design Bean Doc History
http://localhost:8080

This is a Servlet Demo Page

The servlet has received a GET. This is the reply.

```
Starting endpoint port="8080" handler="org.apache.tomcat.service.http.HttpConnectionHandler"
Starting endpoint port="8081" handler="org.apache.tomcat.service.connector.Ajp12ConnectionHandler"
Context log: path="" Socket Exception/
Context log path="" :invoker: init
Context log path="" :foundation2pro.Servlet1: init
```

Figure 22: JBuilder 4 Professional Edition hosts your Web application, enabling you to test and debug from within the JBuilder IDE.

InternetBeans™ Express

JBuilder 4 Professional Edition provides a set of 15+ components that enables dynamic content generation in web pages through both Servlets and JSPs. In addition to generic content, InternetBeans Express uses DataExpress technology to make it easy to display, navigate, update, and append data sets.

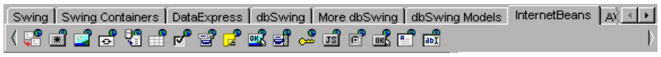


Figure 23: InternetBeans Express components for dynamic Web content generation.

Figure 23 shows the InternetBeans components; these can be viewed as:

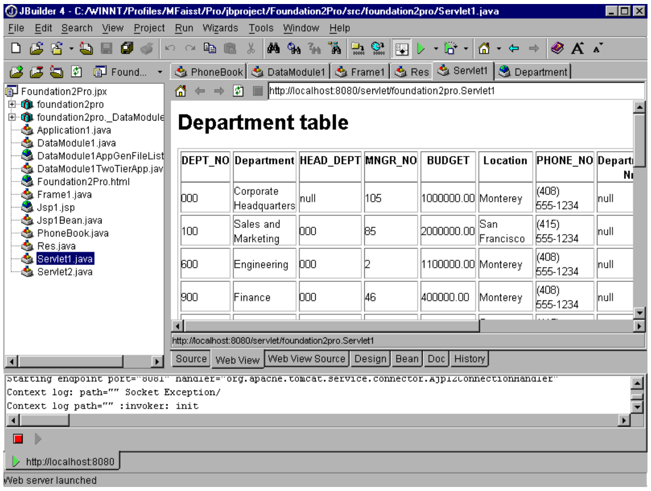
XControl — A generic control that determines at runtime which type of HTML control it is replacing and emulates that control.

IxPageProducer — Reads/parses a static HTML file and inserts dynamic data as text or controls.

xTable — A generic control that determines at runtime which type of HTML control it is replacing and emulates that control.

Other controls — Controls such as ixTextField and ixCheckBox correspond to HTML controls that get their data from an associated dataset.

Most servlets that use InternetBeans Express Components will have their dynamic content based around an ixPageProducer component. As in other Borland development tools such as Delphi™ and C++Builder,™ the Producer component can generate responses from a predesigned Web page and uses dynamic content tags associated with drag-drop controls to provide specific HTML content or perform HTTP tasks. Once created, the data-accessing servlet can be run or debugged from within the JBuilder environment. These components are dropped onto a servlet when placed in Design View within the JBuilder IDE.



http://localhost:8080/servlet/foundation2pro.Servlet1

Department table

DEPT_NO	Department	HEAD_DEPT	MNGR_NO	BUDGET	Location	PHONE_NO	Departi Ni
000	Corporate Headquarters	null	105	1000000.00	Monterey	(408) 555-1234	null
100	Sales and Marketing	000	85	2000000.00	San Francisco	(415) 555-1234	null
600	Engineering	000	2	1100000.00	Monterey	(408) 555-1234	null
900	Finance	000	46	400000.00	Monterey	(408) 555-1234	null

http://localhost:8080/servlet/foundation2pro.Servlet1

```
Starting endpoint port="8081" handler="org.apache.tomcat.service.connector.Ajp12ConnectionHandler"
Context log: path="" Socket Exception/
Context log path="" :invoker: init
```

http://localhost:8080

Web server launched

Figure 24: Producing data accessing thin-client web applications is easy with JBuilder 4 Professional Edition.

InternetBeans Express is a set of components and JSP tag extensions for generating and responding to the presentation layer of a web application. It takes static (template) pages, inserts dynamic content from a live data model, and presents them to the client; then it writes any changes that are posted from the client back into the data model.

Summary

JBuilder offers a unique opportunity for developers to provide solutions using Java technologies. Its award-winning technologies are respected around the globe and by professionals on all three of the platforms for which JBuilder provides unique development support: Solaris,™ Windows,® and Linux.®

Businesses that invest in the right Java development tools for the job will be able to get to market sooner and gain competitive advantage by producing reliable and flexible software systems. JBuilder 4 Professional Edition offers state-of-the-art Java development for those that are serious about producing Java applications. With its enhanced IDE, numerous wizards, advanced JavaBean support, excellent database development capabilities, advanced Web features, and extensive debugging facilities, JBuilder 4 Professional Edition is equipped to take on your Java development challenges and win.

The Borland JBuilder 4 product family does not end here. JBuilder 4 Enterprise Edition takes Java development into even higher realms with Java 2 Enterprise Edition (J2EE) for creating large-scale, end-to-end e-commerce solutions. As well as extending features, such as the team development and debugging support, JBuilder 4 Enterprise Edition comes with VisiBroker® 4.1, world-leading CORBA ORB technology from Borland and Borland® Application Server™ 4.1, offering full support for J2EE and EJB™ 1.1. Borland JBuilder 4 Enterprise Edition provides extensive development, debugging, and deployment capabilities for Enterprise JavaBeans™ (EJBs), and integrated support is provided for leading application servers such as Borland Application Server 4.1 and WebLogic® Server 5.1.

If you have been using JBuilder 4 Foundation Edition for your Java development, isn't it time that you took a look at how the other members of the Borland JBuilder family can help you to provide the solutions that your customers deserve?

Borland®

100 Enterprise Way
Scotts Valley, CA 95066-3249
www.borland.com | 831-431-1000 | Fax: 831-431-4113