Jato Pre-Release Trial Version Read Me First Information

Welcome to the new Powersoft development tool for Java! For now, this product is code-named "Jato" (the previous version was code-named "Starbuck". This is a special pre-release trial version of Jato that is made available through the Jato Early Access Program.

These topics introduce you to the features of Jato and provide the latest information to help you quickly explore the features that are available in this pre-release trial version.

If you have used the earlier version that went by the former name of "Starbuck" and want to see **what is new for Jato**, see New in this build.

This is a pre-release version of Jato. Some of the functionality planned for the product code-named "Jato" is not available or does not function properly in this version. We are providing this software on an "as is" basis so that we can get user feedback early in the product development cycle, and to give you early access to the functionality that has been implemented.

Note: This software is provided for evaluation purposes only. You may not use it to produce retail applications. Consult the license agreement for details.

All company and product names used herein may be the trademarks or registered trademarks of their respective companies.

Important information

The Jato component library reference is not installed by the Jato installation. The reference files are in the pj`clref.zip` file that is included in the download, but you must unzip the file yourself. The reference is in HTML format rather than help format. You will probably want to create a shortcut to the `packages.html` file so that you can easily access the reference.

Two important things to know about Jato are:

- How to choose the Java Virtual Machine (VM) that Jato will use to run your Java programs. For instance the Microsoft VM supports debugging but cannot be used with packages, like the JDBC-ODBC Bridge driver, which use the Sun native call interface. To choose the VM, open the Targets window, click **Run Options** on the **File** menu.
- How to set the ClassPath for a target. This is described in Setting the ClassPath.

Other resources

To get the most out of your trial use of Jato, you may want to download other files or read other information on the Web. Here are some suggested resources:

- At the time of writing, the only browser that supports debugging of Java applets is Microsoft Internet Explorer 3.01 (at the time of writing, 3.01 is the most current version). You can download it from:

  `http://www.microsoft.com/ie/download/`

- This release of Jato does not include Sybase SQL Anywhere or NetImpact Dynamo. You can download a free evaluation version of Sybase SQL Anywhere 5.5 Professional, including NetImpact Dynamo, from:

  `http://www.powersoft.com/download/sqlanyform.html`

  Note: The evaluation version of Sybase SQL Anywhere does not include the Open Server Gateway which you will need to use the jdbcCONNECT driver included with Jato. With other versions of Sybase SQL Anywhere, you can install the Open Server Gateway with the server installation option.

- A white paper and other information on NetImpact Dynamo is available at the Sybase Web site:

  `http://www.sybase.com/products/system11/workplace/dynamo.html`

- This release of Jato installs a beta of the Sybase jdbcCONNECT driver. You can use for direct access to the complete family of Sybase products including SQL Server, SQL Anywhere (using the

Open Server Gateway), Sybase IQ and Replication Server. It also access other databases through Sybase OmniCONNECT, and Sybase DirectCONNECT. For more information on jdbcCONNECT, including documentation, see:

`http://www.sybase.com/products/internet/jdbcconnect/`

- If you downloaded and installed the trial version of SQL Anywhere, or have another ODBC database, you can use JDBC-ODBC Bridge driver from Sun and INTERSOLV. With this build you should use the 1.2001 version (or newer) of the driver (see Important JDBC 1.21 information for instructions). It can be downloaded from:

  `http://splash.javasoft.com/jdbc/`

- The Java language specification, and a Java tutorial, are available at the Sun Java site:

  `http://java.sun.com/nav/download/index.html`

- The JavaBeans specification is available at the JavaSoft site:

  `http://splash.javasoft.com/beans/`

■ What is Jato?
■ Jato and Optima++
■ Installing and uninstalling
■ How to use this pre-release version of Jato
■ Known problems
■ Sending feedback

What is Jato?

Jato is a new development tool for Java. It greatly enhances the productivity of business developers building and deploying Java applets and applications. Leveraging Sybase expertise in business application development and the technology of the award-winning Powersoft Optima++, Jato provides a highly productive, component-based development environment with scalable database connectivity and server-side development.

Jato combines concepts from the Optima++ environment with a rich Java tool set to provide a productive development tool for building multi-platform, business applications.

Jato features include:

The On-Ramp To Java

Jato flattens the learning curve so developers can quickly exploit the benefits of this new programming language. Jato leverages the revolutionary Optima++ development environment to provide a fast on-ramp to Java development. The approachable development environment delivers drag-and-drop programming with the powerful reference card and parameter wizards to allow new developers to quickly get started. As developers grow into the full capabilities of Java, Jato has the depth professional developers require with in-context debugging within the browser, an extensive library of components and classes, seven views for project navigation, and a visual SQL editor.

Powerful Component-Centric Development

Jato provides unparalleled support for both JavaBeans and ActiveX components. Third-party JavaBeans and ActiveX components are seamlessly integrated into Jato with the full power of the new component quickly accessible via drag and drop programming. This revolutionary approach allows developers to become instant experts with new components. Jato also includes a sophisticated set of built-in JavaBeans components allowing developers to quickly assemble multi-platform pure Java applets and applications. Jato maximizes developer productivity through re-use, making it easy for developers to create and extend JavaBeans components to enhance their development toolkit.

Data Smart Applications

Jato offers exceptional database support by providing a powerful set of data-aware components that can be used with any JDBC data source. To build and test database applications Jato includes the award-winning Sybase SQL Anywhere database. For high-performance data access Jato includes Sybase jdbcCONNECT™, a full thin-client JDBC implementation. Sybase jdbcCONNECT provides Java developers native database access in a multi-tier Internet environment. Sybase jdbcCONNECT can be downloaded quickly and on the fly without client installation. Sybase jdbcCONNECT offers high-performance, direct access to the complete family of Sybase products including SQL Server, SQL Anywhere, Sybase IQ, and Replication Server. In addition, jdbcCONNECT also provides high performance access to more than twenty-five Enterprise database servers through Sybase OmniCONNECT.

Simplified Server Development

To maximize the savings from lower cost deployment on the web, Jato dramatically simplifies the development and debugging of ultra-thin client applications where the application logic, business rules and processing are distributed to the server. Jato makes it easy for developers to extend the functionality of their web servers using built-in high-level Java components or by extending the NetImpact Dynamo custom application server with Java servlets. NetImpact Dynamo takes care of the session management, thread pooling, caching and HTML templates allowing developers to maximize their development efforts by focusing on the code specific to their application. Remote debugging makes it easy to seamlessly code and test server-side applications.

Total Integrated Business Solution

Jato is the first Java development tool to take an integrated solution approach to web application development. The innovative Web Application Target extends the Jato Java project to provide developers with a simple way to integrate editing, debugging and publishing of all aspects of a web application including HTML and JavaScript. This extensible approach allows developers to test their Java applets and servlets in-context of their whole web application.

Jato and Optima++

The product code-named “Jato” is closely linked to Optima++; they share the same development environment. The main difference is that Jato uses Java for a programming language, and Optima++ uses C++. Since the upcoming Optima++ 2.0 release and Jato leverage the same technology base, the two products may be integrated to seamlessly enable C++ and Java development in a single environment.

You may even notice places in this preview version of Jato where it says “Optima++” instead of “Jato”!

Powersoft Optima++ is available now in Developer, Professional and Enterprise version, and it has earned the following awards:

- **PC Week Analyst’s Choice**, June 10, 1996
- **Software Development Incredible Product**, September 1996
  (This marked the first time that Software Development has given its top rating to a first-release product.)
- **Byte Best**, October 1996
- **Best of LAN Times**, November 1996
- **Computer Reseller News Recommended**, November 1996
- **Windows Tech Journal Star Tech**, December 1996
- **Lan Magazine (Australia) Client Server Development Tool of the Year**, December 1996
- **PC Magazine (UK) Technical Innovation Award**, January 1997
- **The PC Expert 1997 (France) Technical Excellency Award**, January 1997
- **PC Magazine Editors' Choice Award**, February 4, 1996

For more information on Optima++, including a downloadable test-drive version, visit the Optima++ Web site at:

`http://www.powersoft.com/products/internet/optima/optima2.html`

If you prefer, call us at 1-800-395-3525 in North America. Outside North America call 1-508-287-1500 or contact your local Powersoft representative.

Jato Pre-Release Trial Version Read Me First Information

Installing and uninstalling

These topics provide more information on installing and uninstalling Jato.

- Installing and using Jato on Windows NT
- Ensure that you have sufficient disk space
- Co-existing with Optima++
- Uninstalling Jato

Installing and using Jato on Windows NT

Jato requires NT 4.0 or higher (beta versions are not supported).

You must have administrator permissions to install Jato on Windows NT, in order for some registry settings to be made.

Note that you also need administrator privileges under NT in order to register ActiveX controls.

If you run Jato under NT from an account that does not have full access to the registry then Jato will run but you will encounter problems because it is unable to access the registry. The most common problem will be getting these errors when you build a project:

```
Option requires a path
Processing command line switch -rod=
```

◆ **To solve these problems**

1. Log on from an account with administrator permissions.
2. Open the Registry editor.
3. Select the key **HKEY_LOCAL_MACHINE\Software\Powersoft**
4. Choose **Permissions** from the **Security** menu.
5. Grant **Full Control** of that folder to the **Users** group.

This should allow Jato to run from any account on the machine.

Ensure that you have sufficient disk space

Make sure you that have sufficient disk space to install Jato. For this trial version. You should have at least 100 MB free disk space before downloading and installing all files. (More disk space might be needed if your disk is partitioned to have disk allocation units larger than 16 KB.)

In addition, you should have at least 40 MB available for your Windows swap drive, for Jato to operate efficiently.

Co-existing with Optima++

- Do not install this version of Jato to the same location as Optima++, unless it is a build of Optima++ 2.0 that has the same build number as this build of Jato. (The Jato and Optima++ installation programs check for this.)
- This version of Jato can co-exist with Optima++ 1.*x* (1.0, 1.1, 1.5, etc.) as long as Optima++ is in a different folder.
- This pre-release trial version of Jato is only compatible with Optima++ 2.0 that shares the same build number. (At the time of writing, Optima++ 2.0 is in Beta testing.) You will have to uninstall this particular version to install any different build of Optima++ 2.0.

Uninstalling Jato

Follow these steps to remove Jato from your system.

1. In the Powersoft Jato 2.0 program group, double-click the Uninstall icon.
2. During the uninstall process, you may be asked if you want to remove some shared system DLL files. Unless you know that another program needs those files, you should let the uninstall program delete them.

   IMPORTANT: Because Internet Explorer does not increment registry counters for shared files when it installs the Microsoft Java Virtual Machine, you may have to reinstall Internet Explorer after uninstalling Jato. Registry entries are needed by the VM, so it is not sufficient to prevent Jato from removing the files.
3. If you have saved projects under the Jato folder, the uninstall program will not delete them. However it will copy over the sample projects if you reinstall the samples. If you have made changes to the sample projects, save the files in the projects to different locations to avoid losing your changes.
4. You may need to remove some files under your Jato folder to completely remove Jato from your system. The uninstall program can miss some files that are generated by Jato when it is running.

Jato Pre-Release Trial Version Read Me First Information

How to use this pre-release version of Jato

The following topics provide examples of what you can do with this pre-release trial version of Jato. Keep in mind that the functionality of this release is more limited than what you will find in the product when it is released.

If you have not used Jato before, you should follow the tutorial introduced in Building and debugging an applet to build a Java applet and debug it in a browser. As well as showing you the ease and power of component assembly and debugging in Jato, the tutorial demonstrates the acclaimed programming aids that are unique to Jato and Optima++: **drag and drop programming**, the **Reference Card** and the **Parameter Wizard**. Even if you have experience with Optima++, you should at least skim through the tutorial.

The rest of the topics describe particular features of Jato that you can explore with this pre-release trial version.

New in this build

This build of Jato includes the following changes from earlier previews:

- **Design-time support for JavaBeans and other Java components.** You can now add Java components to the component palette, add them to your form at design time, and use the Reference Card and Parameter Wizard for added components. For more information, see Using and creating JavaBeans.
- **New PictureButton and PictureBox components.** These new components are on the Standard page of the components palette. For more information, see Programming standard objects.
- **Improved menu editor.** For more information, see Using and programming menus.
- **Query editor for composing and testing SQL queries.** To use this, you must have ODBC access as well as JDBC access to your database. For more information see Working with databases.
- **JDBC 1.21.** This build uses the latest JDBC standard. Since there are some incompatibility issues, be sure to see Important JDBC 1.21 information.
- **Sending feedback via Powersoft newsgroup.** You can now submit bugs, and discuss Jato with other users, using a Powersoft newsgroup. For more information, see Sending feedback.

Building and debugging an applet

See A first application in the Jato Getting Started for a step by step introduction to the fundamentals of using Jato, including:

- Designing an applet's user interface by drawing components
- Specifying properties for the applet's objects
- Adding code by typing in the code editor or by using drag and drop programming, the Reference Card and Parameter Wizard.
- Running the application
- Debugging the application in a browser or applet viewer

Creating and using JavaBeans or other packages

All of the components in the "powersoft.jcm" package adhere to the JavaBeans standard. In other words, the native components of Jato are JavaBeans. Follow the tutorial in A first application to see how to use the JavaBeans components that are included with Jato.

For information on creating and using other JavaBeans components, see Using and creating JavaBeans.

The JavaBeans specification is available at:

`http://splash.javasoft.com/beans/`

❑ Jato Pre-Release Trial Version Read Me First Information

❑ How to use this pre-release version of Jato

Using ActiveX controls and servers

ActiveX server components allow you to perform non-visual operations such as business logic or database access. Server components can be run on the web server (as a web server extension) or on the client (embedded in a web page).

The sample described here is a Java applet that uses a simple ActiveX server component to multiply two numbers and return the result. Though the application is not interesting in itself, it does demonstrate a number of key features of ActiveX server components. For more information on using ActiveX servers and controls, see Using ActiveX controls and server components.

The steps to using this samples are:

1. Register the ActiveX server component
2. Add the server component to the Java Component Palette.
3. Open the sample project and run the application.

Registering the control or server

Whenever you use an ActiveX server component or ActiveX control, it must first be registered on the system. Registering a component allows applications such as this sample to use the component. While component registration is a system-wide operation, Jato will register ActiveX server components and ActiveX controls for you.

◆ **To register the ActiveX server component:**

1. If Jato is not already running, start it.
2. From the **Components** menu, select **Register ActiveX Controls**. The component registration dialog is displayed and all registered controls are listed.
3. Click the **Register** button. In the file dialog box, change the file type to **Dynamic-link Libraries (*.dll)**.
4. Locate the sample ActiveX server component on your disk. It will be installed in the Samples\Java_ActiveX sub-folder of the Jato folder.
5. Select the server component (TestE.dll), and click **Open**.
6. Close the registration dialog.

Adding the control or server to the palette

To use an ActiveX server component in the design environment, you must add the control to the Component Palette. When you add the component to the palette, Jato automatically adds the component's methods and properties to the Reference Card. This allows you to use drag-and-drop programming with the ActiveX server component as you would with any native component.

◆ **To add the ActiveX server component to the Component Palette:**

1. On the **Components** menu of the main Jato menu bar, click **Add ActiveX Component(s)**. You are presented with the ActiveX Component Wizard.
2. Choose the control that you want to add to the Component palette.
3. Click the TestE item from the Library list.
4. Click **Finish**.

When Jato has finished, the control defined by the type library appears on the Component palette. Once an ActiveX control resides on the Component palette, you can use it in any project you make.

Running the Java applet

You are now prepared to use the Java_ActiveX sample program. From the **File** menu, select **Open**

Project. The sample is located in the Samples\Java_ActiveX sub-folder of the Jato folder. Select the file called `Java_ActiveX.wxp`.

Both Microsoft Internet Explorer 3.01 and Netscape Navigator 3.01 with the ActiveX plug-in offer support for ActiveX controls. Both of these products are available on the Internet.

◆ **To change the environment that is used to run your applet:**

1. From the **View** menu, select **Targets**. Jato will display the Targets View.
2. In the Targets View, select the Java applet target.
3. From the **File** menu, click **Run Options**.
4. On the General option page, select the third option labeled **Use a web browser**.
5. In the web browser field, enter the full path of your web browser, or click **Browse** to select it from disk.
6. Click **OK**.

Once you have configured the web browser, you can press F5 to run your applet. Jato will open your applet's web page using the web browser that you specified.

Once the application is running, you can change the values in the two fields and click the Multiply button. When you click Multiply, the Java applet retrieves the text from the text boxes, converts the text to integers, and invokes the ActiveX server component's Multiply method.

◆ **To view the code that invokes the Multiply method:**

1. Right-click the Multiply button on the design-time form.
2. On the popup menu, point to Events, and select Click.

❑ Jato Pre-Release Trial Version Read Me First Information

❑ How to use this pre-release version of Jato

Managing Web applications

In Jato, you can use *Web application targets* to manage all files related to an application. For example, a Java Applet target contains all of the .java and .class files associated with the applet. Web application targets allow you to manage an entire web application consisting of any number of web pages, Java applets and applications, web server extensions, image and media files, and so on.

When you run a web application, Jato automatically builds all the dependent targets (such as Java applets), and publishes the files to a location that you specify. It then starts a web browser or a web server to allow you to test your entire application.

The rest of this topic provides a brief exploration of using Web targets in Jato. For more information on Web application targets, see Web application targets.

To make a simple web application target, you can start with a Java applet, create a new web application target, add some files to the web application, then run the web application.

◆ **To make a simple web application:**

1. Start Jato. The default target is a Java applet; this applet will be part of your web application.
2. Create a simple Java applet, for example just place a button on the form.
3. Save the project.
4. Open the Targets View by pressing SHIFT+F7. The Targets View displays the default Java Applet.
5. From the **File** menu, point to **New**, and select **Target**. From the list of targets, select Web Application. Click **Next**.
6. Give the target a name. Click **Next**.
7. Specify the targets that are included in the new web application. If you want to include all the targets, click the **Select all** so that it is checked.
8. Click **Finish**.

Jato presents you with a basic HTML file and creates a new web application target. You can use the Jato text editor or any web editor to modify the HTML file. You should add any files that are used by your web application to the target.

◆ **To add files to your web application:**

1. Open the Targets View by pressing SHIFT+F7.
2. Select the web application target.
3. From the **File** menu, select **Add File**.
4. Select the file you want to add from the disk.

Once your application is assembled, you can choose how Jato will publish your files. Publishing web files means copying the files to a particular location so that they can be viewed by a web browser.

◆ **To set publishing and run options:**

1. Open the Targets View by pressing SHIFT+F7.
2. Select the web application target.
3. From the **File** menu, select **Run Options**.

In the Run Options dialog, you can specify whether your files will be published and the location of the published files. Jato allows you to start a web browser or a web server when the web application is run.

◆ **To configure the web browser**

1. In the Run Options dialog, select **Publish, then run a web browser** on the General page.

2. Click **Configure** to specify which web browser will be used.
3. Select Internet Explorer, Netscape Navigator, or specify the location of another browser.
4. Specify the initial URL for the web browser. For a local file, enter the file's location.

Note: If you want to debug your application in a web browser, you must use Internet Explorer 3.01. (Internet Explorer 3.0 includes an older version of the Java VM than that included with Internet Explorer 3.01 and this build of Jato.)

For more information, see Web application targets.

Team programming

Jato supports the most popular source control systems, and can be easily extended to work with others. Without leaving Jato, you can create a new source control project and exclusively check files in and out for editing. This prevents two people from editing the same file at the same time and losing changes. Most source control systems also provide project management and revision tracking. With graphical systems such as Powersoft ObjectCyle and Microsoft SourceSafe, you can see easily see who is working on what files, and check what changes have been made, when they were made and who made them.

For more information on using revision control with Jato, see Source code control in Jato.

Database applications

The Java component library of Jato uses JDBC to provide database accessibility. The low-level details of JDBC are hidden so that you don't need to learn them. You can simply use bound controls and other components for accessing databases.

Don't overlook the information in Important JDBC 1.21 information.

For an overview of JDBC, see:

`http://splash.javasoft.com/jdbc/`

- Using drivers that require native DLLs
- Setting the ClassPath
- Debugging in Jato
- DataSource URL
- Important JDBC 1.21 information
- Creating a data-aware applet
- Troubleshooting

Using drivers that require native DLLs

Some JDBC drivers require the use of native DLLs. The JDBC-ODBC bridge is one of them.

The JDBC-ODBC bridge uses the Sun native call interface which the Microsoft Virtual Machine does not support. For these kinds of drivers, you need to use the Sun Virtual Machine instead. (Also see Important JDBC 1.21 information for information on how to use the JDBC-ODBC bridge with this build of Jato.)

To use the Sun VM instead of the Microsoft VM, open the Targets window, from the File menu choose Run Options and click on Use Sun's Java interpreter. In most cases, you are not able to use drivers that use native DLLs in browsers like Netscape 3.01 or Internet Explorer 3.01. In that case, you can only use the bridge in a Java application or use appletviewer to run your Java applets that use the bridge.

Setting the ClassPath

If you are using a JDBC driver (such as the JDBC-ODBC bridge) other than the Sybase jdbcCONNECT driver, you need to specify to Jato where the `.class` files for the drivers are located. This is done by setting up the ClassPath.

◆ **To add a driver to your target's ClassPath**

1. Click **Targets** on the **View** menu to open Targets window.
2. In the tree view, expand your target and its **Default Options** to see a **Java Target** item.
3. Right click on the **Java Target** item and click **Properties** to open the Properties for Java Target file(s) property sheet.
4. Click **ClassPath** to go to the ClassPath page.
5. For **Folder**, type in the path up to (but not including) the top-most folder of the package where your driver can be found. If the driver's class files are in a ZIP file, enter the full path of the ZIP file.
6. Click **Add** to add the path to the list.

Important: The Jato ClassPath is separate from a ClassPath environment variable. If you set a ClassPath environment variable it will not be used by Jato.

❑ Jato Pre-Release Trial Version Read Me First Information

❑ How to use this pre-release version of Jato

❑ Database applications

Debugging in Jato

The debugger in Jato can only be used if the Microsoft VM is used.

To debug applications using database components, there are two things that you can do:

TraceToLog property

You can set the **TraceToLog** property in a Transaction and in a Query object. If this property is TRUE at run-time, a file called DB.LOG will be created in the folder in which the applet or application was run. You can check this file for error messages.

If you are running an applet from a browser, the security features of the browser may prevent a file from being written to the local machine. In that case, the file DB.LOG will not be created.

Database log

You can use the database log facility (DbUtil.setLog) to record low-level activities to the log stream. The log stream can be specified by using the **Debug.setLogStream** method. The default log stream is the System.err stream.

It is helpful to execute the above statement in the constructor of the form where transaction and query objects are placed. For example, if the transaction object cannot be connected, you will see the error message in the log stream.

If you are using the Sun VM, you can open the Java Console and the log will be displayed there.

DataSource URL

You need to specify a DataSource URL to make a connection with a Transaction object. The URL is the standard one that is used by normal JDBC programs.

The format of the JDBC URL is driver-dependent. For instance, the JDBC-ODBC bridge uses the format `jdbc:odbc:datasource`, while the Sybase jdbcCONNECT driver uses the format `jdbc:sybase:Tds:host:port`. Refer to the documentation on your particular JDBC driver for details.

Important JDBC 1.21 information

This build of Jato uses JDBC 1.21. JDBC 1.21 was designed for Java 1.1, but until Java 1.1 has been released and widely incorporated into browsers, replacing Java 1.02, there is a compatibility issue that you need to be aware of. This section describes the issue and how you can deal with it in this release of Jato.

With Java 1.1 (currently in beta testing), JDBC becomes part of the Java specification. That is, the JDBC interfaces and the JDBC driver manager are part of Java 1.1. The package name is "java.sql". If you have Java 1.1 installed, the JDBC interfaces are installed locally and so are available for an applet to use.

The situation is much different with Java 1.0.2, which is what browsers and this release of Jato are currently using. Since the "java.sql" package is not part of Java 1.0.2, the classes in the "java.sql" package must be downloaded or installed separately.

However, the Java applet security model forbids downloading of packages that have names starting with "java.", so the extra classes needed for JDBC 1.21 with Java 1.02 must either be installed locally or downloaded under a different package name.

Following JavaSoft's recommendation, JDBC driver vendors have been taking the JDBC interfaces defined by "java.sql" (which are well documented) and redefining them to be in a different package called "jdbc.sql". They then define their drivers to use and return the interfaces defined in "jdbc.sql" as opposed to "java.sql". You can then put your applet, the "jdbc.sql" interfaces and the driver classes on the web server where they can be downloaded and run on the user's machine.

However, this means that a driver vendor may have to support two JDBC drivers - one driver that works with the "java.sql" package (so that users of Java 1.1 can use the driver) and one that works with "jdbc.sql" (so that users of Java 1.0.2 can use the driver). The driver that uses the "java.sql" package will only work in Java 1.1-enabled browsers, but will require less download time since the "java.sql" package is already installed on their machine. The driver that uses the "jdbc.sql" package will work on all browsers but will require more time to download (even using Java 1.1, since the driver doesn't make use of the built-in "java.sql" package).

Thus some drivers, such as the Sybase jdbcCONNECT driver included with Jato, use the "jdbc.sql" package and others, such as the JDBC-ODBC Bridge, use the "java.sql" package.

Important: Jato includes both packages and two sets of the Powersoft classes: one looks for the JDBC classes in the "jdbc.sql" package and the other uses the "java.sql" package. The Jato installation sets Jato up to use the "jdbc.sql" set.

You can switch between the sets of Powersoft classes by using the `switchit.bat` script in your Jato `java\lib` folder.

◆ **To use the "jdbc.sql" set of Powersoft classes**

1. Open a console prompt and change to the `java\lib` folder under your Jato folder. This folder should contain a batch file called `switchit.bat`.
2. Run the command:

   ```
   switchit sybase
   ```

   If there is a folder named `powersoft.syb`, the `powersoft` folder will be renamed to `powersoft.sun` and then the `powersoft.syb` folder will be renamed to `powersoft`. The `powersoft` folder will now have classes that use the "jdbc.sql" package.

When Jato is set up to use the "jdbc.sql" package you can use the included jdbcCONNECT driver or other drivers set up for that package name.

◆ **To use the "java.sql" set of Powersoft classes**

1. Open a console prompt and change to the `java\lib` folder under your Jato folder. This folder should contain a batch file called `switchit.bat`.
2. Run the command:

   ```
   switchit sun
   ```

   If there is a folder named `powersoft.sun`, the `powersoft` folder will be renamed to `powersoft.syb` and then the `powersoft.sun` folder will be renamed to `powersoft`. The `powersoft` folder will now have classes that use the "java.sql" package.

When Jato is set up to use the "java.sql" package you can use the JDBC-ODBC Bridge (1.2001) driver or other drivers set up for that package name.

Creating a data-aware applet

This shows you how to write a simple applet that uses the powerful database components in Jato. The instructions assume you have set up the JDBC-ODBC Bridge driver and a copy of SQL Anywhere 5.0 has been installed on your machine. If not, modify the procedure as required.

Important: To use the JDBC-ODBC Bridge driver, you must run

```
switchit.bat sun
```

in your Jato `java\lib` folder, as described in

You may also wish to examine the JDBCTransaction sample project to see how to use the database components in this release of Jato.

◆ **Create a database application:**

1. Open a new project. By default, an applet target is created for you.
2. Add a Transaction object. Edit the properties, and set:
   Name: `transaction_1`
   JDBC Driver: `sun.jdbc.odbc.JdbcOdbcDriver`
   DataSource URL: `jdbc:odbc:SQL Anywhere 5.0 Sample`
   UserID: `dba`
   Password: `sql`
   AutoConnect: `true`
3. Create a Query object on the form, and set its properties as follows:
   Name: `query_1`
   Transaction: `transaction_1`
   SQL: `select * from employee`
   AutoOpen: `true`
4. Create a data navigator on the form and set:
   DataSource: `query_1`
5. Create a list box and set:
   Bound Control: `true`
   DataSource: `query_1`
   DataColumns: `emp_lname`
   DataTrackRow: `true`
   DataBindAsLookup: `false`
6. Create a text box and set:
   Bound Control: `true`
   DataSource: `query_1`
   DataColumns: `emp_fname`

Now try running it!

Troubleshooting

Have you set up the ClassPath correctly to include the path of your JDBC driver classes?

Can your JDBC driver be used with the Microsoft VM? (The native interfaces of the Sun VM and Microsoft VM are not compatible, so drivers using native interfaces can't work with both VMs.)

Can Windows locate all needed driver DLLs? You may need to copy them to your Windows system folder.

Are there any security restrictions that might prevent the use of sockets?

Have you used the database log?

Have you used the TraceToLog property in the Transaction and Query objects?

Known problems

This is a pre-release version of Jato. Some of the functionality planned for the product code-named “Jato” is not available or does not function properly in this version. We are providing this software on an “as is” basis so that we can get user feedback early in the product development cycle, and to give you early access to the functionality that has been implemented.

Some of the problems you may encounter include:

When you uninstall Jato it can uninstall the Microsoft Java VM. This happens because Internet Explorer does not increment registry counters for shared files when it installs the Microsoft Java Virtual Machine
Solution: Reinstall Internet Explorer after uninstalling Jato.

If you install Internet Explorer 3.0 after Jato, debugging in Jato will not work. Internet Explorer 3.0 includes an older version of the Java VM than that included with Internet Explorer 3.01 and this build of Jato.
Workaround: Install Internet Explorer 3.01 or uninstall and re-install Jato.

Changing the name of a form using its property sheet will not properly rename the generated `.java` file.
Workaround: Use the Classes windows to rename the form.

Changing the variable name for a menu item won't update the name of its click handler (assuming one exists).
Workaround: The name can be changed manually.

On a Java application target, the design-time text property for a form is ignored at run time.

Putting menu items on a root menu bar (not in a popup in the component) will be ignored at runtime.

Jato incorrectly allows you to create two different targets in the same folder, which can adversely affect its operation.
Workaround: Use different folders for each target.

You cannot debug Web application targets.
Workaround: Debug the included applets or applications instead of the Web application target.

If you are using the original Windows 95 version of the file `oleaut32.dll` (dated 07/11/95), Jato will not be able to register ActiveX components.
Solution: Install the Service Pack for Windows 95 or install Microsoft Internet Explorer 3.01 or newer to update your version of `oleaut32.dll`.

Font Sizes are not correct at runtime. Font Colors (Background and Foreground) are not correct as well. Java fonts may not match the Windows ones due to an AWT problem.

The scroll bar performs incorrectly when the thumb is dragged to several different values and the scroll bar is then moved down one line.

Group boxes may sometimes cause components to be hidden at run time.

A combo box with zero items or only one item will not paint correctly due to an AWT problem.

Changing the vertical size of a ComboBox in the property sheet does not work due to the behavior of AWT.

The alignment properties on a Label component do not work at run time.

The Java console is not echoing output that gets sent to it.

There is a problem in the powersoft.jcm.ServerSocket component due to a bug in Sun's JDK 1.0.2. The problem is that you cannot close() a ServerSocket, which is blocked in an accept(), from another thread. This is because in the java.net.ServerSocket component, the close() and accept() methods are synchronized. This may be fixed in JDK 1.1.

The Microsoft Java Virtual Machine may fault or loop when garbage collecting. The problem appears to be caused by memory allocated outside the Java VM. The problem has been seen in Jato for both Web Server extensions and NetImpact Dynamo Java servlets, both of which use coexist with programs that allocate memory outside the Java VM.

Currently there is no way to access queries created in DynamoScript from within a Java plugin. For example, in the template

```
<!--SQL select id from product -->
<!--java classname -->
```

the Java code for classname cannot access the SQL statement.
Workaround: Use connection.CreateQuery() to create a new query.

Sending feedback

The pre-release trial version of this product is provided at no cost, and does not include free technical support. However, Powersoft is interested in receiving your feedback so that it may continue to improve the product.

If you have a bug to report or you would like to see an enhancement to the product, you can contact technical support using either the Powersoft Newsgroup Service or the Powersoft WebExpress. Powersoft will consider all of the feedback that it receives, however the company cannot guarantee that every suggestion will be implemented into subsequent versions of the product.

Powersoft Newsgroup Service

To access the newsgroup, connect to the forums.powersoft.com news server and use the `powersoft.public.code_name_jato.beta` newsgroup. If your newsreader or browser supports it, you can use the URL:

`news://forums.powersoft.com/powersoft.public.code_name_jato.beta`

This news server also has a newsgroup for the jdbcCONNECT driver, called `sybase.public.jdbcconnect`. If your newsreader or browser supports it, you can use the URL:

`news://forums.powersoft.com/sybase.public.jdbcconnect`

Powersoft WebExpress

To run WebExpress, use a World Wide Web (WWW) browser to access:

`http://www.powersoft.com/services/support/optima`

Then select the **WebExpress** tab. The shared **Express Id** for this beta program is 873356 and the **Pin** is 60715. This Web Express ID is only valid for feedback related to the beta program for this downloaded product. It will be deactivated when it is no longer required for the beta program.

For information on WebExpress, see:

`http://www.powersoft.com/services/webexpress.html`

Types of feedback

We encourage you to ask “how to” questions, suggest product enhancements and (especially) report software bugs. The appropriate services to be used to report each type of issue are described below.

“How to” questions

To answer "How to" type questions, first check the on-line documentation that has been provided with the product (`*.hlp` files). Please pay special attention to `Readme` online help (this help file), which contains valuable information pertaining to this beta release. If there is still an issue related the use of our product, please post a message on our Powersoft Newsgroup Service.

The success of the newsgroup will rely on customers, such as you, assisting each other within the newsgroup. Powersoft may monitor this newsgroup from time to time. However, we do not guarantee that we will be able to respond to newsgroup postings.

Product enhancements

Suggestions for improvements to our product can be made using the Powersoft Newsgroup Service. Please post a message in the newsgroup that describes the product enhancement. Please prefix your enhancement request with the text, *** Enhancement Request ***.

Suggestions for the enhancement of our product will be forwarded to the Powersoft product management group. Enhancement requests will be considered for subsequent releases.

Bugs

Bugs can be entered into our issue tracking system using WebExpress. To accomplish this, select the "Submit a New Issue" option from the main menu. The status of a bug can be determined by entering the "Individual Issue" Number from the main menu of WebExpress.

Technical support will review bugs entered via WebExpress within 48 hours of their creation. Technical Support will contact customers directly only if additional information is required.

Powersoft cannot commit to follow-up communications with each customer that contacts the organization.