



BeanMachine for the Web

**By Carol A. Jones, IBM Corp.
Lead Architect, Lotus BeanMachine**

Abstract

Who is building Web sites today? Graphic designers, writers, and marketing and communications people are building them, not Java programmers. In fact, very few Web sites are actually built by professional programmers. That's why Lotus BeanMachine is important: It brings the power of Java to non-programming Web-builders.

This paper will describe how you can use BeanMachine to add Java applets to your Web pages, even if you don't know anything about Java programming. You'll learn how to assemble and connect multimedia parts into a working applet and then how to publish the applet on a Web server. All of BeanMachine's parts are JavaBeans, which are reusable components that adhere to the Java standard. This paper will also explain about why the JavaBeans specification is a key element in making reusable Java components a reality, and you'll see the specification put into practice as you learn how to add new JavaBeans to BeanMachine's palette.

Basics of BeanMachine

So what is BeanMachine all about? From a design point of view, it's about simplicity. You'll find helpful wizards, lots of interesting, reusable samples and artwork, and powerful, commonly used, ready-to-go parts that you can easily assemble into your own applet. From an engineering point of view, it's about JavaBeans. JavaBeans is a specification, defined by Sun Microsystems, for building reusable Java components (these resulting components are also called "JavaBeans"). When Java programmers follow this specification, the classes they produce can be used with BeanMachine. The JavaBeans specification is what makes it possible to easily assemble the parts into the whole, producing Java applets that would have been quite difficult to build otherwise.

Let's take a closer look, starting with a quick tour of BeanMachine's windows and palettes. The main window is called the Composer window. The white rectangle near the upper left corner represents your applet, into which you drop parts from the palette, which is the small window in the upper right of the Composer window.

What kind of parts can you drop? There are three groups of parts that come with BeanMachine. There are Multimedia parts, such as Animation, Audio, Ticker Tape, and Text. There are Networking parts, such as a Database part, a URL Link for linking to other Web pages, and an e-mail part. You'll learn more about these parts a little later. The third category of parts is called Controls, which includes all the basic things you

need to build data entry forms such as buttons and text fields.

After you've dropped some parts into your applet, you'll want to start defining more details about them by setting their properties in the Details window, which is in the lower right of the Composer window. Properties are simply data that define how a part looks or behaves. For instance, the picture property of an Image part would define which .GIF file is displayed. The transition property would indicate whether or not you want a transition effect, such as a dissolve or iris, as the image is being displayed.

Since many of the elements of a Java applet are related to artwork or other multimedia files, the Gallery is an important window. It displays thumbnails of all the artwork in a particular folder. You can display thumbnails from different folders, and the Gallery keeps track of the artwork folders you've used most recently so you can quickly go back to them. By clicking on one of these thumbnails (for example, the thumbnail of a GIF file), you can quickly do two steps in one: You can add the image part to your applet and set its picture property automatically. You don't have to use the Gallery at all, but many people find it easier to work visually, using these thumbnails to help them find the right artwork quickly and easily.

Connecting JavaBeans Together

Every JavaBean has a certain set of properties, methods, and events. Properties are the data that describe the part: its picture, its speed, its width and height, and so on. Every bean has a different list of properties that uniquely define it. Beans are also active objects; that is, they know how to do things. Animations can play and stop. Buttons can show and hide themselves. Again, every kind of bean has a different set of actions it knows how to perform; these are called methods. Events help a bean interact with other beans. An event is a kind of signal that something important has happened: a button was clicked, a transition has finished, or a ticker tape was refreshed. In BeanMachine, every part on the palette is a JavaBean, so it has properties, methods (which BeanMachine calls actions), and events.

On the Connections tab of the Details window, you can connect the events and actions of parts to add logic to your applet. A connection can be thought of as a command or interaction between two parts. They take the form of "when something happens (an event), do something else (an action)." The "do something else" can be any action, such as playing an audio clip, setting a property like background color, or stopping an animation. When you make connections, you decide which events trigger which actions. For example, you can have an animation play when a certain button is clicked.

The Connections page shows you all the connections for the part of your applet that you currently have selected in the Composer window. To make a connection, you click on the different columns to pick the events and actions you want to use. You can have as many connections as you want for a part, and they don't all have to start from the same event. You can copy and paste connections, and you can also reorder them and delete them if you need to.

The Palette

Let's take a deeper look at the palette to get a better idea of what you can assemble and connect to make your own Java applet. There are several categories of parts you can choose from.

Multimedia Category

These parts are mostly for adding nice special effects to your applet. These are the parts you can really have fun with!

Animation	A part for playing a sequence of graphical images	Rollover	An image part that can change when the mouse moves over it
Audio	A part for playing an audio clip	Teletype	A part that displays text one letter at a time with an optional sound
Clock	A part for displaying the date and time	Text	A part for displaying text
Image	A part for a graphical image	Ticker Tape	A moving text area part
Nervous Text	A part that displays text and randomly moves the letters around	Time	A part that works like a stopwatch

Controls Category

These parts are mostly for adding forms to your Web pages. You can collect or present data with these controls.

Button	A button part	List	A list box part
Checkbox	A checkbox part	Panel	A part that can contain other parts and that has its own layout style
Choice	A drop-down list part	Text Area	A multi-line text entry area part
Label	A part for displaying a line of text	Text Field	A single-line text entry field part

Networking Category

These parts are used for networking outside of your applet. Use them to get data from a database or go to different Web pages.

Database	A part that displays data from a database	Mail Link	A part that displays the browser's mail dialog
Headline	A part for displaying a scrolling list of text with URL links	URL Link	A part that links to another Web page

Importing JavaBeans

A JavaBean can constitute an entire applet or only one class. partBeanMachine provides a fast and simple way to import any other Bean and use it just as you would use a JavaBean in BeanMachine.part The Bean Wizard guides you through the steps.

On the Welcome page of the Bean Wizard, you enter the name of the JavaBean that you want to use. Next, you can pick out an icon for the new part and choose which Palette category it will be stored in. There are several other pages in the wizard: one for actions, one for properties, and one for events. You use these pages to decide what will be available in the Details window. You don't have to start from scratch, though, because BeanMachine reads the class and decides what properties, actions, and events the part should have. How does it do this? First, BeanMachine discovers the properties by applying Design Patterns to the names of the methods in the class. Design Patterns are rules of the JavaBeans specification that indicate how "get" and "set" methods of properties should be named. If the author of the Java class has followed the usual naming conventions, BeanMachine will find all the properties that the bean was intended to have. The naming conventions are very simple: The "get" methods should be named "get" plus the property name, and the "set" methods should be named "set" plus the property name.

There's really no way for BeanMachine to know exactly which actions and events you want, so you'll need to check the ones you want on the Actions page of the Bean Wizard. The wizard does try to help you out by checking the show, hide, enable, and disable actions for you. Similarly, for events, you'll get the basic mouse events, unless you check other ones that you want on the Events page of the Bean Wizard.

Java programmers who carefully follow the JavaBeans specification will also provide "BeanInfo" classes with their reusable JavaBeans. If there is such a class, BeanMachine will use it instead of guessing at the properties, events, and actions. You can still make changes on the different pages of the wizard, but the BeanInfo class indicates what the author intended, so you should not need to do anything more.

Writing Java Code

There are times when you need to do more than BeanMachine's basic properties and connections provide. If you know how to program in Java, you can write your own code using BeanMachine's Java window. The methods you write are available in Connections, as actions listed with the part. At the top of the Java window, there is a toolbar to help you get started writing your Java methods:

Paste Part	Pastes in the Java code that accesses a part in your applet	Paste Property	Pastes in the Java code to get or set a property
Paste Action	Pastes in the Java code to call a method	Paste Java	Pastes in a Java statement, such as an if or while statement.

Publishing Your Applet

Now that you know how to build great Java applets, how do you get them onto a Web server? You might have a LAN connection to the server, or maybe you are using a Web server maintained by an Internet service provider. You might even want to publish to a Marimba Castanet Transmitter on a server. In any case, how do you get your applet and all the necessary files on the server so your applet can be viewed and accessed by other people? BeanMachine's Publish Wizard helps you publish almost anywhere. It offers you three choices:

- You can publish to a folder, either on your own computer or on a LAN drive.
- You can publish to a remote system using an FTP account.
- You can publish to a Castanet Transmitter.

BeanMachine can also make a NetObjects Fusion Component if you want to use your applet in an HTML page created by NetObjects Fusion.

Optimizing Download Performance

As applets become more complex, they become larger and take longer to download. If you are downloading separate Java classes, downloading is even slower, because your browser must communicate back to the Web server to get each class. BeanMachine helps you shorten overall download time by compressing all of your applet's classes into one file, which the browser can download all at one time. When you publish your applet, BeanMachine identifies and locates the necessary .class and media files used in your applet and packs them into one compressed file, and then it generates all the necessary HTML tags so your browser will recognize this file.

Sample Applets

Here's a link to the BeanMachine Web site, where you can see the latest samples.
<http://www2.lotus.com/developers/tools/beanmachine/>

Other companies, products, and service names may be trademarks or service marks of others.

Copyright Trademark

[Java Feature Page](#) [Java Home](#)

