BEA Jolt

JoltBeans Developer's Guide

Jolt Release 1.1
Document Edition 1.0
May 1998

Copyright

Restricted Rights Legend

Trademarks or Service Marks

JoltBeans Developer's Guide

Document Edition	Date	Software Version
1.0	May 1998	Jolt Release 1.1

Contents

3. Using JoltBeans

4. JoltBeans Toolkit Class Library Reference

5. Jolt Aware AWT Beans Class Library Reference

Preface

Purpose of This Document

This document describes the BEA JoltBeans product and gives instructions for using a tool for creating applications.

Who Should Read This Document

This document is intended for developers who are interested in building applications using JavaBeans and Jolt. It assumes a familiarity with BEA Jolt, BEA TUXEDO, Integrated Development Environments (IDEs), JavaBeans, and Java programming.

How This Document Is Organized

The *BEA Jolt JoltBeans Developer's Guide* is organized as follows:

- Chapter 1, "Introducing the JoltBeans Toolkit," describes the JoltBeans features and how the beans work.
- Chapter 2, "Installing the JoltBeans Product," provides JoltBeans installation requirements.
- Chapter 3, "Using JoltBeans," describes how developers use the JoltBeans to create applications.

- Chapter 4, “JoltBeans Toolkit Class Library Reference,” is a reference of the JoltBeans toolkit methods and classes.
- Chapter 5, “Jolt Aware AWT Beans Class Library Reference,” is a reference of the AWT JoltBeans methods and classes.

How to Use This Document

This document, *BEA Jolt JoltBeans Developer’s Guide*, is designed primarily as an online, hypertext document. If you are reading this as a paper publication, note that to get full use from this document you should install and access it as an online document via a Web browser.

The following sections explain how to view this document online, and how to print a copy of this document.

Opening the Document in a Web Browser

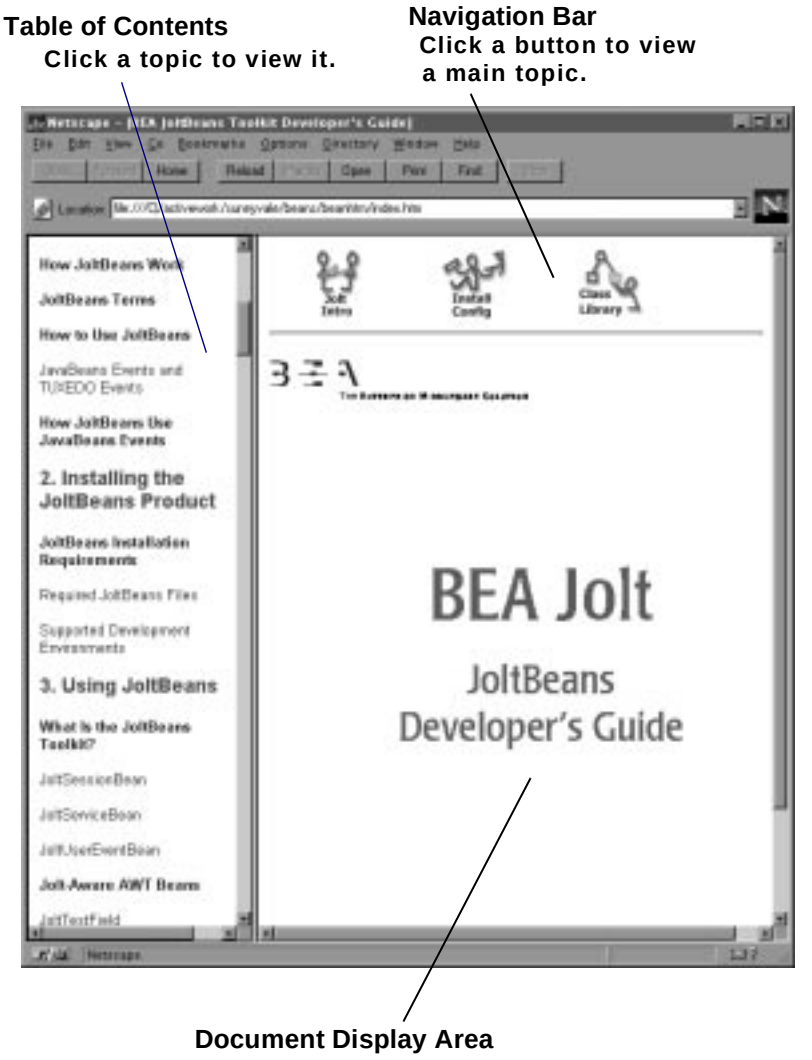
To access the online version of this document, open the following HTML file in a Web browser:

```
/beansdoc/index.htm
```

Note: The online documentation requires a Web browser that supports HTML version 3.0. Netscape Navigator version 2.02 or Microsoft Internet Explorer version 3.0 or later are recommended.

Figure 1 shows the online document with the clickable navigation bar and table of contents.

Figure 1 Online Document Displayed in a Netscape Web Browser



Printing from a Web Browser

You can print a copy of this document, one file at a time, from the Web browser. Before you print, make sure that the chapter or appendix you want is displayed and *selected* in your browser. (To select a chapter or appendix, click anywhere inside the chapter or appendix you want to print. If your browser offers a Print Preview feature, you can use the feature to verify which chapter or appendix you are about to print.)

The BEA Jolt Online Home Page also includes Adobe Acrobat PDF files of all of the online documents. You can use the Adobe Acrobat Reader to print all or a portion of each document.

Documentation Conventions

The following documentation conventions are used throughout this document.

Convention	Item
boldface text	Indicates terms defined in the glossary.
Ctrl+Tab	Indicates that you must press two or more keys sequentially.
italics	Indicates emphasis or book titles.

`monospace text`	Indicates code samples, commands and their options, data structures and their members, data types, directories, and file names and their extensions. Monospace text also indicates text that you must enter from the keyboard.
	Examples:
	`#include <iostream.h> void main ( ) the pointer psz`
	`chmod u+w *`
	`\tux\data\ap`
	`.doc`
	`tux.doc`
	`BITMAP`
	`float`

Convention	Item
`monospace boldface text`	Identifies significant words in code. *Example*: `void` **`commit`** `( )`
`monospace italic text`	Identifies variables in code. *Example*: `String` *`expr`*
UPPERCASE TEXT	Indicates device names, environment variables, and logical operators. *Examples*: LPT1 SIGNON OR
`{ }`	Indicates a set of choices in a syntax line. The braces themselves should never be typed.

`[ ]`	Indicates optional items in a syntax line. The brackets themselves should never be typed.
	Example:
	`buildobjclient [-v] [-o name ] [-f` *`file-list`*`]...`
	`[-l` *`file-list`*`]...`

`|`	Separates mutually exclusive choices in a syntax line. The symbol itself should never be typed.

`...`	Indicates one of the following in a command line:
	♦ That an argument can be repeated several times in a command line
	♦ That the statement omits additional optional arguments
	♦ That you can enter additional parameters, values, or other information
	The ellipsis itself should never be typed.
	Example:
	`buildobjclient [-v] [-o name ] [-f` *`file-list`*`]...`
	`[-l` *`file-list`*`]...`

. . .	Indicates the omission of items from a code example or from a syntax line. The vertical ellipsis itself should never be typed.

Related Documentation

The following sections list the documentation provided with the Jolt software, and other publications related to Jolt and TUXEDO technology.

Jolt Documentation

The Jolt information set consists of the following documents:

BEA Jolt User's Guide

BEA Jolt Release Notes

Note: The BEA Jolt Online Documentation Home Page also includes Adobe Acrobat PDF files of all of the online documents. You can use the Adobe Acrobat Reader to print all or a portion of each document.

BEA Publications

The following BEA publications are also available:

TUXEDO System Reference Manual

TUXEDO System Programmer's Guide, Volumes 1 and 2

Contact Information

The following sections provide information about how to obtain support for the documentation and software.

Documentation Support

If you have questions or comments on the documentation, you can contact the BEA Information Engineering Group by e-mail at **docsupport@beasys.com** or by telephone at **+1.408.542.4193**. (For information about how to contact Customer Support, refer to the following section.)

Customer Support

If you have any questions about this version of BEA Jolt Add-ons, or if you have problems installing and running BEA JoltBeans, contact BEA Customer Support through BEA WebSupport at `www.beasys.com`. You can also contact Customer Support by using the contact information provided on the Customer Support Card, which is included in the product package.

When contacting Customer Support, be prepared to provide the following information:

- Your name, e-mail address, phone number, and fax number
- Your company name and company address
- Your machine type and authorization codes
- The name and version of the product you are using
- A description of the problem and the content of pertinent error messages

1 Introducing the JoltBeans Toolkit

The JoltBeans toolkit provides a JavaBeans compliant interface to BEA Jolt.

The JoltBeans toolkit contains beans which wrap the existing Jolt class library into reusable bean components, such as the JoltSessionBean or the JoltServiceBean. These beans can be customized easily by giving application specific values to properties and connecting them with other bean components.

You can use the JoltBeans toolkit with your Integrated Development Environment (IDE) to create Jolt clients that can access a TUXEDO application.

This section includes the following topics:

- How JoltBeans Work
- JoltBeans Terms
- How to Use JoltBeans
- How JoltBeans Use JavaBeans Events

How JoltBeans Work

The JoltBeans product includes a set of .jar files that you can import to your IDE. For example, if your IDE is Symantec Visual Cafe´, “drag and drop” the .jar file into the Component Library window of Visual Cafe´.

Note: Currently, Symantec Visual Cafe´ 2.5 is the only IDE that has been certified by BEA for use with JoltBeans.

JoltBeans Terms

Refer to the following terms as you work with JoltBeans:

Java Bean
: A reusable software component that can be manipulated visually in a builder tool.

JoltBeans product
: Consists of two sets of Java Beans: JoltBeans toolkit and Jolt aware AWT beans.

Custom GUI element
: A Java GUI class that communicates with JoltBeans. The means of communication can be JavaBeans events, methods, or properties offered by JoltBeans.

Jolt aware bean
: A bean that is source of JoltInputEvents, listener of JoltOutputEvents, or both. Jolt aware beans are a subset of Custom GUI elements that follow beans guidelines.

Jolt aware AWT beans
: JoltList, JoltCheckBox, JoltTextField, JoltLabel, and JoltChoice.

JoltBeans toolkit
: A JavaBeans compliant interface to BEA Jolt, which includes the JoltServiceBean, JoltSessionBean, and JoltUserEventBean.

wiring
: Indicates that a bean is registered as a listener of events from another bean.

How to Use JoltBeans

During development, drag the beans from the JoltBeans component palette of your development environment to the Java form designer for a Jolt client application or applet. You must populate the properties of the beans (see Table 3-6) and set up the event source-listener relationships between various beans of the application or applet. The development tool typically generates the event hook-up code. Finally, add the application logic to the event callbacks.

JavaBeans Events and TUXEDO Events

Java Beans communicate via events. The concept of events in a BEA TUXEDO system and events in a JavaBeans environment differ. In a TUXEDO application, an event is raised from one part of an application to another part of the same application. JoltBeans events are events that communicate between beans. See "Using TUXEDO Event Subscription and Notification with JoltBeans" in Chapter 3, "Using JoltBeans," for more information.

How JoltBeans Use JavaBeans Events

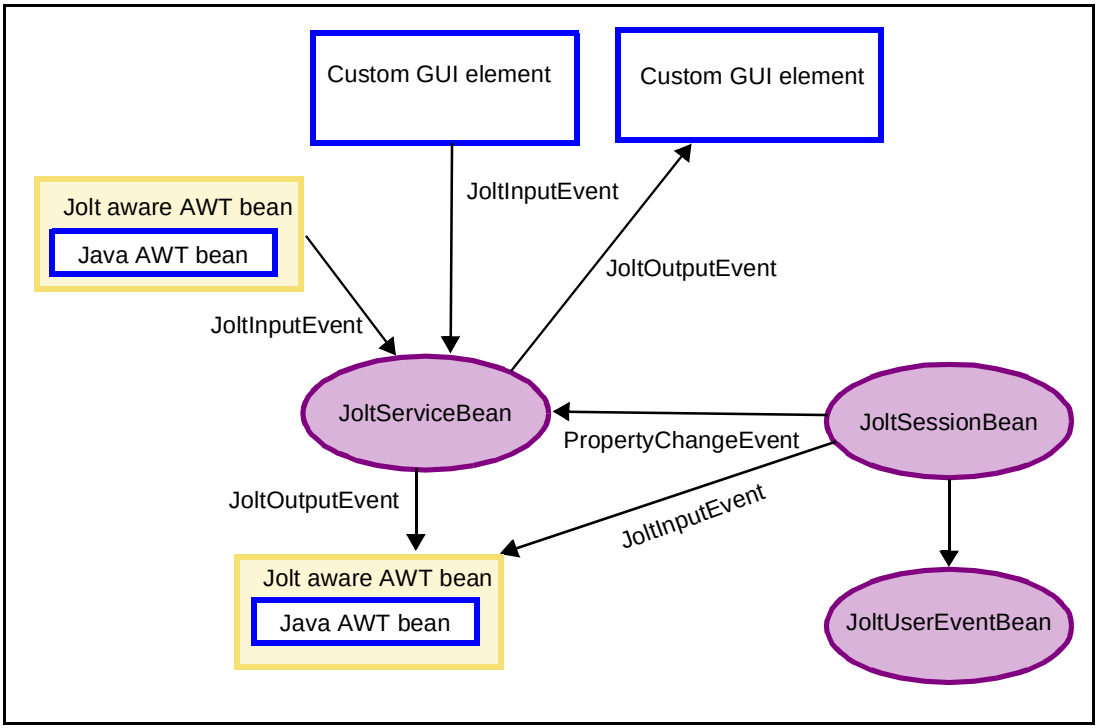
A Jolt client applet or application that has been built using JoltBeans typically consists of Jolt aware AWT beans, such as JoltTextField or JoltList, and JoltBeans, such as JoltServiceBean and JoltSessionBean. The main mode of communication between beans is by JavaBeans events.

Jolt aware beans are sources of JoltInputEvents or listeners of JoltOutputEvents or both. JoltServiceBeans are sources of JoltOutputEvents and listeners of JoltInputEvents.

The Jolt aware AWT beans expose properties and methods so you can link the beans directly to parameters of a TUXEDO service (represented by a JoltServiceBean). Jolt aware beans notify the JoltServiceBean via JoltInputEvent when their content changes. The JoltServiceBean sends a JoltOutputEvent to all registered Jolt aware beans when the reply data is available after the service call. The Jolt aware AWT beans contain logic that update their contents with the corresponding output parameter of the service

Figure 1-1 shows a graphical representation of the possible relationships among the JoltBeans.

Figure 1-1 Possible Interrelationships Among the JoltBeans



CHAPTER

2 Installing the JoltBeans Product

This chapter includes JoltBeans Installation Requirements information.

For additional information about the Jolt 1.1 installation, refer to the *BEA Jolt User's Guide*, *BEA Jolt Release Notes* (for Version 1.1), and *BEA Jolt Release Notes* (for Version 1.1, Volume 2).

JoltBeans Installation Requirements

The following components are required before installing JoltBeans:

- BEA Jolt Version 1.1 or BEA Jolt Version 1.1, Volume 2
- BEA Jolt Patch Level 5 or higher

Note: The BEA Jolt Patch level 5 or higher must be installed before installing the JoltBeans toolkit. Contact BEA Customer Support to request this patch.

Required JoltBeans Files

The .tar file (UNIX) and the .exe file (NT) contain the following:

- `README` — contains additional JoltBeans installation and configuration information.
- `JoltBeanDev.jar` — contains the complete set of files required for the usage of the JoltBeans toolkit in an Integrated Development Environment (IDE).
- `JoltBeanRt.jar` — a subset of `JoltBeanDev.jar` and contains only the files necessary to deploy applets/applications that are using the JoltBeans toolkit.
- `JoltBeanDevAwt.jar` — contains the complete set of files required for the development of applets/applications using Jolt aware AWT beans in an IDE.
- `JoltBeanRtAwt.jar` — a subset of `JoltBeanDevAwt.jar` and contains only the files necessary to deploy Jolt aware AWT beans applets/applications.

Supported Development Environments

Currently, Symantec Visual Cafe´ 2.5 is the only certified and supported IDE.

3 Using JoltBeans

JoltBeans is a collection of beans that you can use to build your own Jolt client applications or applets. During development, take the beans from the JoltBeans toolkit in the component palette of a development tool and position them on the Java form (or forms) of a Jolt client application or applet. Populate the properties of the beans and establish event source-listener relationships between various beans of the application or applet. Typically, the development tool is used to generate the event hook-up code, or you can code the hook-up manually.

The JoltBeans product consists of two sets of Java Beans. The first set, the JoltBeans toolkit, is a beans version of the Jolt API. The second set, Jolt aware AWT beans, is a "Jolt enabled" version for some of the standard JDK 1.1 AWT components. These Jolt aware AWT components help you to build Jolt client GUIs with minimal or no coding.

Note: To use the toolkit, it is recommended that you are familiar with JavaBeans enabled Integrated Development Environments (IDEs). The walkthrough in this chapter is based on Symantec's Visual Cafe´ 2.5 IDE and illustrates the basic steps of building a sample applet.

This chapter contains the following topics:

- Building the Sample Form
- Wiring the JoltBeans Together
- Using the JoltBeans Repository and Setting the Property Values

- JoltBeans Programming Tasks
  - Using Transactions with JoltBeans
  - Using Custom GUI Elements with the JoltService Bean
  - Using TUXEDO Event Subscription and Notification with JoltBeans

What Is the JoltBeans Toolkit?

The JoltBeans toolkit includes the following beans:

- JoltSessionBean
- JoltServiceBean
- JoltUserEventBean

These components encompass the complete Jolt Class Library as Java Beans.

The following sections provide information about the properties for each bean. For additional information, refer to your *BEA Jolt User's Guide* or BEA TUXEDO documentation.

See Chapter 4, "JoltBeans Toolkit Class Library Reference," and Chapter 5, "Jolt Aware AWT Beans Class Library Reference," for additional information about the JoltBeans toolkit and Jolt aware AWT beans classes, constructors, and methods.

JoltSessionBean

The JoltSessionBean represents the TUXEDO session. It encapsulates the functionality of the JoltSession, JoltSessionAttributes, and JoltTransaction classes. The JoltSessionBean offers properties to set session and security attributes, such as send timeout or TUXEDO user name, as well as methods to open and to close a TUXEDO session.

The JoltSessionBean sends a PropertyChange event when the TUXEDO session is established or closed. PropertyChange is a standard bean event defined in the java.beans package. The purpose of this event is to signal other beans about a change of the value of a property in the source bean. In this case, the source is the JoltSessionBean, the targets are JoltServiceBeans or JoltUserEventBeans, and the property changing is the LoggedOn property of the JoltSessionBean. When a logon is successful and a session is established, LoggedOn is set to `true`. After the logoff is successful and the session is closed, the LoggedOn property is set to `false`.

The JoltSessionBean provides methods to control transactions, including beginTransaction(), commitTransaction(), and rollbackTransaction().

Table 3-1 shows the JoltSessionBean properties and descriptions.

Table 3-1 JoltSessionBean Properties

Property	Description
AppAddress	Set the IP address (host name) and port number of the JSL or the Jolt Relay. The format is `//host:port number` (e.g., `myhost:7000`).
AppPassword	Set the TUXEDO application password used at logon, if required.
IdleTimeOut	Set the IDLETIMEOUT value.
inTransaction	Indicate true or false depending if a transaction has been started and not committed or aborted.
LoggedOn	Indicate true or false if a TUXEDO session does or does not exist.
ReceiveTimeOut	Set the RECVTIMEOUT value.
SendTimeOut	Set the SENDTIMEOUT value.
SessionTimeOut	Set the SESSIONTIMEOUT value.

Table 3-1 JoltSessionBean Properties

Property	Description
UserName	Indicate the TUXEDO user name, if required.
UserPassword	Indicate the TUXEDO user, if required.
UserRole	Indicate the TUXEDO user role, if required.

JoltServiceBean

The JoltServiceBean represents a remote TUXEDO service. The name of the service is set as a property of the JoltServiceBean. The JoltServiceBean listens to JoltInputEvents from other beans to populate its input buffer. JoltServiceBean offers the callService() method to invoke the service. JoltServiceBean is an event source for JoltOutputEvents that carry information about the output of the service. After a successful callService(), listener beans are notified via a JoltOutputEvent which carries the reply message.

Although the primary way of changing and querying the underlying message buffer of the JoltServiceBean is via events, the JoltServiceBean also provides methods to access the underlying message buffer directly (setInputValue(…), getOutputValue(…)).

Table 3-2 shows the JoltServiceBean properties and descriptions.

Table 3-2 JoltServiceBean Properties

Property	Description
ServiceName	The name of the TUXEDO service represented by this JoltServiceBean.
Session	The JoltSessionBean associated with the bean that allows access to the TUXEDO client session.
Transactional	Set to true if this JoltServiceBean is to be included in the transaction that was started by its JoltSessionBean.

JoltUserEventBean

The JoltUserEventBean provides access to TUXEDO events. The TUXEDO event to subscribe to or unsubscribe from is defined by setting the appropriate properties of this bean (event name and event filter). The actual event notification is delivered in the form of a JoltOutputEvent from the JoltSessionBean.

Table 3-3 shows the JoltUserEventBean properties and descriptions.

Table 3-3 JoltUserEventBean Properties

Property	Description
EventName	Set the name of the user event represented by the bean.
Filter	Set the event filter
Session	The JoltSessionBean associated with the bean that allows access to the TUXEDO client session.

Jolt Aware AWT Beans

The Jolt aware AWT Beans are inherited from the JDK 1.1 Abstract Windowing Toolkit. They include:

- JoltTextField
- JoltLabel
- JoltList
- JoltCheckbox
- JoltChoice

Note: Refer to Chapter 5, “Jolt Aware AWT Beans Class Library Reference,” for more information.

JoltTextField

A Jolt aware extension of java.awt.TextField that is linked to a specific field in the Jolt input or output buffer by its JoltFieldName property. If the field occurs multiple times, the occurrence this textfield is linked to is specified by the occurrenceIndex property of this bean.

JoltTextField can be connected with JoltServiceBeans in two ways:

- JoltTextField contains parts of the input for a service. A JoltServiceBean may listen to events raised by a JoltTextField. JoltTextField sends JoltInputEvents to its listeners (typically JoltServiceBeans) when its contents changes.
- JoltTextField displays output from a service. In this case, JoltTextField listens to JoltOutputEvents from JoltServiceBeans and updates its contents according to the occurrence of the field to which it is linked.

JoltLabel

This is a Jolt aware extension of java.awt.Label that is linked to a specific field in the Jolt output buffer by its JoltFieldName property. If the field occurs multiple times, the occurrence this textfield is linked to is specified by the occurrenceIndex property of this bean. JoltLabel can be connected with JoltServiceBeans to display output from a service. A JoltLabel listens to JoltOutputEvents from JoltServiceBeans and updates its contents according to the occurrence of the field to which it is linked.

JoltList

This is a Jolt aware extension of java.awt.List that is linked to a specific Jolt field in the Jolt input or output buffer by its JoltFieldName property. If the field occurs multiple times in the Jolt input buffer, the occurrence this list is linked to is specified by the occurrenceIndex property of this bean. JoltList can be connected with JoltServiceBeans in two ways:

- JoltList contains parts of the input for a service. A JoltServiceBean listens to events raised by a JoltList. JoltList sends JoltInputEvents to its listeners when

the selection in the listbox changes. The JoltInputEvent, in this case, is populated with the single value of the selected item.

- JoltList displays output from a service. When used to display the output of a service, JoltList listens to JoltOutputEvents from JoltServiceBeans and updates its contents accordingly with all occurrences of the field to which it is linked.

JoltCheckbox

The JoltCheckbox is a Jolt aware extension of java.awt.Checkbox, that is linked to a specific field in the Jolt input buffer by its JoltFieldName property. If the field occurs multiple times, the occurrence this checkbox is linked to is specified by the occurrenceIndex property of this bean.

It can be connected with JoltServiceBeans to contain parts of the input for a service. A JoltServiceBean may listen to events raised by a JoltCheckbox. JoltCheckbox sends JoltInputEvents to its listeners (typically JoltServiceBeans) when the selection in the checkbox changes. The JoltInputEvent in this case is populated with the `TrueValue` property of data type String (if the box is selected) or `FalseValue` (if the box is unselected).

JoltChoice

The JoltChoice provides a Jolt aware extension of java.awt.Choice, that is linked to a specific field in the Jolt input buffer by its JoltFieldName property. If the field occurs multiple times, the occurrence this choice is linked to is specified by the occurrenceIndex property of this bean.

It can be connected with JoltServiceBeans to contain parts of the input for a service. A JoltServiceBean may listen to events raised by a JoltChoice. JoltChoice sends JoltInputEvents to its listeners (typically JoltServiceBeans) when the selection in the choicebox changes. The JoltInputEvent in this case is populated with the single value of the selected item.

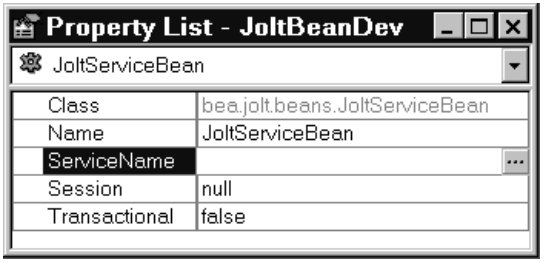
Using the Property List and the Property Editor to Modify the JoltBeans Properties

The values of most JoltBeans properties can be modified by simply editing the right column of the Property List shown in Figure 3-1.

For some properties of JoltBeans, Custom Property Editors are provided.

The Custom Property Editors, accessed from the Property List, include dialog boxes that are used to modify the property values. You can invoke the Custom Property Editors from the Property List by selecting the button with the ellipsis "..." that is next to the value of the corresponding property value. An example is shown in Figure 3-1.

Figure 3-1 Example of the Property List and Ellipsis Button



When the ellipsis button is selected, the Property Editor shown in Figure 3-2 displays.

Figure 3-2 Property Editor Dialog Box

The Custom Property Editors of JoltBeans read cached information. Initially, there is no cached information available, so when the Property Editor is used for the first time, the dialog box is empty. Log on to the Jolt repository and load the property editor cache from the repository.

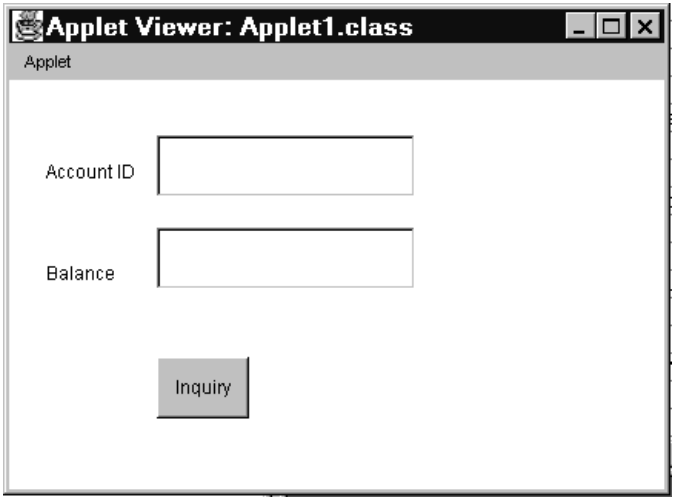
Details of the logon and an example of using the Property List and Property Editor are shown in the "Using the JoltBeans Repository and Setting the Property Values" section of the "JoltBeans Class Library Walkthrough."

JoltBeans Class Library Walkthrough

This walkthrough describes how to build an applet that is used to enter the account ID, click on the Inquiry button, and display the balance of the account shown in Figure 3-3.

This is an example of a completed Java form containing JoltBeans. The applet implements the client functionality for the INQUIRY service of BANKAPP.

Figure 3-3 Sample Inquiry Applet



Refer to Figure 3-5 for an example of each item. To begin, select the following items shown in Table 3-4.

Table 3-4 Required Form Components

Component	Purpose
Applet	A form used to paint the beans in your development environment.
JoltSessionBean	Logs on to a TUXEDO session.
JoltTextField	Gets input from the user (in this case, ACCOUNT_ID).
JoltTextField	Displays the result (in this case, SBALANCE).
JoltServiceBean	Accesses a TUXEDO service. (In this case, INQUIRY from BANKAPP).
Button	Initiates an action.
Label	Describes the field on the applet.

Building the Sample Form

The following example is created using the Visual Cafe´ 2.5 development environment. Follow the steps to build a sample form.

1. Drag and drop the beans (as shown in Figure 3-4) to the desired location (shown in Figure 3-5).

Figure 3-4 JoltBeans and the Form Designer in Visual Cafe´

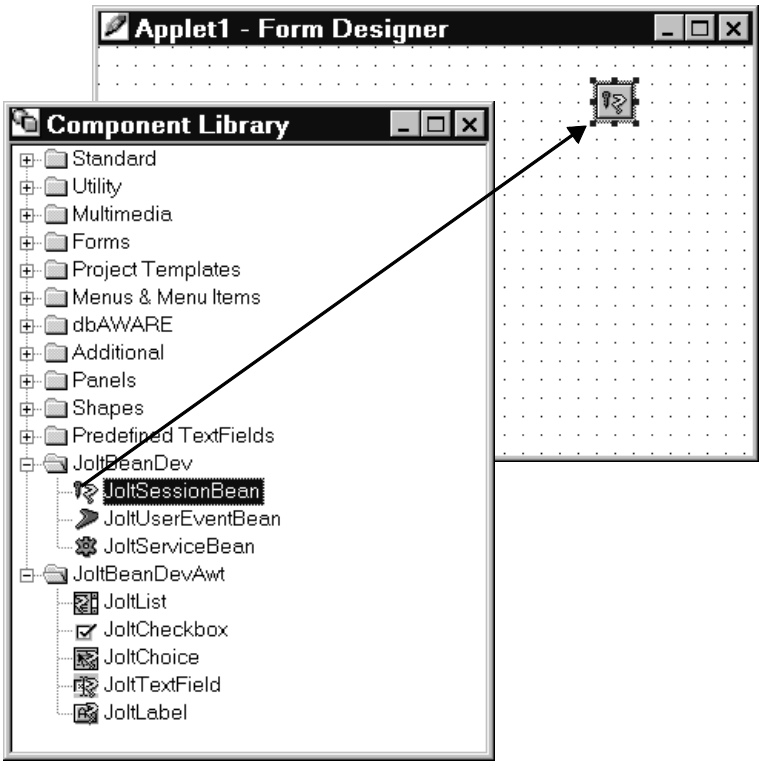


Figure 3-4 shows how a JoltBean is selected and painted to the palette of the Visual Cafe´ development environment.

Figure 3-5 Java Applet Form Designer

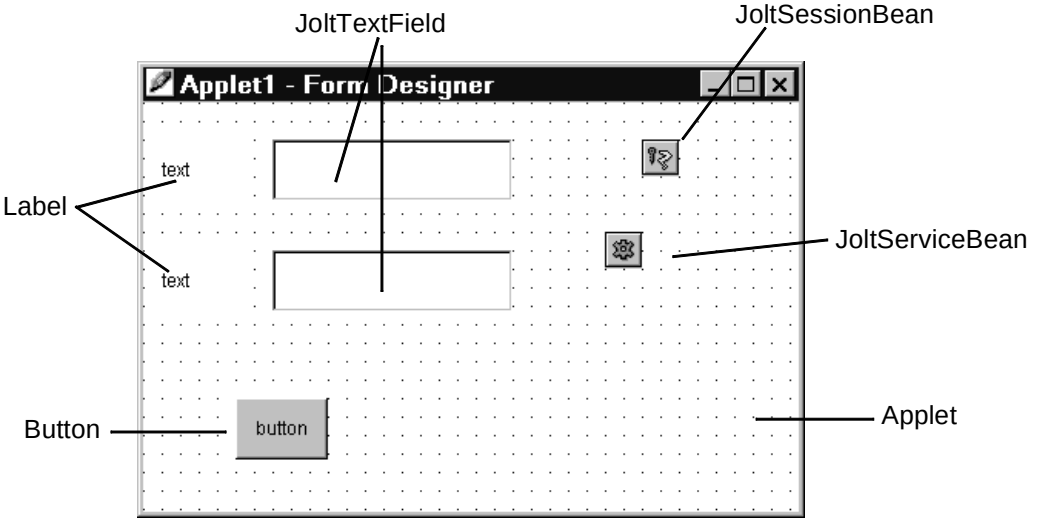


Figure 3-5 shows how the JoltBeans appear in a development environment when the placement of the beans is completed.

2. To modify or customize the buttons, labels or fields, use the Property List. Some JoltBeans use a Custom Property Editor. The example in Figure 3-6 shows how selecting the JoltFieldName of the button property list displays the Custom Property Editor. Set the properties of the beans (e.g., set the JoltFieldName property of the JoltTextField to ACCOUNT_ID. Refer to Figure 3-6).

Note: For complete information on setting and modifying the properties of the JoltBeans, refer to "Using the JoltBeans Repository and Setting the Property Values"section in this chapter.

Table 3-5 specifies the property values that should be set. **Bold** values are required, plain text recommended, and *italic* values are examples of required values that will vary for your configuration.

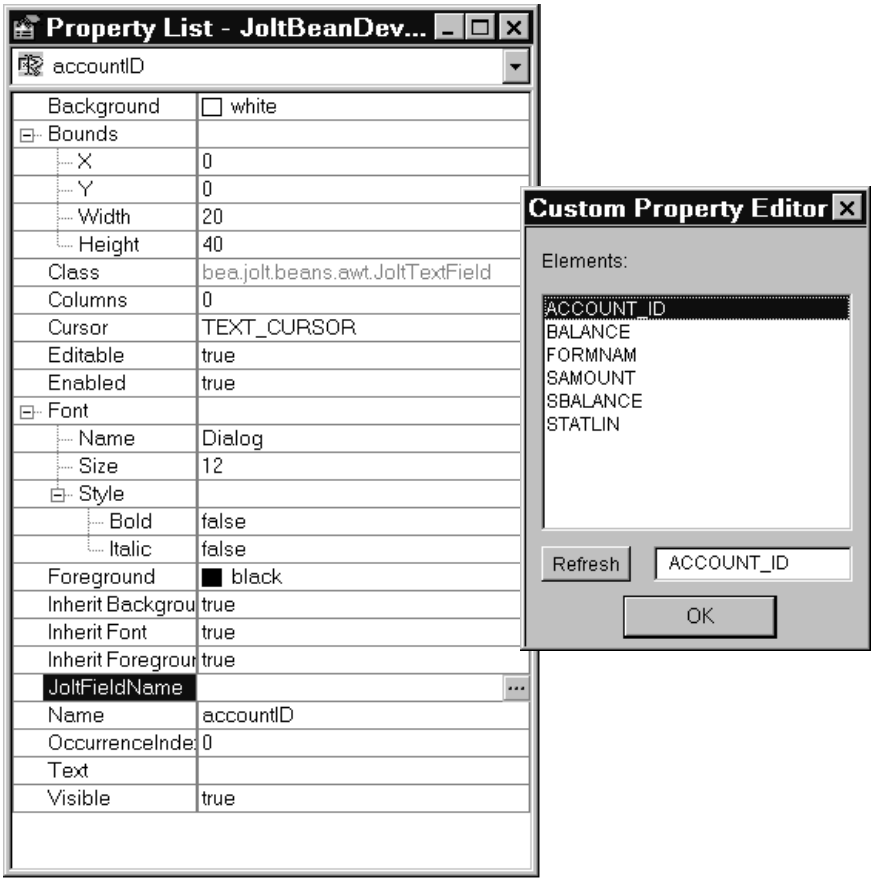
Table 3-5 Required and Recommended Property Values

Bean	Property	Value
label1	Text	Account ID
label2	Text	Balance
JoltTextField1	Name	accountId
JoltTextField1	JoltFieldName	**ACCOUNT_ID**
JoltTextField2	Name	balance
JoltTextField1	JoltFieldName	**SBALANCE**
JoltSessionBean1	AppAddress	*//tuxserv:2010*
JoltServiceBean1	Name	inquiry
JoltServiceBean1	ServiceName	**INQUIRY**
button1	Label	Inquiry

Note: In this walkthrough, the default occurrenceIndex of 0 works for both JoltTextFields.

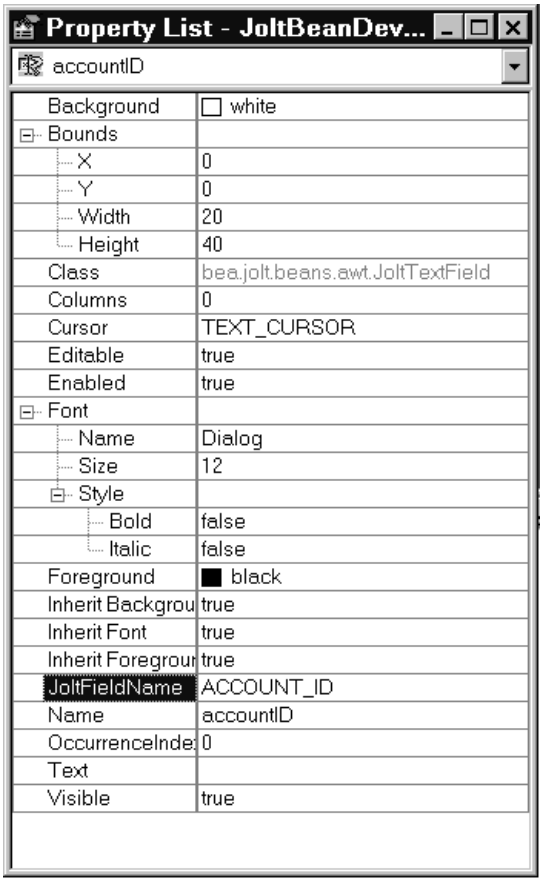
Refer to Table 3-6 and the section, “Using the JoltBeans Repository and Setting the Property Values” for general guidelines on JoltBean properties.

Figure 3-6 Example of JoltTextField Property List and Custom Property Editor



3. To change the value of the JoltFieldName property, click on the ellipsis button of the JoltFieldName in the Property List. The Custom Property Editor displays. Select or type the new field name (in this example, "ACCOUNT_ID"). Select **OK**.

Figure 3-7 Revised JoltFieldName in the JoltTextField Property List



The change is reflected in the Property List shown in Figure 3-7 and on the button shown in Figure 3-8.

Figure 3-8 Example of Applet Form Designer with Property Changes

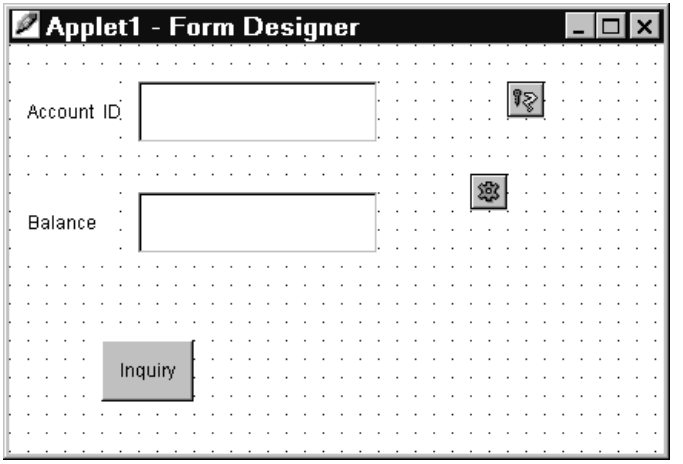


Figure 3-8 is an example of how the text on the Button changes after the label text is added to the Property List fields for these beans.

4. After you set the properties to the right values (refer to Table 3-5 for additional information on the required and recommended property values), define how the beans will interact by wiring them together using the Visual Cafe´ Interaction Wizard similar to the windows shown in Figure 3-10 and Figure 3-11.

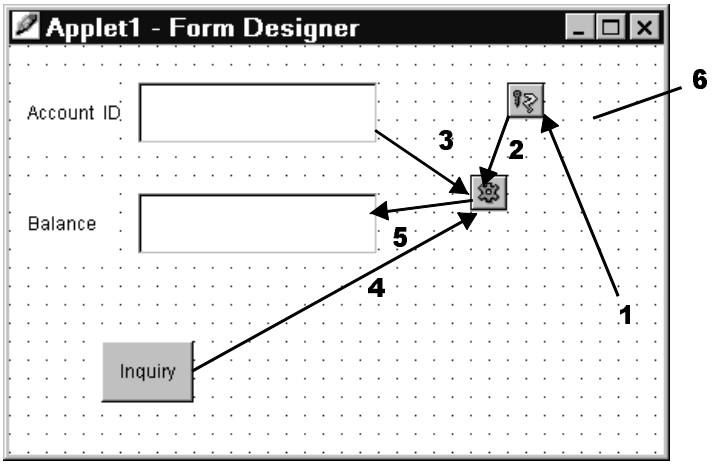
Wiring the JoltBeans Together

After all of the beans are positioned on your form and the properties are set, wire the events together. Save, compile, and run the applet. Figure 3-9 gives an example of the flow to help you determine the order when you are ready to wire the beans.

Wiring the beans allows you to establish event source-listener relationships between various beans of the form. For example, the JoltServiceBean is a listener of ActionEvents from the button and invokes callService() when the event is received. The development environment provides a tool (in this example, Visual Cafe´ Interaction Wizard) to wire beans together. In the absence of this tool, the event source-listener pairs are implemented manually.

For more information, refer to Figure 3-9 and the accompanying steps.

Figure 3-9 Example of How the Form Runs



Step 1: Wire the JoltSessionBean logon

1. Select the Interaction Wizard button. Click in the applet window (not on another bean) and drag a line to the JoltSessionBean. The Interaction Wizard window similar to the example shown in Figure 3-10 displays.

Figure 3-10 Visual Cafe´ Interaction Wizard (Window 1 of 2)

2. Choose ComponentShown as event and select “Logon to the TUXEDO system” as method. Select **Finish**.

Note: The results of completing these two steps will enable the logon() method of the JoltSessionBean to be triggered by an applet (e.g., ComponentShown) that is sent when the applet is opened for the first time.

Step 2: Wire JoltSessionBean to JoltServiceBean via propertyChange

1. Select the Interaction Wizard button. Click on the JoltSessionBean and drag a line to the JoltServiceBean. The Interaction Wizard window similar to the example shown in Figure 3-10 displays.

2. Choose propertyChange as event and select “Handle a property change event...” as the method. Select **Next**. The Interaction Wizard “How it should happen” window similar to the example shown Figure 3-11 displays.

Figure 3-11 Visual Cafe´ Interaction Wizard (Window 2 of 2)

3. Select “A variable or parameter” method. In this example, **event** is the default and the correct value. Select **Finish**.

Note: The results of completing these three steps will enable the JoltSessionBean to send a propertyChange event when logon() completes. The JoltServiceBean listens to this event and associates its service with this session.

Step 3: Wire the accountID JoltTextField as input to the JoltServiceBean using JoltInputEvent

1. Select the Interaction Wizard button. Select the accountID JoltTextField bean and drag a line to the JoltServiceBean. The Interaction Wizard window similar to the example shown in Figure 3-10 displays.
2. Choose dataChanged as event and select "Handle a jolt input event" as method. Select **Next**. The Interaction Wizard "How it should happen" window similar to the example shown in Figure 3-11 displays.
3. Select "A variable or parameter." Select **Finish**.

Note: The results of completing these three steps will enable you to type the account number in the first text field. The JoltFieldName property of this JoltTextField is set to "ACCOUNT_ID". Whenever the text inside this text box changes, it sends a JoltInputEvent to the JoltServiceBean. (The JoltServiceBean listens to JoltInputEvents from this textbox.) The JoltInputEvent object contains the name, value, and occurrence index of the field.

Step 4: Wire Button to JoltServiceBean using JoltAction

1. Select the Interaction Wizard button. Select the Inquiry Button and drag a line to the JoltServiceBean. The Interaction Wizard window similar to the example shown in Figure 3-10 displays.
2. Choose Action as the event. Select "Invoke TUXEDO Service represented by this bean" as the method. Select **Finish**.

Note: The results of completing these two steps will enable the callService() method of the JoltServiceBean to be triggered by an ActionEvent from the Inquiry button.

Step 5: Wire JoltServiceBean to the balance JoltTextField using JoltOutputEvent

1. Select the Interaction Wizard button. Select the JoltServiceBean and drag a line to the AmountJoltTextField bean. The Interaction Wizard window similar to the example shown in Figure 3-10 displays.
2. Choose serviceReturned as the event. Select "serviceReturned..." as the method. Press **Next**. The Interaction Wizard "How it should happen" similar to the example shown in Figure 3-11 displays.
3. Select "A variable or parameter." Select **Finish**.

Note: The results of completing these three steps of the JoltServiceBean sends a JoltOutputEvent when it receives reply data from the remote service. The JoltOutputEvent object contains methods to access fields in the output buffer. The JoltTextField displays the result of the INQUIRY service.

Step 6: Wire the JoltSessionBean logoff

1. Select the Interaction Wizard button. Click in the applet window (not on another bean) and drag a line to the JoltSessionBean. The Interaction Wizard window similar to the example shown in Figure 3-10 displays.
2. Choose componentHidden as event and select "Logoff from the TUXEDO system" as method. Select **Finish**.

Note: The results of completing these two steps will enable the logoff() method of the JoltSessionBean to be triggered by an applet (e.g., componentHidden) that is sent when the applet gets hidden.

Step 7: Compile the applet

Upon completion of the wiring, compile your applet using your development software.

Fill in the empty catch blocks for exceptions. For example, you can add the following error handling code:

```
e.printStackTrace();
```

Check the message window for any compilation errors and exceptions.

Refer to Table 3-6 and Figure 3-12 for additional information.

Using the JoltBeans Repository and Setting the Property Values

Custom Property Editors are provided for the following properties:

- JoltFieldName (Jolt aware AWT beans)
- serviceName (JoltServiceBean)

The Property Editor, accessed from the Property List, includes dialog boxes that are used add or modify the properties. You can invoke the boxes from the Property List by selecting the button with the ellipsis "..." that is next to the value of the corresponding property value.

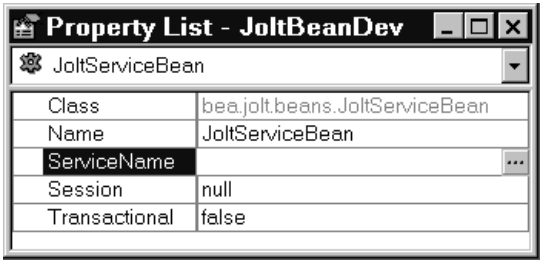
Some JoltBeans require input in the Property List field. The beans are listed in Table 3-6.

Table 3-6 JoltBean Specific Properties

JoltBean	Property	Input Description
JoltSessionBean	appAddress	e.g., `//host:port`
	userName, Password or AppPassword	Type your TUXEDO user name and passwords.
JoltServiceBean	serviceName	e.g., INQUIRY
	isTransactional	Set to true if the service needs to be executed within a transaction. Set isTransactional to false if the service does not require a transaction.
JoltUserEventBean	eventName	Refer to the TUXEDO `tpsubscribe` calls.
	filter	
all Jolt aware AWT beans	joltFieldName	e.g., ACCOUNT_ID
	occurrenceIndex	Multiple fields of the same name. Index starts at 0.
JoltCheckbox	TrueValue and FalseValue	The field value corresponding to the state of the checkbox.

The property editor reads cached information from the repository and returns names of the available services and data elements in a list box. An example of the ServiceName property editor is shown in Figure 3-12:

Figure 3-12 JoltServiceBean Property Editor



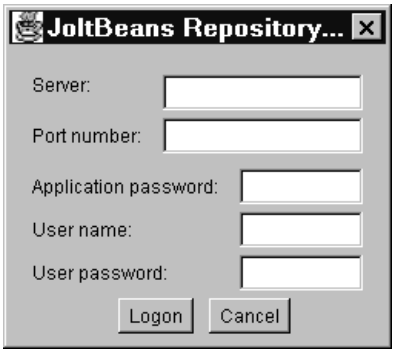
1. Select the service name by clicking on the ellipsis in the ServiceName field shown in Figure 3-12.
2. The Custom Property Editor for ServiceName shown in Figure 3-13 displays.

Figure 3-13 Custom Property Editor for serviceName

Note: If you cannot or do not want to connect to the Repository database, simply type the service name in the text box and proceed to Step 7.

3. If you are not logged on, select **Logon**. The JoltBeans Repository Logon shown in Figure 3-14 displays.

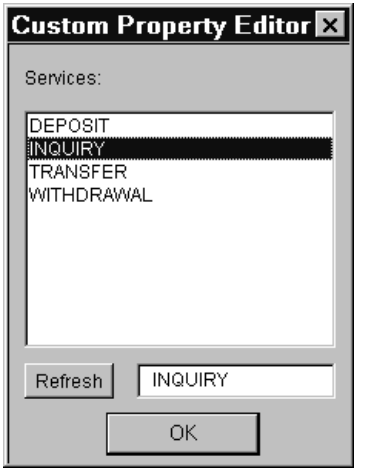
Figure 3-14 JoltBeans Repository Log On



4. Type the TUXEDO or Jolt Relay Machine name for Server and the JSL or Jolt Relay Port Number. Type passwords and user name information (if required) and select **Logon**.

5. The Property Editor loads its cache from the repository.

Figure 3-15 Property Editor with Selected Service



6. Select the appropriate service name in the list box shown in Figure 3-15. Enter the property value (service or field name) directly. A text box is provided. Select **OK** on the property editor dialog in Figure 3-15. The bean property gets set with the contents of the textbox.
7. Select **OK** on the Custom Property Editor dialog shown in Figure 3-15.

JoltBeans Programming Tasks

The additional procedures include:

- Using Transactions with JoltBeans
- Using Custom GUI Elements with the JoltService Bean
- Using TUXEDO Event Subscription and Notification with JoltBeans

Using Transactions with JoltBeans

Your TUXEDO application services may have functionality that will update your database. If so, you can use transactions with JoltBeans (e.g., in BANKAPP, the services TRANSFER and WITHDRAWAL update the database of BANKAPP). If your application service is read-only (e.g., INQUIRY), you do not need to use transactions.

The following example shows how to use transactions with JoltBeans.

1. `setTransactional` (true) is called on the JoltServiceBean. (isTransactional is a boolean property of the JoltServiceBean.)
2. `beginTransaction()` is called on the JoltSessionBean.
3. `callService()` is called on the JoltServiceBean.
4. Depending on the outcome of the service call, `commitTransaction()` or `rollbackTransaction()` is called on the JoltSessionBean.

Using Custom GUI Elements with the JoltService Bean

The JoltBeans product provide a limited set of GUI components that are Jolt enabled. It is also possible to use controls that are not Jolt enabled together with the JoltServiceBean. Figure 3-16 outlines the different ways to link controls that are not Jolt enabled with the JoltServiceBean. Three examples are presented. In Example 1, the GUI element displays output information of the service represented by the JoltServiceBean, in the Examples 2 and 3, the control contains input information.

In Example 1, the GUI element uses an adapter class that implements the JoltOutputListener interface to listen to JoltOutputEvents. The JoltServiceBean as the event source for JoltOutputEvents calls the `serviceReturned()` method of the adapter class when it sends a JoltOutputEvent. Inside `serviceReturned()`, the control's internal data is updated using information from the event object.

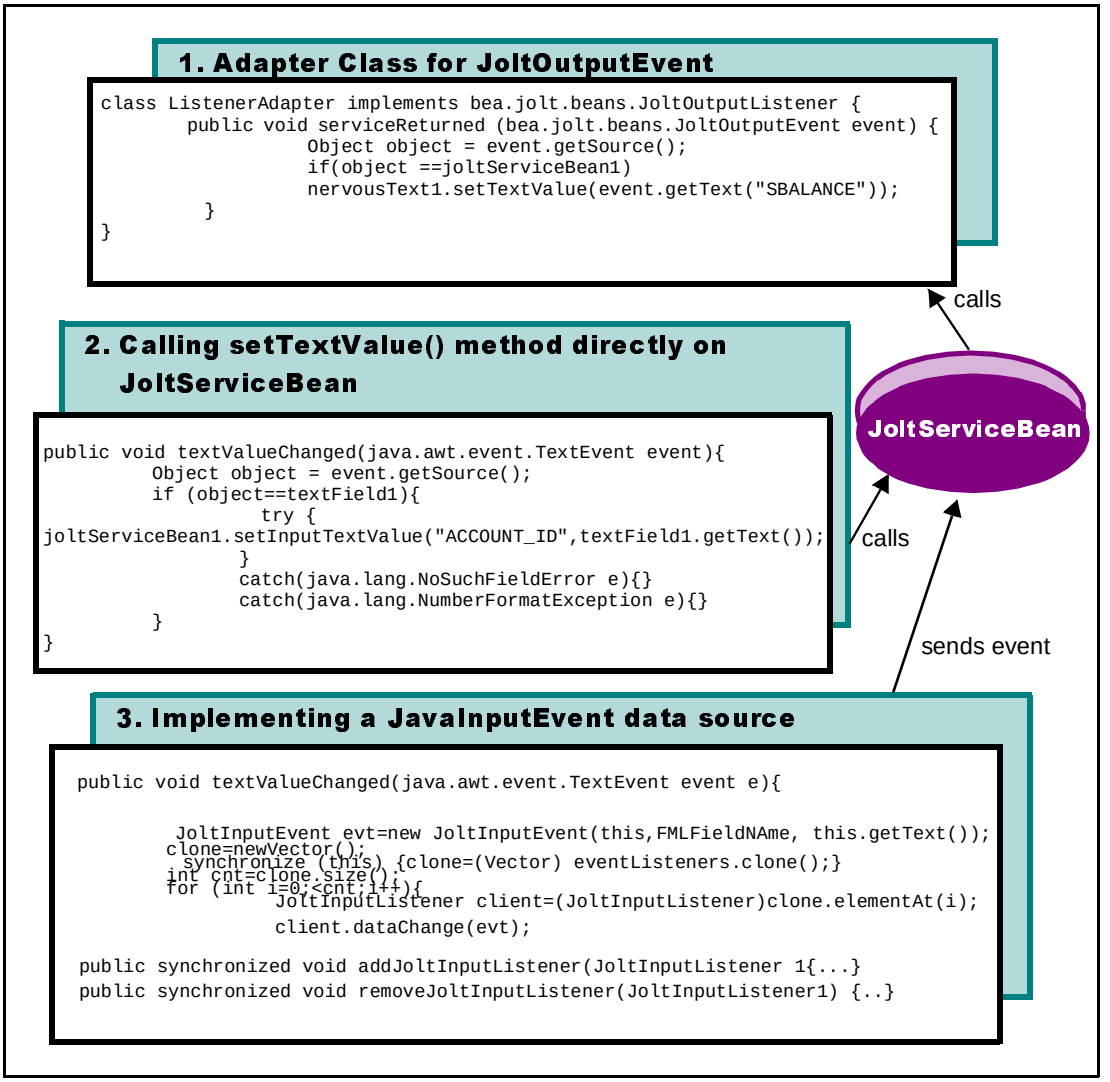
The development tool generates the adapter class when the JoltServiceBean and the GUI element are wired together.

In Example 2, the GUI element calls the `setInputTextValue()` method on the JoltServiceBean. In this example, the GUI element contains data which is input for the TUXEDO service represented by the JoltServiceBean.

In Example 3, the GUI element implements the required methods (addJoltInputListener and removeJoltInputListener) to act as an event source for JoltInputEvents. The JoltServiceBean acts as an event listener for these events. The control sends a JoltInputEvent once its own state changes to keep the JoltServiceBean updated with the input information.

Refer to Figure 3-16 for a graphical representation of using the GUI elements with the JoltServiceBean.

Figure 3-16 Example of Using Custom GUI Elements with the JoltServiceBean



Using TUXEDO Event Subscription and Notification with JoltBeans

TUXEDO supports brokered and unsolicited event notification. Jolt provides a mechanism for Jolt clients to receive TUXEDO events. JoltBeans also include this capability.

Note: TUXEDO event subscription and notification is different than JavaBeans events.

The following example shows how the TUXEDO asynchronous notification mechanism is used in JoltBeans applications.

1. Use the setEventName() and setFilter() methods of the JoltUserEventBean to specify the TUXEDO event to which you want to subscribe.
2. The component that wants to receive the event notifications registers itself as a JoltOutputListener to the JoltSessionBean.
3. `subscribe()` is called on JoltUserEventBean.
4. When the actual TUXEDO event notification arrives, JoltSessionBean sends a JoltOutputEvent to its listeners by calling `serviceReturned()` on them. The JoltOutputEvent object contains the data of the TUXEDO event.

When the client is not interested any more in the event it calls `unsubscribe()` on the JoltUserEventBean.

Note: If the client wants to only subscribe to unsolicited events, use setEventName (`"\\.UNSOLMSG"`) that can be set using the property sheet. `EventName` and `Filter` are properties of the JoltUserEventBean.)

4 JoltBeans Toolkit Class Library Reference

The JoltBeans Toolkit Class Library provides a JavaBeans compliant interface to Jolt. JoltBeans wrap the existing Jolt class library into reusable bean components, such as the JoltSessionBean or the JoltServiceBean. JoltBeans components are not specific to any given application.

Note: For information on the Jolt aware AWT Beans, that include JoltLabel, JoltCheckbox, JoltTextField, JoltChoice, and JoltList, refer to Chapter 5, "Jolt Aware AWT Beans Class Library Reference."

The bea.jolt.beans package includes the following classes:

- JoltInputEvent Class
- JoltOutputEvent Class
- JoltServiceBean Class
- JoltSessionBean Class
- JoltUserEventBean Class

JoltInputEvent Class

```
java.lang.Object
   |
   +----java.util.EventObject
            |
            +----bea.jolt.beans.JoltInputEvent
```

JoltInputEvent carries information about input data to JoltServiceBeans. An instance of JoltInputEvent represents a single input field (possibly with multiple occurrences) to a TUXEDO service. Typically, this event is sent by a GUI element when its data content changes.

JoltInputEvent Constructors

The JoltInputEvent constructors create objects that carry information about input data to JoltServiceBeans. The input field can be a single data element, one of many occurrence of an element, or an entire array. Input data can be either a string or a native datatype (e.g. Float, Double).

JoltInputEvent — single data element

This constructor is used when the input is a single data element and it is represented in a native data format.

Synopsis

`JoltInputEvent``(Object` *`source`*`, String` *`fieldName`*`, Object` *`value`*`)`

Parameters

`source`
Event source

`fieldName`
Name of the input field (Jolt name)

`value`
Value of the field in a native data type, such as Integer or Float

Usage

This constructor is used when the input is a single data element and it is represented in a native data format.

JoltInputEvent — multiple occurrences; only one occurrence set

This constructor is used when the input is a data element with multiple occurrences, but only one occurrence is to be set, and it is represented in a native data format.

Synopsis

```
JoltInputEvent(Object source, String fieldName, Object value,
int pos)
```

Parameters

source
Event source

fieldName
Name of the input field (Jolt name)

value
Value of the field in a native data type, such as Integer or Float

pos
The position of the data element

Usage

This constructor is used when the input is a data element with multiple occurrences, but only one occurrence is to be set, and it is represented in a native data format.

JoltInputEvent — data element with multiple occurrences

This constructor is used when the input is a data element with multiple occurrences and it is represented in a native data format.

Synopsis

```
JoltInputEvent(Object source, String fieldName, Object values[])
```

Parameters

source
Event source

fieldName
Name of the input field (Jolt name)

values
Array of occurrences of the field in a native data type, such as Integer or Float

Usage

This constructor is used when the input is a data element with multiple occurrences and it is represented in a native data format.

JoltInputEvent — single element; String

This constructor is used when the input is a single data element and it is represented as a String.

Synopsis

```
JoltInputEvent(Object source, String fieldName, String textValue)
```

Parameters

source
Event source

fieldName
Name of the input field (Jolt name)

textValue
Value of the field as a String

Usage

This constructor is used when the input is a single data element and it is represented as a String. The JoltInputEvent object will perform the necessary data conversion.

JoltInputEvent — multiple occurrences; only one occurrence set; String

This constructor is used when the input is a data element with multiple occurrences, but only one occurrence is to be set, and it is represented as a String.

Synopsis

```
JoltInputEvent(Object source, String fieldName, String textValue,
int pos)
```

Parameters

source
Event source

fieldName
Name of the input field (Jolt name)

textValue
Value of the field as a String

pos
The position of the data element

Usage

This constructor is used when the input is a data element with multiple occurrences, but only one occurrence is to be set, and it is represented as a String. The JoltInputEvent object performs the necessary data conversion.

JoltInputEvent — multiple occurrences; array of Strings

This constructor is used when the input is a data element with multiple occurrences and it is represented as an array of Strings.

Synopsis

```
JoltInputEvent(Object source, String fieldName,
String textValues[])
```

Parameters

source
Event source

fieldName
Name of the input field (Jolt name)

textValues
Array of occurrences of the field as String

Usage

This constructor is used when the input is a data element with multiple occurrences and it is represented as an array of Strings. The JoltInputEvent object will perform the necessary data conversion.

JoltInputEvent Methods

The following methods are used in conjunction with the JoltInputEvent class:

- getValue
- getValues
- getTextValue
- getTextValues
- getFieldName
- getOccurrenceCount
- getSingleOccurrence
- isText
- isVector

getValue

The `getValue()` method gets the value of the field in its native representation.

Synopsis `Object` **`getValue`**`()`

Usage Get the value in its native representation.

Returns Value of the field.

getValues

The `getValues()` method gets the occurring values of the field in its native representation.

Synopsis `Object[]` **`getValues`**`()`

Usage Get the occurring values in its native representation.

Returns The occurrences of the field.

getTextValue

The `getTextValue()` method gets the value as a String.

Synopsis `String` **`getTextValue`**`()`

Usage Get the value as a String.

Returns The value of the field as a String.

getTextValues

The `getTextValues()` method gets the occurring values as String.

Synopsis `String[]` **`getTextValues`**`()`

Usage Get the occurring values as String.

Returns The occurrences of the field as a String

getFieldName

The `getFieldName()` method gets the name of the field.

Synopsis `String` **`getFieldName`**`()`

Usage Get the name of the field.

Returns The name of the field

getOccurrenceCount

The `getOccurrenceCount()` method gets the number of occurrences of a named item.

Synopsis `int` **`getOccurrenceCount`**`()`

Usage Get the number of occurrences of the field.

Returns The number of occurrences.

getSingleOccurrence

The `getSingleOccurrence()` method gets the position of the field.

Synopsis `int` **`getSingleOccurrence`**`()`

Usage Get the position of the field. These are:

- 0 for single data elements
- i for elements with multiple occurrences of which only one was set
- the array size for multiple occurrences

Returns Position

isText

The `isText()` method returns true if the field type is String, otherwise it returns false.

Synopsis `boolean` **`isText`**`()`

Returns True if the field type is String, otherwise it returns false.

isVector

The `isVector()` method retuns true if the field has multiple occurrences, otherwise it returns false.

Synopsis

```
boolean isVector()
```

Retuns

True if the field has multiple occurrences, otherwise it returns false.

JoltOutputEvent Class

```
java.lang.Object
   |
   +----java.util.EventObject
           |
           +----bea.jolt.beans.JoltOutputEvent
```

JoltOutputEvent is the event class for event objects sent from JoltServiceBeans to GUI elements when the reply from the TUXEDO service is received.

JoltOutputEvent Methods

The following methods are used in conjunction with the JoltOutputEvent class:

- isEventMessage
- getValue — value of a field
- getValue — value of one occurrence of a field
- getValues
- getTextValue — value of a field
- getTextValue — value of one occurrence of a field
- getTextValues

isEventMessage

The `isEventMessage()` method returns a boolean value that is true if this JoltOutputEvent has been constructed for a TUXEDO event. It is false if this JoltOutputEvent has been constructed for a service reply.

Synopsis `boolean` **`isEventMessage`**`()`

Usage Gets the flag indicating the origin of the event.

Returns True if this is a TUXEDO user event or false if this is a service reply.

getValue — value of a field

The `getValue()` method gets the value of a field in the output buffer.

Synopsis

```
Object getValue(String fieldName)
```

Parameters

fieldName
Name of the field (Jolt field name).

Usage

Gets the value of a field in the output buffer.

Returns

Value of the field in native data format.

getValue — value of one occurrence of a field

The `getValue()` method gets the value of one occurrence of a field in the output buffer.

Synopsis

```
Object getValue(String fieldName, int index)
```

Parameters

fieldName
Name of the field (Jolt field name).

index
The occurrence index of the field.

Usage

Gets the value of one occurrence of a field in the output buffer.

Returns

Value of the field in native data format

getValues

The `getValues()` method gets all occurrences of a field in the output buffer.

Synopsis

```
Object[] getValues(String fieldName)
```

Parameters

fieldName
Name of the field (Jolt field name)

Usage

Get all occurrences of a field in the output buffer.

Returns

Array of all occurrences in native data format.

getTextValue — value of a field

The `getTextValue()` method gets the value of a field in the output buffer as String.

Synopsis

```
String getTextValue(String fieldName)
```

Parameters

fieldName
Name of the field (Jolt field name)

Usage Get the value of a field in the output buffer.

Returns Value of the field as String.

getTextValue — value of one occurrence of a field

The `getTextValue()` method gets the value of one occurrence of a field in the output buffer.

Synopsis

```
String getTextValue(String fieldName, int index)
```

Parameters

fieldName
Name of the field (Jolt field name)

index
The occurrence index of the field.

Usage Get the value of one occurrence of a field in the output buffer.

Returns Value of the field as String.

getTextValues

The `getTextValues()` method gets all occurrences of a field in the JoltOutput message buffer.

Synopsis

```
String[] getTextValues(String fieldName)
```

Parameters

fieldName
Name of the field (Jolt field name)

Usage Get all occurrences of a field in the output buffer.

Returns Array of all occurrences as String array.

JoltServiceBean Class

```
java.lang.Object
   |
   +----bea.jolt.beans.JoltServiceBean
```

The JoltServiceBean represents a remote TUXEDO service. It listens to JoltInputEvents from other beans to populate its input buffer. JoltServiceBean offers the callService() method to invoke the service.

JoltServiceBean is an event source for JoltOutputEvents that carry information about output of the service. After a successful callService() listener beans are notified via a JoltOutputEvent that carries the output buffer.

Note: All getOutputValue/getOutputTextValue methods operate on output parameters only. All setInputValue/setInputTextValue methods operate on input parameters only.

JoltServiceBean Constructor

The JoltServiceBean constructor creates a bean representing a Jolt service.

JoltServiceBean

This constructor creates an instance of the JoltServiceBean class.

Synopsis **`JoltServiceBean`**`()`

See Also JoltInputEvent, JoltOutputEvent

JoltServiceBean Methods

The following methods are used in conjunction with the JoltServiceBean class:

- propertyChange
- dataChanged

- setServiceName
- getServiceName
- isTransactional
- setTransactional
- setSession
- getSession
- getOutputValue — value of field
- getOutputValue — value of one occurrence of field
- getOutputValues
- getOutputTextValue — value of field
- getOutputTextValue — value of one occurrence of field
- getOutputTextValues
- setInputValue — value of field
- setInputValue — value of one occurrence of field
- setInputValues
- setInputTextValue — value of field
- setInputTextValue — value of one occurrence of field
- setInputTextValues
- getOccurrenceCount
- clear
- callService
- addJoltOutputListener
- removeJoltOutputListener

propertyChange

The `propertyChange()` method is the event handler for PropertyChange events.

Synopsis
```
void propertyChange(PropertyChangeEvent evt)
```

Parameters
evt
The event object.

Usage Event handler for PropertyChange events. The JoltSessionBean notifies the JoltServiceBean when it logs on and off by raising a PropertyChangeEvent about its LoggedOn property. The logoff of a session, other than the current session, does not affect the JoltServiceBean. This method should not be called directly.

dataChanged

The `dataChanged()` method is the event handler method for JoltInputEvents. The JoltServiceBean updates the input buffer field specified in the JoltInputEvent to the value specified in the event.

Synopsis
```
void dataChanged(JoltInputEvent evt)
```

Parameters
evt
The event object.

Usage Event handler for JoltInputEvents. This method should not be called directly.

setServiceName

The `setServiceName()` method sets the name of the remote service that this bean represents.

Synopsis
```
void setServiceName(String name)
```

Parameters
name
Service name

Usage Sets the name of the remote service that this bean represents.

getServiceName

The `getServiceName()` method gets the name of the remote service that this bean represents.

Synopsis
```
String getServiceName()
```

Usage
Gets the name of the remote service that this bean represents.

Returns
The name of the TUXEDO service.

isTransactional

The `isTransactional()` method returns true if the bean is in transactional mode, otherwise it returns false.

Synopsis
```
boolean isTransactional()
```

Usage
Set to true if this JoltServiceBean is to be included in the transaction that was started by its JoltSessionBean.

Returns
True or false.

setTransactional

The `setTransactional()` method sets the transactional mode of the bean.

Synopsis
```
void setTransactional(boolean mode)
```

Parameters
mode
True or false

Usage
Set to true if this JoltServiceBean is to be included in the transaction that was started by its JoltSessionBean.

setSession

The `setSession()` method is used in cases when the JoltServiceBean is created after the logon event. Otherwise, the JoltServiceBean gets access to a TUXEDO client session by listening to JoltSessionEvents.

Synopsis
```
void setSession(JoltSessionBean value)
```

Parameters
value
The JoltSessionBean that is used by this service bean

Usage Sets the JoltSessionBean associated with this JoltServiceBean.

getSession

The `getSession()` method gets the JoltSessionBean used by this JoltServiceBean.

Synopsis
```
JoltSessionBean getSession()
```

Usage Gets the JoltSessionBean used by this JoltServiceBean.

Returns JoltSessionBean

getOutputValue — value of field

The `getOutputValue()` method gets the value of a field in the output buffer using the field's native type.

Synopsis
```
Object getOutputValue(String fieldName)
```

Parameters
fieldName
Name of the field.

Usage Gets the value of a field in the output buffer using the field's native type.

Returns Value of the field.

Throws NoSuchFieldError

getOutputValue — value of one occurrence of field

The `getOutputValue()` method gets the value of one occurrence of a field in the output buffer using the field's native type.

Synopsis

```
Object getOutputValue(String fieldName, int index)
```

Parameters

fieldName
Name of the field.

index
Index of the field.

Usage Gets the value of one occurrence in the output buffer using the field's native type.

Returns Value of the field.

Throws NoSuchFieldError

getOutputValues

The `getOutputValues()` method gets all of the occurrences of a field in the output buffer using field's native type.

Synopsis

```
Object[] getOutputValues(String fieldName)
```

Parameters

fieldName
The name of the field.

Usage Gets all of the occurrences of a field in the output buffer using field's native type.

Returns The occurrences of the field.

Throws NoSuchFieldError

getOutputTextValue — value of field

The `getOutputTextValue()` method gets the value of a field in the output buffer as a String.

Synopsis

```
String getOutputTextValue(String fieldName)
```

Parameters

fieldName
The name of the field

Usage Gets the value of a field in the output buffer as a `String`.

Returns The value of the field as a String.

Throws NoSuchFieldError

getOutputTextValue — value of one occurrence of field

The `getOutputTextValue()` method gets the value of one occurrence of a field in the output buffer as a String.

Synopsis

```
String getOutputTextValue(String fieldName, int index)
```

Parameters

fieldName
The name of the field.

index
Index of the field.

Usage Gets the value of one occurrence of a field in the output buffer as a `String`.

Returns The value of the field as a String.

Throws NoSuchFieldError

getOutputTextValues

The `getOutputTextValues()` method gets all the occurrences of a field in the output buffer as String.

Synopsis `String[] getOutputTextValues(String fieldName)`

Parameters

`fieldName`
Name of the field.

Usage Gets all the occurrences of a field in the output buffer as `String`.

Returns The occurrences of the field as a String.

Throws NoSuchFieldError

setInputValue — value of field

The `setInputValue()` method sets the value of a field in the input buffer using the field's native type.

Synopsis `void setInputValue(String fieldName, Object value)`

Parameters

`fieldName`
The name of the field.

`value`
Value to be set.

Usage Sets the value of a field in the input buffer using the field's native type.

Throws NoSuchFieldError, ClassCastException

setInputValue — value of one occurrence of field

The `setInputValue()` method sets the value of one occurrence of a field in the input buffer using the field's native type.

Synopsis

```
void setInputValue(String fieldName, int index, Object value)
```

Parameters

fieldName
The name of the field.

index
Index of the field.

value
Value to be set.

Usage Sets the value of one occurrence of a field in the input buffer using the field's native type.

Throws NoSuchFieldError, ClassCastException

setInputValues

The `setInputValues()` method sets all the occurrences of a field in the input buffer using the field's native type.

Synopsis

```
void setInputValues(String fieldName, Object values[])
```

Parameters

fieldName
The name of the field

values
The value to set.

Usage Sets all the occurrences of a field in the input buffer using the field's native type. This method operates using input parameters only.

Throws NoSuchFieldError, ClassCastException

setInputTextValue — value of field

The `setInputTextValue()` method sets the value of a field in the input buffer as a String.

Synopsis

```
void setInputTextValue(String fieldName, String textValue)
```

Parameters

fieldName
The name of the field.

textValue
The value to set.

Usage Sets the value of a field in the input buffer as a `String`.

Throws NoSuchFieldError, NumberFormatException

setInputTextValue — value of one occurrence of field

The `setInputTextValue()` method sets the value of one occurrence of a field in the input buffer as a String.

Synopsis

```
void setInputTextValue(String fieldName, int index,
String textValue)
```

Parameters

fieldName
The name of the field.

index
Index of the field.

textValue
The value to set.

Usage Sets the value of one occurrence of a field in the input buffer as a `String`.

Throws NoSuchFieldError, NumberFormatException

setInputTextValues

The `setInputTextValues()` method sets all the occurrences of a field in the input buffer as String.

Synopsis
```
void setInputTextValues(String fieldName, String textValues[])
```

Parameters

fieldName
The name of the field.

textValues
The values to set.

Usage Sets all the occurrences of a field in the input buffer as `String`.

Throws NoSuchFieldError, ClassCastException

getOccurrenceCount

The `getOccurrenceCount()` method gets the number of occurrences of a field from the output buffer.

Synopsis
```
int getOccurrenceCount(String fieldName)
```

Parameters

fieldName
The name of the field.

Usage Gets the number of occurrences of a field from the output buffer.

Returns The number of occurrences.

Throws NoSuchFieldError

clear

Synopsis
```
void clear()
```

Usage This method clears the underlying input and output buffers.

callService

The `callService()` method invokes the remote service.

Synopsis
```
void callService()
```

Usage
Invokes the remote service.

Throws
ServiceException, TransactionException, ApplicationException

addJoltOutputListener

The `addJoltOutputListener()` method adds a JoltOutputEvent listener.

Synopsis
```
synchronized void addJoltOutputListener(JoltOutputListener listener)
```

Parameters
listener
The event listener to be added.

Usage
Adds a JoltOutputEvent listener.

removeJoltOutputListener

The `removeJoltOutputListener()` method removes a JoltOutputEvent listener.

Synopsis
```
synchronized void removeJoltOutputListener(JoltOutputListener listener)
```

Parameters
listener
The event listener to be removed.

Usage
Removes a JoltOutputEvent listener.

JoltSessionBean Class

```
java.lang.Object
   |
   +----bea.jolt.beans.JoltSessionBean
```

The JoltSessionBean represents the TUXEDO session. It includes the functionality of JoltSession, JoltSessionAttributes, and JoltTransaction classes. The JoltSessionBean offers properties to set session attributes and methods that open and close a TUXEDO session. It also sends a `propertyChange` event for the `LoggedOn` property when the TUXEDO session is established. In addition, the JoltSessionBean provides methods to control transactions. The JoltSessionBean is an event source for JoltOutputEvents. These events are sent if an unsolicited message or TUXEDO user event notification is sent.

JoltSessionBean Constructor

The `JoltSessionBean` constructor creates an instance of the JoltSessionBean.

JoltSessionBean

Synopsis `JoltSessionBean()`

JoltSessionBean Methods

The following methods are used in conjunction with the JoltSessionBean class:

- addJoltOutputListener
- removeJoltOutputListener
- isLoggedOn
- addPropertyChangeListener
- removePropertyChangeListener

- logon
- logoff
- clear
- beginTransaction
- commitTransaction
- rollbackTransaction
- isInTransaction
- getAppAddress
- setAppAddress
- getIdleTimeOut
- setIdleTimeOut
- setReceiveTimeOut
- getReceiveTimeOut
- setSendTimeOut
- getSendTimeOut
- getSessionTimeOut
- setUserName
- getUserName
- setUserRole
- getUserRole
- setUserPassword
- getUserPassword
- setAppPassword
- getAppPassword

addJoltOutputListener

The `addJoltOutputListener()` method adds a JoltOutputEvent listener.

Synopsis
```
synchronized void addJoltOutputListener(JoltOutputListener listener)
```

Parameters
listener
Event listener to be added.

Usage
Adds a JoltOutputEvent listener.

removeJoltOutputListener

The `removeJoltOutputListener()` method removes a JoltOutputEvent listener.

Synopsis
```
synchronized void removeJoltOutputListener(JoltOutputListener listener)
```

Parameters
listener
Event listener to be removed.

Usage
Removes a JoltOutputEvent listener.

isLoggedOn

The `isLoggedOn()` method determines if the session exists.

Synopsis
```
boolean isLoggedOn()
```

Usage
Determines if the session exists.

Returns
True or false.

addPropertyChangeListener

The specified PropertyChangeListener's `propertyChange()` method is called each time the value of the LoggedOn property changes.

Synopsis
```
void addPropertyChangeListener(PropertyChangeListener l)
```

Parameters

l

The PropertyChangeListener.

Usage The specified PropertyChangeListener's `propertyChange()` method is called each time the value of the loggedOn property changes.

removePropertyChangeListener

The `removePropertyChangeListener()` method removes the PropertyChangeListener from the internal list.

Synopsis
```
void removePropertyChangeListener(PropertyChangeListener l)
```

Parameters

l

The PropertyChangeListener.

Usage Removes the PropertyChangeListener from the internal list.

logon

The `logon()` method opens a new session to TUXEDO.

Synopsis
```
void logon()
```

Usage Opens a new session to TUXEDO. If a session is open, it throws a SessionException. Upon successful logon, the LoggedOn property value changes to true.

Throws SessionException

logoff

The `logoff()` method closes the session to TUXEDO.

Synopsis `void` **`logoff`**`()`

Usage Closes the session to TUXEDO. Session attributes are not cleared after logoff. The LoggedOn property value changes to false.

Throws SessionException

clear

The `clear()` method resets all session attributes.

Synopsis `void` **`clear`**`()`

Usage Resets all session attributes.

Throws SessionException

beginTransaction

The `beginTransaction()` method starts the transaction.

Synopsis `void` **`beginTransaction`**`(int` *`timeout`*`)`

Parameters *`timeout`*
Transaction timeout

Usage Starts a transaction. If this method is called twice without commitTransaction or abortTransaction, a TransactionException is raised.

Throws TransactionException

commitTransaction

The `commitTransaction()` method commits the transaction.

Synopsis `void` **`commitTransaction`**`()`

Usage Commits the transaction.

Throws TransactionException

rollbackTransaction

The `rollbackTransaction()` method aborts the transaction.

Synopsis
```
void rollbackTransaction()
```

Usage Aborts the transaction.

Throws TransactionException

isInTransaction

The `isInTransaction()` method returns true or false depending on whether there is a started transaction.

Synopsis
```
boolean isInTransaction()
```

Returns True if in a transaction, that is a transaction has started and not committed or aborted, and false otherwise.

getAppAddress

The `getAppAddress()` method gets the IP address (host name and port number) of the JSL or the Jolt Relay.

Synopsis
```
String getAppAddress()
```

Usage Gets the IP address (host name and port number) of the JSL or the Jolt Relay.

Returns The APPADDRESS.

setAppAddress

The `setAppAddress()` method sets the IP address (host name and port number) of the JSL or the Jolt Relay.

Synopsis
```
void setAppAddress(String value)
```

Parameters *value*

The IP address (`//host:port`).

Usage Sets the IP address (host name and port number) of the JSL or the Jolt Relay.

getIdleTimeOut

The `getIdleTimeOut()` method gets the IDLETIMEOUT attribute.

Synopsis `int` **`getIdleTimeOut`**`()`

Usage Gets the IDLETIMEOUT attribute.

Returns Timeout value.

setIdleTimeOut

The `setIdleTimeOut()` method sets the IDLETIMEOUT attribute.

Synopsis `void` **`setIdleTimeOut`**`(int` *`value`*`)`

Parameters *`value`*
The new IDLETIMEOUT value.

Usage Sets the IDLETIMEOUT attribute.

setReceiveTimeOut

The `setReceiveTimeOut()` method sets the RECVTIMEOUT attribute.

Synopsis `void` **`setReceiveTimeOut`**`(int` *`value`*`)`

Parameters *`value`*
The new RECVTIMEOUT value.

Usage Sets the RECVTIMEOUT attribute.

getReceiveTimeOut

The `getReceiveTimeOut()` method gets the RECVTIMEOUT attribute.

Synopsis `int` **`getReceiveTimeOut`**`()`

Usage Gets the RECVTIMEOUT attribute.

Returns Receive timeout value.

setSendTimeOut

The `setSendTimeOut()` method sets the SENDTIMEOUT attribute.

Synopsis
```
void setSendTimeOut(int value)
```

Parameters
value
The new SENDTIMEOUT value.

Usage
Sets the SENDTIMEOUT attribute.

getSendTimeOut

The `getSendTimeOut()` method gets the SENDTIMEOUT attribute.

Synopsis
```
int getSendTimeOut()
```

Usage
Gets the SENDTIMEOUT attribute.

Returns
Send timeout value.

getSessionTimeOut

The `getSessionTimeOut()` method gets the SESSIONTIMEOUT attribute.

Synopsis
```
String getSessionTimeOut()
```

Usage
Gets the SESSIONTIMEOUT attribute.

Returns
Session timeout value.

setUserName

The `setUserName()` method sets the TUXEDO user name to be used at logon.

Synopsis
```
void setUserName(String value)
```

Parameters
value
TUXEDO user name.

Usage
Sets the TUXEDO user name to be used at logon.

getUserName

The `getUserName()` method gets the TUXEDO user name to be used at logon.

Synopsis
```
String getUserName()
```

Usage Gets the TUXEDO user name to be used at logon.

Returns TUXEDO user name.

setUserRole

The `setUserRole()` method sets the TUXEDO user role to be used at logon.

Synopsis
```
void setUserRole(String value)
```

Parameters
value
TUXEDO user role.

Usage Sets the TUXEDO user role to be used at logon.

getUserRole

The `getUserRole()` method gets the TUXEDO user role to be used at logon.

Synopsis
```
String getUserRole()
```

Usage Gets the TUXEDO user role to be used at logon.

Returns TUXEDO user role.

setUserPassword

The `setUserPassword()` method sets the TUXEDO user password to be used at logon.

Synopsis
```
void setUserPassword(String value)
```

Parameters
value
TUXEDO user password.

Usage Sets the TUXEDO user password to be used at logon.

getUserPassword

The `getUserPassword()` method gets the TUXEDO user password to be used at logon.

Synopsis

```
String getUserPassword()
```

Usage Gets the TUXEDO user password to be used at logon.

Returns TUXEDO user password

setAppPassword

The `setAppPassword()` method sets the TUXEDO application password to be used at logon.

Synopsis

```
void setAppPassword(String value)
```

Parameters

value
TUXEDO application password.

Usage Sets the TUXEDO application password to be used at logon.

getAppPassword

The `getAppPassword()` method gets the TUXEDO application password to be used at logon.

Synopsis

```
String getAppPassword()
```

Usage Gets the TUXEDO application password to be used at logon.

Returns TUXEDO application password.

JoltUserEventBean Class

```
java.lang.Object
   |
   +----bea.jolt.beans.JoltUserEventBean
```

The JoltUserEventBean provides access to TUXEDO events. The TUXEDO event to be subscribed to is defined by setting the appropriate parameters of this bean.

JoltUserEventBean Constructor

The JoltUserEventBean constructor creates an instance of the JoltUserEventBean which can be used to subscribe to a TUXEDO event.

JoltUserEventBean

This constructor subscribes to a specific TUXEDO event or notification.

Synopsis `JoltUserEventBean()`

JoltUserEventBean Methods

The following methods are used in conjunction with the JoltUserEventBean class:

- propertyChange
- setEventName
- getEventName
- setFilter
- getFilter
- setSession
- getSession

- unsubscribe
- subscribe

propertyChange

The `propertyChange()` method is the event handler for PropertyChange events.

Synopsis
```
void propertyChange(PropertyChangeEvent evt)
```

Parameters

evt

The event object.

Usage

Event handler for PropertyChange events. The JoltSessionBean notifies the JoltUserEventBean when it logs on and off by raising a PropertyChangeEvent about its LoggedOn property. The logoff of a session, other than the current session, does not affect the JoltUserEventBean. This method should not be called directly.

setEventName

The `setEventName()` method sets the regular expression of the user event that this bean represents.

Synopsis
```
void setEventName(String name)
```

Parameters

name

The regular expression of the user event.

Usage

Sets the regular expression of the user event that this bean represents. If the client wants to only subscribe to unsolicited events, use setEventName (`"\\.UNSOLMSG"`).

getEventName

The `getEventName()` method gets the regular expression this bean represents.

Synopsis
```
String getEventName()
```

Usage

Gets the regular expression this bean represents.

setFilter

The `setFilter()` method sets the event filter.

Synopsis: void **`setFilter`**(String *`filter`*)

Parameters:

`filter`
The event filter.

Usage: Sets the event filter.

getFilter

The `getFilter()` method gets the event filter.

Synopsis: `String` **`getFilter`**`()`

Usage: Gets the event filter.

Returns: Regular expression

setSession

The `setSession()` method is used when the JoltUserEventBean is created after the logon event.

Synopsis: `void` **`setSession`**`(JoltSessionBean` *`value`*`)`

Parameters:

`value`
JoltSessionBean

Usage: This method is used in cases when the JoltUserEventBean is created after the logon event. Otherwise, the JoltUserEventBean can get access to a TUXEDO client session by listening to propertyChangeEvents.

getSession

The `getSession()` method gets the JoltSessionBean used by this JoltUserEventBean.

Synopsis: `JoltSessionBean` **`getSession`**`()`

Usage: Gets the JoltSessionBean used by this JoltUserEventBean.

Returns: JoltSessionBean

unsubscribe

The `unsubscribe()` method deletes the subscription to an event.

Synopsis `void` **`unsubscribe`**`()`

Usage This method is used to stop subcribing to the event represented by the JoltUserEventBean.

Throws `EventException`, `SessionException`, `JoltException`

subscribe

The `subscribe()` method activates the subscription to an event.

Synopsis `void` **`subscribe`**`()`

Usage Activates the subscription to an event.

Throws `EventException`, `SessionException`, `JoltException`

5 Jolt Aware AWT Beans Class Library Reference

The Jolt aware AWT Beans Class Library provides a Jolt enabled versions of standard AWT components, such as TextField, Label, and List. Using the Jolt aware AWT Beans, you can develop Jolt GUIs with minimal coding.

For information about the JoltBeans toolkit, refer to Chapter 4, "JoltBeans Toolkit Class Library Reference."

The bea.jolt.beans.awt package includes the following classes:

- JoltCheckbox Class
- JoltChoice Class
- JoltLabel Class
- JoltList Class
- JoltTextField Class

JoltCheckbox Class

```
java.lang.Object
   |
   +----java.awt.Component
            |
            +----java.awt.Checkbox
                    |
                    +----bea.jolt.beans.awt.JoltCheckbox
```

The JoltCheckbox class is a Jolt aware extension of java.awt.Checkbox, that is linked to a specific field in the input buffer by its JoltFieldName property. If the field occurs multiple times, the occurrence this checkbox is linked to is specified by the occurrenceIndex property of this bean.

It can be connected with JoltServiceBeans to contain parts of the input for a service. A JoltServiceBean may listen to events raised by a JoltCheckbox. JoltCheckbox sends JoltInputEvents to its listeners (typically JoltServiceBeans) when the selection in the checkbox changes. The JoltInputEvent in this case is populated with the `TrueValue` property of data type String (if the box is selected) or `FalseValue` (if the box is unselected).

JoltCheckbox Constructor

Synopsis `JoltCheckbox()`

JoltCheckbox Methods

The following methods are used in conjunction with the JoltCheckbox class:

- addJoltInputListener
- removeJoltInputListener
- setOccurrenceIndex
- getOccurrenceIndex

- getJoltFieldName
- setJoltFieldName
- getTrueValue
- setTrueValue
- getFalseValue
- setFalseValue

addJoltInputListener

The `addJoltInputListener()` method registers JoltInput listeners.

Synopsis

```
synchronized void addJoltInputListener(JoltInputListener listener)
```

Parameters

`listener`
The listener to be added.

Usage

Registers JoltInput listeners.

removeJoltInputListener

The `removeJoltInputListener()` method unregisters JoltInput listeners.

Synopsis

```
synchronized void removeJoltInputListener(JoltInputListener
listener)
```

Parameters

`listener`
The listener to be removed.

Usage

Unregisters JoltInput listeners.

setOccurrenceIndex

The `setOccurrenceIndex()` method sets the occurrence index of the field represented by this JoltCheckbox.

Synopsis `void` **`setOccurrenceIndex`**`(int` *`occurrence`*`)`

Parameters *`occurrence`*
The occurrence number.

Usage Sets the occurrence index of the field represented by this JoltCheckbox.

getOccurrenceIndex

The `getOccurrenceIndex()` method gets the occurrence index of the field represented by this JoltCheckbox.

Synopsis `int` **`getOccurrenceIndex`**`()`

Usage Get the occurrence index of the field represented by this JoltCheckbox.

getJoltFieldName

The `getJoltFieldName()` method gets the Jolt field name corresponding to this JoltCheckbox.

Synopsis `String` **`getJoltFieldName`**`()`

Usage Gets the Jolt field name corresponding to this JoltCheckbox.

Returns the Jolt field name

setJoltFieldName

The `setJoltFieldName()` method sets the Jolt field name corresponding to this JoltCheckbox.

Synopsis `void` **`setJoltFieldName`**`(String` *`name`*`)`

Parameters *`name`*
The Jolt field name.

Usage Sets the Jolt field name corresponding to this JoltCheckbox.

getTrueValue

The `getTrueValue()` method returns the String value that is represented by this JoltCheckbox if the box is checked.

Synopsis
```
String getTrueValue()
```

Usage
Gets the field value represented by this JoltCheckbox if the box is checked.

Returns
The true value.

setTrueValue

The `setTrueValue()` method sets the String value that is represented by this JoltCheckbox if the box is checked.

Synopsis
```
void setTrueValue(String value)
```

Parameters
value
The true value.

Usage
Sets the field value that is represented by this JoltCheckbox if the box is checked.

getFalseValue

The `getFalseValue()` method gets the field value represented by this JoltCheckbox if the box is unchecked.

Synopsis
```
String getFalseValue()
```

Usage
Gets the field value if the box is unchecked.

Returns
The false value represented by this JoltCheckbox.

setFalseValue

The `setFalseValue()` method sets the field value that is represented by this JoltCheckbox if the box is unchecked.

Synopsis

```
void setFalseValue(String value)
```

Parameters

value

The false value

Usage

Sets the field value that is represented by this JoltCheckbox if the box is unchecked.

JoltChoice Class

```
java.lang.Object
   |
   +----java.awt.Component
           |
           +----java.awt.Choice
                    |
                    +----bea.jolt.beans.awt.JoltChoice
```

The JoltChoice class provides a Jolt aware extension of java.awt.Choice that is linked to a specific field in the input buffer by its JoltFieldName property. If the field occurs multiple times, the occurrence this choice is linked to is specified by the occurrenceIndex property of this bean.

It can be connected with JoltServiceBeans to contain parts of the input for a service. A JoltServiceBean may listen to events raised by a JoltChoice. JoltChoice sends JoltInputEvents to its listeners (typically JoltServiceBeans) when the selection in the choice changes. The JoltInputEvent in this case is populated with the single value of the selected item.

JoltChoice Constructor

Synopsis `JoltChoice()`

JoltChoice Methods

The following methods are used in conjunction with the JoltChoice class:

- addJoltInputListener
- removeJoltInputListener
- setOccurrenceIndex
- getOccurrenceIndex
- getJoltFieldName

- setJoltFieldName
- getItems
- setItems

addJoltInputListener

The `addJoltInputListener()` method registers JoltInput listeners.

Synopsis
```
synchronized void addJoltInputListener(JoltInputListener listener)
```

Parameters
listener
The listener to be added.

Usage
Registers JoltInput listeners.

removeJoltInputListener

The `removeJoltInputListener()` method unregisters JoltInput listeners.

Synopsis
```
synchronized void removeJoltInputListener(JoltInputListener
listener)
```

Parameters
listener
The listener to be removed.

Usage
Unregisters JoltInput listeners.

setOccurrenceIndex

The `setOccurrenceIndex()` method sets the occurrence index of the field represented by this JoltChoice.

Synopsis
```
void setOccurrenceIndex(int occurrence)
```

Parameters
occurrence
The occurrence number.

Usage
Sets the occurrence index of the field represented by this JoltChoice.

getOccurrenceIndex

The `getOccurrenceIndex()` method gets the occurrence index of the field represented by this JoltChoice.

Synopsis `int` **`getOccurrenceIndex`**`()`

Usage Gets the occurrence index of the field represented by this JoltChoice.

getJoltFieldName

The `getJoltFieldName()` method gets the field name corresponding to the field represented by this JoltChoice.

Synopsis `String` **`getJoltFieldName`**`()`

Usage Gets the Jolt field name corresponding to this choice.

Returns The Jolt field name

setJoltFieldName

The `setJoltFieldName()` method sets the field name (Jolt field name) corresponding to this Choice.

Synopsis `void` **`setJoltFieldName`**`(String` *`name`*`)`

Parameters *`name`*

The Jolt field name.

Usage Sets the Jolt field name corresponding to the field represented by this JoltChoice.

getItems

The `getItems()` method gets the selected items corresponding to the field represented by this JoltChoice.

Synopsis `String[]` **`getItems`**`()`

Usage Gets the selected items corresponding to this choice.

Returns The array of selected items.

setItems

The `setItems()` method sets the selected items corresponding to the field represented by this JoltChoice.

Synopsis

```
void setItems(String values[])
```

Parameters

name

The Jolt field name

Usage

Sets the selected items corresponding to this JoltChoice.

JoltLabel Class

```
java.lang.Object
   |
   +----java.awt.Component
           |
           +----java.awt.Label
                    |
                    +----bea.jolt.beans.awt.JoltLabel
```

A Jolt aware extension of java.awt.Label that is linked to a specific field in the output buffer by its JoltFieldName property. If the field occurs multiple times, the occurrence this label is linked to is specified by the occurrenceIndex property of this bean. JoltLabel can be connected with JoltServiceBeans to display output from a service. A JoltLabel listens to JoltOutputEvents from JoltServiceBeans and updates its contents according to the occurrences of the field to which it is linked.

JoltLabel Constructor

```
JoltLabel()
```

JoltLabel Methods

The following methods are used in conjunction with the JoltLabel class:

- serviceReturned
- setOccurrenceIndex
- getOccurrenceIndex
- getJoltFieldName
- setJoltFieldName

serviceReturned

The `serviceReturned()` method is the event handler for JoltOutputEvents.

Synopsis `void` **`serviceReturned`**`(JoltOutputEvent` *`evt`*`)`

Parameters *`evt`*
The event object.

Usage Event handler for JoltOutputEvents. This should not be called directly; it is always called by the JoltServiceBean.

setOccurrenceIndex

The `setOccurrenceIndex()` method sets the occurrence index of the field represented by this JoltLabel.

Synopsis `void` **`setOccurrenceIndex`**`(int` *`occurrence`*`)`

Parameters *`occurrence`*
The occurrence number.

Usage Sets the occurrence index of this field.

getOccurrenceIndex

The `getOccurrenceIndex()` method gets the occurrence index of the field represented by this JoltLabel.

Synopsis `int` **`getOccurrenceIndex`**`()`

Usage Gets the occurrence index of this field.

getJoltFieldName

The `getJoltFieldName()` method gets the Jolt field name corresponding to this Label.

Synopsis `String` **`getJoltFieldName`**`()`

Usage Gets the Jolt field name corresponding to this JoltLabel.

Returns The Jolt field name.

setJoltFieldName

The `setJoltFieldName()` method sets the Jolt field name corresponding to this Label.

Synopsis

```
void setJoltFieldName(String name)
```

Parameters

name

The Jolt field name

Usage

Sets the Jolt field name corresponding to this JoltLabel.

JoltList Class

```
java.lang.Object
   |
   +----java.awt.Component
            |
            +----java.awt.List
                    |
                    +----bea.jolt.beans.awt.JoltList
```

A Jolt aware extension of java.awt.TextField that is linked to a specific field in the Jolt input or output buffer by its JoltFieldName property. If the field occurs multiple times, the occurrence this textfield is linked to is specified by the occurrenceIndex property of this bean.

JoltTextField can be connected with JoltServiceBeans in two ways:

- JoltTextField contains parts of the input for a service. A JoltServiceBean may listen to events raised by a JoltTextField. JoltTextField sends JoltInputEvents to its listeners (typically JoltServiceBeans) when its contents changes.
- JoltTextField displays output from a service. In this case, JoltTextField listens to JoltOutputEvents from JoltServiceBeans and updates its contents according to the occurrence of the field to which it is linked.

JoltList Methods

The following methods are used in conjunction with the JoltList class:

- serviceReturned
- addJoltInputListener
- removeJoltInputListener
- getJoltFieldName
- setJoltFieldName
- setOccurrenceIndex
- getOccurrenceIndex

serviceReturned

The `serviceReturned()` method is the event handler for JoltOutputEvents.

Synopsis `void` **`serviceReturned`**`(JoltOutputEvent` *`evt`*`)`

Parameters *`evt`*
The event object.

Usage Event handler for JoltOutputEvents. This method should not be called directly. It is always called by the JoltServiceBean.

addJoltInputListener

The `addJoltInputListener()` method registers JoltInput listeners.

Synopsis synchronized void **`addJoltInputListener`**(JoltInputListener *`l`*)

Parameters *`l`*
The listener to be added.

Usage Registers JoltInput listeners.

removeJoltInputListener

The `removeJoltInputListener()` method unregisters JoltInput listeners.

Synopsis `synchronized void` **`removeJoltInputListener`**`(JoltInputListener` *`l`*`)`

Parameters *`l`*
The listener to be removed

Usage Unregisters JoltInput listeners.

getJoltFieldName

The `getJoltFieldName()` method gets the Jolt field name corresponding to this JoltList.

Synopsis `String` **`getJoltFieldName`**`()`

Usage Gets the Jolt field name corresponding to this JoltList.

Returns The Jolt field name.

setJoltFieldName

The `setJoltFieldName()` method sets the Jolt field name corresponding to this JoltList.

Synopsis

```
void setJoltFieldName(String name)
```

Parameters

name
The Jolt field name.

Usage

Sets the Jolt field name corresponding to this JoltList.

setOccurrenceIndex

The `setOccurrenceIndex()` method sets the occurrence index of the field represented by this JoltList.

Synopsis

```
void setOccurrenceIndex(int occurrence)
```

Parameters

occurrence
The occurrence number.

Usage

Sets the occurrence index of this field.

getOccurrenceIndex

The `getOccurrenceIndex()` method gets the occurrence index of the field represented by this JoltList.

Synopsis

```
int getOccurrenceIndex()
```

Usage

Gets the occurrence index of this field.

JoltTextField Class

```
java.lang.Object
   |
   +----java.awt.Component
           |
           +----java.awt.TextComponent
                    |
                    +----java.awt.TextField
                             |
                             +----bea.jolt.beans.awt.JoltTextField
```

A Jolt aware extension of java.awt.TextField that is linked to a specific field in the Jolt buffer by its JoltFieldName property. If the field occurs multiple times, the occurrence this textfield is linked to is specified by the occurrenceIndex property of this bean.

JoltTextField can be connected with JoltServiceBeans in two ways:

- JoltTextField contains parts of the input for a service. A JoltServiceBean may listen to events raised by a JoltTextField. JoltTextField sends JoltInputEvents to its listeners (typically JoltServiceBeans) when its contents changes.
- JoltTextField displays output from a service. When used to display the output of a service, JoltTextField listens to JoltOutputEvents from JoltServiceBeans and updates its contents according to the occurrences of the field to which it is linked.

JoltTextField Constructor

The `JoltTextField` constructor creates an instance of the JoltTextField.

JoltTextField Methods

The following methods are used in conjunction with the JoltTextField class:

- addJoltInputListener
- addJoltInputListener
- removeJoltInputListener
- serviceReturned
- getJoltFieldName
- setJoltFieldName
- setOccurrenceIndex
- getOccurrenceIndex

addJoltInputListener

The `addJoltInputListener()` method registers JoltInput listeners.

Synopsis

```
synchronized void addJoltInputListener(JoltInputListener l)
```

Parameters

l

The listener to be added

Usage

Registers JoltInput listeners.

removeJoltInputListener

The `removeJoltInputListener()` method unregisters JoltInput listeners.

Synopsis

```
synchronized void removeJoltInputListener(JoltInputListener l)
```

Parameters

l

The listener to be removed.

Usage

Unregisters JoltInput listeners.

serviceReturned

The `serviceReturned()` method is the handler method for JoltOutputEvents.

Synopsis `void` **`serviceReturned`**`(JoltOutputEvent` *`evt`*`)`

Parameters *`evt`*
The event object.

Usage Handler method for JoltOutputEvents. This method should not be called directly. It is always called by the JoltServiceBean.

getJoltFieldName

The `getJoltFieldName()` method gets the field name corresponding to this JoltTextField.

Synopsis `String` **`getJoltFieldName`**`()`

Usage Gets the Jolt field name corresponding to this JoltTextField.

Returns The Jolt field name.

setJoltFieldName

The `setJoltFieldName()` method sets the field name corresponding to this JoltTextField.

Synopsis `void` **`setJoltFieldName`**`(String` *`name`*`)`

Parameters *`name`*
The Jolt field name.

Usage Sets the Jolt field name corresponding to this JoltTextField.

setOccurrenceIndex

The `setOccurrenceIndex()` method sets the occurrence index of the field represented by this JoltTextField.

Synopsis

```
void setOccurrenceIndex(int occurrence)
```

Parameters

occurrence
The occurrence number.

Usage

Sets the occurrence index of this field.

getOccurrenceIndex

The `getOccurrenceIndex()` method gets the occurrence index of the field represented by this JoltTextField.

Synopsis

```
int getOccurrenceIndex()
```

Usage

Gets the occurrence index of this field.