BEA WebLogic Integration Adapter for Baan®

User Guide

Version 8.1.1
Document Date: October 2003

Copyright

Restricted Rights Legend

Trademarks or Service Marks

Contents

About This Document

1. Introducing the BEA WebLogic Adapter for Baan

2. Introducing Schemas for Baan Integration Objects

3. Defining Application Views for Baan

Index

About This Document

This document describes how to use the BEA WebLogic Adapter for Baan. This document is organized as follows:

- Chapter 1, "Introducing the BEA WebLogic Adapter for Baan," describes the adapter, and how it relates to both Baan business objects and WebLogic Integration.
- Chapter 2, "Introducing Schemas for Baan Integration Objects," describes the schemas for your Baan business objects.
- Chapter 3, "Defining Application Views for Baan," describes application views and how to use them to configure events and services.

Who Should Read This Documentation

This document is intended for the following members of an integration team:

- Integration Specialists—Lead the integration design effort. Integration specialists have expertise in defining the business and technical requirements of integration projects, and in designing integration solutions that implement specific features of WebLogic Integration. The skills of integration specialists include business and technical analysis, architecture design, project management, and WebLogic Integration product knowledge.
- Technical Analysts—Provide expertise in an organization's information technology infrastructure, including telecommunications, operating systems, applications, data repositories, future technologies, and IT organizations. The skills of technical analysts include technical analysis, application design, and information systems knowledge.

- Enterprise Information System (EIS) Specialists—Provide domain expertise in the systems that are being integrated using WebLogic Integration adapters. The skills of EIS specialists include technical analysis and application integration design.
- System Administrators—Provide in-depth technical and operational knowledge about databases and applications deployed in an organization. The skills of system administrators include capacity and load analysis, performance analysis and tuning, deployment topologies, and support planning.

Additional Information

To learn more about the software components associated with the adapter, see the following documents:

- *BEA WebLogic Adapter for Baan Release Notes*

  http://edocs.bea.com/wladapters/baan/docs811/pdf/relnotes.pdf
- *BEA WebLogic Adapter for Baan Installation and Configuration Guide*

  http://edocs.bea.com/wladapters/baan/docs811/pdf/install.pdf
- *Introduction to the BEA WebLogic Adapters*

  http://edocs.bea.com/wladapters/docs81/pdf/intro.pdf
- BEA WebLogic Adapters 8.1 Dev2Dev Product Documentation

  http://dev2dev.bea.com/products/wladapters/index.jsp
- Application Integration documentation

  http://edocs.bea.com/wli/docs81/aiover/index.html

  http://edocs.bea.com/wli/docs81/aiuser/index.html
- BEA WebLogic Integration documentation

  http://edocs.bea.com/wli/docs81/index.html
- BEA WebLogic Platform documentation

  http://edocs.bea.com/platform/docs81/index.html
- Baan documentation

  http://www.baan.com

How to Use This Document

This document is designed to be used in conjunction with *Using the Application Integration Design Console*, available at the following URL:

`http://edocs.bea.com/wli/docs81/aiuser/index.html`

Using the Application Integration Design Console descibes, in detail, the process of defining an application view, which is a key part of making an adapter available to process designers and other users. What *Using the Application Integration Design Console* does *not* cover is the specific information about Adapter for Baan that you need to supply to complete the application view definition. You will find that information in this document.

At each point in *Using the Application Integration Design Console* where you need to refer to this document, you will see a note that directs you to a section in your adapter user guide, with a link to the edocs page for adapters. The following roadmap illustration shows where you need to refer from *Using the Application Integration Design Console* to this document.

Figure 1 Information Interlock with *Using the Application Integration Design Console*

Contact Us!

Your feedback on the BEA WebLogic Adapter for Baan documentation is important to us. Send us e-mail at **docsupport@bea.com** if you have questions or comments. Your comments will be reviewed directly by the BEA professionals who create and update the BEA WebLogic Adapter for Baan documentation.

In your e-mail message, please indicate that you are using the documentation for BEA WebLogic Adapter for Baan and the version of the documentation.

If you have any questions about this version of BEA WebLogic Adapter for Baan, or if you have problems using the BEA WebLogic Adapter for Baan, contact BEA Customer Support through BEA WebSUPPORT at **www.bea.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
- The name and version of the product you are using
- A description of the problem and the content of pertinent error messages

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 simultaneously.
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`

`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. *Example*s: LPT1 SIGNON OR
`{ }`	Indicates a set of choices in a syntax line. The braces themselves should never be typed.

Convention	Item
`[ ]`	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.

CHAPTER 1

Introducing the BEA WebLogic Adapter for Baan

This section introduces the BEA WebLogic Adapter for Baan and describes how the adapter enables integration with Baan business objects and WebLogic Integration.

It includes the following topics:

- About the BEA WebLogic Adapter for Baan
- Getting Started With the Adapter for Baan

About the BEA WebLogic Adapter for Baan

The BEA WebLogic Adapter for Baan connects to your Baan system so that you can easily use your Baan data and functions within your business processes. The adapter provides scalable, reliable, and secure access to your Baan system.

This section includes the following topics:

- Supported Baan Operations for Application Integration
- Supported Services
- Supported Events
- Benefits of the Adapter for Baan

Supported Baan Operations for Application Integration

The Adapter for Baan supports asynchronous, bi-directional message interactions for Baan services, events, and integration objects.

It provides integration with the following Baan operations:

- Access to Baan integration objects using XML to handle both services and events
- Direct invocation of Baan services and business components

Supported Services

The Adapter for Baan supports asynchronous services. To invoke a service, the adapter sends an XML file to Baan containing the service request. Baan returns the XML file containing the service response. The service request XML files are Open Applications Group (OAG) BOD specifications. These OAG documents are converted to JDBC commands that are sent to the RDBMS to update the Baan tables.

The following are the specific OAG documents that are used to update the tables used by specific Baan objects:

- OAG SyncCustomer to update Baan Customer object
- OAG SyncSalesOrder to update Baan Sales Order object
- OAG SyncInventory to update Baan Inventory object
- OAG SyncItem to update Baan Item object
- OAG SyncBOM to update Baan Bill of Materials object
- OAG SyncPO to update Baan Purchase Order object

Supported Events

The Adapter for Baan supports events for the execution of Baan Item, Customer, Sales Order, Inventory, Bill of Materials, and Purchase Order databases. The adapter uses RDBMS triggers to move data from Baan tables to an event table. The adapter polls the event table to send the data to WebLogic Integration for further processing.

Benefits of the Adapter for Baan

The combination of the adapter and WebLogic Integration supplies everything you need to integrate your workflows and enterprise applications with your Baan system. The Adapter for Baan provides these benefits:

- Integration can be achieved without custom coding.
- Business processes can be started by events generated by Baan.
- Business processes can request and receive data from your Baan system using services.
- Adapter events and services are standards-based. The adapter services and events provide extensions to the *J2EE Connector Architecture* (JCA) version 1.0 from Sun Microsystems, Inc. For more information, see the Sun JCA page at the following URL:

  `http://java.sun.com/j2ee/connector/`
- The adapter and WebLogic Integration solution is scalable. The BEA WebLogic Platform provides clustering, load balancing, and resource pooling for a scalable solution. For more information about scalability, see the following URL:

  `http://edocs.bea.com/wls/docs81/cluster/index.html`
- The adapter and WebLogic Integration solution benefits from the fault-tolerant features of the BEA WebLogic Platform. For more information about high availability, see the following URL:

  `http://edocs.bea.com/wli/docs81/deploy/index.html`
- The adapter and WebLogic Integration solution is secure, using the security features of the BEA WebLogic Platform and the security of your Baan system. For more information about security, see the following URL:

  `http://edocs.bea.com/wls/docs81/secintro/index.html`

Getting Started With the Adapter for Baan

This section gives an overview of how to get started using the BEA WebLogic Adapter for Baan within the context of an application integration solution. Integration with Baan involves the following tasks:

- Step 1: Design the Application Integration Solution
- Step 2: Determine the Required Baan Business Workflows
- Step 3: Define Application Views and Configure Services and Events
- Step 4: Integrate with Other BEA Software Components
- Step 5: Deploy the Solution to the Production Environment

Step 1: Design the Application Integration Solution

The first step is to design an application integration solution, which includes (but is not limited to) such tasks as:

- Defining the overall scope of application integration.
- Determining the business process(es) to integrate.
- Determining which WebLogic Platform components will be involved in the integration, such as web services or workflows designed in WebLogic Workshop, portals created in WebLogic Portal, and so on.
- Determining which external systems and technologies will be involved in the integration, such as Baan systems and other EISs.
- Determining which BEA WebLogic Adapters for WebLogic Integration will be required, such as the BEA WebLogic Adapter for Baan. An application integration solution can involve multiple adapters.

This step involves the expertise of business analysts, system integrators, and EIS specialists (including Baan specialists). Note that an application integration solution can be part of a larger integration solution.

Step 2: Determine the Required Baan Business Workflows

Within the larger context of an application integration project, you must determine which specific Baan integration objects are required for services and events to support the business processes in the application integration solution. Or, if you are invoking Baan business services or business components directly, rather than through a workflow, you must determine the tasks you need to complete.

Factors to consider include (but are not limited to):

- Type of Baan integration objects
- Baan transactions involved in business processes
- Logins required to access Baan and perform the required operations
- Whether operations are, from the adapter point of view:
  - services, which notify the Baan system, via an XML document, with a request for action
  - events, which are notifications from the Baan system that trigger workflows

This step involves the expertise of Baan specialists, including analysts and administrators.

Step 3: Define Application Views and Configure Services and Events

The next step is to create an application view that provides an XML-based interface between WebLogic Server and a particular Baan system within your enterprise. If you are accessing multiple Baan systems, you define a separate application view for each Baan system you want to access. To provide different levels of security access (such as “guest” and “administrator”), define a separate application view for each security level.

Once you define an application view, you can configure events and services in that application view that employ the XML schemas that are provided with the adapter. To learn more about schemas, see Chapter 2, “Introducing Schemas for Baan Integration Objects.”

To learn more about defining application views, see Chapter 3, “Defining Application Views for Baan” in conjunction with*Using the Application Integration Design Console*, at the following URL:

`http://edocs.bea.com/wli/docs81/aiuser/index.html`

Step 4: Integrate with Other BEA Software Components

Once you have configured and published one or more application views for Baan integration, you can integrate these application views into other BEA software components, such as workflows or web services created in BEA WebLogic Workshop, or portals built with BEA WebLogic Portal.

For more information, see *Using the Application Integration Design Console*, particularly Chapter 3, "Using Application Views with Application Workflows," at the following URL:

`http://edocs.bea.com/wli/docs81/aiuser/index.html`

Step 5: Deploy the Solution to the Production Environment

After you have designed, built, and tested your application integration solution, you can deploy it into a production environment. The following list describes some of the tasks involved in deploying an application integration:

- Design the deployment.
- Deploy the required components of the BEA WebLogic Platform.
- Install and deploy the BEA WebLogic Adapter for Baan as described in *BEA WebLogic Adapter for Baan Installation and Configuration Guide*
- Deploy your application views and schemas for Baan integration.
- Verify business processes in the production environment.
- Monitor and tune the deployment.

CHAPTER 2

Introducing Schemas for Baan Integration Objects

The WebLogic Adapter for Baan uses XML documents to communicate with your Baan system's integration objects for both services and events. All of the XML documents the adapter sends to, or receives from the Baan system must be defined by schemas.

This section explains schemas and schema repositories. It contains the following topics:

About Schemas

Each service or event the Adapter for Baan uses is defined by a schema. The adapter uses the following schemas:

Service Requests

Service requests are requests for action that your application makes to your Baan system. Requests are defined by request schema. As part of the definition, the request schema defines the input parameters required by the Baan system. The Baan system responds to the request with a service response.

Service Responses

Service responses are the way the Baan system responds to a service request. A service response schema defines this service response. Service requests always have corresponding responses.

Events

Events are generated by the Baan system as a result of activity on that system. You can use these events to trigger an action in your application. For example, the Baan system may generate an event when customer information is updated. If your application must do something when this happens, your application is a consumer of this event. Events are defined by event schema.

About the Schemas Included With the Adapter

Typically, you use the BEA Application Explorer to generate schemas. However, the Baan system does not expose the metadata required by the BEA Application Explorer, so the adapter is packaged with schemas for the following supported Baan objects:

- Customer
- Sales Order
- Item
- Inventory
- Purchase Order
- Bill of Materials

These schemas must be stored in a schema repository.

About Schema Repositories

A schema repository stores schema information.

A schema repository consists of the following elements:

- a manifest file that describes the event and service schemas contained in the repository
- the event and service schemas

When you use the WebLogic Integration Application View Console to create an Application View, a schema repository is automatically created for you. In addition, the Application View creation process also creates a repository manifest and extracts the schemas into the repository.

About the Repository Manifest

Each schema repository has a manifest that describes the repository and its contents. The repository manifest is an XML file named `manifest.xml`. This file is created automatically when the adapter extracts the repository from the EAR file.

The following is an example of a manifest file showing the relationships between events and schemas and service request and response schemas.

Listing 2-1 Sample Manifest File

```
<manifest>
<connection>
</connection>
<schemaref name="SYNC_CUSTOMER_005" alias="SYNC_CUSTOMER_005">
       <request root ="SYNC_CUSTOMER_005" file="SYNC_CUSTOMER_005.xsd"/>
       <response root="SYNC_CUSTOMER_005" file="SYNC_CUSTOMER_005.xsd"/>
</schemaref>
<schemaref name="BaanEvent">
       <event root="BAAN" file="event_Baan_Event_Table.xsd"/>
</schemaref>
</manifest>
```

The repository has a connection section, which can be ignored for this adapter. It also has a schema reference section, named `schemaref`. The schema reference name appears in the drop-down list on the Add Service or Add Event screens in the WebLogic Integration Application

View Console. Each named schema reference can contain three schemas, one of each type. To learn more about schema types, see About the Schemas Included With the Adapter.

Naming Schema Repositories

The schema has a three-part naming convention.

*session_base_directory**adapter**connection_name*

These parts are:

- *session_base_directory*—the schema's session base path, which represents a folder under which multiple sessions of schemas may be held
- *adapter*—the type of adapter (for example, Baan)
- *connection_name*—a name representing a particular instance of the adapter type

Next Steps

The next step is to create an application view. An application view makes the services and events available to applications. To learn more about application views, see Defining Application Views for Baan.

CHAPTER 3

Defining Application Views for Baan

An application view is a business-oriented interface to objects and operations within an EIS.

This section presents the following topics:

How to Use This Document

This document is designed to be used in conjunction with *Using the Application Integration Design Console*, available at the following URL:

`http://edocs.bea.com/wli/docs81/aiuser/index.html`

Using the Application Integration Design Console describes, in detail, the process of defining an application view, which is a key part of making an adapter available to process designers and other users. What *Using the Application Integration Design Console* does *not* cover is the specific information—about connections to your Baan system, as well as supported services and events—that you must supply as part of the application view definition. You will find that information in this section.

At each point in *Using the Application Integration Design Console* where you need to refer to this document, you will see a note that directs you to a section in your adapter user guide, with a link to the edocs page for adapters. The following road map illustration shows where you need to refer from *Using the Application Integration Design Console* to this document.

Figure 3-1 Information Interlock with *Using the Application Integration Design Console*

Before You Begin

Before you define an application view, make sure you have:

- Installed and deployed the adapter according to the instructions in *BEA WebLogic Adapter for Baan Installation and Configuration Guide*.
- Determined which business processes need to be supported by the application view. The required business processes determine the types of services and events you include in your application views. Therefore, you must gather information about the application's business requirements from the business analyst. Once you determine the necessary business

processes, you can define and test the appropriate services and events. For more information, see "Getting Started With the Adapter for Baan" on page 1-4.

- Gathered the connection information for your Baan system.

About Application Views

An application view defines:

- Connection information for the EIS, including login information, connection settings, and so on.
- Service invocations, including the information the EIS requires for this request, as well as the request and response schemas associated with the service.
- Event notifications, including the information the EIS publishes and the event schema for inbound messages.

Typically, an application view is configured for a single business purpose and contains only the services and events required for that purpose. An EIS might have multiple application views, each defined for a different purpose.

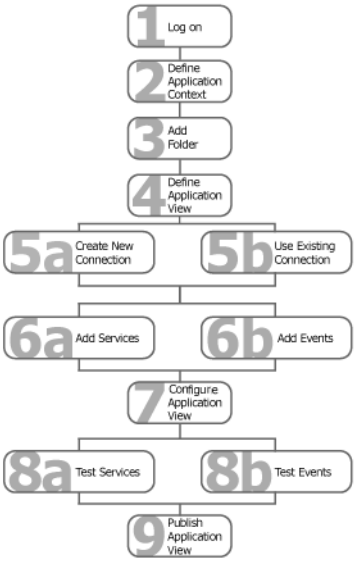
About Defining Application Views

Defining an application view is a multi-step process described in *Using the Application Integration Design Console*, available at the following URL:

`http://edocs.bea.com/wli/docs81/aiuser/index.html`

The information you enter depends on the requirements of your business process and your EIS system configuration. Figure 3-2 summarizes the procedure for defining and configuring an application view.

Figure 3-2 Process for Defining and Configuring an Application View



To define an application view:

1. Log on to the WebLogic Integration Application View Console.
2. Define the application context by selecting an existing application or specifying a new application name and root directory.

   This application will be using the events and services you define in your application view. The application view works within the context of this application.
3. Add folders as required to help you organize application views.
4. Define a new application view for your adapter.
5. Add a new connection service or select an existing one.

   If you are adding a new connection service, see "Defining Service Connection Parameters" on page 3-5 for details about Baan requirements.
6. Add the events and services for this application view.

   See the following sections for details about Baan requirements:

   – "Setting Service Properties" on page 3-6

– "Setting Event Properties" on page 3-9

7. Perform final configuration tasks.

   If you are adding an event connection, see "Defining Event Connection Parameters" on page 3-10 for details about Baan requirements.

8. Test all services and events to make sure they can properly interact with the target Baan system.

   See the following sections for details about Baan requirements:

   – "Testing Services" on page 3-12

   – "Testing Events Using a Service" on page 3-13

   – "Testing Events Manually" on page 3-14

9. Publish the application view to the target WebLogic Workshop application.

   This is the application you specified in step 2. Publishing the application view allows workflow developers within the target application to interact with the newly published application view using an Application View control.

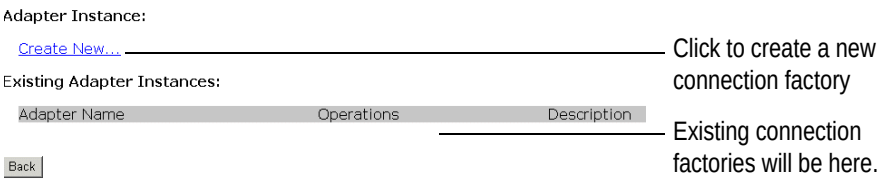
Defining Service Connection Parameters

1 2 3 4 **5** 6 7 8 9

This information applies to "Step 5A, Create a New Browsing Connection" in *Using the Application Integration Design Console*, at the following URL:

`http://edocs.bea.com/wli/docs81/aiuser/index.html`

The Select Browsing Connection page allows you to choose the type of connection factory to associate with the application view. You can select a connection factory within an existing instance of the adapter or create a connection factory within a new adapter instance.

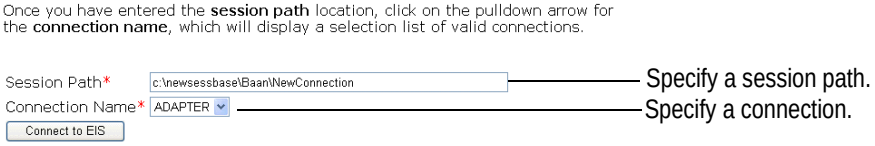


After you enter a connection name and description, you use the Configure Connection Parameters page to specify connection parameters for a connection factory.

To create a new browsing connection:

1. In the Create New Browsing Connections page, enter a connection name and description as described in *Using the Application Integration Design Console*.

   The Configure Connection Parameters page appears to allow you to configure the newly created connection factory within the new adapter instance.



   Note: A red asterisk (*) indicates that a field is required.

2. Specify a session path and connection name.

   This information enables the application view to interact with the target Baan system. You need enter this information only once per application view.

   During creation of an application view, the schemas are extracted from the EAR file into a repository at the location specified in the session path.

3. Click Connect to EIS.

   You return to the Create New Browsing Connections, where you can specify connection pool parameters and logging levels. For more information, see *Using the Application Integration Design Console* at the following URL:

   `http://edocs.bea.com/wli/docs81/aiuser/index.html`

Setting Service Properties

This information applies to "Step 6A, Add a Service to an Application View" in *Using the Application Integration Design Console*, at the following URL:

`http://edocs.bea.com/wli/docs81/aiuser/index.html`

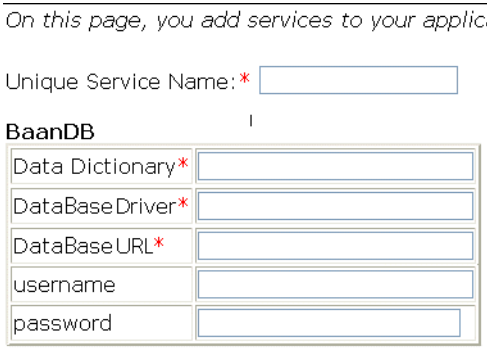
Adapter for Baan uses services to make requests of the Baan system. A service consists of both a request and a response.

To configure a service:

1. Enter a unique service name that describes the function the service performs.

2. Select File from the Select list.

   The Add Services page displays the fields required for this service type.



   Note: A red asterisk (*) indicates that a field is required.

3. Enter the following information:

Table 3-1 Service Parameters

Parameter	Description
Data Dictionary	The full path and file name of the `Baan_Data_Dictionary.xml` file. For more information on this file and its location, see the *BEA WebLogic Adapter for Baan Installation and Configuration Guide*.
DataBaseDriver	The proper name for the JDBC driver that the BEA WebLogic Adapter for Baan uses to connect to the Baan database.
DataBaseURL	JDBC URL required to connect to the database
username	User ID must have the following privileges in the Baan database: connect, insert, update, and delete
password	Password for the user ID

4. See "Common Service and Event Settings" on page 3-8 for information about selecting a schema and configuring logging and tracing.

Common Service and Event Settings

1 2 3 4 5 **6** 7 8 9

This information applies to "Step 6A, Add a Service to an Application View" in *Using the Application Integration Design Console*, at the following URL:

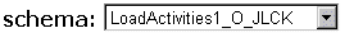
`http://edocs.bea.com/wli/docs81/aiuser/index.html`

You select a schema and select logging options the same way for all services.

To set common service settings:

1. In the Schema list, select the schema you want to use with this service.

   For more information, see Chapter 2, "Introducing Schemas for Baan Integration Objects."

2. Configure logging and tracing for this service, as follows:

   Logging captures information from your adapter and writes it in a log file. Tracing displays runtime information in the console. You set the type and amount of information you wish to capture as part of the final configuration tasks. This is described in detail in *Using the Application Integration Design Console*.

   a. Select the Trace on/off check box to enable tracing for this service. Trace information appears in the runtime console.

   b. Select the Verbose Trace on/off check box to enable additional tracing information for this service. Trace information appears in the runtime console.

   c. Select the Document Trace on/off check box to enable tracing for this service to include the documents sent to and from the adapter. Trace information appears in the runtime console.

3. Click Add to add the service.

   For more information about the next step, see *Using the Application Integration Design Console* at the following URL:

   `http://edocs.bea.com/wli/docs81/aiuser/index.html`

Setting Event Properties

1 2 3 4 5 **6** 7 8 9

This information applies to "Step 6B, Add an Event to an Application View" in *Using the Application Integration Design Console*, at the following URL:

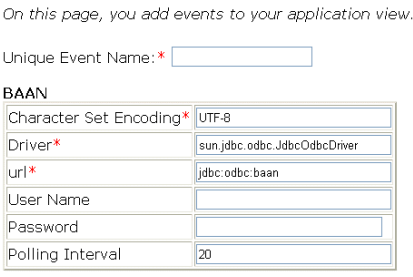
`http://edocs.bea.com/wli/docs81/aiuser/index.html`

An event defines how your application responds to events generated by Baan.

To configure an event:

1. Enter a unique event name that describes the function the event performs.

   The Add Events page displays the fields required for a Baan event.



   Note: A red asterisk (*) indicates that a field is required.

2. Enter the following information:

Table 3-2 Event Parameters

Parameter	Description
Character Set Encoding	The character set encoding for inbound documents. For example, UTF-8.
Driver	The proper name for the JDBC driver the BEA WebLogic Adapter for Baan uses when querying the Baan Event Table.
url	The JDBC URL required to connect to the database

Table 3-2 Event Parameters (Continued)

Parameter	Description
User Name	User ID must have the following privileges in the Baan database: connect, insert, update, and delete Note the following usage consideration for the `iwadapt` user ID: By design, the event does not listen for database changes made by the `iwadapt` user ID. Therefore, if you are using a configuration that uses both a service and an event, you must decide whether to use the `iwadapt` ID. For the event to listen for database changes caused by the service, the service user ID cannot be `iwadapt`. If the event need not listen for changes caused by the service, you can use the `iwadapt` user ID as the triggers ignore these changes by default.
Password	Password for the user ID
Polling Interval	Indicates how often, in seconds, the adapter issues the SQL Query. The default value is 20 seconds.

3. See "Common Service and Event Settings" on page 3-8 for information about selecting a schema and configuring logging and tracing.

Defining Event Connection Parameters

This information applies to "Step 7, Perform Final Configuration Tasks" in *Using the Application Integration Design Console*, at the following URL:

`http://edocs.bea.com/wli/docs81/aiuser/index.html`

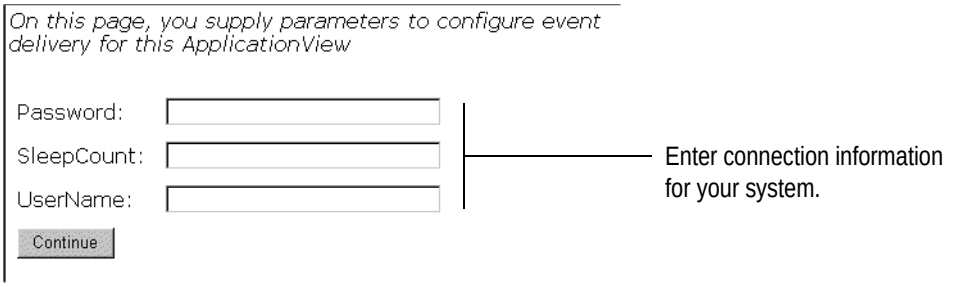
Once you have finished adding services and events and have saved your application view, you must perform some final configuration tasks, including configuring event delivery connections, before testing the services and events. You perform these configuration tasks from the Final Configuration and Testing page.

To define event connection parameters:

1. In Connections area on the Application View Administration page, click Select/Edit.
2. Select Event Connection for this application view.
3. In the Event Connection area, select existing and select an Event Connection. Click OK.

4. Click Event to configure the event delivery parameters.
5. Click the Define link for the connection parameters.

   The Configure Event Delivery Parameters page appears.



   Note: A red asterisk (*) indicates that a field is required.

6. Enter the following information:

Table 3-3 Event Connection Parameters

Parameter	Description
UserName	Your WebLogic Server Administration Console user name, defined in the `startWebLogic` script
Password	The password for your WebLogic Server Administration Console user name
SleepCount	The number of seconds the adapter will wait between polling for events

   The event delivery parameters you enter on this page enable connection to your Baan system and are used when generating events. The parameters are specific to the associated adapter and are defined in the `wli-ra.xml` file within the base adapter.

7. Click Continue to return to the Edit Event Adapter page. Click OK to go back to the Connection Information page.
8. Click Back to return to the Final Configuration and Testing page.

   The Edit Event Adapter page allows you to define event parameters and configure the information that will be logged for the connection factory. Select one of the following settings for the log:

   – Log errors and audit messages

– Log warnings, errors, and audit messages

– Log informational, warning, error, and audit messages

– Log all messages

Note: For maximum tracing, select Log all Messages. This is the recommended setting to use when you are collecting debugging information for BEA support.

The table that follows describes the type of information that each logging message contains.

Table 3-4 Logging message categories

This type of message	Contains
Audit	Extremely important information related to the business processing performed by an adapter.
Error	Information about an error that has occurred in the adapter, which may affect system stability.
Warning	Information about a suspicious situation that has occurred. Although this is not an error, it could have an impact on adapter operation.
Information	Information about normal adapter operations.

Testing Services

This information applies to "Step 8A, Test an Application View's Services" in *Using the Application Integration Design Console*, at the following URL:

`http://edocs.bea.com/wli/docs81/aiuser/index.html`

The purpose of testing an application view service is to evaluate whether that service interacts properly with the target Baan system. When you test a service, you supply any inputs required to start the service. For the Adapter for Baan, the input is in the form of a valid XML string that acts as input for the service.

Note: You can test an application view only if it is deployed and only if it contains at least one event or service.

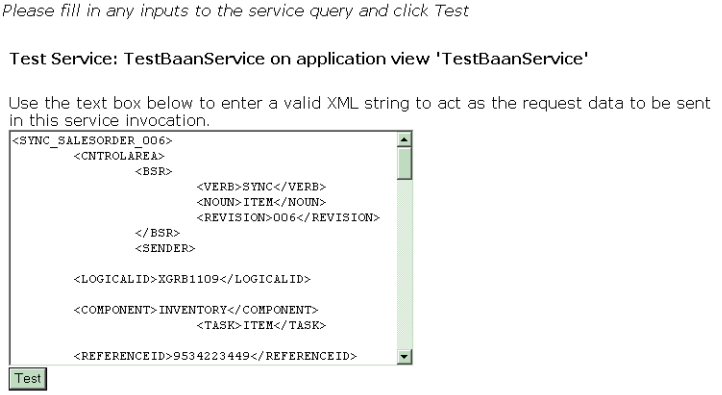
To test a service:

1. In the Application View Administration page, click Test.

   The Summary of Application View page appears.

2. Click Test beside the Baan service to be tested.

3. In the Test Service window, enter the appropriate XML strings for your account.



4. Click Test.

   The results appear in the Test Results window.

Testing Events Using a Service

This information applies to "Step 8B, Test an Application View's Events" in *Using the Application Integration Design Console*, at the following URL:

`http://edocs.bea.com/wli/docs81/aiuser/index.html`

The purpose of testing an application view event is to make sure that the adapter correctly handles events generated by Baan. When you test an event, you can trigger the event using a service or manually.

Note: You can test an application view only if it is deployed and only if it contains at least one event or service.

To test an event:

1. In the Application View Administration page, click Test.

   The Summary of Application View page appears.

2. Click Test beside the Baan event to be tested.

   The Event Test page appears.

3. Select a service that triggers the event you are testing.

4. In the Time field, enter a reasonable period of time to wait, specified in milliseconds, before the test times out (One second = 1000 milliseconds. One minute = 60,000 milliseconds.).

5. Click Test and enter the XML string needed to trigger the service.

   The service is executed.

   – If the test succeeds, the Test Result page appears, showing the event document, the service input document, and the service output document.

   – If the test fails, the Test Result page displays only a Timed Out message.

Testing Events Manually

This information applies to "Step 8B, Test an Application View's Events" in *Using the Application Integration Design Console*, at the following URL:

`http://edocs.bea.com/wli/docs81/aiuser/index.html`

To test an event manually:

1. In the Time field, enter a reasonable period of time to wait, specified in milliseconds, before the test times out (One second = 1000 milliseconds. One minute = 60,000 milliseconds.).

2. Click Test. The test waits for an event to trigger it.

3. Using the triggering Baan application, perform an action that executes the service that, in turn, tests the application view event.

   – If the test succeeds, the Test Result page appears. This page displays the event document from the application, the service input document, and the service output document.

   – If the test fails or takes too long, the Test Result page appears, showing a Timed Out message.

Index