BEA WebLogic Integration Adapters

Introduction

Version 8.1.0
Document Revised: July 2003

Copyright

Restricted Rights Legend

Trademarks or Service Marks

Contents

About This Document

Overview

Concepts

Roles, Responsibilities, and Tasks

Index

About This Document

This document introduces the BEA® WebLogic® Adapters for WebLogic Integration™, which enable fast, simple, and robust enterprise application integration. Compliant with the J2EE Connector Architecture specification, each adapter provides bi-directional, request-response integration with a specific application or technology.

This document covers the following topics:

- Chapter 1, "Overview," provides a high-level overview of the BEA WebLogic Integration Adapters.
  - What are the BEA WebLogic Integration Adapters?
  - How Do Adapters Fit Into the WebLogic Architecture?
  - Types of Adapters
- Chapter 2, "Concepts," introduces the concepts to understand before using adapters in an integration solution.
  - Integration Solutions Involving EIS Integration
  - Components of Integration Solutions Involving EIS Integration
  - Tools for Integration Solutions
  - Run-Time Processing of Services and Events
- Chapter 3, "Roles, Responsibilities, and Tasks," describes roles and responsibilities, and provides a high-level summary of tasks, required to develop an integration solution that involves application integration.

What You Need to Know

This document is intended mainly for application integrators who will use one or more of the BEA WebLogic Integration Adapters to build and deploy an integration solution that involves integration with an enterprise information system (EIS). It assumes a familiarity with the BEA WebLogic Platform™ and with the applications that you want to integrate.

e-docs Web Site

BEA product documentation is available on the BEA corporate Web site. From the BEA Home page, click on Product Documentation or go directly to the "e-docs" Product Documentation page at http://e-docs.bea.com.

How to Print the Document

You can print a copy of this document from a Web browser, one file at a time, by using the File→Print option on your Web browser.

A PDF version of this document is available on the BEA WebLogic Integration Adapters documentation Home page on the e-docs Web site (and also on the documentation CD). You can open the PDF in Adobe Acrobat Reader and print the entire document (or a portion of it) in book format. To access the PDFs, open the BEA WebLogic Integration Adapters documentation Home page, click the PDF files button and select the document you want to print.

If you do not have the Adobe Acrobat Reader, you can get it for free from the Adobe Web site at http://www.adobe.com/.

Related Information

The following documents contain information that is relevant to using the BEA WebLogic Integration Adapters:

- The documentation for the particular adapter(s) that you will use in your integration solution. For a comprehensive list of available adapters, see

  `http://edocs.bea.com/wladapters/docs81/index.html`

- BEA WebLogic Adapters 8.1 edocs Product Documentation at

  `http://edocs.bea.com/wladapters/docs81/index.html`

- BEA WebLogic Adapters 8.1 Dev2Dev Product Documentation at

  `http://dev2dev.bea.com/products/product.jsp?highlight=wla`

- *Introducing Application Integration* in the BEA WebLogic Integration documentation at

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

- *Using Application Integration* in the BEA WebLogic Integration documentation at

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

- BEA WebLogic Integration™ documentation at

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

- BEA WebLogic Platform documentation at

  `http://edocs.bea.com/platform/docs81/index.html`

- Sun Microsystems J2EE Connector Architecture page at

  `http://java.sun.com/j2ee/connector/`

Contact Us!

Your feedback on the BEA WebLogic Integration Adapters 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 Integration Adapters documentation.

In your e-mail message, please indicate that you are using the documentation for the BEA WebLogic Integration Adapters **8.1.0** release.

If you have any questions about this version of BEA WebLogic Integration Adapters, or if you have problems installing and running BEA WebLogic Integration Adapters, 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 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

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`*

Convention	Item
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.

`[ ]`	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

Overview

This section provides an overview of the BEA WebLogic Integration Adapters. This section includes the following topics:

- What are the BEA WebLogic Integration Adapters?
- How Do Adapters Fit Into the WebLogic Architecture?
- Types of Adapters

What are the BEA WebLogic Integration Adapters?

The BEA WebLogic Integration Adapters enable fast, simple, and robust enterprise application integration. Compliant with the J2EE™ Connector Architecture specification from Sun Microsystems, Inc., each adapter provides bi-directional, request-response integration with a specific application, protocol, or technology. The BEA WebLogic Integration Adapters are used with the Application Integration component of BEA WebLogic Integration™. Users purchases the individual adapters they want, and then install them to work in conjunction with BEA WebLogic Integration.

The BEA WebLogic Integration Adapters provide:

- Easy and Fast Integration with Enterprise Information Systems
- Compliance with the J2EE Connector Architecture
- Scalable, Reliable, and Secure Integration Framework

In addition to the BEA WebLogic Integration Adapters, developers can create custom adapters using the WebLogic Integration Adapter Development Kit (ADK). The ADK is a set of tools for implementing the events and services supported by BEA WebLogic Integration. The process of creating custom adapters with the ADK is described in *Developing Adapters* at the following URL:

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

Note: Throughout the rest of this document, the term *adapter* refers to any of the BEA WebLogic Integration Adapters, while the term *custom adapter* refers to any custom adapter that was created using the ADK.

Easy and Fast Integration with Enterprise Information Systems

An *enterprise information system* (EIS) is an application that provides the information infrastructure for an enterprise. An EIS offers its clients a set of services that are made available to clients via local and/or remote interfaces. Examples of EISs include:

- Enterprise Resource Planning (ERP) systems, such as SAP R/3 or PeopleSoft
- Customer Relationship Management (CRM) systems, such as Siebel
- Database systems, such as Oracle

The BEA WebLogic Integration Adapters work with the most popular and widely used EISs. The adapters enable organizations to quickly design and deploy integration solutions that involve existing and future EIS resources. Adapters simplify integration efforts by providing robust, standards based connectivity with various applications within a coherent framework built on top of WebLogic Server.

Compliance with the J2EE Connector Architecture

The BEA WebLogic Integration Adapters are implementations of the *J2EE Connector Architecture* (JCA) version 1.0, from Sun Microsystems, Inc. The JCA is used for integrating J2EE-compliant application servers with enterprise information systems (EIS). The JCA consists of two parts: an EIS-specific resource adapter (such as those provided in the BEA WebLogic Integration Adapters) and an application server (such as BEA WebLogic Server) that the resource adapter plugs into.

The JCA defines a set of contracts, such as transactions, security, and connection management, that a resource adapter must support in order to plug in to an application server. For more information, see the Sun JCA page at the following URL:

`http://java.sun.com/j2ee/connector/`

Scalable, Reliable, and Secure Integration Framework

The BEA WebLogic Integration Adapters are crucial components of a comprehensive enterprise integration framework that provides:

- *Scalability* via the clustering, load balancing, and resource pooling features of BEA WebLogic Platform. In a scalable deployment, adding a linear amount of resources—such as memory, processors, or machines—will result in a corresponding linear increase in throughput while maintaining the same response level.
- *Reliability* via the fault-tolerant and server fail-over features of BEA WebLogic Platform. If a critical problem occurs—such as an e-business application bug or an operating system, hardware, or network failure—integration solutions remain up-and-running.
- *Security* for mission-critical applications and data via the extensive security mechanisms of BEA WebLogic Platform as well as the security of integrated EISs. WebLogic security mechanisms provide user authentication and authorization capabilities, integration with existing security systems (such as LDAP) through security realms, controls in both the presentation and business layers, and firewall interoperability.

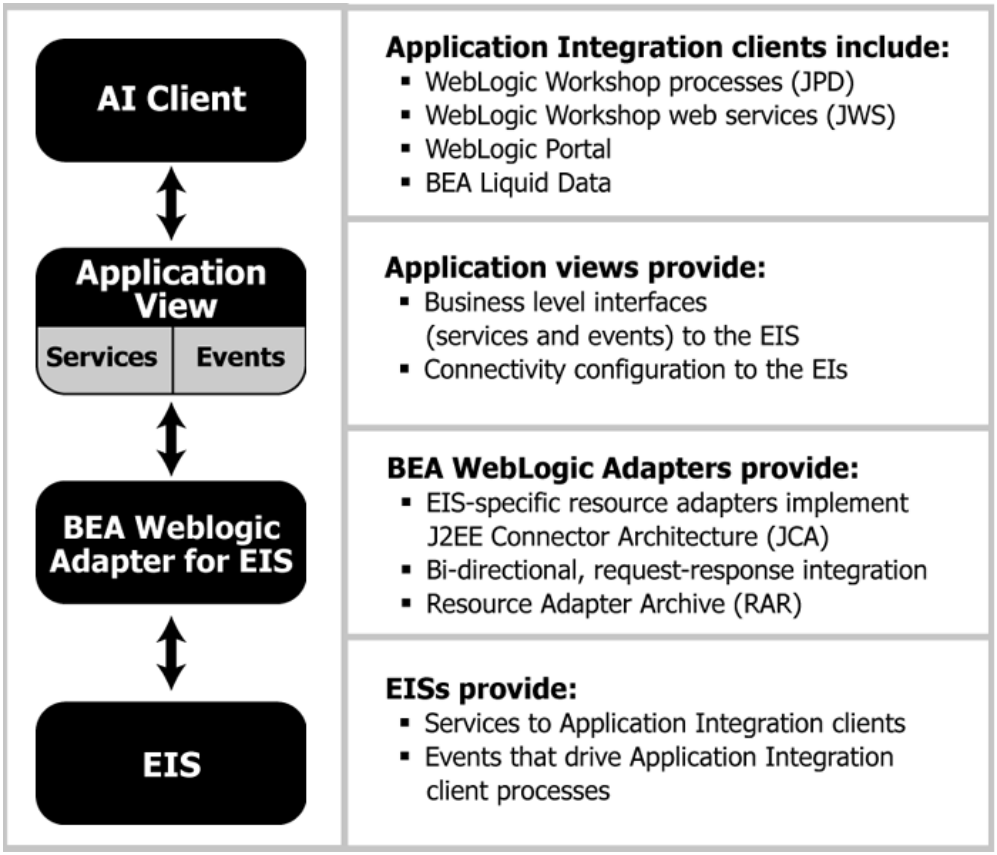
How Do Adapters Fit Into the WebLogic Architecture?

This section describes how the adapters are used in WebLogic Integration. It includes the following topics:

Adapters and Application Integration

Adapters are used in conjunction with the Application Integration component of BEA WebLogic Integration. This component provides a systematic, standards-based architecture for hosting business-oriented interfaces to enterprise applications.

Figure 1-1 Adapters in the Application Integration Architecture



For more information about Application Integration clients, application views, services, and events, see "Components of Integration Solutions Involving EIS Integration" on page 2-2. For general information about Application Integration, see the following documents:

- *Introducing Application Integration* at the following URL:

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

- *Using Application Integration* at the following URL:

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

About BEA WebLogic Integration

BEA WebLogic Integration is the single, unified Workshop IDE platform that provides the functionality businesses can use to:

- develop new applications
- integrate them with existing systems
- streamline business processes
- extend e-business infrastructure through portal gateways.

With WebLogic Integration, application development, business integration, and easy-to-implement portal development has converged into one easy environment—all built on the solid J2EE foundation of WebLogic Server. WebLogic Integration uses Web services to integrate distributed systems inside and outside the organization and utilizes BEA WebLogic Workshop to simplify application development. For more information about BEA WebLogic Integration, see the main documentation page at the following URL:

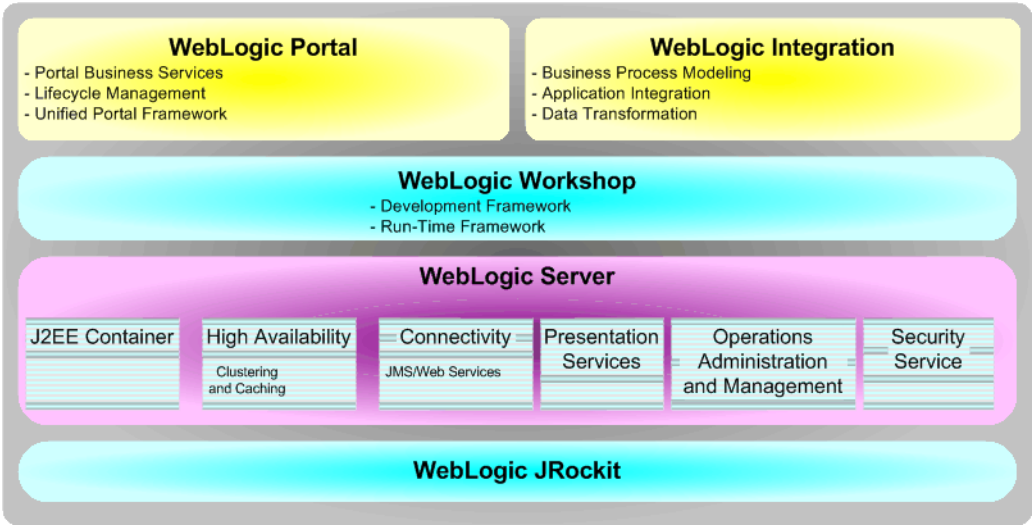
`http://edocs.bea.com/wli/docs81/index.html`

About the BEA WebLogic Platform

BEA WebLogic Platform is a unified, simplified, and extensible platform for developing, deploying, and managing enterprise applications. BEA WebLogic Platform includes a rich development and deployment framework for integrating third-party application packages, custom applications, messaging systems, Web Services, trading partners, and more into your environment.

The following figure shows the BEA WebLogic Platform architecture.

Figure 1-2 WebLogic Platform Architecture



In addition to BEA WebLogic Integration, BEA WebLogic Platform includes the following components:

Table 1-1 Other Components of the BEA WebLogic Platform

Component	Description
BEA WebLogic Portal	Enterprise portal platform that simplifies the production and management of custom-fit portals. WebLogic Portal simplifies, personalizes, and lowers the cost of customer, partner and employee access to information, applications and business processes by leveraging: • a unified portal framework • simplified portal lifecycle management • modular pre-integrated portal business services
BEA WebLogic Workshop	Unified, integrated development framework that makes it easy for all developers—not just J2EE experts—to build powerful, standards-based J2EE applications across the entire WebLogic Platform.

Table 1-1 Other Components of the BEA WebLogic Platform (Continued)

Component	Description
BEA WebLogic Server	Foundation of the platform, providing the core services to ensure a reliable, available, scalable, and high performing execution environment for your platform. WebLogic Server is completely independent of the underlying hardware, operating system, and databases on which it operates.
BEA WebLogic JRocket	High-performance Java Virtual Machine (JVM) optimized for server-side performance and scalability. WebLogic JRockit is supported by all WebLogic Platform components and is fully integrated into the WebLogic Platform package, in addition to being available as a standalone JVM. An optional alternative to the Sun JVM.

For more information about BEA WebLogic Platform, see the documentation at the following URL:

`http://edocs.bea.com/platform/docs81/index.html`

Types of Adapters

Adapters can be classified into the following four types:

Table 1-2 Types of Adapters

Adapter Type	Description	Example
Infrastructure Adapters	Facilitate the integration of existing technology standards and messaging processes with the rest of your enterprise resources.	• BEA WebLogic Integration Adapter for MQSeries® • BEA WebLogic Integration Adapter for RDBMS
Application Adapters	Provide standards-based connectivity between your enterprise applications and the rest of your enterprise resources.	• BEA WebLogic Integration Adapter for Oracle® Applications • BEA WebLogic Integration Adapter for PeopleSoft® • BEA WebLogic Integration Adapter for SAP® • BEA WebLogic Integration Adapter for Siebel®
Format Adapters	Facilitate the integration of existing messaging standards and processes with the rest of your enterprise resources.	• Currently not available in WebLogic Integration 8.1

CHAPTER 2

Concepts

This section describes the concepts you need to understand before you begin to use any of the BEA WebLogic Integration Adapters. This section includes the following topics:

- Integration Solutions Involving EIS Integration
- Components of Integration Solutions Involving EIS Integration
- Tools for Integration Solutions
- Run-Time Processing of Services and Events

Integration Solutions Involving EIS Integration

A *business process* is a set of interconnected business activities with a beginning and an end, and clearly defined inputs and outputs. An *integration solution* involves the integration of one or more related business processes. To integrate with an enterprise information system (EIS), an integration solution uses the appropriate adapter. An integration solution that involves an EIS includes all the information associated with a particular business process: an initiating event, participating EIS and other applications, EIS metadata, data transformation, process definitions, and so on.

Such integration solutions are specific implementations of BEA's Application Integration framework. They can be simple, such as obtaining the status of a given sales order, or more complex, such as automatically adding a new customer in one EIS whenever a new customer is added in another EIS. Integration solutions can span multiple system, geographic, and organizational boundaries. For example, getting bids from different suppliers over the Web involves integration of the buyer's purchasing system and the suppliers' sales and inventory

systems. Integration solutions involving EIS integration can be complete solutions in themselves, or they can be parts of larger integration solutions involving other, non-EIS business processes.

Note: While integration solutions can include the use of custom adapters created using the Adapter Development Kit (ADK), this document focuses on solutions that involve BEA WebLogic Integration Adapters.

Components of Integration Solutions Involving EIS Integration

This section describes the following components of integration solutions that involve EIS integration:

- Enterprise Information Systems
- Resource Adapters
- Application Views
- Application Integration Service Clients and Event Consumers

For a diagram that shows where adapters fit into the Application Integration architecture, see "Adapters and Application Integration" on page 1-3.

Enterprise Information Systems

An *enterprise information system* (EIS) is software that provides the information infrastructure for an enterprise. An EIS offers a set of services to its clients, which are made available to clients via local and/or remote interfaces. An integration solution involves integration with one or more EISs. For examples of EISs, see "Easy and Fast Integration with Enterprise Information Systems" on page 1-2.

Resource Adapters

A *resource adapter* (or simply *adapter*) is a software component that acts as a connector between an EIS and a J2EE application server (such as BEA WebLogic Server). Each adapter provides bi-directional, request-response integration with a specific application or technology. Resource adapters are implementations of the *J2EE Connector Architecture* (JCA) version 1.0, from Sun Microsystems, Inc. For more information, see the Sun JCA page at the following URL:

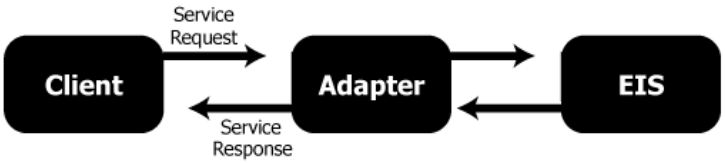
`http://java.sun.com/j2ee/connector/`

Supported Operations

Adapters handle two general types of operations:

- *Services* are request / response communications with the EIS. Client applications submit service requests to the EIS via the adapter, and the adapter returns the EIS response back to the client. For example, a business process might invoke a SAP BAPI or execute a SELECT statement on a database. Responses are either synchronous or asynchronous.

Figure 2-1 Service Invocations



- *Events* are asynchronous, one-way messages received from an EIS. For example, the adapter can receive an IDOC from a SAP system or a message from an MQ system. The adapter routes the EIS message to the appropriate software component via the WebLogic Integration Message Broker and the Application Integration JMS infrastructure.

Figure 2-2 Event Notifications

In effect, a service is a *request for some work to be done* and an event is a *notification that some work has been done*.

At run-time, the EIS and the adapter exchange requests, responses, and events as XML documents. The adapter automatically handles the data translation between the EIS format and the XML format via schemas that are defined at design-time. For more information, see "EIS Metadata, Schemas, and Repositories" on page 2-7.

For detailed information about services and events, see "Run-Time Processing of Services and Events" on page 2-11.

Supported Adapters

In BEA WebLogic Integration, integration solutions can use two kinds of adapters:

- BEA WebLogic Integration Adapters—Users purchase individual pre-built adapters and install them to work in conjunction with BEA WebLogic Integration.
- **Third-party adapters**—Adapters created by BEA partners who participate in BEA's Adapter Partner Program.
- **Custom adapters**—Developers can create custom adapters using the Adapter Development Kit (ADK), which is described in *Developing Adapters* at the following URL:

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

For an overview of adapters, see "What are the BEA WebLogic Integration Adapters?" on page 1-1.

Application Views

An *application view* is a business oriented interface to objects and operations within an EIS. Application views include the information needed to communicate with the EIS as well as configurations for services and events. Application views define:

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

An application view is typically configured for a single business purpose and contains only the services or events required for that business purpose. An EIS might have multiple application views defined for different business purposes. For example, an EIS containing human resources data might have an *HREmployee* application view that provides individual employees with read-only access to their personnel information, and an *HRDataEntry* application view that provides data entry clerks with the ability to add, update, and delete personnel information.

Application views provide a layer of abstraction between applications and the EIS, making it easier for developers and nonprogrammers to access and maintain the services and events exposed by the adapter.

You create application views in either of two ways:

- Using the Application View Console, which is described in "Application View Console" on page 2-10. For detailed information about application views, see "What Are Application Views?" in "Understanding Application Integration" in *Using Application Integration* at the following URL:

  `http://edocs.bea.com/wli/docs81/aiover/2intfra.html`

- Writing custom code. For more information, see "Using Application Views by Writing Custom Code" in *Using Application Integration* at the following URL:

  `http://edocs.bea.com/wli/docs81/aiuser/4usrcust.html`

Application Integration Service Clients and Event Consumers

This section describes clients for service invocations and consumers for event notifications.

Clients for Service Invocations

The following table describes the kinds of clients that invoke services on an EIS via an application view.

Table 2-1 Common Service Clients

Client	Description

BEA WebLogic Workshop:
• business processes
• web services
• portals
	Business processes, web services, and portals all access EIS data via the Application View Control, a Workshop control that provides access to an application view and, therefore, to the services defined for the associated EIS. The Application View control allows a business process engineer to browse the hierarchy of application views and to invoke a service as a business process action.
	Synchronous services are represented as simple methods with a single parameter and a non-void return value. For an illustration of synchronous services, see "Processing Synchronous Service Invocations" on page 2-12. Asynchronous services are represented as both a method with a single parameter (the request), and a callback method with a single parameter (the response). For an illustration of asynchronous services, see "Processing Asynchronous Service Invocations" on page 2-13.
	For more information, see the following topics in the BEA WebLogic Workshop Help System:
	• "Integrating Enterprise Applications"
	• "Building Web Services"
	• "Building Portal Applications"
	at the following URL:
	`http://edocs.bea.com/workshop/docs81/doc/en/core/index.html`
	In addition, see "Using Applications With Business Processes" in *Using Application Integration* at the following URL:
	`http://edocs.bea.com/wli/docs81/aiuser/3usruse.html`

Queries and BEA Liquid Data	BEA Liquid Data for WebLogic can access EIS services via application views that are configured as application view data sources. For more information, see "Configuring Access to Application Views" in the *Liquid Data Administration Guide* at the following URL: `http://edocs.bea.com/liquiddata/docs10/admin/appview.html`
Custom Java Applications	Any Java application that uses the Application View client API (in `com.bea.wlai.client`) can invoke services on an application view. For more information, see "Using Application Views by Writing Custom Code" in *Using Application Integration* at the following URL: `http://e-docs.bea.com/wli/docs81/aiuser/4usrcust.html`

Event Consumers

Adapters deliver events using the WebLogic Integration Message Broker, which provides business processes with a channels-based publish and subscribe communication mechanism. For an illustration, see "Processing Event Notifications at Run-Time" on page 2-15. The following table describes common consumers of events from an EIS.

Table 2-2 Common Event Consumers

Client	Description

BEA WebLogic Workshop:
• business processes
• web services
• portals
	Business processes, web services., and portals can subscribe to events published by the Message Broker via the Message Broker Subscription control (or, for business processes only, a static subscription). The Message Broker control listens for application view events. Events can start business processes in which the start node is configured with "Started with a Message Broker Subscription".
	For more information, see the following topics in the BEA WebLogic Workshop Help System:
	• "Message Broker Subscription Control" in "Message Broker Controls"
	• "Integrating Enterprise Applications"
	at the following URL:
	`http://edocs.bea.com/workshop/docs81/doc/en/core/index.html`
	In addition, see "Receiving Events" in "Using Applications With Business Processes" in *Using Application Integration* at the following URL:
	`http://edocs.bea.com/wli/docs81/aiuser/3usruse.html`

Custom Java Applications	Any Java application that uses the Application View client API (in `com.bea.wlai.client`) can consume events. For more information, see "Using Application Views by Writing Custom Code" in *Using Application Integration* at the following URL: `http://e-docs.bea.com/wli/docs81/aiuser/4usrcust.html`

EIS Metadata, Schemas, and Repositories

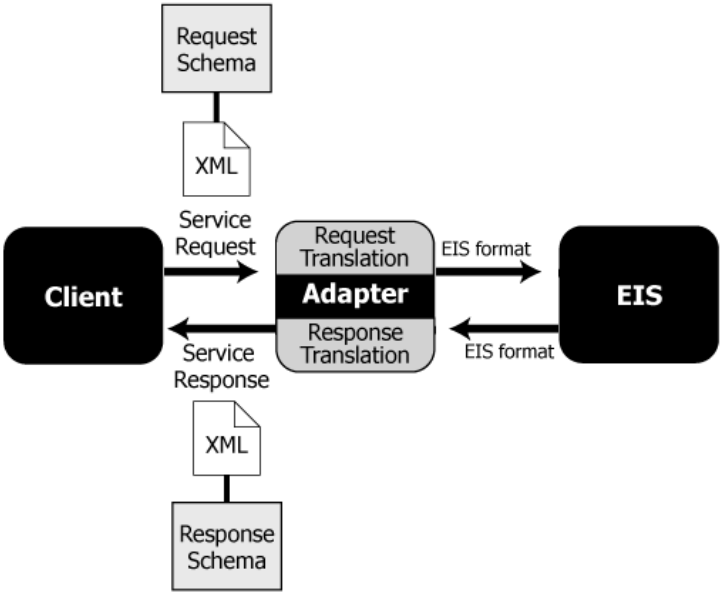
Each EIS uses its own interface to handle service requests and event notifications. For example, SAP provides a BAPI interface that defines the parameters and syntax for BAPI requests and responses. For each EIS, the EIS interface defines the *metadata* that applications can use to integrate with the EIS. The EIS publishes data and expects requests in the format dictated by its interface rules and metadata.

Schemas

At run-time, the EIS and the adapter exchange service requests, service responses, and events via XML documents. The adapter handles the data translation between XML documents and the EIS format, using *schemas* that have been defined at design-time to map the data between XML and the EIS format:

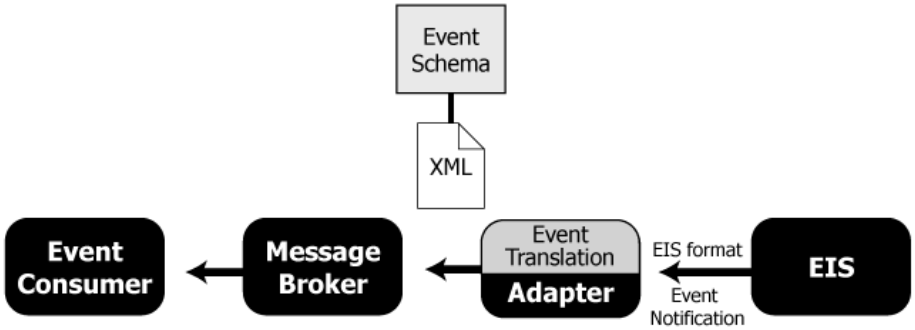
- For service requests, the request arrives at the adapter in the form of an XML document. The adapter uses the *request schema* associated with the service (as defined in the application view) to translate the request to the format that the EIS expects. Similarly, when the adapter receives the response back from the EIS, it uses the *response schema* associated with the service to translate the response to an XML document that the requesting application handles.

Figure 2-3 Adapter Translation for Service Invocations



- For event notifications, the inbound message arrives at the adapter in the format that the EIS uses to publish the event. The adapter uses the *event schema* associated with the event (as defined in the application view) to translate the response to an XML document that the subscribed application handles.

Figure 2-4 Adapter Translation for Event Notifications



At design time, you define a request and a response schema for each service and an event schema for each event that you configure in the application view. For some adapters, such as SAP, you can use the BEA Application Explorer, which is described in "BEA Application Explorer" on page 2-10. For other adapters, you need to create schemas manually. For instructions on how to define schemas for a particular adapter, see the "Generating Schemas" chapter in the *User's Guide* for the adapter(s) that you are using.

Repositories

Once you have created the necessary schemas, you save them in a file-based *repository*, along with a manifest file that associates the schemas with events and services. When you configure application views in the Application View Console, you specify the location of the repository so that the application view can find the schemas as needed. For more information, see the "Defining Application Views" chapter in the *User's Guide* for the adapter(s) that you are using.

Tools for Integration Solutions

This section describes the following tools for designing and deploying integration solutions that involve EIS integration:

- BEA Application Explorer
- Application View Console
- BEA WebLogic Workshop

BEA Application Explorer

The BEA Application Explorer is a design-time tool that you can use to generate schemas for services and events. The BEA Application Explorer incorporates in-depth knowledge of application system environments to query for metadata on specific business objects in the EIS. It uses that metadata to generate the schemas required to build the selected service or event—request and response schemas for services and the event schemas for events. For an introduction to schemas, see "EIS Metadata, Schemas, and Repositories" on page 2-7.

For instructions on how to define schemas for a particular adapter, see the "Generating Schemas" chapter in the *User's Guide* for the adapter(s) that you are using.

Note: For the BEA WebLogic Integration Adapters 8.1, you use the BEA Application Explorer for all adapters *except* the BEA WebLogic Integration Adapter for MQSeries and the BEA WebLogic Adapter for RDBMS.

Application View Console

The Application View Console is a design-time tool that you can use to build application views and configure services and events. For each event or service, the Application View Console allows you to configure connection settings and other relevant information. For an introduction to application views, see "Application Views" on page 2-4.

For instructions on how to create application views using the Application View Console, see the "Defining Application Views" chapter in the *User's Guide* for the adapter(s) that you are using, as well as "Defining an Application View" in "Introduction to Application Integration" in *Using Application Integration* at the following URL:

`http://edocs.bea.com/wli/docs81/aiuser/1usrntr.html`

BEA WebLogic Workshop

BEA WebLogic Workshop is an integrated development environment for building enterprise-class applications on the BEA WebLogic Platform. WebLogic Workshop is both a design-time tool for building business processes, web services, and portals, and a run-time environment for running business processes.

BEA WebLogic Workshop provides the following mechanisms for integrating with EISs:

- The Application View control lets a developer invoke Application View services both synchronously and asynchronously.

- The Message Broker Subscription control lets event consumers subscribe to Message Broker channels and listen for events that the adapter has received from the EIS and has published via the Message Broker.
- The JMS control lets business processes listen to the `WLAI_EVENT_QUEUE` JMS queue and start when an event is received on the applicable `WLAI_EVENT_TOPIC`.

  Note: To start business processes based on events, the start nodes should be configured with "Started with a Message Broker Subscription."

For detailed information, see the BEA WebLogic Workshop Help System at the following URL:

`http://edocs.bea.com/workshop/docs81/doc/en/core/index.html`

Run-Time Processing of Services and Events

This section provides a high-level overview of how adapters process services and events at run-time. It contains the following topics:

- Processing Service Invocations at Run-Time
- Processing Event Notifications at Run-Time

The procedures in this section provide simplified, high-level (non-programmer) descriptions of the process. For sample code, see "Code for Sample Java Class" in "Using Application Views by Writing Custom Code" in *Using Application Integration* at the following URL:

`http://edocs.bea.com/wli/docs81/aiuser/4usrcust.html`

Processing Service Invocations at Run-Time

Service invocations can be either synchronous or asynchronous:

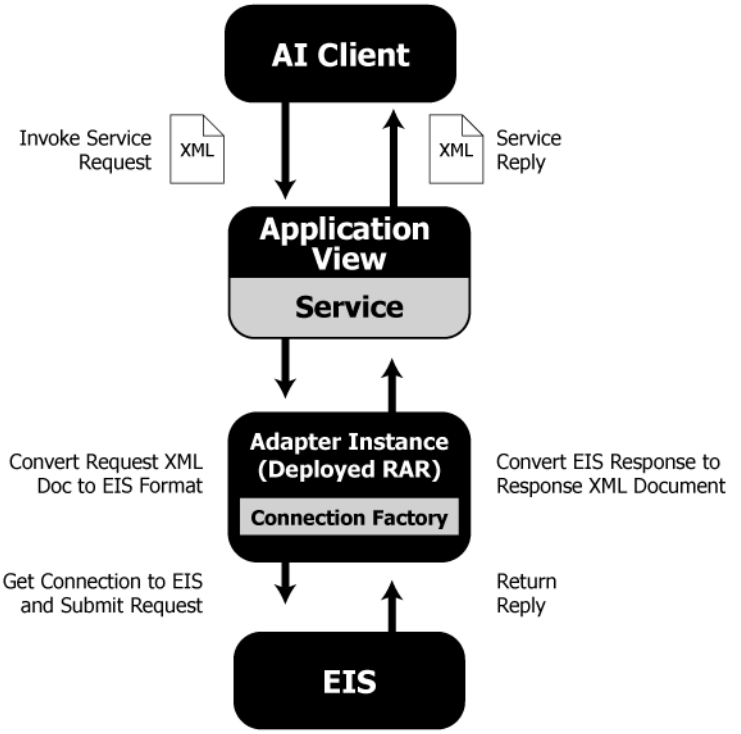
- For *synchronous service invocations*, the client application stops until it receives the response from the EIS.
- For *asynchronous service invocations*, the client application continues, but polls periodically for the response from the EIS or receives the response in a callback method.

The steps for processing service invocations differ when invoked synchronously or asynchronously.

Processing Synchronous Service Invocations

This section walks through the process of a synchronous service invocation at run-time, as shown in the following figure.

Figure 2-5 Run-Time Processing of a Synchronous Service Invocation



The following procedure describes, at a high level, how a synchronous service invocation is processed at run-time:

1. The client application invokes a given service on a given application view (`invokeService` method), specifying the service name, application name, and the request document.

   The client application specifies the response document as the return value to the `invokeService` method.

2. Based on the service invoked, the client instance of the application view obtains a connection to the EIS from the connection factory defined in the adapter instance, and then establishes a connection to the EIS.

3. The client instance of the application view requests that the adapter execute the service request (`execute` method).
4. Upon receiving the request document, the adapter:
   - Translates the request document into the appropriate EIS format using the request schema that was configured for the service.
   - Submits the request to the EIS for processing using the appropriate communications technology for the EIS.
5. The EIS processes the request and returns the response.
6. Upon receiving the response from the EIS, the adapter:
   - Translates the response to the XML format using the response schema that was configured for the service.
   - Returns the response document to the client.
7. The client receives the response document as the return value to the `invokeService` method and processes it accordingly.

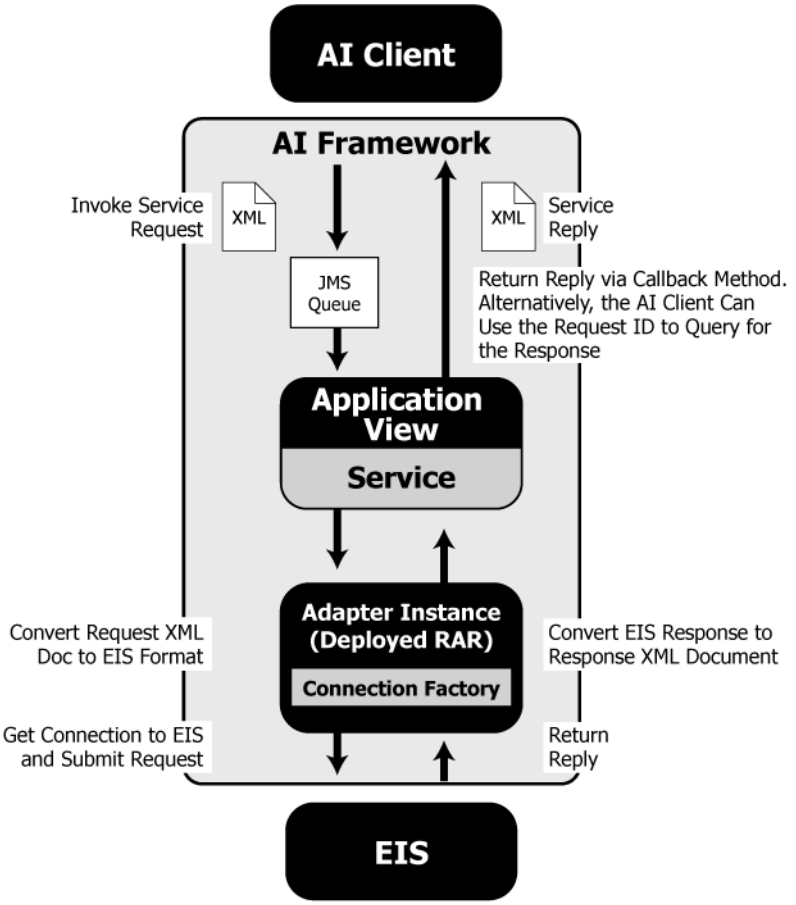
Processing Asynchronous Service Invocations

This section walks through the process of an asynchronous service invocation at run-time. Asynchronous service invocations can be initiated in either of two ways:

- If the service is invoked on a business process using an Application View control, then the service has already been configured for asynchronous invocation. At design time in WebLogic Workshop, while configuring an application view control, you specify an asynchronous invocation by selecting the service in the Services to Invoke Asynchronously list in the Application View Browser window.
- If the service is invoked programmatically, then the client application specifies the request ID as the return value to the method indication. The client application also specifies a callback handler method to match a response with the request ID and to receive the response as a response document.

The following figure shows how an asynchronous service invocation is processed at run-time.

Figure 2-6 Run-Time Processing of an Asynchronous Service Invocation



The following procedure describes, at a high level, how an asynchronous service invocation is processed at run-time:

1. The client application invokes a given service on a given application view, specifying the service name, application name, and the request document.
2. The request document is put into a JMS queue.
3. The Application Integration framework has a message-driven bean that pulls the request document off the JMS queue and invokes the adapter.

4. Upon receiving the request document, the adapter completes the following operations:
   - Converts the request document to the appropriate EIS format.
   - Submits the request to the EIS for processing.
5. The EIS processes the request and returns the response.
6. Upon receiving the response from the EIS, the adapter completes the following operations:
   - Retrieves the request ID and matches responses to the appropriate request ID.
   - Translates the response to the XML format using the response schema that was configured for the service.
7. The response document returns to the client application in either of the following ways:
   - The adapter returns the response document to the client application via the callback method.
   - The client application uses the request ID to query for the response document.
8. The client application receives the response document and processes it accordingly.

Processing Event Notifications at Run-Time

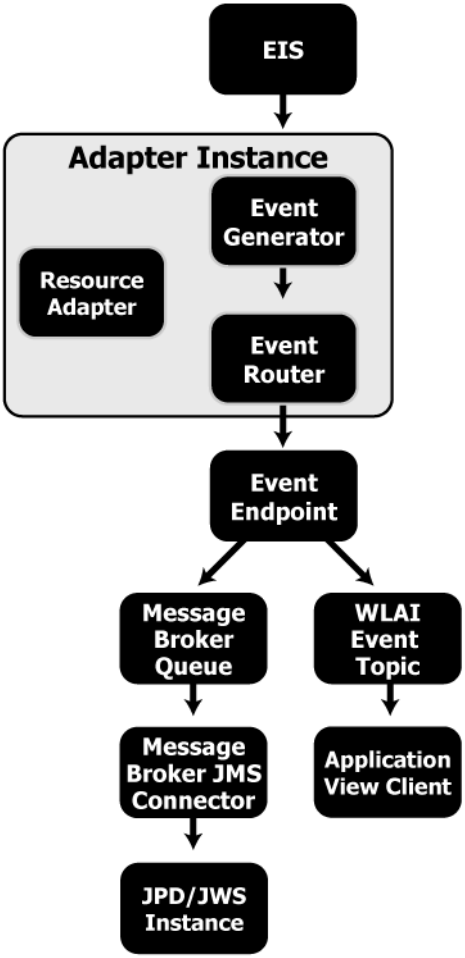
This section walks through the process of an event notification at run-time. Event notifications are always asynchronous and are published to two queues:

- **Message Broker queue**. For business processes (JDP files) or web services (JWS files) defined in WebLogic Workshop. Clients are subscribed to listen for events on specific topics in the Message Broker queue. The Message Broker uses topic files that describe the type of data being published to any given Message Broker topic. The topic file associates a topic name with a schema definition. At design time, when defining the application view, the project associated with the event must be specified. Thereafter, when an application view is deployed, a topic file containing event type information (comprised of the application name, application view name, and event name) is saved in the WebLogic Workshop application directories.
- **Application Integration JMS queue** (`WLAI_EVENT_TOPIC` JMS topic). A message-driven bean (the WLI-AI Event Processor) listens on the `WLAI_EVENT_QUEUE` distributed destination and publishes a copy of the event to the `WLAI_EVENT_TOPIC`.
  The `WLAI_EVENT_TOPIC` is a distributed JMS topic that handles the delivery of events to remote Application View clients.

At design-time, event consumers must be configured to listen for the event on either of these queues. In addition, the EIS must be configured to send event messages to a particular destination so that the adapter can receive it via the EIS-specific communications protocol.

The following figure shows how an event notification is processed at run-time.

Figure 2-7 Run-Time Processing of an Asynchronous Service Invocation



The following procedure describes, at a high level, how an event notification is processed at run-time:

1. The adapter receives the event message from the EIS.
2. The adapter translates the event message to the appropriate XML format using the event schema that was configured for the event.
3. The Event Generator receives the inbound message and then posts it to the Event Router.
4. The Event Router forwards the event message to the Event Endpoint.
5. The Event Endpoint sends the event XML document to two queues:
   - **WebLogic Integration Message Broker** for delivery to any subscribers to the Message Broker topic. Subscribers use either a Message Broker Subscription control or, for business processes only, a Message Broker static subscription.
   - **Application Integration JMS queue** (`WLAI_EVENT_TOPIC`) for delivery to remote or local Application Integration clients who are listening to this topic.
6. Event consumers who have subscribed to the event receive the event XML document and process it accordingly.

CHAPTER 3

Roles, Responsibilities, and Tasks

This section provides an overview of the roles and tasks required to create integration solutions using one or more of the BEA WebLogic Integration Adapters. This section includes the following topics:

- Roles and Responsibilities
- Process for Creating Integration Solutions
- Where To Go From Here

Roles and Responsibilities

The following sections describe the roles that must be fulfilled for members of an integration solution team:

- Application Integration Specialists
- EIS Specialists
- Technology Specialists

A successful integration solution requires input from all of these participants. Depending on the solution, one person may assume multiple roles and all roles might not be required.

Application Integration Specialists

Application integration specialists lead the implementation of an integration solution and drive the design effort. Application integration specialists are knowledgeable about the features and

capabilities of the WebLogic Integration product, particularly the Application Integration component. They consult with EIS specialists to determine requirements, map those requirements to WebLogic Integration features, and design an integration solution's architecture. Integration specialists are responsible for the end-to-end solution and have experience in the following areas:

- Business and technical analysis
- Architecture design
- Project management

EIS Specialists

EIS specialists are experts in the enterprise information systems (EIS) that are part of the integration solution. An EIS specialist provides the information needed to integrate the EIS into the integration solution, including external interfaces, connection protocols, EIS metadata and data formats, and EIS behaviors. EIS specialists are knowledgeable about all aspects of the applicable EIS system and they have experience in the following areas:

- Technical analysis
- Integration solution design
- In-depth knowledge of the organization's EIS deployment and operations

Technology Specialists

Technology specialists are experts in the various technologies used in integration solutions. Examples of technology specialists include:

- Java developers
- database administrators
- system administrators
- infrastructure specialists, such as experts in network, intranet, extranet, and mail infrastructure

Process for Creating Integration Solutions

This section provides a high level, end-to-end view of the process of creating integration solutions that involve EIS integration. It includes the following topics:

- Phase 1: Design the Solution
- Phase 2: Build the Solution
- Phase 3: Deploy and Manage the Solution

This section is a hypothetical or idealized solution designed to showcase product features rather than a suggestion of how to execute a plan. It is intended to supplement any methodologies or processes already used in your organization to build and deploy integration solutions.

Phase 1: Design the Solution

This phase involves two steps: defining the components of the integration solution and creating a detailed, end-to-end design.

Step 1: Define the Components of the Solution

The first phase is to define the components of an integration solution, which includes (but is not limited to) the following tasks:

- Determine which business process(es) will be involved in the integration solution.
- Determine which external EISs and other technologies will be involved in the integration, as well as any external EIS interfaces involved in the business process(es) that you are integrating.
- Determine which WebLogic Platform components will be involved in the integration solution, such as:
  - web services, business processes, or portals designed in WebLogic Workshop
  - BEA Liquid Data for WebLogic
  - custom applications
- Determine which adapters will be required, including BEA WebLogic Integration Adapters and, if applicable, custom adapters. An integration solution can involve multiple adapters.

For more information, see “Components of Integration Solutions Involving EIS Integration” on page 2-2.

Step 2: Create a Detailed, End-to-End Design of the Solution

Once you have defined the components of the solution, you need to create a detailed design that specifies:

- Any service invocations, including:
  - the schemas required for requests and responses (as described in "EIS Metadata, Schemas, and Repositories" on page 2-7)
  - the client application that will initiate each service request and handle the response (as described in "Clients for Service Invocations" on page 2-5)
  - whether each service invocation will be synchronous or asynchronous
  - other requirements that pertain to the associated EISs
- Any event notifications, including:
  - the event schema required (as described in "EIS Metadata, Schemas, and Repositories" on page 2-7)
  - the event consumer(s) that will subscribe to, and listen for, events initiated by the EIS, as described in "Event Consumers" on page 2-7
  - the configuration required on each EIS to publish events to destination where the adapters can receive them
  - other requirements that pertain to the associated EISs
- Any requirements for connecting to the EIS, such as login credentials, network connections, specialized configuration, and so on.
- Any specialized business logic, such as transaction processing.
- Any other components of the integration solution, such as business processes, web services, portals, and so on.

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

Phase 2: Build the Solution

The next phase is to build the solution using the design-time tools described in "Tools for Integration Solutions" on page 2-9. Build tasks include:

- Purchase, install, and configure the WebLogic Platform and any adapters.
- Create the schemas for services and events according to the "Generating Schemas" chapter in your adapter documentation, using the BEA Application Explorer if appropriate for your adapter.
- Create the application views that provide an XML-based interface between WebLogic Server and the EIS. For each application view, you configure connection information, services, and events.
- Build and integrate with other BEA software components as required. For example, you might need to build business processes, web services, or portals in WebLogic Workshop and configure them to invoke services or receive and process event notifications. Similarly, you might need to construct queries in Liquid Data for WebLogic that access application views as data sources. For instructions, see the documentation associated with the BEA software component you are using.
- Test the end-to-end solution, making sure that all of its components interact correctly and produce the desired results.

This phase involves the expertise of technical specialists, such as designers (of business processes, web services, portals, and queries), developers, system integrators, database administrators, EIS specialists, and so on.

Phase 3: Deploy and Manage the Solution

The final phase is to deploy the integration solution in a production environment and monitor its ongoing operation.

- Design the deployment.
- Deploy the required components of the BEA WebLogic Platform.
- Install and deploy the adapter as described in the adapter's *Installation and Configuration Guide*.
- Deploy any application views and schemas for EIS integration.
- Verify business processes in the production environment.

- Monitor, tune, and troubleshoot the deployment.

This phase requires system administrators, network administrators, network operators, and specialists who operate the infrastructure of your organization.

Where To Go From Here

To begin using one of the BEA WebLogic Integration Adapters in your integration solution, refer to the "Getting Started" section of the Introduction in the adapter's *User's Guide*.

Index