



# BEA WebLogic Application Integration

A Component of BEA WebLogic Integration

## Product Overview

BEA WebLogic Application Integration Release 2.0  
Document Edition 2.0  
July 2001

## Copyright

Copyright © 2001 BEA Systems, Inc. All Rights Reserved.

## Restricted Rights Legend

This software and documentation is subject to and made available only pursuant to the terms of the BEA Systems License Agreement and may be used or copied only in accordance with the terms of that agreement. It is against the law to copy the software except as specifically allowed in the agreement. This document may not, in whole or in part, be copied photocopied, reproduced, translated, or reduced to any electronic medium or machine readable form without prior consent, in writing, from BEA Systems, Inc.

Use, duplication or disclosure by the U.S. Government is subject to restrictions set forth in the BEA Systems License Agreement and in subparagraph (c)(1) of the Commercial Computer Software-Restricted Rights Clause at FAR 52.227-19; subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013, subparagraph (d) of the Commercial Computer Software--Licensing clause at NASA FAR supplement 16-52.227-86; or their equivalent.

Information in this document is subject to change without notice and does not represent a commitment on the part of BEA Systems. THE SOFTWARE AND DOCUMENTATION ARE PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND INCLUDING WITHOUT LIMITATION, ANY WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. FURTHER, BEA Systems DOES NOT WARRANT, GUARANTEE, OR MAKE ANY REPRESENTATIONS REGARDING THE USE, OR THE RESULTS OF THE USE, OF THE SOFTWARE OR WRITTEN MATERIAL IN TERMS OF CORRECTNESS, ACCURACY, RELIABILITY, OR OTHERWISE.

## Trademarks or Service Marks

BEA, WebLogic, Tuxedo, and Jolt are registered trademarks of BEA Systems, Inc. How Business Becomes E-Business, Operating System for the Internet, Liquid Data, BEA WebLogic E-Business Platform, BEA Builder, BEA Manager, BEA eLink, BEA Campaign Manager for WebLogic, BEA WebLogic Commerce Server, BEA WebLogic Personalization Server, BEA WebLogic Process Integrator, BEA WebLogic Collaborate, BEA WebLogic Enterprise, BEA WebLogic Server, and BEA WebLogic Integration are trademarks of BEA Systems, Inc.

All other product names may be trademarks of the respective companies with which they are associated.

### **BEA WebLogic Application Integration**

<b>Document Edition</b>	<b>Part Number</b>	<b>Date</b>	<b>Software Version</b>
2.0	N/A	July 2001	2.0

---

# Contents

## About This Document

What You Need to Know .....	v
e-docs Web Site .....	vi
How to Print the Document .....	vi
Related Information .....	vi
Contact Us! .....	vii
Documentation Conventions .....	viii

## 1. Introduction to BEA WebLogic Application Integration

The Application Integration Problem .....	1-1
BEA WebLogic Integration - The End-to-End Integration Solution .....	1-2
BEA WebLogic Application Integration - The EAI Solution .....	1-4
Integration Framework .....	1-4
Application View .....	1-5
Design-Time GUI .....	1-8
Adapter Development Kit (ADK) .....	1-8
Design-time Framework .....	1-9
Run-time Framework .....	1-9
Logging/Auditing Framework .....	1-9
Packaging Framework .....	1-9
Sample Adapters .....	1-10
The BEA WebLogic Application Integration Plug-In for WebLogic	
Process Integrator .....	1-10

## 2. Understanding Adapters

What Is a J2EE-Compliant Adapter? .....	2-1
Service Adapters .....	2-2

---

Event Adapters .....	2-2
Adapters and Their Components .....	2-3
Adapter Design-Time Component .....	2-3
Adapter Run-Time Component .....	2-4
Application Integration-to-EIS (Service Adapter Component) .....	2-4
EIS-to-Application Integration (Event Adapter Component) .....	2-5

### **3. Using BEA WebLogic Application Integration**

Scenarios for Using BEA WebLogic Application Integration .....	3-1
Setting the Stage for the Integration Scenarios .....	3-2
Integration Within An Enterprise .....	3-2
The Problem .....	3-2
The Solution with BEA WebLogic Application Integration: As a Component of BEA WebLogic Integration .....	3-3
Understanding the Integration Solution Design .....	3-4
e-Commerce Integration .....	3-5
The Problem .....	3-6
The End-to-End Solution with BEA WebLogic Integration (and Application Integration) .....	3-6
Understanding the End-to-End Integration Solution Design .....	3-8
Roadmap for Using BEA WebLogic Application Integration .....	3-9

---

# About This Document

*BEA WebLogic Application Integration Product Overview* is organized as follows:

- “Introduction to BEA WebLogic Application Integration” provides an overview of BEA WebLogic Application Integration. It provides detailed discussion of all major concepts, features, and component architecture for Application Integration.
- “Understanding Adapters” provides information about J2EE-compliant adapters and their components.
- “Using BEA WebLogic Application Integration” provides two scenarios for using and a roadmap for using BEA WebLogic Application Integration.

## What You Need to Know

This document is intended for the following users:

- **Technical Analyst**—responsible for configuring integration solutions and administering a WebLogic Integration environment, including application server, application integration, business process modeling, and B2B integration. This guide assumes that the technical analyst has knowledge of the entire system from end to end.
- **Business Analyst**—works with the technical analysts to ensure accuracy of the business interface functionality, to create application views, and to use application views within an enterprise.

---

# e-docs Web Site

BEA product documentation is available on the BEA corporate Web site. From the BEA Home page, click 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 Application Integration 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 Application Integration 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 resources are also available:

- BEA WebLogic Server documentation (<http://www.edocs.com>)
- BEA WebLogic Process Integrator documentation (<http://www.edocs.com>)
- XML Schema Specification (<http://www.w3.org/TR/xmlschema-1/>)
- The Sun Microsystems, Inc. J2EE Connector Architecture Specification (<http://java.sun.com/j2ee/connector/>)

- 
- The Sun Microsystems, Inc. Java site (<http://www.javasoft.com/>)

## Contact Us!

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

In your e-mail message, please indicate that you are using the documentation for the BEA WebLogic Application Integration 2.0 release.

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

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

- Your name, e-mail address, phone number, and fax number
- Your company name and company address
- Your machine type and authorization codes
- The name and version of Application Integration 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.

<b>Convention</b>	<b>Item</b>
<b>boldface text</b>	Indicates terms defined in the glossary.
<b>Ctrl+Tab</b>	Indicates that you must press two or more keys simultaneously.
<i>italics</i>	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. <i>Examples:</i> chmod u+w * c:\startSever .doc wls.doc BITMAP float
<b>monospace boldface text</b>	Identifies significant words in code. <i>Example:</i> void <b>commit</b> ( )
<i>monospace italic text</i>	Identifies variables in code. <i>Example:</i> String <i>expr</i>
UPPERCASE TEXT	Indicates device names, environment variables, and logical operators. <i>Examples:</i> LPT1 SIGNON OR



---

Convention	Item
...	Indicates one of the following in a command line: <ul style="list-style-type: none"> <li>■ That an argument can be repeated several times in a command line</li> <li>■ That the statement omits additional optional arguments</li> <li>■ That you can enter additional parameters, values, or other information</li> </ul> <i>Example:</i> <pre>import com.sap.rfc.exception.* . . . ;</pre>
. . .	Indicates the omission of items from a code example or from a syntax line. The vertical ellipsis itself should never be typed.

---



# 1 Introduction to BEA WebLogic Application Integration

BEA WebLogic Application Integration provides a WebLogic-based integration solution that supports existing and forthcoming standards for connecting applications both within and between enterprises. BEA WebLogic Application Integration provides a means to define communication endpoints, which may be included in a process flow using BEA WebLogic Process Integrator or using custom Java code to form a complete integration solution.

This guide provides information on the following subjects:

- The Application Integration Problem
- BEA WebLogic Integration - The End-to-End Integration Solution
- BEA WebLogic Application Integration - The EAI Solution

## The Application Integration Problem

Historically, enterprise information systems such as packaged applications (ERP, CRM, SCM) and legacy mainframe systems have been the center of the modern IT environment, driving enterprise-wide business functions and providing critical data for enterprise operations. However, these applications were designed to meet specific functional requirements, not to support complex e-business processes that span

multiple applications and departments within and beyond the enterprise. To support today's business requirements, application integration projects are needed to link these enterprise systems not only with one another, but with the Web and wireless applications that provide the ability to share real-time information and services with prospects, customers, and suppliers around the world.

Having already invested heavily in "best-of-breed" business applications, corporations are demanding that seamless bridges be built between existing business applications to bind them into a single, unified enterprise application. Enterprise application integration (EAI) provides a means for the unrestricted sharing of data and business processes among any connected application and data sources in the enterprise without the necessity of changing the original applications or data structures.

The widespread adoption of standards such as Java 2 Enterprise Edition (J2EE) and Extensible Markup Language (XML) have laid the groundwork for an application integration solution that can provide support for business processes, management, and integration within the enterprise and provide support for business process management beyond the firewalls of the enterprise.

BEA WebLogic Application Integration, a component of BEA WebLogic Integration, builds upon standards such as J2EE and XML to provide the application integration solution. Working in conjunction with the other components of BEA WebLogic Integration, BEA WebLogic Application Integration helps to provide an end-to-end integration solution. For more information about BEA WebLogic Integration, see "BEA WebLogic Integration - The End-to-End Integration Solution".

## **BEA WebLogic Integration - The End-to-End Integration Solution**

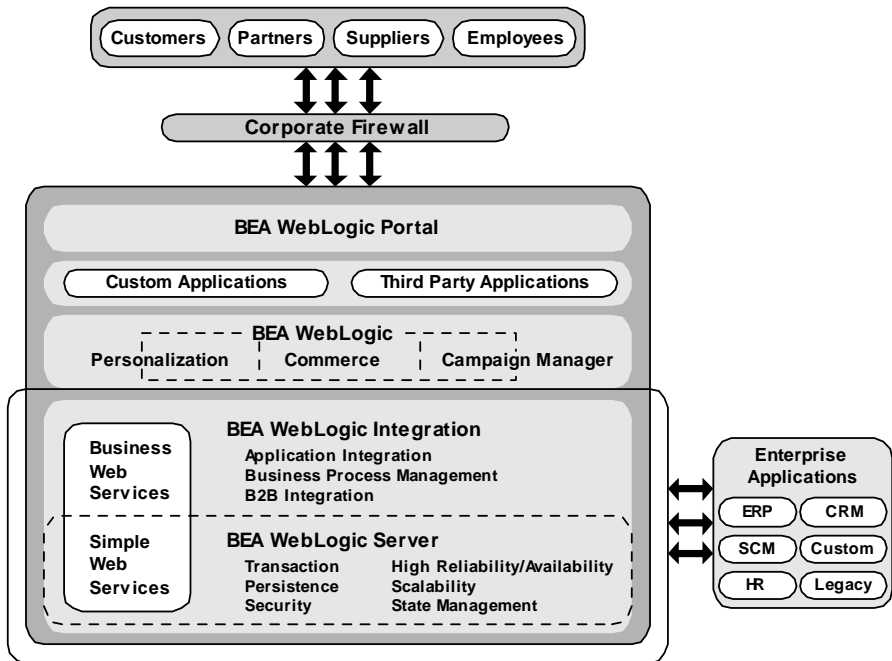
BEA WebLogic Integration provides a single, standards-based platform that enables organizations to develop, deploy, and integrate applications and business processes within the enterprise and across the extended enterprise to suppliers, distributors, and other business partners. It delivers application server, application integration, business process management, and B2B integration functionality that supports a "build to integrate" approach to enterprise application life-cycle management.

The goal of BEA WebLogic Integration is to speed development and deployment efforts, minimize the pain of integration and provide a lower total cost of ownership for IT Application server, BEA WebLogic Server. BEA WebLogic Integration is comprised of the following components:

- BEA WebLogic Process Integrator - to provide business process management
- BEA WebLogic Collaborate - to provide B2B integration
- BEA WebLogic Application Integration - to provide enterprise application integration
- BEA WebLogic XML/Non-XML Translator - to provide XML to non-XML translation capabilities

Figure 1-1 provides a graphical overview of BEA WebLogic Integration. For more information about BEA WebLogic Integration, see the *BEA WebLogic Integration Product Overview*.

**Figure 1-1 BEA WebLogic Integration**



# BEA WebLogic Application Integration - The EAI Solution

BEA WebLogic Application Integration, as a component of BEA WebLogic Integration, leverages the standards provided by J2EE specifications and Java Message Services (JMS) to provide a robust and scalable integration platform for integrating applications such as Enterprise Resource Planning (ERP) and Customer Relationship Management (CRM) with the Web and wireless applications using adapters. Adapters are software components that provide an interface between an application and an integration server. BEA WebLogic Application Integration also supports consistent data representation via XML.

BEA WebLogic Application Integration provides the tools and framework for creating custom adapters that can expand the reach of the application integration solution to any application in your enterprise. Application Integration provides tools that allow you to expose an application's functionality and data formats in a J2EE standard-based way so they can be used by all clients of BEA WebLogic Integration.

As part of the complete BEA WebLogic Integration solution, BEA WebLogic Application Integration is integrated and interoperable with the application server, business process management and B2B integration components of the solution.

BEA WebLogic Application Integration consists of two key components:

- Integration Framework
- Adapter Development Kit (ADK)
- Sample Adapters
- The BEA WebLogic Application Integration Plug-In for WebLogic Process Integrator

## Integration Framework

The Integration Framework component provides a systematic, standards-based architecture for hosting business-oriented interfaces to enterprise applications.

The features provided in Integration Framework include:

- Application View
- Design-Time GUI

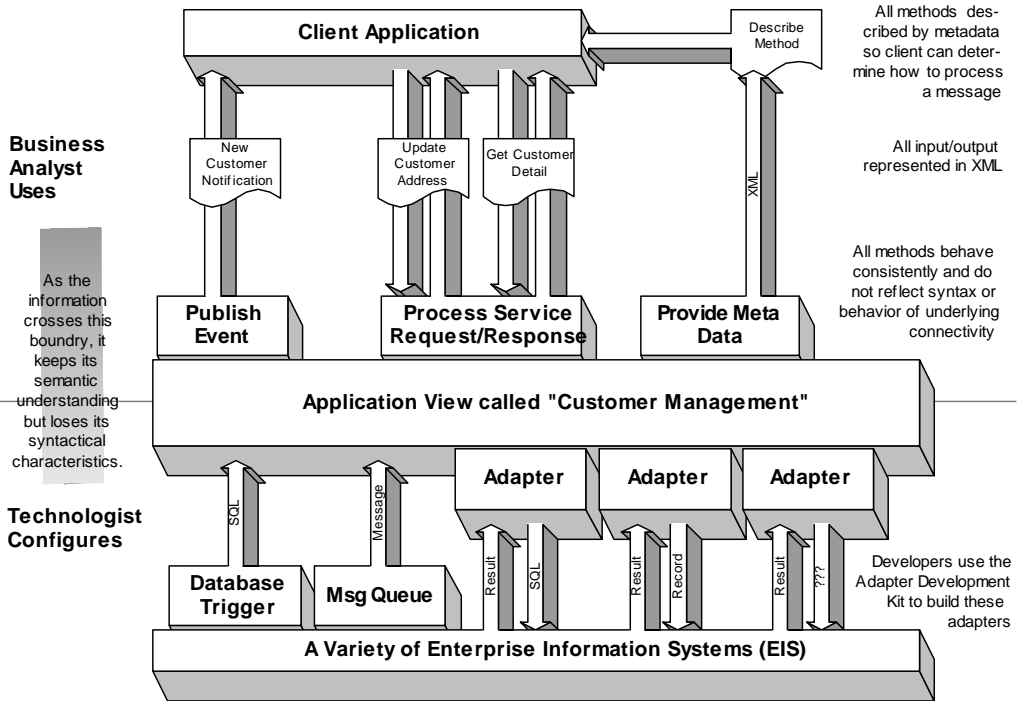
## Application View

The *application view* is the cornerstone of BEA WebLogic Application Integration's Integration Framework. It provides a *view* of the application capabilities exposed by an adapter that a user can customize to meet specific business needs. A user tailors an application view for a specific business purpose, and as a result, the application view provides an effective alternative to the "one size fits all" approach that most applications provide for the design of their client interface. The application view allows you to define for it only the business capabilities that directly apply to your business purpose. You can customize the capabilities by naming, describing and defining their data requirements.

The application view also provides the abstraction layer between the technical implementation of an adapter and the business usage of the capabilities exposed by the adapter. The application view is a standardized, self-describing interface to an application.

The business-level view of an application's capabilities provides a logical separation between the programmer and the business analyst. For example, this abstraction enables a business analyst to create records in a database without having to know SQL. Figure 1-2 provides a diagram of the application view at work within the BEA WebLogic Application Integration environment.

**Figure 1-2 The Application View**



The application view enables the business user to define integration points by browsing fields and data of a specific application. For example, by defining application views, a user could define integration endpoints for a customer management system, sales, order management system, or purchase order management system. Users can create an application view for each business use and deploy and manage these applications views using the Integration Framework component. The application view provides the following features:

- Consistent Data Representation with XML
- Events and Services
- Self Describing Metadata
- Common Protocol
- Bidirectional Communication Support



## Consistent Data Representation with XML

The application view requires an XML request message and provides an XML response message for each service. Registered event listeners also receive event information as XML. The application view relies on the adapter to perform the translation of the EIS specific format to and from XML. Because of this, the business user does not need to understand the language of the application. The application view provides a common view in XML for all Enterprise Information Systems (EIS) for which an appropriate adapter is available.

## Events and Services

The application view, via an underlying adapter, supports events and services for a particular business use. *Events* enable messages generated by an application to be managed following a publish/subscribe model. *Services* act as business functions that may be invoked by a user. Service invocations cause messages to be sent to an application following the request/response model. Both events and service requests/responses are passed through the system as XML documents. For more information about events and services, see “Events and Services.”

## Self Describing Metadata

The application view uses an XML schema as metadata to describe the XML information for events, service requests and responses. This metadata helps a user to understand the data requirements of the event or service of interest.

## Common Protocol

A common protocol makes application integration possible because each application needs to be integrated only with the common protocol rather than with every other application. With a common protocol, all applications communicate using a standard language, in this case, XML.

BEA WebLogic Application Integration implements a common protocol based on events and services that are represented by XML documents. Adapters expose an application’s functionality and data formats as events and services in an XML format via the application view so they can be used by all clients of BEA WebLogic Integration.

## Bidirectional Communication Support

The J2EE specification does not currently provide guidelines for an EIS to initiate communication with the application server or client. BEA WebLogic Application Integration provides this capability via the event adapter. For more information about event adapters, see “Event Adapters”.

## Design-Time GUI

The Integration Framework component of BEA WebLogic Application Integration provides a design-time GUI, provides a way for you to create application views which add services and events to application views, and to deploy, test, and edit application views.

The Application View Management Console is available in the GUI to help you access, organize, and edit all application views in your enterprise. You can use the Application View Management Console to create new folders and to add new application views to the folders. These folders allow you to organize your application views according to your own navigation scheme, regardless of the adapter the application view belongs to.

For detailed information about creating, using, and managing application views via the BEA WebLogic Application Integration design-time GUI, see the *BEA WebLogic Application Integration User Guide*.

## Adapter Development Kit (ADK)

The Adapter Development Kit (ADK) facilitates the construction of adapters that will run in any J2EE-compliant environment. These adapters can have additional, value-added functionality when hosted within the Integration Framework component of BEA WebLogic Application Integration.

The ADK provides tools to allow adapter developers to create bidirectional adapters that support the business-level concepts of events and services. An adapter developed using the ADK can provide access to an application’s metadata, which can then be used to define events and services (via the application view) The ADK provides run-time support for events and services. It also provides support constructing thin, JSP-based user interfaces for use in defining application views based on the adapter.

The ADK provides a collection of frameworks to support the development, testing, packaging, and distribution of adapters:

- Design-time Framework
- Run-time Framework
- Logging/Auditing Framework
- Packaging Framework

## **Design-time Framework**

The design-time framework provides a tool set to support the development of a user interface for an adapter. The user interface may be used for defining and deploying application views. This is a powerful feature because it makes the adapter capabilities easier to use, giving the adapter a wider audience.

## **Run-time Framework**

The ADK run-time framework provides a complete, extensible event generator to support the development of event adapters. To support the development of service adapters, the runtime framework provides a complete, but minimally functional J2EE-compliant adapter. Adapter developers save time coding and debugging time by starting with and extending this base framework.

## **Logging/Auditing Framework**

Logging is an essential feature of an adapter. When an adapter encounters an error or warning condition, it cannot stop processing and wait for an end-user to respond, so it must provide an accurate audit log for tracking errors. The ADK logging framework gives adapter developers the ability to log internationalized and localized messages to multiple output destinations.

## **Packaging Framework**

The ADK packaging framework provides a tool set for packaging an adapter for delivery to a customer. This framework makes the creation of various archive and environment files easy for adapter developers.

## Sample Adapters

BEA WebLogic Application Integration provides two J2EE-compliant sample adapters built with the BEA WebLogic Application Integration Adapter Development Kit (ADK): the DBMS sample adapter and the Email sample adapter. These adapters provide concrete examples for adapter providers of how one might use the ADK to construct an adapter.

For the DBMS adapter, a relational database was used as the EIS because it allows adapter providers to focus on the adapter/ADK specifics, rather than become bogged-down in understanding a particular proprietary EIS. For more information about the DBMS sample adapter, see “Introduction to the DBMS Sample Adapter.”

For the Email adapter, Java mail was used as the EIS because it allows adapter providers to focus on the adapter/ADK specifics. For more information about the eMail sample adapter, see “Introduction to the eMail Sample Adapter.”

## The BEA WebLogic Application Integration Plug-In for WebLogic Process Integrator

WebLogic Process Integrator provides a GUI-based environment for designing business process workflows. The BEA WebLogic Application Integration plug-in for BEA WebLogic Process Integrator provides a way to use application views created with BEA WebLogic Application Integration in Process Integrator workflows. For more information about using the plug-in, see “Using Application Views in WebLogic Process Integrator.”

# 2 Understanding Adapters

BEA WebLogic Application Integration provides a WebLogic-based integration solution that supports existing and forthcoming standards for integrating applications.

This guide provides information on the following subjects:

- What Is a J2EE-Compliant Adapter?
- Adapters and Their Components

## What Is a J2EE-Compliant Adapter?

An adapter is a software component that provides an interface between an application and an integration server. A J2EE adapter is an adapter that conforms to the J2EE specification as defined by Sun. Such an adapter can be hosted in any J2EE-compliant application server. This allows a simple adapter to be used in a wide range of software environments.

The advent of standards for adapters will also encourage developers to write a wider variety of adapters, thus providing more choices for implementing applications that make use of one or more EIS. An adapter developed using the Application Integration ADK and running in the BEA WebLogic Application Integration environment can provide value-added features that go beyond those of J2EE Connector Architecture. These features include the ability to generate events and to expose XML interfaces to services.

Adapters developed with the Application Integration ADK can also run outside of the WebLogic Server environment, but they will not have the ability to generate events and expose application view interfaces. For information about adapter components, see “Adapters and Their Components”.

BEA WebLogic Application Integration provides the tools to build and use two types of adapters:

- Service Adapters
- Event Adapters

## Service Adapters

Service adapters are J2EE adapters with extended capabilities. Service adapters receive an XML request document from a client and invoke a specific function in the underlying EIS. They are consumers of XML and may or may not provide a response. There are two types of service adapters: asynchronous and synchronous. Service adapters perform the following four functions:

- Receiving service requests from a client (an XML document) as input.
- Transforming the XML request document into the EIS-specific format. The request document conforms to the request XML schema for the service. The request XML schema is based on metadata in the EIS.
- Invoking the underlying function in the EIS and wait for its response.
- Transforming the response from the EIS-specific data format to an XML document that conforms to the response XML schema for the service. The response XML schema is based on metadata in the EIS.

## Event Adapters

Event adapters are designed to propagate information from an EIS into the Application Integration environment. This type of adapter can be described as a publisher of information.

There are two basic types of event adapters: in-process and out-of-process. In-process event adapters execute within the same process as the EIS. Out-of-process adapters execute in a separate process.

The ADK implements basic event generation support capabilities in the form of base Java classes that may be extended. This allows adapter developers to focus on the EIS-specific aspects of the adapter. Within the BEA WebLogic Application Integration environment, all event adapters perform the following three functions:

- Responding to “events” that occur inside the running EIS and extract data about the event from the EIS into the adapter.
- Transforming event data from the EIS specific format to an XML document that conforms to the XML schema for the event. The XML schema is based on metadata in the EIS.
- Propagating the event into the Application Integration environment that delivers it to one or more clients.

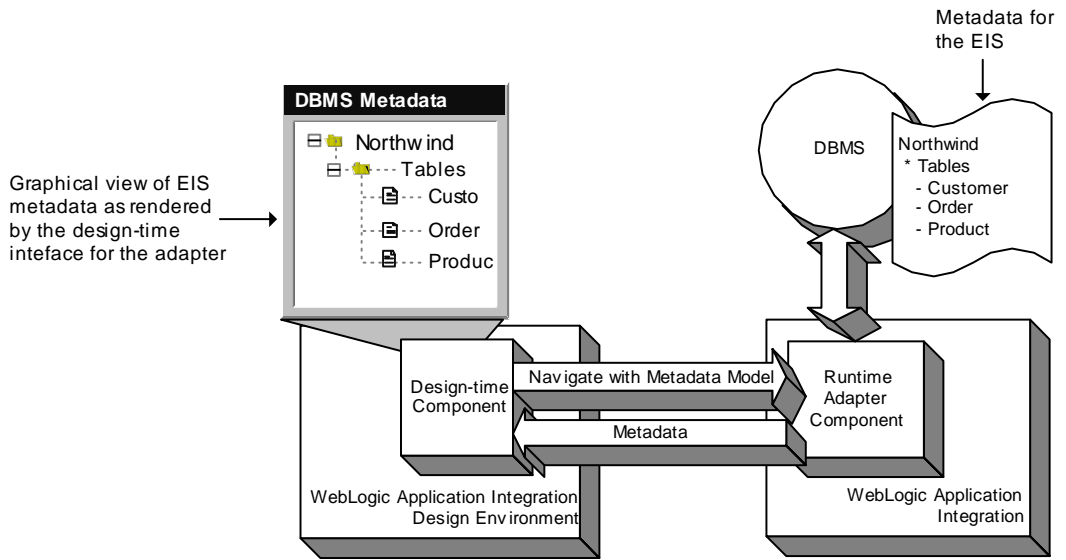
## Adapters and Their Components

An adapter is composed of two major components. The first component is called the design-time component, which leverages web interfaces to enable metadata browsing and definitions of application views, events, and services. The second component is called the run-time component. At runtime it handles generation of events and the invocation of system-level functions in response to requests by the application view.

### Adapter Design-Time Component

The design-time component of an adapter is implemented as a series of Java Server Pages (JSP) pages. The ADK provides many of these JSPs, which an adapter developer can extend. This component is responsible for presenting a view of the metadata within the EIS instance. The design-time component is connected to the view by making use of the adapter’s run-time component to communicate with the EIS. Figure 2-1 shows the design-time component and its relationship to the run-time adapter component.

**Figure 2-1 Adapter Design-Time Component**



## Adapter Run-Time Component

An adapter is capable of supporting both Application Integration-to-EIS communication, in the form of services, and EIS-to-Application Integration communication in the form of events. These two types of communication are handled by two distinct software components within the adapter:

- Application Integration-to-EIS (Service Adapter Component)
- EIS-to-Application Integration (Event Adapter Component)

### Application Integration-to-EIS (Service Adapter Component)

The software component of an adapter that is responsible for handling service invocations, called the service adapter component, is an industry standard, J2EE specification-compliant adapter. The J2EE engine hosts this component and manages physical connections to the EIS, login authentication, and transaction management as specified by the J2EE standard.



## **EIS-to-Application Integration (Event Adapter Component)**

The software component of an adapter that is responsible for handling event generation and delivery is called the EIS-to-Application Integration adapter component. This software component leverages industry standard JMS technologies to allow an EIS to notify interested parties of events within itself. This capability is not addressed in the J2EE specification is unique to Application Integration. The event adapter component is composed of an event generator component and an event router component.

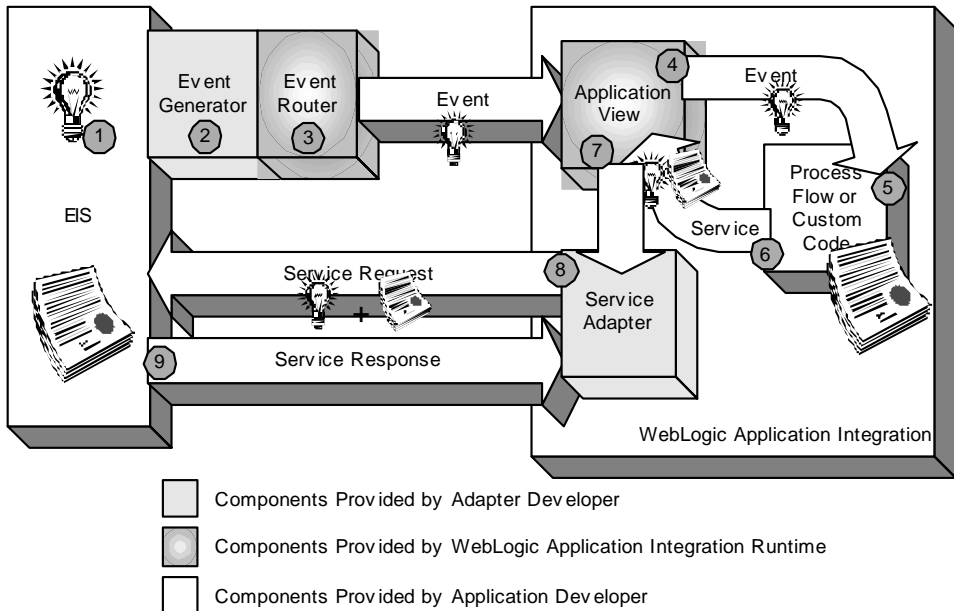
### Event Generator

The event generator, which is specifically designed for each EIS, is responsible for detecting events within the EIS and generating an event object that reflects the event and the context in which it occurred. The event generator then passes the event object to the event router for delivery to clients subscribed.

### Event Router

The event router is a generic software component provided by the Integration Framework component that manages subscriptions to event notifications by subscribed clients and handles the delivery of event objects to those clients when event objects of a given type are published by the event generator. Event subscribers are application views that subscribe to the event types that have been configured within them. The event generator is capable of delivering a single event object to any number of subscribing application views in any number of Application Integration instances. This allows a single EIS system to supply event notifications to multiple Application Integration instances.

**Figure 2-2 Adapter Run-time Component**



This figure illustrates the following steps:

1. Some event occurs within the EIS.
2. The event generator detects the event and checks to see if anyone is subscribed to this type of event. To do this, the event generator checks the types of events that have subscriptions, as maintained by the event router. The event generator creates an event object containing pertinent data about the event and sends it to the event router.
3. The event router sends the event object to all application views that are subscribed to the type of event object it received from the event generator.
4. The application view receives the event object, and notifies any listeners that have registered for the type of event. It is assumed that a WebLogic Process Integrator process flow or some custom business logic code is listening for the designated type of event.
5. The process flow or custom code processes the event object and its data and determines that it should invoke a service in response to the event.

6. The application view combines the data from the event object with data it has in its state variables and makes the service invocation.
7. The application view accepts the service request and determines which interaction to request on the resource adapter. This mapping from service name to adapter interaction allows the service to be business-centric and not system function-centric.
8. The resource adapter carries out the request interaction with the EIS and passes the data it received from the application view.

The EIS processes the request and responds to the adapter. This response propagates back out through the application view and is received by the process flow or custom code component. This component may take further action based on the response.



# 3 Using BEA WebLogic Application Integration

BEA WebLogic Application Integration provides a WebLogic-based integration solution that supports existing and forthcoming standards for connecting applications.

This guide provides information for the following subjects:

- Scenarios for Using BEA WebLogic Application Integration
- Roadmap for Using BEA WebLogic Application Integration

## Scenarios for Using BEA WebLogic Application Integration

This section provides two scenarios for integrating applications with BEA WebLogic Application Integration. The first scenario demonstrates the benefits of Application Integration when integrating internal enterprise applications. The second scenario demonstrates the benefits of the end-to-end WebLogic integration solution for B2C selling and B2B collaboration on the Web.

This section provides information on the following topics:

- Setting the Stage for the Integration Scenarios
- Integration Within An Enterprise
- e-Commerce Integration

## Setting the Stage for the Integration Scenarios

CrankItUp.com is a fictitious company that provides a vast product catalog of mountain bike components on the Web. CrankItUp.com is a customer-centric business that handles product orders for several distributors that do not have a strong Web presence. Its strategy is to provide a highly personalized portal system in which partners and customers can efficiently carry out a variety of integrated services online, such as purchasing products, checking order status, and obtaining return authorizations.

CrankItUp.com adopts a best-of-breed approach to its information processing needs. It uses a customer relationship management system (CRM) to manage the primary information about a customer. For enterprise resource planning (ERP), they use an application that leverages the logistics and financial modules of a well known ERP system.

## Integration Within An Enterprise

Customer records are created by CrankItUp.com's CRM application, but the order processing application requires knowledge of these customers in order to fulfill sales order requests (for example, to verify product availability, credit, and shipping information). Because CrankItUp.com has chosen to use a particular CRM application as the primary repository for customer information, they need to propagate customer information from their CRM system to their order processing system, thus integrating these two systems.

### The Problem

CrankItUp.com's integration problem is twofold: First, the CRM application and the order processing application were not designed to communicate with each other. Second, each application uses a different format to represent a customer. To integrate the two applications, they need a middleware solution that can solve both dimensions of the integration problem.

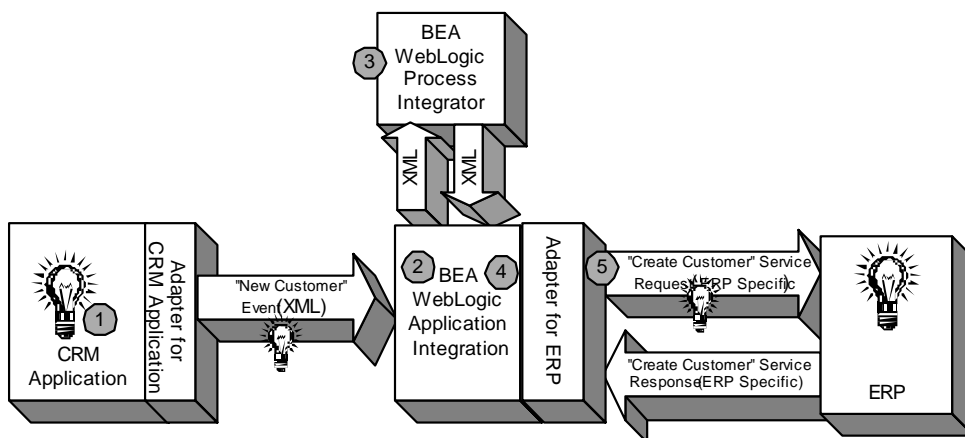
## The Solution with BEA WebLogic Application Integration: As a Component of BEA WebLogic Integration


To enable the integration solution, CrankItUp.com employs several BEA WebLogic Integration software components:

- **BEA WebLogic Server** to provide the platform on which the integration solution will run.
- **BEA WebLogic Application Integration** to enable the use of the CRM and ERP as end points of an integration solution.
- **BEA WebLogic Process Integrator** to allow the endpoints of the integration to be tied together into a working business process.
- **Adapters** developed with the Application Integration Adapter Development Kit for both the CRM system and the order processing system. The software components plug in to Application Integration and allow clients of Application Integration to communicate with an otherwise incompatible system.

Figure 3-1 illustrates how the CRM and order processing application are integrated using these components:

**Figure 3-1 Solution with BEA WebLogic Server**



 New Customer event within the CRM application (Information about this event is propagated by an event object)

### 3 *Using BEA WebLogic Application Integration*

---

- 1  
A customer is created within the CRM system. The CRM adapter generates an event object that contains information (in XML format) about the customer that was just created.
- 2  
An application view within BEA WebLogic Application Integration receives the “New Customer” event and passes the event object on to BEA WebLogic Process Integrator for processing. For detailed information about application views, see “Application View.”
- 3  
BEA WebLogic Process Integrator hosts a business process that is started upon receipt of a “New Customer” event. The business process is configured to push the new customer data out to the order processing application by calling a “CreateCustomer” service on an application view. For detailed information about application views, see “Application View.”
- 4  
The “CreateCustomer” service represents the “CreateCustomer” functionality within the order processing application. As input, this service takes XML data representing a customer and returns a response in XML indicating the success or failure of the operation. Process Integrator passes the XML data received from the “New Customer” event to the service. The service then uses the order processing application adapter to invoke the correct functionality within that application.
- 5  
The order processing application adapter communicates with the order processing application to create a new customer and returns an XML document indicating the results.

## **Understanding the Integration Solution Design**

This first solution was designed using tools provided with Application Integration and Process Integrator. If CrankItUp.com’s business requirements change in the future, the designers can change this integration solution using BEA WebLogic Integration tools without modifying source code.



BEA WebLogic Integration stores the configuration of an integration solution within its repository, thus allowing this information to be retrieved at runtime. The configuration information controls the execution of the integration solution within the BEA WebLogic Integration environment.

A business analyst uses the BEA WebLogic Integration design environment to design the integration solution:

1. Define the endpoints of the integration solution related to the CRM application.

Because the CRM system is the source system in this scenario, the endpoint is defined in terms of events. The business analyst defines the events of interest within the CRM system. The event data is defined in terms of XML schema.

2. Define the endpoints of the integration solution related to the order processing application.

Because the order processing application is the target system in this scenario, this end point is defined in terms of services. The business analyst defines the services that will handle order-related requests to the order processing system. Each service's request and response data is defined in terms of XML schema.

3. Define the Process Integrator business process to tie together the CRM and order processing endpoints.

The business process makes use of the XML schemas that define the events and the request/response data for services to ensure that data from incoming events is properly mapped to the request data for services.

## e-Commerce Integration

To build customer loyalty, CrankItUp.com focuses on providing excellent customer service and improving its Web site. As a result, CrankItUp.com cannot store large inventories of their products. Instead, they must optimize their supply-chain through collaborative supply chain-planning activities with their business partners to reduce their inventory and operating costs.

## The Problem

CrankItUp.com's problem is that it must maintain visibility into the supply chain and assurance of fulfillment and delivery. First, systems that enable business-to-business (B2B) e-commerce transactions must support varying levels of traffic and throughput. They must be 100% reliable and must ensure the integrity and security of all interactions. Second, the collaboration systems must be flexible to support changing processes, changing partnerships, and changing business models. Integration with business partners' systems requires a solution built on open industry standards such as XML, Java, and HTTP.

## The End-to-End Solution with BEA WebLogic Integration (and Application Integration)

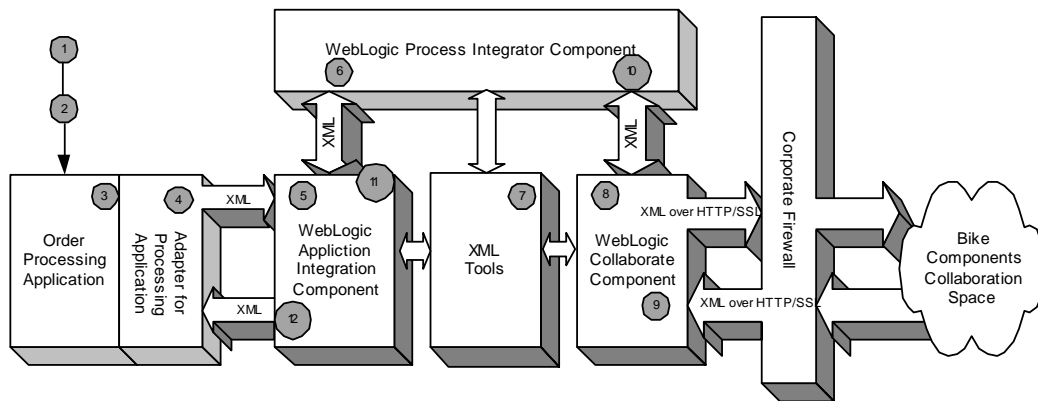
To enable the integration solution, CrankItUp.com employs several BEA WebLogic Integration software components:

- **BEA WebLogic Server** to provide the platform on which the integration solution will run.
- **BEA WebLogic Application Integration** to enable the use of the CRM and ERP as end points of an integration solution.
- **BEA WebLogic Process Integrator** to allow the end points of the integration to be tied together into a working business process.
- **BEA WebLogic Collaborate** to allow the use of remote collaboration partners as end points of an integration solution.
- **Adapters** developed with the Application Integration Adapter Development Kit for both the CRM system and the order processing system. The software components plug in to Application Integration and allow clients of Application Integration to communicate with an otherwise incompatible system.

**Note:** Adapters written without the help of the BEA WebLogic Application Integration ADK can still be used with BEA WebLogic Application Integration as long as they meet certain requirements (XCCI interface).

Figure 3-2 illustrates how the end-to-end solution enables full integration with business partners for the Internet:

Figure 3-2 The End-to-End Solution



- 1 A customer requests to purchase a new crank set from the CrankItUp.com Web site. The request is sent to the order processing application.
- 2 The sales order is processed by the order processing application. CrankItUp.com has an optimized supply chain, and the product is available for immediate shipping.
- 3 The order processing application analyzes the current inventory for this product and finds only 60 crank sets left. Based on historical demand statistics, the current inventory will last five more days. This initiates a “Restock Product” event in the order processing application. This “Restock Product Event” contains detailed information on the product, number of units required, and the date for which the units are required.
- 4 The order processing adapter generates an event object that contains information (as XML) about the “Restock Product” event that just occurred.
- 5 An application view within Application Integration receives the “Restock Product” event and passes the event object on to Process Integrator for processing.

- 6 Process Integrator has been configured to respond to “Restock Product” events. When a “Restock Product” event occurs, Process Integrator initiates a new business process.
- 7 The first step in the business process is to transform the “Restock Product” event information from the order processing application XML format to the standard XML vocabulary (for example: RosettaNet, BizTalk) established between CrankItUp.com and its suppliers. Process Integrator leverages a standard eXtensible Style Language Transformation (XSLT) engine to perform the transformation.
- 8 When the event data transformation is complete, Process Integrator sends the transformed XML document to WebLogic Collaborate.
- 9 WebLogic Collaborate sends the “Restock Product” information securely over the Internet to all subscribers of the CrankItUp.com collaboration space. WebLogic Collaborate coordinates responses from suppliers and returns them to Process Integrator.
- 10 As Process Integrator receives responses from suppliers via WebLogic Collaborate, it evaluates the responses and chooses the optimal response. In this case, the optimal response is a function of the supplier’s ability to deliver within five days and the lowest price per unit.
- 11 Once Process Integrator determines the optimal response, it calls Application Integration to invoke an application view implemented for order processing.
- 12 The order processing application view now invokes a service called “AvailableToSupply.”

## Understanding the End-to-End Integration Solution Design

These steps provide information about the design of the end-to-end solution:

1. Define the endpoint for Order Processing application (OP).

You must notify the system of the supply chain “restock” events so an event is configured on the application view to represent “Restock” events in OP. You must also allow for updating OP to indicate parts have been ordered so no new restock event happens until needed again. As a result, a service is defined on the application view to make this capability available.

2. Define the B2B conversations as the supply chain endpoint.

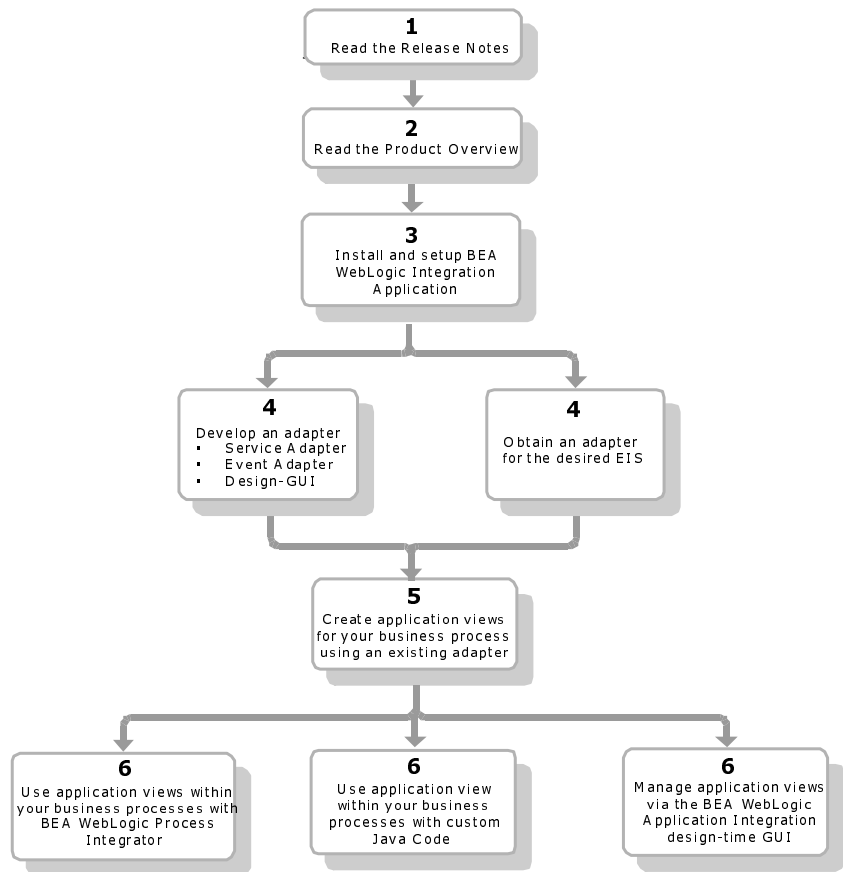
To interact with the supply chain, you must define a B2B conversation that allows interactions with suppliers. Using a conversation definition, you can ask for a supplier to give you quotes on availability and process. You can also use a conversation to buy the parts from a supplier.

3. Define the process to tie these endpoints together using BEA WebLogic Process Integrator.

## Roadmap for Using BEA WebLogic Application Integration

The steps outlined in Figure 3-3 provide you with a high-level guideline to all of the tasks and processes that you must perform to install, configure, and work with BEA WebLogic Application Integration. Think of these steps as a roadmap to guide you through the process and to point you to the resources available to help you.

**Figure 3-3 Roadmap for Using BEA WebLogic Application Integration**



1. Read the Release Notes.

Read the *BEA WebLogic Application Integration Release Notes* to ensure that you have up-to-date information about this product release. The Release Notes are provided in printed form in the product package and are also available from the BEA customer support web site.

2. Read the Product Overview.

Read the *BEA WebLogic Application Integration Product Overview* for information about how Application Integration works and how to use its components. This document is available on the BEA WebLogic Integration

documentation CD that is provided in the product box, and it is available at <http://edocs.bea.com>.

3. Install and set up BEA WebLogic Application Integration.

Read the *BEA WebLogic Application Integration Installation and Setup Guide* for information about preparing your system to install BEA WebLogic Application Integration, installing the application, and performing post-installation setup and setup configuration tasks. The Installation and Setup Guide is provided in printed form in the product package and are also available in PDF format at <http://edocs.bea.com>.

4. Develop an adapter or obtain an adapter.

You must have an adapter installed in the BEA WebLogic Application Integration environment before you can begin creating application views. You may either obtain an existing third-party adapter, or if an adapter that meets your needs has not been developed, you will need to develop it (or have it developed).

Read the *BEA WebLogic Application Integration Development Guide* for complete information on developing J2EE-compliant adapters. The Development Guide contains instructions for developing service adapters, event adapters, and a design-time GUI for the adapters as well as appendices for the sample DBMS adapter and the sample email adapter.

The Development Guide is available on the BEA WebLogic Integration documentation CD in PDF format and HTML format and at <http://edocs.bea.com>.

5. Create application views for your business process using an existing adapter.

Read the *BEA WebLogic Application Integration User Guide* to get an overview of the integration framework component of BEA WebLogic Application Integration and complete instructions for creating application views. The user guide is available on the BEA WebLogic Integration documentation CD in PDF format and HTML format and at <http://edocs.bea.com>.

6. Read the *BEA WebLogic Application Integration User Guide* for instructions on defining application views, using defined and deployed application views, services and events in workflows and within WebLogic Process Integrator.

Read the *BEA WebLogic Application Integration User Guide* for instructions on custom Java code to represent the business process into which you will use a deployed application view.

### 3 Using BEA WebLogic Application Integration

---

Read the *BEA WebLogic Application Integration User Guide* for an overview of the BEA WebLogic Application Integration design-time GUI and instructions for managing application views using the BEA WebLogic Application Integration design-time GUI.

The *BEA WebLogic Application Integration User Guide* is available on the BEA WebLogic Integration documentation CD in PDF format and in HTML format and at <http://edocs.bea.com>.



---

# Index

- A**
- Adapter 1-8
  - Adapter Development Kit 3-3, 3-6
  - Adapter Development Kit (ADK) 1-8, 2-1
    - Logging/Auditing Framework 1-9
    - Packaging Framework 1-9
    - runtime framework 1-9
  - Adapters 1-4, 3-3, 3-6
    - components 2-3
    - development 3-11
    - event 2-2, 2-5
    - event generator 2-5
    - event router 2-5
    - run-time component 2-4
    - service 2-2, 2-4
  - Application integration 1-2
  - application server 1-2, 2-1
  - Application view 1-5, 2-6, 3-4, 3-7
    - bi-directional communication support 1-8
      - creating 3-11
      - events and services 1-7
      - protocol 1-7
      - XMLtranslation 1-7
  - Application View Management Console 1-8
  - architecture 1-4
  - authentication 2-4
- B**
- B2B 3-6
    - collaboration 3-1
    - conversations 3-9
    - integration 1-2, 1-3
  - B2C 3-1
  - BEA WebLogic Integration 1-2
  - BizTalk 3-8
  - Business analyst v
- C**
- common protocol 1-7
  - Component
    - adapter design-time 2-3
    - adapter run-time 2-4
  - custom code 2-6
  - Customer Relationship Management 1-4, 3-2
  - customer support contact information vii
- D**
- Design-Time GUI 1-8
  - documentation, where to find it vii
- E**
- e-commerce transactions 3-6
  - edocs vi
  - Endpoint 1-1
  - endpoints 3-9
  - Enterprise application integration 1-2, 1-3
  - Enterprise Resource Planning 1-4, 3-2
  - Event adapter 2-2, 2-5

---

event generator 1-9, 2-5, 2-6

event object 3-4

event router 2-5, 2-6

Events 1-7

Extensible Markup Language 1-2

eXtensible Style Language Transformation  
(XSLT) 3-8

## G

GUI 1-8

Application View Management Console  
1-8

## H

HTTP 3-6

## I

Integration

e-Commerce 3-5

scenarios 3-2

within an enterprise 3-2

Integration Framework 1-4

application view 1-5

Design-Time GUI 1-8

## J

J2EE

Connector Architecture 1-4, 2-1

engine 2-4

J2EE Connector Architecture Specification

vi

Java 3-6

Java 2 Enterprise Edition 1-2

Java classes 2-3

Java Message Services (JMS) 1-4

Java Messaging Service (JMS) 2-5

Java Server Pages 2-3

## M

Metadata 1-7, 1-8, 2-3

## P

printing product documentation vi

process flow 1-1, 2-6

process management 1-2, 1-3

protocol 1-7

## R

Related Information

J2EE Connector Architecture

Specification vi

Sun Microsystems Java site vii

WebLogic Server documentation vi

XML Schema Specification vi

related information vi

Roadmap 3-10

RosettaNet 3-8

## S

Service adapter 2-2, 2-4

Services 1-7

Standards

HTTP 3-6

Java 3-6

XML 3-6

Sun Microsystems vii

Sun Microsystems, Inc. Java site vii

support

technical vii

## T

Technical analyst v

transaction management 2-4

---

## U

User interface 1-8

## W

WebLogic Collaborate 1-3, 3-6, 3-8

WebLogic Process Integrator 1-3, 3-3, 3-4, 3-6

WebLogic Server vi, 3-3, 3-6

WebLogic XML/Non-XML Translator 1-3

## X

XCCI interface 3-6

XML 1-7, 3-6

- data 3-4

- document 3-4

- format 3-4

- interface 2-1

- metadata 1-7

- request document 2-2

- schema 1-7, 3-5

XML Schema Specification vi

XML translation 1-3

XSLT 3-8