BEA WebLogic Integration™

Introducing Application Integration

Release 7.0
Document Date: June 2002

Copyright

Restricted Rights Legend

Trademarks or Service Marks

Introducing Application Integration

Part Number	Date	Software Version
N/A	June 2002	7.0

Contents

About This Document

1. Introduction to Application Integration

2. Understanding the Integration Framework

3. Understanding the ADK

4. Understanding the Development Kit Adapters

5. Understanding the Application Integration Plug-In for BPM

Index

About This Document

Introducing Application Integration is organized as follows:

- "Introduction to Application Integration" provides an overview of application integration. It provides detailed discussion of all major concepts, features, and component architecture for application integration.
- "Understanding the Integration Framework" provides information about using an adapter's design-time GUI to integrate applications.
- "Understanding the ADK" provides information about developing adapters using the Adapter Development Kit (ADK) provided with WebLogic Integration.
- "Understanding the Development Kit Adapters" provides information about the fully functional EIS adapters provided with the ADK.
- "Understanding the Application Integration Plug-In for BPM" provides information about the plug-in that makes application integration functionality available through the business process management (BPM) tools.

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 that includes an application server, and tools for application integration, business process modeling, and B2B integration. In this document, we assume that the technical analyst is knowledgeable about the entire system, from end to end.

- *Business Analyst*–Works with the technical analysts to ensure accuracy of the business interface functionality. Also provides requirements for application views and business processes that use those application views.
- *System Administrator*–Sets up, tunes, and maintains WebLogic Integration servers. Also installs and maintains adapters used with WebLogic Integration.

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 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 WebLogic 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://edocs.bea.com`)
- BEA WebLogic Integration Studio documentation (`http://edocs.bea.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 WebLogic 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 WebLogic Integration documentation.

In your e-mail message, please indicate which release of the WebLogic Integration documentation you are using.

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

Convention	Item
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:
	`chmod u+w *`
	`c:\startSever`
	`.doc`
	`wls.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 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 *Example*: `import com.bea.adapter.cci.* . . . ;`
. . .	Indicates the omission of items from a code example or from a syntax line. The vertical ellipsis itself should never be typed.

CHAPTER

1 Introduction to Application Integration

WebLogic Integration provides a standards-based integration solution for connecting applications both within and between enterprises. Specifically, this solution provides a means to integrate applications by defining communication endpoints, either in custom code or in a process flow defined with the WebLogic Integration Studio.

WebLogic Integration provides the following tools for integrating applications:

- Integration framework
- Adapter Development Kit (ADK)
- EIS adapters
- Application integration plug-in for the WebLogic Integration Studio

By using these tools, you can integrate all your enterprise information systems (EIS). Typical IT organizations use several highly specialized applications. Without a common integration platform, integration of such applications requires extensive, highly specialized development efforts.

WebLogic Integration makes use of adapters to establish a single enterprise-wide framework for integrating current or future applications. Adapters greatly simplify your integration efforts because they allow you to integrate each application with a single application server, and thus avoid the need to integrate every application with every other application.

WebLogic Integration helps you use these adapters to define business-focused interfaces to an EIS. These interfaces, called *application views*, provide a simple, consistent, self-describing interface to services and events in an application. Unlike adapter users, application view users are not required to have intimate knowledge of the EIS or its client interface. As a result, users who are not programmers, such as technical analysts, can use application views. These same users may be incapable of using an adapter directly, due to lack of familiarity with the EIS.

The following sections give a brief overview of each of the application integration tools provided by WebLogic Integration. For details, see *Developing Adapters* and *Using Application Integration.*

CHAPTER

2 Understanding the Integration Framework

The integration framework provided by WebLogic Integration offers a standards-based architecture for hosting *application views*: business-oriented interfaces to enterprise applications.

This section includes the following topics:

- Understanding Adapters
- What Are Application Views?
- What Are Design-Time GUIs?

Understanding Adapters

Application integration uses an integration framework that is based on adapters and application views to help you integrate applications in your enterprise. Instead of *hardwiring* your enterprise systems together, you can use the integration framework to build *adapters*: software components that connect enterprise systems to an application server. Once you deploy an adapter for an EIS, other components and applications can use that adapter to access data on the EIS.

What Are Application Views?

Application views provide a layer of abstraction between an adapter and the EIS functions exposed by that adapter. By using application views, you can simplify the procedure you must perform to access adapters. Instead of accessing an EIS by directly invoking it, you can simply edit the adapter's application views, create new application views, or delete obsolete ones. This layer of abstraction, formed by application views, makes it easy for nonprogrammers to maintain the services and events exposed by the adapter.

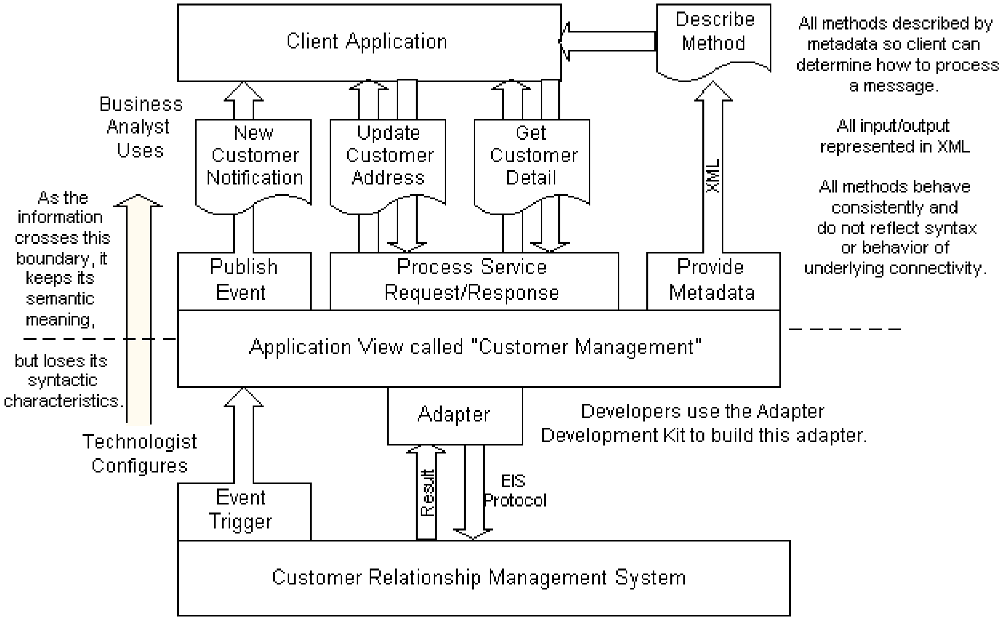
Each application view defines a set of business functions on one adapter's EIS. After an adapter is created, you can use its Web-based interface to define your own application views. Such application views allow you to display the application capabilities exposed by an adapter.

If you are a business analyst or technical analyst and you define an application view using an adapter, you can customize the application view for a specific business purpose. The business purpose is defined by the business analyst. For example, if you define a "Customer Management" application view on an adapter for a CRM (Customer Relationship Management) system, then you are likely to add only services and events that are related to customer management. You can, however, create application views that are as inclusive as necessary. Because application views can be customized for a specific business purpose, they work much better than the "one size fits all" approach used by many other EAI systems.

Understanding Application Views

The business-level view of an application's capabilities provides a logical dividing line between the focus of the programmer and that of the technical analyst. For example, with a business-level view, a technical analyst can create records in a database without knowing SQL. Figure 2-1 provides a diagram of an application view at work in an application integration environment.

Figure 2-1 How Application Views Work


Client Application
Describe Method
All methods described by metadata so client can determine how to process a message.
All input/output represented in XML
All methods behave consistently and do not reflect syntax or behavior of underlying connectivity.
Business Analyst Uses
New Customer Notification
Update Customer Address
Get Customer Detail
XML
As the information crosses this boundary, it keeps its semantic meaning,
but loses its syntactic characteristics.
Publish Event
Process Service Request/Response
Provide Metadata
Application View called "Customer Management"
Adapter
Developers use the Adapter Development Kit to build this adapter.
Technologist Configures
Event Trigger
Result
EIS Protocol
Customer Relationship Management System

Main Features of Application Views

Because WebLogic Integration uses the application view as its primary user interface for adapters, it offers several features not commonly found in competing EAI technologies:

- Use of XML as a Common Language Among Applications
- Use of Service and Event Definitions to Expose Application Capabilities
- Use of XML Schemas to Define the Data for Services and Events
- Support of Bidirectional Communication in Adapters

The remainder of this section provides descriptions of these functions.

Use of XML as a Common Language Among Applications

In an EAI scenario, it is much easier and more efficient to use one common data format to integrate every EIS with WebLogic Server than it is to use a variety of custom, proprietary data formats to integrate each EIS with every other EIS. When a common data format is used, all applications communicate using a standard language. WebLogic Integration uses XML, the increasingly popular data interchange format, as its common data format.

In the WebLogic Integration environment, virtually all messages are sent as XML documents:

- For each service, application views require an XML request message and provide an XML response message.
- When events are generated, registered event listeners receive event information as XML messages. The application view relies on its adapter to translate the EIS-specific format into and from XML.

Because an adapter translates an application's data format using XML, business analysts do not need to understand that format themselves. If you are a business analyst and you want to use an adapter, you need to know only how to define and use application views. Best of all, because all adapters use a similar Web-based interface for defining application views, it is easy to learn to use current and future adapters. Thus XML simplifies the use of EAI for developers and business analysts alike.

Use of Service and Event Definitions to Expose Application Capabilities

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 and subscribe model. *Services* are business functions that may be invoked by a user. Service invocations cause messages to be sent to an application following a request/response model. Events, service requests, and service responses are all passed through the system as XML documents.

Use of XML Schemas to Define the Data for Services and Events

Each application view uses an XML schema as *metadata*: that is, as information about the XML information for events, service requests, and service responses. This metadata helps users understand the data requirements of any application view event or service.

Support of Bidirectional Communication in Adapters

Currently, the J2EE Connector Architecture Specification version 1.0 does not provide guidelines governing how an EIS initiates communication with an application server or client. WebLogic Integration provides this communication capability via event adapters.

Who Uses Adapters and Application Views?

In your enterprise, several people may share the responsibilities of maintaining adapters, application views, and the services and events for those application views. In most enterprises, these people belong to one of three groups:

- System administrators
- Adapter developers
- Adapter users

System Administrators

If you are responsible for installing the WebLogic Integration Adapter Development Kit (ADK), then you are the *system administrator* mentioned in this document and in *Using Application Integration.*

Adapter Developers

If you are a software developer or a high-level technician in your enterprise, you are probably an *adapter developer*. Adapter developers commonly use the ADK to develop new adapters for EIS systems and design the accompanying Common Client Interface (CCI) for the adapter. The CCI is an interface that the adapter user interacts with to design application views representing functionality associated with a resource adapter. For more information, see *Developing Adapters.*

Adapter Users

If you are a business analyst, EIS specialist, or technical analyst in your enterprise, you are probably an *adapter user*. Adapter users do not usually create adapters, but they may provide specifications to adapter developers. Once an adapter is created, application views for it are often defined and managed by its users.

How Responsibilities Are Distributed Among Users

Table 2-1 shows how tasks and responsibilities are typically divided among users.

Table 2-1 Common Jobs and Their Owners

Task	System Administrator	Adapter Developer	Adapter User
Installing WebLogic Integration	X		
Developing an adapter		X	
Developing a user interface for defining application views		X	
Configuring WebLogic Integration to deploy an adapter	X	X	
Defining an application view			X
Configuring the connection parameters of application views			X
Adding and testing services and events			X
Setting up application view folders			X

What Are Design-Time GUIs?

The integration framework provided by WebLogic Integration provides a means for developers to create the Common Client Interface (CCI) for each adapter. The CCI enables applications components and Enterprise Application integration (EAI) frameworks to drive interactions across heterogeneous EISs using a common client API. An adapter's design-time GUI enables nonprogrammers to rapidly create, deploy, test, and edit application views, which they can customize by adding services and events.

Creating Application Views Using a Design GUI

The primary purpose of an adapter's design GUI is to allow you to define, deploy, and test application views. For detailed information about defining application views, see *Using Application Integration.*

Managing Application Views with the Console

The Application View Console helps you access, organize, and edit all application views in your enterprise. You can use the Application View Console to create new folders and then add new application views to them. These folders allow you to organize your application views according to your own navigation scheme, regardless of the adapter used by the application view.

For detailed information about managing application views, see "Using the Application View Console" in *Using Application Integration.*

When to Define Application Views and When to Write Custom Code

Using an adapter's design-time GUI is not the only way to expose the functionality of an EIS, but it is usually the most convenient method. To support service invocations and events, you can define application views, or you can write custom code that accomplishes equivalent functions. At a minimum, you must define application views for each adapter, to expose the functions provided by the adapter's application. However, if your users require a greater than average degree of control, you may also write custom code that allows them to access the resources of an adapter. You must decide whether the needs of your enterprise can best be met by defining application views, writing your own code, or implementing a combination of both methods.

When to Define Application Views

Most EIS applications can be integrated into your system easily by defining application views. You may want to define application views in the following situations:

- You have multiple EIS systems in your enterprise, and you lack developers with a detailed, thorough knowledge of them.
- You want to use the WebLogic Integration Studio to construct and manage business processes.
- You need to update the parameters of an adapter or one of its processes.

When to Write Custom Code Instead of Defining Application Views

In general, you should write a custom interface to an adapter only in the following situations:

- You have only one EIS system in your enterprise.
- Your developer has thorough, detailed knowledge of each EIS involved in the business processes being coded.
- You do not need to use the coded functions in the WebLogic Integration Studio.
- You do not anticipate any need to change your code.

3 Understanding the ADK

If you are an adapter provider or developer, you can use the WebLogic Integration Adapter Development Kit (ADK) to create your own J2EE-compliant EIS adapters. The ADK is a collection of four frameworks, each of which comprises tools and Java classes. Together, these frameworks let you quickly develop adapters that you can easily test, package, and distribute. For details about using the ADK frameworks to create adapters, see *Developing Adapters.*

This section provides information about the following four component frameworks:

- Design-Time Framework
- Run-Time Framework
- Logging and Auditing Framework
- Packaging Framework

Design-Time Framework

An adapter's design-time interface lets you define and deploy application views on the instance of WebLogic Server that is hosting WebLogic Integration. Whenever you build an adapter with the WebLogic Integration ADK, you can also develop a design-time user interface for it. The design-time user interface for an adapter can be accessed from any common Web browser. The interface allows users who are not programmers to interact with the adapter without writing code. For example, by using a user interface, business analysts can log in to an adapter and define their own custom application views. Without a design-time user interface, an adapter can be used only by highly technical users.

To facilitate the development of design-time user interfaces for adapters, the ADK includes a set of Java classes and tools known as the *design-time framework*. This framework is a powerful feature because it allows users who are not programmers to use an adapter. By simplifying the development process, the ADK's design-time framework expands the audience for the adapter and upgrades the role played by business analysts in the implementation of business processes.

Run-Time Framework

The ADK provides a *run-time framework*: a complete, extensible event generator that supports the development of event adapters. To help you develop service adapters, the run-time framework provides a J2EE-compliant adapter that offers a complete set of the minimum functions. Adapter developers can save coding and debugging time by starting with this base framework and extending it to meet the needs of their enterprise.

Logging and Auditing Framework

If you are an administrator, it is essential to create adapters that automatically log alert messages on your system that can be audited later. To make it easy to develop an adapter with built-in logging and auditing support, the ADK includes a logging and auditing framework. Any adapter you develop can generate internationalized and localized alert messages and can deliver these messages to multiple output destinations.

Packaging Framework

If you are a third-party adapter provider, you can use the ADK packaging framework when preparing your adapter for delivery to a customer. This framework makes it easy to create the archive and environment files required for packaging.

CHAPTER

4 Understanding the Development Kit Adapters

The BEA WebLogic Integration Adapter Development Kit (ADK) provides sample adapters to get you started developing your own adapters: a DBMS adapter and a sample adapter. If you are developing your own adapters using the ADK, we recommend that you begin by studying the adapters supplied with the kit. Although these adapters are generic and simple, they serve as excellent examples of the types of adapters you can build using the ADK.

How the Kit Adapters Were Developed

All the development kit adapters were developed using the ADK. Although the ADK makes it possible to develop sophisticated adapters, the adapters provided in the kit have been kept simple deliberately, to make them easy to dissect and understand.

How to Use the Kit Adapters

If you are an adapter provider or developer, we recommend that you study the kit adapters to increase your knowledge of the ADK and to determine how you can use the adapters as models for your own. All kit adapters are based on version 1.3 of the J2EE (Java 2 Platform, Enterprise Edition) from Sun Microsystems. For details about the ADK, see *Developing Adapters.* The following sections provide details about the kit adapters.

DBMS Adapter

WebLogic Integration provides a DBMS adapter that integrates WebLogic Server with a simple relational database that it uses as its EIS. The DBMS adapter serves as a good example for adapter developers and providers who want to understand the adapter and ADK without having to learn an unfamiliar proprietary database system.

If you do not have a suitable database to use with the DBMS adapter, you can use the PointBase database included with WebLogic Integration.

The DBMS adapter supports the following functions:

- Retrieving all database records
- Retrieving sets of database records
- Writing database records
- Receiving notifications when a record is updated, inserted, or deleted from a table

For more information, see "Learning to Develop Adapters Using the DBMS Sample Adapter" in *Developing Adapters.*

Sample Adapter

The sample adapter is provided as a template for new adapters. It includes a design-time component, a service adapter, and an event adapter. The source code clearly documents the structure of the service adapter and the event adapter.

The sample adapter is accompanied by a simple EIS implementation that demonstrates how related events can be triggered by invoking services. Each component contains comments indicating where you must supply your own adapter-specific logic.

CHAPTER

5 Understanding the Application Integration Plug-In for BPM

In addition to the tools required to integrate your applications, WebLogic Integration also provides a powerful set of tools that enables you to achieve process-level integration among your applications. Chief among these business process management (BPM) tools is the WebLogic Integration Studio, a graphical user interface that allows business analysts to graphically construct and maintain business process workflows that integrate applications. The Studio is described in detail in *Using the WebLogic Integration Studio.*

The WebLogic Integration Studio and other BPM tools cannot work with application view services and events until you deploy the application integration plug-in. Once you deploy the plug-in, however, business analysts and other nonprogrammers can use the consistent, familiar interface provided by the Studio to invoke services and receive notification of events from workflows. For more information, see "Configuring Workflow Resources" in *Using the WebLogic Integration Studio*.

Use the WebLogic Integration functions and tools in the following order:

1. Establish application-level integration among all the applications in your enterprise, using the tools described in this document.
2. Develop adapters and define application views, using the tools described in this document.
3. Open the WebLogic Integration Studio and create business process workflows.
4. Incorporate your application view services and events with your workflows.

5. Deploy your workflows.
6. If you need to update existing workflows, adapters, services, or events, you can do so in the Studio. By making changes in this way, you can avoid time-consuming compiles, which are not required when you use the Studio.

For details about using the WebLogic Integration Studio to build business process workflows that include application view services and events, see "Using Application Views in the Studio" in *Using Application Integration*.

Index