



BEA WebLogic Integration™

Introducing
Application Integration

Version 2.1
Document Date: October 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, Jolt, Tuxedo, and WebLogic are registered trademarks of BEA Systems, Inc. BEA Builder, BEA Campaign Manager for WebLogic, BEA eLink, BEA Manager, BEA WebLogic Commerce Server, BEA WebLogic E-Business Platform, BEA WebLogic Enterprise, BEA WebLogic Express, BEA WebLogic Integration, BEA WebLogic Personalization Server, BEA WebLogic Portal, BEA WebLogic Server and How Business Becomes E-Business are trademarks of BEA Systems, Inc.

All other trademarks are the property of their respective companies.

Introducing Application Integration

Part Number	Date	Software Version
N/A	October 2001	2.1

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 Application Integration

Application Integration.....	1-1
------------------------------	-----

2. Understanding the Integration Framework

Understanding Adapters	2-1
Application Views: Exposing Application Functions	2-2
Understanding Application Views	2-2
Main Features of Application Views	2-3
Use of XML as a Common Language Among Applications	2-4
Use of Service and Event Definitions to Expose Application Capabilities	2-4
Use of XML Schemas to Define the Data for Services and Events....	2-5
Support of Bidirectional Communication in Adapters	2-5
Who Uses Adapters and Application Views?	2-5
System Administrators.....	2-5
Adapter Developers.....	2-5
Adapter Users.....	2-6
How Responsibilities are Distributed Among Users	2-6
Design-time GUIs: Web-based Interfaces for Creating Application Views	2-7

Creating Application Views Using the Design-time GUI.....	2-7
Managing Application Views with the WebLogic Integration Application View Console	2-7
When to Define Application Views and When to Write Custom Code	2-8
When to Define Application Views	2-8
When to Write Custom Code Instead of Defining Application Views.....	2-8
3. Understanding the ADK	
Components Included in the ADK	3-1
Design-Time Framework.....	3-2
Run-Time Framework	3-2
Logging and Auditing Framework	3-3
Packaging Framework	3-3
4. Understanding the Included Adapters	
How the Included Adapters Were Developed	4-1
How to Use the Included Adapters.....	4-2
E-mail Adapter	4-2
DBMS Adapter	4-2
Sample Adapter	4-3
5. Understanding the Application Integration Plug-in for BPM	
Understanding the Application Integration Plug-in for BPM	5-2

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 included Adapter Development Kit (ADK).
- “Understanding the Included Adapters” provides information on the two fully-functional EIS adapters included with the ADK.
- “Understanding the Application Integration Plug-in for BPM” provides information on the plug-in for business process management (BPM).

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, provides requirements for application views, and provides requirements for business processes that use those application views.
 - **System Administrator**—sets up, tunes, and maintains WebLogic Integration servers, installs and maintains adapters installed for use in 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 that you are using the documentation for the WebLogic Integration 2.1 release.

If you have any questions about this version of WebLogic Integration, or if you have problems installing and running 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 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.
<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
monospace boldface text	Identifies significant words in code. <i>Example:</i> void commit ()
monospace italic text	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">■ 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 <i>Example:</i> <pre>import com.bea.adapter.cci.* . . . ;</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 Application Integration

The application integration component of WebLogic Integration provides a standards-based integration solution for connecting applications both within and between enterprises. Application integration provides a means to form a complete integration solution by defining communication endpoints. The communication endpoints may be defined in a process flow using WebLogic Integration Studio or implemented using custom code.

This section provides information on the following subject:

- Application Integration

Application Integration

Application integration is one of the primary components of BEA WebLogic Integration. In turn, application integration is comprised of the following components:

- The Integration Framework
- The Adapter Development Kit (ADK)
- Included EIS adapters
- The application integration plug-in for WebLogic Integration Studio

1 Introduction to Application Integration

You can use application integration to achieve integration among all of your enterprise information systems (EIS). Typical IT organizations are comprised of several highly-specialized applications. Without a common integration platform to facilitate application-level integration, integration of these applications requires extensive, highly specialized development efforts.

Application integration makes use of adapters to establish an enterprise-wide, united framework for integrating any current or future application. Adapters greatly simplify your integration efforts because they allow you to integrate each application to the application server instead of integrating each application to every other application.

Application integration allows you to use these adapters to define business-focused interfaces to an EIS called application views. Application views provide a simple, self-describing, consistent interface to services and events in the application. Application views make use of the adapter for an EIS. Unlike adapters, an application view does not require its users to have intimate knowledge of the EIS or its client interface. This allows non-programmers, such as technical analysts, to use application views. These same users may be incapable of using the adapter directly due to lack of familiarity with the EIS.

The following sections give a brief overview of each of the components of application integration. For details on the components and implementation, see [Developing Adapters](#) and [Using Application Integration](#).

2 Understanding the Integration Framework

The integration framework provided by application integration provides a systematic, standards-based architecture for hosting application views, which are business-oriented interfaces to enterprise applications.

The features provided in Integration Framework include:

- Application Views: Exposing Application Functions
- Design-time GUIs: Web-based Interfaces for Creating Application Views

Understanding Adapters

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

Application Views: Exposing Application Functions

Using application views can help your enterprise by simplifying the way you access adapters. Application views provide a layer of abstraction between an adapter and the EIS functions exposed by that adapter. Instead of accessing an EIS by directly programming 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 non-programmers to maintain the services and events exposed by the adapter.

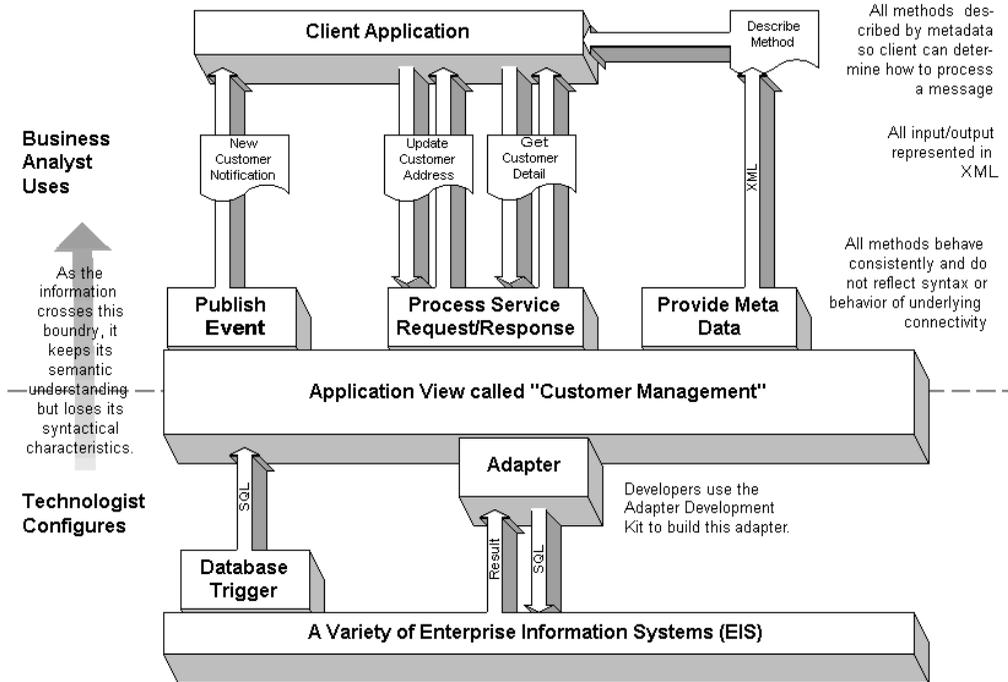
Each application view is specific to one adapter and defines a set of business functions on the adapter's EIS. After an adapter is created, you can use the adapter's Web-based interface to define your own application views. Such application views provide a *view* of the application capabilities exposed 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 are defining a "Customer Management" application view on an adapter for a CRM (Customer Relationship Management) system, then you would normally add only the services and events related to customer management. You can create application views that are as inclusive as you need them to be. Because you can customize application views for a specific business purpose, application views 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 separation between the programmer and the technical analyst. For example, this abstraction enables a technical analyst to create records in a database without having to know SQL. Figure 2-1 provides a diagram of the application view at work within the application integration environment.

Figure 2-1 How Application Views Work



Main Features of Application Views

Because WebLogic Integration uses application views as its primary user interface for adapters, WebLogic Integration exhibits a number of features not commonly found in competing EAI technologies. Application views provide the following features:

- 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

Use of XML as a Common Language Among Applications

In an EAI scenario, it is much easier to use one common data format to integrate each EIS with WebLogic Server than it is to use a variety of custom, proprietary data formats to integrate each EIS with every other EIS. A common data format makes application integration efficient because each application needs to be integrated only with the common data format rather than with every other application. With a common data format, all applications communicate using a standard language. WebLogic Integration uses XML, the widely-embraced 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 XML request messages and provide an XML response message.
- When events are generated, registered event listeners receive event information as XML. The application view relies on its adapter to translate the EIS-specific format to and from XML.

Because an adapter uses XML to translate the application's data format, business analysts do not need to understand the application's data format. If you are a business analyst and you want to use an adapter, you only need to know 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. XML makes EAI smooth both for developers and business analysts.

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/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.

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

Each application view uses an XML schema as metadata to describe the XML data 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

The J2EE specification does not currently provide guidelines for an EIS to initiate communication with the application server or client. Application integration provides this 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 their services and events. In most enterprises, these people can be grouped into one of three categories:

- System administrators
- Adapter developers
- Adapter users

System Administrators

If you are the one responsible for installing the WebLogic Integration Adapter Development Kit (ADK), then you are the system administrator referred to throughout this document and [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 user interface that the adapter user will interact with when creating application views for the adapter. For information on developing adapters, 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 develop adapters but may provide specifications to the adapter developer. Once an adapter is developed, adapter users normally define and manage its application views.

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 the user interface used for defining application views		X	
Configuring WebLogic Integration to deploy an adapter	X	X	
Defining an application view			X
Configuring connection parameters of application views			X
Adding and testing services and events			X
Setting up application view folders			X

Design-time GUIs: Web-based Interfaces for Creating Application Views

The integration framework provided by application integration provides a means for developers to provide a design-time GUI for each adapter. An adapter's design-time GUI lets non-programmers rapidly create, deploy, test, and edit application views, which they can customize by adding services and events.

Creating Application Views Using the Design-time GUI

The primary purpose of an adapter's design-time 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 WebLogic Integration Application View Console

The Application View 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 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 used by the application view.

For detailed information about managing application views via the Application View Console, see “[Using the WebLogic Integration 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. To support service invocation and events, you can define application views, or you can write custom code that accomplishes equivalent functions. Normally, for each adapter, you will define application views to expose the application functions. However, for those who require more control, you may also write custom code to access the resources of an adapter. For your enterprise, you must decide whether to define application views, write your own code, or implement a combination of both methods.

When to Define Application Views

You can define application views to easily integrate most EIS applications. In general, define application views in the following situations:

- You have more than one EIS system in your enterprise, and you lack developers who have detailed, thorough knowledge of all of the systems.
- You want to use WebLogic Integration Studio to construct and manage business processes.
- You may need to update the parameters of the adapter or one of its processes.

When to Write Custom Code Instead of Defining Application Views

In general, write custom code as an interface to an adapter only in the following situations:

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

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.

This section provides information on the following topics:

- Components Included in the ADK
 - Design-Time Framework
 - Run-Time Framework
 - Logging and Auditing Framework
 - Packaging Framework

Components Included in the ADK

The ADK consists of a collection of tools and Java class frameworks. Together, these frameworks let you quickly develop adapters that you can easily test, package, and distribute. There are four frameworks of tools and Java classes:

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

For details on using the ADK frameworks to develop adapters, see the document [Developing Adapters](#).

Design-Time Framework

An adapter's design-time interface lets you define and deploy application views on a WebLogic Integration server. In WebLogic Integration, when you build any adapter with the ADK, you can develop a design-time user interface for it. An adapter's design-time user interface is accessed from any common Web browser and lets non-programmers interact with the adapter without writing code. For example, if an adapter has a user interface, business analysts can log in to the adapter and define their own custom application views. Without a design-time user interface, an adapter is usable 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 design-time framework provides a tool set to support the development of a user interface for an adapter. A design-time interface is a powerful feature because it makes it easy for non-programmers to use an adapter. This ease-of-use gives the adapter a wider audience and upgrades the role of the business analysts when implementing business processes.

Run-Time Framework

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

Logging and Auditing Framework

If you are an administrator, it is essential for your system's adapters to automatically log alert messages 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 included with application integration. The packaging framework provides Java classes and tools for packaging an adapter for delivery to a customer. This framework makes it easy to create the necessary archive and environment files.

3 *Understanding the ADK*

4 Understanding the Included Adapters

If you plan on using the BEA WebLogic Integration Adapter Development Kit (ADK) to develop adapters, WebLogic Integration includes three adapters to get you started: the e-mail adapter, the DBMS adapter, and the sample adapter. If you are developing your own adapters using the ADK, you should study these adapters to aid your development efforts. Although these included adapters are generic and simple, they serve as excellent examples of the types of adapters you can build using the ADK.

How the Included Adapters Were Developed

All of the included adapters were developed using the ADK. Using the ADK, it is possible to develop adapters that are far more sophisticated than the included adapters. These adapters were kept simple because their purpose is to be dissected and understood by adapter providers and developers.

How to Use the Included Adapters

If you are an adapter provider or developer, you should study the included adapters to increase your ADK knowledge and to provide a model for your own adapter development projects. Each adapter is summarized below. All are based on version 1.3 of the J2EE (Java 2 Platform, Enterprise Edition) from Sun Microsystems. For details on the ADK, see [Developing Adapters](#).

E-mail Adapter

Application integration includes an e-mail adapter that integrates WebLogic Server with a simple e-mail server. For this e-mail adapter, a simple Java mail server is used as the EIS. If you are an adapter developer or adapter provider, this lets you focus on the details of the adapter and the ADK instead of an unfamiliar e-mail EIS.

Features of the e-mail adapter:

- Sending mail
- Receiving mail

For more information on the e-mail adapter, see “[The E-mail Adapter](#)” in *Developing Adapters*.

DBMS Adapter

Application integration includes a DBMS adapter that integrates WebLogic Server with a simple relational database. The DBMS adapter uses a simple relational database as its EIS. If you have no existing, suitable database to use with the DBMS adapter, you can use the Cloudscape database included with WebLogic Integration. However, the Cloudscape database is intended to be used only for running sample applications. BEA, Inc. does not support the use of Cloudscape in a production environment. 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.

Features of the DBMS adapter:

- Retrieving all database records
- Retrieving sets of database records
- Writing database records

For more information on the DBMS adapter, see “[The DBMS Adapter](#)” in *Developing Adapters*.

Sample Adapter

In addition to the e-mail adapter and DBMS adapter, a sample adapter is included. The sample adapter acts as a template that you can use as the starting point for developing a new adapter. The sample adapter includes a design-time component, service adapter, and event adapter. The source code clearly documents the structure of the service adapter and event adapter. The sample adapter also is accompanied by a simple EIS implementation to demonstrate how invoking services can trigger related events. Each of these components contain comments indicating where to supply your own adapter-specific logic.

4 *Understanding the Included Adapters*

5 Understanding the Application Integration Plug-in for BPM

WebLogic Integration comes with a powerful process engine known as business process management (BPM), formerly known as WebLogic Process Integrator. After you use application integration to achieve application-level integration in your enterprise, use WebLogic Integration Studio, a BPM component, to achieve process-level integration among your applications. If you are a business analyst, BPM lets you graphically construct and maintain business process workflows that integrate your applications. For details on BPM, see [Using the WebLogic Integration Studio](#).

BPM does not automatically work with application view services and events created with application integration. To make these two components work together, application integration includes a plug-in component for WebLogic Integration Studio. The application integration plug-in gives BPM access to application view services and events. After you install the application integration plug-in for BPM, business analysts and other non-programmers can use its consistent, familiar interface to interact with all services and events.

This section provides information on the following topics:

- Understanding the Application Integration Plug-in for BPM

Understanding the Application Integration Plug-in for BPM

The application integration plug-in is an add-on component for WebLogic Integration Studio that enables WebLogic Integration Studio to automatically recognize and use application views. If you are a business analyst, this combination of software lets you set up business process workflows that incorporate the application view services and events.

WebLogic Integration Studio is used for establishing process-level integration among applications and systems. In most cases, you will use WebLogic Integration Studio after using application integration to establish application-level integration for each application in your enterprise. Once you develop adapters and define application view services and events, you can use them in business process workflows. Once your business process workflows are deployed, WebLogic Integration Studio lets you easily update existing workflows, adapters, services, and events without requiring time-consuming compile cycles.

For details on using WebLogic Integration Studio to build business process workflows that include application view services and events, see [“Using Application Views in Business Process Management”](#) in *Using Application Integration*.

Index

- A**
- Adapter Development Kit (ADK)
 - runtime framework 3-2
 - Application view
 - events and services 2-4
 - protocol 2-4
 - Application View Management Console 2-7
 - application views
 - when to define 2-8
 - architecture 2-1
- B**
- Business analyst vi
- C**
- common protocol 2-4
 - custom code
 - for defining application views 2-8
 - customer support contact information vii
- D**
- documentation, where to find it vi
- E**
- edocs vii
 - Endpoint 1-1
 - event generator 3-2
 - Events 2-4
- G**
- GUI
 - Application View Management Console 2-7
- I**
- Integration Framework 2-1
 - Design-Time GUI 2-7
- J**
- J2EE Connector Architecture Specification vii
- P**
- printing product documentation vi
 - process flow 1-1
 - protocol 2-4
- R**
- Related Information
 - J2EE Connector Architecture Specification vii
 - Sun Microsystems Java site vii
 - WebLogic Server documentation vii
 - XML Schema Specification vii
 - related information vi

S

Services 2-4

Sun Microsystems vii

Sun Microsystems, Inc. Java site vii

support

technical vii

System administrator vi

T

Technical analyst v

U

User interface 2-7

W

WebLogic Server vii

X

XML

schema 2-5

XML Schema Specification vii