



BEA WebLogic Integration™

Introducing
BEA WebLogic 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 BEA WebLogic Integration

Part Number	Date	Software Version
880-001001-002	October 2001	2.1

Contents

About This Document

Who Should Read This Document viii
Contact Us! ix

1. The Road to e-Business Integration

The Integration Challenge 1-2
Integration Solutions 1-3
BEA WebLogic Integration..... 1-3
 Business Process Management..... 1-4
 Application Integration 1-5
 B2B Integration 1-6
 Data Integration 1-7

2. Business Process Management

Developing Business Processes..... 2-2
 Design..... 2-2
 Execution..... 2-5
 Monitoring and Optimization..... 2-6
Run-Time Interaction 2-6
APIs for Developing BPM Client Applications 2-7
Plug-In Framework..... 2-8
Sample Business Processes 2-9
 Sales Order 2-9
 Plug-In 2-9

3. Application Integration

Application Integration Framework	3-2
Hosting J2EE CA-Based Adapters	3-2
Application Views	3-3
Integration with Business Processes	3-7
Adapter Development Kit	3-7
Sample DBMS Adapter	3-8

4. B2B Integration

B2B Integration Framework	4-2
Conversation Definition and Monitoring	4-2
Trading Partner Configuration and Management	4-4
Business Protocol Support	4-7
Collaboration Agreement Definition and Management	4-7
Security Services	4-8
Lightweight Client Support	4-8
Integration with Business Processes	4-8
APIs and Logic Plug-Ins for B2B Application Development	4-9
Samples	4-10
Hello Partner	4-10
Channel Master	4-11
RosettaNet 2.0 Security	4-11
Lightweight Client	4-12
Messaging API	4-12
EDI Integration	4-13
Sample EDI Application	4-14

5. Data Integration

Data Translation	5-2
Integration with Business Processes	5-3
Samples	5-4
Data Transformation	5-6
Integration with Business Processes	5-7

A. Documentation Roadmap

WebLogic Integration.....	A-2
Business Process Management.....	A-3
Application Integration.....	A-4
B2B Integration	A-4
Data Integration	A-6



About This Document

This document provides an overview of WebLogic Integration. It first outlines the integration challenges e-businesses face with the wide variety of systems they use to power their enterprises. It then describes the application integration, B2B integration, business process management, and data integration functionality WebLogic Integration provides to integrate those business systems.

This document is the first in a series of four documents that provide an overview of WebLogic Integration, and that explain how the functionality provided by WebLogic Integration is used at various stages in the design, development, and deployment of integrated solutions. Readers should start with this document to gain a comprehensive understanding of the functionality provided by WebLogic Integration, and then proceed to the other three documents in the series:

- *Learning to Use BEA WebLogic Integration*—Describes a sample integrated application. The sample application is a supply-chain hub that connects with business partners, automates a number of business processes, and integrates back-end enterprise information systems. Readers learn how to set up and run the sample application, and understand how an integrated solution is architected and developed using WebLogic Integration.
- *Designing BEA WebLogic Integration Solutions*—Describes how to design and architect WebLogic Integration solutions. Readers learn about good design principles following recommended best practices.
- *Deploying BEA WebLogic Integration Solutions*—Explains how to move an integrated solution from a development to a production environment. Readers learn about configuring, scaling, porting, and performance tuning integrated applications.

Once you are familiar with the contents of the overview documents, you can proceed to the detailed documentation about the functionality provided by WebLogic Integration.

This document is organized as follows:

- Chapter 1, “The Road to e-Business Integration,” describes typical integration problems e-businesses face, the roadblocks to successful integration, and the functionality WebLogic Integration provides to solve those integration problems.
- Chapter 2, “Business Process Management,” describes the functionality that enables the development and monitoring of complex, enterprise-wide e-business processes that span systems, applications, and people.
- Chapter 3, “Application Integration,” describes the Integration Framework, the Adapter Development Kit, and the functionality that enables the integration of enterprise systems, such as ERP, CRM, and legacy mainframes.
- Chapter 4, “B2B Integration,” describes the features that facilitate collaborative commerce activities through the creation, routing, and management of XML-based messages among trading partners.
- Chapter 5, “Data Integration,” describes the capabilities that enable the translation of data types, such as binary to XML, and the transformation of XML documents using the data mapping features of Contivo Analyst.
- Appendix A, “Documentation Roadmap,” provides task-oriented directions to the detailed documentation that describes the functionality provided by WebLogic Integration.

Who Should Read This Document

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

- **Integration Specialists**—Lead the integration design effort. Integration specialists have expertise in defining the business and technical requirements of integration projects, and in designing integration solutions that implement specific features of WebLogic Integration. The skills of integration specialists include business and technical analysis, architecture design, project management, and WebLogic Integration product knowledge.
- **Business Analysts**—Provide expertise in an organization’s business processes, procedures, policies, business rules, and resources. The skills of business

analysts include business analysis, process design and modeling, and operations knowledge.

- **Technical Analysts**—Provide expertise in an organization’s information technology infrastructure, including telecommunications, operating systems, applications, data repositories, future technologies, and IT organizations. The skills of technical analysts include technical analysis, application design, and information systems knowledge.
- **Enterprise Information System (EIS) Specialists**—Provide domain expertise in the systems that are being integrated using WebLogic Integration adapters. The skills of EIS specialists include technical analysis and application integration design.
- **System Administrators**—Provide in-depth technical and operational knowledge about databases and applications deployed in an organization. The skills of system administrators include capacity and load analysis, performance analysis and tuning, deployment topologies, and support planning.

Contact Us!

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

In your e-mail message, please indicate that you are using the documentation for the BEA WebLogic Integration 2.1 release.

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.bea.com**. You can also contact Customer Support by using the contact information provided on the Customer Support Card, which is included in the product package.

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

- Your name, e-mail address, phone number, and fax number
- Your company name and company address

-
- Your machine type and authorization codes
 - The name and version of the product you are using
 - A description of the problem and the content of pertinent error messages

1 The Road to e-Business Integration

Businesses operate today in a diverse environment. They interact with a wide variety of clients, both inside and outside of the enterprise, and they rely on disparate systems and processes to power their business activities. In this kind of environment, businesses face an integration challenge. To fully maximize their resources, businesses strive to bring together their internal systems and processes to gain operational efficiency, and they strive to extend those systems and processes to their trading partners to increase revenue.

WebLogic Integration is a single, unified platform that provides the functionality businesses can use to develop new applications, integrate them with existing systems, streamline business processes, and connect with trading partners. Using WebLogic Integration, businesses can integrate enterprise adapters for application integration, collaboration arrangements between trading partners, and business processes, resulting in composite applications that provide end-to-end business integration.

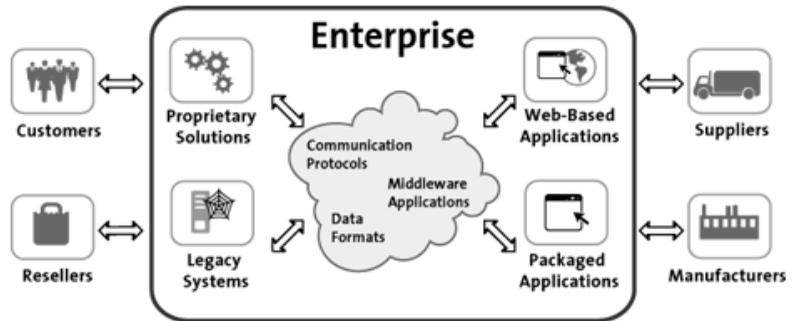
To begin our introduction to WebLogic Integration, we look at business systems today, and the functionality WebLogic Integration provides to integrate those systems. These topics are discussed in the following sections:

- The Integration Challenge
- Integration Solutions
- BEA WebLogic Integration

The Integration Challenge

Modern business systems are a complex mix of Web-based, wireless, custom-built, commercial, and legacy applications. Many of these systems—especially legacy applications—are deeply entrenched, and represent a substantial investment in development, deployment, training, and maintenance.

Figure 1-1 Business Systems Today



As Figure 1-1 suggests, business systems that drive the enterprise have some or all of the following characteristics:

- Use a variety of communication protocols: FTP, TCP/IP, HTTP, IIOP, and SOAP, among others
- Include middleware applications for distributed computing, based on a variety of standards and architectures: CORBA, RMI, and DCOM, among others
- Support a variety of data formats: binary, relational, proprietary, and XML, among others
- Include different implementation technologies: C, Cobol, and JCL, among others
- Run on a variety of different platforms: Microsoft Windows, variants of UNIX, VAX/VMS, and S/390, among others

To take full advantage of this variety of systems and processes, businesses look to integration solutions that can integrate their internal systems with those of their trading partners to manage complex business relationships.

Integration Solutions

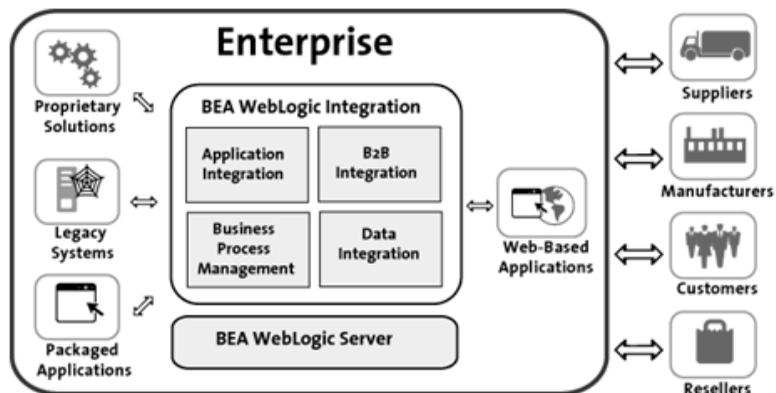
Integration solutions address two kinds of integration: *intra-enterprise* and *inter-enterprise*. Intra-enterprise integration involves linking together the types of applications shown in Figure 1-1, and the internal business processes that interact with them. It also includes mapping data between the various systems.

Inter-enterprise integration, on the other hand, extends a corporation's internal data and processes by linking them with customers and other corporate enterprises. This type of integration enables heterogeneous enterprises to interact in diverse business transactions that can be complex and long-running.

BEA WebLogic Integration

WebLogic Integration is a single, unified platform that provides the functionality to integrate business systems within an enterprise, and to link those systems in a collaborative arrangement with an organization's trading partners.

Figure 1-2 BEA WebLogic Integration



Underlying this functionality is the industry-leading J2EE application server, BEA WebLogic Server, which provides the critical infrastructure needed to develop integrated solutions, including transaction management, security, fault tolerance, persistence, and clustering.

WebLogic Integration enables intra- and inter-enterprise integration by providing functionality in the following areas:

- Business process management
- Application integration
- B2B integration
- Data integration

The following sections outline the key features in each of these areas of functionality.

Business Process Management

The business process management functionality of WebLogic Integration enables integration specialists to develop, run, and maintain complex e-business processes that integrate existing enterprise systems, cross-enterprise applications, and human decision makers. The following table lists the key features of business process management.

Table 1-1 Business Process Management Features

Key Feature	Description
Design, monitoring, and run-time interaction with business processes	WebLogic Integration Studio provides the functionality integration specialists use to design and monitor business processes, and to interact with running processes.
Run-time interaction with business processes	WebLogic Integration Worklist enables end users to interact with running business processes.

Table 1-1 Business Process Management Features (Continued)

Key Feature	Description
APIs for BPM client application development	Used to develop a variety of custom client applications: <ul style="list-style-type: none"> ■ Configuration clients ■ Design clients ■ Run-time management clients ■ Monitoring clients
Plug-in framework	Software that enables: <ul style="list-style-type: none"> ■ Development of plug-ins that extend the default design or run-time behavior of workflow components ■ Integration of other areas of functionality in WebLogic Integration with business processes
Sample business processes	Examples that show how to use the Studio to design business processes, and how to develop plug-ins

Application Integration

The application integration functionality of WebLogic Integration greatly simplifies the integration of existing internal enterprise systems with each other and with new e-business applications. The following table lists the key features of application integration.

Table 1-2 Application Integration Features

Key Feature	Description
Application integration framework	<ul style="list-style-type: none"> ■ Standards-based architecture for hosting J2EE Connector Architecture (J2EE CA)-based adapters ■ Application View Console for the development and management of application views

Table 1-2 Application Integration Features (Continued)

Key Feature	Description
Integration with business processes	Plug-in framework enables the integration of application views with business processes
Adapter Developer Kit (ADK)	Collection of frameworks that developers can use to develop J2EE CA-compliant adapters
Sample adapters	J2EE CA-compliant sample adapters that developers can use to learn how to develop adapters using the ADK

B2B Integration

The B2B integration functionality of WebLogic Integration enables the integration of trading partners over the Internet, and the integration of EDI environments with WebLogic Integration. The following table lists the key features of B2B integration features.

Table 1-3 B2B Integration Features

Key Feature	Description
B2B integration framework	<ul style="list-style-type: none">■ Services for messaging, connectivity, and business protocols that support collaboration between trading partners■ B2B Console for the configuration and management of conversations, trading partners, collaboration agreements, business protocols, and more
Integration with business processes	Plug-in framework enables the integration of trading partner conversations with business processes
APIs and logic plug-ins for B2B application development	<ul style="list-style-type: none">■ APIs to develop custom management and messaging applications, and to enable message logging■ Logic plug-ins to enable specialized processing of business messages

Table 1-3 B2B Integration Features (Continued)

Key Feature	Description
Sample applications	Examples that developers can use to model B2B integration solutions
Business protocol support	Support for RosettaNet, cXML, and XOCP
EDI integration	Functionality that includes Power.Enterprise!, an EDI-capable system, to enable the integration of WebLogic Integration with EDI environments

Data Integration

The data integration functionality of WebLogic Integration enables the translation of binary data to XML, and the transformation of XML grammars. The following table lists the key features of data integration.

Table 1-4 Data Integration Features

Key Feature	Description
Data translation	<ul style="list-style-type: none"> ■ Enables the translation of binary data to XML, and vice versa ■ Format Builder tool for the development of binary-to-XML maps
Data translation samples	Examples that illustrate the use of workflow actions for data translation
Data transformation	<ul style="list-style-type: none"> ■ Enables the transformation of one XML grammar to another ■ Contivo Analyst for the development of XSL stylesheets that map XML grammars

Table 1-4 Data Integration Features (Continued)

Key Feature	Description
Integration with business processes	<ul style="list-style-type: none">■ Plug-in framework enables the integration of data translation functionality with business processes■ XSL Transform workflow action enables the integration of data transformation functionality with business processes

2 Business Process Management

Business processes are fundamental to the success of a business. They consolidate company-wide tasks, actions, decisions, and events into a continuous flow that keeps business activity moving forward. Business processes can be internal to a business, coordinating simple interactions between different departments, or they can be public processes that orchestrate the collaboration between trading partners. Business processes can be as brief as a single transaction, or they can be long-running, spanning days or weeks.

The key to integrating applications, systems, and people into a single, integrated solution is the ability to manage business processes. WebLogic Integration provides comprehensive functionality that enables the integration of diverse applications into a single system within an enterprise, and the coordinated exchange of information between trading partners outside of the enterprise.

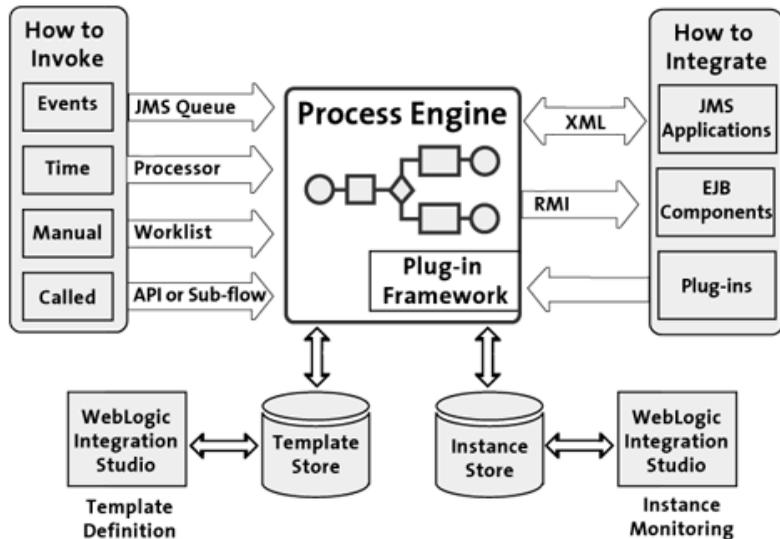
The following sections describe the business process management (BPM) functionality provided by WebLogic Integration:

- Developing Business Processes
- Run-Time Interaction
- APIs for Developing BPM Client Applications
- Plug-In Framework
- Sample Business Processes

Developing Business Processes

WebLogic Integration provides the functionality needed to design, execute, and monitor complex, enterprise-wide processes that span applications, systems, and people. WebLogic Integration includes a Java-based process engine that manages the run-time execution of business processes throughout the enterprise.

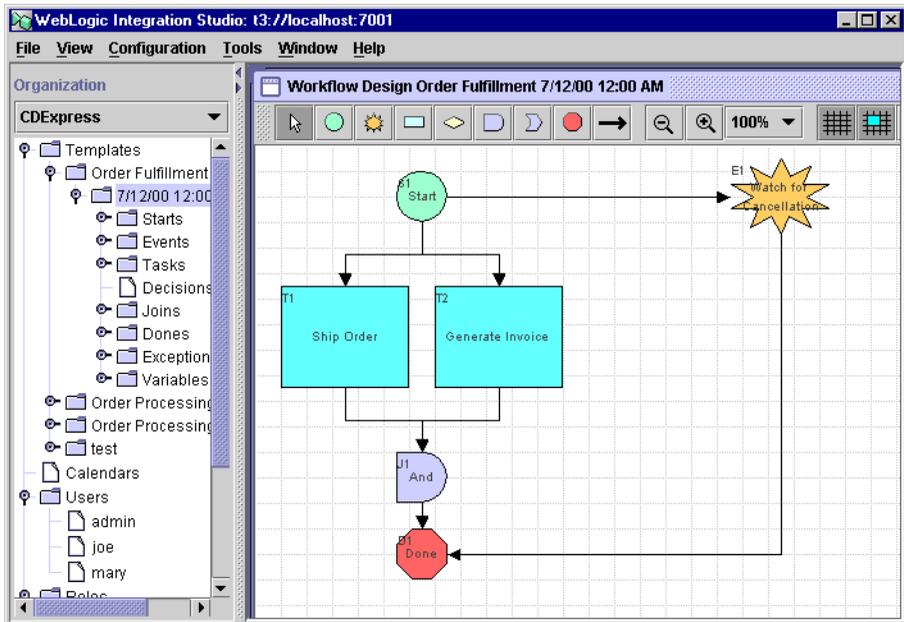
Figure 2-1 Business Process Management Functionality



Design

To design processes, integration specialists use the WebLogic Integration Studio. The Studio provides a graphical interface in which familiar flowchart elements are available for *workflow* modeling. A workflow is a graphical representation of a business process.

Figure 2-2 WebLogic Integration Studio



To model workflows, integration specialists first identify the required elements, and then, using the Studio, they define the process flow, arranging individual elements into the proper sequence to produce a workflow template definition. Template definitions created using the Studio are saved in the template store, which is implemented through a JDBC database, such as Oracle or SQL Server.

The template definition contains geometrical shapes called *nodes* that graphically depict the flow of a business process with all of the necessary activities linked in the proper sequence. The Studio provides the following nodes.

Table 2-1 Workflow Nodes

Shape	Name	Represents
	Start	Beginning of a workflow
	Task	Unit of work or discrete activity in the workflow

Table 2-1 Workflow Nodes

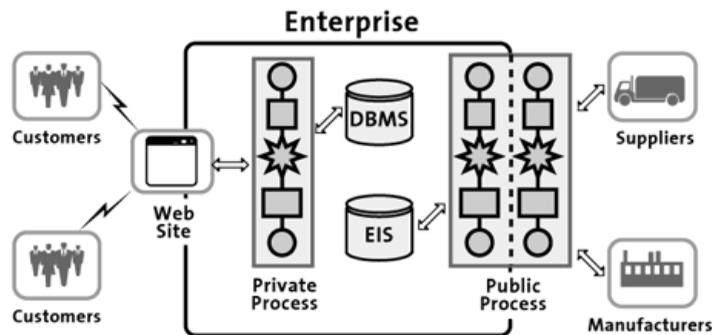
Shape	Name	Represents
	Decision	Condition in the workflow that must be evaluated
	Event	Wait state that can be activated by the receipt of an XML message
	And Join	Merge of two separate paths of a workflow. Both paths must complete executing before the flow continues.
	Or Join	Merge of two separate paths of a workflow. One of the paths must complete executing before the flow continues.
	Done	End of the workflow

Nodes can include *actions*, which are units of work that are performed when a node is activated in the workflow. The Studio provides a wide variety of actions. Some actions, such as *Start Workflow*, can be used to manage an entire workflow. Other actions, as shown in Figure 2-1, can integrate a workflow with external software, such as when an action calls an EJB method or sends an XML message to another application.

Public and Private Business Processes

Using the Studio, integration specialists can develop both *public* and *private* business processes. In the B2B integration environment, public processes are also known as *collaborative* processes. To develop public processes using the Studio, the B2B integration plug-in must be installed. For more information about the B2B integration plug-in, see “Plug-In Framework” on page 2-8.

Figure 2-3 Public and Private Processes



Public processes enable the interaction of trading partners in a collaborative, business-to-business arrangement by choreographing the exchange of business messages between them. These inter-enterprise processes have unique requirements that exceed those of private processes. The design and management of these processes is subject to the agreement of the trading partners. They are part of a formal contract that defines the content and semantics of the messages exchanged by the partners, and they may be standardized across an industry or industry segment.

Private processes are internal to an enterprise. They are not exposed outside of the enterprise, and customers or trading partners do not interact with them directly. A private process might handle the interaction between an enterprise and its customers, but the interaction takes place through a Web site. For example, placing an order for a computer on a Web site might trigger a private process that performs the following tasks: inventory check, credit card validation, transmission of a confirmation email, order building, and order shipment.

Execution

The underlying process engine manages the execution of business processes, automatically sequencing and implementing the prescribed process elements. During execution, XML is used for data representation, and JMS is used for messaging between workflows and other applications.

As Figure 2-1 shows, business processes can be started in a number of ways, depending on their design. Specifically, they can be:

- Called by an application or another workflow
- Invoked manually by people using a client application, such as the WebLogic Integration Worklist
- Triggered by the receipt of an event notification in the form of an XML message. The XML message can come from an application, another workflow, or a trading partner.
- Timed to start automatically at a predefined date and time

At run time, a workflow is converted into a workflow instance. For any workflow, multiple instances can run at the same time. Running instances of a workflow are saved in an instance store, which is implemented through a JDBC database, such as Oracle or SQL Server.

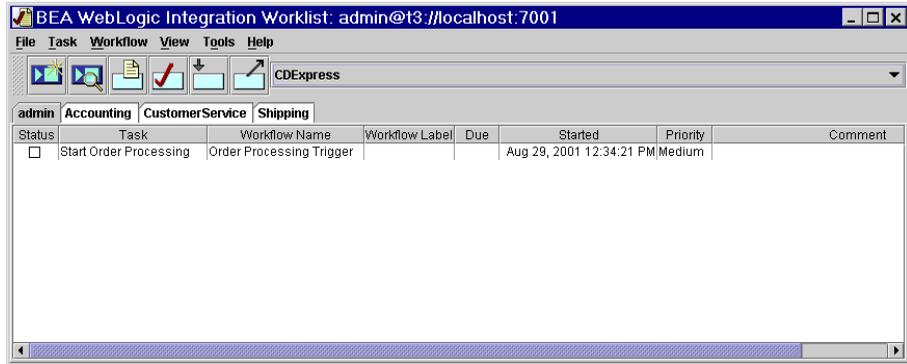
Monitoring and Optimization

Using the Studio, integration specialists can monitor the ongoing effectiveness of business processes by viewing processes in real time, and by collecting run-time statistics for reports. They can then tap into this data to evaluate processes, optimize performance and throughput, and increase uptime. Using this data, integration specialists can also modify ongoing business processes without necessarily interrupting them.

Run-Time Interaction

Business processes are sometimes designed to be started or stopped by people, or to include tasks that must be performed by people. These tasks might include making discretionary decisions, handling exceptions, or troubleshooting problems. WebLogic Integration provides an application called the Worklist that people can use to start and stop processes, and to interact with a running process. Using the Worklist, people can handle business process tasks assigned to them, such as making a decision about a customer's credit limit, or they can respond to messages from a process.

Figure 2-4 WebLogic Integration Worklist



APIs for Developing BPM Client Applications

WebLogic Integration provides application programming interfaces (APIs) that developers can use to create the following kinds of custom client applications:

- Configuration clients—Enable end users to configure security realms, business operations, event keys, and business calendars. The Studio uses the APIs to implement its configuration functionality.
- Design clients—Enable end users to perform a variety of workflow design tasks, such as developing workflow template definitions, creating workflow nodes, and importing and exporting workflow data. The Studio uses the APIs to implement its design-time functionality.
- Monitoring clients—Enable end users to monitor business processes in real time, and to collect run-time statistics for reports. The Studio uses the APIs to implement its workflow monitoring functionality.
- Run-time interaction clients—Enable end users to perform workflow interaction tasks, such as starting and stopping workflows, and executing workflow tasks assigned to them. The Worklist uses the APIs to implement its run-time interaction functionality.

Plug-In Framework

WebLogic Integration includes a *framework* that supports the development of *plug-ins*. Plug-ins are Java classes that extend the business process management functionality of WebLogic Integration. Developers can use the plug-in framework to integrate other products and technologies into the WebLogic Integration environment. Developers can also use plug-ins to modify the default design-time or run-time behavior of the following workflow components that are available in the Studio:

- Start, Event, and Done nodes
- Task actions
- Properties of workflow templates and template definitions
- Functions
- Message types
- Variable types

WebLogic Integration also includes the following plug-ins that enable integration specialists to integrate other areas of WebLogic Integration functionality with business processes:

- Application integration plug-in—Provides access from the Studio to *application views* deployed on WebLogic Server. Application views define access to enterprise systems and data. Integration specialists can define actions in workflow nodes that invoke application views to access enterprise systems and data. For details about application views, see Chapter 3, “Application Integration.”
- B2B integration plug-in—Provides workflow actions and extends workflow node properties in the Studio so that they can be used in a B2B integration environment. Integration specialists can use those actions and extended node properties to create public workflows that implement each *role* in a *conversation*. For details about public processes, roles, and conversations, see Chapter 4, “B2B Integration.”
- Data integration plug-in—Provides workflow actions in the Studio for translating binary data into XML and vice versa, and a binary variable type to hold binary data during translation. Integration specialists can specify those

actions in workflow nodes to perform data translation. For details about translating binary data to XML, see Chapter 5, “Data Integration.”

Sample Business Processes

WebLogic Integration provides samples that show how to use the Studio to design business processes, and that illustrate how to develop plug-ins that extend the business process management functionality provided by WebLogic Integration.

Sales Order

The sales order sample consists of three workflows: order processing, order fulfillment, and order-processing trigger. The workflows illustrate many of the most commonly used features of the Studio. They are packaged in a ready-to-use version so that integration specialists and analysts can use them for demonstrations of running workflows. The WebLogic Integration documentation provides a tutorial, called [Learning to Use BPM with WebLogic Integration](#), which describes the workflows, and teaches integration specialists how to develop them using the Studio.

Plug-In

The plug-in sample, loosely based on the sales order sample, consists of two workflows: order processing and order fulfillment. The workflows contain plug-ins that illustrate how the default functionality in certain workflow nodes can be extended. For example, the Start node in the order processing workflow contains a plug-in-defined event that shows how a workflow can be triggered by a custom event. When the plug-in is used, the Start node is triggered only by a custom XML event; it cannot be triggered by any of the default methods defined in the Start node.

3 Application Integration

Historically, enterprise information systems (EIS), such as legacy mainframe systems, and, more recently, commercial applications, such as ERP and CRM, have been at the center of the modern IT environment, driving enterprise-wide business functions, and providing critical data for enterprise operations. Today these systems are still vital to many businesses, but they are not the only systems on which businesses rely. In e-business environments, these systems must be linked not only with one another, but also with Web and wireless applications that enable the real-time exchange of information and services by customers, suppliers, and trading partners around the world.

To meet this demand, WebLogic Integration provides an application integration framework that enables intra-enterprise integration through the use of *adapters*. WebLogic Integration also provides an Adapter Developer Kit (ADK) that businesses can use to develop custom adapters. Using adapters, businesses can share data and business processes between connected applications without having to change their original applications or data structures.

The following sections describe the application integration functionality provided by WebLogic Integration:

- Application Integration Framework
- Integration with Business Processes
- Adapter Development Kit
- Sample DBMS Adapter

Application Integration Framework

The application integration framework provides the following key features to enable the integration of diverse enterprise systems:

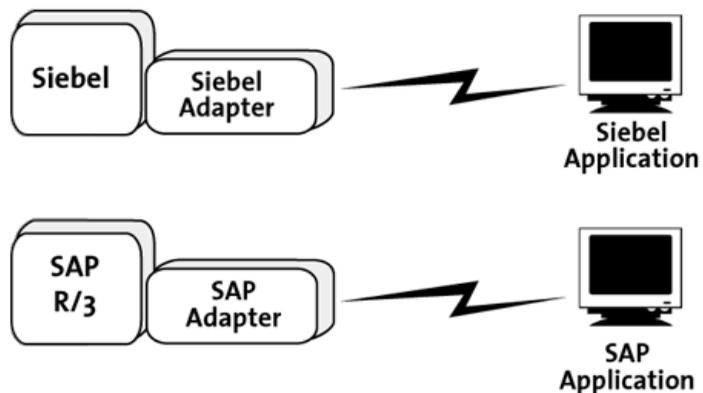
- Standards-based architecture for hosting J2EE Connector Architecture (J2EE CA)-based adapters that connect enterprise applications to WebLogic Server
- Application views for both event and service adapters

The following sections provide details about each of these features.

Hosting J2EE CA-Based Adapters

The traditional approach to enterprise application integration is to connect individual systems with each other; in effect, to hardwire the systems together. This approach is fraught with complexity because connecting different systems requires an intimate, low-level knowledge of each proprietary technology. To remove this complexity, businesses use adapters.

Figure 3-1 EIS Adapters



Adapters are software components that applications can use to access enterprise data programmatically. For example, an adapter can use a Java class to represent enterprise data, and it can provide methods that applications can invoke to access the data. When an application invokes an access method, the adapter executes the method to retrieve the enterprise data.

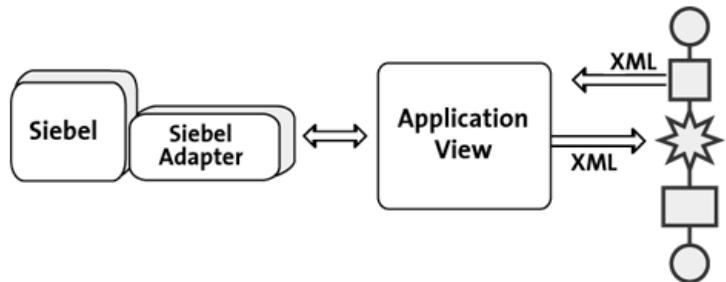
The widespread adoption of standards, such as J2EE and XML, has laid the groundwork for a standardized approach to the development of adapters. Perhaps the most significant of these standards for application integration is the J2EE Connector Architecture (CA). J2EE CA provides a standardized approach for the development of adapters for all types of applications—from legacy mainframe applications, such as CICS, to packaged applications, such as PeopleSoft, Siebel, and SAP. The adoption of standards such as J2EE CA enables businesses to develop adapters that work on any J2EE-compliant application server, such as WebLogic Server.

WebLogic Integration builds on this standardized approach in its application integration framework by providing a standards-based architecture for hosting J2EE CA-based adapters. Using the ADK, developers can build J2EE CA-compliant adapters, and then deploy them in the integration framework to connect enterprise applications to WebLogic Server.

Application Views

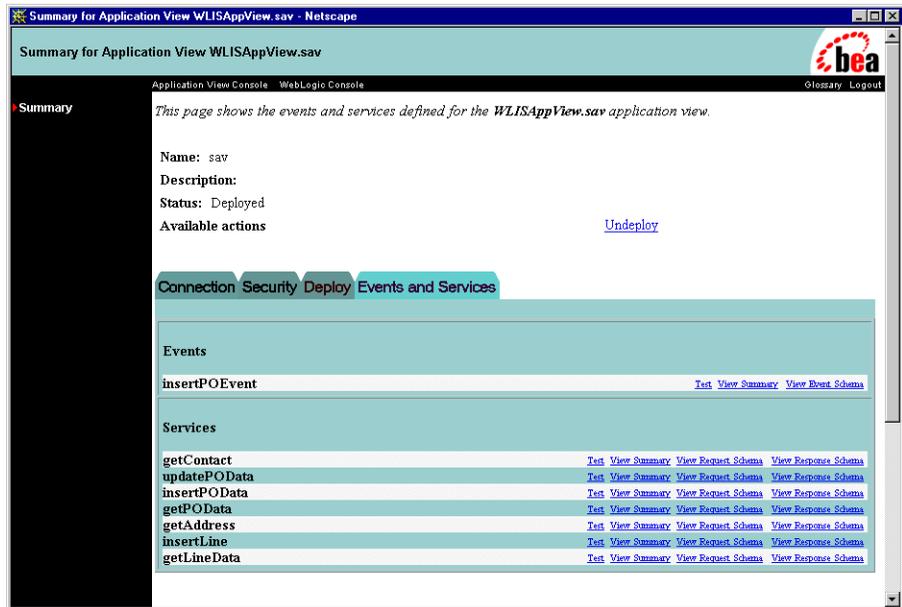
The integration framework goes one step beyond the programmatic access adapters provide to enterprise systems with an innovation that makes it possible to expose existing information systems as business services. This innovation is called the *application view* of the adapter. The application view provides a business-oriented way for integration specialists to access enterprise data without worrying about the programmatic details defined in an adapter.

Figure 3-2 Adapters and Application Views



The idea behind application views is quite simple. First, a developer codes an adapter that connects the enterprise system to WebLogic Server. The adapter exposes the functionality in an application that accesses enterprise data. The functionality the adapter exposes might, for example, update records in a database using SQL statements, or it might request information from an SAP system using its BAPI or IDOC interfaces. Then, a business analyst, working with the developer, defines an application view of the adapter using the Application View Console provided by WebLogic Integration.

Figure 3-3 WebLogic Integration Application View Console



The application view abstracts the technical details in the adapter so that a business analyst can invoke the functionality in the enterprise application exposed by the adapter without worrying about the technical implementation. For example, to update records in a database using SQL statements, a business analyst can define an application view called `UpdateRecords`. An EIS specialist supplies the SQL statements that update the database records, and includes them in the application view definition.

At run time, the application view called `UpdateRecords` is invoked by an XML message from a business process. The application view, in turn, calls the adapter, passing the SQL statements to it. The adapter accesses the enterprise data and returns the result to the `UpdateRecords` application view, which returns the result in another XML message to the business process.

Application View Benefits

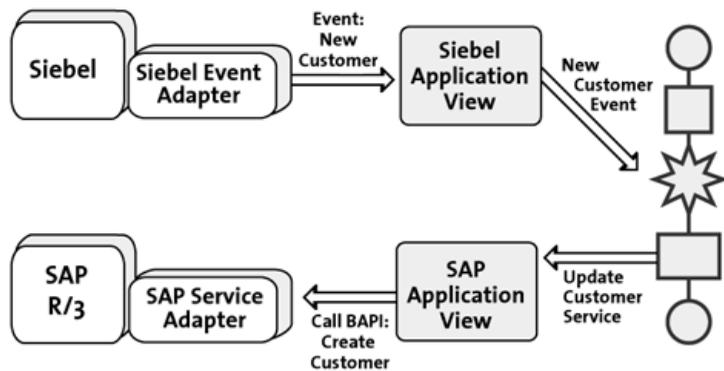
Using application views makes it possible to expose existing enterprise applications as business services that accept XML data as input and return XML data as output. Using XML as the common format to represent enterprise data makes it possible to treat

different systems in similar ways. The basic difference between integrating with a SAP system and a Siebel system, from the perspective of the business analyst, is the content of the XML messages. The technical details of the system being integrated are made transparent, enabling the business analyst to focus on the business service being called to solve a business problem.

Application View Services and Events

Adapters can be developed as *service* or *event* adapters. A service adapter invokes a specific function in the enterprise system to which it is connected. An event adapter propagates information from the enterprise system to the integration environment. Businesses can add any number of services and events to each application view definition to support specific types of transactions between WebLogic Server and the target enterprise system.

Figure 3-4 Application View Services and Events



When added to an application view, each event responds to a particular enterprise application state. When a particular event is triggered, the event adapter extracts data about the event from the enterprise system, and propagates the data in an XML document to the WebLogic Integration environment using an application view.

Similarly, when added to an application view, each service represents a particular function in the enterprise system. When the application view receives an XML document that maps to a particular service, it invokes the corresponding function in the enterprise system using the service adapter. The enterprise system may or may not send a response to the service, depending on how it is designed.

Integration with Business Processes

WebLogic Integration provides a plug-in framework that enables the integration of application views with business processes. To integrate application views with business processes, integration specialists do the following:

- Define the application view using the Application View Console
- Deploy the application view on WebLogic Server
- Define a business process using the WebLogic Integration Studio, specifying an application integration action in one of the workflow nodes that invokes the application view

Using this scheme, application views can be updated without disturbing the business processes that use them, even if the implementation details of the application views change.

Adapter Development Kit

WebLogic Integration provides an Adapter Development Kit (ADK) that businesses can use to develop service and event adapters that run in any J2EE-compliant environment, such as WebLogic Server. The ADK provides the following collection of frameworks for the development, testing, packaging, and distribution of adapters:

- Design-time framework—Provides the tools that enable developers to create a Web-based GUI for defining and deploying application views.
- Run-time framework—Provides a complete, extensible event generator to support the development of event adapters. To support the development of service adapters, the run-time framework provides a complete, but minimally functional J2EE-compliant adapter. Adapter developers can save coding time by starting with and extending this base framework.
- Logging and auditing framework—Provides the developer with the support needed to develop error, audit, and debugging logs that comply with the Java Logging Specification.

- Packaging framework—Provides a set of tools for packaging adapters so they can be installed, configured, and uninstalled on WebLogic Server in a consistent manner.

Sample DBMS Adapter

WebLogic Integration includes a sample DBMS adapter. The sample is a J2EE CA-compliant adapter built using the ADK that provides a way for an application to access a relational database. Developers can use the sample adapter to learn how to develop adapters using the ADK.

The sample adapter includes the following features:

- GUI that enables developers to connect to a Cloudscape, Oracle, or SQL Server database, and to browse the catalogs, schemas, tables, and columns of the underlying database.
- Support for the development and testing of application views with events and services. Services are provided to select, insert, delete, and update entries in the underlying database.
- Documentation that describes the code and Javadoc so developers can understand how to use and extend the classes of the ADK to build a J2EE CA-compliant adapter.

4 B2B Integration

Enterprises interact with customers, suppliers, and trading partners in different ways. Many businesses use electronic data interchange (EDI) for business communication. Using this technology, business systems exchange structured messages using agreed upon message standards over private networks run by EDI service companies. Other businesses engage in collaborative arrangements with a variety of trading partners over the Internet. In these arrangements, trading partners need to link existing back-end applications, databases, and customers together so they can engage in real-time business transactions using a variety of business protocols.

WebLogic Integration provides the next-generation infrastructure that enables this level of inter-enterprise integration. WebLogic Integration provides a B2B integration framework for messaging, connectivity, and business protocols that businesses can use to develop collaborative arrangements with numerous trading partners over the Internet. WebLogic Integration also enables businesses to integrate EDI environments with WebLogic Integration.

The following sections describe the B2B integration functionality provided by WebLogic Integration:

- B2B Integration Framework
- Integration with Business Processes
- APIs and Logic Plug-Ins for B2B Application Development
- Samples
- EDI Integration

B2B Integration Framework

The B2B integration framework provides the following key features to enable collaboration between trading partners:

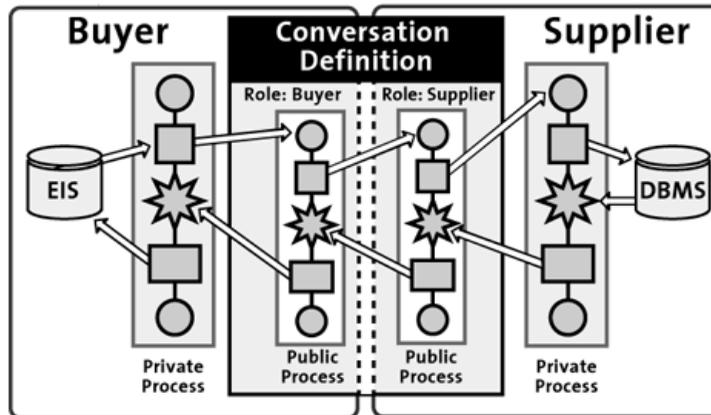
- Conversation definition and monitoring
- Trading partner configuration and management
- Business protocol support
- Collaboration agreement definition and management
- Security services
- Lightweight client support

The following sections provide details about each of these features.

Conversation Definition and Monitoring

In an e-business environment, collaboration between trading partners occurs through the exchange of business messages that contain XML or nonXML documents in a secure, choreographed arrangement called a *conversation*. A conversation is, quite simply, a series of business messages exchanged between trading partners.

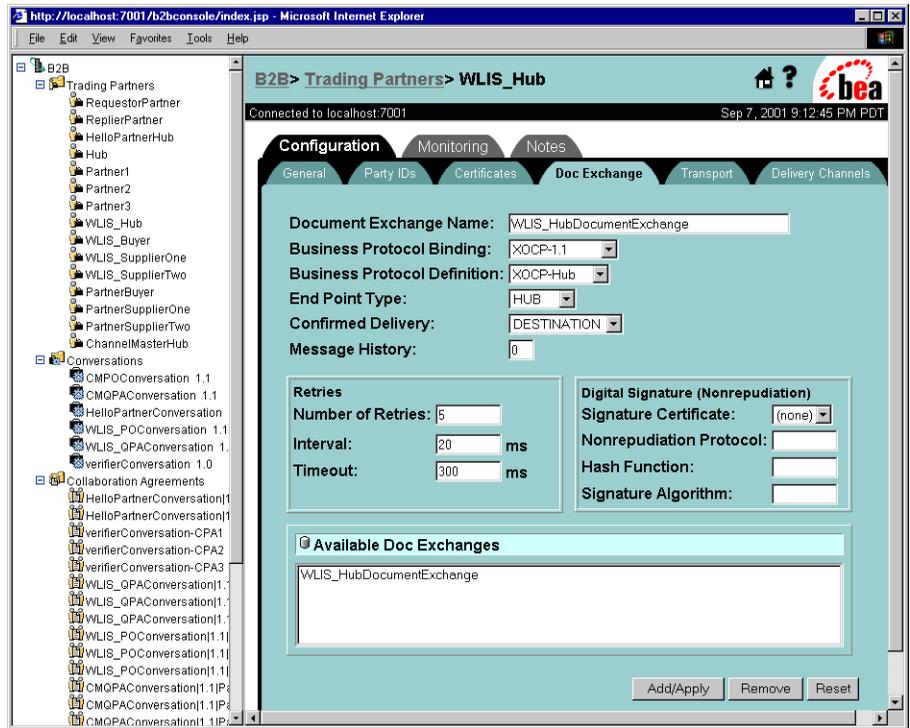
Figure 4-1 Conversation Between Trading Partners



As Figure 4-1 shows, the composition of business messages and the sequence of an exchange is typically handled by *collaborative* or *public* business processes. The composition and sequence of messages can also be handled by Java messaging applications. Conversations can be complex and long-running, or they can be short-lived. Each conversation has a unique name. Each participant in a conversation has a *conversation role*, such as that of a buyer or a supplier in a supply-chain arrangement.

All the details of a conversation, including its name and version, the roles of the participants, and the business protocols it uses, are specified in a *conversation definition*. Integration specialists create conversation definitions and monitor running conversations using the WebLogic Integration B2B Console. The following figure shows configuration information for a trading partner.

Figure 4-2 WebLogic Integration B2B Console



Trading Partner Configuration and Management

An e-commerce community is formed when a trading partner joins other trading partners to pursue a common business objective. An e-commerce community can exist in different forms, and for different purposes. It might, for example:

- Span multiple departments within a company to manage inventory across the company
- Span multiple companies across firewalls and over the Internet to manage a supply chain or a multistep purchasing arrangement

- Include trading partners both within a company and in other companies, such that one or more trading partners interact with trading partners in other companies

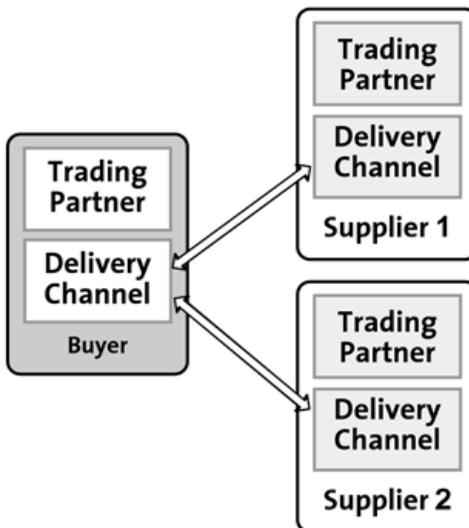
To participate in the conversations of an e-commerce community, integration specialists use the B2B Console to configure trading partners. Specifically, they assign trading partners the names by which they will be known in the conversation, and they specifying the delivery channels to be used for the exchange of business messages.

A delivery channel defines how a trading partner sends and receives messages. It also specifies the business protocol to be used in the conversation (such as XOCP), the transport protocol (such as HTTP), and security parameters. Trading partners can configure their delivery channels to communicate directly with each other in a peer-to-peer configuration, or through an intermediary when deployed in a hub-and-spoke configuration.

Peer-to-Peer Configuration

In a peer-to-peer configuration, trading partners communicate with each other directly through their respective delivery channels, using the RosettaNet or cXML business protocol.

Figure 4-3 Peer-to-Peer Configuration

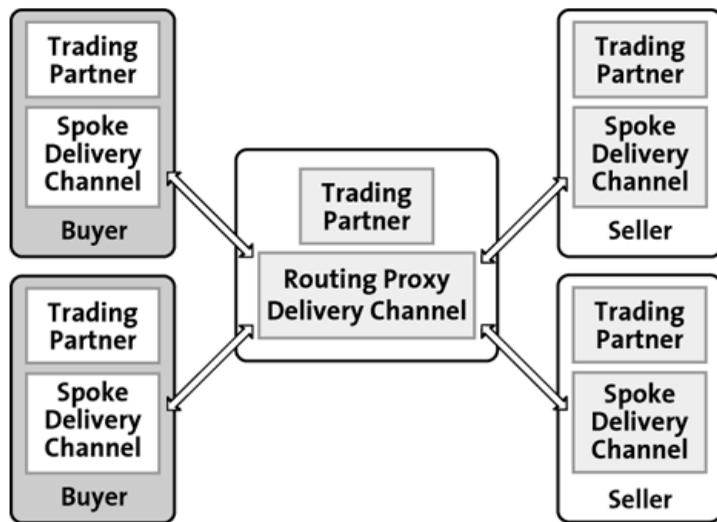


In this type of configuration, a single trading partner is the controlling entity, and the other trading partners are integrated with it. A peer-to-peer configuration might be used to integrate an enterprise with its suppliers in a supply-chain arrangement.

Hub-and-Spoke Configuration

In a hub-and-spoke configuration, trading partners communicate with each other through an intermediary, or routing-proxy delivery channel, using the XOCP business protocol.

Figure 4-4 Hub-and-Spoke Configuration



In this type of configuration, a single trading partner is the hub that mediates the exchange of messages between the other trading partners enlisted in the conversation. The hub trading partner can perform such tasks as routing and filtering messages, or it can provide customized services to the other trading partners in the conversation. A hub-and-spoke configuration might be used to link buyers and sellers in an electronic marketplace.

Business Protocol Support

Trading partners can use any of the following business protocols supported by WebLogic Integration:

- RosettaNet Implementation Framework 1.1 and 2.0—An industry-wide standard that defines communication protocols for supply-chain partners on a global basis. RosettaNet provides specifications for the RosettaNet Implementation Framework (RNIF), the Partner Interface Processes (PIPs), and business and technical dictionaries.
- cXML (Commerce eXtensible Markup Language)—A protocol that is designed to link buyers and suppliers in arrangements created to allow buyers to browse catalogs and submit purchase orders to suppliers. Buyers can browse a supplier’s catalog directly, or businesses can use an Ariba Commerce Server Network to mediate messages between buyers and suppliers. The B2B integration framework provides a cXML API so that trading partners can send and receive cXML messages.
- XOCP (eXtensible Open Collaboration Protocol)—A BEA-specific protocol that is designed to manage long-lived conversations. XOCP supports message multicasting. It also enables businesses to define a number of Quality of Service (QoS) settings, including message durability, length of the timeout period, and number of retry attempts.

Collaboration Agreement Definition and Management

Collaboration agreements are a central component of B2B integration because they tie together all the pieces that we have been discussing: conversations and roles, collaborative business processes, and trading partners and delivery channels. Integration specialists use the B2B Console to define collaboration agreements, which map trading partners to the roles specified in conversation definitions. Collaboration agreements reference the delivery channel each trading partner uses, and the business process each role uses to define the sequence in which a role exchanges business messages with other roles in a conversation.

Security Services

The security services provided by the B2B integration framework are built on top of the security services provided by WebLogic Server. They include the following features:

- An SSL-based secure platform for conversations
- Certificate verification that can be used to authenticate the identities of trading partners
- Digital signatures that can be attached to business messages being exchanged by trading partners
- Support for nonrepudiation of origin and nonrepudiation of receipt, which are often required by law for critical business messages
- Data encryption for business protocols that require this support

Lightweight Client Support

The B2B integration framework provides support for lightweight clients so that small and medium-size enterprises, or enterprises with few or no back-end integration requirements, can participate in e-business communities simply and inexpensively. Such enterprises can use a Web browser or a file-sharing client to communicate with other trading partners, one of which must have WebLogic Integration deployed. By using lightweight client support, trading partners can enlist other partners in shared business processes without requiring them to commit any in-house technical resources, and they can leverage their integration solutions across a wider audience.

Integration with Business Processes

WebLogic Integration provides a plug-in framework for creating collaborative or public business processes that implement the various roles in a conversation. To create public processes, integration specialists do the following:

- Define a trading partner conversation using the B2B Console. Information about the conversation includes a definition of the roles, and references to the business protocols and public business processes used by the roles in the conversation. This information is stored in the WebLogic Integration repository.
- Define public processes using the WebLogic Integration Studio, specifying workflow actions, events, and the node properties required to implement each role in a conversation.
- Once the public processes are tested and the conversation runs successfully, export the repository information to an XML file using the B2B Console. Other trading partners can then import the file and use it to set up their environment so they can participate in the conversation.

Public business processes are typically integrated with private business processes. The design and definition of private processes are specific to a particular organization. Private processes are not visible from outside an organization, and they are usually integrated with back-end business systems. As Figure 4-1 shows, a private process can execute an internal business activity, such as retrieving information from a database, and then it can return the result to a public process, which, in turn, forwards the result to a trading partner.

APIs and Logic Plug-Ins for B2B Application Development

WebLogic Integration provides the following application programming interfaces (APIs) and Java classes that developers can use for the creation of B2B applications:

- **Managed Beans (MBeans)**—BEA implementation of the Java Management Extensions (JMX) Managed Beans API. This API enables developers to create custom applications to monitor aspects of run-time conversations, such as message traffic and run-time statistics for server nodes, delivery channels, business transaction definitions, trading partners, and business messages. The functionality provided by the API is comparable to that provided by the B2B Console.

- **Messaging API**—Used to develop XOCP applications to exchange business messages. An XOCP application implements a trading partner role to manage a conversation and to handle business messages.
- **Logging API**—Used to develop B2B applications that can write error, warning, and information messages to a message log.
- **Logic plug-ins**—Java classes that an intermediary can use to intercept a business message and then to perform additional processing on the message, such as routing and filtering the message, integrating the message with a back-end system, validating the content of the message, or logging the message in a transaction log.

Samples

WebLogic Integration provides the following samples that integration specialists can use to model B2B integration solutions:

- Hello Partner
- Channel Master
- RosettaNet 2.0 Security
- Lightweight Client
- Messaging API

Hello Partner

The Hello Partner sample demonstrates how two or more trading partners can participate in business communication in a hub-and-spoke configuration. Each trading partner includes public and private business processes.

The public processes handle communication between the partners, exchanging messages using the XOCP protocol. The private processes handle the message content processing. The private processes interact with the public ones to deliver messages to trading partners, and with an associated Java application to process message content.

Channel Master

The Channel Master sample shows how a large trading partner uses WebLogic Integration to automate its supply chain.

The interactions between the trading partners occur in the following sequence:

1. The channel master buyer broadcasts a query for the pricing and availability of a particular item to two supplier trading partners. The broadcast provides an example of multicast communication using the XOCB protocol.
2. The two supplier trading partners send a reply quote with the price and availability of the requested item to the buyer trading partner, demonstrating point-to-point communications with the buyer.
3. The buyer selects a supplier and sends a purchase order.
4. The selected supplier returns a purchase order acknowledgment.

The interactions between the buyer and the selected supplier also demonstrate point-to-point communication.

RosettaNet 2.0 Security

The RosettaNet sample shows how WebLogic Integration can be used to implement RosettaNet 2.0 PIP 3A2 and PIP 0A1 using workflows. This sample shows two trading partners exchanging business messages that conform to the RosettaNet 2.0 PIP 3A2 standard.

The interactions between the trading partners occur in the following sequence:

1. A customer trading partner sends a price and availability request to a supplier trading partner.
2. The supplier returns an acknowledgment, indicating that it received the request.
3. The supplier then returns a price and availability response.
4. The customer sends an acknowledgment to the supplier, indicating that it received the response.

Lightweight Client

The Lightweight Client sample demonstrates how communication can take place between a requestor and a replier trading partner when they do not have WebLogic Integration installed. Communication is achieved using two types of lightweight clients. The requestor trading partner uses a Web browser client, and the replier uses a file-sharing client.

In the sample, the requestor uses a Web browser to access JSPs served from a remote installation of WebLogic Integration that is configured to handle lightweight clients. The JSPs create mailboxes that the replier then accesses using its file-sharing client.

Messaging API

The Messaging API sample demonstrates two message delivery methods: synchronous messaging and deferred synchronous messaging. When synchronous messaging is used, a trading partner that sends a message waits for a return from the receiving partner before it continues to execute other tasks. When deferred synchronous messaging is used, the sending partner does not wait for a return from the receiving partner, but continues executing other tasks.

The interactions between the trading partners occur in the following sequence:

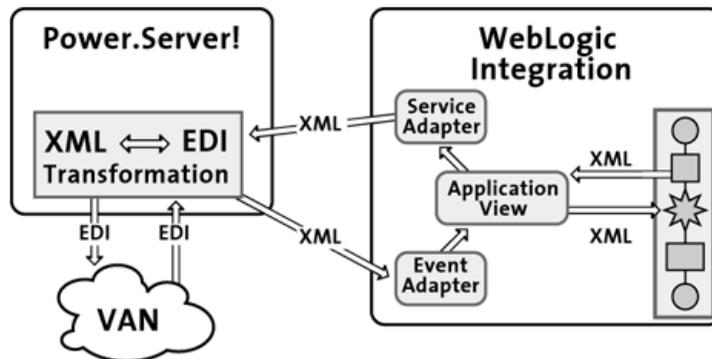
1. Partner 1 sends a deferred synchronous message to Partner 2. A logic plug-in delays the delivery of the message for fifteen seconds. Logic plug-ins are Java classes that can intercept and process business messages at run time
2. During the delay, Partner 1 sends a synchronous message to Partner 3.
3. When control returns to Partner 1, it checks to see that the first message was received by Partner 2.
4. Partner 3 sends a reply to Partner 1.
5. Partner 2 then sends a reply to Partner 1.

EDI Integration

Electronic data interchange (EDI) is a means by which businesses communicate with each other electronically in a structured, standardized way. In an EDI environment, businesses place orders or perform financial transactions by exchanging business messages between computer systems. EDI messages are structured according to an agreed upon message standard, and are processed automatically without human intervention. The structured data in EDI messages is exchanged over private networks called value added networks (VANs). VANs are managed and supported by EDI service companies, and function like electronic post offices, routing electronic messages from sender to recipient systems.

EDI integration functionality connects EDI environments with WebLogic Integration, making it possible to integrate XML-based transactions with EDI-based transactions.

Figure 4-5 EDI Integration



To connect EDI and WebLogic Integration, WebLogic Integration provides J2EE CA-compliant service and event adapters that connect to Power.Enterprise!, an EDI-capable system. WebLogic Integration also includes an application view that integration specialists can use to integrate business processes with the EDI system.

Using the application view service in a business process, applications can invoke specific functionality in the EDI system using an XML message. Using the application view event, the EDI system can propagate information to WebLogic Integration using

an EDI message. The conversion from EDI to XML, and vice versa, is handled by the EDI-to-XML transformation engine bundled with Power.Server!, the EDI server provided by Power.Enterprise!

Sample EDI Application

WebLogic Integration provides an EDI sample application that demonstrates how WebLogic Integration and Power.Enterprise! can be used to exchange EDI purchase-order information over a VAN. In the sample application, a supplier trading partner uses the EDI integration functionality of WebLogic Integration to connect to a buyer over a VAN.

The interactions between the buyer and supplier occur in the following sequence:

1. A buyer trading partner submits an EDI purchase order over a VAN to the supplier
2. The EDI-to-XML transformation engine bundled with Power.Server! converts the purchase order to XML
3. The XML document triggers a business process in the supplier application. The business process generates an XML purchase order acknowledgement
4. The supplier forwards the acknowledgement to the transformation engine which converts it to EDI, and then forwards it over a VAN to the buyer

5 Data Integration

Heterogeneous environments are now commonplace in modern enterprises. These environments are home to heterogeneous applications, each with its own data format. This variety of data formats can make it difficult to integrate data across business systems. Before different systems can understand the data they receive, the data must be translated or transformed into the appropriate format. To facilitate the integration of data across different systems, businesses are adopting XML as the standard format for information exchange.

The adoption of XML as a standard format for data exchange greatly simplifies both intra- and inter-enterprise integration. Within an enterprise, various data formats can be translated into XML, and then seamlessly exchanged with other applications. XML documents can be further transformed into other XML grammars, and then exchanged efficiently over the Internet between trading partners.

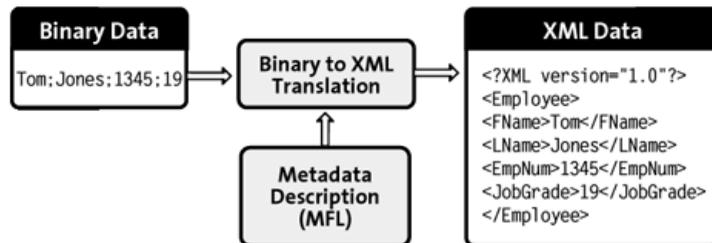
WebLogic Integration provides support for the translation of legacy data into XML, and the transformation of XML documents into other XML grammars. The following sections describe the data integration functionality WebLogic Integration provides:

- Data Translation
- Data Transformation

Data Translation

To support the integration of data from diverse enterprise applications, WebLogic Integration enables the translation of binary data to XML, and vice versa. When binary data is translated to XML, structured binary data is converted into an XML document so that the data can be accessed using standard XML parsing methods.

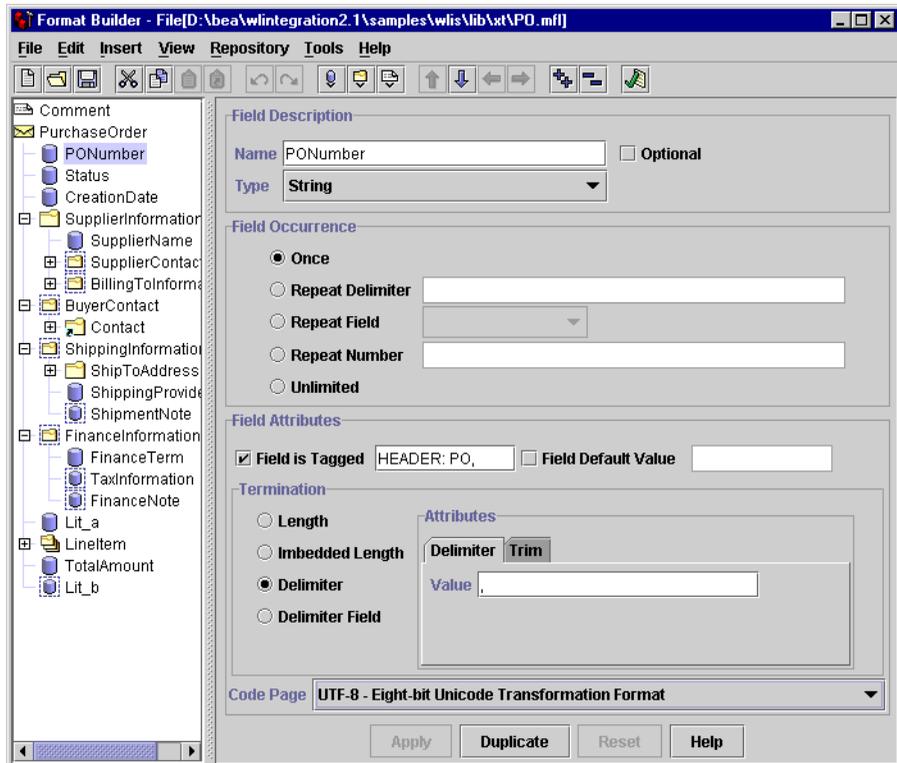
Figure 5-1 Translating Binary Data into XML



To translate binary data, developers first create a metadata description of the record layout in the binary data. The metadata includes descriptions of the fields in the records, including field types (such as string or floating point) and field names. The metadata is stored in a Message Format Language (MFL) file. At run time, the MFL document is used to translate binary data into an XML document, and vice versa.

To create MFL files, integration specialists use the Format Builder tool provided by WebLogic Integration. Integration specialists can also use the Format Builder to produce a DTD file or XML schema that describes the XML document created from the translation.

Figure 5-2 Format Builder



To perform the translation, applications can invoke the translation functionality programmatically, or business processes can invoke the functionality from within their workflows. The resulting XML documents can be used directly in XML-enabled applications or to start other business processes. They can also be transformed into other XML grammars.

Integration with Business Processes

WebLogic Integration provides a plug-in framework that enables the integration of data translation with business processes. To integrate data translation functionality with business processes, integration specialists can use two workflow actions:

- Translate Binary to XML
- Translate XML to Binary

Each action is used to specify an MFL file that contains a metadata description of the binary data, and the location of the source or target data, whether binary or XML.

To use these actions for data translation, integration specialists define a business process using the WebLogic Integration Studio, specifying one of the data translation actions in a workflow node. At run time, the specified action is executed, translating the data into the appropriate format.

Triggering Business Processes Using Binary Data

Business processes are triggered by the receipt of an XML document. WebLogic Integration provides a data translation plug-in that enables binary data to trigger business processes by translating the binary data into XML. The functionality in the plug-in is called the *event handler*.

The publication of a JMS message to an internal JMS queue invokes the event handler. The event handler causes the binary data in the JMS message to be translated into an XML document. The XML document can then be used to trigger a business process.

The use of the event handler is demonstrated in the Servlet sample application supplied with WebLogic Integration. For details about the Servlet sample, see the following section.

Samples

WebLogic Integration provides three samples that illustrate the integration of the data translation functionality with business processes, and the use of the Format Builder to create message format definitions.

Servlet Applicaton

The Servlet application installs a servlet that uses the binary data event handler to convert binary data to XML. The servlet can be accessed through a Web browser. The servlet accepts requests for the conversion of binary data to XML, and responds by displaying the generated XML data. In addition, the servlet posts the XML data to an internal JMS queue that can be used to trigger a business process.

EJB Application

This EJB application simulates a business process in which data flows from a human resources system to a payroll system. In the process, data flows in the following sequence:

1. The business process is initiated manually when an end user enters payroll data.
2. Using the payroll data, the process obtains employee data from a legacy payroll system that uses binary data.
3. The binary data is translated to XML so that salary information for the employee can be calculated.
4. The result of the calculation is translated back to binary format. It is then sent back to the payroll system for further processing.

Purchase Order

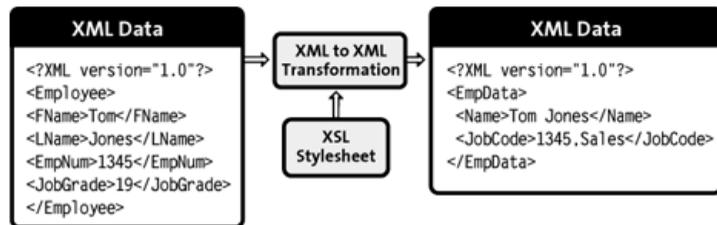
The Purchase Order sample demonstrates how to create DTD and MFL files using the Format Builder. The sample consists of DTD, MFL, and data files that contain binary data. Developers can use the data files as input to the Format Builder to develop their own DTD and MFL files. They can use the DTD and MFL files supplied with the sample to verify that the files they create are correct.

Data Transformation

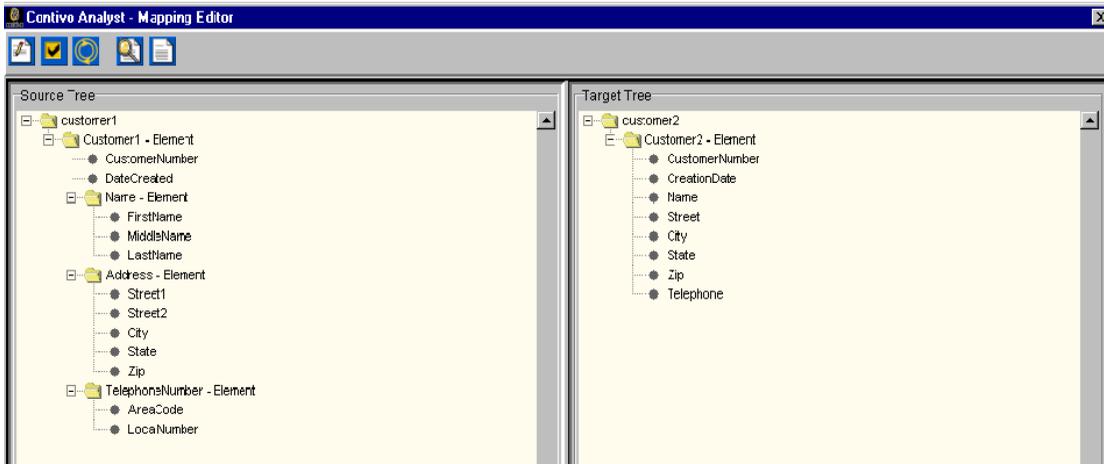
Once binary data is translated into an XML document, a business can transform an XML document into a specific XML grammar, such as RosettaNet or ebXML. The process of transforming an XML document into another XML grammar or display format is called *XML transformation*.

XSLT is a specialized XML language that defines rules for translating an XML document into another XML or nonXML document. An XSL stylesheet specifies which elements in an XML document are to be transformed, and how they are to be transformed. The stylesheet maps elements in a source XML document to elements in a target XML document with a different XML grammar.

Figure 5-3 XML Transformation



To transform an XML document, integration specialists first create an XSL stylesheet that describes how one XML grammar is mapped to another XML grammar. Integration specialists can create the XSL stylesheet manually, using the design-time GUI provided by Contivo Analyst, or they can use the auto-mapping functionality in Contivo Analyst and Contivo eService.

Figure 5-4 Contivo Analyst Mapping Editor

To create the XSL stylesheet, integration specialists first need to specify the DTD or XML schema for the source and target XML documents. They can retrieve the DTD and schema files from the WebLogic Integration repository using Contivo Analyst. Once they create the XSL stylesheet, they can store it in the WebLogic Integration repository.

The actual transformation is performed at run-time by the XSL transformation engine bundled with WebLogic Server. Integration specialists define when the transformation is to occur in a business process developed using the business process management functionality provided by WebLogic Integration.

Integration with Business Processes

WebLogic Integration provides an XSL Transform action to integrate data transformation with business processes. To integrate data transformation functionality with business processes, integration specialists define a business process using the WebLogic Integration Studio, specifying the XSL Transform action in a workflow node.

At run time, the XSL Transform action defined in the business process invokes the transformation engine bundled with WebLogic Server. The transformation engine uses the XSL stylesheet sheet stored in the WebLogic Integration repository to map

elements from the source XML document to the target document. The resulting XML document can be used to complete a business process, it can be sent for further processing to a trading partner, or it can be used in other XML-enabled applications.

A Documentation Roadmap

WebLogic Integration provides extensive functionality that analysts and developers can use to simplify and standardize application development, deployment, and business integration for an enterprise. To help you understand and use this functionality effectively, we provide a comprehensive library of documentation at the following Web site:

<http://edocs.bea.com>

The following sections list the documentation that is available for WebLogic Integration:

- WebLogic Integration
- Business Process Management
- Application Integration
- B2B Integration
- Data Integration

WebLogic Integration

This section lists the documentation available for WebLogic Integration.

Table A-1 WebLogic Integration Documentation

To . . .	Read . . .
Learn about the functionality provided by WebLogic Integration	<i>Introducing BEA WebLogic Integration</i>
Set up and run the WebLogic Integration sample application	<i>Learning to Use BEA WebLogic Integration</i>
Design and architect WebLogic Integration solutions following recommended best practices	<i>Designing BEA WebLogic Integration Applications</i>
Move an integrated solution from a development environment to a production environment	<i>Deploying BEA WebLogic Integration</i>
Install WebLogic Integration	<i>Installing BEA WebLogic Integration</i>
Start and stop WebLogic Integration design and administration tools, and to configure a custom domain	<i>Starting, Stopping, and Customizing BEA WebLogic Integration</i>
Migrate to the current release of WebLogic Integration	<i>Migrating to BEA WebLogic Integration Release 2.1</i>
Program WebLogic Integration applications using the APIs	<i>BEA WebLogic Integration Javadoc</i>
Understand WebLogic Integration terminology	<i>BEA WebLogic Integration Glossary</i>
Learn about new features, supported platforms, and known limitations	<i>BEA WebLogic Integration Release Notes</i>
Solve common problems when using WebLogic Integration	<i>Troubleshooting Tips and FAQs for BEA WebLogic Integration</i>

Business Process Management

This section lists the documentation available for the business process management functionality provided by WebLogic Integration.

Table A-2 Business Process Management Documentation

To . . .	Read . . .
Develop and monitor business processes using the WebLogic Integration Studio	<ul style="list-style-type: none"> ■ Learning to Use BPM with WebLogic Integration ■ Using the WebLogic Integration Studio ■ Online Help for the WebLogic Integration Studio
Manage and interact with business processes using the WebLogic Integration Worklist	Using the WebLogic Integration Worklist
Develop custom client applications to manage and interact with business processes	Programming BPM Client Applications
Develop custom code that extends the business process management functionality provided by WebLogic Integration	Programming BPM Plug-Ins for WebLogic Integration

Application Integration

This section lists the documentation available for the application integration functionality provided by WebLogic Integration.

Table A-3 Application Integration Documentation

To . . .	Read . . .
Learn about the application integration functionality and the ADK	Introducing Application Integration
Configure and deploy application views, and learn how to integrate them into business processes	Implementing Application Integration
Develop event and service adapters, and a design-time interface for using an adapter	Developing Adapters with Application Integration
Use the application integration plug-in in business processes	Online Help for Application Integration

B2B Integration

This section lists the documentation available for the B2B integration functionality provided by WebLogic Integration.

Table A-4 B2B Integration Documentation

To . . .	Read . . .
Learn about the B2B integration functionality	Introducing B2B Integration
Use the B2B Console to administer and monitor trading partner conversations	<ul style="list-style-type: none"> ■ Administering B2B Integration ■ Online Help for the WebLogic Integration B2B Console

Table A-4 B2B Integration Documentation (Continued)

To . . .	Read . . .
Configure trading partner authentication, authorization, and other security settings	<i>Implementing Security with B2B Integration</i>
Configure and run the B2B sample applications	<i>Running the B2B Integration Samples</i>
Create business processes that implement trading partner conversations	<i>Creating Workflows for B2B Integration</i>
Use the RosettaNet business protocol for communication between trading partners	<i>Implementing RosettaNet for B2B Integration</i>
Use the cXML business protocol for communication between trading partners	<i>Implementing cXML for B2B Integration</i>
Create logic plug-ins that perform specialized processing of business processes	<i>Programming Logic Plug-Ins for B2B Integration</i>
Develop management applications that monitor run-time conversation activities	<i>Programming Management Applications for B2B Integration</i>
Use the Messaging API to develop Java applications to exchange XOCP business messages	<i>Programming Messaging Applications for B2B Integration</i>
Use the Logging API to enable B2B applications to write messages to the message log	<i>Writing Messages to the B2B Integration Log</i>
Configure WebLogic Integration and Power.Enterprise!, and run the EDI sample application	<i>Using EDI with WebLogic Integration</i>

Data Integration

This section lists the documentation available for the data integration functionality provided by WebLogic Integration.

Table A-5 Data Integration Documentation

To . . .	Read . . .
Develop and test message format definitions using the Format Builder	<i>Integrating Data with WebLogic Integration</i>
Use the data integration plug-in in business processes	<ul style="list-style-type: none">■ <i>Guidelines for Data Integration Plug-Ins</i>■ <i>Online Help for Data Integration Plug-Ins</i>