



BEA WebLogic Platform™

**Introducing BEA
WebLogic Platform**

Release 7.0 Service Pack 6
Document Date: March 2005

Copyright

Copyright © 2005 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, BEA Liquid Data for WebLogic, BEA WebLogic Server, Built on BEA, Jolt, JoltBeans, SteelThread, Top End, Tuxedo, and WebLogic are registered trademarks of BEA Systems, Inc. BEA Builder, BEA Campaign Manager for WebLogic, BEA eLink, BEA Manager, BEA MessageQ, BEA WebLogic Commerce Server, BEA WebLogic Enterprise, BEA WebLogic Enterprise Platform, BEA WebLogic Enterprise Security, BEA WebLogic Express, BEA WebLogic Integration, BEA WebLogic Java Adapter for Mainframe, BEA WebLogic JDriver, BEA WebLogic JRockit, BEA WebLogic Log Central, BEA WebLogic Personalization Server, BEA WebLogic Platform, BEA WebLogic Portal, BEA WebLogic Server Process Edition, BEA WebLogic WorkGroup Edition, BEA WebLogic Workshop, and Liquid Computing are trademarks of BEA Systems, Inc. BEA Mission Critical Support is a service mark of BEA Systems, Inc. All other company and product names may be the subject of intellectual property rights reserved by third parties.

All other trademarks are the property of their respective companies.

Contents

About This Document

What You Need to Know	vi
e-docs Web Site	vi
How to Print the Document	vi
Related Information	vii
Contact Us!	vii
Documentation Conventions	viii
Introducing BEA WebLogic Platform	1
WebLogic Platform Features	2
WebLogic Platform Benefits	3
Enterprise Challenges Confronted	4
Summary	4
Developing J2EE Applications	6
Employing the Power and Flexibility of WebLogic Server	6
Developing Enterprise-Class Web Services	9
What Are Web Services?	10
Web Service Standards	11
Web Service Development Options	12
Developing Web Services with WebLogic Workshop	13
Developing Web Services Using WebLogic Workshop	14
Visually Designing Web Services	15
Editing Web Service Source Code	17
Testing and Debugging Web Services	18
Deploying and Managing Web Services Using WebLogic Workshop	19
Building Enterprise Portals	19
What Is a Portal?	20

WebLogic Portal Features	22
WebLogic Portal Tools.....	24
Portal Management Tool.....	25
E-Business Control Center	26
Integrating the Enterprise	29
Integrating Applications	30
Managing Business Processes	32
Integrating B2B Collaborations.....	33
Translating and Transforming Data.....	35
Managing the Enterprise.....	38
Managing Application Server Resources	39
Configuring Domains	40
Managing Additional Features	40
Managing Additional WebLogic Portal Features.....	41
Managing Additional WebLogic Integration Features.....	43
Building a WebLogic Platform Solution	43
WebLogic Platform Feature Interoperation.....	44
Building an Online Sales Portal	46
Building a Product Inventory Portal.....	48
Building an Employee HR Portal	49
Extending the Platform	50
Conclusion	52

Index

About This Document

This document introduces BEA WebLogic Platform, the unified, simplified, and extensible platform solution. This document is organized as follows:

- “Introducing BEA WebLogic Platform” introduces the main features and benefits of WebLogic Platform, and outlines the enterprise business challenges that it confronts.
- “Developing J2EE Applications” describes the BEA WebLogic Platform development solution and highlights the features of BEA WebLogic Server that are essential to developing and deploying mission-critical J2EE applications, including enterprise-class Web services.
- “Developing Web Services with WebLogic Workshop” describes the BEA WebLogic Workshop development framework that empowers application developers, as well as J2EE experts, to rapidly develop, test, and deploy enterprise-class Web Service applications.
- “Building Enterprise Portals” describes the unified framework for building enterprise portals using BEA WebLogic Portal.
- “Integrating the Enterprise” details the solution for integrating applications and business processes from within and across the enterprise using BEA WebLogic Integration.
- “Managing the Enterprise” describes the integrated operations, administration, and management tools for managing the application development and run-time environments.
- “Building a WebLogic Platform Solution” describes several examples that use WebLogic Platform to build enterprise business solutions, and explain how the WebLogic Platform features interoperate.
- “Extending the Platform” describes the options that are available to customers interested in extending the standards-based WebLogic Platform.

What You Need to Know

This document is intended for new or existing BEA customers that want to learn about the features and benefits of BEA WebLogic Platform.

e-docs Web Site

BEA product documentation is available on the BEA corporate Web site. From the BEA Home page, click on Product Documentation or go directly to the Product Documentation page at the following URL:

<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 WebLogic Platform 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 Platform 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 the following URL:

<http://www.adobe.com/>

Related Information

The following documents contain information describing features and components of WebLogic Platform.

- *Introduction to BEA WebLogic Server*
- *Getting Started with BEA WebLogic Workshop*
- *BEA WebLogic Portal Development Guide*
- *Introducing BEA WebLogic Integration*
- *Introduction to WebLogic Platform 7.0 Security*

Contact Us!

Your feedback on the WebLogic Platform 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 Platform documentation.

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

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

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

- Your name, e-mail address, phone number, and fax number
- Your company name and company address
- Your machine type and authorization codes

-
- The name and version of the product you are using
 - A description of the problem and the content of pertinent error messages

Documentation Conventions

The following documentation conventions are used throughout this document.

Convention	Item
boldface text	Indicates terms defined in the glossary.
Ctrl+Tab	Indicates that you must press two or more keys simultaneously.
<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> <pre>#include <iostream.h> void main () the pointer psz chmod u+w * \tux\data\ap .doc tux.doc BITMAP float</pre>
monospace boldface text	Identifies significant words in code. <i>Example:</i> <pre>void commit ()</pre>
<i>monospace italic text</i>	Identifies variables in code. <i>Example:</i> <pre>String <i>expr</i></pre>

Convention	Item
UPPERCASE TEXT	Indicates device names, environment variables, and logical operators. <i>Examples:</i> LPT1 SIGNON OR
{ }	Indicates a set of choices in a syntax line. The braces themselves should never be typed.
[]	Indicates optional items in a syntax line. The brackets themselves should never be typed. <i>Example:</i> buildobjclient [-v] [-o name] [-f file-list]... [-l file-list]...
	Separates mutually exclusive choices in a syntax line. The symbol itself should never be typed.
...	Indicates one of the following in a command line: <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 The ellipsis itself should never be typed. <i>Example:</i> buildobjclient [-v] [-o name] [-f file-list]... [-l file-list]...
.	Indicates the omission of items from a code example or from a syntax line. The vertical ellipsis itself should never be typed.

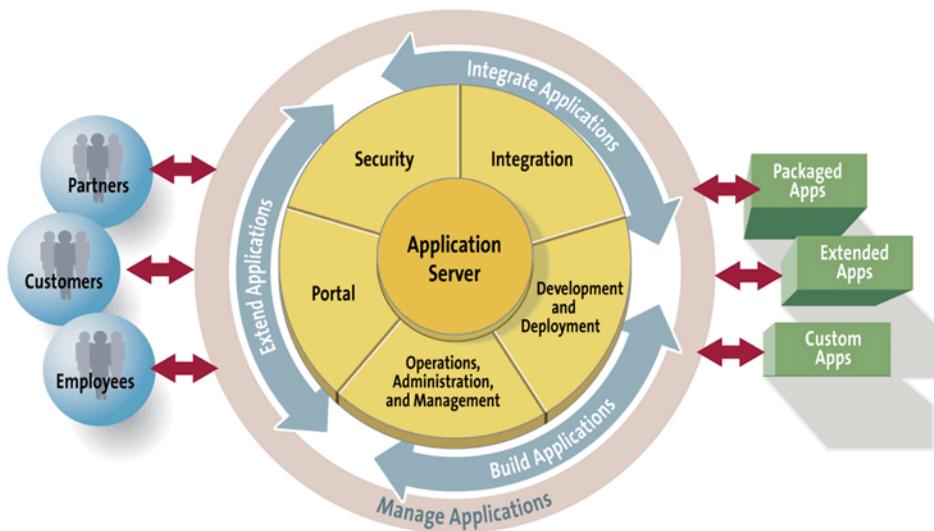


Introducing BEA WebLogic Platform

BEA WebLogic Platform represents the integration of the existing BEA WebLogic product suite into a unified, simplified, and extensible platform. WebLogic Platform merges all of the features of the existing BEA WebLogic products—application server, development, portal, and integration—into a highly integrated solution, combining the benefits of a common application infrastructure with an easy-to-use, robust framework.

The following figure shows the WebLogic Platform common application infrastructure.

Figure 1. BEA WebLogic Platform



WebLogic Platform Features

The following features comprise WebLogic Platform, as shown in Figure 1.

Table 1. WebLogic Platform Features

Feature	Description
Application Server	Solid foundation, via the proven BEA WebLogic Server, for rapidly developing, deploying, and managing e-business applications, including a common application run-time environment.
Development and Deployment	Integrated development, deployment, and run-time environment that includes: <ul style="list-style-type: none">■ Rich, flexible framework for developing and deploying J2EE applications, including enterprise-class Web services, through WebLogic Server.■ WebLogic Workshop, a graphical development framework that empowers application developers, as well as J2EE experts, to build powerful Web services quickly.■ Common application run-time environment through WebLogic Server.
Portal	Unified framework for building enterprise portals using BEA WebLogic Portal that includes: <ul style="list-style-type: none">■ Set of services and tools that enable you to efficiently build, launch, and maintain high-performance enterprise portals.■ Framework that facilitates the creation, customization, and management of multiple enterprise portals and portlets with rich user interfaces.■ Ability to adapt the portal content to the browsing behavior and characteristics of the audience based on a set of predefined rules.■ Portal administration tools for accessing WebLogic Portal functionality and resources, and reducing the burden on IT.■ Standards-based approach to reduce portal integration costs and leverage Web services for application integration.■ Common application run-time environment through WebLogic Server.

Table 1. WebLogic Platform Features (Continued)

Feature	Description
Integration	<p>Complete solution for developing, deploying, and integrating applications and business processes from within and across the enterprise using BEA WebLogic Integration that includes:</p> <ul style="list-style-type: none"> ■ Integration of enterprise information systems via standards-based integration technology, including enterprise resource planning (ERP), supply chain management (SCM), human resource (HR), and customer relationship management (CRM), as well as custom and legacy applications. ■ Collaboration between suppliers and partners, and the automation of business process workflows. ■ Common application run-time environment through WebLogic Server.
Security	Common security framework for building secure applications.
OA&M	Common operations, administration, and management (OA&M) framework for managing the application development, deployment, and run-time environments.

WebLogic Platform Benefits

Through its common application infrastructure, WebLogic Platform delivers the following benefits:

- Common models for packaging, download, installation, licensing, documentation, support, and release schedules
- Shared set of services and enterprise functionality
- Coexistence and interoperability across the platform
- Common application server run-time environment through the proven WebLogic Server
- Common OA&M and security frameworks
- Universal configuration wizard for generating application domains
- Seamless out-of-the-box experience, including access to the WebLogic Platform sample and QuickStart application for getting started using WebLogic Platform

Enterprise Challenges Confronted

The *unified*, *simplified*, and *extensible* architecture of WebLogic Platform confronts the business challenges facing your enterprise, as highlighted in the following table.

Table 2. Enterprise Challenges Confronted

Feature	Enterprise Business Challenge Confronted
<i>Unified architecture</i> —combining development, portal, and integration features	Customers are faced with installing, learning, integrating, and supporting a high volume of diverse applications.
<i>Simplified development, deployment, and management</i> —empowering developers to realize business objectives	Complexity of development, deployment, and management is compounded by distributed, heterogeneous technologies and platforms.
<i>Extensible architecture</i> —supporting a robust, scalable platform	Dependency on multiple, proprietary systems impedes integration and growth opportunities.

Summary

This section has introduced the main features and benefits of WebLogic Platform, and outlined the enterprise business challenges that it confronts.

The following sections explore WebLogic Platform features in more detail, and describe how the common application infrastructure enables you to *build*, *integrate*, and *extend* business processes across a collection of disparate systems.

Use the topics outlined in the following table to access the information most relevant to your business objectives.

Table 3. WebLogic Platform Topics

To learn how to . . .	Refer to the following topic . . .
Develop and deploy mission-critical J2EE applications	“Developing J2EE Applications” on page 6 describes the WebLogic Platform development solution and highlights the features of WebLogic Server that are essential to developing and deploying mission-critical J2EE applications, including enterprise-class Web services.

Table 3. WebLogic Platform Topics (Continued)

To learn how to . . .	Refer to the following topic . . .
Develop enterprise-class Web services to integrate distributed resources	“Developing Enterprise-Class Web Services” on page 9 describes how to develop enterprise-class Web services, which provide a standard way to integrate distributed systems.
Use WebLogic Workshop development framework to develop Web services	“Developing Web Services with WebLogic Workshop” on page 13 describes the WebLogic Workshop development framework that empowers application developers, as well as J2EE experts, to rapidly develop, test, and deploy enterprise-class Web service applications.
Create integrated, personalized portals for employees, customers, and business partners	“Building Enterprise Portals” on page 19 describes the unified framework for building enterprise portals using WebLogic Portal.
Develop, deploy, and integrate applications and business processes from within and across the enterprise	“Integrating the Enterprise” on page 29 details the solution for integrating applications and business processes from within and across the enterprise using WebLogic Integration.
Manage platform security features	Introduction to WebLogic Platform 7.0 Security details security assumptions and concepts used in WebLogic Platform 7.0. Introduction to WebLogic Server Security describes in more detail the architecture of the WebLogic Server security framework.
Manage the application development and run-time environments	“Managing the Enterprise” on page 38 describes the integrated management tools for managing the application development, deployment, and run-time environments.
Build a complete WebLogic Platform solution to meet business objectives	“Building a WebLogic Platform Solution” on page 43 provides examples that demonstrate how to use WebLogic Platform to build an enterprise business solution, and describes WebLogic Platform feature interoperation.
Extend the platform	“Extending the Platform” on page 50 describes the options that are available to customers interested in extending the standards-based WebLogic Platform.

Developing J2EE Applications

WebLogic Platform increases developer productivity by providing solutions that address the needs of its users. For example:

- *Enterprise developers*—WebLogic Server delivers a rich, flexible framework to promote the rapid development and deployment of J2EE applications, including enterprise-class Web services.
- *Application developers*—WebLogic Workshop empowers application developers, as well as J2EE experts, to rapidly develop, test, and deploy enterprise-class Web service applications.

The following sections describe the features of WebLogic Server that promote the rapid development of J2EE applications, including enterprise-class Web services. WebLogic Workshop is described in “Developing Web Services with WebLogic Workshop” on page 13.

Employing the Power and Flexibility of WebLogic Server

To promote the rapid development of your J2EE applications, you need a solid, reliable foundation. BEA WebLogic Server provides that foundation, serving as the backbone for WebLogic Platform.

WebLogic Server delivers the features that are essential for developing and deploying mission-critical J2EE applications. Furthermore, WebLogic Server is *proven*: it is used by more developers, independent software vendors (ISVs), application service providers (ASPs), and system integrators (SIs) than any other application server.

The following table highlights the features of WebLogic Server that promote the rapid development of J2EE applications.

Table 4. WebLogic Server Feature Highlights

Feature	Description
<i>Leverages J2EE component technologies and services</i>	<p>WebLogic Server supports the Sun Microsystems Java 2 Platform, Enterprise Edition (J2EE) version 1.3 component technologies and services, representing the first independently developed Java application server to achieve J2EE certification. J2EE includes specifications for:</p> <ul style="list-style-type: none"> ■ Java Server Pages (JSPs), Java Servlets, and Enterprise Java Beans (EJBs)—develop scalable, reusable, component-based applications. ■ Web applications—deploy a collection of Web resources such as JSPs, servlets, and HTML pages. ■ J2EE Connector Architecture (CA)—integrate Enterprise Information System (EIS) through resource adapters, also referred to as connectors. ■ Java Database Connectivity (JDBC)—enable standardized database access. ■ Java Message Service (JMS)—communicate between applications through the asynchronous exchange of messages, and simplify application development by providing a standard interface. ■ Java Naming and Directory Interface (JNDI)—access named resource and file systems. ■ Managing clustering, security, transactions, and other core services. For more information, see “Managing the Enterprise” on page 38. <p>J2EE simplifies the development of multitier enterprise applications by managing many of the application lifecycle details and enabling the application developer to focus on the business requirements, rather than managing the infrastructure. The J2EE component-based model enables you to develop a core set of business applications that are both portable and scalable, and contributes to increased productivity and faster time-to-market.</p> <p>For more information on J2EE component technologies and services, refer to the Sun Web site at the following URL:</p> <p>http://java.sun.com/j2ee/sdk_1.3/index.html</p>
<i>Enables rapid development and deployment of enterprise-class Web services</i>	<p>WebLogic Server enables you to rapidly develop and deploy enterprise-class Web services, providing a standard way to integrate distributed systems. Web services are shared applications that distributed clients and servers access to exchange information over the Internet.</p> <p>Web service development is explored in more detail in the section “Developing Enterprise-Class Web Services” on page 9.</p>

Table 4. WebLogic Server Feature Highlights (Continued)

Feature	Description
<i>Supports exchange of enterprise messages</i>	<p>A <i>message</i> is a request, report, and/or event that contains the information needed to coordinate communication between different applications. Messages provide a level of abstraction, allowing you to separate details about the destination system from the application code.</p> <p>WebLogic Server delivers a powerful messaging platform, based on Java Messaging Service (JMS), to enable distributed, loosely coupled communication between applications. Specifically, JMS:</p> <ul style="list-style-type: none">■ Enables Java applications sharing a messaging system to exchange messages.■ Simplifies application development by providing a standard interface for creating, sending, and receiving messages. <p>For more information about WebLogic JMS features, see <i>Programming WebLogic JMS</i> on the BEA e-docs Web site at the following URL:</p> <p>http://e-docs.bea.com/wls/docs70/jms/index.html</p>
<i>Supports integration of new and legacy Enterprise Information System (EIS) applications</i>	<p>WebLogic Server supports the integration of new and legacy EIS applications through the following features:</p> <ul style="list-style-type: none">■ Web services—access and re-use shared applications to exchange information over the Internet.■ J2EE Connector Architecture (CA)—integrate Enterprise Information System (EIS) through resource adapters.■ BEA WebLogic/Tuxedo Connector (WTC)—implement bidirectional access between WebLogic Server and Tuxedo to develop, manage, and deploy C/C++ or Cobol applications based on procedural, Common Object Request Broker Architecture (CORBA), and ATMI programming models.■ WebLogic jCOM—implement bidirectional access between Java/J2EE objects deployed in WebLogic Server, and Microsoft ActiveX components available within the Microsoft Office family of products, VisualBasic and C++ objects, and other COM/DCOM-compliant environments.

Table 4. WebLogic Server Feature Highlights (Continued)

Feature	Description
<i>Simplifies application deployment</i>	<p>WebLogic Server deployment is a simple two-phase process:</p> <ol style="list-style-type: none"> 1. Package the J2EE application components. 2. Deploy the J2EE application component package files on WebLogic Server. <p>WebLogic Server delivers the following tools to assist in the application deployment process:</p> <ul style="list-style-type: none"> ■ BEA WebLogic Builder—graphical tool for assembling a J2EE application module, creating and editing its deployment descriptors, and deploying it to a WebLogic Server. ■ BEA WebLogic EJBGen—command-line tool that uses Javadoc markup to generate EJB deployment descriptor files and home, local, and remote interfaces automatically. ■ BEA WebLogic Administration Console Deployment Descriptor Editor—graphical tool for editing deployment descriptor files. ■ BEA XML Editor—graphical tool for editing XML files. <p>For more information about application deployment, see <i>Deployment</i> in the WebLogic Server documentation on the BEA e-docs Web site, at the following URL:</p> <p>http://e-docs.bea.com/wls/docs70/deployment.html</p>

For more information about WebLogic Server, see the WebLogic Server documentation on the BEA e-docs Web site at the following URL:

<http://e-docs.bea.com/wls/docs70/index.html>

The following sections explore the development of enterprise-class Web services in more detail.

Developing Enterprise-Class Web Services

Web services provide a standard means of communication between disparate applications within the enterprise and over the Internet. WebLogic Platform provides you with the tools and infrastructure you need to rapidly develop and deploy enterprise-class Web services that integrate distributed systems and provide you with the reliability, scalability, and availability you have come to expect from WebLogic.

What Are Web Services?

Web services make software application resources available to distributed clients over a network in a standardized way. Because Web services rely on standard methods of communication, applications that support these standards can take advantage of the resources that Web services expose, regardless of the internal implementation of the client application or Web service.

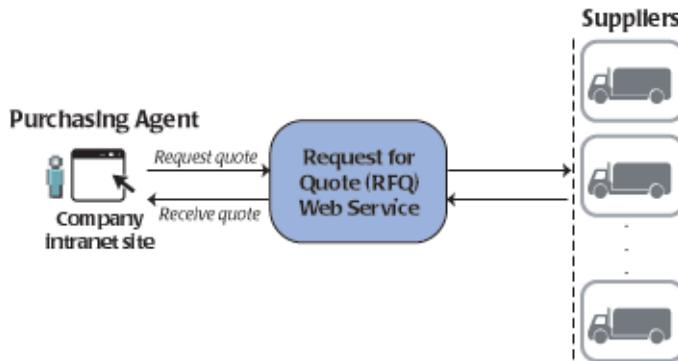
Web services are ideal for the enterprise because they are:

- *Loosely coupled.* Web services communicate using a *public contract* (Web Services Definition Language, or WSDL, file) that describes the XML messages to be exchanged, but leaves the underlying implementation details to each application. Loosely-coupled Web services enable communication between applications written in different languages and running on different operating systems.
- *Coarse grained.* Web services integrate at the business level, exchanging documents like POs and invoices rather than single data elements. Coarse-grained Web services allow greater flexibility when one part of the system changes.
- *Asynchronous.* Web services can communicate asynchronously, as well as synchronously, so that they operate the way your business does. Asynchronous Web services can perform operations that may require resources that are not immediately available, without blocking other requests or slowing down the system.

Web services are used to perform discrete business processes, such as accessing a product inventory, checking stock quotes, or exchanging user profile information.

For example, the following figure shows how a purchasing agent can access a company intranet site to check inventory and submit a request for quotes (RFQ)—using an asynchronous Web service—to various suppliers on low-inventory items.

Figure 2. Web Service Example



Each supplier receives an RFQ request and returns price and availability information. All communication takes place asynchronously, and information is posted to the intranet site as it is returned from suppliers. Based on the information returned, the purchasing agent reviews the quotes, selects a supplier, and submits a purchase order.

Web Service Standards

To expose their resources and ensure interoperability among applications that exist on heterogeneous hardware and software platforms, Web services adhere to the following communication standards:

- *Describe their functionality through a Web Services Definition Language (WSDL) file, which may be published through a public registry.*

A WSDL file describes how the Web service operates and how other software applications can interface with it. You can think of a WSDL file as the instruction manual for a Web service, explaining how a user can access the resources provided by the Web service. WSDLs are publicly accessible, providing enough detail to enable potential clients to operate the Web service.

WebLogic Server delivers:

- Built-in support for WSDL.
- Automatic generation of WSDL scripts when a Web service is deployed.
- An embedded Universal Description, Discovery, and Integration (UDDI) registry, providing a distributed, Web-based location registry for searching Web services.

-
- An API for searching and updating the UDDI or any third-party registry.
 - *Communicate with other applications via XML messages.*

XML messages provide a common language that different applications can use to communicate over a network. To use a Web service, an application sends an XML message containing a request to perform an operation. The Web service responds by sending an XML message containing the results of that operation.

Typically, XML messages are formatted according to Simple Object Access Protocol (SOAP) syntax. SOAP specifies a standard format for applications to call other application methods and pass data. WebLogic Server delivers a high-performance SOAP implementation. Non-SOAP forms of XML messages are also possible, depending on the specific requirements of the Web service.

The XML message type and syntax requirements are defined in the WSDL file, making the Web service available to any client application capable of sending and receiving XML messages that meet those requirements.

- *Receive requests and send responses using standard network protocols, such as HTTP (HyperText Transfer Protocol) and JMS (Java Message Service).*

WebLogic Server supports the standard network protocols used during Web service communication.

Web Service Development Options

BEA supports two approaches to developing enterprise-class Web services:

- Use BEA WebLogic Workshop to develop, test, and deploy enterprise-class Web service applications. WebLogic Workshop is a graphical development framework that empowers application developers, as well as J2EE experts, to build powerful Web services quickly. The next section explores this approach in more detail.
- Use your favorite IDE or text editor to develop Web services, and leverage the Ant tasks provided by WebLogic Server that help you generate Web service components (such as data type serializers and deserializers, a client JAR file, and a Web service Home page) and to package all the pieces of a WebLogic Web service into a deployable EAR file. For more information, see *Programming WebLogic Web Services* in the WebLogic Server documentation on the BEA e-docs Web site at the following URL:

<http://e-docs.bea.com/wls/docs70/webserv/index.html>

Developing Web Services with WebLogic Workshop

BEA WebLogic Workshop is a graphical development environment that empowers application developers, as well as J2EE experts, to rapidly develop, test, and deploy enterprise-class Web service applications.

WebLogic Workshop enables developers to visually design robust Web services, without needing to understand the complexities of J2EE and object-oriented programming. Rather than learn complex APIs, developers can focus on the application logic, setting simple properties (expressed as Javadoc annotations) to access sophisticated functionality in the framework.

WebLogic Workshop provides all of the features commonly found in a development environment, including code editing, debugging, and project management. Tightly integrated build, deploy, and test phases are managed transparently to provide fast iterative development. Web services built using WebLogic Workshop are deployed automatically on WebLogic Server, streamlining deployment and providing consistency of service across the enterprise.

WebLogic Workshop consists of the following components:

- *Visual development environment*—WebLogic Workshop delivers an intuitive GUI-based integrated development environment (IDE) for developing Web services, supporting the development, test, and debug phases as a single, consistent process.
- *Run-time framework*—WebLogic Workshop delivers a run-time environment for deploying and managing Web services created with WebLogic Workshop.
- *Format for Web service applications*—Web services generated using WebLogic Workshop are stored as Java Web Services (JWS) files. A JWS file contains the Java code which determines how the Web service behaves, and consists of standard Java class definitions with annotations, expressed using Javadoc syntax, to define additional functionality. You can think of a JWS file as a Java class that communicates with the outside world through XML messages.

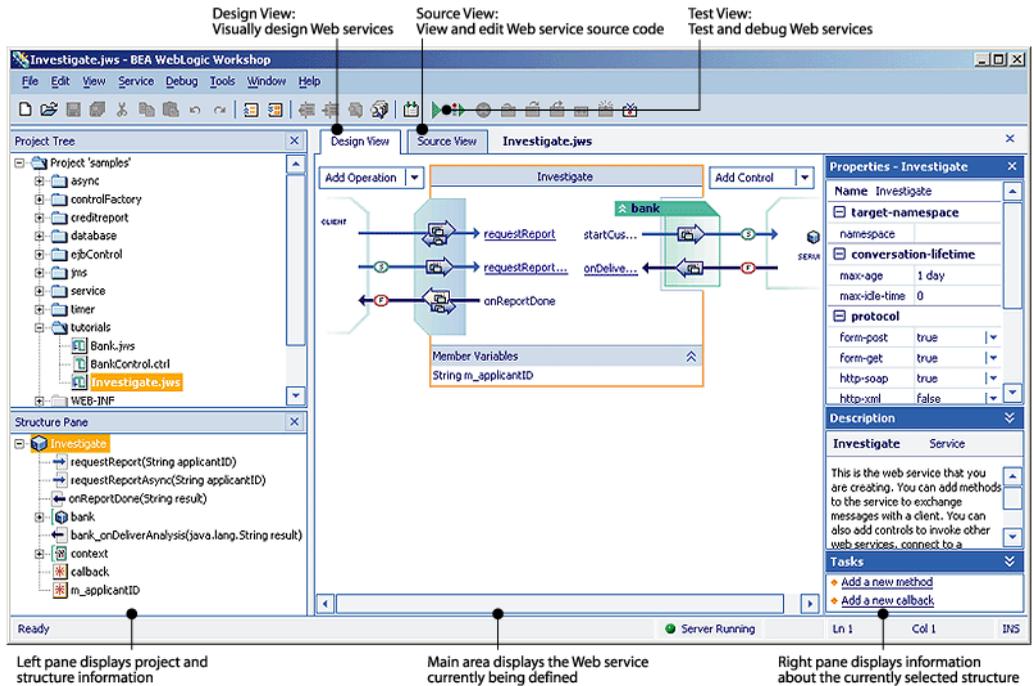
The WebLogic Workshop visual development and run-time environments are described in the following sections. For more information about WebLogic Workshop, see the WebLogic Workshop documentation on the BEA e-docs Web site at the following URL:

<http://e-docs.bea.com/workshop/docs70/index.html>

Developing Web Services Using WebLogic Workshop

The WebLogic Workshop IDE, shown in the following figure, provides a powerful development environment for building Web services.

Figure 3. Developing Web Services—WebLogic Workshop IDE



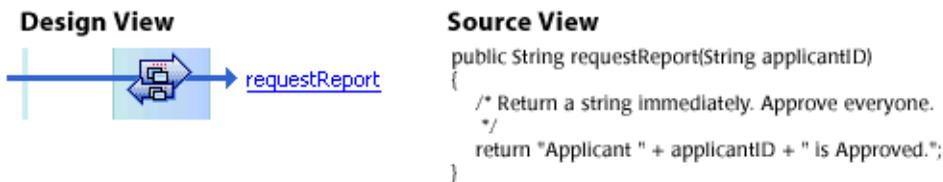
The main area of the WebLogic Workshop IDE displays the Web service that is currently being defined, and the surrounding panes display information that is related to the current Web service.

As shown in the previous figure, WebLogic Workshop provides access to the following three views of your JWS file:

- Design View—visually design Web services
- Source View—view and edit Web service source code
- Test View—test and debug Web services

WebLogic Workshop supports two-way editing, ensuring that updates that are made in the Design View are automatically reflected in the Source View, and vice versa. For example, when you add the `requestReport` method in the Design View, the corresponding source code is generated automatically, and can be viewed from the Source View, as shown in the following figure.

Figure 4. WebLogic Workshop Two-Way Editing

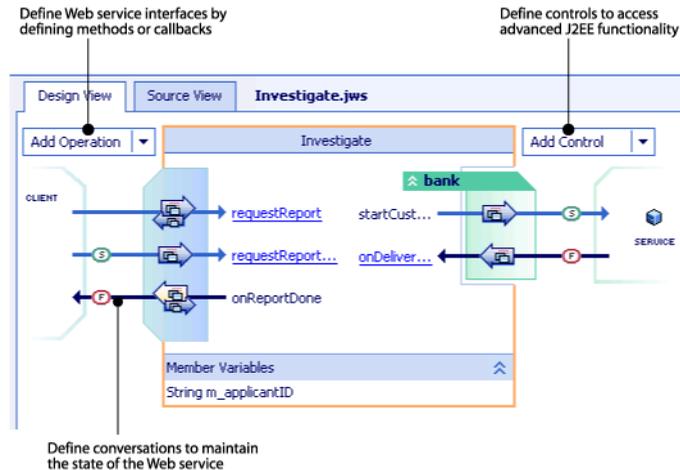


The following sections describe the design, source, and test views in more detail.

Visually Designing Web Services

The Design View, shown in the following figure, enables you to visually design Web services and their interactions with client applications and data resources, such as databases and other Web services.

Figure 5. Visually Designing Web Services—Design View



While designing a Web service, you can define:

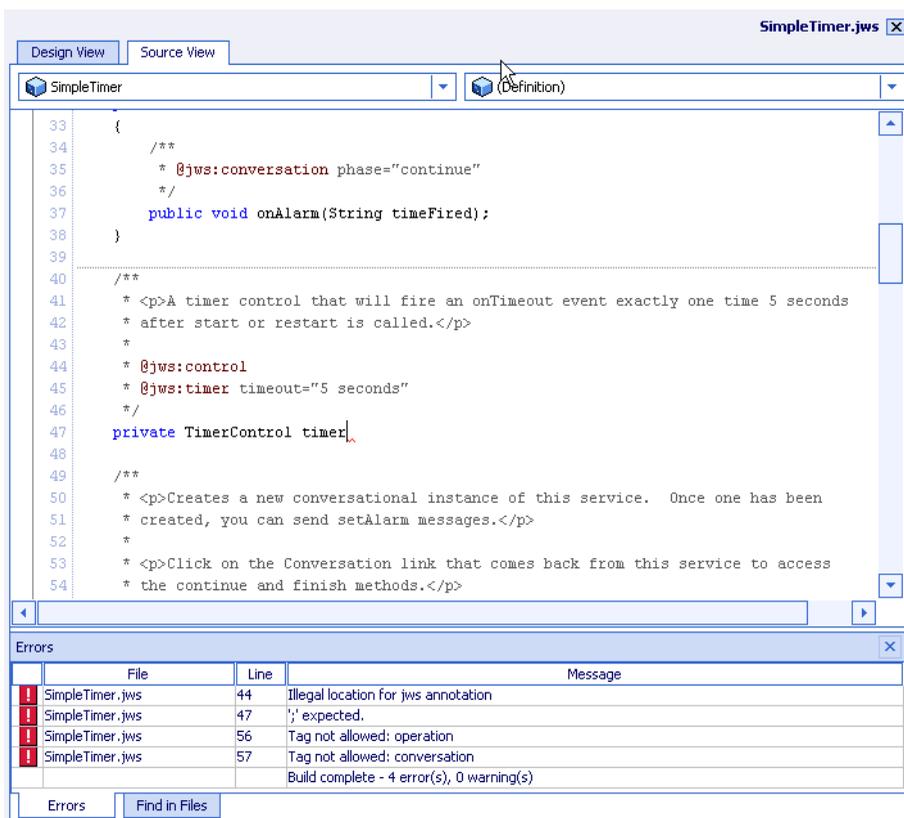
- Web service interfaces, including *methods* to expose the functionality of the Web service, and *callbacks* to return results to clients asynchronously.
- Controls to:
 - Access advanced J2EE functionality, including EJBs, databases, JMS queues.
 - Execute an operation at specific time intervals.
 - Access other Web services.
- Properties (expressed as Javadoc annotations) to access sophisticated functionality in the framework. See the Properties Pane in Figure 3. Properties include:
 - Conversations to maintain the state of your Web service over a long-running sequence of asynchronous calls. The icons shown in Figure 5 indicate the start and finish points of a conversation.
 - Message buffers to store requests and responses on a nonvolatile queue to ensure that they are processed.
 - Protocols supported by the interface.

- XML maps to control how data is converted between XML messages and Java objects. The Map and Interface Editor makes it easy to define parameter and return XML maps, and build the interface to your application.

Editing Web Service Source Code

The Source View, shown in the following figure, enables you to edit Web service source code directly to customize generated code and implement the application logic.

Figure 6. Editing Web Service Source Code—Source View



The Source View supports productivity features such as auto-complete, parameter hints, and as-you-type error highlighting. When you compile a file with syntax errors, the Error pane displays error messages at the bottom of the Source View. Syntax errors are also marked by red underlines in your code. Double-clicking on a particular message moves the cursor next to the offending code.

Testing and Debugging Web Services

The Test View, shown in the following figure, enables you to test Web services. The Test View consists of a dynamically generated HTML page that enables you to call service methods with specific parameter values.

Figure 7. Testing and Debugging Web Services—Test View

The screenshot shows the 'Investigate.jws Web Service' interface. At the top, there are tabs for 'Overview', 'Console', 'Test Form', and 'Test XML'. The URL is 'http://localhost:7001/ FinancialServices/ Investigate.jws'. Below the tabs are links for 'Start operations' and 'Continue this conversation'. On the left, a 'Message Log' shows a log entry for 'requestReportAsync' with ID '1014089615040'. The main area displays the details of the service request, processing request, and buffered service response.

```
Investigate.jws Web Service Created by BEA WebLogic Workshop  
Overview Console Test Form Test XML http://localhost:7001/ FinancialServices/ Investigate.jws  
Start operations  
Continue this conversation  
Message Log Refresh  
1014089615040  
▶ requestReportAsync  
Clear Log  
Service Request  
Submitted at Mon Feb 18 19:33:35 PST 2002  
requestReportAsync  
.EXPLORE = .INVOKE  
.EXPLOREOVERRIDE = .TEST  
applicantID = 123-45-6789  
.CONVERSATIONID = 1014089615040  
Processing Request  
Submitted at Mon Feb 18 19:33:39 PST 2002  
requestReportAsync  
Buffered Service Response  
Submitted at Mon Feb 18 19:33:39 PST 2002  
requestReportAsync  
<requestReportAsyncHttpGetOutBodyType xmlns="http://openuri.org/">  
</requestReportAsyncHttpGetOutBodyType>
```

The Message Log on the left-hand side of the window displays messages that are sent to and received from the Web service during testing. You can click on a log entry to display the request and response details of the message.

While testing, you can use WebLogic Workshop integrated debugger to debug your Web service. The debugger allows you to set breakpoints, step through your code line-by-line, view local variables, set watches on variables, and view the call stack and exception information.

Deploying and Managing Web Services Using WebLogic Workshop

WebLogic Workshop run-time environment provides the framework to easily deploy and manage Web services created with WebLogic Workshop.

WebLogic Workshop run-time environment reads a JWS file and automatically performs the following tasks:

- Compiles and deploys Web services, generating the required EJBs, messages queues, and other J2EE resources to deploy the application, and creating the deployment descriptors for each EJB.
- Instantiates controls used by the Web service.
- Generates WSDL files for describing the Web service.
- Generates Java client code for invoking the Web service.
- Generates an EAR file containing any collection of Web services for standard deployment to a production server.

The following section explores how to build enterprise portals, which can interact with Web services to integrate distributed systems.

Building Enterprise Portals

BEA WebLogic Portal delivers a unified framework for building enterprise portals that:

- Enables you to build, launch, and maintain portals that are interactive, adaptable to the needs of the audience, and customizable based on visitor preferences.

-
- Facilitates the creation, customization, and administration of multiple portals.
 - Simplifies, personalizes, and reduces the cost of self-service access to information, applications, and business processes.
 - Increases IT productivity through simplified portal development, deployment, and management by leveraging a common infrastructure and enabling component reuse.
 - Runs in the common application run-time environment, provided by WebLogic Server.

The following sections define what a portal is, and explore the WebLogic Portal features and design tools.

For more information about WebLogic Portal, see the WebLogic Portal documentation on the BEA e-docs Web site at the following URL:

<http://e-docs.bea.com/wlp/docs70/index.html>

What Is a Portal?

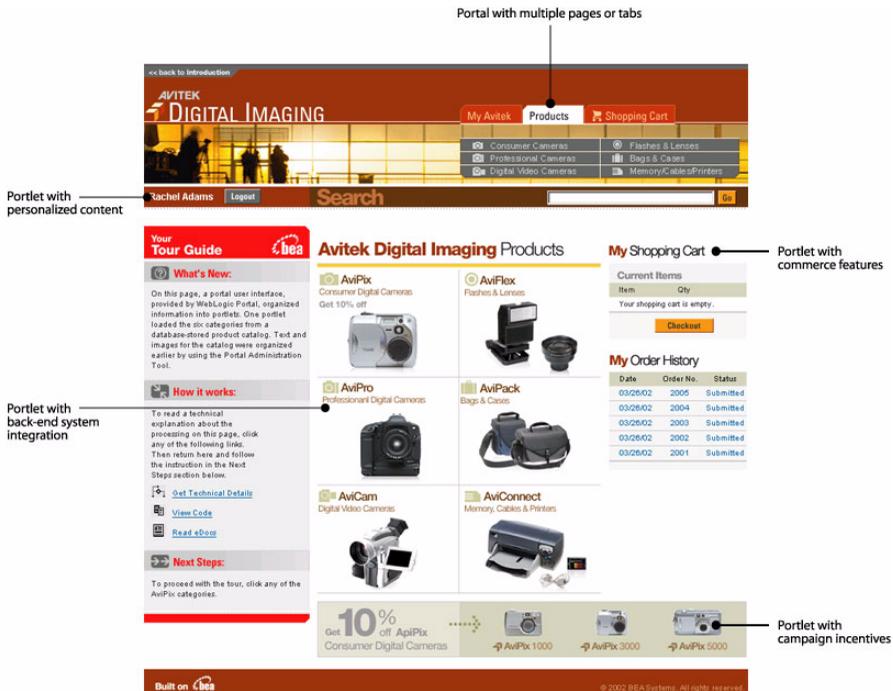
A *portal* provides a single point of access to enterprise data and applications, presenting a unified and personalized view of that information to employees, customers, and business partners.

There are three main types of portals:

- Business-to-business (B2B) for exchanging information between business partners.
- Business-to-consumer (B2C) for tailoring content, applications, and services to different types of consumers.
- Business-to-employee (B2E) for enhancing member or employee intranet Web sites.

The following figure provides an example of a Business-to-Consumer portal.

Figure 8. Portal Example



As shown in the previous figure, a portal manages and organizes content through the use of portlets. A *portlet* is a standard user interface that displays content or applications within a section of a portal page. Multiple portlets can appear within a portal, and they can display personalized content, and commerce and campaign services, and integrate with back-end systems. These concepts are explored in more detail in the following section.

WebLogic Portal Features

The following table describes the main features of WebLogic Portal.

Table 5. WebLogic Portal Features

Feature	Description
Portal Foundation Services	<p>Provides a framework to create, customize, and manage multiple enterprise portals and portlets with rich user interfaces.</p> <p>Portal foundation services include:</p> <ul style="list-style-type: none">■ Presentation services to enable the creation of portal pages containing JSP-based portlets and customized page layouts. WebLogic Portal provides editors and wizards to facilitate the creation and administration of portals and portlets.■ Layout, skin, and portlet selection to define portal appearance. <i>Layouts</i> define the arrangement of portlets on a portal page. <i>Skins</i> define the overall look-and-feel of the portal. <i>Portlet selection</i> defines which portlets appear on a page.■ System and service integration with third-party content management systems, LDAP servers, and search engines.■ Commerce and campaign services including predefined commerce templates for catalog, search/browse, shopping cart, order management, payment service, and discount features, and for providing back-end system integration.

Table 5. WebLogic Portal Features (Continued)

Feature	Description
Personalization and Interaction Management	<p>Adapts the content of a Web site to the browsing behavior and characteristics of the audience based on a set of predefined rules.</p> <p>Personalization can be implicit or explicit: <i>Implicit</i> personalization adapts the site using online browsing behavior. <i>Explicit</i> personalization further adapts the site for known visitors, using online profile and offline visitor database information to provide relevant personalized information.</p> <p>Personalization and interaction management includes:</p> <ul style="list-style-type: none"> ■ Support for defining personalization rules. ■ Events for directing actions such as displaying advertisements, sending promotional e-mails, and enabling discounts. ■ Behavior Tracking to develop trend and perspective analyses and optimize Web site content. ■ Campaign management to plan and manage multiple targeted promotional campaigns or interactions. ■ Placeholders to identify locations on a Web site where promotions will appear. ■ Rules-based customer segments to classify users by similar attributes gathered from user profiles and dynamic data, such as number of visits and value of the products purchased on the site.
Intelligent Administration	<p>Facilitates portal administration by providing portal users with the tools for accessing WebLogic Portal functionality and resources, and reducing the burden on IT.</p> <p>Intelligent administration includes:</p> <ul style="list-style-type: none"> ■ Delegated administration to create an administration hierarchy to delegate and distribute workload. ■ Rules-based entitlements to establish a set of rules that determines which users or groups are able to view specific portal content. Entitlements dynamically define user permissions based on business rules, contributing to a reduction in administration overhead. ■ Webflow to control the navigation between Web and portal pages and separate the content shown to visitors from the underlying business logic.

Table 5. WebLogic Portal Features (Continued)

Feature	Description
Integration Services	<p>Applies standards-based approach to reduce portal integration costs and leverage Web services for application integration.</p> <p>Integration services include:</p> <ul style="list-style-type: none">■ Unified user profiles (UUP) that provide a single view of the user across multiple data sources including user profiles, existing user databases, and legacy applications.■ Pipeline components to facilitate process-level communication and dataflow between Web applications and enterprise information system, including WebLogic Integration. A <i>Pipeline component</i> is a discrete unit of business logic for tasks such as calculating a discount or validating a credit card number.■ Portlets that can be built to integrate with enterprise information systems. For example, Web service portlets discover and interact with Web services, or portlets can call WebLogic Integration APIs.

WebLogic Portal Tools

WebLogic Portal provides the following tools for accomplishing fundamental design and management tasks:

- Portal Management Tool, which is part of the BEA WebLogic Portal Administration Tools

Note: For more information about the WebLogic Portal Administration Tools, see “Managing Additional WebLogic Portal Features” on page 41.

- E-Business Control Center, which is a standalone graphical tool

These tools are described in the following sections.

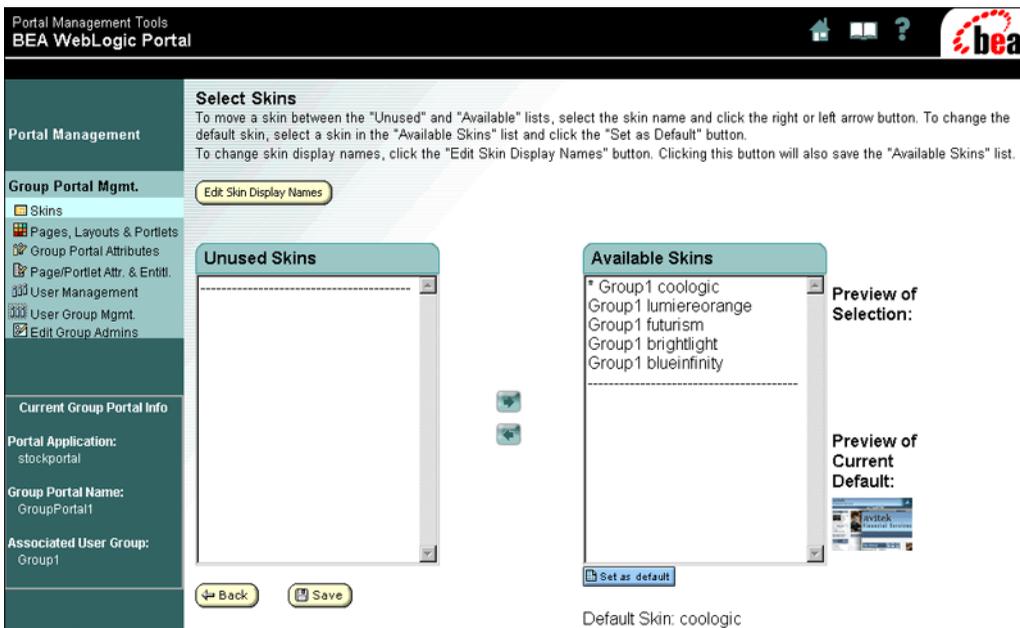
Portal Management Tool

The Portal Management tool is part of the browser-based WebLogic Portal Administration tools, runs on the application server, and is used to accomplish the following tasks:

- Creating and managing the default portal content
- Managing the customization of content for specific group members through the use of group portals

For example, you can define the default skin appearance of a portal for all of the members of a group portal using the Select Skins page that is accessible from the Portal Management tool, as shown in the following figure.

Figure 9. Defining Default Portal Skins—Portal Management Tool

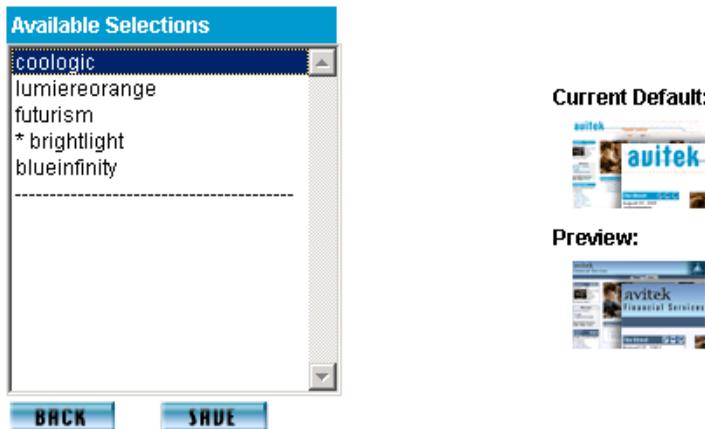


The following figure shows how a Web site visitor can select skins for a sample portal. (Visitor-specific settings do not impact the default portal settings.)

Figure 10. Selecting Skins Using Sample Portal Visitor Tools

Select Skins

To select a default skin, choose a skin name from the "Available Selections" list and click the "Save" button



E-Business Control Center

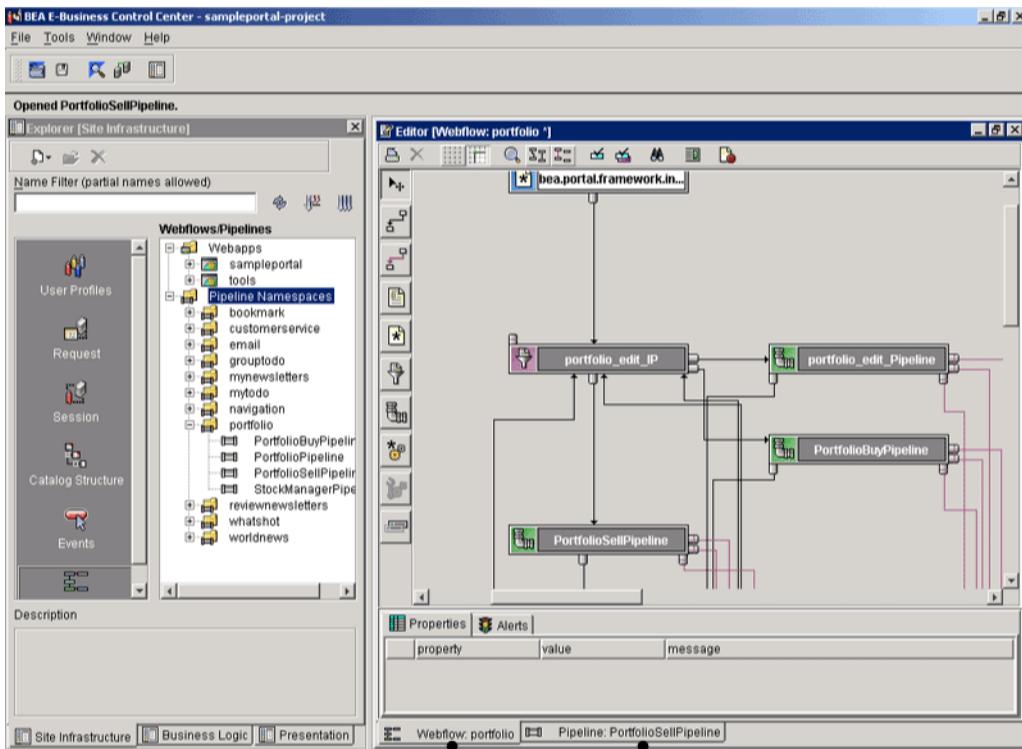
E-Business Control Center is a standalone graphical tool that simplifies the tasks required to create and maintain site infrastructure, business logic, and presentation resources for enterprise portals and sites built using WebLogic Portal. The E-Business Control Center enables you to define the design, functionality, and visitor attributes of portals and portlets with limited coding knowledge of Java, Java Server Pages (JSPs) or HTML.

The E-Business Control Center provides a number of useful editors and wizards for accomplishing fundamental portal design tasks, including the Webflow and Pipeline Editor, Portal wizard, and Portlet wizard.

Webflow and Pipeline Editor

The Webflow and Pipeline Editor, shown in the following figure, enables you to specify navigation between pages and simplifies inter-portlet communication.

Figure 11. Defining Webflow and Pipelines—E-Business Control Center



Webflow Editor: Define Webflow to specify the order in which pages are displayed to your Web site visitors.

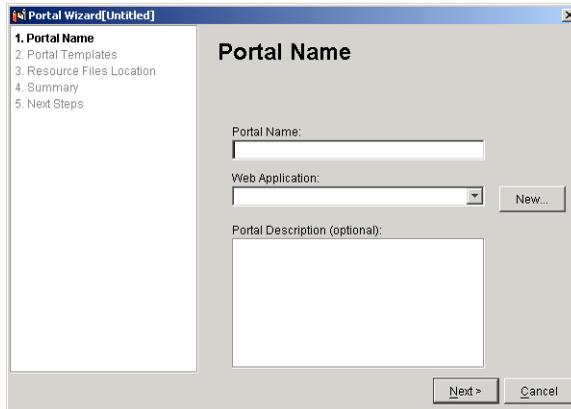
Pipeline Editor: Define Pipeline Components to initiate the execution of specific business process tasks.

Webflows and Pipelines are flexible mechanisms designed to help you separate and manage the presentation and business logic that comprise your applications, without the need for advanced programming skills.

Portal and Portlet Wizards

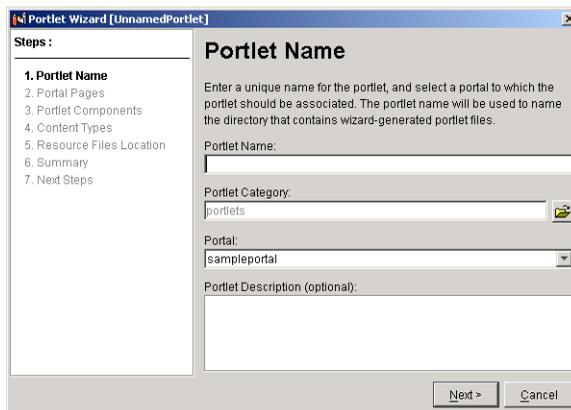
The Portal wizard, shown in the following figure, enables you to easily create a new portal based on an existing template.

Figure 12. Creating a Portal—Portal Wizard



Similarly, the Portlet wizard, shown in the following figure, enables you to easily create a new portlet and specify its components.

Figure 13. Creating a Portlet—Portlet Wizard



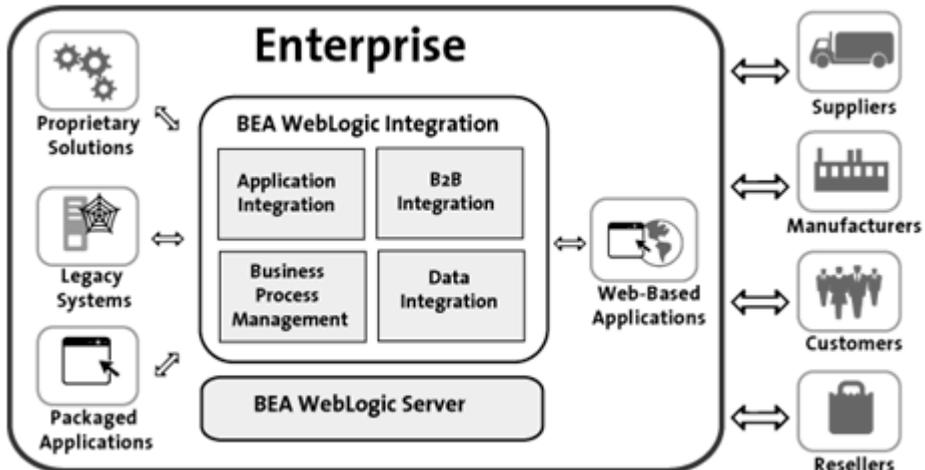
The Portal and Portlet wizards collect the required information and automatically generate a new portal and/or portlet, respectively, based on the information specified.

Integrating the Enterprise

Today's global enterprise requires an integration solution that enables applications to exchange business-critical information in real-time. Integration initiatives may be *intra-enterprise*, linking your back-office systems, or *inter-enterprise*, linking internal data and processes with customers and other corporate enterprises.

BEA WebLogic Integration, shown in the following figure, delivers a complete solution for developing, deploying, and integrating applications and business processes from within and across the enterprise.

Figure 14. BEA WebLogic Integration Framework



As shown in the previous figure, WebLogic Integration supports the following key services.

Table 6. WebLogic Integration Services

Service	Description
Application Integration	Integrate information systems and new applications across the enterprise.
Business Process Management (BPM)	Streamline management of enterprise business processes.
B2B Integration	Collaborate with business trading partners and manage B2B interactions.
Data Integration	Integrate data from diverse enterprise applications.

WebLogic Integration runs in the common application run-time environment, provided by WebLogic Server.

The following sections describe WebLogic Integration services in more detail.

For more information about WebLogic Integration, see the WebLogic Integration documentation on the BEA e-docs Web site at the following URL:

<http://e-docs.bea.com/wli/docs70/index.html>

Integrating Applications

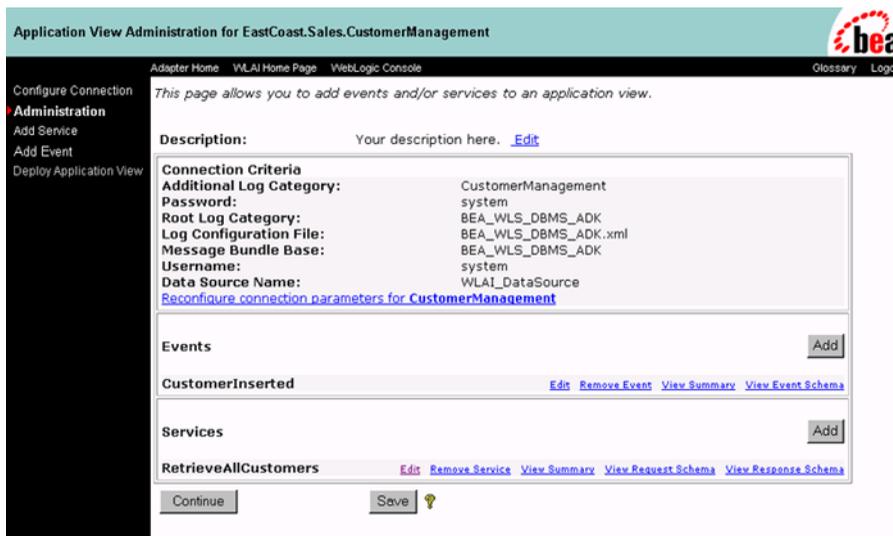
Application integration enables you to integrate information systems and new applications across the enterprise, including enterprise resource planning (ERP), supply chain management (SCM), human resource (HR), and customer relationship management (CRM), as well as custom and legacy applications.

Using adapters, businesses can share data and business processes between connected applications without having to modify their original applications or data structures. You can develop and host J2EE Connector Architecture (CA)-compliant service and event adapters using the Adapter Developer Toolkit (ADK). A *service adapter* invokes a specific function in the enterprise information system to which it is connected. An *event adapter* propagates information from the enterprise information system to the integration environment.

Application views provide an abstract view of an adapter, exposing existing enterprise applications as business services that accept XML data as input and return XML data as output. Application views enable integration specialists to access enterprise data without worrying about the technical implementation.

The Application View Console, shown in the following figure, is a graphical user interface (GUI) that offers an easy way to access, organize, and edit all the application views in your enterprise.

Figure 15. Managing Application Views—Application View Console



Using the Application View Console, you can define and manage application views for event and service adapters. By storing your application views in folders, you can organize them according to your own navigation scheme, regardless of the adapters to which the individual application views belong.

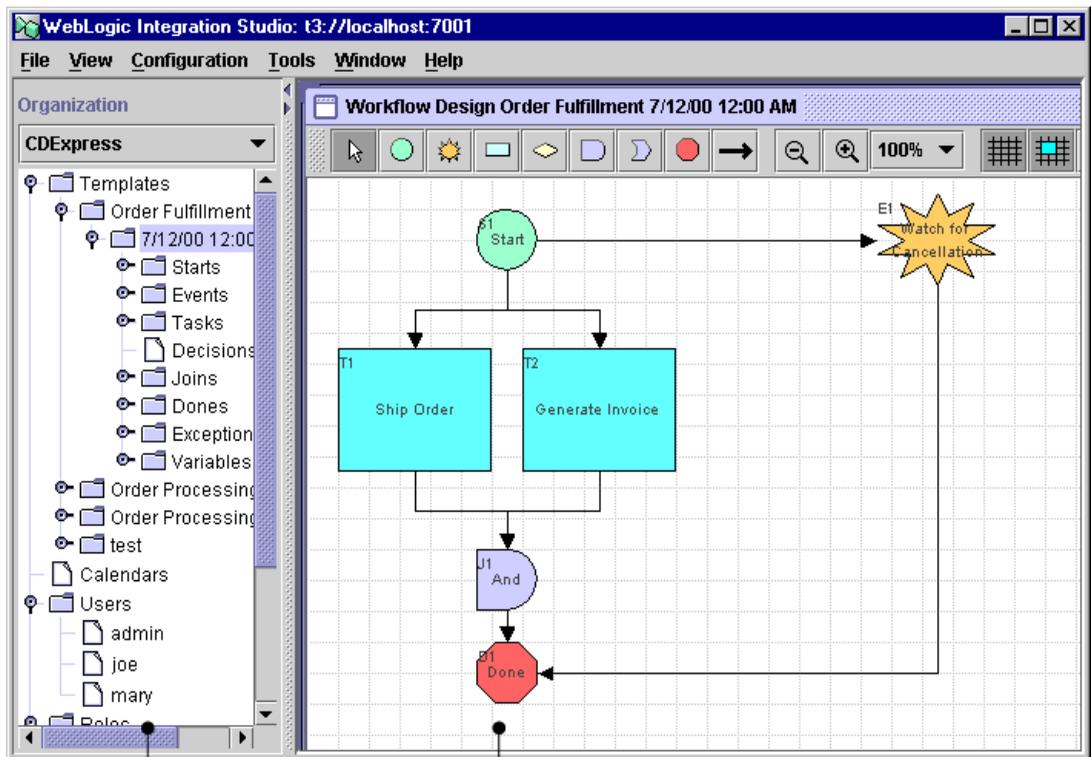
To extend the application integration functionality, you can integrate application views with BPM business processes using the BPM plug-in framework.

Managing Business Processes

Business process management (BPM) enables you to streamline the management of enterprise business processes. A *business process* consolidates company-wide tasks, actions, decisions, and events, and can span applications, systems, and people.

Using WebLogic Integration Studio, you can design and monitor business processes, as shown in the following figure.

Figure 16. Defining and Monitoring Business Processes—Studio



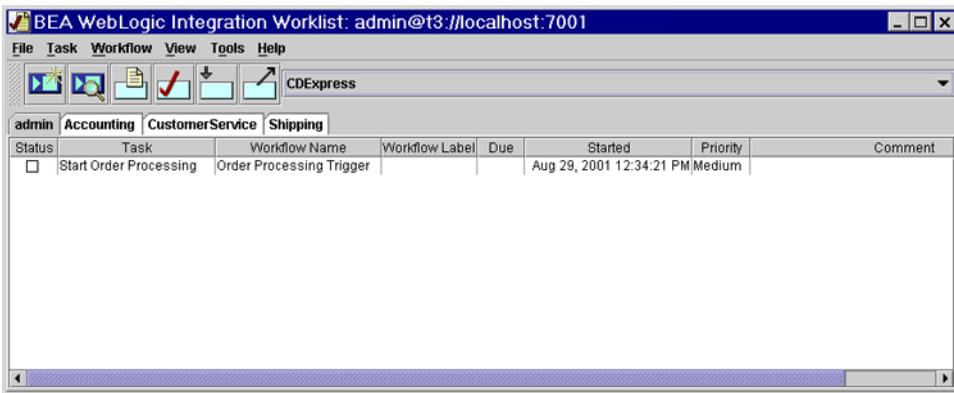
Left pane displays the workflow components for the currently selected organization

Right pane displays the currently selected workflow component

The Studio provides a graphical interface in which familiar flowchart elements are available for business process workflow modeling to graphically represent a business process. The Studio supports workflow design, workflow monitoring, and data administration functions.

During execution, you can interact with business processes as an end user using WebLogic Integration Worklist, as shown in the following figure.

Figure 17. Interacting with Business Processes—Worklist



Using the Worklist, you can manage the current organization, start business processes, and manage tasks.

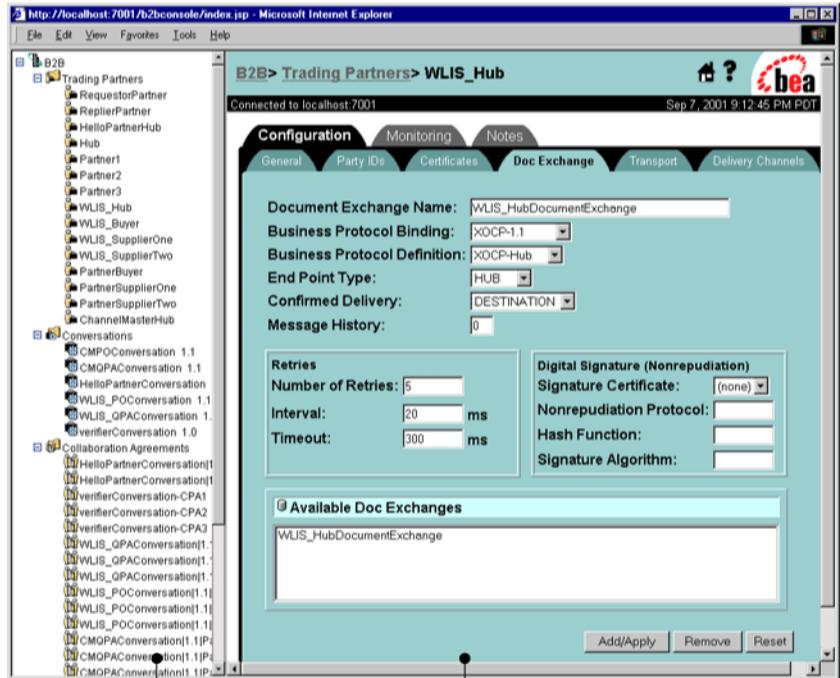
To extend the BPM functionality, you can develop custom client applications and/or plug-ins using the BPM API. WebLogic Integration delivers predefined plug-ins for connecting with application integration, business-to-business (B2B) integration, and data integration functionality.

Integrating B2B Collaborations

B2B integration enables you to collaborate with business trading partners and manage B2B interactions.

Using the WebLogic Integration B2B Console, you can configure and manage these collaborations and B2B interactions, as shown in the following figure.

Figure 18. Configuring and Managing B2B Integration—B2B Console



Left pane displays hierarchically arranged set of nodes that represent the entities (such as trading partners) linked through B2B integration

Right pane displays the interactive pages through which you manage the currently selected node

Using the B2B Console, you can configure and manage:

- Conversations. Within the context of B2B, a *conversation* consists of a series of business messages exchanged between trading partners.
- Trading partners and their respective delivery channel configurations. *Delivery channels* define how a trading partner sends and receives messages. In a *peer-to-peer* configuration, trading partners communicate directly with each other through their delivery channels. In a *hub-and-spoke* configuration, trading partners communicate with each other through an intermediary, or routing-proxy, delivery channel. The delivery channel also defines the business protocol (eXML Message Service or RosettaNet) to be used in the conversation, the transport protocol, and security parameters.

- Collaboration agreements to map trading partners to the roles specified in conversation definitions.

To extend the B2B integration functionality, you can:

- Develop custom management and messaging applications, and enable message logging using the B2B integration API.
- Enable specialized processing of business messages using the logic plug-ins.
- Use Power.Enterprise!, an electronic data interchange (EDI)-capable system, to enable the integration of WebLogic Integration with EDI environments.
- Integrate trading partner conversations with BPM business processes using the BPM plug-in framework.

Translating and Transforming Data

Data integration enables you to integrate data from diverse enterprise applications using XML as the standard format.

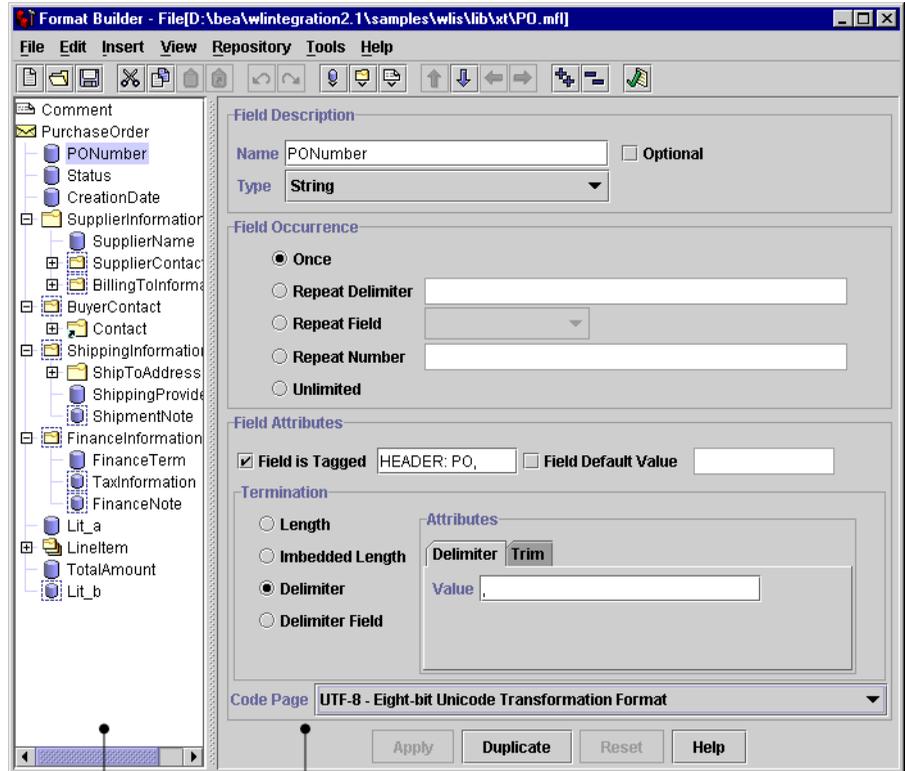
Using the data integration functionality, you can:

- Translate binary data to XML, and vice versa.
- Transform data to XML grammars, such as ebXML or RosettaNet.

To translate data, you must define a binary-to-XML map, specifically a metadata description of the record layout in the binary data, and store the information as a Message Format Language (MFL) file.

To facilitate this process, you can use Format Builder, as shown in the following figure.

Figure 19. Defining Data Translations—Format Builder



Left pane displays the structural information for the data format

Right pane displays information about the currently selected data element

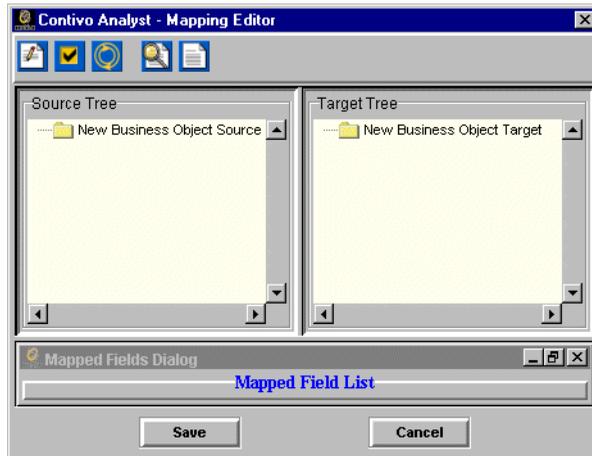
You can use Format Builder to:

- Create format descriptions derived from structural and detailed analyses of your data, and store them as MFL.
- Test format descriptions before applying them to your actual data.

To transform data, you develop XSL style sheets that describe how one XML grammar is mapped to another. You can create the XSL style sheet manually, using the design-time GUI provided by Contivo Analyst, or use the auto-mapping functionality in Contivo Analyst and Contivo eService.

The following figure shows the Contivo Analyst Mapping Editor.

Figure 20. Defining Data Transformations—Contivo Analyst Mapping Editor



To extend the data integration functionality, you can integrate with BPM business processes as follows:

- Integrate data translation functionality with business processes using the BPM plug-in framework.
- Integrate data transformation functionality with business processes using the XSL Transform workflow action in WebLogic Integration Studio.

Managing the Enterprise

As noted previously, WebLogic Platform features execute within the application server run-time environment provided by WebLogic Server, and make use of a set of interrelated, sophisticated systems resources.

Examples of system resources include:

- *Clusters*—distributing the workload of your application across multiple WebLogic Server instances. Clusters support replication, load balancing, and failover services to ensure scalability and high availability of your e-commerce systems.
- *Security*—providing a comprehensive, flexible security architecture, enabling you to control and track access to system resources. Supported security features include access control, cryptography-based privacy, user authentication, user- and group-level Access Control Lists (ACLs), Realms, Secure Sockets Layer (SSL), and Digital Certificates.
- *Transactions*—ensuring data integrity by guaranteeing that database transactions are performed accurately, in accordance with strict guidelines. Supported features include distributed transactions, updating multiple resource managers in a coordinated manner, and two-phase commits, guaranteeing that transactions are committed by all participating databases.

WebLogic Platform delivers a set of integrated management tools for managing application server resources across the enterprise.

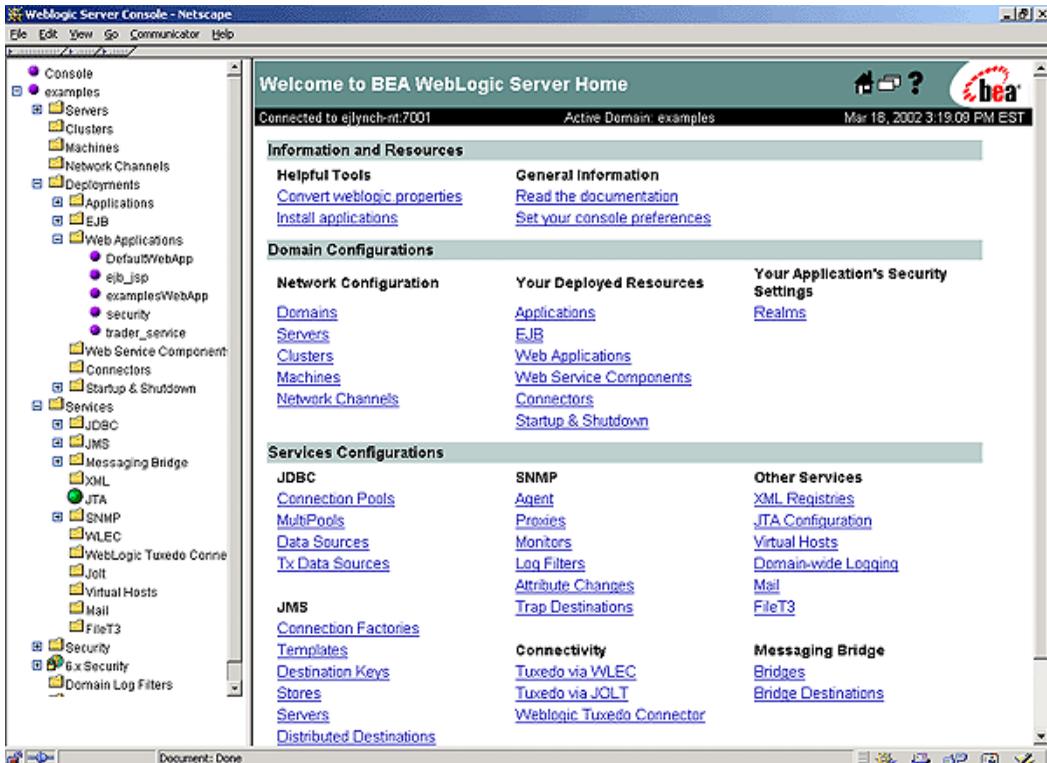
The primary tool for accomplishing the core management tasks is the WebLogic Server Administration Console Web-based interface. WebLogic Server also provides a command-line interface for developing sophisticated system management scripts, or for use with non-GUI-based systems. Additional, value-add tools are provided for configuring domains, and managing specific features of WebLogic Portal and WebLogic Integration.

WebLogic Platform management tools are described in more detail in the following sections.

Managing Application Server Resources

To manage application server resources, use the WebLogic Server Administration Console, shown in the following figure.

Figure 21. Managing Application Server Resources—Administration Console



The Administration Console is a Web-based interface to the WebLogic Administration Service, an implementation of Sun's Java Management Extension (JMX) standard that provides facilities for managing server resources.

You can perform the following set of core management tasks using the Administration Console:

- Start, stop, monitor, and configure system resources.
- Monitor and evaluate system performance.
- Balance load on servers or connection pools.
- Detect and correct problems.
- Deploy Web applications, EJBs, or other resources.

A command-line interface is also provided for developing sophisticated system management scripts, or for use with non-GUI-based systems.

Configuring Domains

The BEA Configuration Wizard enables you to create custom domains. A domain is the basic administrative unit for WebLogic Server, and consists of an interrelated set of WebLogic Server resources that are managed as a unit. A domain includes one or more WebLogic Server instances, and may also include WebLogic Server clusters.

For more information about the Configuration Wizard, see *Using the Configuration Wizard* on the BEA e-docs Web site at the following URL:

<http://edocs.bea.com/platform/docs70/configwiz/index.html>

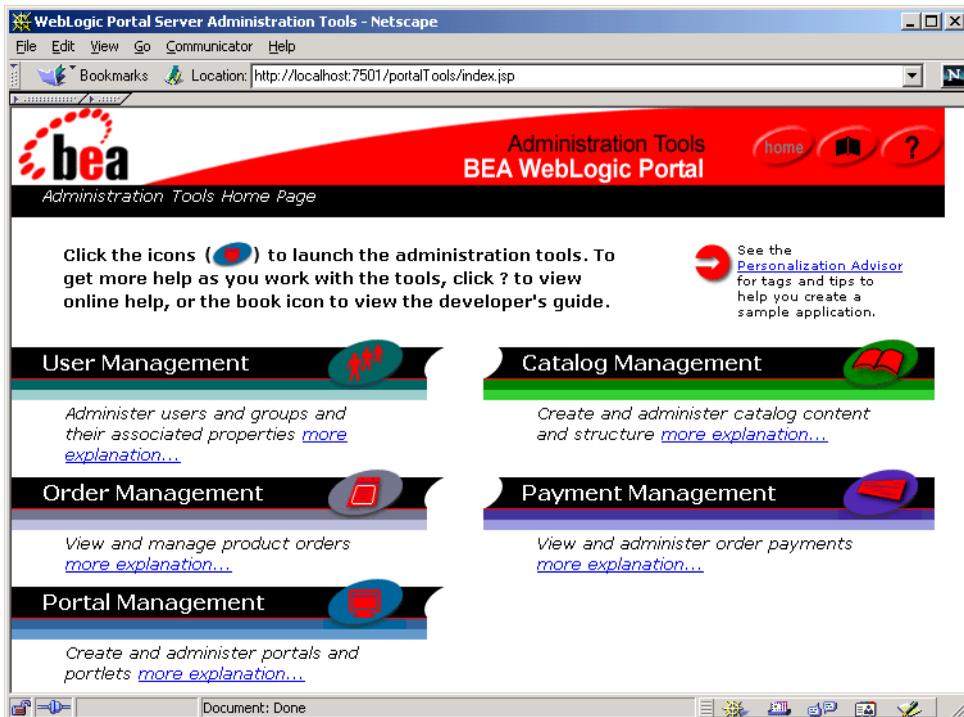
Managing Additional Features

In addition to the WebLogic Server Administration Console, you can use the tools described in this section to manage additional features of WebLogic Portal and WebLogic Integration.

Managing Additional WebLogic Portal Features

To manage specific features of WebLogic Portal, use the WebLogic Portal Administration Tools, shown in the following figure.

Figure 22. Managing WebLogic Portal Features—Administration Tools



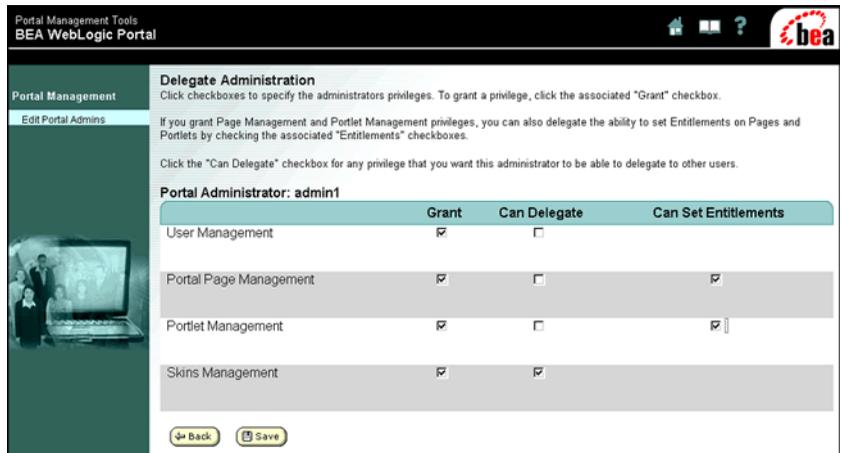
The following table describes each WebLogic Portal Administration Tool, highlighting the additional features that can be managed.

Table 7. WebLogic Portal Management Tools

Tool	Description
User Management	Manage visitor and administrative users, group memberships, and user profile attributes.
Order Management	View and modify existing customer orders.
Portal Management	Create and administer portals and portlets.
Catalog Management	Manage the behavior and content of the product catalog.
Payment Management	View and modify payment transactions.

WebLogic Portal supports the delegated administration of users, portal presentation, and entitlements management using the Delegate Administration tool shown in the following figure.

Figure 23. Delegating Administration Tasks—Delegate Administration Tool



Delegated administration enables you to create an administration hierarchy to delegate and distribute workload, or provide backup administrative support. For example, you may want to delegate portal presentation management to members of a Design group.

Managing Additional WebLogic Integration Features

To manage WebLogic Integration features, use the tools described in the following table.

Table 8. WebLogic Integration Feature Management

Service	Description
Application Integration	To manage application views for event and service adapters, use the Application View Console, shown in the figure “Managing Application Views—Application View Console” on page 31.
BPM	To manage and monitor BPM business processes, use the WebLogic Integration Studio, shown in the figure “Defining and Monitoring Business Processes—Studio” on page 32.
B2B Integration	To manage B2B integrations, use the B2B Console, shown in the figure “Configuring and Managing B2B Integration—B2B Console” on page 34.

Building a WebLogic Platform Solution

The following sections provide examples that demonstrate how to use WebLogic Platform to build an enterprise business solution. The examples show how WebLogic Platform enables a fictitious digital camera equipment supply company, Avitek Digital Imaging, to meet its business objectives.

Avitek Digital Imaging, a subsidiary of Avitek Corporation, is a leading manufacturer of audio, video communications, and information technology products for the consumer and professional markets. Avitek Corporation is a large, global enterprise, and its employees are geographically dispersed. To stock product inventory, employees must elicit quotes from multiple suppliers.

Avitek Digital Imaging’s key objectives are to:

- Increase company revenue by establishing an online, personalized sales channel.
- Increase employee productivity and reduce costs by automating the product purchasing process.

-
- Increase HR productivity by facilitating employee access to public and personal HR information.

The WebLogic Platform solution consists of:

- B2C Internet Web site enabling customers to browse the Avitek Digital Imaging product catalog, select an item, and submit an order, and, for industry professionals, receive personalized content.
- B2B purchasing portal enabling Avitek Digital Imaging purchasing agents to interact with suppliers and replenish its product inventories.
- B2E intranet Web site enabling Avitek Digital Imaging employees to access all public and personal HR material online.

With the exception of the last solution, the B2E intranet Web site, the solutions described are based on the WebLogic Platform sample application, as described in *Tour of the BEA WebLogic Platform Sample Application*.

Before diving into the detailed example descriptions, it is important to understand the WebLogic Platform feature interoperation that enables the business objectives demonstrated. The following section describes WebLogic Platform feature interoperation.

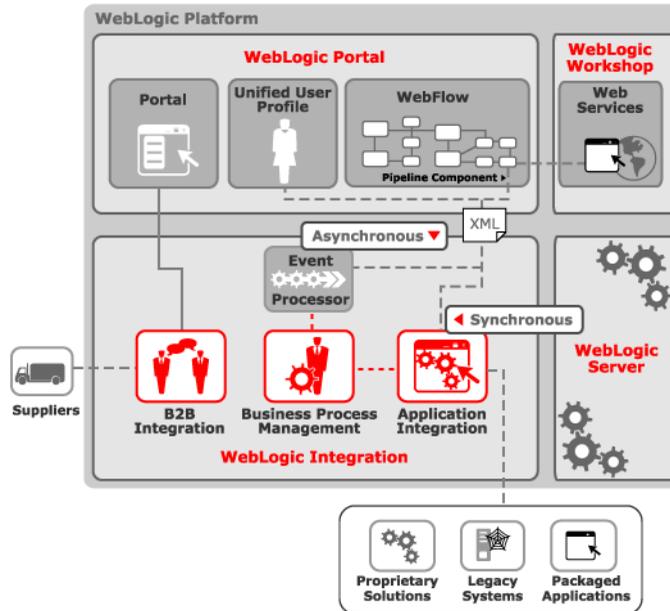
WebLogic Platform Feature Interoperation

WebLogic Platform provides a standards-based, *build-to-integrate* approach that enables companies to develop and deploy applications, rapidly integrate with existing systems, automate business processes, and connect with business partners.

To support its build-to-integrate approach and enable WebLogic Platform feature interoperation, specific integration entry points are available to facilitate process-level communication and data flow between front-end Web applications and heterogeneous back-end systems. An *integration entry point* defines a place at which interoperation can take place.

The following figure shows WebLogic Platform integration entry points.

Figure 24. WebLogic Platform Integration Entry Points



As shown in the previous figure, integration entry points include:

- *Web services*—integrate disparate systems.
- WebLogic Integration features, including:
 - *Application integration*—call a J2EE Connector Architecture (CA) adapter directly to integrate with back-end systems. Typically, this is used for synchronous communication.
 - *BPM*—communicate synchronously through calls to the API or asynchronously via JMS. Extend BPM functionality using the BPM plug-in framework.
 - *B2B integration*—communicate through a message API or the BPM plug-in collaboration functionality. Use JSP tag library to build thin-client interfaces that can communicate with a hosted B2B integration enabler.

-
- WebLogic Portal features, including:
 - *Portlets*—discover and interact with Web services or call WebLogic Integration APIs.
 - *Unified user profiles (UUP)*—maintain a single view of the user across multiple data sources including user profiles, existing user databases, and legacy applications.
 - *Webflow/Pipeline components*—manage flow of business logic and site presentation. Enable process-level communication and data flow between Web applications and enterprise information system, including WebLogic Integration features.

The following sections provide examples of how to build enterprise business solutions using WebLogic Platform, while demonstrating the interoperation scenarios described in this section.

Building an Online Sales Portal

One of Avitek’s main business objectives is to increase company revenue and maintain a competitive edge by establishing an online, personalized sales portal. Using WebLogic Platform, Avitek builds a B2C portal allowing customers to browse the Avitek Digital Imaging product catalog, select an item, and submit an order, and, for industry professionals, receive personalized content.

The following summarizes the WebLogic Platform features that Avitek employed when building the online sales portal:

- WebLogic Portal powers the front-end B2C Web application.
- Application integration communicates with back-end enterprise systems to check product availability.
- BPM streamlines the order fulfillment process.
- Web services supply product evaluation and payment authorization functionality.

The following table describes the online sales portal user interactions and the corresponding WebLogic Platform features at work behind the scenes.

Table 9. Online Sales Portal User Interactions

User Interaction	WebLogic Platform Features at Work
Customer logs into the Avitek Digital Imaging Web site	B2C portlet, built using WebLogic Portal, presents personalized view of the commerce portal.
Customer accesses product evaluation portlet to evaluate potential purchase	Product evaluation Web service, developed using WebLogic Workshop, integrates with a back-end database system and displays product evaluation information.
Customer browses product catalog and adds item to the shopping cart	Pipeline component calls an application integration adapter to communicate synchronously with a back-end inventory system to verify product availability. If item is available, portal adds item to shopping cart and calculates total cost of order. Applicable customer discounts are also applied at this time. If item is not available, portal displays “temporarily out of stock” message.
Customer proceeds to checkout screen, providing shipping address, billing information, and credit card information	Payment authorization Web service, developed using WebLogic Workshop, integrates with an external payment processor to authorize payment information. If authorized, portal displays confirmation message; otherwise, portal displays authorization failure message.
Customer confirms order	Pipeline component generates an XML message and passes it to the WebLogic Integration Event Processor for entry into the BPM workflow. BPM dequeues XML message and passes the order to a back-end order management system.

Building a Product Inventory Portal

Avitek's second business objective is to increase employee productivity and reduce costs by automating the product purchasing process. Using WebLogic Platform, Avitek builds a B2B product inventory portal that allows its purchasing agents to interact with suppliers and replenish its product inventories.

The following summarizes the WebLogic Platform features that Avitek employed in the product inventory portal:

- WebLogic Portal powers the front-end B2B Web application.
- B2B integration provides the functionality to enable purchase agent and suppliers to collaborate using standards-based conversations.
- BPM coordinates tasks among multiple systems.

The following table describes the product inventory portal user interactions and the corresponding WebLogic Platform features at work, behind the scenes.

Table 10. Product Inventory Portal User Interactions

User Interaction	WebLogic Platform Features at Work
Purchase agent logs into the Avitek product inventory portal	B2B portlet, built using WebLogic Portal, presents personalized view of the purchasing portal.
Purchase agent checks parts inventory for specific product	Portlet uses WebLogic Integration JSP tag library to communicate with the hosted B2B integration enabler, and displays parts inventory for selected product.
Purchase agent selects a part and initiates a quote for price and availability (QPA)	Portlet uses WebLogic Integration JSP tag library to trigger BPM to initiate a QPA workflow process. As part of the workflow, B2B integration sends QPA to suppliers and receives replies. BPM subsequently assigns a task to the purchase agent, returning the quotes and prompting the agent to select a supplier.
Purchase agent selects supplier and initiates a purchase order (PO)	BPM resumes workflow prompting B2B integration to send PO to selected supplier. B2B integration sends PO and receives acknowledgement from supplier.
Purchase agent checks PO status	B2B integration receives an acknowledgement back from the supplier and updates the portlet.

Building an Employee HR Portal

Avitek’s final business objective is to increase HR productivity by facilitating employee access to public and personal HR information. Using WebLogic Platform, Avitek builds a B2E intranet Web site that allows its employees to access all public and personal HR material online.

The following summarizes the WebLogic Platform features that Avitek employed in the employee HR portal:

- WebLogic Portal powers the front-end B2E Web application.
- Application integration communicates with back-end HR and benefits systems.
- BPM streamlines the benefit enrollment process.

The following table describes the employee HR portal user interactions and the corresponding WebLogic Platform features at work, behind the scenes.

Table 11. Employee HR Portal User Interactions

User Interaction	WebLogic Platform Features at Work
Employee logs into Avitek employee HR portal	B2E portlet, developed using WebLogic Portal, presents personalized view of the HR portal.
Employee browses public HR information	Pipeline component calls an application integration adapter to communicate synchronously with a back-end HR system to display requested information.
Employee browses personal benefits information	Pipeline component calls an application integration adapter to communicate synchronously with a back-end benefits database to display benefits information.
Employee enrolls in benefits and/or updates personal information	Pipeline component generates an XML message and passes it to the WebLogic Integration Event Processor for entry into the BPM workflow. BPM dequeues XML message and passes the order to a back-end benefits system.

Extending the Platform

As noted previously, WebLogic Platform delivers an extensible architecture, supporting a robust, scalable framework and enabling you to leverage current and future assets.

WebLogic Platform delivers extensibility through its support of:

- Standards-based J2EE technologies and services.
- Web services, which provide a standard way to integrate distributed systems.
- Other WebLogic Server EIS integration options, described in Table 4, including J2EE Connector Architecture (CA), WebLogic/Tuxedo Connector (WTC), and WebLogic jCOM.
- Application integration, as described in “Integrating Applications” on page 30.
- Java Management Extensions (JMX) for integrating with third-party management and monitoring tools, that also provides a Service Provider Interface (SPI) for managing security.
- Third-party Java IDEs, such as JBuilder and WebGain.
- MQ Series as a transactional resource manager.

BEA is committed to the development community, working closely with its customers, and offering assistance via the following resources:

- *BEA eSupport* (<http://support.bea.com/welcome.jsp>): Industry-leading support from BEA Customer Support, providing you with the technical expertise to effectively deliver proactive, responsive services for your critical e-business initiatives.
- *BEA dev2dev Online* (<http://dev2dev.bea.com>): Technical resources for designing and deploying applications using WebLogic Platform.
- *BEA Newsgroups* (<http://forums.bea.com>): Forum to exchange ideas, share best practices and lessons learned and discuss and obtain technical support from your peers.

- *BEA Portal Solutions Catalog*
(<http://dev2dev.bea.com/products/wlportal/psc/index.jsp>): Online resource center for businesses to design and develop enterprise portal solutions. The BEA Portal Solution Center simplifies portal design and development by providing secure, self-service access to a catalog of portlets, software integrations, and pre-designed portal solutions from BEA and leading systems integrators and software vendors.

BEA offers individual licensing programs to independent software vendors (ISVs), third-party developers, or partners that want to build their own products on top of BEA core technologies. BEA is committed to investing in strategic partnerships to provide complete e-business solutions, and works closely with each partner to deliver complete e-business solutions.

Benefits of a partnership with BEA include:

- *Build on a proven, reliable foundation*—WebLogic Server is the de facto standard application server, and is used by more developers, independent software vendors (ISVs), application service providers (ASPs), and system integrators (SIs) than any other application server.
- *Increase productivity*—achieve rapid development of e-commerce applications that use the J2EE standards for application services.
- *Reduce cost structure*—leverage existing assets and reduce number of products to learn, integrate, support, and maintain.
- *Grow your business*—achieve scalability and high-availability through open standards and clustering, supporting aggressive growth rates in a customer base.

For more information about BEA partner programs, see:

- BEA Partners at the following URL:
<http://www.bea.com/framework.jsp?CNT=index.htm&FP=/content/partners>
- *WebLogic Server Partner's Guide* on the BEA e-docs Web site at the following URL:
<http://e-docs.bea.com/wls/docs70/isv/index.html>

Conclusion

This document has explored how WebLogic Platform, through its unified, simplified, and extensible architecture, contributes to an increase in developer productivity, resulting in faster time-to-market and a reduced cost structure.

To access more information about WebLogic Platform, refer to the information sources provided in the following table.

Table 12. WebLogic Platform Information Sources

To access . . .	See . . .
A starting point for first-time users to evaluate, learn, and use WebLogic Platform	WebLogic Platform QuickStart application, providing quick access to the WebLogic Platform sample application and tour and other samples, useful tools for accomplishing specific development tasks, and the online documentation. QuickStart appears when you initially install WebLogic Platform. To invoke QuickStart manually on Windows, select BEA WebLogic Platform 7.0 > QuickStart from the Start menu. To invoke QuickStart manually on UNIX, perform the following steps: <ol style="list-style-type: none">1. Log in to the target UNIX system.2. Open a command-line shell.3. Go to the /common/bin subdirectory of the WebLogic Platform installation. For example: <code>cd /home/bea/weblogic700/common/bin</code>4. Enter the following command: <code>sh quickstart.sh</code>
A running sample application that demonstrates a WebLogic Platform solution	<i>Tour of the BEA WebLogic Platform Sample Application</i>
Detailed information on installing and using the WebLogic Platform	WebLogic Platform documentation on the BEA e-docs Web site at the following URL: http://e-docs.bea.com/platform/docs70/index.html

Table 12. WebLogic Platform Information Sources (Continued)

To access. . .	See . . .
Technical resources for designing and deploying applications using WebLogic Platform	BEA dev2dev Online at the following URL: http://dev2dev.bea.com
Forum to exchange ideas, share best practices and lessons learned, and discuss and obtain technical support from your peers	BEA Newsgroups at the following URL: http://forums.bea.com



Index

- A**
- adapters 30
 - Administration Console 39
 - Administration Tools, portal 41
 - application developer solutions 6
 - application integration. *See* integration
 - application server
 - description of 2, 6
 - managing 39
 - more information 9
 - Application View Console 31
 - application views 31
 - applications, J2EE. *See* J2EE
- B**
- B2B Console 34
 - B2B integration
 - description of 33
 - integration entry points 45
 - managing 43
 - more information 20, 30
 - B2B portal
 - description of 20
 - example 48
 - B2C portal
 - description of 20
 - example 46
 - B2E portal
 - description of 20
 - example 49
 - BEA WebLogic Integration. *See* integration
 - BEA WebLogic Portal. *See* portal
 - BEA WebLogic Server. *See* application server
 - BEA WebLogic Workshop. *See* WebLogic Workshop
 - benefits of platform 3
 - BPM
 - description of
 - integration entry points 45
 - managing 43
 - more information 30
 - build solutions 43
 - business process management. *See* BPM
- C**
- callbacks, Web service 16
 - clusters 38
 - collaboration agreements, B2B 35
 - commerce and campaign services 22
 - configuration
 - domains 40
 - hub-and-spoke, B2B 34
 - peer-to-peer, B2B 34
 - Contivo Analyst 37
 - controls, Web service 16
 - conversations
 - B2B integration 34
 - Web service 16

D

- data integration 35
- delegated administration, portal 23, 42
- delivery channels 34
- deployment
 - description of 2, 9
 - more information 9
- Deployment Descriptor Editor 9
- Design View 15
- development environment 2
- domains, configuring 40

E

- E-Business Control Center 26
- EIS applications, integrating 8
- enterprise challenges confronted 4
- enterprise developer solutions 6
- Enterprise Information System applications,
 - integrating 8
- event adapter 30
- extensible architecture 4

F

- features of platform
 - extensible architecture 4
 - illustration of 1
 - interoperation of 44
 - simplified architecture 4
 - summary of 2
 - unified architecture 4
- Format Builder 36

H

- hub-and-spoke configuration, B2B 34

I

- integration

- application 30
- B2B 33
- BPM 32
- data 35
- description of 3, 30
- EIS applications 8
- entry points 45
- framework 29
- J2EE Connector Architecture 8
- managing 43
- more information 20
- portal services 22
- summary of services 30
- WebLogic jCOM 8
- WebLogic/Tuxedo Connector 8
- interoperation 44

J

- J2EE
 - accessing functionality via Web services 16
 - Connector Architecture 8
 - deploying applications 9
 - developing applications 6
 - leveraging 7
- Java Message Service. *See* JMS
- Java Web Services file 13
- javadoc annotations, Web service 16
- JMS 7, 8, 12
- JWS file 13

L

- layout, portal 22

M

- management
 - application server 39
 - core tasks 40

- domains 40
- integration 43
- overview 38
- portal 41

Message Format Language file 35

messages

- buffers, Web service 16
- exchanging enterprise 8
- JMS 7, 8, 12
- XML 12

methods, Web service 16

MFL file 35

O

operations, administration, and management
3

P

peer-to-peer configuration, B2B 34

personalization, portal 23

Pipeline components

- as integration entry points 46
- description of 24

portal

- Administration Tools 41
- appearance 22
- B2B 20, 48
- B2C 20, 46
- B2E 20, 49
- building 19
- commerce and campaign services 22
- delegated administration 23, 42
- description of 2, 20
- E-Business Control Center 26
- example 21, 46, 48, 49
- features 22
- foundation services 22
- integration entry points 46
- intelligent administration 23

- layout 22
- Management tool 25
- managing 41
- more information 20
- personalization 23
- Pipeline component 24
- portlet selection 22
- presentation services 22
- rules-based entitlements 23
- skins 22, 25
- system and service integration 22
- tools 24
- types 20
- unified user profile 24, 46
- Webflow and Pipeline editor 27
- Webflow component 23, 46
- wizard 28

portlet

- as integration entry point 46
- description of 21
- integration 24
- selection 22
- wizard 28

presentation services, portal 22

properties, Web service 16

protocols, Web service 16

Q

QuickStart application 3, 52

R

rules-based entitlements 23

S

sample application and tour 44

security

- description of 3
- managing 38

- more information 5
- service adapter 30
- simplified architecture 4
- skins, portal 22, 25
- solutions
 - application developer 6
 - B2B portal 48
 - B2C portal 46
 - B2E portal 49
 - employee HR portal 49
 - enterprise developer 6
 - online sales portal 46
 - overview 43
 - product inventory portal 48
- Source View 17
- Studio 32

T

- Test View 18
- topic summary 4
- tour and sample application 44
- trading partners 34
- transaction management 38
- transformation, data 35
- translation, data 35
- two-way editing, WebLogic Workshop 15

U

- unified architecture 4
- unified user profiles
 - as integration entry points 46
 - description of 24
- UUP
 - as integration entry point 46
 - description of 24

W

- Web services

- as integration entry points 45
- callbacks 16
- controls 16
- conversations 16
- deploying and managing 19
- description of 7, 10
- developing 9
- developing with WebLogic Workshop 13
- development options 12
- editing source code 17
- example 11
- message buffers 16
- methods 16
- properties 16
- protocols 16
- standards 11
- testing and debugging 18
- visually designing 15
- WebLogic Workshop IDE 14
- WSDL file 11
- XML maps 17
- Webflow and Pipeline Editor 27
- Webflow components
 - as integration entry points 46
 - description of 23
- WebLogic Builder 9
- WebLogic EJBGen 9
- WebLogic Integration. *See* integration
- WebLogic jCOM 8
- WebLogic Portal. *See* portal
- WebLogic Server. *See* application server
- WebLogic Workshop
 - description of 13
 - Design View 15
 - IDE 14
 - more information 14
 - run-time environment 19
 - Source View 17
 - Test View 18
 - two-way editing 15

WebLogic/Tuxedo Connector 8

wizard

- portal 28

- portlet 28

Worklist 33

WSDL file 11

WTC 8

X

XML

- editor 9

- maps 17

- messages 12

