



BEA WebLogic Platform™

Introducing WebLogic Platform

Version 8.1
Document Date: July 2003
Part Number: 885-001002-003

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 JRocket, 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

Introduction to WebLogic Platform

Components of WebLogic Platform	1
WebLogic Server—Foundation for the Platform	2
WebLogic Workshop—Development and Run-Time Framework	3
WebLogic Portal—Window into Data and Applications	4
WebLogic Integration—Connecting Enterprise Solutions	4
WebLogic JRockit—High-Performance JVM	5
Benefits of WebLogic Platform	5
Common Application Infrastructure	6
Common Development Paradigm	6
Applications You Can Create in Workshop	9
WebLogic Server Programming Support	11
Solutions You Can Implement with WebLogic Platform	11
About the End-to-End WebLogic Platform Tour	11
Developing the Intranet Portal Using WebLogic Portal	12
Controlling the User Path Using Page Flows	13
Using a Web Service to Invoke Business Processes	13
Developing the Order Management System Using WebLogic Integration	14
Using a WebLogic Workshop Enterprise Java Bean Control to Track Orders	16
End-to-End WebLogic Platform Tour Online Demos	16
For More Information About WebLogic Platform	18
Contact Us!	25

Introduction to WebLogic Platform

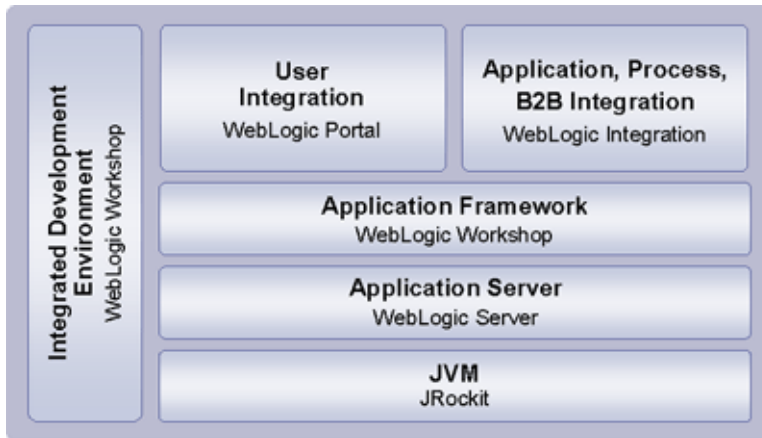
BEA WebLogic Platform is a unified, simplified, and extensible platform for building, extending, integrating, deploying, and managing applications. WebLogic Platform includes a rich development and deployment framework for integrating valuable resources and tools into your environment: third-party application packages, custom applications, messaging systems, Web services, trading partners, and more.

Based on a foundation of open standards that are leveraged throughout, WebLogic Platform ensures interoperability, flexibility, and choice, enabling *integration* with and between existing applications and resources, and *extensibility* through ISV applications and solutions.

This document is intended for software developers and managers who are new to WebLogic Platform and who want to understand its features and benefits.

Components of WebLogic Platform

The WebLogic Platform product comprises multiple component products that can be used independently, or in combination, as required for your application. These component products, shown in the following diagram, are referred to as *components* for the remainder of this document.



The highlights of each component are described in the following sections.

WebLogic Server—Foundation for the Platform

WebLogic Server delivers an easy-to-use, industrial application infrastructure for building and integrating distributed applications. WebLogic Server centralizes application services, such as Web server functionality, business components, and access to back-end enterprise systems. It is based on standards such as J2EE, Web services, and XML, and it provides standards-based integration to enable application integration and investment protection. WebLogic Server also provides enterprise-level security and powerful administration facilities.

WebLogic Server provides the foundation for WebLogic Platform. The WebLogic Workshop, WebLogic Portal, and WebLogic Integration components, and all applications built with these components, utilize the WebLogic Server run-time environment to meet the demands of applications that span one or more enterprises. Enterprise-class run-time capabilities provided by WebLogic Server include the following.

This feature . . .	Provides the ability to . . .
Reliability	Recover from a system failure with no noticeable interruption in service, and to protect session state. When session state is protected, any failures that occur can be fully masked from the user and the application.
Scalability	Scale up to meet demand seamlessly and dynamically.
Security	Provide a single, unified security framework for all deployments—including portal applications, integration applications, and custom J2EE applications—that integrates best-of-breed security tools into a flexible, unified system.

WebLogic Workshop—Development and Run-Time Framework

WebLogic Workshop is a unified, integrated development framework that makes it easy for all developers—not only J2EE experts—to build powerful, standards-based J2EE applications across the entire WebLogic Platform. The WebLogic Workshop architecture includes the following basic components:

- **Workshop Integrated Development Environment (IDE)**

A graphical environment that you can use for building, testing, and debugging WebLogic Platform applications. The Workshop IDE's intuitive user interface lets you design your application visually and hides the complexity of J2EE programming.

- **Workshop Run-Time Framework**

A program that compiles the annotated code created in the IDE and automatically implements the appropriate J2EE components, such as EJBs, JSPs, and JDBC connections, that are required to build and run the specified application.

The WebLogic Workshop IDE supports the ability to build applications that include any combination of the following:

- Enterprise-class Web services
- Web applications
- Java controls
- Enterprise Java Beans

- Portals
- Business processes

WebLogic Portal—Window into Data and Applications

WebLogic Portal simplifies the production and management of custom-fit portals. WebLogic Portal simplifies, personalizes, and lowers the cost of access to information, applications, and business processes, whether the person who needs such access is an employee, a partner, or a customer. WebLogic Portal achieves this goal by leveraging a unified portal framework, simplified portal lifecycle management, and modular portal business services.

To speed the development and administration of portal applications, WebLogic Portal provides the following features.

With this feature . . .	WebLogic Portal provides . . .
Portal business services	Modular business services that support enterprise portal and application development. These services enhance developers' ability to build custom portals and administrators' ability to tailor the experience of a portal user.
Lifecycle management	Functionality and tools tailored to specific users, including application developers, JSP developers, HTML and graphic designers, system administrators, portal administrators, and business analysts.
Unified portal framework	An extensible, scalable, and standards-based framework that supports the development of flexible user interfaces to provide secure access to information, applications, and business processes.

WebLogic Integration—Connecting Enterprise Solutions

WebLogic Integration provides a development and run-time framework for unifying business process management, application integration, and B2B integration into a single, flexible environment.

With this feature . . .	WebLogic Integration provides . . .
Business process management	Environment for quickly modeling, automating, and analyzing business processes that controls the flow of business activity by coordinating the interoperation of applications, business users, and trading partners. Based on Java Specification Request (JSR) 207, Process Definition for Java.
Enterprise resource access	Utilities and other resources for providing prompt and secure access to resources inside and outside the enterprise by business users, trading partners, and enterprise software applications.
Dynamic integration services	Toolkit that includes adapters for delivering native connectivity to leading enterprise packaged applications, legacy systems, and mainframes; a message broker for providing high-reliability, asynchronous messaging inside and outside the company; data transformation for translating between XML and non-XML messages; and Web services.

WebLogic JRockit—High-Performance JVM

WebLogic JRockit is a high-performance JVM optimized for server-side performance and scalability. WebLogic JRockit is supported by all WebLogic Platform components and is now fully integrated into the WebLogic Platform package. It is also available as a standalone JVM.

The following features make WebLogic JRockit a strong match for WebLogic Platform:

- Support for J2SE 1.4.1, JVMPI, and JVMDI
- Integration with third-party profiling, development, and management tools
- Support for 32-bit and 64-bit Intel architecture on Windows and Linux systems

WebLogic JRockit is included with WebLogic Platform and is also available as a standalone product.

Benefits of WebLogic Platform

WebLogic Platform provides a single, highly integrated, high-performance, reliable solution, combining the advantages of a common application infrastructure with an easy-to-use, robust application development and management framework for the following classes of applications:

- Portal initiatives

- Enterprise integration
- Custom development

The following sections describe the commonly used application infrastructure and development paradigm supported by WebLogic Platform.

Common Application Infrastructure

WebLogic Platform's common application infrastructure includes the following:

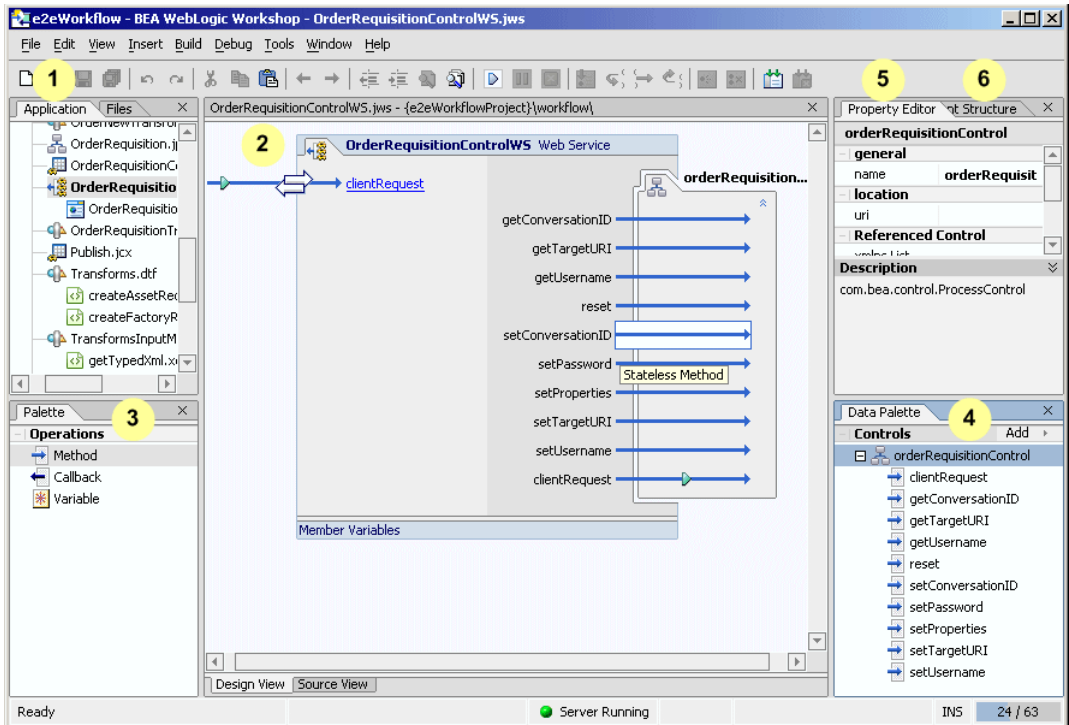
- **Common framework for easy installation on target systems** that supports the selection of Platform components to install; control of software usage through a single licensing mechanism; and support for commonly used hardware, operating system, and database combinations
- **Common configuration of Platform components** that supports the easy setup for application development, migration of applications to production environments, and the flexibility to accommodate the addition of Platform components to existing applications in response to new business requirements
- **A common development paradigm.** The Workshop visual programming model is based on the notion of controls—simple components that have methods, events, and properties with a visual representation in the design view—and annotated Java code, which lets you declaratively specify behavior and focus on handling events and calling methods, instead of writing complex object-oriented infrastructure code
- **A highly reliable, available, scalable, extensible, standards-based, and high-performing foundation**—WebLogic Server—that allows you to incorporate flexibility, extensibility, and choice into your IT solutions
- **A unified and simplified management architecture**, populated by distributed, heterogeneous technologies and platforms, that enables developers and administrators to realize business objectives

Common Development Paradigm

WebLogic Workshop's unified architecture makes it possible for developers to learn only a single programming model for building and integrating the full suite of WebLogic Platform applications, including Web services, Web applications, portals, and business processes. Its development environment provides an intuitive user interface that lets you design your applications visually. It also abstracts the more complex J2EE programming. Applications you build in the IDE are constructed from high-level components rather than low-level API calls.

The most common way in which developers become familiar with WebLogic Platform is by experimenting with application development or by prototyping initial applications. The Workshop IDE is shown in [Figure 1](#).

Figure 1 WebLogic Workshop IDE



The Workshop IDE simplifies application construction by providing a set of graphical elements, including the following:

1. The Application window is an area in which you can create, view, and edit files for your application. This window presents a hierarchical view of application files, making it easy to add or select files to work on.
2. A visual document workspace that presents two-way editing of your application: the Design View mode and the Source View mode. The Design View is a handy graphical tool for making coarse-grained modifications to your application, such as the addition of controls and method declarations. The Source View is useful for entering code, such as when you want to flesh out

method logic. In the preceding figure, a Web service is being edited in the Design View. When you select an entity in the document workspace, the properties of that entity are displayed in the Property Editor.

3. The Palette contains often-used items that you may want to add to the entity in the document workspace. To create a new instance of a particular type of item, simply drag an existing instance of the desired item from the Palette to the document workspace.
4. The Data Palette displays a structural-list view of either the methods or attributes of controls and tags. For example, in the preceding figure, the Data Palette lists the methods of the `OrderRequisitionControl` process control. As you drag items from the Data Palette to the document workspace, code is inserted in the current entity to access the item being dragged.
5. The Property Editor enables you to view and set properties for the entity that is currently selected in the document workspace. For example, if you are creating a portal, you can customize the look and feel of that portal by setting its properties in the Property Editor.
6. The Document Structure window allows you to display a structured view of the components for the file that is currently selected in the document workspace.

The IDE also includes the following:

- **Java controls**—Components that have a predefined set of methods, events, and properties that encapsulate business logic. For example, you may have a control that can perform a database query, receive a JMS message, or execute business logic and return a result to the Workshop application. Java controls provide a powerful means of abstracting code so that a developer can focus on handling events and calling methods, instead of writing complex object-oriented J2EE infrastructure code.
- **Java annotations**—Code that conveys the appropriate run-time application behavior specified by the components you add, such as controls and property attributes. As you construct applications, the IDE adds these Java annotations, along with the required standard Java files. The Workshop run-time framework interprets these annotations and automatically implements the appropriate J2EE components, such as EJBs, JSPs, and JDBC connections, required to build the specified application.
- **Debugging tool**—Workshop supports an advanced cross-component debugging system. For example, this allows you to step from JSP code, to page flow code, and to code inside custom controls that you have written. Moreover, the debugging framework is leveraged by the other Platform products, so you can even debug from a business process into custom control code or JSPs. The JSP debugger lets you step through the execution of an individual JSP page and view the output stream as it is being constructed. Debugging performance has also increased significantly, and immediate mode and multithreaded debugging support have been added.

- Source editing features—Workshop’s source code editor supports many popular productivity features. For example:
 - When you use a class that has not yet been imported into your code, Workshop automatically prompts you to import that library.
 - Holding down the Ctrl key and hovering over functions and variables provides additional information and links, respectively, to their definitions or declarations.
 - Basic source editing features like *auto completion* have also been enhanced and expanded beyond Java functions and variables. For example, an `href` tag in a JSP or HTML page “autocompletes” to any file or image in the project.
- Online help—Tool that provides comprehensive reference information and hands-on instructions, as well as a rich set of tutorials that show how to build each type of application supported in Workshop.

Applications You Can Create in Workshop

While you are developing applications in Workshop, the IDE provides easy-to-use access to views into those applications, in both the graphical Design View mode and the programmatic Source View mode. It also offers editors and palettes appropriate for each type of application, including the following:

- Web applications

To help you build dynamic, sophisticated Web applications, WebLogic Workshop provides Java page flows. A page flow is a Java class that controls the behavior of Web application components through the use of Javadoc annotations, methods, and, in most cases, forms. Java page flows provide an abstraction layer that helps you maintain efficiency in the following aspects of your Web application:

 - Connections between pages
 - Business logic actions that can be executed by pages
 - Bindings between pages and back-end data sources
- Enterprise-Class Web Services

WebLogic Workshop makes it very easy to create and deploy Web services that adhere to Web services standards, such as Simple Object Access Protocol (SOAP) and Web Services Description Language (WSDL). Web services are accessed via XML messages. WebLogic Workshop provides tools for accessing and processing XML files received by Web services and for formatting XML responses. Web services built with WebLogic Workshop automatically leverage the power, reliability, security, and scalability of WebLogic Server, making them truly enterprise-class.

- Java controls

The Java controls model is extensible so that all users, including independent software vendors, can build tightly-coupled business logic components that seamlessly plug in to the Workshop's Integrated Development Environment (IDE). A key feature of a Java control is its ability to contain other controls. By nesting, you can use controls to wrap potentially complex business logic, along with access to resources, while exposing a relatively simple interface to developers. Once a control is created, it can be leveraged across all applications developed with Workshop.

- Enterprise Java Beans

Enterprise Java Beans (EJBs) are server-side Java components of enterprise applications. The J2EE Specification defines the types and capabilities of EJBs, as well as the environments, or containers, in which EJBs are deployed and executed. WebLogic Workshop provides a built-in control, called the EJB control, that makes it easy to use an existing, deployed EJB in your application. WebLogic Workshop also enables you to create new EJBs.

- Portal applications

WebLogic Portal provides a Workshop extension that includes designers for building portals, portlets, portal foundation services, personalization and interaction management, and integration services. These designers are graphical tools that enable you to create a sophisticated portal in minutes, by using a drag-and-drop function to manipulate data and setting properties on a grid. For example, if you create a portal file that appears with a default page and drag a sample JSP from the file tree onto that default page, a portlet is created for you automatically. You can then save the file and view the portal in a browser.

- Business processes

WebLogic Integration provides a Workshop extension for graphically implementing business processes that integrate applications, business users, trading partners, and other resources in the enterprise. Using the business process extension, the developer can define the process flow, the resources the process interacts with and the data that is passed to and from those resources, and the appropriate exception handling logic. The Mapper extension allows the developer to graphically map between XML, non-XML, and Java data within the business process. A set of integration controls is provided for use within the business process for integrating to a range of resources.

All these applications can be combined into a single application, all built in the same tool, and all sharing the same run-time characteristics provided by WebLogic Server.

Note that Workshop is a file-based system. All files created in Workshop are standard Java classes with annotations. While Workshop has many productivity features built into it, developers have the flexibility to use other IDEs for building applications that run on the Workshop run-time framework.

WebLogic Server Programming Support

WebLogic Platform also provides and supports traditional WebLogic Server programming environments that allow developers to work within a level of complexity that is appropriate for them. For developers who prefer a flexible, low-overhead, custom development environment that includes off-the-shelf editors, such as Emacs or vi, WebLogic Server provides custom Ant tasks for the repetitive tasks required to build, configure, and deploy WebLogic Server applications. Any application built with these tools can be easily integrated into any other application that has been developed in the Workshop IDE.

Solutions You Can Implement with WebLogic Platform

WebLogic Platform provides a comprehensive infrastructure for combining custom development, portal initiatives, and enterprise application integration in a single development and run-time environment. To demonstrate the kinds of solutions you can implement, the WebLogic Platform distribution includes a sample application, called the End-to-End WebLogic Platform Tour. This sample application introduces you to the principal features of the WebLogic Platform product. It also provides a launching point from which you can get started with the development and integration of your own applications.

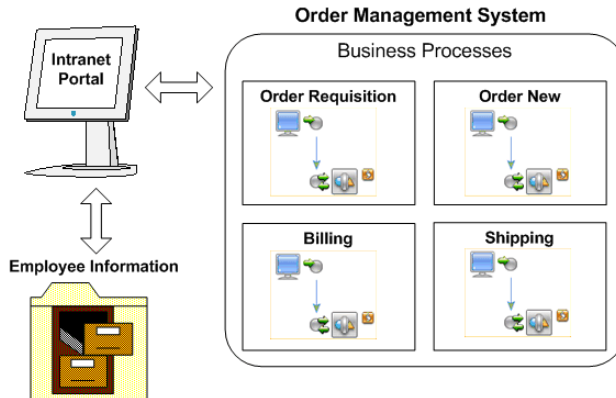
This section describes the End-to-End WebLogic Platform Tour: it summarizes the structure of the application and explains how key components of the application are developed with WebLogic Workshop. This section also emphasizes ways in which you can extend this application to encompass an even broader range of resources, giving you a robust and comprehensive implementation.

About the End-to-End WebLogic Platform Tour

The WebLogic Platform Tour depicts a scenario in which Avitek, a major vendor of electronic office equipment, has developed a corporate intranet using WebLogic Platform. The corporate intranet enables employees and managers to view information from an employee information database and manage office equipment purchases. These capabilities are provided by two systems:

- The Employee Information system comprises a database with a “profile” of each employee. Employees can view the information in their profiles.
- The Office Equipment Order Management system integrates multiple business processes, such as Order Requisition, Order New, Billing, and Shipping, that are required in the life cycle of an equipment purchase. Each business process, in turn, invokes application modules that execute the back-end business logic and database transactions needed by the system.

The WebLogic Platform Tour shows how these functions can be integrated using WebLogic Platform.



The following sections explain how select components of the WebLogic Platform Tour are implemented with WebLogic Workshop:

- [Developing the Intranet Portal Using WebLogic Portal](#)
- [Controlling the User Path Using Page Flows](#)
- [Using a Web Service to Invoke Business Processes](#)
- [Developing the Order Management System Using WebLogic Integration](#)
- [Using a WebLogic Workshop Enterprise Java Bean Control to Track Orders](#)

This section concludes with links to a set of interactive demos from the documentation for the WebLogic Platform Tour. You can invoke these demos to see how to develop specific application components in Workshop.

Developing the Intranet Portal Using WebLogic Portal

The underlying purpose of the intranet portal described in the WebLogic Platform Tour is to provide Avitek's employees with a single point of access to the company's applications and data. Specifically, the intranet portal must do the following:

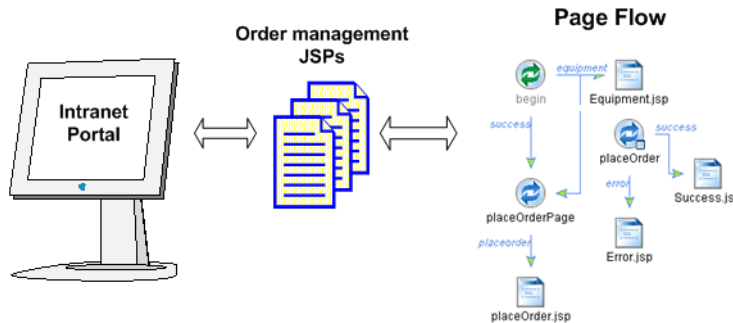
1. Provide access to the Office Equipment Order Management System
2. Give employees access to their profiles in the employee database
3. Provide secure access to applications and data

You can use the WebLogic Workshop Portal Extensions to quickly create a portal that has these features and more. WebLogic Portal provides the ability to integrate existing Web applications,

whether developed with WebLogic Workshop or with other tools and application environments, into a single portal that provides a unified view of applications. Web applications integrated into the WebLogic Portal environment, in turn, provide user access to back-end application functionality, databases, and other IT resources.

Controlling the User Path Using Page Flows

The application accessed by this intranet portal consists of multiple Web pages. These Web pages need to be linked together into a unified set, so that an employee's path is well coordinated, and so that a variety of usage scenarios, including error conditions, can be handled appropriately. This linkage can be performed by a WebLogic Workshop feature called page flows, as shown in the following figure.



A page flow links together multiple Web pages in an application and coordinates the user's path through those pages and the associated flow of data. A page flow consists of a Java class with annotations, methods, and forms that are specifically designed to control the behavior of application components. Page flows are based on the Apache Struts framework.

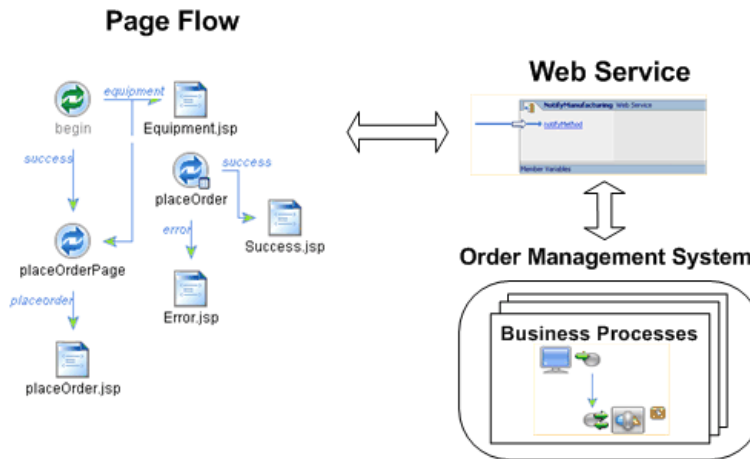
Using a Web Service to Invoke Business Processes

A Web service makes software application resources available over networks using standard technologies. Because Web services are based on standard interfaces, including XML, HTTP, SOAP, and WSDL, they can communicate even if they are running on different operating systems and are written in different languages. For this reason, Web services provide an excellent approach for building distributed applications that must incorporate diverse systems over a network, such as the intranet portal and Office Equipment Order Management System in the WebLogic Platform Tour.

WebLogic Workshop provides the tools and infrastructure you need to build and deploy enterprise-class Web services based on these standard technologies. WebLogic Workshop's

environment for building Web services is both simple enough for the application developer and powerful enough for the J2EE expert.

The WebLogic Platform Tour uses a Web service to invoke the Order Requisition business process in the Office Equipment Order Management System, shown in the following figure. This Web service enables the application accessed by the intranet portal to be completely decoupled from the back-end business logic of the Office Equipment Order Management System.



Developing the Order Management System Using WebLogic Integration

The Office Equipment Order Management System in the WebLogic Platform Tour is an application consisting of business processes that do the following:

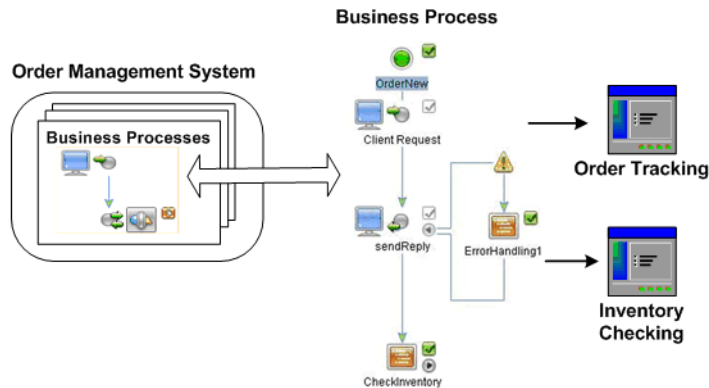
- Control the lifecycle of an office equipment order
- Enable users to interact with business processes by using a WebLogic Integration feature called the Worklist
- Employ built-in Java controls, developed using WebLogic Workshop, that simplify access to business logic and enterprise resources

The Office Equipment Order Management System integrates these applications so that they can exchange documents and data efficiently. Some of the business logic and enterprise resources to be integrated can be extended to include the following:

- Custom software built by the company

- Prepackaged ERP applications provided by an independent software vendor
- Legacy systems that the company must continue using because it has made such big investments in them

The following figure shows how a business process fits into the Office Equipment Order Management System software to control the execution flow of various operations in that system.



WebLogic Integration provides the following capabilities to support the application integration and interoperability requirements of this Office Equipment Order Management System:

- *Business process management* provides an environment for quickly modeling, automating, and analyzing business processes that coordinate the interoperation of applications along with human interactions, to control the flow of business activity. For example, when a new equipment order arrives from a portal, a business process can send a message to the inventory application, to check for availability. Or, the business process can send a message to a manager for approval, and branch the process based on the response received.
- *Enterprise resource access* makes it possible to integrate multiple order management applications. The WebLogic Platform can be extended to include trading partner applications that coordinate online transactions with external suppliers. For example, equipment purchases might be coordinated by a RosettaNet business process, which could be integrated into the system.
- *Dynamic integration services* include:
 - Adapters for delivering native connectivity to packaged enterprise applications. For example, the Office Equipment Order Management System might include an adapter for a prepackaged application that manages inventory.

- Message brokers provide highly reliable, asynchronous messaging, both inside and outside the company. For example, the message broker in the Office Equipment Order Management System can be configured to listen for incoming purchase order responses from trading partners.
- Data transformation for translating between XML and non-XML messages, which simplifies the exchange of documents by two or more of the applications in the Office Equipment Order Management System.

Using a WebLogic Workshop Enterprise Java Bean Control to Track Orders

EJB controls make it easy for you to use an existing, deployed EJB from within an application. Normally, to obtain access to an Enterprise Java Bean (EJB), you must perform several preparatory operations: look up the EJB in the JNDI registry, obtain its home interface, and then obtain an instance of that EJB.

WebLogic Workshop uses the notion of an EJB control to simplify access to an EJB. Once you have created an EJB control, any application you create in WebLogic Workshop—a Web service, a business process, a page flow—can use the control to access the EJB’s business methods directly. The EJB control manages communication with the EJB for you, including all JNDI lookup, interface discovery, and EJB instance creation and management.

The WebLogic Platform Tour provides an EJB to track the status of office equipment orders. The business processes in the Office Equipment Order Management System use an EJB control to simplify the task of accessing this EJB when updating or displaying the status of an order.

End-to-End WebLogic Platform Tour Online Demos

The documentation for the End-to-End WebLogic Platform Tour includes the following online demos that show how to use WebLogic Workshop to build specific parts of the Tour application.

This demo . . .	Shows how to . . .
Designing a Portal	Use the portal design tools
Designing a Portlet	Use the portlet design tools
Designing a Page Flow	Use the page flow design tool
Creating JSP Content	Use the JSP design tools

This demo . . .	Shows how to . . .
Building a Control	Create and add a database Java control
Designing a Business Process	Design a business process using the business process design tools

Together with its online documentation and demos, the WebLogic Platform Tour provides a comprehensive introduction to the sorts of applications and components you can build with WebLogic Workshop. Give the Tour a try and find out how easy it is to build and deploy enterprise-class applications with WebLogic Platform!

For More Information About WebLogic Platform

The following table provides links to useful information in the WebLogic Platform documentation set, such as topics that teach you how to use the tools for implementing, deploying, configuring, and monitoring applications and resources that run on WebLogic Platform.

Table 1 WebLogic Platform Topics

To learn about . . .	See the following . . .
Getting started with the WebLogic Platform components	<i>Installing BEA WebLogic Platform</i> available at: http://e-docs.bea.com/platform/docs81/install/index.html
	<i>Creating WebLogic Configurations Using the Configuration Wizard</i> available at: http://e-docs.bea.com/platform/docs81/configwiz/index.html
	<i>Supported Configurations</i> available at: http://e-docs.bea.com/platform/suppconfigs/index.html
	<i>BEA WebLogic Platform Tour Guide</i> , available at: http://e-docs.bea.com/platform/docs81/tour/index.html
	<i>Introduction to WebLogic Server and WebLogic Express</i> , available at: http://e-docs.bea.com/wls/docs81/intro/index.html
	<i>WebLogic Workshop Help</i> , available at: http://e-docs.bea.com/workshop/docs81/doc/en/core/index.html
	<i>Introducing WebLogic Integration</i> , available at: http://e-docs.bea.com/wli/docs81/overview/index.html
	<i>Building Portal Applications</i> , available at: http://e-docs.bea.com/workshop/docs81/doc/en/portal/overview/BuildingPortalAppsOV.html
	<i>Using WebLogic JRockit 1.4.2 SDK</i> , available at: http://e-docs.bea.com/wljrookit/docs142/userguide/intro.html

Table 1 WebLogic Platform Topics (Continued)

To learn about . . .	See the following . . .
Using WebLogic Workshop	
WebLogic Workshop IDE tutorials	See the following topics in the <i>WebLogic Workshop Help</i> , available at: http://e-docs.bea.com/workshop/docs81/doc/en/core/index.html <ul style="list-style-type: none"> • Tutorial: Java Control • Tutorial: Web Services • Tutorial: Page Flow • Part One: Building Your First EJBs • Portal Tutorials • Tutorial: Building Your First Business Process • Tutorial: Building Your First Data Transformation
Developing J2EE applications using WebLogic Workshop	See the following topics in the <i>WebLogic Workshop Help</i> , available at: http://e-docs.bea.com/workshop/docs81/doc/en/core/index.html <ul style="list-style-type: none"> • The WebLogic Workshop Development Environment • Java and XML Basics • Working with Java Controls • Developing Web Applications • Building Web Services • Designing Asynchronous Interfaces • Handling XML with XMLBeans • Configuring a Data Source • Developing Enterprise JavaBeans
Deploying Applications	See the following topics in the <i>WebLogic Workshop Help</i> , available at: http://e-docs.bea.com/workshop/docs81/doc/en/core/index.html <ul style="list-style-type: none"> • WebLogic Workshop Security Overview • Deploying an Application to a Production Server

Table 1 WebLogic Platform Topics (Continued)

To learn about . . .	See the following . . .
Using WebLogic Integration	
Using the WebLogic Integration components	<i>Introducing WebLogic Integration</i> , available at: http://e-docs.bea.com/wli/docs81/overview/index.html
	<i>Introducing Trading Partner Integration</i> , available at: http://e-docs.bea.com/wli/docs81/tpintro/index.html
	<i>Using the Worklist</i> , available at: http://e-docs.bea.com/wli/docs81/worklist/index.html
	<i>WebLogic Integration 8.1 Upgrade Guide</i> , available at: http://e-docs.bea.com/wli/docs81/upgrade/index.html
	<i>WebLogic Integration Solutions Best Practices FAQ</i> , available at: http://e-docs.bea.com/wli/docs81/design/index.html
	<i>Managing WebLogic Integration Solutions</i> , available at: http://e-docs.bea.com/wli/docs81/manage/index.html
	<i>Deploying WebLogic Integration Solutions</i> , available at: http://e-docs.bea.com/wli/docs81/deploy/index.html
	<i>Introducing Application Integration</i> , available at: http://e-docs.bea.com/wli/docs81/aiover/index.html
	<i>Using the Application Integration Design Console</i> , available at: http://e-docs.bea.com/wli/docs81/aiuser/index.html
	BEA WebLogic Adapters documentation Web site available at: http://e-docs.bea.com/wladapters/docs81/index.html
<i>Developing Adapters</i> , available at: http://e-docs.bea.com/wli/docs81/devadapt/index.html	

Table 1 WebLogic Platform Topics (Continued)

To learn about . . .	See the following . . .
	<p>See also the following topics in the <i>WebLogic Workshop Help</i>, available at: http://e-docs.bea.com/workshop/docs81/doc/en/core/index.html</p> <ul style="list-style-type: none"> • Tutorial: Building Your First Business Process • Tutorial: Building Your First Data Transformation • Guide to Building Business Processes • Guide to Data Transformation • Using Integration Controls
Using WebLogic Portal	
Using the WebLogic Portal components	<p>The following topics in the <i>WebLogic Workshop Help</i>, available at: http://e-docs.bea.com/workshop/docs81/doc/en/core/index.html</p> <ul style="list-style-type: none"> • Building Portal Applications • Portal Overview • Developing Portal Applications • Developing Personalized Applications • Developing Portal User Interfaces • Assembling Portal Applications • Securing Portal Applications • Deploying Portal Applications • Portal Reference <p>WebLogic Administration Portal Online Help, available at the following location, describes tasks and information about managing portal applications: http://e-docs.bea.com/wlp/docs81/adminportal/index.html</p>
Using WebLogic Server	
WebLogic Server tutorials	<p>Application Examples and Tutorials Web page, available at the following location, provides a description of the Avitek Medical Records sample applications: http://e-docs.bea.com/wls/docs81/samples.html</p> <p>Avitek Medical Records Clustering Tutorials, available at the following location, guide you through the process of creating and configuring a new cluster: http://e-docs.bea.com/wls/docs81/medrec_cluster/index.html</p>

Table 1 WebLogic Platform Topics (Continued)

To learn about . . .	See the following . . .
Creating applications	<p><i>Developing WebLogic Server Applications</i>, available at the following location, describes the different types of applications you can build on WebLogic Server using the J2EE programming model. This document also provides a high-level procedure for building and deploying those applications, and information about programming.</p> <p>http://e-docs.bea.com/wls/docs81/programming/index.html</p>
Programming for WebLogic Server	<p>The following documents are available at the WebLogic Server Programming Web page, available from the following location:</p> <p>http://e-docs.bea.com/wls/docs81/api.html</p> <ul style="list-style-type: none">• <i>Developing Web Applications for WebLogic Server</i>• <i>Programming WebLogic Enterprise JavaBeans</i>• <i>Programming WebLogic Server J2EE Connectors</i>• <i>Programming WebLogic HTTP Servlets</i>• <i>Programming WebLogic JSP</i>• <i>Programming WebLogic JDBC</i>• <i>Programming WebLogic JMS</i>• <i>Programming WebLogic JNDI</i>• <i>Programming WebLogic JTA</i>• <i>Programming WebLogic Management Services with JMX</i>• <i>Programming WebLogic RMI</i>• <i>Programming WebLogic RMI over IIOP</i>• <i>Programming WebLogic XML</i>• <i>Using WebLogic Server Clusters</i>• <i>Programming WebLogic Web Services</i>• <i>Using Applets with WebLogic Server</i>• <i>Programming WebLogic jCOM</i>• <i>Programming WebLogic JSP Tag Extensions</i>• <i>Programming WebLogic Server for Wireless Services</i>• <i>Programming WebLogic Security</i>• <i>Developing Security Providers for WebLogic Server</i>• <i>Using WebLogic Logging Services</i>

Table 1 WebLogic Platform Topics (Continued)

To learn about . . .	See the following . . .
Deploying WebLogic Server applications	<p><i>Deploying WebLogic Server Applications</i>, available at the following location, provides a comprehensive set of WebLogic Server deployment topics: http://e-docs.bea.com/wls/docs81/deployment/index.html</p>
Using WebLogic JRockit	
Using the JRockit JVM	<p><i>Using WebLogic JRockit 1.4.2 SDK</i>, available at the following location, provides instructions for using WebLogic JRockit SDK and JVM on the Windows and Linux platforms: http://e-docs.bea.com/wljrockit/docs142/userguide/index.html</p> <p><i>Tuning WebLogic JRockit 1.4.2 JVM</i>, available at the following location, explains how to tune the JRockit JVM to optimize your application's performance: http://e-docs.bea.com/wljrockit/docs142/tuning/index.html</p>
Managing the WebLogic Platform Environment	
Configuring the platform environment and applications	<p><i>Configuring and Managing WebLogic Server</i>, available at the following location, which provides links to instructions for starting, configuring, and monitoring WebLogic Server. http://e-docs.bea.com/wls/docs81/adminguide/index.html</p> <p><i>Creating WebLogic Configurations Using the Configuration Wizard</i>, available at the following location, which explains how to create and extend WebLogic domains, and also how to use the Template Builder to create templates and template extensions from existing domains. http://e-docs.bea.com/platform/docs81/configwiz/index.html</p> <p>The following topics are available from the WebLogic Server System Administration Web page, available at the following location: http://e-docs.bea.com/wls/docs81/admin.html</p> <ul style="list-style-type: none"> • Overview of WebLogic Server System Administration • Configuring Web Server Functionality for WebLogic Server • Using WebLogic Server Clusters

Table 1 WebLogic Platform Topics (Continued)

To learn about . . .	See the following . . .
Security	<p><i>Security in WebLogic Platform 8.1</i>, available at the following location, gives an overview of the WebLogic Platform security service and provides links to instructions for configuring, programming, and managing security.</p> <p>http://e-docs.bea.com/platform/docs81/secintro/secure.html</p> <p>See also the WebLogic Server Security topic page, available at the following location, which provides a list of topics with information about configuring, managing, and programming WebLogic Server security.</p> <p>http://e-docs.bea.com/wls/docs81/security.html</p>
Fault isolation and troubleshooting	<p><i>WebLogic Server Performance and Tuning</i>, available at the following location, which describes how to tune all the components of your WebLogic Server environment for optimal performance. These components include the WebLogic Server software itself, applications that run on WebLogic Server, the JVM, the operating system, and the network.</p> <p>http://e-docs.bea.com/wls/docs81/perform/index.html</p>
Extending the WebLogic Platform Environment	
BEA Tuxedo	<p><i>WebLogic Tuxedo Connector Administration Guide</i>, available at the following location, which describes how to configure and administer the WebLogic Tuxedo Connector, which enables WebLogic Server applications and Tuxedo services to interoperate.</p> <p>http://e-docs.bea.com/wls/docs81/wtc_admin/index.html</p>
Adapters for integrating enterprise applications into WebLogic Platform	<p>BEA WebLogic Adapters 8.1 online documentation, available at the following location, a set of online manuals for all the BEA adapters for enterprise application integration.</p> <p>http://e-docs.bea.com/wladapters/docs81/index.html</p>
BEA Liquid Data for WebLogic	<p>BEA Liquid Data for WebLogic online documentation, available at the following location, a set of online manuals that describe the product overview, the Liquid Data Avitek sample, administration, building queries and data views, and more.</p> <p>http://e-docs.bea.com/liquiddata/docs81/index.html</p>

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 BEA 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

