



BEA WebLogic Integration™

**Introducing BEA
WebLogic Integration**

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 JRockit, 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 AquaLogic, BEA AquaLogic Data Services Platform, BEA AquaLogic Enterprise Security, BEA AquaLogic Service Bus, BEA AquaLogic Service Registry, 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

Introduction to WebLogic Integration

| | |
|---|----|
| Topics Included in this Section | 1 |
| Integration Component of WebLogic Platform | 2 |
| Benefits of a Common Application Framework | 4 |
| Unified Approach to Enterprise Integration. | 6 |
| Interoperability of WebLogic Integration 8.5 SP5 with AquaLogic Service Bus | 7 |
| Features of BEA WebLogic Integration | 8 |
| Business Process Management | 10 |
| Synchronous and Asynchronous Business Processes | 13 |
| Stateless and Stateful Business Processes | 14 |
| Business Process Calls. | 14 |
| Importing and Exporting BPEL Processes | 14 |
| Web Services Available as Business Process Resources | 15 |
| Data Transformation | 16 |
| Message Broker | 19 |
| WebLogic Integration Controls | 21 |
| Application View Control | 22 |
| Dynamic Transformation Control | 22 |
| ebXML Control | 22 |
| Email Control | 23 |
| File Control | 23 |
| Http Control | 23 |

| | |
|--|----|
| MB Publish and MB Subscription Controls | 24 |
| MQSeries Control | 24 |
| Process Control | 26 |
| RosettaNet Control | 26 |
| Service Broker Control | 26 |
| Task and Task Worker Controls | 27 |
| TIBCO RV Control | 27 |
| TPM Control | 27 |
| WLI JMS Control | 28 |
| XML MetaData Cache Control | 28 |
| Worklist System | 28 |
| Trading Partner Integration | 31 |
| Application Integration and Adapters | 34 |
| Administration and Management | 36 |
| Business Process Instance Monitoring and Process Configuration | 39 |
| System Configuration | 40 |
| Event Generator Management | 41 |
| XML MetaData Cache Management | 44 |
| Related Documents | 44 |

Index

Introduction to WebLogic Integration

BEA WebLogic Integration™ 8.5 Service Pack 5 (SP5) is a unified solution for integrating business systems within an enterprise. WebLogic Integration provides a development and run-time framework that unifies all the components of business integration into a single flexible environment. The components include: business process management, data transformation, trading partner integration, connectivity, message brokering, application monitoring, and user interaction.

WebLogic Integration combines divergent components of the business integration picture, including ERP, CRM, legacy applications, business users, supply chains, and trading partners. It provides a versatile development environment for rapid business integration with simplified production and management of these components.

Topics Included in this Section

Integration Component of WebLogic Platform™

Describes how the WebLogic Integration product is integrated as a component of the WebLogic Platform.

Unified Approach to Enterprise Integration

Describes how WebLogic Integration can help solve your enterprise integration needs.

Interoperability of WebLogic Integration 8.5 SP5 with AquaLogic Service Bus™

Describes the interoperability between WebLogic Integration and AquaLogic Service Bus.

Features of BEA WebLogic Integration

Describes the various WebLogic Integration components that help build complex integrated solutions.

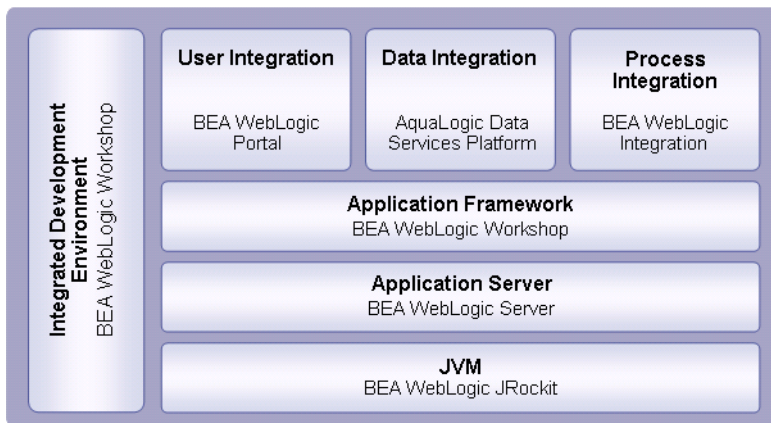
Integration Component of WebLogic Platform

WebLogic Integration is a **component** of the WebLogic Platform that provides functionality for businesses to develop new applications, integrate applications with existing systems, streamline business processes, and extend e-business infrastructure through portal gateways.

WebLogic Integration provides a single environment to build an integrated enterprise application. Be it business process integration (from business process modeling to integrating enterprise adapters), custom application development using robust Web services and controls, or developing a portal to provide employees, partners, and customers with an integrated view of applications and data.

As shown in the following diagram, the WebLogic Platform comprises multiple components that you can use independently, or in combination, as required for your application.

Figure 1 WebLogic Platform



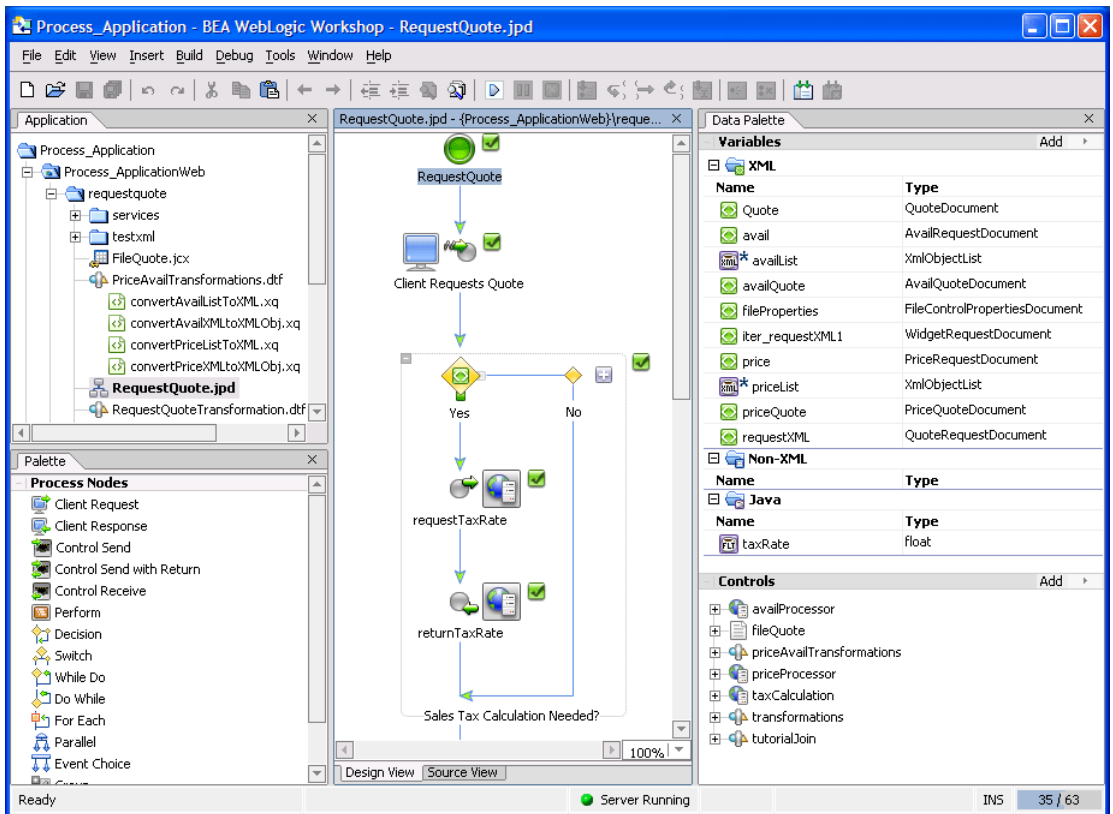
BEA WebLogic Server[®], the industry-leading J2EE application server, provides the critical infrastructure needed to develop integrated solutions which include security, transaction management, fault tolerance, persistence, and clustering.

Leveraging WebLogic Server as the underlying deployment environment, WebLogic Integration uses Web services to integrate distributed systems inside and outside an organization. WebLogic

Integration then uses the BEA WebLogic Workshop® framework to simplify application development.

As a seamless component of the WebLogic Workshop environment, WebLogic Integration uses all the available resources for extending integration applications. As shown in Figure 2, the graphical tools used to edit business processes are available in the same WebLogic Workshop environment as the controls, Web services, and portal-building tools.

Figure 2 WebLogic Integration Business Process in the WebLogic Workshop IDE



Benefits of a Common Application Framework

While WebLogic Integration provides all the elements of application integration, your integration project will encompass more. You will have to write application logic and expose business processes to the user. You will need to customize the user interface. You might also have to build a web-based application on top of your application. This is where the new unified platform provides an ease-of-use advantage. Given the same framework and the same resource interfaces, you can use Web services, access the J2EE layer for specialized application logic, and use NetUI and Portal resources to allow your user to interact with the business process. Once you build the integration application, you can continue to stay within the interface design environment (IDE) to build the user interface. You can use the JSP editor to create forms for data entry and use Page groups to orchestrate the flow of information across multiple Web pages. You can use Portal to host your Web interface and customize the user experience.

The WebLogic Workshop framework for development, integration and portal applications includes the following benefits:

Table 1 Benefits of a Common Framework

| Feature | Benefit |
|---------------------------|---|
| Unified programming model | <ul style="list-style-type: none"> • Event-driven programming model based on procedural logic development • Control-based environment for constructing processes and abstracting resources • Abstracts low-level technical details of J2EE APIs and back-end resources |
| Common look-and-feel | <ul style="list-style-type: none"> • Unified visual development model across WebLogic Workshop, WebLogic Integration, and WebLogic Portal® components |
| Annotated Java code model | <ul style="list-style-type: none"> • Lets you focus on behavior, event handling and method invocation instead of writing complex code • Provides meta data-driven application construction |

Table 1 Benefits of a Common Framework

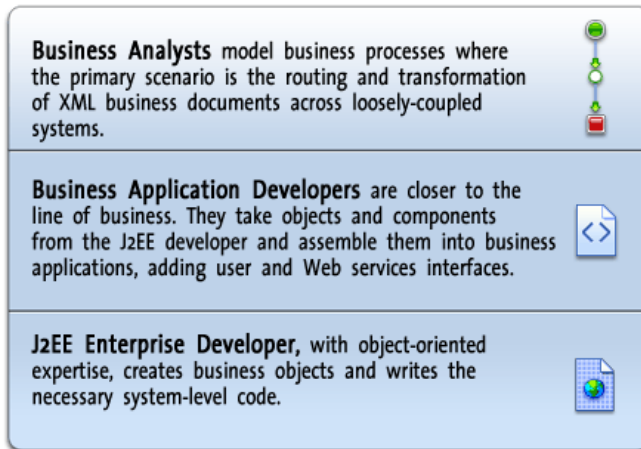
| Feature | Benefit |
|---|---|
| Web services | <ul style="list-style-type: none"> • Natively built, extensible and integrated at the enterprise level • Adheres to Web services standards, such as Simple Object Access Protocol (SOAP) and Web Services Description Language (WSDL) • Implemented processes automatically accessible as Web services • Exposes processes as Web services and invokes internal and third-party Web services from IDE components • Accessible using XML messages |
| Common project and deployment model | <ul style="list-style-type: none"> • Application encapsulation through J2EE mechanisms like WAR and EAR files |
| Controls | |
| Simple visual components | <ul style="list-style-type: none"> • Easy-to-use Workshop environment GUI • Behavior defined through methods, events, and properties |
| Extensible architecture | <ul style="list-style-type: none"> • Application components built in WebLogic Workshop automatically become controls that can be reused anywhere within the environment • Developers and ISVs can develop custom controls |
| Consistent mechanism for representing resources | <ul style="list-style-type: none"> • All resources look the same—abstracts resource specific details • Reduces development learning curve |
| Composition | <ul style="list-style-type: none"> • Controls can invoke other controls |
| Java component with Java methods | <ul style="list-style-type: none"> • Easier access to J2EE resources |
| Complete access to J2EE API | <ul style="list-style-type: none"> • J2EE developers can build the logic at the J2EE layer, package it as a control, and make it available to the application developer or integration specialist |

Unified Approach to Enterprise Integration

Today, businesses operate in a diverse environment. They interact with a variety of clients, both inside and outside the enterprise, and rely on disparate systems and processes to power their business activities. In this kind of environment, businesses face an integration challenge. To fully maximize their resources, businesses strive to bring together their internal systems and processes to gain operational efficiency, and they strive to extend those systems to increase revenue. Gaps exist between business integration needs and the tools available to fulfill IT requirements.

Traditional business process management tools are targeted at helping the business analysts paint a high-level picture of integration. However, implementing that picture requires staff with scarce and expensive specialized knowledge of proprietary integration and deployment environments. WebLogic Integration provides the tools to create the visual models for an integration solution in the same highly productive environment that the architect and developer use to implement the solution. [Figure 3](#) illustrates the layers in an IT organization that need to communicate to build an effective integration solution.

Figure 3 WebLogic Integration Unifies the Different IT Layers Needed to Build an Integration Solution



By providing access to enterprise resources such as messaging, adapters, and integration controls (coupled to business process modeling and data transformation), the WebLogic Integration environment equips analysts and developers, who need not have specialized knowledge of the deployment environment, with a means to quickly implement and bind business processes to IT resources.

Within the WebLogic Workshop framework, WebLogic Integration and its robust Web services and controls architecture support a business process layer of abstraction and a common language for requirements gathering, validating implementation, and monitoring run-time execution. By straddling development and integration environments, WebLogic Integration cuts through the accumulation of proprietary integration technologies, making the integration effort easier and less costly.

WebLogic Integration optimizes enterprise integration by recognizing and reflecting the following design principles:

- **Loosely-coupled Integrations:** Easier to maintain than traditional tight and rigid integrations
- **Asynchronous Communication:** Critical to conducting and protecting business operations when communication goes down
- **Coarse-grained Communication:** Key to maximizing the efficiency of typically high cost communication between loosely coupled systems

WebLogic Integration enables the reuse of technical skills across the entire lifecycle of building, integrating, and deploying applications. The following sections outline the key WebLogic Integration features available in the WebLogic Workshop environment.

Interoperability of WebLogic Integration 8.5 SP5 with AquaLogic Service Bus

AquaLogic Service Bus is an enterprise infrastructure software. It is a key product in the family of configuration-driven infrastructure software, called Service Infrastructure, targeted at the implementation, deployment and on-going operations of service oriented architectures (SOA).

AquaLogic Service Bus brings together the functionality of an Enterprise Service Bus with service management functionality in a single unified product targeted at deploying SOA with configuration-driven integration of services and applications. This converged approach delivers a scalable, intelligent messaging, routing and transformation layer supporting heterogeneous end-points, integrated with capabilities for service registration, monitoring, and lifecycle management.

AquaLogic Service Bus supports the following message types: SOAP 1.1, JMS with headers, IBM WebSphere[®] MQ with native headers, Un-enveloped XML, Raw Data, and SOAP with attachments. AquaLogic Service Bus supports the following transport protocols: HTTP, HTTPS,

HTTP/SOAP, JMS, FTP, SMTP, and File based protocols. MQ is supported via foreign JMS provider.

AquaLogic Service Bus integrates seamlessly with BEA products - WebLogic Server, WebLogic Integration, WebLogic Portal, AquaLogic Data Services Platform, and WebLogic Adapters - for service creation, consumption and orchestration.

WebLogic Integration 8.5 SP5 can be leveraged by AquaLogic Service Bus as a service endpoint. WebLogic Integration business processes and composite applications can act as business services (service providers) or as service clients (service consumers). Configuring WebLogic Integration as a service client can be done by importing the service client WSDL from AquaLogic Service Bus into WebLogic Integration, generating a JWS control, and adding a node in the JPD to invoke the service. Exposing WebLogic Integration to AquaLogic Service Bus can be done by generating the WSDL file from the existing JPD and registering with AquaLogic Service Bus as a business service. WebLogic Integration can also interface via FTP, Email, or File event generators or controls. These services are registered in AquaLogic Service Bus.

For more information, see the AquaLogic Service Bus 2.0 documentation available at the following URL:

<http://e-docs.bea.com/alsb/docs20/index.html>

Features of BEA WebLogic Integration

WebLogic Integration provides a set of enhanced enterprise integration capabilities within the Workshop environment, for improved developer productivity. Following is a list of design and run-time integration features provided by WebLogic Integration:

Table 1 Features of WebLogic Integration

| Feature | Description |
|---|---|
| Business Process Management | Enables users to model business process scenarios and implement integration and automation in terms of business goals. |
| Data Transformation | Transforms data from one format to another using simple GUI drag-and-drop feature. Allows any-to-any data transformation of incoming and outgoing data (in any combination of structured XML, non-XML, or Java data), transformation within a process flow, multiple input sources, and complex operations such as joins, unions, and grouping by key fields. |

Table 1 Features of WebLogic Integration

| Feature | Description |
|---|--|
| Message Broker | Provides rules-based message routing; using a channels-based publish-subscribe broker to transport events in a loosely-coupled manner. Enables high-performance and low latency message routing between applications. |
| WebLogic Integration Controls | Enables non-experts to achieve rapid integration results by dragging and dropping simple component interfaces that represent the resources being integrated. Over a dozen pre-built controls are available out-of-the-box for access to database, file, HTTP, messaging, service broker, and human interaction with enterprise resources. |
| Worklist System | Provides controls-based functionality that allows interaction from end users, such as task creators, task workers, and task administrators, to business processes for handling process exceptions, approvals, status tracking and so forth. Includes centralized user and group management and user rules and authorization for secure participation within processes. |
| Trading Partner Integration | <p>Enables rapid, secure online-connection with suppliers and customers through leading standard protocols such as RosettaNet and ebXML. These protocols provide:</p> <ul style="list-style-type: none"> • secure messaging, digital signatures and encryption • recoverable and trackable messages • dynamic configuration updates <p>Accommodates a full range of partners—from full Hub to Hub interaction and lightweight partner client (BEA WebLogic Integration - Business Connect) to zero weight client access via portal, browser, or FTP access.</p> <p>Manages trading partner profiles with streamlined import and export of configurations.</p> |

Table 1 Features of WebLogic Integration

| Feature | Description |
|--|--|
| Application Integration and Adapters | WebLogic Integration provides access from WebLogic Workshop controls to the Application Integration Design Console, Application Views, pre-built BEA WebLogic Adapters, and custom adapters. |
| Administration and Management | <p>Provides administrators full and secure visibility into the distributed integration environment using WebLogic Integration Administration Console. The console provides integration-focused lifecycle management of:</p> <ul style="list-style-type: none"> • business processes • deployed applications • message broker traffic • enterprise adapter health and usage • trading partner activity and parameters • roles assigned to various users and groups • worklists |

The following sections provide more information about WebLogic Integration features available in the WebLogic Workshop environment.

Business Process Management

WebLogic Integration business process management (BPM) allows the user to model and execute business processes that span multiple internal systems, external resources, and users. From the BPM perspective, the enterprise is a set of business services that are accessed through controls and can be orchestrated to model a business process. WebLogic Integration supports synchronous and asynchronous communications, and stateless and stateful processes.

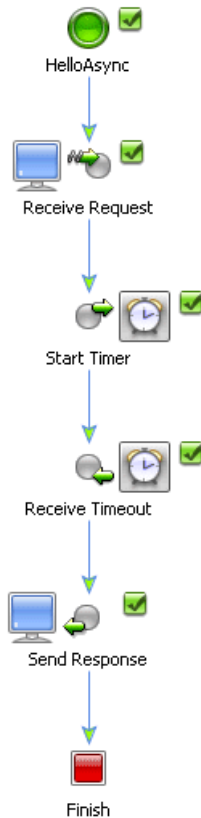
The BPM functionality of WebLogic Integration enables corporate developers to develop, run, and maintain complex e-business processes that integrate existing enterprise systems, cross-enterprise applications, and human decision makers. The following table lists the key features of WebLogic Integration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

Table 2 Business Process Management Features

| Key Feature | Description |
|---|---|
| Unified access to resources through controls | <ul style="list-style-type: none"> • View business activities as services and model business process to orchestrate • Business process seamlessly interacts with users, applications, back-end resources, and resources inside and outside the firewall |
| New simplified structured business processes | <ul style="list-style-type: none"> • XML for flow • Java for operations |
| Graphical business process editing for high-level integration scenarios | <ul style="list-style-type: none"> • Message-based, transformation routing • Business processes are Java classes. Business process (JPD) files also contain the metadata that describes the business process logic. • Two-way editing in Source View and Design View |
| Support for Java code in business process nodes | <ul style="list-style-type: none"> • One click away from Java code |
| Process implementation optimization for performance | Support for: <ul style="list-style-type: none"> • stateless synchronous processes • stateless asynchronous processes • stateful asynchronous processes |

The business process engine offers flexibility to create business processes graphically, allowing you to focus on the application logic rather than on implementation details. [Figure 4](#) illustrates a graphical representation of a business process.

Figure 4 Graphical Representation of Business Process



As you design business processes in WebLogic Workshop using the graphical tools in Design View, WebLogic Workshop writes source code to a business process file (a JPD file). You can view and, if required, edit the code in Source View. WebLogic Workshop is robust and enables you to edit and view large JPDs, with greater than 50 nodes defined in the process. To minimize code mishaps, protected sections are clearly indicated in the Source View. If you try to modify or remove any of the protected sections, warnings are displayed in the Design View.

Using node builders, you can create, open, and edit transformation maps. Node builders can be opened from within the IDE.

For more information, see

- [Tutorial: Building Your First Business Process](#), available in the WebLogic Workshop Help at:

<http://edocs.bea.com/workshop/docs81/doc/en/integration/tutorial/tutWLIProcessIntro.html>

- *Guide to Building Business Processes*, available in the WebLogic Workshop online help at:
<http://edocs.bea.com/workshop/docs81/doc/en/integration/wfguide/wfguideIntro.html>

Synchronous and Asynchronous Business Processes

A business process can be synchronous or asynchronous, based on the method that is used to invoke the process.

A synchronous process is invoked by a synchronous method, which is a Client Request with Return node. So, the synchronous process returns a response to the client after the process is executed. A synchronous business process can also contain asynchronous operations, but these must be added after the starting event in the process flow. That is, at run time, the asynchronous processes are executed after the synchronous starting event is completed.

An asynchronous process is invoked by an asynchronous method. In other words, the starting event in an asynchronous process is represented by an Asynchronous node. This includes business processes that are invoked via a Client Request node, an Asynchronous Subscription node, or one of several Client Request or Subscription nodes (that is, an Event Choice node). An asynchronous process can call synchronous or asynchronous methods without additional configuration.

You can also enable synchronous clients to interact with business processes that have asynchronous interactions with resources. For example, a synchronous WebLogic Workshop client, such as a JSP or Portal page that uses a Java control, might need to invoke a business process and then block. While the client is blocking, the business process may perform asynchronous activities, such as enqueueing a JMS message and waiting for a JMS receive, and then return the response to the client, after which the client unblocks.

To enable synchronous clients to interact with business processes that have asynchronous interactions with resources, you can create a business process with a Client Request node with an attribute property called `sync/async callback name`. This Client Request node property holds the name of the callback method used by the associated Client Response node. The Client Request and Client Response nodes delineate the activities (including asynchronous activities) that occur while the client is blocking. After setting this property, instead of generating a synchronous WSDL file, you need to use the Sync-Async Control to generate a sync-to-async WSDL file for the process.

Stateless and Stateful Business Processes

Business processes can either be stateless or stateful. Typically, stateless business processes provide high performance because there is no overhead of maintaining state in a database through the transaction.

For a stateful business process, the state of the process is persisted in a database. Stateful business processes are used when the requirement for data reliability and recovery is high. However, since state is persisted, there could be an impact on the performance of the process. To minimize the adverse impact on performance, non-persistent stateful business processes can be defined at design-time. When you set the properties of the business process, three types of persistence can be defined. The persistence types are:

- **Always** where state is always persisted in a database
- **Never** where state is stored only in memory (and never in a database) and lasts only for the current session
- **On overflow** where state is persisted only after a number (specified in the Max Bean in Cache deployment descriptor file) is reached

Business Process Calls

Business processes can expose their functionality to clients in several ways, including through WSDL files, Process controls, Service Broker controls, and JPD Proxies. Process controls and Service Broker controls can only be used between Workshop components. The Workshop component may be a Web service or a business process that is deployed on the same domain or different domain. Any Java client, including standalone Java applications, EJBs, JSPs and Servlets, can communicate with any business process using JPD proxies. When the clients use JPD proxies, they actually make Java method calls using Remote Method Invocation (RMI). In other words, a JPD proxy is a RMI client to a business process.

Importing and Exporting BPEL Processes

Business Process Execution Language for Web services (BPEL) is a Web service orchestration standard that can be used to define transactional business processes. With BPEL, activities that are a part of a business process can be expressed as a Web service. These services can then be orchestrated to ensure control over the entire process. Processes written in BPEL are stored as `.bpe1` files and can be run on any platform or product that is compliant with the BPEL specification.

BPEL 1.1 is a draft specification proposed by BEA, IBM, and Microsoft. However, this specification will be superseded by BPEL 2.0, which is expected to be formalized as a standard under the aegis of OASIS.

BPEL Import and Export tools are provided to enable design-time interoperability with other tools that support the BPEL 1.1 specification. However, in certain cases, run-time semantics are not guaranteed, especially when there are functional mismatches between JPD and BPEL, or between various expression languages including differences between XQuery, XPath, and XSLT. Run-time semantics are also not guaranteed when they involve vendor extensions, external artifacts, or environment settings. For the above reasons, the imported and exported files should be reviewed and modified, based on requirements, to make sure that they run properly.

For more information, see *BPEL Import and Export User Guide* available at: <http://e-docs.bea.com/wli/docs85/bpel/index.html>.

BPEL Import Tool

You can use the BPEL Import tool to import a BPEL 1.1 compliant file into a JPD file, where it can be used in the WebLogic Workshop design environment. While the main orchestration logic of the BPEL file is imported into a JPD file, it is not expected that the imported JPD file will be immediately executable in WebLogic Workshop. You will need to manipulate the JPD file in WebLogic Workshop to get the imported process to run.

BPEL Export Tool

You can use the BPEL Export tool to export the semantics of a JPD file into BPEL where it can be used in a BPEL-compatible design environment. BPEL code exported using the BPEL Export tool is BPEL 1.1 compliant and can be used in design environments compliant with BPEL 1.1. While the main orchestration logic of the JPD is exported to BPEL, it is not expected that the exported BPEL file will be immediately executable in the target environment. You will need to manipulate the BPEL file in the target environment to get the exported process to run, or to get close to the run-time semantics.

Web Services Available as Business Process Resources

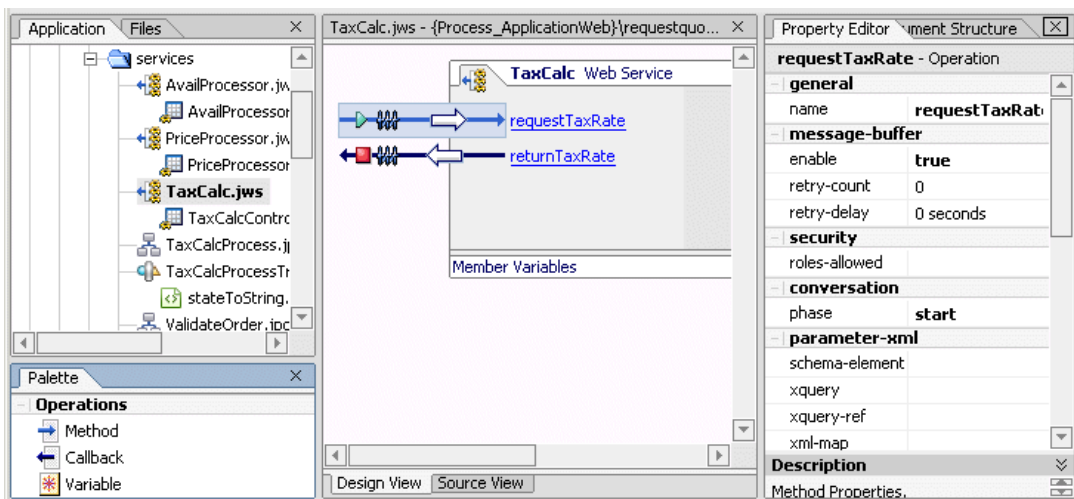
WebLogic Integration leverages Web services, asynchronous communication, and XML messaging at the platform level. At this level, these services can be used across internal and external integrations, giving application developers the tools to simplify development and integration of loosely coupled and asynchronous applications.

WebLogic Integration provides native support for Web services including Web service security and reliable messaging. Web services can be invoked from within a WebLogic Integration business process and business processes can be exposed as a Web Service. They can be made available as resources to other applications and application components. To learn more, see “Web Service Features in Business Processes” in the *Annotations Reference* located at the following URL:

<http://e-docs.bea.com/workshop/docs81/doc/en/integration/javadoc-tag/jpd/webservicefeatures.html>

The following figures illustrates a Web service invoked from a business process.

Figure 5 Web Services Available as Resources to Integration



Data Transformation

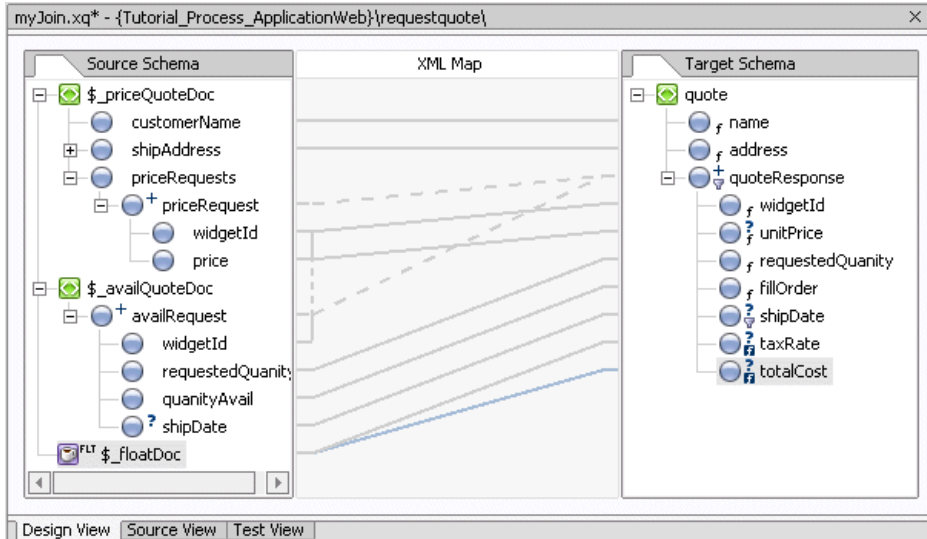
Data transformation enables you to translate between XML, non-XML, and Java data formats allowing you to rapidly integrate heterogeneous applications regardless of the format used to represent data. The data transformation functionality is available through control and data transformations that can be packaged as controls and reused across multiple business processes and applications.

In WebLogic Workshop business processes, XML data can be transformed using either XQuery expressions or eXtensible Stylesheet Language Transformations (XSLTs). While WebLogic

Integration provides functionality for executing existing XSLTs in business processes, it also offers a new and easier path to data transformation through XQuery.

WebLogic Integration features a powerful visual data mapping tool, **XQuery Transformation Mapper**, as shown in [Figure 6](#). The Mapper gives you the ability to generate complex transformations easily using drag-and-drop operations. The mapper functionality of WebLogic Workshop enables the conversion of data of different types. For example, XML data can be transformed from an XML document valid to one XML Schema to another XML document valid to a different XML Schema.

Figure 6 XQuery Transformation Data Mapper



The following table lists the key features of data transformation.

Table 1 Data Transformation Features

| Feature | Description |
|-------------------------------------|--|
| Data transformation | <ul style="list-style-type: none"> • Transformations packaged as controls that can be treated as resources and reused across multiple processes and integration solutions. • Transforms data between any of the following input-output data types: XML data, Non-XML data, Java Primitives, and Java classes. • Allows multiple input sources to a transformation. • Supports complex relations and constraints including joins, unions, and grouping by key fields. • Enables transformation of XML grammars |
| Integration with business processes | <ul style="list-style-type: none"> • Enables the transformation of data in a business process using transformations written in XQuery or XSLT languages. • Transforms data received as an incoming message into the business process. • Transforms data before the business process sends an outgoing message. • Transforms data inside the business process. |
| XQuery Transformation Mapper | <ul style="list-style-type: none"> • Provides visual modeling tool for transformation between any combination of XML, Non-XML, and Java data formats. • Enables the visual transformation of data from one format to another through drag-and-drop mechanism. Functions and parameters can also be dragged and dropped. • Enables adding of complex conditional expressions to transformations. • Supports nested If-Then-Else expressions • Allows data conversion from a non-repeating source element to a repeating target group and from a repeating source group to a non-repeating target element. • Supports the use of recursive schemas in transformations. |

Table 1 Data Transformation Features (Continued)

| Feature | Description |
|------------------------------|---|
| Data transformation tutorial | Step-by-step tutorial that illustrates the use of business process actions for data transformation. |
| Format Builder tool | Enables creation of metadata to describe non-XML data. |

For more information, see:

- [Tutorial: Building Your First Data Transformation](#), available in the WebLogic Workshop Help at:

<http://edocs.bea.com/workshop/docs81/doc/en/integration/dttutorial/tutWLIDataTransIntro.html>

- [Guide to Data Transformation](#), available in the WebLogic Workshop Help at:

<http://edocs.bea.com/workshop/docs81/doc/en/integration/dtguide/dtguideIntro.html>

Message Broker

WebLogic Integration implements a Message Broker that provides business processes with a channels-based publish and subscribe communication mechanism. This enables business processes to communicate in a loosely-coupled, anonymous manner using a business-naming paradigm. For example, a Purchase Order routing process subscribes to the *New Order Entered* channel and as each new order message is published to that channel, the process is activated. Each business process can specify the channels to which it publishes and subscribes.

Publishers can broadcast messages without knowing who is going to receive these messages. The consumers of these messages can be any one of the different types of listeners. Consumers, such as business processes and other back-end resources, can subscribe to Message Broker channels. At run-time, you can add new publishers and new subscribers.

WebLogic Integration supports the ability to start a business process as a result of receiving a synchronous message from a Message Broker channel. A synchronous subscription start causes the subscribed business process to run in the same transaction as the publisher.

Message Broker supports Event Generators that can publish events from external sources to Message Broker channels. WebLogic Integration supports Email, File, HTTP, JMS, MQSeries RDBMS, TIBCO RV, and Timer event generators. WebLogic Integration adapters, hosted in the Application Integration framework, publish events from packaged applications to channels.

Table 2 Message Broker Features

| Feature | Description |
|-------------------------|---|
| Publish and Subscribe | <ul style="list-style-type: none"> • High-performance throughput • Channels via event generators—business processes dynamically subscribe to Message Broker channels via controls. Business processes can also specify static subscriptions to channels at the Start node. In this case, a business process is started when it receives a message from a Message Broker channel to which it is subscribed. • Subscriptions can start a new business process or be used within a running business process to block a message. • XQuery filters to refine message selection on subscription |
| Message Broker Channels | <ul style="list-style-type: none"> • Static and dynamic binding to channels • Channel typing explicitly defines the structure of the message that the channel can route. • Defined channel-naming hierarchy • Event generators publish messages to channels from external sources. • Dynamically bind event generators to channels at run time using WebLogic Integration Administration Console. • WebLogic adapters can publish events to channels. |
| Message Routing | <ul style="list-style-type: none"> • Routing from event generators to stateless business processes • Single transaction (No need to save state) |
| Rules | <ul style="list-style-type: none"> • Business process logic defines trigger, transformation, and routing rules. • Short-running rules not long-running processes • Stateless business processes implement <i>rules</i> |

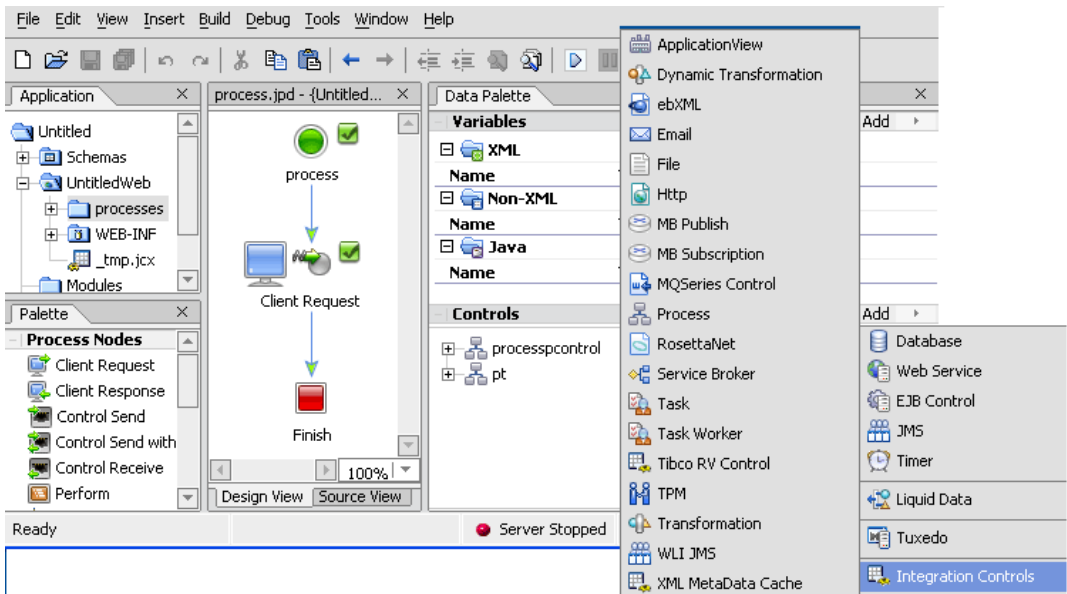
For more information, see “Message Broker” in *Managing WebLogic Integration Solutions*, available at <http://e-docs.bea.com/wli/docs85/manage/msgbroker.html>.

WebLogic Integration Controls

WebLogic Integration provides a set of out-of-the-box controls that enable you to start integration projects with a portfolio of resources. Figure 7 shows a sample menu of controls. These WebLogic Integration controls provide access to external resources to enhance developer productivity. WebLogic Integration applications also have access to external resources through the WebLogic Workshop high-level controls. The WebLogic Workshop framework provides a consistent mechanism for interacting with resources across all Workshop, Integration, and Portal components.

For more information, see *Using Integration Controls* available at <http://e-docs.bea.com/workshop/docs81/doc/en/integration/controls/controlsIntro.html>.

Figure 7 Out-of-the-Box Controls



WebLogic Integration controls provide access to Web services and to any J2EE resources such as JMS, EJB, and Databases (JDBC).

Note: Integration controls are available in WebLogic Workshop only if you are licensed to use WebLogic Integration.

The following sections describe the available WebLogic Integration controls.

Application View Control

The Application View control allows Web Services or business processes to access an enterprise application using an Application View. This control is designed to make it easy for you to use an existing, deployed application view from within your business process. The Application View must be created using Application Integration Design Console before it can be referenced using an Application View control. For more information, see “[Application Integration and Adapters](#)” on page 34. For more information about the Application View control, see:

<http://e-docs.bea.com/workshop/docs81/doc/en/integration/controls/controlsAppView.html>

Dynamic Transformation Control

The Dynamic Transformation control provides a business process with the ability to dynamically select which transformation is invoked at run-time. So, the format of the input and output data need not to be specified during business process design-time. Instead, the business process identifies the input data and transforms it to the desired output format during run-time.

The Dynamic Transformation control uses transformations that have already been created, such as those transformations created with the Transformation control. You need to make sure that the required transformation has been defined and tested before you create the Dynamic Transformation control. This control has different base methods that have been defined for various transformation types like XQuery, XSL, and MFL. Custom methods for executing different types of transformations can also be created. Using these methods, the transformation type is identified and executed at run-time. For more information, see:

<http://e-docs.bea.com/workshop/docs81/doc/en/integration/controls/controlsDynamicTrans.html>

ebXML Control

The ebXML protocol (Electronic Business using eXtensible Markup Language) is a modular suite of specifications that enables enterprises of any size and in any geographical location to conduct business over the Internet. It is sponsored by UN/CEFACT and OASIS. To learn about ebXML, see <http://www.ebXML.org>.

The ebXML control enables WebLogic Workshop business processes to exchange business messages and data with trading partners by using the ebXML protocol. This control supports both the ebXML 1.0 and ebXML 2.0 messaging services. For more information, see “[Trading Partner Integration](#)” on page 31. For more information about the ebXML control, see:

<http://e-docs.bea.com/workshop/docs81/doc/en/integration/controls/controlsebXML.html>

Email Control

The Email control enables WebLogic Integration business processes to send e-mail to a specific destination. To receive e-mail, you must use the Email event generator. The event generator then publishes incoming e-mails to Message Broker channels.

You use WebLogic Integration Administration Console to create and manage event generators. For more information, see “[Event Generator Management](#)” on page 41.

For more information about the Email control, see:

<http://e-docs.bea.com/workshop/docs81/doc/en/integration/controls/controlsEmail.html>

File Control

A File control is used to perform operations on a file. It allows business processes to read, write, or append to a file in a file system. In addition, the File control supports file manipulation operations such as copy, rename, and delete. You can also retrieve a list of the files stored in a specific directory. The files can be one of the following types: XmlObject, RawData (binary), or String. The methods available to a File control depend on the type of data contained in the file. For a file of type String or XmlObject, you can specify the character set encoding. When processing a file, you can specify the delimiter for a file as the record size. If no delimiter is specified, the file is processed one line at a time. In the case of files of type String, you can also specify number of bytes or any character as a delimiter. For more information, see:

<http://e-docs.bea.com/workshop/docs81/doc/en/integration/controls/controlsFile.html>

Http Control

HTTP protocol is used for communication between a client and a server. The Http control is used to provide outgoing HTTP access to WebLogic Workshop clients. This control can be used to work with HTTP requests and process responses. The Http control supports the Get and Post request methods for data transfer. By using the Get mode, you can send your business data along with the URL. By using the Post mode, you can send large amount of data like Binary, XML and String documents to the server within the body of the request. You can specify Http control properties in an annotation, or pass dynamic properties via an XML variable. Using this control, you can send an HTTP or HTTPS (Secure HTTP) request to a URL and receive specific HTTP response header and body data. For more information, see:

<http://e-docs.bea.com/workshop/docs81/doc/en/integration/controls/controlsHTTP.html>

MB Publish and MB Subscription Controls

The Message Broker resource provides a publish and subscribe message-based communication model for WebLogic Integration business processes, and includes a powerful message filtering capability. For more information, see “[Message Broker](#)” on page 19.

Two Message Broker controls, MB Publish and MB Subscription, are available, in WebLogic Workshop, to your business processes. The Publish control is used to publish messages to Message Broker channels. You bind the Message Broker channel to the Publish control when you declare the control, but it can be overridden dynamically. You can add additional methods to your extension (subclass) of the Message Broker Publish control.

The Subscription control is used to dynamically subscribe to channels and receive messages. You bind the channel and optionally, an XQuery expression for filtering messages, when you create an instance of the control for your business process. The bindings cannot be overridden dynamically. The Subscription control interface includes methods that allow your business process to subscribe to and unsubscribe from the bound Message Broker channel.

In addition to the dynamic subscriptions you design at Control nodes in your business process, you can design static subscriptions at Start nodes to receive messages from Message Broker channels. That is, a business process can be initiated by subscribing synchronously or asynchronously to a Message Broker channel and starting via an event.

For more information about the Message Broker controls, see:

<http://e-docs.bea.com/workshop/docs81/doc/en/integration/controls/controlsBroker.html>

MQSeries Control

MQSeries is a messaging service queue provided by IBM that enables message transfer between applications. The sending application puts a message on a Queue, and the receiving application gets the message from the Queue.

The MQSeries control enables WebLogic Integration business processes to send and receive messages using MQSeries queues. Using the MQSeries control, you can send and receive Binary, XML, and String messages. You can specify MQSeries control properties while configuring the MQSeries control or dynamically at run-time. By default, the MQSeries control handles transactions implicitly for each PUT and GET method individually, without having to set an explicit transaction boundary. However, transaction boundaries can also be set explicitly.

Using SSL, you can enable both one way authentication (server-side) and two-way authentication (client-side).

For more information, see:

<http://e-docs.bea.com/workshop/docs81/doc/en/integration/controls/controlsMQ.html>

Disclaimer

In the WebLogic Integration product, BEA provides and supports MQSeries (now known as WebSphere® MQ) integration through the following options:

- *BEA WebLogic Integration JMS Control within the WebLogic Workshop IDE*
To learn more, see “[WLI JMS Control](#)” in *Using Integration Controls* available at <http://e-docs.bea.com/workshop/docs81/doc/en/integration/controls/controlsJMS.html>.
- *JMS Event Generators in BEA WebLogic Integration Administration Console*
To learn more, see “[Event Generators](#)” in *Managing WebLogic Integration Solutions* available at <http://e-docs.bea.com/wli/docs85/manage/evntgen.html>.
- *Async Sample illustrating the use of JMS with MQSeries*
To learn more, see “[Async Binary Update Sample](#)” available at http://e-docs.bea.com/wli/docs85/sol_samples/async_binary.html.
- *BEA WebLogic Integration MQSeries Control within the WebLogic Workshop IDE*
To learn more, see “[MQSeries Control](#)” in *Using Integration Controls* available at <http://e-docs.bea.com/workshop/docs81/doc/en/integration/controls/controlsMQ.html>.
- *BEA Adapter for MQSeries*
To learn more, see BEA [WebLogic Adapter for MQSeries 8.1 Documentation](#) available at <http://e-docs.bea.com/wladapters/mq/docs81/index.html>.

The WebLogic Integration Control and Adapter rely on use of the MQSeries APIs. IBM has not currently certified the MQSeries APIs for BEA WebLogic Integration 8.1 and 8.5 products. To resolve issues for any aspect of the MQSeries Control or Adapter that is not part of the BEA WebLogic Integration product or is specific to IBM WebSphere MQ, the APIs might require a workaround. JMS-based MQ integration, however, does not have the same support-related limitations as WebSphere MQ APIs. For a current list of BEA products that IBM has explicitly stated it supports in conjunction with WebSphere MQ, see the following URL:

http://www-306.ibm.com/software/integration/mqfamily/platforms/supported/wsmq_for_solari_s_5_3.html

Process Control

The Process control is used to send requests to, and receive callbacks from another business process. The Process control is typically used to call a sub process from a parent process. Process control invocations are Java Remote Method Invocation (RMI) calls.

You can create a Process control by inserting it from the list of Integration controls, or you can generate the Process control using the business process file (JPD file). When you insert the Process control, you can specify the target business process and the start method associated with that business process. Optionally, you can use the Query Builder to decide at run-time which one of multiple subprocesses can be called by the Process control. When you generate the Process control using the JPD file, a JCX file is automatically created for that control. Double-click this JCX file (as displayed in the Application tab) to view the control in the Design View, or drag it to the Data Palette and use any of the methods associated with this Process control.

For more information, see:

<http://e-docs.bea.com/workshop/docs81/doc/en/integration/controls/controlsProcess.html>

RosettaNet Control

RosettaNet is a business protocol that enables enterprises to conduct business over the Internet. To learn about RosettaNet, see <http://www.rosettanet.org>.

The RosettaNet control enables WebLogic Workshop business processes exchange business messages and data with trading partners using the RosettaNet protocol. You can use the RosettaNet control only in initiator business processes to manage the exchange of RosettaNet business messages with participants. The RosettaNet control supports RosettaNet version 1.1 and 2.0 of the implementation framework. For more information, see “Trading Partner Integration” on page 31.

For more information, see:

<http://e-docs.bea.com/workshop/docs81/doc/en/integration/controls/controlsRosettaNet.html>

Service Broker Control

The Service Broker control allows a business process to send requests to and receive callbacks from another business process, a Web service, or a Web service or business process defined in a Web Service Description Language (WSDL) file. The Service Broker control is an extension of the Web Service control. For more information, see:

<http://e-docs.bea.com/workshop/docs81/doc/en/integration/controls/controlsService.html>

Task and Task Worker Controls

The *Task control* and the *Task Worker control* are Worklist controls that enable you to introduce user assigned tasks and task management to WebLogic Integration, so you can build a Worklist system.

These Worklist controls enable the automated manipulation, creation, and management of Tasks. A Task instance represents a unit of work that requires completion within a certain time period. After the work is completed, you can use the Task instance to represent a detailed record of that unit of work. For more information, see “[Worklist System](#)” on page 28. For more information about the Worklist controls, see:

<http://e-docs.bea.com/workshop/docs81/doc/en/integration/controls/controlsWorklist.html>

TIBCO RV Control

TIBCO® Rendezvous™ is a messaging software provided by TIBCO. It enables exchange of data across applications running on distributed platforms. TIBCO RV control is a WebLogic Integration control that enables seamless connection to, and transfer of data with TIBCO Rendezvous using the Rendezvous daemon. It enables communication via many of the features provided by TIBCO Rendezvous, including Certified Message Delivery and Distributed Queue. The sending and receiving applications can be on multiple platforms, as long as the Rendezvous daemon is running on the host machine, or is remotely accessible to the host.

For more information, see *TIBCO Rendezvous Control and Event Generator User Guide*, available at: <http://e-docs.bea.com/wli/docs85/tibcorv/index.html>.

Disclaimer

Use of the TIBCO RV control and event generator with BEA WebLogic Integration in no manner confers or grants the right to use TIBCO Rendezvous including "dynamic libraries". In order to use such TIBCO products, the user of the TIBCO RV control and event generator must obtain a valid license from TIBCO. See <http://www.tibco.com> for information on how to obtain a licensed copy of Rendezvous.

TPM Control

The TPM (Trading Partner Management) control provides WebLogic Workshop business processes and Web services with query (read-only) access to trading partner and service information stored in the TPM repository. Information like trading partner name or business ID, default trading partner, basic and extended trading partner properties, default bindings (ebXML or RosettaNet), services, service profiles, and service profile bindings (ebXML, RosettaNet, or

Web service bindings) can be queried and retrieved from the TPM repository. However, only active profile services and active trading partners can access this repository. Since access to the repository is read-only, you cannot modify trading partner and service information. These details can only be modified by using WebLogic Integration Administration Console. For more information, see “[Trading Partner Integration](#)” on page 31. For more information, see: <http://e-docs.bea.com/workshop/docs81/doc/en/integration/controls/controlsTPM.html>

WLI JMS Control

Java Message Service (JMS) is a Java API for communicating with messaging systems. The WLI JMS control enables WebLogic Workshop business processes to easily interact with messaging systems that provide a JMS implementation.

Each WLI JMS control is associated with a specific facility of the messaging system. Once a WLI JMS control is defined, business processes can use it like any other WebLogic Workshop control.

For more information, see:

<http://e-docs.bea.com/workshop/docs81/doc/en/integration/controls/controlsJMS.html>

XML MetaData Cache Control

The XML MetaData Cache control is used in business processes to access and retrieve XML metadata maintained in XML MetaData Cache. This cache is managed by WebLogic Integration Administration Console or the MBean API, which allows users to create their own NetUI-based consoles. The XML MetaData Cache is a global, domain-wide cache. So, data maintained in the cache can be accessed by any business process that is deployed in that domain. The cache can also be used for sharing data within a cluster. The cache is primarily used to maintain configuration metadata. Data is stored as key-value pairs where key is of type String and the value contains XML data. Data from the cache is made available permanently through file-based storage. For each XML document that is added to the cache, a new XML MetaData Cache file is created. The XML MetaData Cache control in a business process uses the key to retrieve XML metadata associated with the key value, from the cache. For more information, see:

<http://e-docs.bea.com/workshop/docs81/doc/en/integration/controls/controlsXMLMetadata.html>

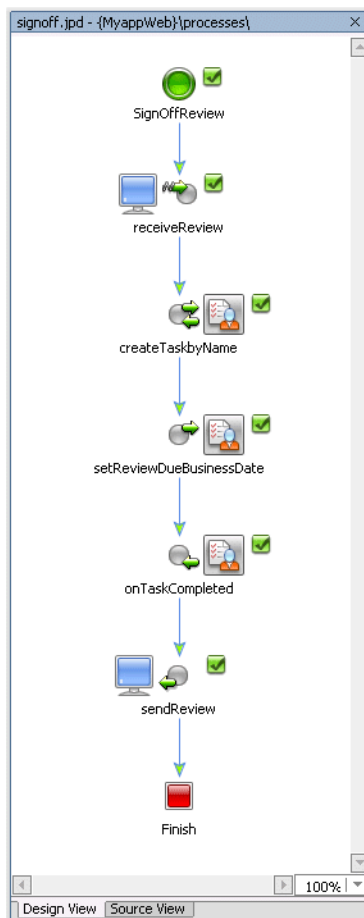
Worklist System

WebLogic Integration delivers rapid access to integration with business users through the Worklist system. The Worklist system supports capabilities to manage users, groups, and roles and manage the routing of tasks to the people in an enterprise. It enables people to collaborate in

business processes including assigning tasks, tracking the status of tasks, handling approvals and so on. Integral to the flow of work are actions such as receiving, approving, modifying, and routing documents. The documents that often accompany work activities provide the background required for people to complete tasks, which are the central component of all Worklist systems.

In the WebLogic Workshop environment, WebLogic Integration provides two controls, the Task control and the Task Manager control, to support the Worklist functionality. [Figure 8](#) illustrates Worklist controls in a business process.

Figure 8 Worklist Controls Add Human Interaction to Business Processes



The Task control and the Task Manager control are server-side components managed by the Workshop framework. They expose Java interfaces that can be invoked directly from the business processes. The Task control creates an instance of a Task, manages its state, data and so on. The Task Worker control assumes ownership of Tasks, works on them, completes them, and provides administrative privileges including starting, stopping, deleting, and assigning. The Worklist system provides functionality that enables end users, such as task creators, task workers, and task administrators, to interact with running business processes for handling process exceptions, approvals, and status tracking.

While a sample Worklist user interface (Worklist client) is provided, you can also create a custom user interface to manage the Worklist components of your applications.

In the run-time Worklist system, a Task instance is a particular object that represents a work assignment in the real world. Task instances are part of the WebLogic Integration server and exist independent of any controls or business processes. Multiple business processes can interact with a Task throughout its lifecycle concurrently. Tasks remain in the run time indefinitely unless they are explicitly deleted or purged by the WebLogic Integration purging process.

Task instances have built-in data values for defining how work should be performed, who should do it, by when it needs to be completed, task assignees, due date, owner, state, priority and so on. You can also use these data values to capture what actually was done when the task is completed.

Operations are used to create new tasks, alter task states or data values, delete tasks, or read information about an existing task. Some operations allow combinations of these actions in a single step. For example, operations can be used to create tasks, modify task properties and so on.

Task data like the task state, due dates, changes in owners and assignees and so on can be tracked for generating reports and compiling statistics related to the task. To optimize performance, tracking task data can be done as and when needed. In addition, this data can be periodically purged.

To find out information about a task, task queries can be used. These queries are analogous to SQL and databases tables. The task data values can be used to specify the search criteria. The results returned by the queries contain information about tasks in the run-time Worklist system that meet the specified criteria. These results can be sorted based on your requirements. Results are sorted based on integer values associated with each sort criteria. Rather than view all the results returned for the task query, you can use the [WorklistScrollableResultManager Interface](#) to limit the number of results that are returned. Additionally, this interface allows you to scroll backward and forward to fetch different ranges of results.

You can use WebLogic Integration Administration Console to administer and manage tasks in the Worklist, business calendars, task properties, and other features.

For more information, see

- *Using the Worklist System*, available at:
<http://edocs.bea.com/wli/docs85/worklist/index.html>
- *Tutorial: Building a Worklist Application*, available at:
<http://edocs.bea.com/wli/docs85/wltutorial/index.html>

Trading Partner Integration

WebLogic Integration allows you to automate and manage relationships with your trading partners. You can streamline your business processes with customers, suppliers, distributors, and other partners to get a top-down view of business transactions across the value chain.

The following table summarizes WebLogic Integration's trading partner integration capabilities.

Table 1 Trading Partner Integration in WebLogic Integration

| Feature | Description |
|--|---|
| Visual public and private process integration | WebLogic Integration leverages the unified programming model and run-time framework of WebLogic Workshop to provide an end-to-end business process integration, with easily implemented controls and templates. |
| Support for leading B2B industry protocols and standards | WebLogic Integration supports the following protocols and standards: ebXML 1.0 and 2.0, RosettaNet 1.1 and 2.0, and Web services. |
| Trading Partner Management (TPM) and repository access | WebLogic Integration provides sophisticated Trading Partner Management capabilities through the unified WebLogic Integration Administration Console. This console enables administrators to easily manage a central repository of trading partner profile information, including protocol bindings used for secure message exchanges between trading partners, services representing public processes, security, and bulk import and export capabilities. Authorized business processes and Web services can dynamically access trading partner information via easily implemented controls. In addition to the administration console, MBean APIs are also provided so that third-party MBean clients can be written to access the TPM repository. |
| Easy access to run-time information | WebLogic Integration provides flexible run-time tracking, auditing, and reporting capabilities to show a top-down view of trading partner activities and business transactions across the value chain. |

Table 1 Trading Partner Integration in WebLogic Integration (Continued)

| Feature | Description |
|--|--|
| High performance and availability | WebLogic Integration provides fast and reliable business message exchanges between trading partners. It supports clustered configuration for scalability and fail-over, message persistence, low-level acknowledgements and receipts, and transactional integrity. |
| High security, auditing, and non-repudiation | WebLogic Integration ensures private, secure, and reliable business message exchanges among trading partners using transport level security with SSL and message level security with digital signature and encryption. The certificates and private keys used for various purposes are stored in protected keystores while the passwords are encrypted and stored in the WebLogic Integration PasswordStore. |
| Interoperability | WebLogic Integration interoperates with a wide range of B2B servers from other vendors. In addition, WebLogic Integration Business Connect, a lightweight B2B server, is designed for small trading partners who do not have their own B2B server. For trading partners who want a zero-install solution, WebLogic Integration can easily be extended to offer a browser or FTP interface. |

Figure 9 shows basic interactive business processes between trading partners. The Buyer business process sends an order to the Seller using an agreed-upon business protocol (ebXML or RosettaNet). The Seller business process receives the request, writes the order to a database, receives an invoice from an internal back-end system, and then sends the invoice to the Buyer using the same business protocol.

This message exchange is known as a *conversation*. In the conversation, the initiating trading partner (Seller) is known as the *initiator*, while the responding trading partner is known as the *participant*.

Figure 9 Interactive Business Processes Between Trading Partners

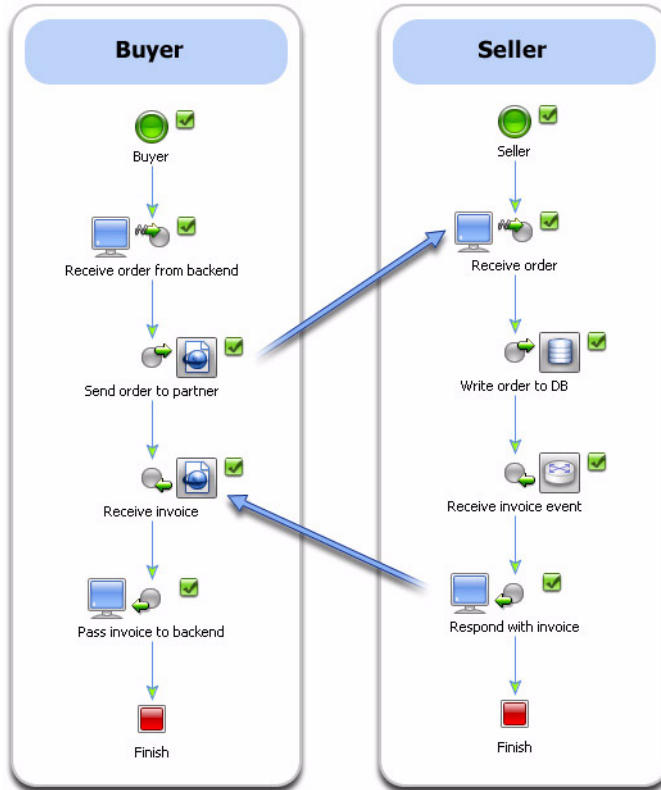
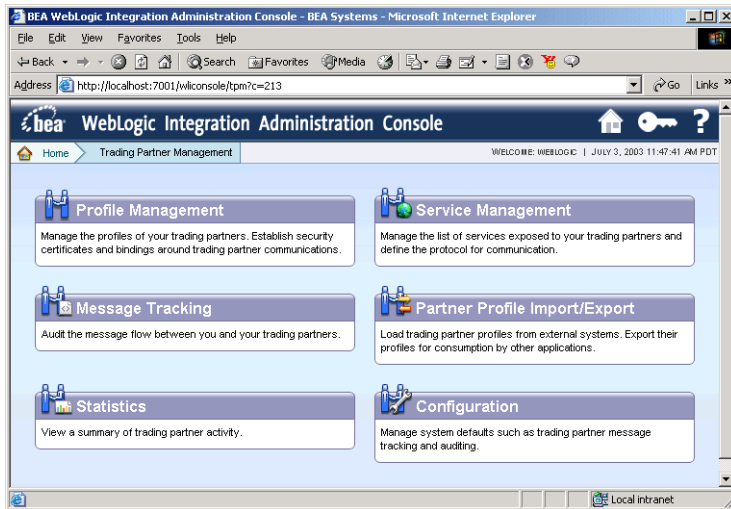


Figure 10 shows the Trading Partner Management home page in WebLogic Integration Administration Console, which allows administrators to manage trading partner profiles, security certificates, protocol bindings, services, message tracking and auditing, trading partner activity, system defaults, and importing and exporting trading partner profile information.

For more information, see *Introducing Trading Partner Integration*, available at:

<http://edocs.bea.com/wli/docs85/tpintro/index.html>

Figure 10 Trading Partner Management in WebLogic Integration Administration Console



Application Integration and Adapters

WebLogic Integration provides an application integration framework comprising the Application View control in the Workshop environment, the Application Integration Design Console, and support for pre-built BEA and custom adapters. The application integration framework allows you to link existing systems and new applications using standards-based architecture to host J2EE Connector Architecture (J2EE-CA)-based adapters.

With the Application View control, you can invoke application view services both synchronously and asynchronously, and start a new business process when an EIS event occurs. In both the service and event cases, the developer uses XML and mapping tools to interact with the Application View control. The developer need not understand the protocol or client API for the enterprise application. Events are delivered using the MB Subscription control. Message Broker integration is provided by publishing all application view events to the Message Broker through its API.

Application Integration Design Console allows integration specialists to configure adapters without coding, including application introspection, abstracting application logic, and inputs and outputs through Application Views.

WebLogic Integration provides a set of J2EE CA-based adapters to integrate with back-end systems, including packaged business applications from major vendors. WebLogic Integration supports leading enterprise applications and technologies with pre-built BEA WebLogic

Adapters for MQSeries, RDBMS, PeopleSoft, SAP, Siebel, and Oracle applications. In addition, it supports the development of J2EE CA-based custom adapters using an Adapter Development Kit (ADK). These adapters are exposed to the WebLogic Workshop framework via the Application View control. To configure an Application View control, the application expert uses the WebLogic configuration tool to configure the adapter and define relevant high-level business operations and events.

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

Table 1 Application Integration Features

| Key Feature | Description |
|--|--|
| Application Integration | <ul style="list-style-type: none"> Standards-based architecture for hosting J2EE CA-based adapters. Exposed as Application View control in the business process. |
| JSP-based Application Integration Design Console | <ul style="list-style-type: none"> Enables the development and management of Application Views. Allows EIS specialists to configure adapters via application introspection. Defines application view services and events. |
| Adapters | <ul style="list-style-type: none"> J2EE CA 1.0-based adapter infrastructure with extensions. Service adapters expose application services to WebLogic Integration. Event adapters publish asynchronous, unsolicited messages from the application to Message Broker. BEA Adapters available for MQSeries, PeopleSoft, RDBMS, SAP, Siebel, and Oracle applications. |
| Adapter Developer Kit (ADK) | <ul style="list-style-type: none"> J2EE CA-compliant sample adapters that developers can use to learn how to develop adapters using the ADK. |

For more information, see

- *Introducing Application Integration*, available at:

<http://edocs.bea.com/wli/docs85/aiover/index.html>

- *Using Application Integration Design Console*, available at:
<http://edocs.bea.com/wli/docs85/aiuser/index.html>
- *Developing Adapters*, available at:
<http://edocs.bea.com/wli/docs85/devadapt/index.html>
- *BEA WebLogic Adapters 8.1* online documentation, a set of online manuals for all BEA package adapters for enterprise application integration, available at:
<http://edocs.bea.com/wladapters/docs81/index.html>

Administration and Management

WebLogic Integration provides a simplified secure browser-based administration console for run-time management and administration analysis. WebLogic Integration Administration Console allows centralized configuration, maintenance, and monitoring of integrated resources, including audit trails as well as associated security and role information, extensible with JMX interfaces to third-party tools. This console is designed for the applications administrator, that is, an administrator who needs an integration-focused view of business processes, messaging activity, and monitoring of deployed applications.

WebLogic Integration separates run-time administration from offline analysis by maintaining two logical database stores. The online administration database contains run-time data about the integration engine, business process states, and message history. This repository is designed for performance—to scale and to retrieve information as quickly as possible while maintaining its data in an optimized format. According to configurable archiving policies, this online repository is periodically archived to an offline data store. Data archiving enables analysis of process data, task and message histories by third-party tools through SQL to archived databases.

Using WebLogic Integration Administration Console, you can monitor and manage process flows, message broker activities, work lists, Application Views, and trading partners from a centralized location. [Figure 11](#) illustrates the WebLogic Integration Administration Console.

Figure 11 Unified, Browser-based Administration Console



WebLogic Integration Administration Console addresses the following operations, management, and administration features:

Table 1 WebLogic Integration Administration Console Features

| Feature | Description |
|--|--|
| Business Process Instance Monitoring and Process Configuration | Deploy and configure business process types, set Service Level agreement (SLA) limits, and manage tracking level on individual business process types. View statistics on running business process instances and filter instance lists. Terminate, delete, and suspend business process instances. |
| Message Broker | Manage all aspects of the Message Broker. View and manage channels, filters, and subscription rules. Monitor the volumes of messages routed through the message broker, or view subscribers to the various channels. Track and reset channel message counts for periodic reporting. For more information, see “Message Broker” in <i>Managing WebLogic Integration Solutions</i> , available at http://e-docs.bea.com/wli/docs85/manage/msgbroker.html . |

Table 1 WebLogic Integration Administration Console Features (Continued)

| Feature | Description |
|--------------------------------------|---|
| Worklist Administration | <p>Manage Worklist users, groups, and tasks. Configure rules for Business Calendars and monitor progress of task completion against due dates. Perform queries to show individual workloads and reassign tasks to speed task progress.</p> <p>For more information, see “Worklist Administration” in <i>Managing WebLogic Integration Solutions</i>, available at http://e-docs.bea.com/wli/docs85/manage/worklist.html.</p> |
| Trading Partner Management | <p>Manage and monitor all trading partner profiles. Configure information about trading partners, including agreed on service access and communication channels. Trading partner information can be maintained individually or bulk-loaded and exported. All messages are tracked and archived for detailed partner auditing. Monitor partner activity through event statistics.</p> <p>For more information, see “Trading Partner Management” in <i>Managing WebLogic Integration Solutions</i>, available at http://e-docs.bea.com/wli/docs85/manage/tpm.html.</p> |
| Application Integration | <p>Monitor enterprise adapter health, tune adapters for automatic suspension and failover, and swap adapters to leverage different enterprise application servers.</p> <p>For more information, see “Application Integration” in <i>Managing WebLogic Integration Solutions</i>, available at http://e-docs.bea.com/wli/docs85/manage/ai.html.</p> |
| User, Group, Role Management | <p>Manage users and groups within the system using tasks. Define new roles for users and allocate roles to groups or users.</p> <p>For more information, see “User Management” in <i>Managing WebLogic Integration Solutions</i>, available at http://e-docs.bea.com/wli/docs85/manage/users.html.</p> |
| System Configuration | <p>Set general configuration and security. Configure security for processes (both authorization and execution policies), channels, and trading partner communications.</p> |

Table 1 WebLogic Integration Administration Console Features (Continued)

| Feature | Description |
|---|--|
| Event Generator Management | Configure environment events that spawn new processes. Environment events can include file updates, reception of email, or placing messages on a JMS queue. |
| Business Calendar Configuration | Maintain calendars that can be used for process scheduling or task assignment. For more information, see “Business Calendar Configuration” in <i>Managing WebLogic Integration Solutions</i> , available at http://e-docs.bea.com/wli/docs85/manage/businesscalendar.html . |
| XML MetaData Cache Management | Maintain configuration metadata in XML MetaData Cache, a global and domain-wide cache. An XML MetaData Cache control can be defined in a business process to retrieve data stored in the cache. |

Business Process Instance Monitoring and Process Configuration

You can use the Process Instance Monitoring resource to view summary statistics that reflect system health, view status and statistics of process instances, view an interactive or printable process instance graph and so on. The information displayed in the Process Instance Monitoring module is based on the tracking data stored in the run-time database. A combination of system-level and process-level properties control the type of data that is available.

You can configure a process to archive the variable values in the process. This can be done both at the system level or at the individual process level. So, it is possible to track the value that was assigned to the process variables while the process is running and track the variable values after the process completes, terminates, or aborts.

A process instance that calls another process instance using a Process control is deemed to have a parent-child relationship. You can use WebLogic Integration Administration Console to monitor parent as well as child processes and navigate between these processes. However, the parent-child navigation functionality is limited to instances invoked via the Process control. Instances started by the Service Control or Service Broker Control are not identified as child instances.

For more information, see “Process Instance Monitoring” in *Managing WebLogic Integration Solutions* available at <http://e-docs.bea.com/wli/docs85/manage/processmonitoring.html>

You can use the Process Configuration module to:

- View process type information and locate specific processes for configuration
- View or update process type properties, such as display name, tracking level, and reporting data policy
- View or update the security policies for a process
- Activate or deactivate a non-versioned process
- Configure the activation time for a newly deployed process version, or rollback to a previous version
- View an interactive or printable process type graph
- View or update the selectors used to dynamically set control attributes for a Process control or Service Broker control

For more information, see “Process Configuration” in *Managing WebLogic Integration Solutions* available at <http://e-docs.bea.com/wli/docs85/manage/processconfig.html>

System Configuration

The System Configuration module in WebLogic Integration Administration Console is used to set general configuration and security parameters. You can configure security for processes (both authorization and execution policies), channels, and trading partner communications. You can use this module to:

- View the configuration for tracking, reporting, and purging data.
Events that are generated by a process instance and a Worklist task instance can be tracked. Similarly, trading partner message history can also be tracked.
- Configure the reporting data and purge processes, the reporting database, and the default tracking level and reporting data policy.

The tracking level can be set to optimize performance and meet system requirements. The tracking data is maintained in the run-time database. The reporting data stream can be enabled or disabled. If the reporting data stream is enabled, the specified reporting database is populated by a near real-time data stream. Because the reporting database is populated by a near real-time stream, it is possible to see a snapshot of the data for process instances that contain partial data.

- Manually start and stop the purge process.

In order to optimize performance, the amount of tracking data stored in the run-time database should be kept to a minimum. To help ensure this, data is periodically purged from the run-time database, as set by an administrator. If the data is required for reporting and analysis, the administrator can enable the transfer of tracking data suitable for reporting to an offline database.

- Manage the password store and the password aliases.

The password store provides for the secure storage of the passwords used by controls, event generators, and other WebLogic Integration components. Each required password is defined in the password store and associated with a password alias. This alias can then be referenced in the annotations of process definitions (*.jpd), control extensions (*.jcx), and event generator configuration files (wliconfig/*EventGen.xml).

- Configure the server for application integration and configure the Worklist task creation role.

The application integration configuration, debug status (enabled or disabled), JMS connection factory, and repository root directory can be managed. In addition, you can set, edit, and view the worklist task creation role. This is the role that is authorized to create worklist tasks.

For more information, see “System Configuration” in *Managing WebLogic Integration Solutions* available at <http://e-docs.bea.com/wli/docs85/manage/system.html>.

Event Generator Management

Event generators trigger events in response to activities that occur in the systems associated with them. Each event generator is associated with a Message Broker channel. When a system event occurs, event generators publish this information using these channels. So, you need to configure a set of channel rules for each event generator.

The various types of event generators that are supported by WebLogic Integration are: File, Email, JMS, Timer, RDBMS, HTTP, MQSeries, and TIBCO RV. These event generators can be managed, deployed, and monitored using WebLogic Integration Administration Console. Event generators can be deployed both on standalone servers as well as on clusters. For more information about deploying event generators on clusters, see “Deploying Event Generators” in *Understanding WebLogic Integration Clusters*.

For more information about, see “Event Generators” in *Managing WebLogic Integration Solutions* available at <http://e-docs.bea.com/wli/docs85/manage/evntgen.html>.

Table 2 Various Types of Event Generators

| Event Generator | Description |
|------------------------|--|
| File | File event generator polls files or directories on a local drive or FTP server and generates an event when an operation is performed on the file. The operation can be read, write, or append file. In addition, file manipulation operations such as copy, rename, and delete can also be monitored. The monitored files can be one of the following types: XmlObject, RawData (binary), or String. Details of the file operations are published on Message Broker channels as XML or binary objects. |
| Email | Email event generator polls e-mail accounts for messages and publishes the contents to Message Broker channels. |
| JMS | JMS event generator polls JMS queues and topics for messages. These messages are then published using the Message Broker channels. |
| Timer | Timer event generators create events at user designated times and publish the events to Message Broker channels. When the Timer event generator detects that a designated time has passed, it publishes a message to a Message Broker channel. The message content can be specified in the channel rules defined for the event generator. These are always XML messages. |
| RDBMS | RDBMS event generators can be used to generate an event when an insert, update, or delete action takes place on any database table associated with the event generator. The RDBMS event generator can also be used to query data available in a database. These updates are then published on Message Broker channels. |
| HTTP | The HTTP event generator is a servlet, which takes HTTP requests, checks for the content type, and then publishes the messages to Message Broker channels. The HTTP event generator supports synchronous subscription with return and also supports a Multipart response. |

Table 2 Various Types of Event Generators

| Event Generator | Description |
|-----------------|---|
| MQSeries | <p>MQSeries is the messaging service queue provided by IBM. The MQSeries event generator polls this message queue for any messages and publishes them through Message Brokers channels.</p> |
| | <p>Disclaimer</p> |
| | <p>In the WebLogic Integration product, BEA provides and supports MQSeries (now known as WebSphere® MQ) integration through the following options:</p> |
| | <ul style="list-style-type: none"> <li data-bbox="427 595 1210 743"> <p>• <i>BEA WebLogic Integration JMS Control within the WebLogic Workshop IDE</i> To learn more, see “WLI JMS Control” in <i>Using Integration Controls</i> available at http://e-docs.bea.com/workshop/docs81/doc/en/integration/controls/controlsJMS.html.</p> <li data-bbox="427 760 1210 847"> <p>• <i>JMS Event Generators in BEA WebLogic Integration Administration Console</i> To learn more, see “Event Generators” in <i>Managing WebLogic Integration Solutions</i> available at http://e-docs.bea.com/wli/docs85/manage/evntgen.html.</p> <li data-bbox="427 864 1210 951"> <p>• <i>Async Sample illustrating the use of JMS with MQSeries</i> To learn more, see “Async Binary Update Sample” available at http://e-docs.bea.com/wli/docs85/sol_samples/async_binary.html.</p> <li data-bbox="427 968 1210 1142"> <p>• <i>BEA WebLogic Integration MQSeries Control within the WebLogic Workshop IDE</i> To learn more, see “MQSeries Control” in <i>Using Integration Controls</i> available at http://e-docs.bea.com/workshop/docs81/doc/en/integration/controls/controlsMQ.html.</p> <li data-bbox="427 1159 1210 1246"> <p>• <i>BEA Adapter for MQSeries</i> To learn more, see BEA WebLogic Adapter for MQSeries 8.1 Documentation available at http://e-docs.bea.com/wladapters/mq/docs81/index.html.</p> |
| | <p>The WebLogic Integration Control and Adapter rely on use of the MQSeries APIs. IBM has not currently certified the MQSeries APIs for BEA WebLogic Integration 8.1 and 8.5 products. To resolve issues for any aspect of the MQSeries Control or Adapter that is not part of the BEA WebLogic Integration product or is specific to IBM WebSphere MQ, the APIs might require a workaround. JMS-based MQ integration, however, does not have the same support-related limitations as WebSphere MQ APIs. For a current list of BEA products that IBM has explicitly stated it supports in conjunction with WebSphere MQ, see the following URL:</p> |
| | <p>http://www-306.ibm.com/software/integration/mqfamily/platforms/supported/wsmq_for_solaris_5_3.html.</p> |

Table 2 Various Types of Event Generators

| Event Generator | Description |
|-----------------|---|
| TIBCO RV | <p>TIBCO Rendezvous™ is a messaging software provided by TIBCO. It enables exchange of data across applications running on distributed platforms. TIBCO RV event generator listens for messages on a Rendezvous subject and raises events to the Message Broker on receiving the desired message. The messages are received in most formats supported by Rendezvous, converted to binary and then published on the Message Broker channels. These messages can be of type XML, String, and the TIBCO proprietary Rendezvous Message format.</p> <p>Disclaimer</p> <p>Use of the TIBCO RV control and event generator with BEA WebLogic Integration in no manner confers or grants the right to use TIBCO Rendezvous including "dynamic libraries". In order to use such TIBCO products, the user of the TIBCO RV control and event generator must obtain a valid license from TIBCO. See http://www.tibco.com for information on how to obtain a licensed copy of Rendezvous.</p> |

XML MetaData Cache Management

XML MetaData Cache is a global, domain-wide cache. Data maintained in this cache can be accessed by any business process that is deployed in that domain. The cache can also be used for sharing data within a cluster. This cache can be managed by using WebLogic Integration Administration Console or the MBean API, which allows users to create their own NetUI-based consoles. The cache is primarily used to maintain configuration metadata. Data is stored as key-value pairs where key is of type String and the value contains XML data. Data from the cache is made available permanently through file-based storage. For each XML document added to the cache, a new XML MetaData Cache file is created. The Admin Server needs to be up and running if an administrator wants to modify the cache. An XML MetaData Cache control can be defined in a business process to retrieve data stored in the cache. The control uses the key to retrieve XML metadata associated with the key value, from the cache.

For more information about, see "XML Cache" in *Managing WebLogic Integration Solutions* available at <http://e-docs.bea.com/wli/docs85/manage/xmlcache.html>.

Related Documents

The following table provides links to useful information in the WebLogic Integration documentation set, such as topics that teach you how to use the tools for implementing, deploying, and managing integration applications and resources.

Table 3 WebLogic Integration Topics

| To learn about... | See ... |
|--|--|
| Developing business processes | <p data-bbox="485 427 1233 482"><i>Tutorial: Building Your First Business Process</i>, available in the WebLogic Workshop Help at: http://edocs.bea.com/workshop/docs81/doc/en/integration/tutorial/tutWLIProcessIntro.html</p> <p data-bbox="485 562 1233 618"><i>Guide to Building Business Processes</i>, available in the WebLogic Workshop online help at: http://edocs.bea.com/workshop/docs81/doc/en/integration/wfguide/wfguideIntro.html</p> |
| Implementing data transformations | <p data-bbox="485 708 1233 763"><i>Tutorial: Building Your First Data Transformation</i>, available in the WebLogic Workshop Help at: http://edocs.bea.com/workshop/docs81/doc/en/integration/dttutorial/tutWLIDataTransIntro.html</p> <p data-bbox="485 843 1233 933"><i>Guide to Data Transformation</i>, available in the WebLogic Workshop Help at: http://edocs.bea.com/workshop/docs81/doc/en/integration/dtguide/dtguideIntro.html</p> |
| Using WebLogic Integration controls | <p data-bbox="485 960 1233 987"><i>Using Integration Controls</i>, available in the WebLogic Workshop Help at: http://edocs.bea.com/workshop/docs81/doc/en/integration/controls/controlsIntro.html</p> |
| Adding worklist capabilities to a business process | <p data-bbox="485 1078 1233 1105"><i>Using the Worklist System</i>, available at: http://edocs.bea.com/wli/docs85/worklist/index.html</p> <p data-bbox="485 1156 1233 1220"><i>Tutorial: Building a Worklist Application</i>, available at: http://edocs.bea.com/wli/docs85/wltutorial/index.html</p> |
| Trading partner integration | <p data-bbox="485 1246 1233 1274"><i>Introducing Trading Partner Integration</i>, available at: http://edocs.bea.com/wli/docs85/tpintro/index.html</p> |

Table 3 WebLogic Integration Topics

| To learn about... | See ... |
|--|--|
| B2B controls | <p><i>RosettaNet Control</i>, available in the WebLogic Workshop Help at: http://edocs.bea.com/workshop/docs81/doc/en/integration/controls/controlsRosettaNet.htm</p> <p><i>ebXML Control</i>, available in the WebLogic Workshop Help at: http://edocs.bea.com/workshop/docs81/doc/en/integration/controls/controlslebXML.html</p> <p><i>TPM Control</i>, available in the WebLogic Workshop Help at: http://e-docs.bea.com/workshop/docs81/doc/en/integration/controls/controlsTPM.html</p> |
| Managing integration applications | <p><i>Managing Integration Solutions</i>, available at: http://edocs.bea.com/wli/docs85/manage/index.html</p> |
| Application integration | <p><i>Introducing Application Integration</i>, available at: http://edocs.bea.com/wli/docs85/aiover/index.html</p> <p><i>Using Application Integration Design Console</i>, available at: http://edocs.bea.com/wli/docs85/aiuser/index.html</p> |
| Creating custom adapters | <p><i>Developing Adapters</i>, available at: http://edocs.bea.com/wli/docs85/devadapt/index.html</p> |
| Connecting your EIS with BEA WebLogic Adapters | <p><i>BEA WebLogic Adapters 8.1</i> online documentation, a set of online manuals for all BEA package adapters for enterprise application integration, available at: http://edocs.bea.com/wladapters/docs81/index.html</p> |
| TIBCO RV Control and Event Generator | <p><i>TIBCO Rendezvous Control and Event Generator User Guide</i>, available at: http://e-docs.bea.com/wli/docs85/tibcorv/index.html</p> |
| BPEL Import and Export Tools | <p><i>BPEL Import and Export User Guide</i>, available at: http://e-docs.bea.com/wli/docs85/bpel/index.html</p> |
| Upgrading to WebLogic Integration 8.5 SP5 | <p><i>WebLogic Integration 8.5 Upgrade Guide</i>, available at: http://e-docs.bea.com/wli/docs85/upgrade/index.html</p> |
| Deploying WebLogic Integration applications | <p><i>Deploying WebLogic Integration Solutions</i>, available at: http://edocs.bea.com/wli/docs85/deploy/index.html</p> |

Index

A

- adapters -34
- application integration -34
- Application View control -22
- AquaLogic Service Bus -7

B

- BPEL processes, importing and exporting -14
- BPM -10
- business process calls -14
- business process configuration -39
- business process instance, monitoring -39
- business process management -10
- business process resources, Web services -15
- business processes, stateless and stateful -14
- business processes, synchronous and asynchronous -13

C

- common application framework, benefits -4

D

- data transformation -16
- Dynamic Transformation control -22

E

- ebXML control -22
- Email control -23
- Email event generator -42
- enterprise integration, unified approach -6
- Event Generators -41

F

- File control -23
- File event generator -42

H

- Http control -23
- HTTP event generator -42

I

- Integration component of WebLogic Platform -2

J

- JMS event generator -42

M

- MB Publish control -24
- MB Subscription control -24
- Message Broker -19
- MQSeries Control -24
- MQSeries event generator -43

P

- Process control -26

R

- RDBMS event generator -42
- RosettaNet control -26

S

- Service Broker control -26
- System configuration -40

T

- Task control -27
- Task Worker control -27
- TIBCO RV control -27
- TIBCO RV event generator -44
- Timer event generator -42
- TPM control -27

Trading Partner Integration -31

W

WebLogic Integration 8.5 and AquaLogic

Service Bus -8

WebLogic Integration Administration Console

-36

WebLogic Integration controls -21

Application View -22

Dynamic Transformation -22

ebXML -22

Email -23

File -23

Http -23

Message Broker controls -24

MQSeries -24

Process -26

RosettaNet -26

Service Broker -26

TIBCO RV -27

TPM -27

WLI JMS -28

Worklist -27

XML MetaData Cache -28

WebLogic Integration, features -8

WLI JMS control -28

Worklist system -28

X

XML MetaData Cache -44

XML MetaData Cache control -28