BEA WebLogic Integration™

Introducing BEA WebLogic Integration

Version 9.2
Document Date: November 2006

Copyright

Restricted Rights Legend

Trademarks and Service Marks

Contents

Introduction to WebLogic Integration

BEA WebLogic Integration 9.2 is a unified solution for integration business systems within an enterprise. WebLogic Integration provides a development and run-time framework that unifies all the components of business integration--business process management, data transformation, trading partner integration, connectivity, message brokering, application monitoring, and user interaction--into a single flexible environment. WebLogic Integration 9.2 reduces the cost of management and operations by providing highly reliable, stable, scalable, and mission critical integration solutions.

WebLogic Integration combines the divergent pieces of the business integration picture, including ERP, CRM, legacy applications, business users, supply chains, and trading partners, by providing a versatile development environment that delivers rapid business integration with simplified production and management.

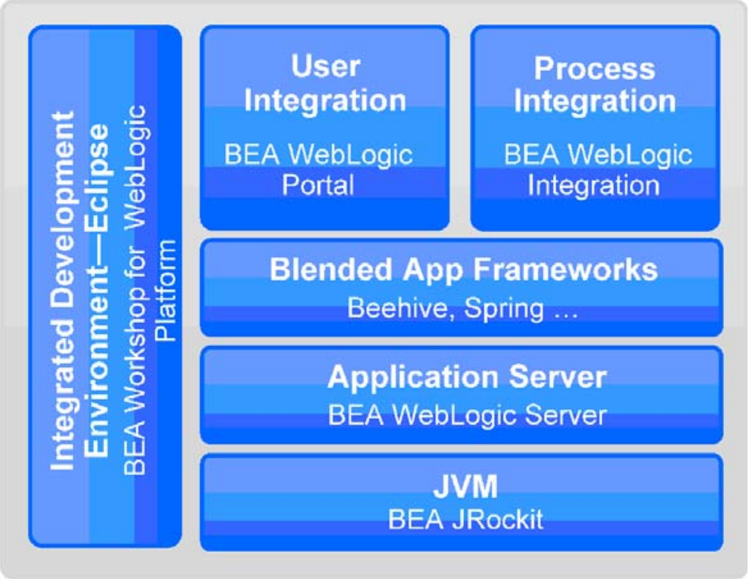
Integration Component of WebLogic Platform

WebLogic Integration is part of the WebLogic Platform that provides functionality for businesses to use to develop new applications, create both businesses and web services, integrate them 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 whether it is for 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.

Figure 1 illustrates the WebLogic Platform that comprises of multiple components that you can use independently, or in combination, as required for your application.

Figure 1 WebLogic Platform

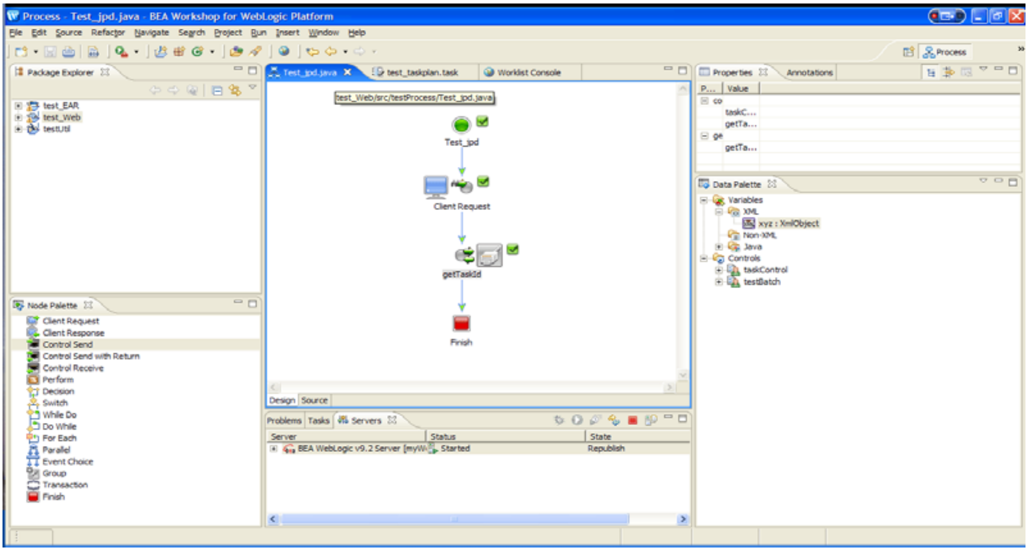


BEA WebLogic Server, the industry-leading J2EE application server, provides the critical infrastructure needed to develop integrated solutions, including security, transaction management, fault tolerance, persistence, and clustering.

Leveraging WebLogic Server as the underlying deployment environment, WebLogic Integration uses Web services to integrate distributed systems inside and outside an organization and utilizes the BEA Workshop for WebLogic Platform framework to simplify application development.

WebLogic Integration works seamlessly in the BEA Workshop for WebLogic Platform environment. WebLogic Integration uses the power of BEA Workshop for WebLogic Platform to provide a robust set of tools to develop and extend integration applications. WebLogic Integration provides graphical tools you use to edit your business processes, and human interaction task plans. These tools are available in the same BEA Workshop for WebLogic Platform environment as your controls, Web services, and portal-building tools as shown in Figure 2.

Figure 2 WebLogic Integration Business Process in the BEA Workshop for WebLogic Platform IDE



Benefits of a Common Application Framework

While WebLogic Integration provides a robust set of general purpose tools, your integration solution will require some of your own custom behavior. You may write application logic or expose business processes to the user for human tasks such as business approvals. You may customize the user interface that your user's see when providing the business approvals. You may choose to build a Web-based application on top of your integration application that provides a custom web portal for your users. BEA's common application framework allows you to develop all these components of your solution in the same environment. This is where the WebLogic unified platform can provide you with an ease-of-use advantage.

In the same framework and the integrated development environment, you can define business processes, 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 your integration application, you can stay within the IDE to build your 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 Eclipse-based BEA Workshop for WebLogic Platform framework for development, integration and portal applications includes the following benefits as listed in Table 1:

Table 1 Benefits of a Common Framework

Feature	Benefit
Industry Standard IDE	• Eclipse-based Integrated Development Environment allows you to leverage the power of the extended Eclipse community for skills, ideas, discussion groups, and plug-ins • Improve developer collaboration within and across projects • Take advantage of prevalent Eclipse familiarity to quickly bring new team members on board
Unified programming model	• Event-driven programming model based on procedural logic development • Control-based environment for constructing processes and abstracting resources • Abstracts the low-level technical details of the J2EE APIs and back-end resources
Common look-and-feel	• Provides a unified developer experience across BEA Workshop for WebLogic Platform, Integration, and Portal components
Annotated Java code model	• Enables you to specify behavior and focus on handling events and calling methods, instead of writing complex code • Provides meta data-driven application construction • Annotations based on JSR-175 standards
Web services	• Natively built, extensible, and integrated at the enterprise level • Expose processes as Web services and invoke internal and third-party Web services from IDE components • Implemented processes automatically accessible as a Web service • Follow Web services standards, such as Simple Object Access Protocol (SOAP) and Web Services Description Language (WSDL) • Accessed via XML messages

Table 1 Benefits of a Common Framework

Feature	Benefit
Common project and deployment model	• Application encapsulation through J2EE mechanisms – WAR and EAR files
Controls	
Simple visual components	• High-level, easy-to-use interface represented as visual elements in the Workshop environment • Behavior defined through methods, events, and properties
Extensible architecture	• Application artifacts built in BEA Workshop for WebLogic Platform automatically become controls that can be reused anywhere within the environment • Developers and ISVs can develop custom controls
Consistent mechanism for representing resources	• All resources look the same—abstracts resource specific details • Reduces development learning curve
Composition	• Controls can invoke other controls
Java component with Java methods	• Easier access to J2EE resources
Complete access to J2EE API	• 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

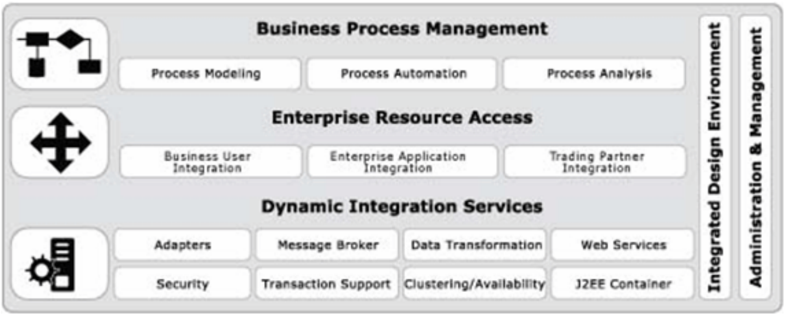
Unified Approach to Enterprise Integration

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

Whether your starting point is 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, you have one environment to build your integrated enterprise application.

BEA Workshop for WebLogic Platform represents rapid integration with BEA Workshop for WebLogic Platform.

Figure 3 Rapid Integration with BEA Workshop for WebLogic Platform


Business Process Management
Process Modeling
Process Automation
Process Analysis
Enterprise Resource Access
Business User Integration
Enterprise Application Integration
Trading Partner Integration
Dynamic Integration Services
Adapters
Message Broker
Data Transformation
Web Services
Security
Transaction Support
Clustering/Availability
J2EE Container
Integrated Design Environment
Administration & Management

By providing access to enterprise resources such as messaging, adapters, and integration controls coupled to business process modeling, human interaction workflow modeling, and data transformation, the WebLogic Integration environment equips IT staff with the means to quickly implement and bind business processes to IT resources without specialized knowledge of the deployment environment.

Within the BEA Workshop for WebLogic Platform framework with its robust Web services and controls architecture, WebLogic Integration supports a business process layer of abstraction and a common language for requirements gathering, for validating implementation, and for monitoring runtime execution. By bridging the gap between the development and integration environments, WebLogic Integration cuts through the accumulation of proprietary integration technologies and makes the integration effort easier and less costly.

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

- **Loosely coupled integrations** are easier to maintain than traditional tight and rigid integrations.
- **Asynchronous communication** is critical to conducting and protecting business operations when communication goes down.

- **Coarse-grained communication** is key to maximizing the efficiency of typically high cost communication between loosely coupled systems.

With a common environment that recognizes that code must be written for integration solutions and that applications require integration to communicate, WebLogic Integration enables the reuse of technical skills across the entire life-cycle of building, integrating, and deploying applications. The following section outlines the key WebLogic Integration features available in the BEA Workshop for WebLogic Platform environment.

Interoperability of WebLogic Integration with AquaLogic Service Bus

AquaLogic Service Bus is enterprise infrastructure software. It is a key product in the Service Infrastructure family of configuration-driven infrastructure software, 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, text, 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. It also supports BEA WebLogic Tuxedo and EJB.

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 9.2 can be leveraged by AquaLogic Service Bus as a service end point. 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.

BEA now offers AquaLogic Integrator, a cost-effective solution to build, connect and manage integrated process services. AquaLogic Integrator delivers the best solution for IT to integrate, deploy and manage process-driven services inside and outside the enterprise, by combining the power and flexibility of BEA WebLogic Integration for process integration with the high-performance stateless mediation of BEA AquaLogic Service Bus.

For more information, see the AquaLogic Service Bus 2.5 documentation.

Features of BEA WebLogic Integration

WebLogic Integration contains an Eclipse-based IDE framework, that in turn contains many perspectives such as the Process and Task Plan perspective. Each perspective has their own Editors, Views, Resources, Projects and Facets.

WebLogic Integration provides a set of enhanced enterprise integration-specific capabilities within the Workshop environment that are designed for improved developer productivity. WebLogic Integration provides the following design-time and runtime integration features as listed in Table 2:

Table 2 Features

Feature	Description
Eclipse-Based Integration Development Environment (IDE)	Contains an open source IDE framework, that follows the best practices of Eclipse, and also functions with Eclipse plug-ins. WebLogic Integration uses Eclipse concepts such as Workspace, Workbench, Editors, Views, Resources, Projects, Perspectives, and Facets.
Business Process Management	Enables users to model business process scenarios and orchestrate integration and automation in terms of business goals
Data Transformation	Transforms data from one format to another using simple drag-and-drop mapping. 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 and unions.

Table 2 Features

Feature	Description
Message Broker	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, low latency message routing between applications.
Event Generators	Event generators publish messages to Message Broker channels in response to system events.
WebLogic Integration Controls	Enable 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. The controls container is based on Apache Beehive.
Worklist System	WebLogic Integration 9.2 delivers a new full-featured Worklist system to manage end-user interaction for process exceptions, approvals, and status tracking. You can create a reusable sequence of end-user steps that can be used with one or many business processes by using a drag-and-drop design interface, out-of-the-box portlets, and generate automated forms. Worklist also includes centralized user and group management and user rules and authorization for secure participation within processes.
Trading Partner Integration	Enables rapid, secure online-connection with suppliers and customers through leading standard protocols such as RosettaNet and ebXML. These protocols provide: • Secure messaging, digital signatures and encryption • Recoverable and trackable messages • Dynamic configuration updates 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. Manages trading partner profiles with streamlined import and export of configurations.

Table 2 Features

Feature	Description
Application Integration and Adapters	Application Integration (AI) provides access from Workshop controls to the Application Integration Design Console, Application Views, pre-built BEA WebLogic Adapters, and custom adapters.
Administration and Management	Administration Console provides integration-focused lifecycle management of running business processes, deployed applications, message broker traffic, enterprise adapter health and usage, trading partner activity and parameters, and worklists, giving administrators full, secure visibility into the distributed integration environment.
BPEL Import and Export	WebLogic Integration supports the import and export of BPEL 1.1 processes.

Eclipse-Based Integration Development Environment

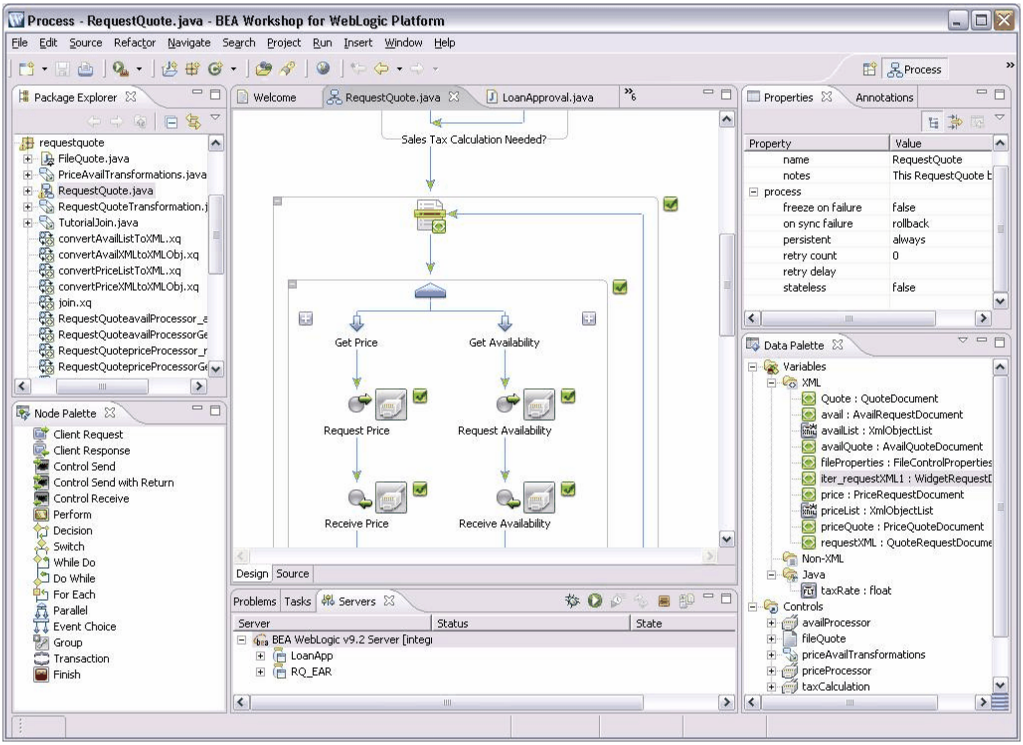
WebLogic Integration contains an IDE framework based on Eclipse 3.1.2. When you create a WebLogic Integration application, you will use the following elements in Eclipse:

- Workbench
- Workspace
- Editors
- Views
- Resources
- Facets
- Perspectives

Workbench

The Workbench refers to the development environment window. The Workbench provides a central place for creation, management, navigation, and control of various workspace resources. It can house several projects. You can work on several projects simultaneously as in Figure 4, without re-opening the IDE for every type of project.

Figure 4 Workbench

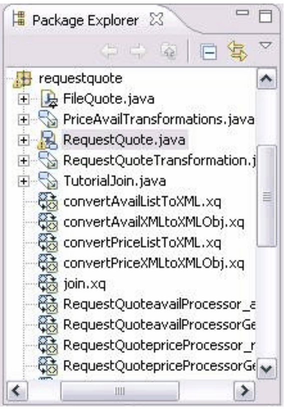

Process - RequestQuote.java - BEA Workshop for WebLogic Platform
File Edit Source Refactor Navigate Search Project Run Insert Window Help
Process
Package Explorer
requestquote
FileQuote.java
PriceAvailTransformations.java
RequestQuote.java
RequestQuoteTransformation.j
TutorialJoin.java
convertAvailListToXML.xq
convertAvailXMLtoXMLObj.xq
convertPriceListToXML.xq
convertPriceXMLtoXMLObj.xq
join.xq
RequestQuoteavailProcessor_a
RequestQuoteavailProcessorGe
RequestQuotepriceProcessor_r
RequestQuotepriceProcessorGe
Node Palette
Client Request
Client Response
Control Send
Control Send with Return
Control Receive
Perform
Decision
Switch
While Do
Do While
For Each
Parallel
Event Choice
Group
Transaction
Finish
Welcome
RequestQuote.java
LoanApproval.java
Sales Tax Calculation Needed?
Get Price
Get Availability
Request Price
Request Availability
Receive Price
Receive Availability
Design
Source
Problems
Tasks
Servers
Server
Status
State
BEA WebLogic v9.2 Server [integ
LoanApp
RQ_EAR
Properties
Annotations
Data Palette
Variables
XML
Quote : QuoteDocument
avail : AvailRequestDocument
availList : XmlObjectList
availQuote : AvailQuoteDocument
fileProperties : FileControlProperties
iter_requestXML1 : WidgetRequestD
price : PriceRequestDocument
priceList : XmlObjectList
priceQuote : PriceQuoteDocument
requestXML : QuoteRequestDocume
Non-XML
Java
taxRate : float
Controls
availProcessor
fileQuote
priceAvailTransformations
priceProcessor
taxCalculation

Property	Value
name	RequestQuote
notes	This RequestQuote b
process	
freeze on failure	false
on sync failure	rollback
persistent	always
retry count	0
retry delay	
stateless	false

Workspace

The Workspace is the root directory which holds the various projects, folders, sub-folders, files, libraries, and other artifacts. It also houses a .metadata folder containing the Eclipse log files and plug-in folder for the Eclipse workspace as shown in Figure 5.

Figure 5 Workspace


Package Explorer
requestquote
FileQuote.java
PriceAvailTransformations.java
RequestQuote.java
RequestQuoteTransformation.j
TutorialJoin.java
convertAvailListToXML.xq
convertAvailXMLtoXMLObj.xq
convertPriceListToXML.xq
convertPriceXMLtoXMLObj.xq
join.xq
RequestQuoteavailProcessor_a
RequestQuoteavailProcessorGe
RequestQuotepriceProcessor_r
RequestQuotepriceProcessorGe

Editors

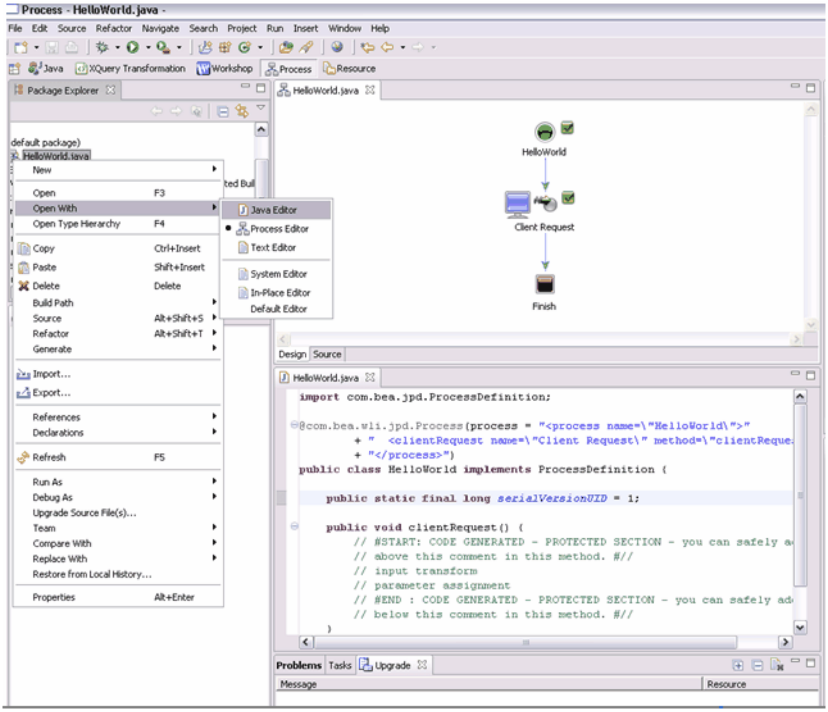
The Editor opens a specific type of file and allows you to edit the content. It also helps you to visualize the file content as required. You can open multiple editors but only one can be active at a time in a workbench. Editors are stacked by default, but you can tile them and thereby even view them simultaneously. You can associate different files with different editors. Double click a file to open the file and invoke its associated editor.

BEA Workshop for WebLogic Platform IDE supports special editors such as:

- Process Editor
- Task Plan Editor
- XQuery Transformation Editor
- Format Builder Editor

These editors are available in the central area of the workspace as in Figure 6.

Figure 6 Editors



```
import com.bea.jpd.ProcessDefinition;

@com.bea.wli.jpd.Process(process = "<process name=\"HelloWorld\">"
        + "  <clientRequest name=\"Client Request\" method=\"clientReque
        + "</process>")
public class HelloWorld implements ProcessDefinition {

    public static final long serialVersionUID = 1;

    public void clientRequest() {
        // #START: CODE GENERATED - PROTECTED SECTION - you can safely a
        // above this comment in this method. #//
        // input transform
        // parameter assignment
        // #END : CODE GENERATED - PROTECTED SECTION - you can safely ad
        // below this comment in this method. #//
    }
```

Views

Views support editors and provide an alternative presentation of workbench resources. You can use these views to easily navigate through the resources. Every View has its own menu, and in most cases contains tool bars as well. A view can appear alone, or stacked with other views.

Some of the standard Eclipse views are:

- Package Explorer
- Navigator

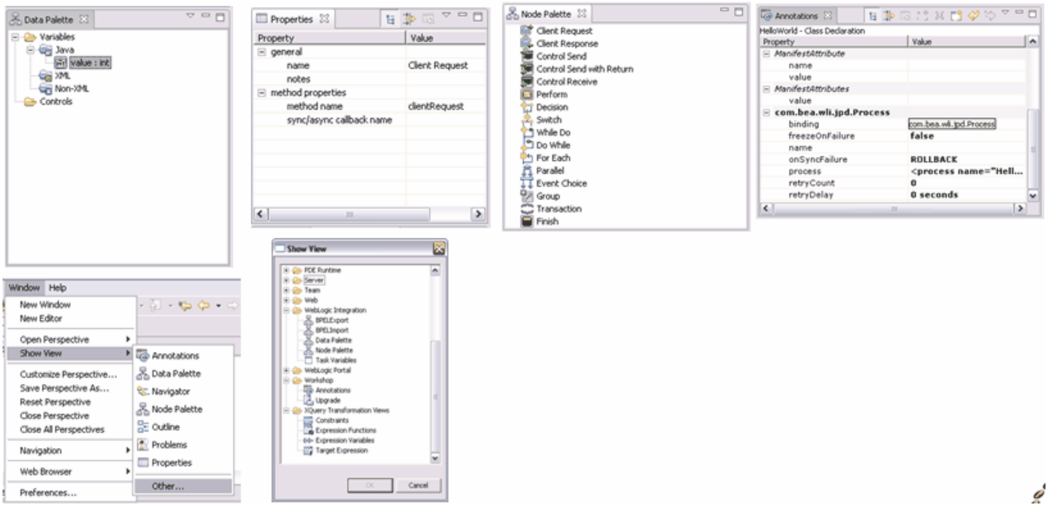
  Use Package Explorer and Navigator to view projects and source files as in Figure 8.
- Problems
- Properties
- Error log

Some of the BEA Workshop for WebLogic Platform specific views are:

- Node Palette
- Data Palette
- Annotations view
- Server view

The Views support Editors and allow you to present and navigate through information using various methods as shown in Figure 7.

Figure 7 Views



Property	Value
general	
name	Client Request
notes	
method properties	
method name	clientRequest
sync/async callback name	

Property	Value
ManifestAttribute	
name	
value	
ManifestAttributes	
value	
com.bea.wli.jpd.Process	
binding	com.bea.wli.jpd.Process
freezeOnFailure	**false**
name	
onSyncFailure	**ROLLBACK**
process	**<process name="Hell...**
retryCount	**0**
retryDelay	**0 seconds**

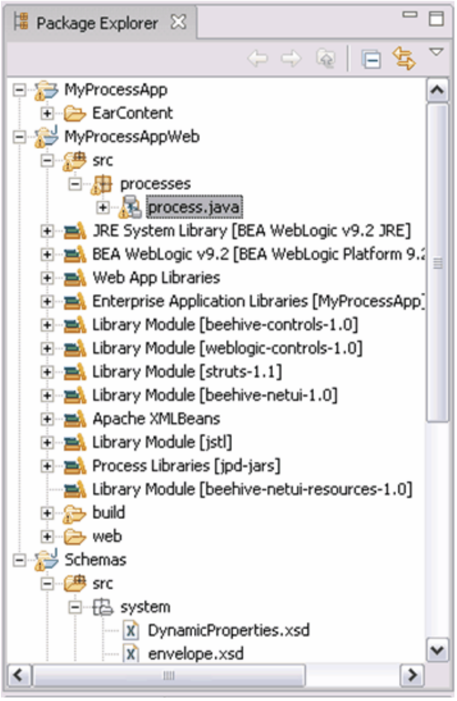
Resources

A Workspace houses three types of resources - Projects, Folders and Files. A Project is a collection of files and folders. A Project is a container within which you can organize the various resources related to the project. Every project is associated with a metadata file called .project. This file contains information such as project name, referenced projects, associated validators/builders, and applied facets.

Resources are organized in a tree-like structure. The Workspace contains projects. Every project contains folders with multiple files.

You can view projects using the Package Explorer as in Figure 8.

Figure 8 Package Explorer View of Projects



Package Explorer

- MyProcessApp
  - EarContent
- MyProcessAppWeb
  - src
    - processes
      - process.java
  - JRE System Library [BEA WebLogic v9.2 JRE]
  - BEA WebLogic v9.2 [BEA WebLogic Platform 9.2
  - Web App Libraries
  - Enterprise Application Libraries [MyProcessApp]
  - Library Module [beehive-controls-1.0]
  - Library Module [weblogic-controls-1.0]
  - Library Module [struts-1.1]
  - Library Module [beehive-netui-1.0]
  - Apache XMLBeans
  - Library Module [jstl]
  - Process Libraries [jpd-jars]
  - Library Module [beehive-netui-resources-1.0]
  - build
  - web
- Schemas
  - src
    - system
      - DynamicProperties.xsd
      - envelope.xsd

Facets

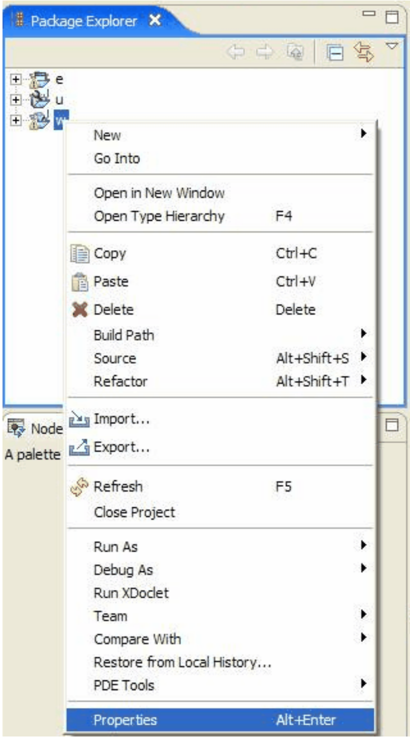
When you create a Project, you need to specify the type of the project to determine the capabilities of that Project. For example, you need to add the libraries, set the various compiler options, decide the various publishing tasks, set the build path, add or remove various validators, builders, and processors. You can define a Facet that contains all these details instead of manually setting this every time you create a project. When a project is created, you can apply the specific facet and set all these details for that project.

Each WebLogic project has two core facets:

- An enabler facet that specifies the type of project
- An extension facet that specifies the features added

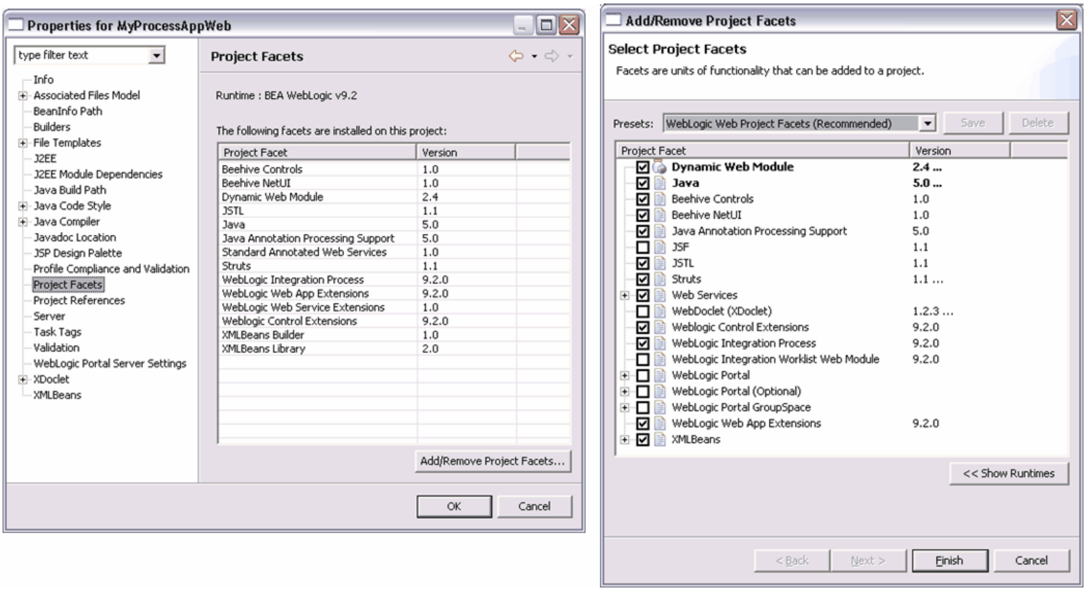
You can add and remove project capabilities using facets. Facets are pre-configured into projects using the process application wizard. Add and remove facets as in Figure 9.

Figure 9 Select Facets



Select Properties to proceed to Figure 10.

Figure 10 Add and Remove Facets



Project Facets

Runtime : BEA WebLogic v9.2

The following facets are installed on this project:

Project Facet	Version
Beehive Controls	1.0
Beehive NetUI	1.0
Dynamic Web Module	2.4
JSTL	1.1
Java	5.0
Java Annotation Processing Support	5.0
Standard Annotated Web Services	1.0
Struts	1.1
WebLogic Integration Process	9.2.0
WebLogic Web App Extensions	9.2.0
WebLogic Web Service Extensions	1.0
Weblogic Control Extensions	9.2.0
XMLBeans Builder	1.0
XMLBeans Library	2.0

Select Project Facets

Facets are units of functionality that can be added to a project.

Presets: WebLogic Web Project Facets (Recommended)

Project Facet	Version
☑ **Dynamic Web Module**	**2.4 ...**
☑ **Java**	**5.0 ...**
☑ Beehive Controls	1.0
☑ Beehive NetUI	1.0
☑ Java Annotation Processing Support	5.0
☐ JSF	1.1
☑ JSTL	1.1
☑ Struts	1.1 ...
☑ Web Services	
☐ WebDoclet (XDoclet)	1.2.3 ...
☑ Weblogic Control Extensions	9.2.0
☑ WebLogic Integration Process	9.2.0
☐ WebLogic Integration Worklist Web Module	9.2.0
☐ WebLogic Portal	
☐ WebLogic Portal (Optional)	
☐ WebLogic Portal GroupSpace	
☑ WebLogic Web App Extensions	9.2.0
☑ XMLBeans	

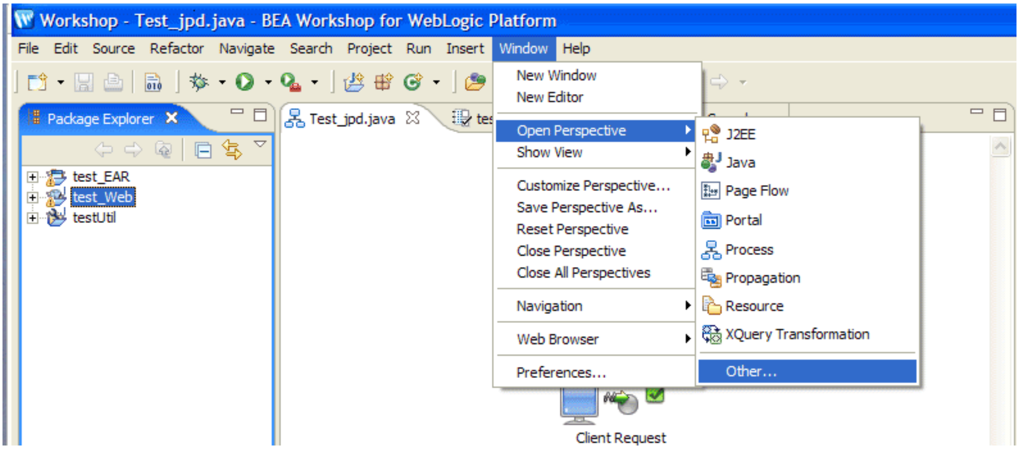
Perspectives

Perspectives define the initial set of Views and their associated layout in the Workbench. A perspective provides a set of functions. You can configure perspectives to define a collection of views, their layouts, and actions for specific tasks as in Figure 11.

The WebLogic Integration perspectives are as follows:

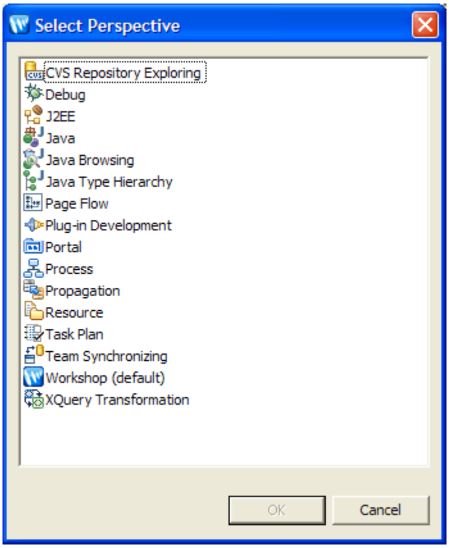
- Process
- Task Plan
- XQuery Transformation

Figure 11 Perspectives



Select **Window→Open Perspective→Other...** to obtain a list of perspectives as shown in Figure 12.

Figure 12 Select Perspective



Business Process Management

WebLogic Integration 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.

As you design business processes using the graphical tools (**Design** view) in BEA Workshop for WebLogic Platform, it writes source code to a business process file. When you do need to write Java code, it is always available with single click access (**Source** view). The business process management 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 end-user decision makers.

Table 3 lists the key features of WebLogic Integration business process management.

Table 3 Business Process Management Features

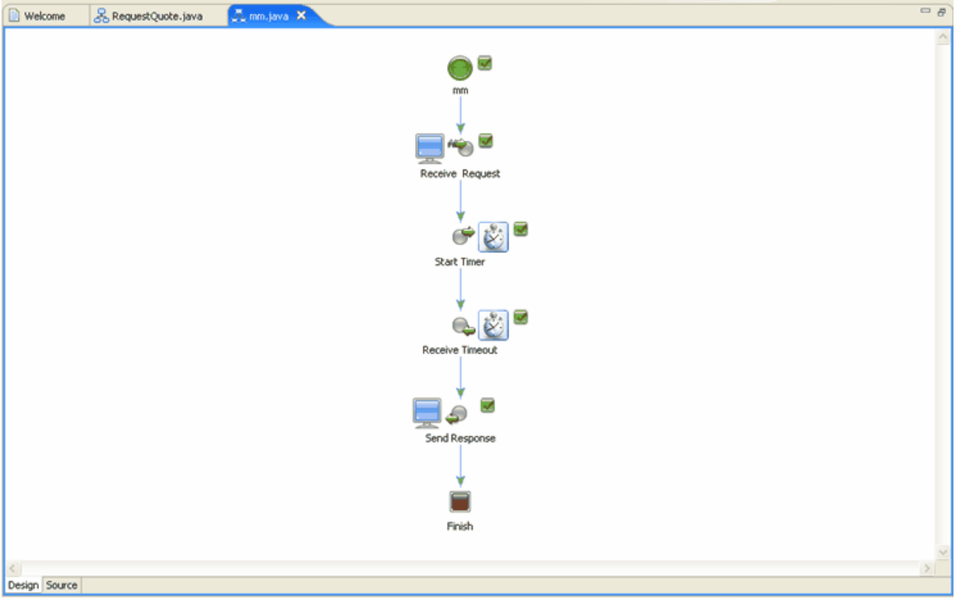
Key Feature	Description
Unified access to resources through Controls	• View business activities as services and orchestrate them as using a business process • Business process seamlessly interacts with users, applications, back-end resources, and resources inside and outside the firewall
New simplified structured business processes	• XML for the flow • Java for the operations
Graphical business process editing for high-level integration scenarios	• Message-based, transformation routing • Business Processes are Java classes. Business process files contain the metadata that describes the business process logic. • Two-way editing in Design and Source view

Table 3 Business Process Management Features (Continued)

Key Feature	Description
Support for Java code in business process nodes	• One click away from Java coding
Process implementation optimization for performance	Support for: • Stateless synchronous processes • Stateless asynchronous processes • Stateful asynchronous processes

WebLogic Integration allows you to create a business process by simply dragging and dropping icons into a logical flow. This approach allows you to define the overall application logic without having to focus on implementation details. Figure 13 illustrates the visual business process editor.

Figure 13 Graphical Representation of Business Process



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 features 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 and made available as resources to other applications and application components.

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.

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 a control and data transformations can be packaged as controls and re-used across multiple business processes and applications.

In BEA Workshop for WebLogic Platform 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, a standards-based query language with the familiar simplicity of SQL-like expressions and a easy-to-use data mapping tool.

WebLogic Integration features a powerful visual data mapping tool, the *XQuery Mapper*, that gives you the ability to generate complex transformations easily with a drag-and-drop simplicity. Figure 14 illustrates the XQuery Mapper. The mapper functionality of BEA Workshop for WebLogic Platform enables the conversion of data of different types. For example, XML data can be transformed from XML data valid to one XML Schema to another XML document valid to a different XML Schema.

Figure 14 XQuery Mapper

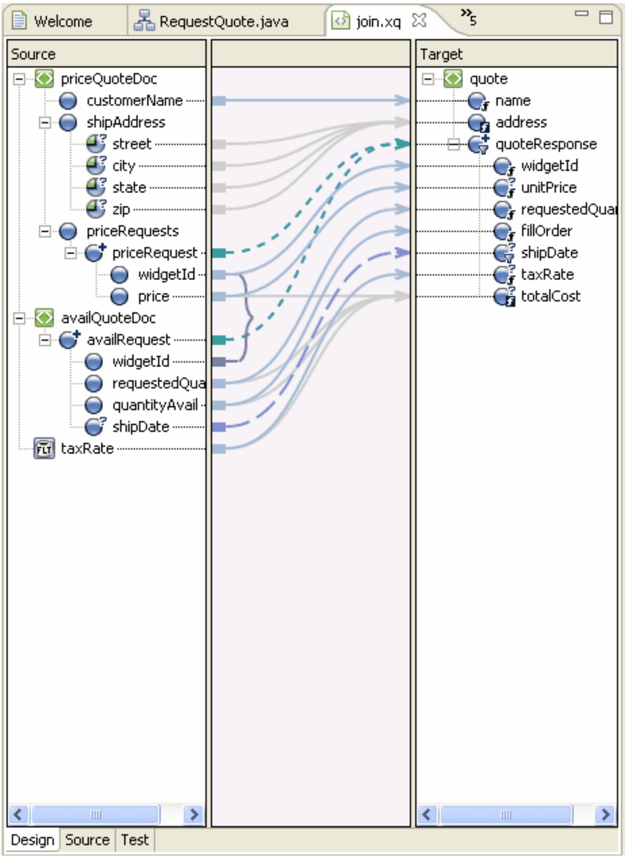


Table 4 lists the key features of data transformation:

Table 4 Data Transformation Features

Feature	Description
Data transformation	• Transformations packaged as controls that can be treated as resources and reused across multiple processes and integration solutions • Transformation of 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, and unions. • Enables transformation of XML grammars
Integration with business processes	• Enables the transformation of data in a business process using transformations written in XQuery or XSLT languages • Transform data received as an incoming message into the business process • Transform data before the business process sends an outgoing message • Transform data inside the business process
XQuery Mapper	• 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
Format Builder tool	• Enables creation of metadata to describe non-XML data

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 can subscribe 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 a couple of different types of listeners. Consumers, such as business processes and other back end resources, can subscribe to Message Broker channels. In this way, the Message Broker facilitates a loosely coupled interface. At run time, you can add new publishers and new subscribers.

The Message Broker supports Event Generators that can publish events from external sources to Message Broker channels. WebLogic Integration supports E-mail, File, JMS, HTTP, MQ Series, RDBMS, TIBCO RV, and Timer event generators. WebLogic Integration adapters, hosted in the Application Integration framework, publish events from packaged applications to channels. Table 5 lists the features of Message Broker:

Table 5 Message Broker Features

Feature	Description
Publish and Subscribe	• 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 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 can be used to refine message selection on subscription.
Message Broker Channels	• 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.

Table 5 Message Broker Features

Feature	Description
Message Routing	• Routing from Event Generators to stateless business processes. • Single transaction (No need to save state).
Rules	• Business process logic defines trigger, transformation, and routing rules. • Short-running rules not long-running processes. • Stateless business processes implement *rules*.

Event Generators

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, MQ Series, 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, see "Event Generators" in *Managing WebLogic Integration Solutions*.

Table 6 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.

Table 6 Types of Event Generators

Event Generator	Description
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 6 Types of Event Generators

Event Generator	Description
MQ Series	MQ Series is the messaging service queue provided by IBM. The MQ Series event generator polls this message queue for any messages and publishes them through Message Brokers channels. **Disclaimer** In the WebLogic Integration product, BEA provides and supports MQ Series (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 MQ Series* To learn more, see "Async Binary Update Sample" available at http://e-docs.bea.com/wli/docs81/sol_samples/async_binary.html *BEA WebLogic Integration MQ Series Control within the WebLogic Workshop IDE* To learn more, see "MQ Series Control" in *Using Integration Controls* available at http://e-docs.bea.com/workshop/docs81/doc/en/integration/controls/controlsMQ.html *BEA Adapter for MQ Series* To learn more, see *BEA WebLogic Adapter for MQ Series 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 MQ Series APIs. IBM has not currently certified the MQ Series APIs for BEA WebLogic Integration 8.1 and 8.5 products. To resolve issues for any aspect of the MQ Series 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 work around. 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_solaris_5_3.html.

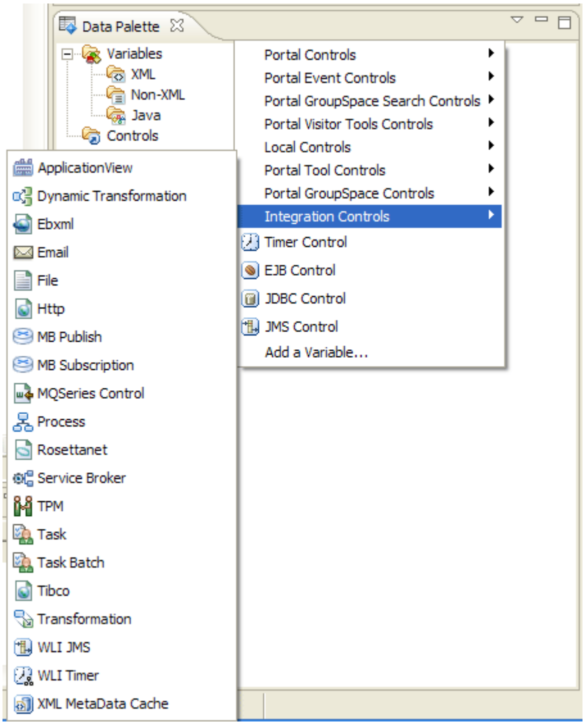
Table 6 Types of Event Generators

Event Generator	Description
TIBCO RV	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. **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.

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 15 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. BEA Workshop for WebLogic Platform provides a consistent mechanism for interacting with resources across all BEA Workshop for WebLogic Platform, WebLogic Integration, and WebLogic Portal components.

Figure 15 Out-of-the Box Controls



WebLogic Integration controls are now based on the Apache Beehive open source framework (http://beehive.apache.org/). Beehive is a lightweight component framework that helps programmers encapsulate application logic and leverage metadata annotations into their programming model. Developers can create custom controls or use the pre-built system controls.

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

The following sections describe the controls delivered with WebLogic Integration:

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.

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.

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.

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.

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.

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.

Message Broker Publish and 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.

Two Message Broker controls, Publish and 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.

MQ Series Control

MQ Series 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 MQ Series control enables WebLogic Integration business processes to send and receive messages using MQ Series queues. Using the MQ Series control, you can send and receive Binary, XML, and String messages. You can specify MQ Series control properties while configuring the MQ Series control or dynamically at run-time. By default, the MQ Series 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).

In the WebLogic Integration product, BEA provides and supports MQ Series (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/docs81/manage/evntgen.html

Asynchronous Sample Illustrating the use of JMS with MQ Series

To learn more, see "Async Binary Update Sample" available at http://e-docs.bea.com/wli/docs81/sol_samples/async_binary.html

BEA WebLogic Integration MQ Series Control within the WebLogic Workshop IDE

To learn more, see "MQ Series Control" in *Using Integration Controls* available at http://e-docs.bea.com/workshop/docs81/doc/en/integration/controls/controlsMQ.html

BEA Adapter for MQ Series

To learn more, see *BEA WebLogic Adapter for MQ Series 8.1 Documentation* available at http://e-docs.bea.com/wladapters/mq/docs81/index.html

Disclaimer

The WebLogic Integration Control and Adapter rely on use of the MQ Series APIs. IBM has not currently certified the MQ Series APIs for BEA WebLogic Integration. To resolve issues for any aspect of the MQ Series 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 work around. 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_solaris_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. 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 process file, a control extension file is automatically created for that control. Double-click this control extension 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.

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.

Service Broker Control

The Service Broker control allows a business process to send requests to and receive callbacks from another business process or business process defined in a Web Service Description Language (WSDL) file.

The Service Broker Control has been included in WebLogic Integration 9.2 for backward compatibility with WebLogic Integration 8.1. The Service Broker Control in 9.2 does not support the latest version of web services standards, such as WS-Addressing, WS-Security, and SOAP 1.2. In order to take advantage of the latest web services standards, please use the web service control delivered with BEA Workshop for WebLogic Platform.

Task and Task Batch Controls

The Task control and the Task Batch 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. For more information, see Worklist System.

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.

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.

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.

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.

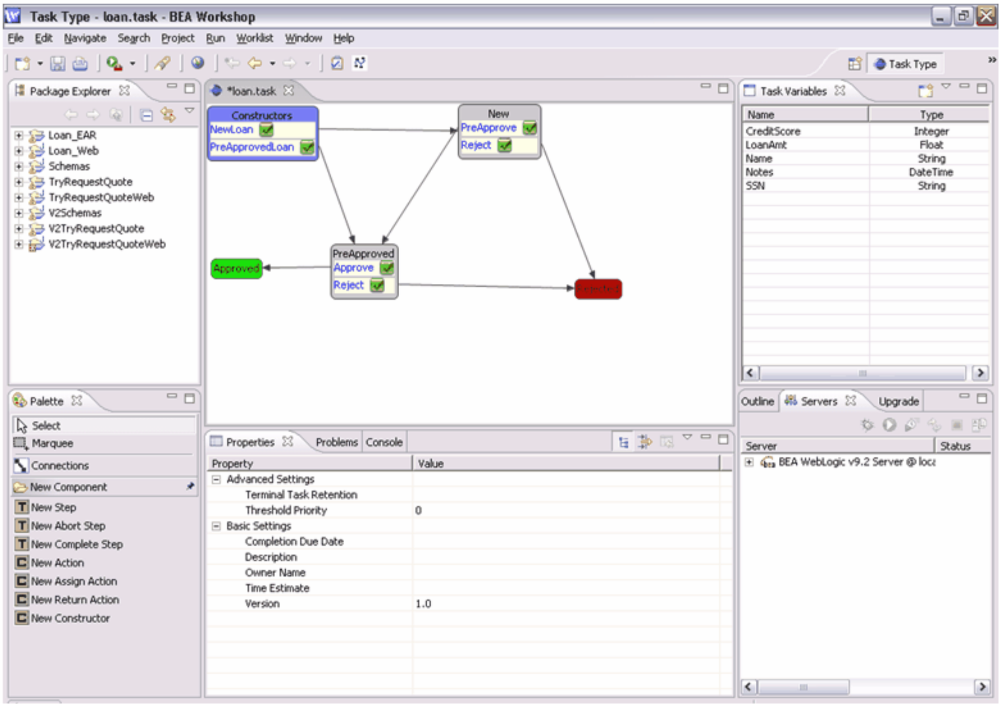
Worklist System

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 people within an enterprise. It enables people to collaborate in business processes including assigning tasks, tracking the status of tasks, handling approvals, and more. Actions such as receiving, approving, modifying, and routing documents are integral to the flow of work. The documents that often accompany work activities provide the background required for people to complete tasks.

A task is the central element within Worklist. In Worklist 9.2, we allow designers can string together tasks that are made up of multiple steps into a Task Plan. Each step in a task plan represents a distinct phase toward the completion of an overall business goal, such as approving a loan request. As business users work on each step, they can choose between one or more actions whose completion will advance them to the appropriate next step based on the rules and conditions defined in the Task Plan. WebLogic Integration provides a drag-and-drop design environment for you to define Task Plans that will be available to business users. The design canvas is delivered as a "perspective" within BEA Workshop for WebLogic Platform, as shown in Figure 16.

Business processes can deal with task instances via controls and message broker. The Task Control can be used to interact with a single task instance in a business process. The Task Batch Control which replaces the Task Worker Control in 8.x, can be used to interact with multiple task instances with queries and bulk operations. In WebLogic Integration 9.2, the Worklist events related to any task instance are published to a Message Broker channel. This allows a business process to subscribe to and filter specific events to act on them, enabling a loosely-coupled integration between the Worklist and the business process.

Figure 16 Task Plan Perspective used in Worklist applications

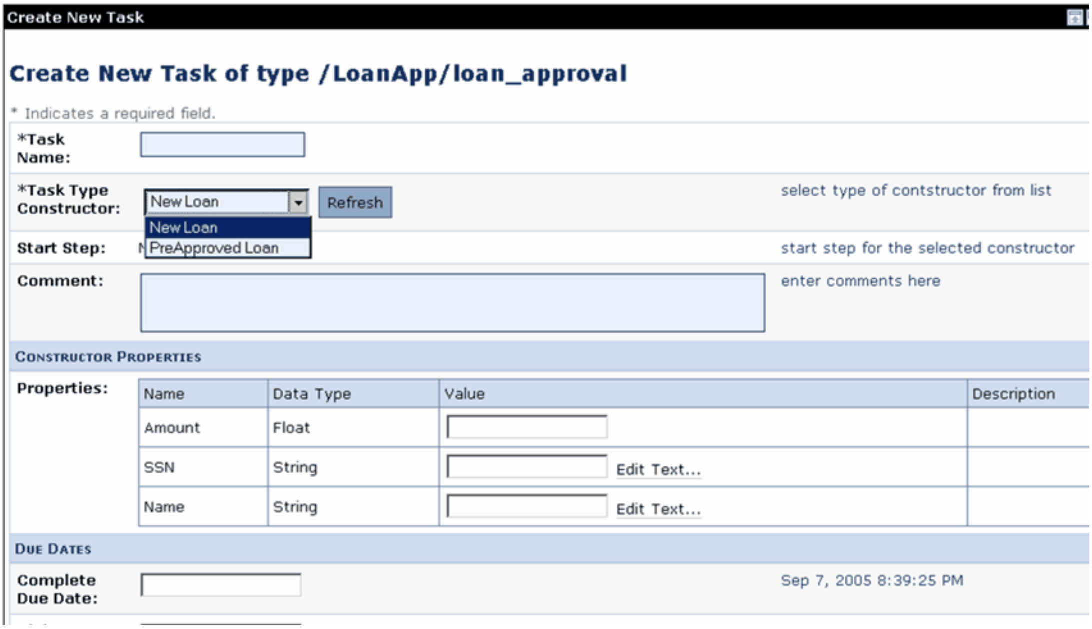


Name	Type
CreditScore	Integer
LoanAmt	Float
Name	String
Notes	DateTime
SSN	String

Property	Value
Advanced Settings	
Terminal Task Retention	
Threshold Priority	0
Basic Settings	
Completion Due Date	
Description	
Owner Name	
Time Estimate	
Version	1.0

As you create your Task Plan, WebLogic Integration automatically generates an end-user facing form that can be used to capture data from business users. It is possible to customize these forms to better suit the look and feel of your environment. Figure 17 shows an example of a generated form.

Figure 17 Automatically Generated Form for Capturing end User Data



Create New Task

Create New Task of type /LoanApp/loan_approval

* Indicates a required field.

*Task Name:		
*Task Type Constructor:	New Loan New Loan PreApproved Loan Refresh	select type of contstructor from list
Start Step:	N	start step for the selected constructor
Comment:		enter comments here

CONSTRUCTOR PROPERTIES

Properties:

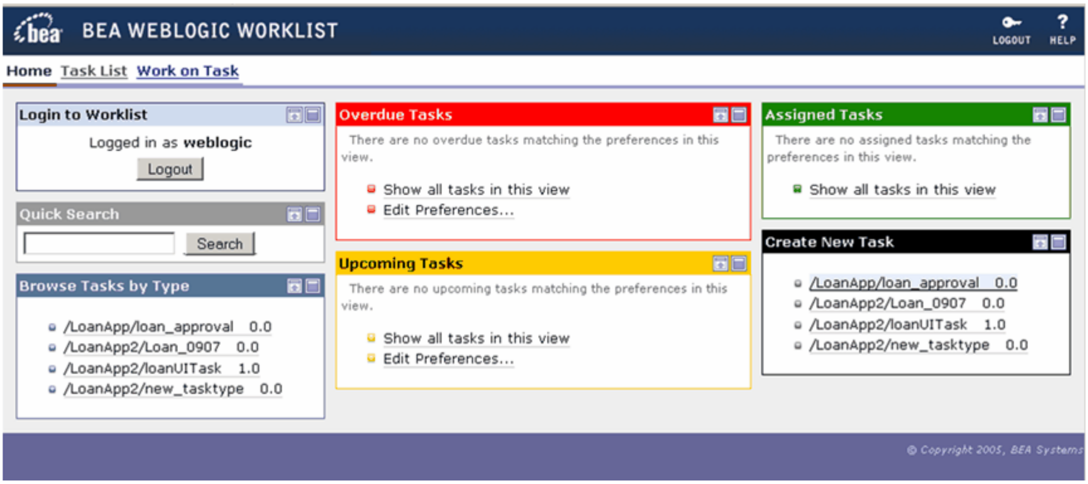
Name	Data Type	Value	Description
Amount	Float		
SSN	String	Edit Text...	
Name	String	Edit Text...	

DUE DATES

Complete Due Date:		Sep 7, 2005 8:39:25 PM

WebLogic Integration also contains a set of out-of-the-box portlets that business users can use to manage their tasks. There is a **User** and a **Manager** portal. The Worklist User Portal provides facilities for end-users to claim tasks, alter task properties and take actions on tasks. The Worklist Manager Portal allows managers to supervise the work done by the users that they manage. The Manager Portal provides peer group views of tasks, their type, and status. It also allows the manager to view the details of subsets of tasks. Figure 18 shows the Worklist User Portal and Figure 19 shows the Worklist Manager Portal. You can customize these portals or embed one or more of these portlets into your own portal.

Figure 18 Worklist User Portal



BEA WEBLOGIC WORKLIST

LOGOUT HELP

Home Task List Work on Task

Login to Worklist

Logged in as **weblogic**

Logout

Quick Search

Search

Browse Tasks by Type

- /LoanApp/loan_approval 0.0
- /LoanApp2/Loan_0907 0.0
- /LoanApp2/loanUITask 1.0
- /LoanApp2/new_tasktype 0.0

Overdue Tasks

There are no overdue tasks matching the preferences in this view.

- Show all tasks in this view
- Edit Preferences...

Upcoming Tasks

There are no upcoming tasks matching the preferences in this view.

- Show all tasks in this view
- Edit Preferences...

Assigned Tasks

There are no assigned tasks matching the preferences in this view.

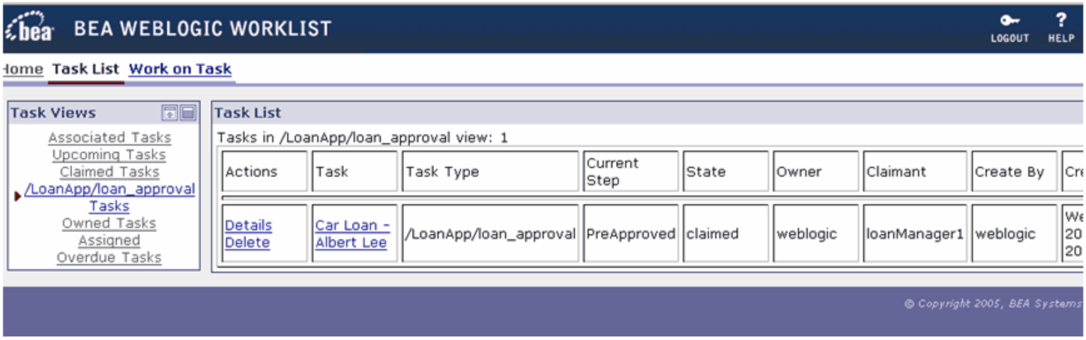
- Show all tasks in this view

Create New Task

- /LoanApp/loan_approval 0.0
- /LoanApp2/Loan_0907 0.0
- /LoanApp2/loanUITask 1.0
- /LoanApp2/new_tasktype 0.0

© Copyright 2005, BEA Systems

Figure 19 Worklist Manager Portal



BEA WEBLOGIC WORKLIST

LOGOUT HELP

Home Task List Work on Task

Task Views

- Associated Tasks
- Upcoming Tasks
- Claimed Tasks
- /LoanApp/loan_approval Tasks
- Owned Tasks
- Assigned
- Overdue Tasks

Task List

Tasks in /LoanApp/loan_approval view: 1

Actions	Task	Task Type	Current Step	State	Owner	Claimant	Create By	Cr
Details Delete	Car Loan - Albert Lee	/LoanApp/loan_approval	PreApproved	claimed	weblogic	loanManager1	weblogic	We 20 20

© Copyright 2005, BEA Systems

For more information on Worklist, see About WebLogic Integration Worklist.

Trading Partner Integration

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

Table 7 summarizes WebLogic Integration's trading partner integration capabilities.

Table 7 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 BEA Workshop for WebLogic Platform to provide 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, audit, and reporting capabilities to show a top-down view of trading partner activities and business transactions across the value chain.
High performance and availability	WebLogic Integration provides fast and reliable business message exchanges between trading partners, supporting the clustered configuration for scalability and fail-over, message persistence, low-level acknowledgements and receipts, and transactional integrity.

Table 7 Trading Partner Integration in WebLogic Integration (Continued)

Feature	Description
High security, auditing, and non-repudiation	WebLogic Integration ensures the 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 kept in protected keystores while the passwords are kept in encrypted forms in the WebLogic Integration PasswordStore.
Interoperability	WebLogic Integration interoperates with a wide range of B2B servers from other vendors. For trading partners who want a zero-install solution, WebLogic Integration can easily be extended to offer a browser or FTP interface.

Figure 20 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.

Figure 20 Interactive Business Processes Between Trading Partners

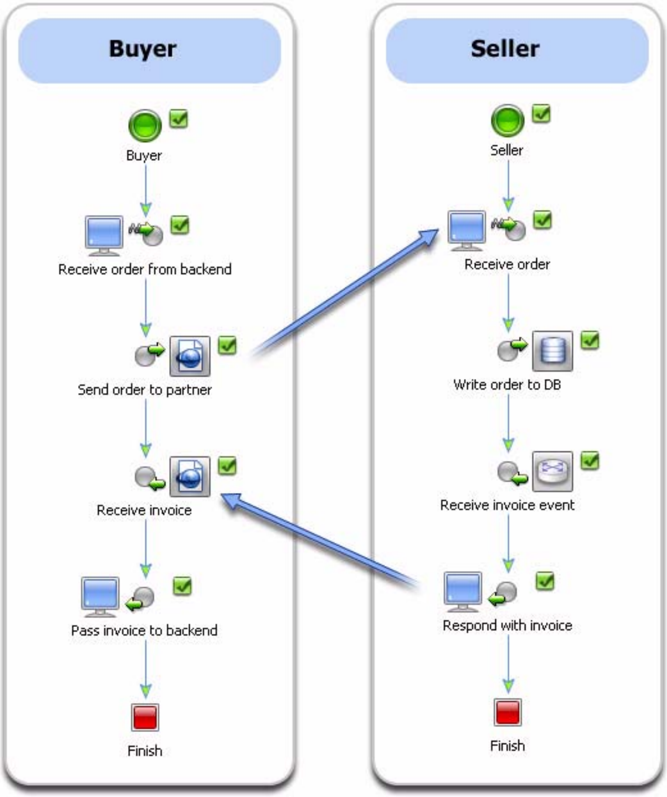
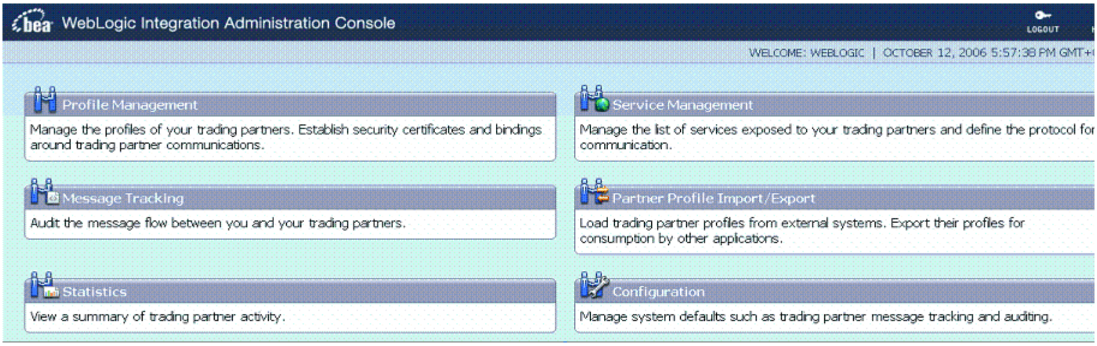


Figure 21 shows the Trading Partner Management home page in the WebLogic Integration Administration Console, which allows administrators to manage trading partner profiles, security certificates, protocol bindings, services, message tracking and auditing, trading partner activity, system defaults, and importing and exporting trading partner profile information.

Figure 21 Trading Partner Management in the WebLogic Integration Administration Console



bea
WebLogic Integration Administration Console
LOGOUT
WELCOME: WEBLOGIC | OCTOBER 12, 2006 5:57:38 PM GMT+

Profile Management

Manage the profiles of your trading partners. Establish security certificates and bindings around trading partner communications.

Service Management

Manage the list of services exposed to your trading partners and define the protocol fo communication.

Message Tracking

Audit the message flow between you and your trading partners.

Partner Profile Import/Export

Load trading partner profiles from external systems. Export their profiles for consumption by other applications.

Statistics

View a summary of trading partner activity.

Configuration

Manage system defaults such as trading partner message tracking and auditing.

Application Integration and Adapters

WebLogic Integration provides an application integration framework comprised of 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 particular protocol or client API for the enterprise application.

The Application 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 Peoplesoft, SAP, Siebel, and Oracle Applications. These adapters are exposed to the BEA Workshop for WebLogic Platform 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.

Table 8 lists the key features of application integration.

Table 8 Application Integration Features

Key Feature	Description
Application Integration	• Standards-based architecture for hosting J2EE CA-based adapters. • Exposed as Application View control in the business process.
JSP-based Application Integration Design Console	• Enables the development and management of Application Views. • Allows EIS specialist to configure adapters via application introspection. • Defines application view services and events.

Table 8 Application Integration Features (Continued)

Key Feature	Description
Adapters	• 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 Oracle Applications, PeopleSoft, SAP, and Siebel applications.
XML MetaData Cache Management	• Global, domain-wide cache. Data maintained in this cache can be accessed by any business process that is deployed in that domain. Use the cache to share data within a cluster. Use the WebLogic Integration Administration Console or the MBean API to manage this cache, that allows users to create their own NetUI-based consoles. • Maintains 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. • Creates a new XML MetaData Cache file for each XML document added to the cache. The Administration Server needs to be up and running if an administrator wants to modify the cache. • Contains XML MetaData Cache controls that are defined in a business process to retrieve data stored in the cache. The controls use the key to retrieve XML metadata associated with the key value, from the cache. • For more information, see XML Cache in Managing WebLogic Integration Solutions.

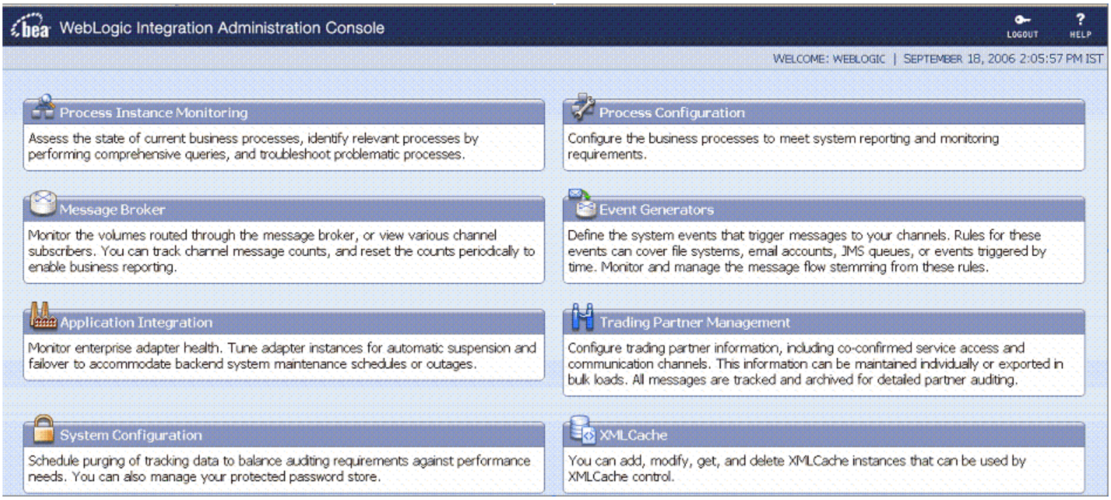
Administration and Management

WebLogic Integration provides a simplified secure browser-based administration console for run-time management and administration analytics. The WebLogic Administration Console allows centralized configuration, maintenance, and monitoring of integrated resources, including audit trails as well as associated security and role information. It is extensible with JMX interfaces to third-party tools. The WebLogic Integration Administration 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 analytics 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 a 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 by third-party tools through SQL to archived databases. Task history is maintained in a separate offline store (called the Worklist reporting store) from the JPD state. This store can either be the default reporting store provided with the Worklist, or a custom reporting store that you provide.

Using the WebLogic Integration Administration Console, you can monitor and manage process flows, message broker activities, application views, and trading partners in one place. Figure 22 illustrates the WebLogic Integration Administration Console.

Figure 22 Browser-based Administration Console



WELCOME: WEBLOGIC | SEPTEMBER 18, 2006 2:05:57 PM IST

Process Instance Monitoring

Assess the state of current business processes, identify relevant processes by performing comprehensive queries, and troubleshoot problematic processes.

Process Configuration

Configure the business processes to meet system reporting and monitoring requirements.

Message Broker

Monitor the volumes routed through the message broker, or view various channel subscribers. You can track channel message counts, and reset the counts periodically to enable business reporting.

Event Generators

Define the system events that trigger messages to your channels. Rules for these events can cover file systems, email accounts, JMS queues, or events triggered by time. Monitor and manage the message flow stemming from these rules.

Application Integration

Monitor enterprise adapter health. Tune adapter instances for automatic suspension and failover to accommodate backend system maintenance schedules or outages.

Trading Partner Management

Configure trading partner information, including co-confirmed service access and communication channels. This information can be maintained individually or exported in bulk loads. All messages are tracked and archived for detailed partner auditing.

System Configuration

Schedule purging of tracking data to balance auditing requirements against performance needs. You can also manage your protected password store.

XMLCache

You can add, modify, get, and delete XMLCache instances that can be used by XMLCache control.

The WebLogic Integration Administration Console addresses the following operations, management, and administration features as listed in Table 9

Table 9 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.
Trading Partner Management	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.
System Configuration	Set general configuration and security. Configure security for processes (both authorization and execution policies), channels, and trading partner communications.
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.
Application Integration	Monitor enterprise adapter health, tune adapters for automatic suspension and failover, and swap adapters to leverage different enterprise application servers.
XML Metadata Cache	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/docs81/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/docs81/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 (`wliconsfig/*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/docs81/manage/system.html.

BPEL Import and Export

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 `.bpel` 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 business process 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, review and modify the imported and exported files based on requirements, to make sure that they run properly.

BPEL Import Tool

You can use the BPEL Import tool to import a BPEL 1.1 compliant file into a business process file, where it can be used in the BEA Workshop for WebLogic Platform design environment. While the main orchestration logic of the BPEL file is imported into a business process file, it is not expected that the imported business process file will be immediately executable in BEA Workshop for WebLogic Platform. You will need to manipulate the business process file in WebLogic Integration Eclipse IDE before it can run as a business process.

BPEL Export Tool

You can use the BPEL Export tool to export the semantics of a business process 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 business process 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.

Related Documents

Table 10 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 10 WebLogic Integration Topics

To learn about...	See the following...

Developing business processes	*Tutorial: Building Your First Business Process*, available in the BEA Workshop for WebLogic Platform Help at:
	`http://edocs.bea.com/workshop/docs92/doc/en/integration/tutorial/tutWLIProcessIntro.html`
	Guide to Business Processes, available in the BEA Workshop for WebLogic Platform online help at:
	`http://edocs.bea.com/workshop/docs92/doc/en/integration/wfguide/wfguideIntro.html`

Implementing data transformations	*Building Your First Data Transformation Tutorial*, available in the BEA Workshop for WebLogic Platform Help at:
	`http://edocs.bea.com/workshop/docs92/doc/en/integration/dttutorial/tutWLIDataTransIntro.html`
	Guide to Data Transformation, available in the BEA Workshop for WebLogic Platform Help at:
	`http://edocs.bea.com/workshop/docs92/doc/en/integration/dtguide/dtguideIntro.html`

Using WebLogic Integration controls	*Using Integration Controls*, available in the BEA Workshop for WebLogic Platform Help at: `http://edocs.bea.com/workshop/docs92/doc/en/integration/controls/controlsIntro.html`

Adding worklist capabilities to a business process	*Using the Worklist System*, available at:
	`http://edocs.bea.com/wli/docs92/worklist/index.html`
	Tutorial: Building a Worklist Application, available at:
	`http://edocs.bea.com/wli/docs92/wltutorial/index.html`

Table 10 WebLogic Integration Topics

To learn about...	See the following...
Trading partner integration	*Introducing Trading Partner Integration*, available at: `http://edocs.bea.com/wli/docs81/tpintro/index.html`

B2B controls	*RosettaNet Control*, available in the BEA Workshop for WebLogic Platform Help at:
	`http://edocs.bea.com/workshop/docs81/doc/en/integration/controls/controlsRosettaNet.htm`
	ebXML Control, available in the BEA Workshop for WebLogic Platform Help at:
	`http://edocs.bea.com/workshop/docs81/doc/en/integration/controls/controlslebXML.html`
	TPM Control, available in the BEA Workshop for WebLogic Platform Help at:
	`http://e-docs.bea.com/workshop/docs81/doc/en/integration/controls/controlsTPM.html`

Managing integration applications	*Managing Integration Solutions*, available at: `http://edocs.bea.com/wli/docs81/manage/index.html`

Application integration	*Introducing Application Integration*, available at:
	`http://edocs.bea.com/wli/docs81/aiover/index.html`
	Using Application Integration Design Console, available at:
	`http://edocs.bea.com/wli/docs81/aiuser/index.html`

Creating custom adapters	*Developing Adapters*, available at: `http://edocs.bea.com/wli/docs81/devadapt/index.html`
Connecting your EIS with BEA WebLogic Adapters	BEA WebLogic Adapters 9.2 online documentation, available at the following location, a set of online manuals for all BEA package adapters for enterprise application integration: `http://edocs.bea.com/wladapters/docs92/index.html`
Upgrading from BEA WebLogic 7.0 versions to WebLogic 9.2	*WebLogic Integration 9.2 Upgrade Guide*, available at: `http://e-docs.bea.com/wli/docs92/upgrade/index.html`
Deploying WebLogic Integration applications	*Deploying WebLogic Integration Solutions*, available at: `http://edocs.bea.com/wli/docs81/deploy/index.html`