



BEA WebLogic Integration Adapter for J. D. Edwards® OneWorld®

User Guide

Version 8.1.1
Document Revised: October 2003

Copyright

Copyright © 2003 BEA Systems, Inc. All Rights Reserved.

Portions Copyright © 2003 iWay Software. All Rights Reserved.

Restricted Rights Legend

This software and documentation is subject to and made available only pursuant to the terms of the BEA Systems License Agreement and may be used or copied only in accordance with the terms of that agreement. It is against the law to copy the software except as specifically allowed in the agreement. This document may not, in whole or in part, be copied photocopied, reproduced, translated, or reduced to any electronic medium or machine readable form without prior consent, in writing, from BEA Systems, Inc.

Use, duplication or disclosure by the U.S. Government is subject to restrictions set forth in the BEA Systems License Agreement and in subparagraph (c)(1) of the Commercial Computer Software-Restricted Rights Clause at FAR 52.227-19; subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013, subparagraph (d) of the Commercial Computer Software--Licensing clause at NASA FAR supplement 16-52.227-86; or their equivalent.

Information in this document is subject to change without notice and does not represent a commitment on the part of BEA Systems. THE SOFTWARE AND DOCUMENTATION ARE PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND INCLUDING WITHOUT LIMITATION, ANY WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. FURTHER, BEA Systems DOES NOT WARRANT, GUARANTEE, OR MAKE ANY REPRESENTATIONS REGARDING THE USE, OR THE RESULTS OF THE USE, OF THE SOFTWARE OR WRITTEN MATERIAL IN TERMS OF CORRECTNESS, ACCURACY, RELIABILITY, OR OTHERWISE.

Trademarks or Service Marks

BEA, Jolt, Tuxedo, and WebLogic are registered trademarks of BEA Systems, Inc. BEA Builder, BEA Campaign Manager for WebLogic, BEA eLink, BEA Liquid Data for WebLogic, BEA Manager, BEA WebLogic Commerce Server, BEA WebLogic Enterprise, BEA WebLogic Enterprise Platform, BEA WebLogic Express, BEA WebLogic Integration, BEA WebLogic Personalization Server, BEA WebLogic Platform, BEA WebLogic Portal, BEA WebLogic Server, BEA WebLogic Workshop and How Business Becomes E-Business are trademarks of BEA Systems, Inc.

All other trademarks are the property of their respective companies.

Contents

About This Document

Who Should Read This Documentation.	viii
Additional Information	viii
How to Use This Document.	ix
Contact Us!	x
Documentation Conventions	xi

Introducing the BEA WebLogic Adapter for J.D. Edwards OneWorld

About the BEA WebLogic Adapter for J.D. Edwards OneWorld.	1-1
Supported J. D. Edwards OneWorld Operations for Application Integration	1-2
Executing J. D. Edwards OneWorld Business Functions	1-2
Accessing Data Stored in J. D. Edwards OneWorld	1-2
Supported Services	1-3
Propagating External Events into J. D. Edwards OneWorld	1-3
Accessing Services with the Interoperability Framework.	1-3
Supported Events	1-4
Accessing Events with the Interoperability Framework	1-5
Benefits of the Adapter for J. D. Edwards OneWorld	1-6
Getting Started With the Adapter for J. D. Edwards OneWorld	1-7
Step 1: Design the Application Integration Solution	1-7
Step 2: Determine the Required J. D. Edwards OneWorld Business Workflows.	1-8

Step 3: Generate Schemas for J. D. Edwards OneWorld Integration Objects	1-9
Step 4: Define Application Views and Configure Services and Events.	1-9
Step 5: Integrate with Other BEA Software Components	1-10
Step 6: Deploy the Solution to the Production Environment.	1-10

Generating Schemas for J. D. Edwards OneWorld Integration Objects

Before You Begin	2-2
Generating Java Wrappers for your J. D. Edwards OneWorld Objects	2-2
About the BEA Application Explorer	2-2
Types of Schemas Generated by the BEA Application Explorer	2-3
Service Requests	2-3
Service Responses	2-3
Events	2-3
About the Process for Defining Schemas	2-3
Starting the BEA Application Explorer	2-4
Setting the Session Path	2-4
Managing J. D. Edwards OneWorld Connections	2-5
Creating a New Connection	2-5
Using an Existing Connection	2-7
Disconnecting from J. D. Edwards OneWorld	2-7
Removing Connections	2-8
Managing Schemas	2-8
Creating Schemas for Services	2-8
Creating Schemas for Events	2-10
Removing Schemas	2-12
Next Steps	2-13

Defining Application Views for J. D. Edwards OneWorld

How to Use This Document	3-2
Before You Begin	3-2
About Application Views	3-3
About Defining Application Views	3-3
Defining Service Connection Parameters	3-5
Setting Service Properties	3-6
Common Service and Event Settings	3-8
Setting Event Properties	3-9
J. D. Edwards OneWorld System Event	3-9
J. D. Edwards OneWorld File Event	3-10
J. D. Edwards OneWorld FTP Event	3-12
Defining Event Connection Parameters	3-14
Testing Services	3-16
Testing Events Using a Service	3-17
Testing Events Manually	3-17
Enabling Outbound Transaction Processing	A-1
Modifying the jde.ini File	A-2
Types of Coexistence	B-1
Advantages of Using Coexistence	B-2
Disadvantages of Using Coexistence	B-2
Installation Guidelines	B-2
Single-Function Request	C-1
Multiple-Function Request	C-2

Index

About This Document

This document describes how to use the BEA WebLogic Adapter for J.D. Edwards OneWorld. This document is organized as follows:

- [Chapter 1, “Introducing the BEA WebLogic Adapter for J.D. Edwards OneWorld,”](#) describes the adapter, how it relates to both J. D. Edwards OneWorld business objects and WebLogic Integration.
- [Chapter 2, “Generating Schemas for J. D. Edwards OneWorld Integration Objects,”](#) describes how to generate schemas for your J. D. Edwards OneWorld business objects using the BEA Application Explorer.
- [Chapter 3, “Defining Application Views for J. D. Edwards OneWorld,”](#) describes application views and how to use them to configure events and services.
- [Appendix A, “Configuring J. D. Edwards OneWorld for Outbound Transaction Processing,”](#) describes how to enable outbound transaction processing in OneWorld and modify the `jde.ini` file for XML support.
- [Appendix B, “Coexistence Between J. D. Edwards OneWorld and J. D. Edwards WorldSoftware,”](#) describes types of coexistence and the advantages and disadvantages of using coexistence.
- [Appendix C, “Sample Files,”](#) provides examples of the `jdeRequest` and `jdeResponse` XML structures for executing business functions within OneWorld.

Who Should Read This Documentation

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

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

Additional Information

To learn more about the software components associated with the adapter, see the following documents:

- *BEA WebLogic Adapter for J. D. Edwards OneWorld Release Notes*
<http://edocs.bea.com/wl.adapters/jde/docs811/pdf/relnotes.pdf>
- *BEA WebLogic Adapter for J. D. Edwards OneWorld Installation and Configuration Guide*
<http://edocs.bea.com/wl.adapters/jde/docs811/pdf/install.pdf>
- *BEA Application Explorer Installation and Configuration Guide*
<http://edocs.bea.com/wl.adapters/bae/docs811/pdf/install.pdf>
- *Introduction to the BEA WebLogic Adapters*
<http://edocs.bea.com/wl.adapters/docs81/pdf/intro.pdf>
- BEA WebLogic Adapters 8.1 Dev2Dev Product Documentation

<http://dev2dev.bea.com/products/wlapters/index.jsp>

- Application Integration documentation

<http://edocs.bea.com/wli/docs81/aiover/index.html>

<http://edocs.bea.com/wli/docs81/aiuser/index.html>

- BEA WebLogic Integration documentation

<http://edocs.bea.com/wli/docs81/index.html>

- BEA WebLogic Platform documentation

<http://edocs.bea.com/platform/docs81/index.html>

- J. D. Edwards OneWorld documentation

<http://www.jdedwards.com>

How to Use This Document

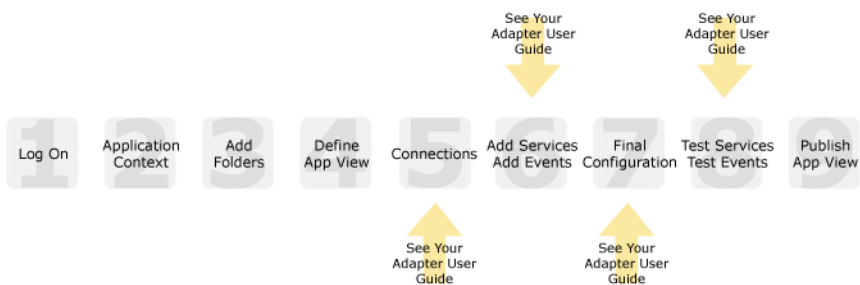
This document is designed to be used in conjunction with *Using the Application Integration Design Console*, available at the following URL:

<http://edocs.bea.com/wli/docs81/aiuser/index.html>

Using the Application Integration Design Console describes, in detail, the process of defining an application view, which is a key part of making an adapter available to process designers and other users. What *Using the Application Integration Design Console* does *not* cover is the specific information about Adapter for J. D. Edwards OneWorld that you need to supply to complete the application view definition. You will find that information in this document.

At each point in *Using the Application Integration Design Console* where you need to refer to this document, you will see a note that directs you to a section in your adapter user guide, with a link to the edocs page for adapters. The following roadmap illustration shows where you need to refer from *Using the Application Integration Design Console* to this document.

Figure 1 Information Interlock with *Using the Application Integration Design Console*



Contact Us!

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

In your e-mail message, please indicate that you are using the documentation for BEA WebLogic Adapter for J.D. Edwards OneWorld and the version of the documentation.

If you have any questions about this version of BEA WebLogic Adapter for J.D. Edwards OneWorld, or if you have problems using the BEA WebLogic Adapter for J.D. Edwards OneWorld, contact BEA Customer Support through BEA WebSUPPORT at www.bea.com. You can also contact Customer Support by using the contact information provided on the Customer Support Card which is included in the product package.

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

- Your name, e-mail address, phone number, and fax number
- Your company name and company address
- Your machine type and authorization codes
- The name and version of the product you are using
- A description of the problem and the content of pertinent error messages

Documentation Conventions

The following documentation conventions are used throughout this document.

Convention	Item
boldface text	Indicates terms defined in the glossary.
Ctrl+Tab	Indicates that you must press two or more keys simultaneously.
<i>italics</i>	Indicates emphasis or book titles.
monospace text	Indicates code samples, commands and their options, data structures and their members, data types, directories, and file names and their extensions. Monospace text also indicates text that you must enter from the keyboard. <i>Examples:</i> <pre>#include <iostream.h> void main () the pointer psz chmod u+w * \tux\data\ap .doc tux.doc BITMAP float</pre>
monospace boldface text	Identifies significant words in code. <i>Example:</i> <pre>void commit ()</pre>
<i>monospace italic text</i>	Identifies variables in code. <i>Example:</i> <pre>String <i>expr</i></pre>
UPPERCASE TEXT	Indicates device names, environment variables, and logical operators. <i>Examples:</i> <pre>LPT1 SIGNON OR</pre>
{ }	Indicates a set of choices in a syntax line. The braces themselves should never be typed.

Convention	Item
[]	<p>Indicates optional items in a syntax line. The brackets themselves should never be typed.</p> <p><i>Example:</i></p> <pre>buildobjclient [-v] [-o name] [-f file-list]... [-l file-list]...</pre>
	<p>Separates mutually exclusive choices in a syntax line. The symbol itself should never be typed.</p>
...	<p>Indicates one of the following in a command line:</p> <ul style="list-style-type: none"> • That an argument can be repeated several times in a command line • That the statement omits additional optional arguments • That you can enter additional parameters, values, or other information <p>The ellipsis itself should never be typed.</p> <p><i>Example:</i></p> <pre>buildobjclient [-v] [-o name] [-f file-list]... [-l file-list]...</pre>
.	<p>Indicates the omission of items from a code example or from a syntax line. The vertical ellipsis itself should never be typed.</p>

Introducing the BEA WebLogic Adapter for J.D. Edwards OneWorld

This section introduces the BEA WebLogic Adapter for J.D. Edwards OneWorld and describes how the adapter enables integration with J. D. Edwards OneWorld business objects and WebLogic Integration.

It includes the following topics:

- [About the BEA WebLogic Adapter for J.D. Edwards OneWorld](#)
- [Getting Started With the Adapter for J. D. Edwards OneWorld](#)

About the BEA WebLogic Adapter for J.D. Edwards OneWorld

The BEA WebLogic Adapter for J.D. Edwards OneWorld connects to your J. D. Edwards OneWorld system so that you can easily use your J. D. Edwards OneWorld data and functions within your business processes. The adapter provides scalable, reliable, and secure access to your J. D. Edwards OneWorld system.

This section includes the following topics:

- [Supported J. D. Edwards OneWorld Operations for Application Integration](#)
- [Supported Services](#)
- [Supported Events](#)
- [Benefits of the Adapter for J. D. Edwards OneWorld](#)

Supported J. D. Edwards OneWorld Operations for Application Integration

The Adapter for J. D. Edwards OneWorld supports synchronous and asynchronous, bi-directional message interactions for J. D. Edwards OneWorld Business Services, Business Components, and Integration Objects.

It provides integration with the following J. D. Edwards OneWorld operations:

- Access to J. D. Edwards OneWorld integration objects using XML to handle both services and events
- Direct invocation of J. D. Edwards OneWorld Master Business Functions by passing an XML document into OneWorld through the J. D. Edwards OneWorld ThinNet API.

The BEA WebLogic Adapter for J.D. Edwards OneWorld supports the following integration access methods:

- J. D. Edwards OneWorld ThinNet API
- J. D. Edwards OneWorld XML
- J. D. Edwards unedited transaction tables (Z tables)

Executing J. D. Edwards OneWorld Business Functions

You can use the BEA WebLogic Adapter for J.D. Edwards OneWorld to invoke a OneWorld business function, such as add account, update account, or delete account. You can also use the adapter as part of an integration effort to connect OneWorld with non-OneWorld systems. The adapter can detect an event in OneWorld by receiving an XML document, or it can execute one or more J. D. Edwards Master Business Functions by passing an XML document into OneWorld through the J. D. Edwards OneWorld ThinNet API.

Accessing Data Stored in J. D. Edwards OneWorld

J. D. Edwards OneWorld provides multiple methods and technologies to provide interoperability. It has three supported entry points:

- Flat files
- Database tables
- Master Business Function (MBF) interactive calls

Supported Services

The Adapter for J. D. Edwards OneWorld supports three types of integration access: J. D. Edwards OneWorld ThinNet API, J. D. Edwards OneWorld XML, and J. D. Edwards OneWorld unedited transaction tables (Z tables). In each case, the adapter sends a J. D. Edwards OneWorld XML file to J. D. Edwards OneWorld to cause a J. D. Edwards OneWorld business event.

Propagating External Events into J. D. Edwards OneWorld

What is a service from the point of view of the adapter is an external event from the point of view of J. D. Edwards OneWorld. When integrating external events into OneWorld using flat file input, the files are imported through a batch program and placed on an unedited transaction table. The records on the transaction table are processed by a batch program, which makes calls to the appropriate MBF.

With the database table method, the process bypasses the first step in the flat file method, and records are written directly to the unedited transaction table. Once they are on the table, a batch process is run, which makes calls to the appropriate MBF.

The third method, calling the MBF directly, bypasses the batch processing altogether and provides synchronous access to OneWorld.

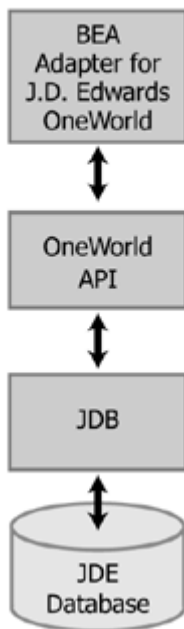
Accessing Services with the Interoperability Framework

Figure 1-1 illustrates the inbound processing framework. In the inbound process, the service adapter uses the J. D. Edwards OneWorld ThinNet API to communicate with the OneWorld application. Using the ThinNet API, the service adapter can execute one or more Master Business Functions (MBFs) in a single Unit Of Work (UOW). If any of the MBFs fail, then the entire UOW fails, preventing partial updates. Because the service adapter executes the MBFs, validation of

data, business rules, and communications to the underlying database are handled by the OneWorld application.

Figure 1-1 J. D. Edwards OneWorld Inbound Processing Framework

J.D. Edwards Inbound Processing Framework



Supported Events

The Adapter for J. D. Edwards OneWorld supports events generated by J. D. Edwards OneWorld. In each case, the adapter picks up an XML file and passes it to an event variable within a business process.

Integrating a J. D. Edwards OneWorld event with external systems is handled in much the same way as the inbound process, except in reverse. The determination of whether a transaction needs to be integrated with an external system is maintained in the Data Export Control table. When a transaction needs to be integrated, the MBF handles logging of all additions, changes, and

deletions to the unedited transaction table. Once the transaction's information is written to the table, a key for that record is sent from the MBF to the subsystem data queue.

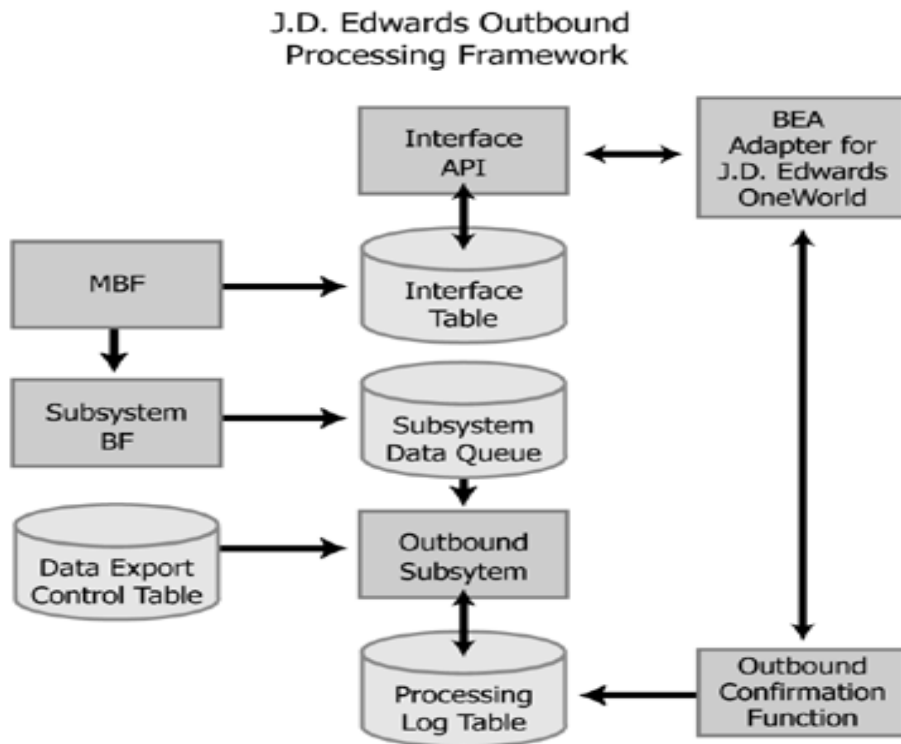
The subsystem data queue triggers the processing of the new record by launching an outbound subsystem batch process, which is generic and handles all outbound transactions. The outbound subsystem then accesses the Data Export Control table to determine the business function or external process to run.

Accessing Events with the Interoperability Framework

Figure 1-2 illustrates the outbound processing framework. In the outbound process, the event starts when a specific MBF is executed in the J. D. Edwards OneWorld environment. The MBF writes the required information for the event into the appropriate interface table and then notifies the subsystem Batch Function (BF) that an event occurred. The subsystem BF then places an entry about the event on the Subsystem Data Queue.

The outbound subsystem retrieves the data queue entry and looks on the Data Export Control table for the external processes to notify. The outbound subsystem then calls the BEA WebLogic Adapter for J.D. Edwards OneWorld event listener with the notification of the event. The event listener passes the notification to the BEA event generator associated with the event. The event generator then uses the J. D. Edwards OneWorld ThinNet API to retrieve the appropriate event information from the interface table.

Figure 1-2 J. D. Edwards OneWorld Outbound Processing Framework



Benefits of the Adapter for J. D. Edwards OneWorld

The combination of the adapter and WebLogic Integration supplies everything you need to integrate your workflows and enterprise applications with your J. D. Edwards OneWorld system. The Adapter for J. D. Edwards OneWorld provides these benefits:

- Integration can be achieved without custom coding.
- Business processes can be started by events generated by J. D. Edwards OneWorld.
- Business processes can request and receive data from your J. D. Edwards OneWorld system using services.

- Adapter events and services are standards-based. The adapter services and events provide extensions to the *J2EE Connector Architecture* (JCA) version 1.0 from Sun Microsystems, Inc. For more information, see the Sun JCA page at the following URL:

<http://java.sun.com/j2ee/connector/>

- The adapter and WebLogic Integration solution is scalable. The BEA WebLogic Platform provides clustering, load balancing, and resource pooling for a scalable solution. For more information about scalability, see the following URL:

<http://edocs.bea.com/wls/docs81/cluster/index.html>

- The adapter and WebLogic Integration solution benefits from the fault-tolerant features of the BEA WebLogic Platform. For more information about high availability, see the following URL:

<http://edocs.bea.com/wli/docs81/deploy/index.html>

- The adapter and WebLogic Integration solution is secure, using the security features of the BEA WebLogic Platform and the security of your J. D. Edwards OneWorld system. For more information about security, see the following URL:

<http://edocs.bea.com/wls/docs81/secintro/index.html>

Getting Started With the Adapter for J. D. Edwards OneWorld

This section gives an overview of how to get started using the BEA WebLogic Adapter for J.D. Edwards OneWorld within the context of an application integration solution. Integration with J. D. Edwards OneWorld involves the following tasks:

- [Step 1: Design the Application Integration Solution](#)
- [Step 2: Determine the Required J. D. Edwards OneWorld Business Workflows](#)
- [Step 3: Generate Schemas for J. D. Edwards OneWorld Integration Objects](#)
- [Step 4: Define Application Views and Configure Services and Events](#)
- [Step 5: Integrate with Other BEA Software Components](#)
- [Step 6: Deploy the Solution to the Production Environment](#)

Step 1: Design the Application Integration Solution

The first step is to design an application integration solution, which includes (but is not limited to) such tasks as:

- Defining the overall scope of application integration.
- Determining the business process(es) to integrate.
- Determining which WebLogic Platform components will be involved in the integration, such as web services or workflows designed in WebLogic Workshop, portals created in WebLogic Portal, and so on.
- Determining which external systems and technologies will be involved in the integration, such as J. D. Edwards OneWorld systems and other EISes.
- Determining which BEA WebLogic Adapters for WebLogic Integration will be required, such as the BEA WebLogic Adapter for J.D. Edwards OneWorld. An application integration solution can involve multiple adapters.

This step involves the expertise of business analysts, system integrators, and EIS specialists (including J. D. Edwards OneWorld specialists). Note that an application integration solution can be part of a larger integration solution.

Step 2: Determine the Required J. D. Edwards OneWorld Business Workflows

Within the larger context of an application integration project, you must determine which specific J. D. Edwards OneWorld integration objects and workflows are required to handle services and events to support the business processes in the application integration solution. Or, if you are invoking J. D. Edwards OneWorld business services or business components directly, rather than through a workflow, you must determine the tasks you need to complete.

Factors to consider include (but are not limited to):

- Type of J. D. Edwards OneWorld integration objects, workflows, and transport used to access the J. D. Edwards OneWorld system.
- J. D. Edwards OneWorld transactions involved in business processes
- Logins required to access J. D. Edwards OneWorld transports and perform the required operations
- Whether operations are, from the adapter point of view:
 - services, which notify the J. D. Edwards OneWorld system, via an XML document, with a request for action, and, in addition, whether such services should be processed synchronously or asynchronously

- events, which are notifications from the J. D. Edwards OneWorld system that trigger workflows

This step involves the expertise of J. D. Edwards OneWorld specialists, including analysts and administrators.

Step 3: Generate Schemas for J. D. Edwards OneWorld Integration Objects

After identifying the J. D. Edwards OneWorld integration objects and workflows required for the application integration solution, you must generate the XML schemas that will be used to exchange data with one or more J. D. Edwards OneWorld systems:

- Services require two XML schemas: one for the J. D. Edwards OneWorld request and another for the J. D. Edwards OneWorld response.
- Events require a single XML schema to handle the data sent by the J. D. Edwards OneWorld system.

You must use the J. D. Edwards OneWorld utility GenJava to generate Java wrappers for your OneWorld objects. Then you use the BEA Application Explorer tool to generate schemas for J. D. Edwards OneWorld objects. To learn more about the GenJava utility, see your J. D. Edwards OneWorld documentation. To learn more about schemas, see [Chapter 2, “Generating Schemas for J. D. Edwards OneWorld Integration Objects.”](#)

Step 4: Define Application Views and Configure Services and Events

After you create the schemas for your J. D. Edwards OneWorld services or events, you create an application view that provides an XML-based interface between WebLogic Server and a particular J. D. Edwards OneWorld system within your enterprise. If you are accessing multiple J. D. Edwards OneWorld systems, you define a separate application view for each J. D. Edwards OneWorld system you want to access. To provide different levels of security access (such as “guest” and “administrator”), define a separate application view for each security level.

Once you define an application view, you can configure events and services in that application view that employ the XML schemas that you created in [“Step 3: Generate Schemas for J. D. Edwards OneWorld Integration Objects” on page 1-9](#). To learn more about generating schemas, see [Chapter 2, “Generating Schemas for J. D. Edwards OneWorld Integration Objects.”](#)

To learn more about defining application views, see [Chapter 3, “Defining Application Views for J. D. Edwards OneWorld”](#) in conjunction with *Using the Application Integration Design Console*, at the following URL:

<http://edocs.bea.com/wli/docs81/aiuser/index.html>

Step 5: Integrate with Other BEA Software Components

Once you have configured and published one or more application views for J. D. Edwards OneWorld integration, you can integrate these application views into other BEA software components, such as workflows or web services created in BEA WebLogic Workshop, or portals built with BEA WebLogic Portal.

For more information, see *Using the Application Integration Design Console*, particularly Chapter 3, “Using Application Views with Application Workflows,” at the following URL:

<http://edocs.bea.com/wli/docs81/aiuser/index.html>

Step 6: Deploy the Solution to the Production Environment

After you have designed, built, and tested your application integration solution, you can deploy it into a production environment. The following list describes some of the tasks involved in deploying an application integration:

- Design the deployment.
- Deploy the required components of the BEA WebLogic Platform.
- Install and deploy the BEA WebLogic Adapter for J.D. Edwards OneWorld as described in *BEA WebLogic Adapter for J. D. Edwards OneWorld Installation and Configuration Guide*
- Deploy your application views and schemas for J. D. Edwards OneWorld integration.
- Verify business processes in the production environment.
- Monitor and tune the deployment.

Generating Schemas for J. D. Edwards OneWorld Integration Objects

The Adapter for J. D. Edwards OneWorld uses XML documents to communicate with your J. D. Edwards OneWorld system's integration objects for both services and events. The format of these XML documents is determined by Java wrappers you generate using the J. D. Edwards GenJava tool. After you generate the Java wrappers, you can use them as input to the BEA Application Explorer to generate the schemas that the adapter will use.

This section explains how to use the BEA Application Explorer to generate schemas. It contains the following topics:

- [Before You Begin](#)
- [Generating Java Wrappers for your J. D. Edwards OneWorld Objects](#)
- [About the BEA Application Explorer](#)
- [Starting the BEA Application Explorer](#)
- [Setting the Session Path](#)
- [Managing J. D. Edwards OneWorld Connections](#)
- [Managing Schemas](#)

Before You Begin

Before you begin to generate schema for the Adapter for J. D. Edwards OneWorld, you must:

- Download and install the BEA Application Explorer software. To learn more, see the *BEA Application Explorer Installation and Configuration Guide* at the following URL:

<http://edocs.bea.com/wladapters/docs81/index.html>

- Obtain the information necessary to connect to your J. D. Edwards OneWorld system. Contact your J. D. Edwards OneWorld administrator for this information.
- Use the J. D. Edwards OneWorld GenJava utility to generate Java wrappers for your J. D. Edwards OneWorld objects. See your J. D. Edwards OneWorld documentation for instructions on how to use GenJava to generate Java wrappers for your J. D. Edwards OneWorld objects.

Note: Only the generation of service schemas requires you to use the GenJava tool to generate the wrappers and the GenJava Repository. For more information on building the J. D. Edwards OneWorld Master Business Function Repository, see the *J. D. Edwards Interoperability Guide for OneWorld Xe*.

Generating Java Wrappers for your J. D. Edwards OneWorld Objects

Before you can generate schemas for your J. D. Edwards OneWorld objects, you must use the J. D. Edwards OneWorld utility GenJava to generate Java wrappers for these objects. See your J. D. Edwards OneWorld documentation for information on using this utility. Once you have generated these wrappers, you can use the BEA Application Explorer to generate schemas for these objects. To learn more about generating schemas, see [About the BEA Application Explorer](#).

About the BEA Application Explorer

The BEA Application Explorer uses intelligence about J. D. Edwards OneWorld combined with metadata provided by J. D. Edwards OneWorld Java wrappers to generate the schemas required to build application view services and events.

This section contains the following topics:

- [About the Process for Defining Schemas](#)
- [Types of Schemas Generated by the BEA Application Explorer](#)

Types of Schemas Generated by the BEA Application Explorer

Each service or event the Adapter for J. D. Edwards OneWorld uses must be defined by a schema. The BEA Application Explorer generates XML schemas for:

- [Service Requests](#)
- [Service Responses](#)
- [Events](#)

Service Requests

Service requests are requests for action that your application makes to your J. D. Edwards OneWorld system. Requests are defined by request schema. As part of the definition, the request schema defines the input parameters required by the J. D. Edwards OneWorld system. The J. D. Edwards OneWorld system responds to the request with a service response.

Service Responses

Service responses are the way the J. D. Edwards OneWorld system responds to a service request. A service response schema defines this service response. Service requests always have corresponding responses.

Events

Events are generated by the J. D. Edwards OneWorld system as a result of activity on that system. You can use these events to trigger an action in your application. For example, the J. D. Edwards OneWorld system may generate an event when customer information is updated. If your application must do something when this happens, your application is a consumer of this event. Events are defined by event schema.

About the Process for Defining Schemas

The process for defining XML schemas includes the following steps:

1. [Starting the BEA Application Explorer.](#)
2. [Setting the Session Path.](#)

The BEA Application Explorer uses this path to create the directory for the schemas.

3. [Creating a New Connection](#) or [Using an Existing Connection.](#)

4. [Creating Schemas for Services](#) and [Creating Schemas for Events](#).

Starting the BEA Application Explorer

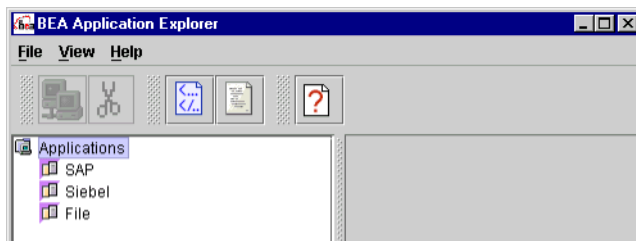
You use the BEA Application Explorer to generate service request schemas, service response schemas, and event schemas. The schemas you create are published in the WebLogic Integration repository.

You must supply the BEA Application Explorer with the location of the integration object you are using.

To start the BEA Application Explorer:

1. Open the BEA Application Explorer.
 - In Windows, choose Windows Start→Programs→BEA Application Explorer.
 - On UNIX, run the startup script `beabse.sh` or the Java command `java com.ibi.common.ui.StartPanel`.

The BEA Application Explorer window appears.



Setting the Session Path

The session path determines the directory where the BEA Application Explorer places your generated XML schemas and connection information. Your schemas are stored here:

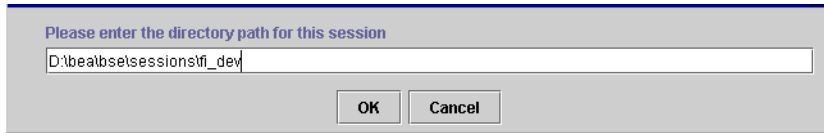
- On Windows:
`session_path\JDEDWARDSOW\connection_name\schemas`
- On UNIX: `session_path/JDEDWARDSOW/connection_name/schemas`

Here, *connection_name* is the value you specify when you select a connection. To learn more about selecting a connection, see [“Managing J. D. Edwards OneWorld Connections.”](#)

To set the session path:

1. From the File menu, choose Session.

The Enter Session Path window appears, displaying a default path.



2. Do one of the following:

- To accept the default session path, click OK.
- To specify a different path, enter the path and click OK.

Specifying a different path allows you to group your schema according to project, or other logical group.

Managing J. D. Edwards OneWorld Connections

The BEA Application Explorer must connect to your J. D. Edwards OneWorld system before you can generate schemas. Therefore, you must first define a connection to your J. D. Edwards OneWorld system.

This section includes the following topics:

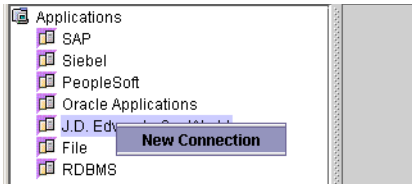
- [Creating a New Connection](#)
- [Using an Existing Connection](#)
- [Disconnecting from J. D. Edwards OneWorld](#)
- [Removing Connections](#)

Creating a New Connection

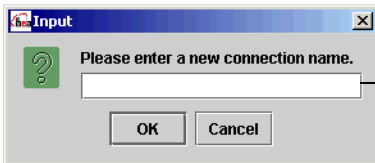
If you are creating a new connection, be sure to check that you have the correct information for your J. D. Edwards OneWorld system.

To create a new connection:

1. In the left pane of the BEA Application Explorer window, under Applications right-click J. D. Edwards OneWorld → New Connection.



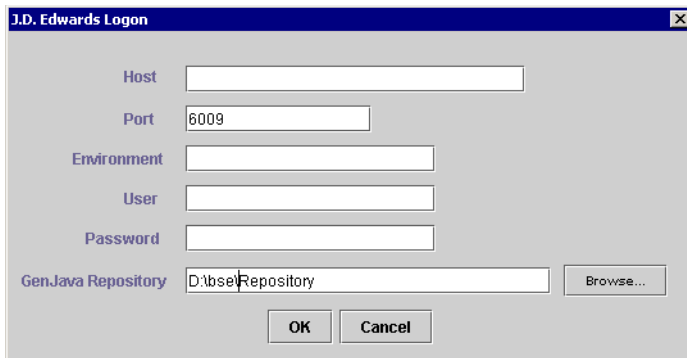
The BEA Application Explorer prompts you for a connection name.



Enter a name for this connection.

2. Enter a name for this connection and click OK.

The J. D. Edwards OneWorld Logon dialog box appears.



3. Enter the parameters for your system.

- Host: The name of the server on which J. D. Edwards OneWorld is running, or its IP address.
- Port: The port number the server is listening on.
- Environment: The J. D. Edwards OneWorld environment.
- User: A valid user ID for J. D. Edwards OneWorld.
- Password: The password associated with the user ID.
- GenJava Repository: The location of the GenJava-generated Master Business Function wrappers.

For more information on these parameters, see your J. D. Edwards OneWorld documentation, or ask your J. D. Edwards OneWorld system administrator.

4. Click OK.

The new connection appears under the J. D. Edwards OneWorld node in the BEA Application Explorer window. You can now view business objects and services, as well as all available integration objects in your J. D. Edwards OneWorld system.

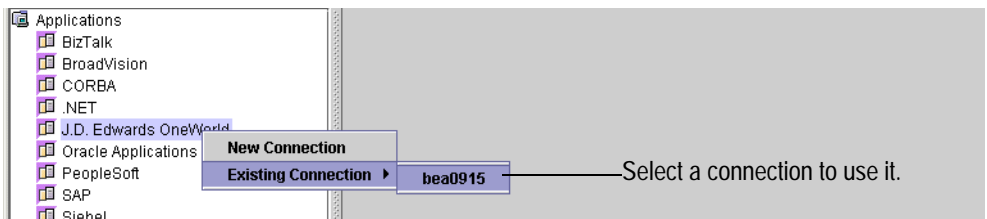
Using an Existing Connection

You can use an existing connection rather than creating a new one.

To use an existing J. D. Edwards OneWorld connection:

1. In the left pane of the BEA Application Explorer window, under Applications right-click J. D. Edwards OneWorld→Existing Connection→*your connection*.

The connection appears below the J. D. Edwards OneWorld node.



2. If the connection parameters do not correspond to your system, edit them in the J. D. Edwards OneWorld Logon Window.
3. Click OK.

Disconnecting from J. D. Edwards OneWorld

The BEA Application Explorer allows you to disconnect from J. D. Edwards OneWorld.

To disconnect from J. D. Edwards OneWorld:

- In the left pane of the BEA Application Explorer, right-click on the connection. Choose Disconnect.

This disconnects from J. D. Edwards OneWorld, and the connection icon change to indicate that it is not currently connected. To re-establish the connection, right-click on the connection and choose Connect.

Removing Connections

The BEA Application Explorer allows you to remove connections when you no longer need them.

To remove a connection:

- In the left pane of the BEA Application Explorer, right-click on the connection. Choose Remove.

Managing Schemas

You need to create a schema for each service and event your application uses. You use the BEA Application Explorer to create these schemas.

This section explains:

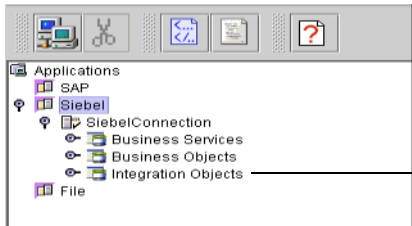
- [Creating Schemas for Services](#)
- [Creating Schemas for Events](#)
- [Removing Schemas](#)

Creating Schemas for Services

Services require two schemas, one for the request and one for the response. Services always have these two schema, even if the response is not used by your application.

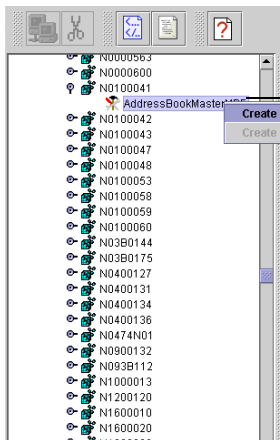
To create a schema for a service:

1. Start BEA Application Explorer. To learn more, see [Starting the BEA Application Explorer](#).
2. Set the session path. This determines where the BEA Application Explorer places your schemas. To learn more, see [Setting the Session Path](#).
3. Select or create a connection to J. D. Edwards OneWorld. To learn more, see [Managing J. D. Edwards OneWorld Connections](#).
4. Expand the tree under Applications → J. D. Edwards OneWorld → *connection name* → Integration Objects to see the items for which you may create a schema. If you cannot expand the tree beneath J. D. Edwards OneWorld, you have not set a connection for J. D. Edwards OneWorld.



Expand the list of integration objects.

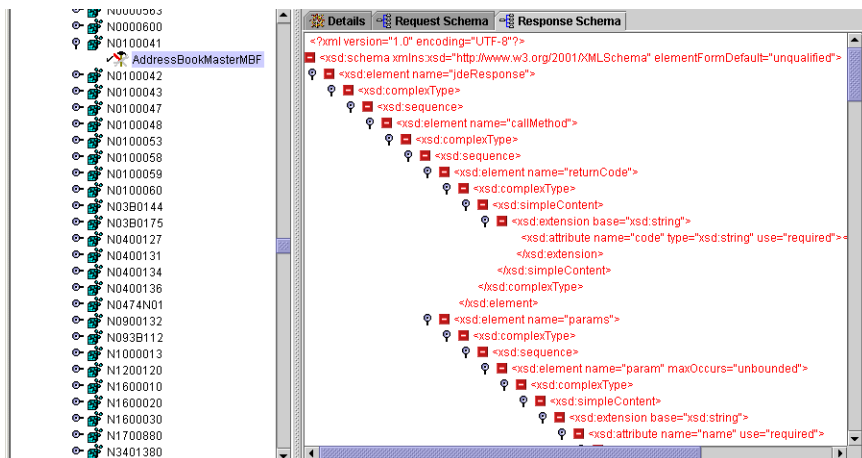
5. Select the integration object for this schema.



Select an integration object for this schema.

6. Right-click the item for which you wish to create the schema and choose Create Service Schemas.

The BEA Application Explorer displays tabs that show the request and response schemas.



The BEA Application Explorer creates a directory structure within the working directory you identified earlier. In this example, the working directory is `C:\BEA\BEASCHEMAS`.

Within this directory, the BEA Application Explorer creates a folder called J. D. Edwards OneWorld as well as subfolders to hold the schemas for each configured J. D. Edwards OneWorld connection. In this example, the schemas have been created in the folder called `jde1`, and the BEA Application Explorer adds the following items to the folder

`C:\BEA\BEASCHEMAS\JDEDWARDSOW\jde1:`

- `manifest.xml`
- `service_AcceptOrRejectOrder1-1-FA22.xsd`
- `service_AcceptOrRejectOrder1-1-FA22_response.xsd`

You have successfully created service request and response schemas for this integration object.

Creating Schemas for Events

The BEA WebLogic Adapter for J.D. Edwards OneWorld enables the processing of J. D. Edwards OneWorld business function events through the J. D. Edwards OneWorld ThinNet API.

The event request begins with the configuration of the OneWorld environment to produce the desired event information. Once configured, the OneWorld environment notifies the BEA-supplied J. D. Edwards OneWorld Event Listener of an event. The listener then passes the pertinent information to the event generator for retrieval of the event information using the J. D. Edwards OneWorld ThinNet API. The information is retrieved from J. D. Edwards Z files, which are used to hold outbound event data.

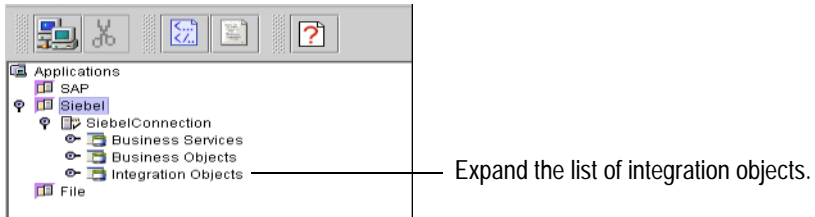
To learn more about configuring your environment, see [Appendix A, “Configuring J. D. Edwards OneWorld for Outbound Transaction Processing.”](#)

Events only require one schema. There is no response expected when J. D. Edwards OneWorld generates an event.

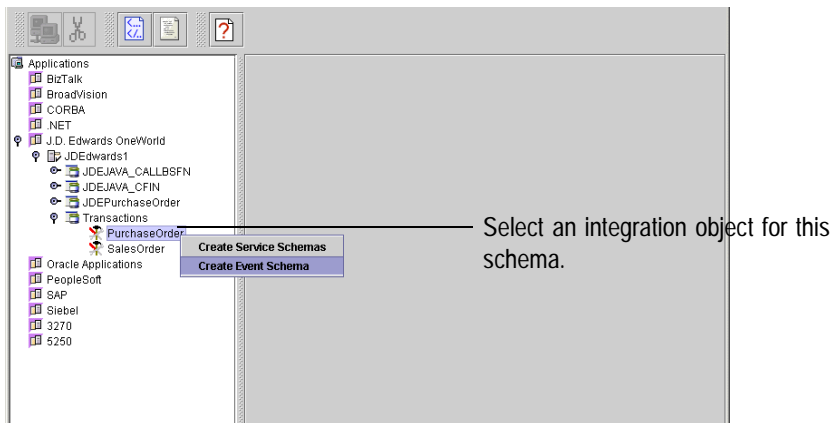
To create a schema for an event:

1. Start BEA Application Explorer. To learn more, see [Starting the BEA Application Explorer](#).
2. Set the session path. This determines where the BEA Application Explorer places your schemas. To learn more, see [Setting the Session Path](#).
3. Select or create a connection to your EIS. To learn more, see [Managing J. D. Edwards OneWorld Connections](#).

- Expand the tree under Applications → J. D. Edwards OneWorld → *connection name* → Integration Objects to see the items for which you may create a schema. If you cannot expand the tree beneath J. D. Edwards OneWorld, you have not set a connection for J. D. Edwards OneWorld.

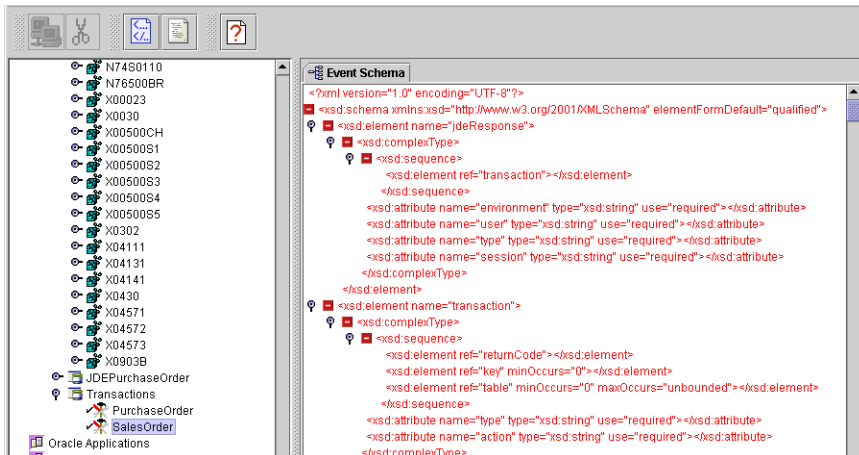


- Select the integration object for this schema.



- Right-click the item for which you wish to create the schema and choose Create Event Schemas.

The BEA Application Explorer displays a tab that shows the event schema. If you have service schemas for this item, the BEA Application Explorer also displays tabs with those request and response schemas.



Within this directory, the BEA Application Explorer creates a folder called J. D. Edwards OneWorld as well as subfolders to hold the schemas for each configured J. D. Edwards OneWorld connection. In this example, the schemas have been created in the folder called jde1, and the BEA Application Explorer adds the following items to the folder C:\BEA\BEASCHEMAS\JDEDWARDSOW\jde1:

- manifest.xml
- event_AcceptOrRejectOrder1-1-FA22.xsd

You have successfully created an event schema for this integration object.

Removing Schemas

To remove a schema:

1. Right-click on an integration object for which there is at least one schema.

If there is an event schema defined for this integration object, the menu has a Remove Event Schemas option.

If there are service schemas defined for this integration object, the menu has a Remove Event Schema option.

2. Choose the appropriate option.

Next Steps

After you have defined schemas for your events and services, the next step is to create an application view. An application view makes the services and events available to applications. To learn more about application views, see [Defining Application Views for J. D. Edwards OneWorld](#).

Defining Application Views for J. D. Edwards OneWorld

An application view is a business-oriented interface to objects and operations within an EIS.

This section presents the following topics:

- [How to Use This Document](#)
- [Before You Begin](#)
- [About Application Views](#)
- [About Defining Application Views](#)
- [Defining Service Connection Parameters](#)
- [Setting Service Properties](#)
- [Setting Event Properties](#)
- [Defining Event Connection Parameters](#)
- [Testing Services](#)
- [Testing Events Using a Service](#)
- [Testing Events Manually](#)

How to Use This Document

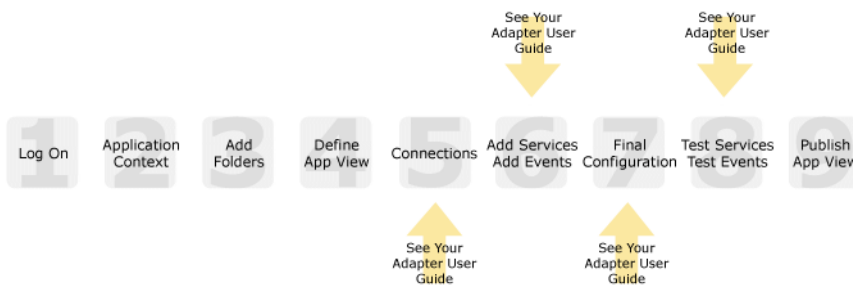
This document is designed to be used in conjunction with *Using the Application Integration Design Console*, available at the following URL:

<http://edocs.bea.com/wli/docs81/aiuser/index.html>

Using the Application Integration Design Console describes, in detail, the process of defining an application view, which is a key part of making an adapter available to process designers and other users. What *Using the Application Integration Design Console* does *not* cover is the specific information—about connections to your J. D. Edwards OneWorld system, as well as supported services and events—that you must supply as part of the application view definition. You will find that information in this section.

At each point in *Using the Application Integration Design Console* where you need to refer to this document, you will see a note that directs you to a section in your adapter user guide, with a link to the edocs page for adapters. The following road map illustration shows where you need to refer from *Using the Application Integration Design Console* to this document.

Figure 3-1 Information Interlock with *Using the Application Integration Design Console*



Before You Begin

Before you define an application view, make sure you have:

- Installed and deployed the adapter according to the instructions in *BEA WebLogic Adapter for J. D. Edwards OneWorld Installation and Configuration Guide*.
- Determined which business processes need to be supported by the application view. The required business processes determine the types of services and events you include in your application views. Therefore, you must gather information about the application's business requirements from the business analyst. Once you determine the necessary business

processes, you can define and test the appropriate services and events. For more information, see [Getting Started With the Adapter for J. D. Edwards OneWorld](#).

- Gathered the connection information for your J. D. Edwards OneWorld system. To learn more about the connection information needed by the BEA Application Explorer for your J. D. Edwards OneWorld system, see [Before You Begin](#).

About Application Views

An application view defines:

- Connection information for the EIS, including login information, connection settings, and so on.
- Service invocations, including the information the EIS requires for this request, as well as the request and response schemas associated with the service.
- Event notifications, including the information the EIS publishes and the event schema for inbound messages.

Typically, an application view is configured for a single business purpose and contains only the services and events required for that purpose. An EIS might have multiple application views, each defined for a different purpose.

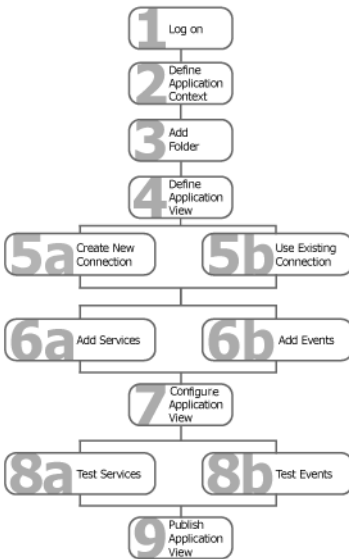
About Defining Application Views

Defining an application view is a multi-step process described in *Using the Application Integration Design Console*, available at the following URL:

<http://edocs.bea.com/wli/docs81/aiuser/index.html>

The information you enter depends on the requirements of your business process and your EIS system configuration. [Figure 3-2](#) summarizes the procedure for defining and configuring an application view.

Figure 3-2 Process for Defining and Configuring an Application View



To define an application view:

1. Log on to the WebLogic Integration Application View Console.
2. Define the application context by selecting an existing application or specifying a new application name and root directory.

This application will be using the events and services you define in your application view. The application view works within the context of this application.

3. Add folders as required to help you organize application views.
4. Define a new application view for your adapter.
5. Add a new connection service or select an existing one.

If you are adding a new connection service, see [Defining Service Connection Parameters](#) for details about J. D. Edwards OneWorld requirements.

6. Add the events and services for this application view.

See the following sections for details about J. D. Edwards OneWorld requirements:

- [Setting Service Properties](#)

- [Setting Event Properties](#)
7. Perform final configuration tasks.

If you are adding an event connection, see [Defining Event Connection Parameters](#) for details about J. D. Edwards OneWorld requirements.
 8. Test all services and events to make sure they can properly interact with the target J. D. Edwards OneWorld system.

See the following sections for details about testing the BEA WebLogic Adapter for J.D. Edwards OneWorld:

 - [Testing Services](#)
 - [Testing Events Using a Service](#)
 - [Testing Events Manually](#)
 9. Publish the application view to the target WebLogic Workshop application.

This is the application you specified in step 2. Publishing the application view allows workflow developers within the target application to interact with the newly published application view using an Application View control.

Defining Service Connection Parameters



This information applies to “Step 5A, Create a New Browsing Connection” in *Using the Application Integration Design Console*, at the following URL:

<http://edocs.bea.com/wli/docs81/aiuser/index.html>

The Select Browsing Connection page allows you to choose the type of connection factory to associate with the application view. You can select a connection factory within an existing instance of the adapter or create a connection factory within a new adapter instance.

Adapter Instance:

[Create New...](#) _____

Click to create a new connection factory

Existing Adapter Instances:

Adapter Name	Operations	Description

Existing connection factories will be here.

[Back](#)

After you enter a connection name and description, you use the Configure Connection Parameters page to specify connection parameters for a connection factory.

To create a new browsing connection:

1. In the Create New Browsing Connections page, enter a connection name and description as described in *Using the Application Integration Design Console*.

The Configure Connection Parameters page appears to allow you to configure the newly created connection factory within the new adapter instance.

On this page, you supply parameters to connect to your EIS

The BEA Application Explorer generates schema information for a session stored at a location that must be known to the general adapter. Enter this session location here. A session can support multiple connections.

Once you have entered the **session path** location, click on the pulldown arrow for the **connection name**, which will display a selection list of valid connections.

Session Path*	<input type="text" value="D:\Program Files\BEA Systems\BEA Application Explorer\sessions"/>	Specify a session path.
Connection Name*	<input type="text" value="IDES"/>	Specify a connection.
<input type="button" value="Connect to EIS"/>		

Note: A red asterisk (*) indicates that a field is required.

2. Specify a session path and connection name.

This information enables the application view to interact with the target J. D. Edwards OneWorld system. You need enter this information only once per application view.

3. Click Connect to EIS.

You return to the Create New Browsing Connections, where you can specify connection pool parameters and logging levels. For more information, see *Using the Application Integration Design Console* at the following URL:

<http://edocs.bea.com/wli/docs81/aiuser/index.html>

Setting Service Properties



This information applies to “Step 6A, Add a Service to an Application View” in *Using the Application Integration Design Console*, at the following URL:

<http://edocs.bea.com/wli/docs81/aiuser/index.html>

Adapter for J. D. Edwards OneWorld uses services to make requests of the J. D. Edwards OneWorld system. A service consists of both a request and a response. The Adapter for J. D. Edwards OneWorld supports the following services that use the following methods to access J. D. Edwards OneWorld:

- Flat Files
- Database
- Master Business Functions

To configure a service:

1. Enter a unique service name that describes the function the service performs.

The Add Services page displays the fields required for this service.

Unique Service Name: *

server*	<input type="text"/>
JD Edwards Port*	<input type="text" value="6009"/>
user*	<input type="text" value="JDEUser"/>
Password*	<input type="password" value="*****"/>
environment*	<input type="text" value="DV7333"/>

schema:

Note: A red asterisk (*) indicates that a field is required.

2. Enter the following information:

Table 3-1 File Service Parameters

Parameter	Description
server	The IP address of the machine running the J. D. Edwards OneWorld application
J. D. Edwards Port	The port used by J. D. Edwards OneWorld
User	The user ID required to access OneWorld
Password	The password associated with the specified user ID
Environment	The OneWorld-specified environment value

3. See [“Common Service and Event Settings” on page 3-8](#) for information about selecting a schema and configuring logging and tracing.

Common Service and Event Settings

1 2 3 4 5 **6** 7 8 9

This information applies to “Step 6A, Add a Service to an Application View” in *Using the Application Integration Design Console*, at the following URL:

<http://edocs.bea.com/wli/docs81/aiuser/index.html>

You select a schema and select logging options the same way for all services.

To set common service settings:

1. In the Schema list, select the schema you want to use with this service.

To learn more about schemas, see [Chapter 2, “Generating Schemas for J. D. Edwards OneWorld Integration Objects.”](#)

schema:

2. Configure logging and tracing for this service, as follows:

Logging captures information from your adapter and writes it in a log file. Tracing displays runtime information in the console. You set the type and amount of information you wish to capture as part of the final configuration tasks. This is described in detail in *Using the Application Integration Design Console*.

settings

Trace on/off	<input checked="" type="checkbox"/>
Verbose Trace on/off	<input checked="" type="checkbox"/>
Document Trace on/off	<input checked="" type="checkbox"/>

- a. Select the Trace on/off check box to enable tracing for this service. Trace information appears in the runtime console.
 - b. Select the Verbose Trace on/off check box to enable more detailed tracing for this service. Trace information appears in the runtime console.
 - c. Select the Document trace check box to enable recording of additional trace information for deeper troubleshooting.
3. Click Add to add the service.

For more information about the next step, see *Using the Application Integration Design Console* at the following URL:

<http://edocs.bea.com/wli/docs81/aiuser/index.html>

Setting Event Properties

1 2 3 4 5 **6** 7 8 9

This information applies to “Step 6B, Add an Event to an Application View” in *Using the Application Integration Design Console*, at the following URL:

<http://edocs.bea.com/wli/docs81/aiuser/index.html>

An event defines how your application responds to events generated by J. D. Edwards OneWorld.

The Adapter for J. D. Edwards OneWorld supports these events:

- [J. D. Edwards OneWorld System Event](#)
- [J. D. Edwards OneWorld File Event](#)
- [J. D. Edwards OneWorld FTP Event](#)

J. D. Edwards OneWorld System Event

To configure a J. D. Edwards OneWorld system event:

1. Select JDESYS as the event type.
2. Enter a unique event name that describes the function the event performs.

The Add Events page displays the fields required for this event.

On this page, you add events to your application view.

Unique Event Name: *

Select: JDESYS

server*	<input type="text" value="bvision01"/>
JD Edwards Port*	<input type="text" value="6009"/>
user*	<input type="text" value="JDE"/>
password*	<input type="password" value="....."/>
environment*	<input type="text" value="DV7333"/>
encoding*	<input type="text" value="ISO-8859-1"/>
Local port*	<input type="text" value="4551"/>

schema: SalesOrder

Note: A red asterisk (*) indicates that a field is required.

3. Enter the following information:

Table 3-2 System Event Parameters

Parameter	Description
server	The IP address of the machine running the OneWorld application
J. D. Edwards port	The port used by the OneWorld system
user	The user ID required to access the OneWorld system
password	The password associated with the specified user ID.
environment	The OneWorld-specified environment value.
encoding	The type of encoding to use in XML. Accept the default value
Local port	The the port used by the event generator for accepting OneWorld event notifications.

4. See [“Common Service and Event Settings” on page 3-8](#) for information about selecting a schema and configuring logging and tracing.

J. D. Edwards OneWorld File Event

You can use a file event to provide error handling in the case where the J. D. Edwards OneWorld is up and running, but WebLogic Integration is unavailable. To learn more about setting up your J. D. Edwards OneWorld system to write events to files when the WebLogic Integration system is unavailable, see “Installing and Configuring the J. D. Edwards OneWorld Event Listener” in the *BEA WebLogic Adapter for J. D. Edwards OneWorld Installation and Configuration Guide*. Configuring a file event will allow the adapter to retrieve the event once WebLogic Integration is available.

To configure a J. D. Edwards OneWorld file event:

1. Select `JDE_FILE` as the event type.
2. Enter a unique event name that describes the function the event performs.

The Add Events page displays the fields required for this event.

On this page, you add events to your application view.

Unique Event Name:*

Select: **JDE_FILE** ▾

Location*	<input type="text"/>
Sort*	<input type="checkbox"/>
Polling Interval	<input type="text"/>
server*	bvision01
JD Edwards Port*	6009
user*	JDE
password*	●●●●●●●●●●
environment*	DV7333
encoding*	UTF-8

Note: A red asterisk (*) indicates that a field is required.

3. Enter the following information:

Table 3-3 File Event Parameters

Parameter	Description
Location	The directory path where the event payload is to be saved in the event of an error. Do not include the file extension.
Sort	If true, documents are sorted by arrival time. This maintains the sequence of the events, but slows performance.
Polling Interval	The maximum interval between checks for new documents. The format of this value is xxH:xxM:xxS. For example 1H:2M:3S is one hour, two minutes and three seconds. The default value is 2 seconds. If you set the timeout to 0, the listener will check once for documents and then terminate. The higher you set this value, the fewer system resources will be used. However, a higher value means that the listener will take longer to respond to a stop command.
server	The IP address of the machine running the OneWorld application
J. D. Edwards port	The port used by the OneWorld system
user	The user ID required to access the OneWorld system
password	The password associated with the specified user ID.

Table 3-3 File Event Parameters (Continued)

Parameter	Description
environment	The OneWorld-specified environment value.
encoding	The type of encoding to use in XML. Accept the default value

4. See [“Common Service and Event Settings” on page 3-8](#) for information about selecting a schema and configuring logging and tracing.

J. D. Edwards OneWorld FTP Event

You can use an FTP event to provide error handling in the case where the J. D. Edwards OneWorld is up and running, but WebLogic Integration is unavailable. To learn more about setting up your J. D. Edwards OneWorld system to write events to files when the WebLogic Integration system is unavailable, see “Installing and Configuring the J. D. Edwards OneWorld Event Listener” in the *BEA WebLogic Adapter for J. D. Edwards OneWorld Installation and Configuration Guide*. Configuring an FTP event will allow the adapter to retrieve the event once WebLogic Integration is available.

To configure a J. D. Edwards OneWorld FTP event:

1. Select JDE_FTP as the event type.
2. Enter a unique event name that describes the function the event performs.

The Add Events page displays the fields required for this event.

On this page, you add events to your application view.

Unique Event Name*:

Select: JDE_FTP

Host name*	<input type="text"/>
FTP port	<input type="text"/>
User ID*	<input type="text"/>
Password*	<input type="text"/>
Location*	<input type="text"/>
Polling interval	<input type="text"/>
server*	bvision01
JD Edwards Port*	6009
user*	JDE
password*	●●●●●●●●●●
environment*	DV7333
encoding*	UTF-8

Note: A red asterisk (*) indicates that a field is required.

3. Enter the following information:

Table 3-4 FTP Event Parameters

Parameter	Description
Host Name	The name of the FTP host.
FTP Port	The FTP system port. Defaults to the FTP default port.
User ID	User name to use when connecting to the FTP host
Password	Password associated with the User ID
Location	The directory path on the FTP host from which to retrieve files. This is the directory in where the event payload is to be saved in the event of an error.
Polling Interval	<p>The maximum interval between checks for new documents. The format of this value is xxH:xxM:xxS. For example 1H:2M:3S is one hour, two minutes and three seconds. The default value is 2 seconds. If you set the timeout to 0, the listener will check once for documents and then terminate.</p> <p>The higher you set this value, the fewer system resources will be used. However, a higher value means that the listener will take longer to respond to a stop command.</p>

Table 3-4 FTP Event Parameters (Continued)

Parameter	Description
server	The IP address of the machine running the OneWorld application
J. D. Edwards port	The port used by the OneWorld system
user	The user ID required to access the OneWorld system
password	The password associated with the specified user ID.
environment	The OneWorld-specified environment value.
encoding	The type of encoding to use in XML. Accept the default value

Note: A red asterisk (*) indicates that a field is required.

4. Enter the following information:

Defining Event Connection Parameters



This information applies to “Step 7, Perform Final Configuration Tasks” in *Using the Application Integration Design Console*, at the following URL:

<http://edocs.bea.com/wli/docs81/aiuser/index.html>

Once you have finished adding services and events and have saved your application view, you must perform some final configuration tasks, including configuring event delivery connections, before testing the services and events. You perform these configuration tasks from the Final Configuration and Testing page.

To define event connection parameters:

1. In Connections area on the Application View Administration page, click Select/Edit.
2. In the Event Connection area, click Event to edit the default event connection.

The Configure Event Delivery Parameters page appears.

On this page, you supply parameters to configure event delivery for this ApplicationView

Password:

SleepCount:

UserName:

Enter connection information for your system.

3. Enter the following information:

Table 3-5 Event Connection Parameters

Parameter	Description
username	Your WebLogic Server Administration Console user name, defined in the startWebLogic script
password	The password for your WebLogic Server Administration Console user name
SleepCount	The number of seconds the adapter will wait between polling for events

The event delivery parameters you enter on this page enable connection to your J. D. Edwards OneWorld system and are used when generating events. The parameters are specific to the associated adapter and are defined in the `wli-ra.xml` file within the base adapter.

4. Click Save to save your event delivery parameter settings. Click Continue to return to the Edit Event Adapter page, and then click Back to return to the Final Configuration and Testing page.

The Edit Event Adapter page allows you to define event parameters and configure the information that will be logged for the connection factory. Select one of the following settings for the log:

- Log errors and audit messages
- Log warnings, errors, and audit messages
- Log informational, warning, error, and audit messages
- Log all messages

Note: For maximum tracing, select Log all Messages. This is the recommended setting to use when you are collecting debugging information for BEA support.

The table that follows describes the type of information that each logging message contains.

Table 3-6 Logging message categories

This type of message	Contains
Audit	Extremely important information related to the business processing performed by an adapter.
Error	Information about an error that has occurred in the adapter, which may affect system stability.
Warning	Information about a suspicious situation that has occurred. Although this is not an error, it could have an impact on adapter operation.
Information	Information about normal adapter operations.

Testing Services



This information applies to “Step 8A, Test an Application View’s Services” in *Using the Application Integration Design Console*, at the following URL:

<http://edocs.bea.com/wli/docs81/aiuser/index.html>

The purpose of testing an application view service is to evaluate whether that service interacts properly with the target J. D. Edwards OneWorld system. When you test a service, you supply any inputs required to start the service. For the Adapter for J. D. Edwards OneWorld, the input is in the form of a valid XML string (the `jdeRequest` XML files created based on the generated schema) that acts as input for the service.

Note: You can test an application view only if it is deployed and only if it contains at least one event or service.

To test a service:

1. In the Application View Administration page, click the Test link beside the service to be tested.

The Test Services page appears.

2. In the Test Service window, copy the appropriate XML strings from the JDE XML document.
3. Click Test.

The results appear in the Test Results window.

Testing Events Using a Service



This information applies to “Step 8B, Test an Application View’s Events” in *Using the Application Integration Design Console*, at the following URL:

<http://edocs.bea.com/wli/docs81/aiuser/index.html>

The purpose of testing an application view event is to make sure that the adapter correctly handles events generated by J. D. Edwards OneWorld. When you test an event, you can trigger the event using a service or manually.

Note: You can test an application view only if it is deployed and only if it contains at least one event or service.

To test an event:

1. In the Application View Administration page, click the Test link beside the event to be tested. The Test Events page appears.
2. Select Service and choose a service that triggers the event you are testing.
3. In the Time field, enter a reasonable period of time to wait, specified in milliseconds, before the test times out (One second = 1000 milliseconds. One minute = 60,000 milliseconds.).
4. Click Test and enter the XML string needed to trigger the service.

The service is executed.

- If the test succeeds, the Test Result page appears, showing the event document, the service input document, and the service output document.
- If the test fails, the Test Result page displays only a Timed Out message.

Testing Events Manually



This information applies to “Step 8B, Test an Application View’s Events” in *Using the Application Integration Design Console*, at the following URL:

<http://edocs.bea.com/wli/docs81/aiuser/index.html>

To test an event manually:

1. In the Time field, enter a reasonable period of time to wait, specified in milliseconds, before the test times out (One second = 1000 milliseconds. One minute = 60,000 milliseconds.).
2. Click Test. The test waits for an event to trigger it.
3. In your J. D. Edwards OneWorld system, execute a J. D. Edwards OneWorld Workflow that sends J. D. Edwards OneWorld XML to the adapter.
4. Using the J. D. Edwards OneWorld client, perform an action that triggers the event.
 - If the test succeeds, the Test Result page appears. This page displays the event document from the application, the service input document, and the service output document.
 - If the test fails or takes too long, the Test Result page appears, showing a Timed Out message.

Configuring J. D. Edwards OneWorld for Outbound Transaction Processing

J. D. Edwards OneWorld provides the ability to specify outbound functionality for business functions. This is accomplished by a processing option that controls how a transaction is written.

This section describes how to enable outbound transaction processing in OneWorld and modify the `jde.ini` file for XML support. It includes the following topics:

- [Enabling Outbound Transaction Processing](#)
- [Modifying the `jde.ini` File](#)

Enabling Outbound Transaction Processing

To enable outbound transaction processing:

1. Right-click the application that contains the processing options for the transaction's Master Business Functions. See Appendix B of the *J. D. Edwards Interoperability Guide for OneWorld Xe* for a list of these options.
2. Select Prompt for Values from the shortcut menu.
3. Click either the Outbound tab or the Interop tab.
4. Enter the transaction type. The OneWorld Event Listener processes only the *after* image for the business function, so setting the *before* image function is not required.
5. The processing of outbound data involves the use of the following:
 - The Data Export Control table

- The Processing Log table

The Data Export Control table manages the flow of the outbound data to third-party applications. OneWorld allows for the subscription of multiple vendor-specific objects for an interoperability transaction. The records in the Data Export Control table are used to determine the vendor-specific objects to call from the Outbound Subsystem batch process (R00460), or the Outbound Scheduler batch process (R00461).

To use the data export controls:

- On the Work With Data Export Controls panel, click Add.
- Provide values for the Transaction Type and Order Type fields.
- For each detail row, enter either a batch process name or version, or a function name and the library.
- To launch the vendor-specific object for an add or insert, enter 1. Do the same for the update, delete, and inquiry actions.
- Enter 1 in the Launch Immediately column.

The Processing Log table contains all the information about the OneWorld event, including the transaction type, order type, and sequence number from the Data Export Control table, and if the columns are filled in, the batch process and version number and the successfully processed columns.

For more information on configuring J. D. Edwards OneWorld for outbound processing, see the chapter titled “Detailed Tasks for OneWorld Operations” in the *J. D. Edwards Interoperability Guide for OneWorld Xe*.

Modifying the jde.ini File

Since the adapter uses XML for the transfer of information to and from J. D. Edwards OneWorld, you must configure the OneWorld environment to handle XML. This is easily accomplished through modification of the OneWorld `jde.ini` file.

The following is a sample of the modifications required to implement XML support.

1. Add the following blocks:

```
[JDENET_KERNEL_DEF6]
;krnlName=CALL OBJECT KERNEL
;dispatchDLLName=jdekrnl.dll
;dispatchDLLFunction=_JDEK_DispatchCallObjectMessage@28
```



```

;maxNumberOfProcesses=10
;numberOfAutoStartProcesses=0
krnlName=CALL OBJECT KERNEL
dispatchDLLName=XMLCallObj.dll
dispatchDLLFunction=_XMLCallObjectDispatch@28
maxNumberOfProcesses=10
numberOfAutoStartProcesses=0

[JDENET_KERNEL_DEF15]
krnlName=XML TRANSACTION KERNEL
dispatchDLLName=XMLTransactions.dll
dispatchDLLFunction=_XMLTransactionDispatch@28
maxNumberOfProcesses=1
numberOfAutoStartProcesses=1

```

2. Change the following block:

Note: Change maxKernelRanges to 15, as shown.

```

[JDENET]
serviceNameListen=6009
serviceNameConnect=6009
maxNetProcesses=5
maxNetConnections=400
maxKernelProcesses=50
maxKernelRanges=15
netTrace=1
ServiceControlRefresh=5
MonitorOption=0 0 0 0 0 0 0

```

For specific information on setting up your J. D. Edwards OneWorld environment for XML support, see the chapter titled “Setting the `jde.ini` File for XML” in the *J. D. Edwards Interoperability Guide for OneWorld Xe*.

Coexistence Between J. D. Edwards OneWorld and J. D. Edwards WorldSoftware

Coexistence between J. D. Edwards OneWorld and J. D. Edwards WorldSoftware provides the functionality for the two systems to share the same set of business data files at the same time without conflict. This is a preferred migration option to transition from WorldSoftware to OneWorld.

This section describes types of coexistence and the advantages and disadvantages of using coexistence. It includes the following topics:

- [Types of Coexistence](#)
- [Installation Guidelines](#)

Types of Coexistence

- **Functional coexistence:** WorldSoftware and OneWorld can access the same DB2/400 database.
- **Multi-platform coexistence:** Customers who want to unify a heterogeneous network of platforms and databases can use OneWorld to achieve an integrated solution. End users are shielded from the platform and database differences.
- **Vertical coexistence:** Customers use part of the WorldSoftware suite while using other vertical offerings, which are on OneWorld. For example, you might choose to migrate your G/L, A/P, and A/R to the OneWorld software while continuing to use the logistics and distribution verticals on the WorldSoftware side because of complex custom modifications you have written and have not yet migrated to OneWorld.

Advantages of Using Coexistence

- Provides ability for WorldSoftware to implement an E-commerce solution quickly.
- Allows gradual migration to J. D. Edwards OneWorld.
- Allows users to choose which system they are more comfortable with.
- Allows users to choose which system performs best to accomplish the same task.

Disadvantages of Using Coexistence

- Increased amount of DASD
- Dual maintenance needed
- Limited CNC configurations

Installation Guidelines

During the installation process, the client installs OneWorld Xe using a custom coexistence plan. OneWorld Xe creates data sources that point to native WorldSoftware files. OneWorld Xe runs against the same business data as WorldSoftware. The data resides on AS/400. The customer's users can run J. D. Edwards applications in OneWorld or WorldSoftware. Note that not all applications can coexist. A customer must choose to run those applications either in WorldSoftware or in OneWorld.

Below are tables that show which applications from OneWorld and WorldSoftware coexist. For complete information, contact J. D. Edwards support.

Table B-1 FOUNDATION

FOUNDATION	WorldSoftware	OneWorld	COEXIST
Generic Text	X	X	YES
Data Dictionary	X	X	YES
Supplemental Data	X	X	YES
Next Numbers			NO
Vertex			NO

Table B-2 FINANCIALS

FINANCIALS	WorldSoftware	OneWorld	COEXIST
Accounts Receivable	X	X	NO
Property Management	X	X	NO
Intercompany Settlements	X	X	YES
Multi-Site Consolidation	X	X	NO
Account Reconciliation			
Bank Statement Processing	X	X	NO
XRT-CERG		X	NO
Fixed Assets	X	X	YES
Interoperability		X	NO
EPS		X	NO
Euro Conversion	X	X	YES
Service Billing	X	X	NO
Contract Billing	X	X	NO
Contract Management	X		NO

Table B-3 HR/PAYROLL

HR/PAYROLL	WorldSoftware	OneWorld	COEXIST
Payroll	X	X	NO
HR	X	X	YES
Benefits Module	X	X	NO
Self Service Applications		X	NO

Table B-4 DISTRIBUTION

DISTRIBUTION	WorldSoftware	OneWorld	COEXIST
Generating a Proposal		X	NO
Customer Self Service		X	NO
Sales Order – Prepayment Processing		X	NO
Creating Intercompany Orders	X	X	YES
Advance Pricing into Procurement		B733.1	NO
Price Approvals		X	NO
Ship & Debit		X	NO
Outbound Transportation		X	NO
Inbound Transportation		X	NO

Table B-5 MANUFACTURING

MANUFACTURING	WorldSoftware	OneWorld	COEXIST
Quality Manufacturing	A73	B733.1	NO
Repetitive Manufacturing	A73	B733.1	NO
SynQuest		X	NO
Configurator		X	NO
Custom Works		X	NO
Active Supply Chain		X	NO
Forecast Pricing		X	NO
On-Line Simulation		X	NO

Table B-5 MANUFACTURING

MANUFACTURING	WorldSoftware	OneWorld	COEXIST
Over/Under Completions		X	NO
Copy Configured Items		X	NO
Store and Forward Configured Item		X	NO
Assembly Inclusion		X	NO
Equipment and Plant Maintenance – Approvals and Routing		X	NO
Interoperability		X	NO
CSMS		X	NO

Note: The “X” in the above tables represents the module available on that platform.

Sample Files

The BEA WebLogic Adapter for J.D. Edwards OneWorld supports the `jdeRequest` and `jdeResponse` XML structures for executing business functions within J. D. Edwards OneWorld. The use of J. D. Edwards OneWorld XML provides the following benefits:

- Aggregation of business function calls into a single object
- Availability of the J. D. Edwards OneWorld ThinNet API
- Access to both Z files and business functions

These benefits, combined with the power of WebLogic Server, provide the most scalable, efficient, and powerful integration solution with J. D. Edwards OneWorld.

This section describes the `jdeRequest` and `jdeResponse` XML structures for executing business functions within J. D. Edwards OneWorld. The code for these samples is not reproduced here. You can obtain these samples from the following URL:

http://commerce.bea.com/products/weblogicadapters/wladapters_samples.jsp

This section includes the following topics:

- [Single-Function Request](#)
- [Multiple-Function Request](#)

Single-Function Request

This sample, `GeteffectiveAddress.xml`, is a single-function call to J. D. Edwards OneWorld. In a single-function request, there is only one `callMethod` specified within the XML object. The

result of this request is a standard `jdeResponse` document. The sample ZIP file also contains a sample response document for this request, `GeteffectiveAddress_response.xml`.

Multiple-Function Request

The samples contain two requests that are multiple function requests:

- `SalesOrder.xml`
- `Purchase_Order.xml`

Both of these samples illustrate multiple-function call to J. D. Edwards OneWorld. The result of this request is a standard `jdeResponse` document with multiple sections. In a multiple-function request, there is more than one `callMethod` specified within the XML object. Notice that the XML contains return parameter specifications as well as file cleanup logic.

The sample ZIP file also contains sample response documents for these requests:

- `SalesOrder_response.xml`
- `Purchase_Order_response.xml`

The Purchase Order response document contains individual return codes for each `callMethod` executed. In addition, this method returns the order number assigned for the Purchase Order.

The Sales Order response document specifies error messages to returned in the document if an error occurs. These error messages can be used within a business process.

Index

A

- adapter
 - benefits 1-6
- Application Explorer
 - about the BEA Application Explorer 2-2
 - starting 2-4
- application views
 - adding events to 3-9
 - adding services to 3-6
 - events, adding 3-9
 - final configuration tasks 3-14
 - overview of defining 3-3
 - preparing to define 3-2
 - services, adding 3-6
 - services, testing 3-16
 - testing events manually 3-17
 - testing events using a service 3-17
 - testing services 3-16
- auditing events 3-15

B

- BEA WebLogic Adapter for J. D. Edwards OneWorld, overview of 1-1
- benefits of adapter 1-6

C

- configuring a J. D. Edwards file event 3-10
- configuring a J. D. Edwards FTP event 3-12
- configuring a J. D. Edwards system event 3-9
- customer support contact information x

D

- default session path, changing 2-4
- defining schemas 2-3

E

- events
 - about 2-3
 - adding to application views 3-9
 - auditing 3-15
 - testing 3-17
 - testing manually 3-17
 - testing using a service 3-17

G

- getting started 1-7

J

- J. D. Edwards OneWorld connections
 - creating a new connection 2-5
 - editing an existing connection 2-7
- J. D. Edwards OneWorld file event
 - configuring 3-10
- J. D. Edwards OneWorld FTP event
 - configuring 3-12
- J. D. Edwards OneWorld system event
 - configuring 3-9

L

- logging 3-15

P

- product support x

R

- related information viii

S

schemas

- defining 2-3

- events 2-3

- service requests 2-3

- service responses 2-3

service requests 2-3

service responses 2-3

services

- adding to application views 3-6

- testing 3-16

session path, changing 2-4

support x

supported J. D. Edwards OneWorld operations

1-2

T

technical support x