

# Oracle9iAS InterConnect Adapter for JDEdwards

Installation and User's Guide

Release 2 (9.0.2)

February 2002

Part No. A95451-01

Part No. A95451-01

Copyright © 2002 Oracle Corporation. All rights reserved.

The Programs (which include both the software and documentation) contain proprietary information of Oracle Corporation; they are provided under a license agreement containing restrictions on use and disclosure and are also protected by copyright, patent and other intellectual and industrial property laws. Reverse engineering, disassembly or decompilation of the Programs, except to the extent required to obtain interoperability with other independently created software or as specified by law, is prohibited.

The information contained in this document is subject to change without notice. If you find any problems in the documentation, please report them to us in writing. Oracle Corporation does not warrant that this document is error-free. Except as may be expressly permitted in your license agreement for these Programs, no part of these Programs may be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without the express written permission of Oracle Corporation.

If the Programs are delivered to the U.S. Government or anyone licensing or using the programs on behalf of the U.S. Government, the following notice is applicable:

**Restricted Rights Notice** Programs delivered subject to the DOD FAR Supplement are "commercial computer software" and use, duplication, and disclosure of the Programs, including documentation, shall be subject to the licensing restrictions set forth in the applicable Oracle license agreement. Otherwise, Programs delivered subject to the Federal Acquisition Regulations are "restricted computer software" and use, duplication, and disclosure of the Programs shall be subject to the restrictions in FAR 52.227-19, Commercial Computer Software - Restricted Rights (June, 1987). Oracle Corporation, 500 Oracle Parkway, Redwood City, CA 94065.

The Programs are not intended for use in any nuclear, aviation, mass transit, medical, or other inherently dangerous applications. It shall be the licensee's responsibility to take all appropriate fail-safe, backup, redundancy, and other measures to ensure the safe use of such applications if the Programs are used for such purposes, and Oracle Corporation disclaims liability for any damages caused by such use of the Programs.

Oracle is a registered trademark, and Oracle*MetaLink*, Oracle Store, Oracle9i, Oracle9iAS Discoverer, SQL\*Plus, and PL/SQL are trademarks or registered trademarks of Oracle Corporation. Other names may be trademarks of their respective owners.

---

---

# Contents

<b>Send Us Your Comments .....</b>	<b>vii</b>
<b>Preface.....</b>	<b>ix</b>
Intended Audience .....	x
Documentation Accessibility .....	x
Organization.....	x
Related Documentation .....	xi
Conventions.....	xii
<b>1 Introduction</b>	
<b>J.D. Edwards Communication Architecture .....</b>	<b>1-2</b>
<b>Software Requirements .....</b>	<b>1-2</b>
<b>Platforms.....</b>	<b>1-3</b>
<b>Required Files.....</b>	<b>1-3</b>
<b>Customizing the jdeinterop.ini File.....</b>	<b>1-3</b>
<b>2 Installation and Configuration</b>	
<b>Installing the J. D. Edwards OneWorld Adapter.....</b>	<b>2-2</b>
Preinstallation Tasks .....	2-2
Installation Tasks.....	2-2
<b>J. D. Edwards OneWorld Adapter Configuration.....</b>	<b>2-4</b>
Using the Application Parameter.....	2-5
adapter.ini Initialization Parameter File .....	2-6
Hub.ini .....	2-6

	Agent Connection Parameters .....	2-7
	J. D. Edwards OneWorld Adapter Parameters .....	2-12
	<b>Starting the J. D. Edwards OneWorld Adapter</b> .....	2-12
<b>3</b>	<b>Supported J. D. Edwards Interfaces</b>	
	<b>Using the J. D. Edwards OneWorld Adapter</b> .....	3-2
	<b>Creating an Implemented Procedure</b> .....	3-2
	Importing Attributes from J.D. Edwards .....	3-6
	<b>Creating a Subscribed Event</b> .....	3-8
<b>4</b>	<b>Using the Configuration Editor</b>	
	<b>Configuration Editor</b> .....	4-2
	<b>Log into J.D. Edwards</b> .....	4-4
	Default Credentials.....	4-4
<b>5</b>	<b>Using the J. D. Edwards Adapter</b>	
	<b>Control Flow Overview</b> .....	5-2
	Design Time.....	5-2
	Runtime.....	5-2
	Apartment Threading of J.D. Edwards OneWorld Business Functions.....	5-3
	Connection Pooling.....	5-3
	<b>Business Functions</b> .....	5-3
	Handling String Values.....	5-4
	Using the MATH_NUMERIC Type.....	5-6
	Exponents .....	5-7
	Invalid Values .....	5-8
	Precision for Operations .....	5-8
	Currency .....	5-8
	Typedef of MATH_NUMERIC.....	5-8
	Typedef of JDEDATE.....	5-9
<b>6</b>	<b>Custom Package Creation and Installation</b>	
	<b>Defining a Custom Package</b> .....	6-2
	<b>Custom Package Creation and Installation Prerequisites</b> .....	6-2

Related Documentation .....	6-2
Retrieving the Package Management Reference Document .....	6-3
Retrieving the Product Packing Tips and Techniques Reference Document .....	6-3
<b>Creating a Custom Package</b> .....	6-4
Basic Steps for Packing the Source Files and Defining the Custom Package .....	6-4
Basic Steps for Making the Custom Package Portable .....	6-6
<b>Installing the Custom Package ACBREL1</b> .....	6-7
<b>List of Modules</b> .....	6-9

## 7 Limitations

<b>Querying and Retrieving Lists of Records</b> .....	7-2
Controlling Iteration.....	7-2

## Index



---

---

# Send Us Your Comments

**Oracle9iAS InterConnect Adapter for JDEdwards Installation and User's Guide, Release 2 (9.0.2)**

**Part No. A95451-01**

Oracle Corporation welcomes your comments and suggestions on the quality and usefulness of this document. Your input is an important part of the information used for revision.

- Did you find any errors?
- Is the information clearly presented?
- Do you need more information? If so, where?
- Are the examples correct? Do you need more examples?
- What features did you like most?

If you find any errors or have any other suggestions for improvement, please indicate the document title and part number, and the chapter, section, and page number (if available). You can send comments to us in the following ways:

- Electronic mail: [iasdocs\\_us@oracle.com](mailto:iasdocs_us@oracle.com)
- FAX: 650-506-7407 Attn: Oracle9i Application Server Documentation Manager
- Postal service:  
Oracle Corporation  
Oracle9i Application Server Documentation  
500 Oracle Parkway, M/S 2op3  
Redwood Shores, CA 94065  
USA

If you would like a reply, please give your name, address, telephone number, and (optionally) electronic mail address.

If you have problems with the software, please contact your local Oracle Support Services.





---

# Preface

This preface contains these topics:

- [Intended Audience](#)
- [Documentation Accessibility](#)
- [Organization](#)
- [Related Documentation](#)
- [Conventions](#)

## Intended Audience

This guide is intended for those who perform the following tasks:

- install applications
- maintain applications

## Documentation Accessibility

Our goal is to make Oracle products, services, and supporting documentation accessible, with good usability, to the disabled community. To that end, our documentation includes features that make information available to users of assistive technology. This documentation is available in HTML format, and contains markup to facilitate access by the disabled community. Standards will continue to evolve over time, and Oracle Corporation is actively engaged with other market-leading technology vendors to address technical obstacles so that our documentation can be accessible to all of our customers. For additional information, visit the Oracle Accessibility Program Web site at

<http://www.oracle.com/accessibility/>

**Accessibility of Code Examples in Documentation** JAWS, a Windows screen reader, may not always correctly read the code examples in this document. The conventions for writing code require that closing braces should appear on an otherwise empty line; however, JAWS may not always read a line of text that consists solely of a bracket or brace.

**Accessibility of Links to External Web Sites in Documentation** This documentation may contain links to Web sites of other companies or organizations that Oracle Corporation does not own or control. Oracle Corporation neither evaluates nor makes any representations regarding the accessibility of these Web sites.

## Organization

This document contains:

### **Chapter 1, "Introduction"**

This chapter describes the J.D. Edwards OneWorld adapter and the hardware and software requirements.

## **Chapter 2, "Installation and Configuration"**

This chapter describes the installation and configuration procedures for the J.D. Edwards OneWorld adapter.

## **Chapter 3, "Supported J. D. Edwards Interfaces"**

This chapter describes the supported J.D. Edwards OneWorld adapter interfaces.

## **Chapter 4, "Using the Configuration Editor"**

This chapter provides information on using the Configuration Editor to configure the J.D. Edwards OneWorld adapter.

## **Chapter 5, "Using the J. D. Edwards Adapter"**

This chapter provides information on using the J.D. Edwards OneWorld adapter.

## **Chapter 6, "Custom Package Creation and Installation"**

This chapter provides custom package creation and installation procedures for the J.D. Edwards OneWorld adapter.

## **Chapter 7, "Limitations"**

This chapter provides the limitations of the J.D. Edwards OneWorld adapter.

## **Related Documentation**

For more information, see these Oracle resources:

- *Oracle9iAS InterConnect User's Guide* in the Oracle9i Application Server (Oracle9iAS) Documentation Library
- *Oracle9i Application Server Installation Guide*
- *Oracle9iAS InterConnect Adapter Configuration Editor User's Guide*

In North America, printed documentation is available for sale in the Oracle Store at

<http://oraclestore.oracle.com/>

Customers in Europe, the Middle East, and Africa (EMEA) can purchase documentation from

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

Other customers can contact their Oracle representative to purchase printed documentation.

To download free release notes, installation documentation, white papers, or other collateral, please visit the Oracle Technology Network (OTN). You must register online before using OTN; registration is free and can be done at

<http://otn.oracle.com/admin/account/membership.html>

If you already have a username and password for OTN, then you can go directly to the documentation section of the OTN Web site at

<http://otn.oracle.com/docs/index.htm>

## Conventions

This section describes the conventions used in the text and code examples of this documentation set. It describes:

- [Conventions in Text](#)
- [Conventions in Code Examples](#)
- [Conventions for Microsoft Windows Operating Systems](#)

## Conventions in Text

We use various conventions in text to help you more quickly identify special terms. The following table describes those conventions and provides examples of their use.

Convention	Meaning	Example
<b>Bold</b>	Bold typeface indicates terms that are defined in the text or terms that appear in a glossary, or both.	When you specify this clause, you create an <b>index-organized table</b> .
<i>Italics</i>	Italic typeface indicates book titles or emphasis.	<i>Oracle9i Database Concepts</i> Ensure that the recovery catalog and target database do <i>not</i> reside on the same disk.
UPPERCASE monospace (fixed-width) font	Uppercase monospace typeface indicates elements supplied by the system. Such elements include parameters, privileges, datatypes, RMAN keywords, SQL keywords, SQL*Plus or utility commands, packages and methods, as well as system-supplied column names, database objects and structures, usernames, and roles.	You can specify this clause only for a NUMBER column. You can back up the database by using the BACKUP command. Query the TABLE_NAME column in the USER_TABLES data dictionary view. Use the DBMS_STATS.GENERATE_STATS procedure.
lowercase monospace (fixed-width) font	Lowercase monospace typeface indicates executables, filenames, directory names, and sample user-supplied elements. Such elements include computer and database names, net service names, and connect identifiers, as well as user-supplied database objects and structures, column names, packages and classes, usernames and roles, program units, and parameter values. <b>Note:</b> Some programmatic elements use a mixture of UPPERCASE and lowercase. Enter these elements as shown.	Enter sqlplus to open SQL*Plus. The password is specified in the orapwd file. Back up the datafiles and control files in the /disk1/oracle/dbs directory. The department_id, department_name, and location_id columns are in the hr.departments table. Set the QUERY_REWRITE_ENABLED initialization parameter to true. Connect as oe user. The JRepUtil class implements these methods.
lowercase italic monospace (fixed-width) font	Lowercase italic monospace font represents placeholders or variables.	You can specify the <i>parallel_clause</i> . Run <i>Uold_release</i> .SQL where <i>old_release</i> refers to the release you installed prior to upgrading.

## Conventions in Code Examples

Code examples illustrate SQL, PL/SQL, SQL\*Plus, or other command-line statements. They are displayed in a monospace (fixed-width) font and separated from normal text as shown in this example:

```
SELECT username FROM dba_users WHERE username = 'MIGRATE';
```

The following table describes typographic conventions used in code examples and provides examples of their use.

Convention	Meaning	Example
[ ]	Brackets enclose one or more optional items. Do not enter the brackets.	DECIMAL ( <i>digits</i> [ , <i>precision</i> ])
{ }	Braces enclose two or more items, one of which is required. Do not enter the braces.	{ENABLE   DISABLE}
	A vertical bar represents a choice of two or more options within brackets or braces. Enter one of the options. Do not enter the vertical bar.	{ENABLE   DISABLE} [COMPRESS   NOCOMPRESS]
...	Horizontal ellipsis points indicate either: <ul style="list-style-type: none"><li>■ That we have omitted parts of the code that are not directly related to the example</li><li>■ That you can repeat a portion of the code</li></ul>	CREATE TABLE ... AS <i>subquery</i> ;  SELECT <i>col1</i> , <i>col2</i> , ... , <i>coln</i> FROM employees;
.	Vertical ellipsis points indicate that we have omitted several lines of code not directly related to the example.	
Other notation	You must enter symbols other than brackets, braces, vertical bars, and ellipsis points as shown.	acctbal NUMBER(11,2); acct CONSTANT NUMBER(4) := 3;
<i>Italics</i>	Italicized text indicates placeholders or variables for which you must supply particular values.	CONNECT SYSTEM/ <i>system_password</i> DB_NAME = <i>database_name</i>

Convention	Meaning	Example
UPPERCASE	Uppercase typeface indicates elements supplied by the system. We show these terms in uppercase in order to distinguish them from terms you define. Unless terms appear in brackets, enter them in the order and with the spelling shown. However, because these terms are not case sensitive, you can enter them in lowercase.	<pre>SELECT last_name, employee_id FROM employees; SELECT * FROM USER_TABLES; DROP TABLE hr.employees;</pre>
lowercase	<p>Lowercase typeface indicates programmatic elements that you supply. For example, lowercase indicates names of tables, columns, or files.</p> <p><b>Note:</b> Some programmatic elements use a mixture of UPPERCASE and lowercase. Enter these elements as shown.</p>	<pre>SELECT last_name, employee_id FROM employees; sqlplus hr/hr CREATE USER mjones IDENTIFIED BY ty3MU9;</pre>

## Conventions for Microsoft Windows Operating Systems

The following table describes conventions for Microsoft Windows operating systems and provides examples of their use.

Convention	Meaning	Example
Choose Start >	How to start a program.	To start the Oracle Database Configuration Assistant, choose Start > Programs > Oracle - <i>HOME_NAME</i> > Configuration and Migration Tools > Database Configuration Assistant.
File and directory names	File and directory names are not case sensitive. The following special characters are not allowed: left angle bracket (<), right angle bracket (>), colon (:), double quotation marks ("), slash (/), pipe ( ), and dash (-). The special character backslash (\) is treated as an element separator, even when it appears in quotes. If the file name begins with \\, then Windows assumes it uses the Universal Naming Convention.	<pre>c:\winnt "\"system32 is the same as C:\WINNT\SYSTEM32</pre>

Convention	Meaning	Example
C:\>	<p>Represents the Windows command prompt of the current hard disk drive. The escape character in a command prompt is the caret (^). Your prompt reflects the subdirectory in which you are working. Referred to as the <i>command prompt</i> in this manual.</p>	C:\oracle\oradata>
	<p>The backslash (\) special character is sometimes required as an escape character for the double quotation mark (") special character at the Windows command prompt. Parentheses and the single quotation mark (') do not require an escape character. Refer to your Windows operating system documentation for more information on escape and special characters.</p>	<pre>C:\&gt;exp scott/tiger TABLES=emp QUERY=\ "WHERE job='SALESMAN' and sal&lt;1600\" C:\&gt;imp SYSTEM/password FROMUSER=scott TABLES=(emp, dept)</pre>
<i>HOME_NAME</i>	<p>Represents the Oracle home name. The home name can be up to 16 alphanumeric characters. The only special character allowed in the home name is the underscore.</p>	C:\> net start OracleHOME_ NAME_TNSListener



Convention	Meaning	Example
<i>ORACLE_HOME</i> and <i>ORACLE_BASE</i>	<p>In releases prior to Oracle8i release 8.1.3, when you installed Oracle components, all subdirectories were located under a top level <i>ORACLE_HOME</i> directory that by default used one of the following names:</p> <ul style="list-style-type: none"> <li>■ C:\orant for Windows NT</li> <li>■ C:\orawin95 for Windows 95</li> <li>■ C:\orawin98 for Windows 98</li> </ul> <p>This release complies with Optimal Flexible Architecture (OFA) guidelines. All subdirectories are not under a top level <i>ORACLE_HOME</i> directory. There is a top level directory called <i>ORACLE_BASE</i> that by default is C:\oracle. If you install Oracle9i release 1 (9.0.1) on a computer with no other Oracle software installed, then the default setting for the first Oracle home directory is C:\oracle\ora90. The Oracle home directory is located directly under <i>ORACLE_BASE</i>.</p> <p>All directory path examples in this guide follow OFA conventions.</p>	Go to the <i>ORACLE_BASE\ORACLE_HOME\rdms\admin</i> directory.



---

# Introduction

The J.D. Edwards OneWorld adapter enables you to use J.D. Edwards business functions with your system. This book discusses procedures for setting up the J.D. Edwards OneWorld adapter and some J.D. Edwards specific information.

This chapter discusses the following:

- [J.D. Edwards Communication Architecture](#)
- [Software Requirements](#)
- [Platforms](#)
- [Required Files](#)
- [Customizing the jdeinterop.ini File](#)

## J.D. Edwards Communication Architecture

The J.D. Edwards OneWorld adapter provides access to J.D. Edwards (JDE) business functions. J.D. Edwards communicates between client and server machines using a proprietary messaging architecture called JDENet. This is implemented using TCP/IP as a transport, with a default port of 6009.

Calls to J.D. Edwards business functions require two messages. For an Oracle9iAS InterConnect user, these two messages are wrapped in a single call to the business function. The first message responds with the location of the server that processes the business function. This is accomplished by performing a lookup in a set of tables called Object Configuration Mapping.

The second message sends a formatted message buffer containing the arguments to pass to or from J.D. Edwards to the appropriate server, and waits for a reply. The buffers are formatted according to the type definitions of the underlying C++ structures.

## Software Requirements

[Table 1-1](#) lists the systems to which the adapter connects:

**Table 1-1 Supported Systems**

Component Support	Required Components
J. D. Edwards OneWorld Adapter	XE (B73.3.3)
J.D. Edwards Enterprise server	NT, UNIX
JDK	1.1.8 and above

The J.D. Edwards ACBREL1 custom package is required to use the J.D. Edwards OneWorld adapter.

**See Also:**

- [Chapter 6, "Custom Package Creation and Installation"](#)
- *Oracle9i Application Server Installation Guide*, Appendix C for hardware requirements

## Platforms

The J.D. Edwards OneWorld adapter supports the following platforms:

- Windows NT 4.0 with service pack 6
- Windows 2000
- Solaris 2.6, Solaris 7 (2.7), and Solaris 8 (2.8) – compatible with the C++ workshop compiler version 6.0 on Solaris
- HP-UX B.11.00

## Required Files

J.D. Edwards OneWorld adapter uses the following .jar files at both runtime and design time:

- Connector.jar
- Kernel.jar
- JDEJAccess.jar
- JDEActionalInterop.jar

Kernel.jar and Connector.jar must be copied from the machine where J.D. Edwards OneWorld is installed to the machine where J.D. Edwards OneWorld adapter is installed.

The files must be copied to the following directories:

```
...\oai\9.0.2\lib\Kernel.jar  
...\oai\9.0.2\lib\Connector.jar
```

## Customizing the jdeinterop.ini File

The Connect.jar and Kernel.jar files use the jdeinterop.ini file to configure connections parameters to the J.D. Edwards OneWorld machine. You must edit ... \oai\9.0.2\config\JDE\jdeinterop.ini to include some settings the server may need. The default drive for the file is c:, however, you can change this location. Information is organized by section, for example, [JDENET]. Sections are listed in the order they are found in the software.

---

---

**Note:** Make sure you include the following path to the jdeinterop.ini file in the J.D. Edwards Configuration Editor:

...\oai\9.0.2\config\JDE\jdeinterop.ini.

---

---

**See Also:** ["Configuration Editor"](#) on page 4-2

#### [JDENET]

##### Setting Typical Value Purpose

EnterpriseServerTimeout = 90000 – The timeout value for a request to the enterprise server.

maxPoolSize = 30 – The JDENET socket connection pool size.

#### [SERVER]

##### Setting Typical Value Purpose

glossaryTextServer = JDED:6010 – The enterprise server and port that provide glossary text information. This is the server that returns text descriptions for errors. This is often the same host and port as the JDEdwards OneWorld application server. There may be more than one glossary server for different supported language encodings.

codePage = 1252 – The encoding scheme.

- 1252 English and Western European
- 932 Japanese
- 950 Traditional Chinese
- 936 Simplified Chinese
- 949 Korean

**[LOGS]****Setting Typical Value Purpose**

log = c:\jas.log - This is the file used by interoperability components to log significant information.

debuglog = c:\jasdebug.log—Location of debug log file.

Debug = FALSE

**[DEBUG]****Setting Typical Value Purpose**

JobFile = c:\Interop.log—Location of error file.

DebugFile = c:\InteropDebug.log—Location of debug file.

log = c:\net.log—Location of log file.

debugLevel = 0 - 12—Debug levels.

The following table describes debug levels:

Debug Level	Description
0	None. Logging is turned off and only errors are written to the JobFile.
2	Errors (error messages).
4	System Errors (exception messages).
6	Warning Information.
8	Minimum Trace (Key operations. For example, Login, Logoff, Business Function calls).
10	Trouble Shooting Information (Help).
12	Complete Debug Information (Logs everything).

The odd values are reserved for future levels to be added. By default, you typically do not need to trace. However, tracing is useful for debugging:

NetTraceLevel= 0

The following table describes trace levels:

Trace Level	Description
0	No trace.

Trace Level	Description
1	Refers to the Record process ID, thread ID, and the available socket status when a new connection is added and the socket pool is searched.
2	Includes all the information in Trace Level 1, and also traces every call made in the connection manager class.
3	Includes all information in Trace Level 2, and also traces <code>getPort()</code> calls and <code>getHost()</code> calls.

By default, you do not need to trace. However, tracing is useful for debugging.

### [INTEROP]

#### Setting Typical Value Purpose

`enterpriseServer = JDED` - This value is the name of the host server. Make sure this value is the same value you enter in the Host Name field in the JDE Credentials and Default Credentials panels in the Configuration Editor.

`port = 6010` - This value is the port number that is used to exchange data. Make sure this value is the same value you enter in the Port Number field in the JDE Credentials and Default Credentials panels in the Configuration Editor.

`inactive_timeout = 1200000` - The timeout value for a transaction in auto commit mode. If the user is inactive for this amount of time (in milliseconds), the interop server logs off the user. You can change this value to a shorter period of time.

`manual_timeout = 300000` - The timeout value for a transaction in manual commit mode.

`Repository c:\Oracle\Ora90\oai\9.0.2\lib` - Points to the location of the repository directory. This must point to the directory containing the `JDEActionalInterop.jar` file. This repository directory is `%OAI_INSTALL_DIR%\lib` where `%OAI_INSTALL-DIR%` is the Oracle9iAS InterConnect installer directory, `...\oai\9.0.2`.

### [CORBA]

#### Setting Typical Value Purpose

`Multithread = 1` - Set to 1 for multithread support for CORBA.

`Objects = CORBA::Connector;CORBA::OneWorldVersion` - The objects for the CORBA server to create at startup. Also replaces the `-DIORFILENAME =` command line option, for example: `CORBA::Connector=connector.ior`.



---

# Installation and Configuration

This chapter describes installation and configuration of the J.D. Edwards OneWorld adapter. This chapter discusses the following topics:

- [Installing the J. D. Edwards OneWorld Adapter](#)
- [J. D. Edwards OneWorld Adapter Configuration](#)
- [Starting the J. D. Edwards OneWorld Adapter](#)

## Installing the J. D. Edwards OneWorld Adapter

This section contains these topics:

- [Preinstallation Tasks](#)
- [Installation Tasks](#)

### Preinstallation Tasks

The J. D. Edwards OneWorld adapter must be installed in one of the following Oracle homes:

- An existing Oracle9i Application Server Oracle home
- An existing Oracle9i Application Server Infrastructure Database Oracle home
- An existing Oracle9iAS InterConnect Oracle home
- A new Oracle home (the installer creates this for you)

Consult the *Oracle9i Application Server Installation Guide* before proceeding with the J.D. Edwards OneWorld adapter installation. This guide includes information on:

- CD-ROM mounting
- Oracle Universal Installer startup
- Oracle9iAS InterConnect software, hardware, and system requirements
- Oracle9iAS InterConnect installation

---

---

**Note:** Oracle9iAS InterConnect Hub is installable through the Oracle9iAS InterConnect Hub installation type. You must install the Oracle9iAS InterConnect Hub before proceeding with the J.D. Edwards OneWorld adapter installation.

---

---

### Installation Tasks

To install the J.D. Edwards OneWorld adapter:

1. Click **Next** on the Welcome page.  
The File Locations page displays.
2. Enter the following information in the Destination fields:
  - Name—The Oracle home name.

- Path—The full path to the Oracle home in which to install the J.D. Edwards OneWorld adapter.

---

---

**Note:** Do not change the path specified in the Source field. This is the location on the CD-ROM from which to install the J.D. Edwards OneWorld adapter.

---

---

3. Click **Next**.

The Installation Types page displays.

4. Select Oracle9iAS InterConnect Adapters and click **Next**.

The Available Product Components page displays.

5. Select Oracle9iAS InterConnect J. D. Edwards OneWorld Adapter and click **Next**.

6. If the J.D. Edwards OneWorld adapter is not being installed on the same computer as Oracle9iAS InterConnect Hub and another adapter is not installed in the current Oracle home, the Oracle9iAS InterConnect Hub Database screen appears. Enter the following information about the Oracle9iAS InterConnect Hub to use:

- Host Name—The hostname of the computer on which Oracle9iAS InterConnect Hub database is installed.
- Port Number—The port number of the computer.
- Database SID—The system identifier (SID) of the Oracle9iAS InterConnect Oracle9iAS Metadata Repository.
- Password—The password for the Oracle9iAS Metadata Repository schema.

The Oracle9iAS Metadata Repository stores metadata used by Oracle9iAS InterConnect to coordinate communication between components.

7. Click **Next**.

The Oracle9iAS InterConnect Adapter Configuration page displays.

8. Enter the application associated with the J.D. Edwards OneWorld adapter. White spaces or blank spaces are not allowed. The default value is `myJDEApp`.

9. Click **Next**. Complete the fields for any other components selected for installation, such as other adapters. When finished, the Summary page displays.

10. Click **Install** to install the J.D. Edwards OneWorld adapter and other selected components. The J.D. Edwards OneWorld adapter is installed in the following directory:

Platform	Directory
Windows	%ORACLE_HOME%\oai\9.0.2\adapters\Application
UNIX	\$ORACLE_HOME/oai/9.0.2/adapters/Application

## J. D. Edwards OneWorld Adapter Configuration

Table 2-2, Table 2-3, and Table 2-4 describe executable files, configuration files, and directories. These files and directories are accessible from the directory shown in Table 2-1:

**Table 2-1 J.D. Edwards OneWorld adapter Directory**

Platform	Directory
UNIX	\$ORACLE_HOME/oai/9.0.2/adapters/Application
Windows	%ORACLE_HOME%\oai\9.0.2\adapters\Application

**Table 2-2 Executable Files**

File	Description
start.bat (Windows)	Takes no parameters, starts the adapter.
start (UNIX)	
stop.bat (Windows)	Takes no parameters; stops the adapter.
stop (UNIX)	
ignoreErrors.bat (Windows)	If an argument is specified, then the given error code will be ignored. If no argument is specified, then all error codes specified in the ErrorCodes.ini will be ignored.
ignoreErrors (UNIX)	

**Table 2–3 Configuration Files**

File	Description
ErrorCodes.ini (Windows and UNIX)	Should contain one error code per line.
adapter.ini (Windows and UNIX)	Consists of all the initialization parameters which the adapter reads at startup. Refer to Appendix A for a typical adapter.ini file.

**Table 2–4 Directories**

File	Description
persistence	The messages are persisted in this directory. This directory or its contents should not be edited.
logs	The logging of adapter activity is done in subdirectories of the log directory. Each new run of the adapter creates a new subdirectory in which logging is done in an oailog.txt file.

## Using the Application Parameter

Adapters do not have integration logic. The J.D. Edwards OneWorld adapter has a generic transformation engine that processes metadata from the repository as runtime instructions to do transformations. The application defines for an adapter what its capabilities are. For example, it can define what messages it can publish, what messages it can subscribe to, and what are the transformations to perform. The application parameter allows the adapter to become smart in the context of the application to which it is connected. It allows the adapter to retrieve from the repository only that metadata that is relevant to the application. The application parameter must match the corresponding application that will be defined in *iStudio* under the Applications folder.

If you are using pre-packaged metadata, after importing the pre-packaged metadata into the repository, start up *iStudio* to find the corresponding application (under the Applications folder in *iStudio*) to use as the application for the adapter you are installing (unless the package you are using provides directions for what the application should be).

## adapter.ini Initialization Parameter File

This section contains these topics:

- [Hub.ini](#)
- [Agent Connection Parameters](#)
- [J. D. Edwards OneWorld Adapter Parameters](#)

### Hub.ini

The J.D. Edwards OneWorld adapter connects to the hub database using parameters from the `hub.ini` file located in the hub directory. The following table lists the parameter name, a description for each parameter, the possible and default values, and an example.

Parameter	Description	Example
<code>hub_username</code>	The name of the hub database schema (or username). Possible values are valid hub database username. There is no default value.	<code>hub_username=myhub</code>
<code>hub_password</code>	The password for the hub database user. Possible values are the valid password for the hub database user. There is no default value.	<code>hub_password=manager</code>
<code>hub_host</code>	The name of the machine hosting the hub database. Possible values are the valid machine name. There is no default value.	<code>hub_host=mpjoshpc</code>
<code>hub_instance</code>	The valid SID of the hub database. There is no default value.	<code>hub_instance=orcl</code>
<code>hub_port</code>	The TNS listener port number for the HUB database instance. There is no default value.	<code>hub_port=1521</code>
<code>repository_name</code>	The valid name of the repository this adapter talks to. There is no default value.	<code>repository_name=myrepo</code>

## Agent Connection Parameters

The J.D. Edwards OneWorld adapter connects to the spoke application using parameters from the `adapter.ini` file. The following table lists the parameter name, a description for each parameter, the possible and default values and an example.

Parameter	Description	Example
<code>application</code>	The name of the application this adapter connects to. This must match with the name specified in iStudio during creating of metadata. Any alphanumeric string can be used. There is no default value.	<code>application=aqapp</code>
<code>partition</code>	The partition this adapter handles as specified in iStudio. Any alphanumeric string is a possible value. There is no default value.	<code>partition=germany</code>
<code>instance_number</code>	To have multiple adapter instances for the given application with the given partition, each adapter should have a unique instance number. Possible values are any integer greater than 1. There is no default value.	<code>instance_number=1</code>
<code>agent_log_level</code>	Specifies the amount of logging necessary. Possible values are: 0=errors only 1=status and errors 2=trace, status, and errors The default value is 1.	<code>agent_log_level=2</code>
<code>agent_subscriber_name</code>	The subscriber name used when this adapter registers its subscription. The possible value is a valid Oracle Advanced Queuing subscriber name and there is no default value.	<code>agent_subscriber_name=aqapp</code>
<code>agent_message_selector</code>	Specifies conditions for message selection when registering its subscription with the hub. The possible value is a valid Oracle Advanced Queuing message selector string. There is no default value.	<code>agent_message_selector=recipient_list like '%aqapp,%'</code>
<code>agent_reply_subscriber_name</code>	The subscriber name used when multiple adapter instances for the given application with the given partition are used. Optional if there is only one instance running. The possible value is application name (parameter: <code>application</code> ) concatenated with instance number (parameter: <code>instance_number</code> ). There is no default value.	If <code>application=aqapp,</code> <code>instance_number=2,</code> then, <code>agent_reply_subscriber_name=aqapp2</code>

Parameter	Description	Example
agent_reply_message_selector	Used only if multiple adapter instances for the given application with the given partition. The possible value is a string built using concatenating application name (parameter:application) with instance number (parameter:instance_number). There is no default value.	If application=aqapp, instance_number=2, then agent_reply_message_selector=recipient_list like '%,aqapp2,%'
agent_tracking_enabled	Specifies if message tracking is enabled. Set to false to turn off all tracking of messages. Set to true to track messages with tracking fields set in iStudio. Possible values are true or false. The default value is true.	agent_tracking_enabled=true
agent_throughput_measurement_enabled	Specifies if throughput measurement is enabled. Set to true to turn on all throughput measurements. Possible values are true or false. The default value is true.	agent_throughput_measurement_enabled=true
agent_use_custom_hub_dtd	Specifies if a custom DTD should be used for the common view message when handing it to the hub. By default adapters use an Oracle9iAS InterConnect-specific DTD for all messages sent to the hub as other Oracle9iAS InterConnect adapters will be retrieving the messages from the hub and know how to interpret them. Set to true if for every message, the DTD imported for the message of the common view is to be used instead of the Oracle9iAS InterConnect DTD. Only set to true if a Oracle9iAS InterConnect adapter is not receiving the messages from the hub. Possible values are true or false. There is no default value.	agent_use_custom_hub_dtd=false
agent_metadata_caching	Specifies the metadata caching algorithm. Possible values are: <ul style="list-style-type: none"> <li>■ startup—Cache everything at startup. This may take a while if there are a lot of tables in the repository.</li> <li>■ demand—Cache metadata as it is used.</li> <li>■ none—No caching. This slows down performance.</li> </ul> The default value is demand.	agent_metadata_caching=demand



Parameter	Description	Example
agent_dvm_table_caching	<p>Specifies the DVM caching algorithm. Possible values are:</p> <ul style="list-style-type: none"> <li>▪ startup—Cache all DVM tables at startup. This may take a while if there are a lot of tables in the repository.</li> <li>▪ demand—Cache tables as they are used.</li> <li>▪ none—No caching. This slows down performance.</li> </ul> <p>The default value is demand.</p>	agent_dvm_table_caching=demand
agent_lookup_table_caching	<p>Specifies the lookup table caching algorithm. Possible values are:</p> <ul style="list-style-type: none"> <li>▪ startup—Cache all lookup tables at startup. This may take a while if there are a lot of tables in the repository.</li> <li>▪ demand—Cache tables as they are used.</li> <li>▪ none—No caching. This slows down performance.</li> </ul> <p>The default value is demand.</p>	agent_lookup_table_caching=demand
agent_delete_file_cache_at_startup	<p>With any of the agent caching methods enabled, metadata from the repository is cached locally on the file system.</p> <p>Set this parameter to <code>true</code> to delete all cached metadata on startup.</p> <p>Note: After changing metadata or DVM tables for this adapter in iStudio, you must delete the cache to guarantee access to the new metadata or table information.</p> <p>Possible values are <code>true</code> or <code>false</code>. The default value is <code>false</code>.</p>	agent_delete_file_cache_at_startup=false
agent_max_ao_cache_size	Specifies the maximum number of application objects' metadata to cache. Possible values are any integer greater than 1. The default value is 200.	agent_max_ao_cache_size=200
agent_max_co_cache_size	Specifies the maximum number of common objects' metadata to cache. Possible values are any integer greater than 1. The default value is 100.	agent_max_co_cache_size=100
agent_max_message_metadata_cache_size	Specifies the maximum number of messages' metadata to cache (publish/subscribe and invoke/implement). Possible values are any integer greater than 1. The default value is 200.	agent_max_message_metadata_cache_size=200

Parameter	Description	Example
agent_max_dvm_table_cache_size	Specifies the maximum number of DVM tables to cache. Possible values are any integer greater than 1. The default value is 200.	agent_max_dvm_table_cache_size=200
agent_max_lookup_table_cache_size	Specifies the maximum number of lookup tables to cache. Possible values are any integer greater than 1. The default value is 200.	agent_max_lookup_table_cache_size=200
agent_max_queue_size	Specifies the maximum size that internal Oracle9iAS InterConnect message queues can grow. Possible values are any integer greater than 1. The default value is 1000.	agent_max_queue_size=1000
agent_persistence_queue_size	Specifies the maximum size that internal Oracle9iAS InterConnect persistence queues can grow. Possible values are any integer greater than 1. The default value is 1000.	agent_persistence_queue_size=1000
agent_persistence_cleanup_interval	Specifies how often the persistence cleaner thread should run. Possible values are any integer greater than 30000. The default value is 60000.	agent_persistence_cleanup_interval=60000
agent_persistence_retry_interval	Specifies how often the persistence thread should retry when it fails to push a Oracle9iAS InterConnect message. Possible values are any integer greater than 5000. The default value is 60000.	agent_persistence_retry_interval=60000
service_path	Windows only. The value that the environment variable PATH should be set to. Path is set to the specified value before forking the Java VM. Typically, all directories containing all necessary DLLs should be listed here. Possible values are the valid path environment variable setting. There is no default value.	service_path=%JREHOME%\bin;D:\oracle\ora902\bin
service_classpath	The classpath used by the adapter Java VM. If a custom adapter is developed and as a result, the adapter is to be used to pick up any additional jars, add the jars to the existing set of jars being picked up. Possible values are the valid classpath. There is no default value.	service_classpath=D:\oracle\ora902\oai\902\lib\oai.jar;%JREHOME%\lib\il8n.jar;D:\oracle\ora902\jdbc\classes12.zip
service_class	The entry class for the Windows NT service. The possible value is oracle/oai/agent/service/AgentService. There is no default value.	service_class=oracle/oai/agent/service/AgentService
service_max_java_stack_size	Windows only. The maximum size to which the Java VM's stack can grow. Possible values are the valid Java VM maximum native stack size. The default value is the default for the Java VM.	service_max_java_stack_size=409600

Parameter	Description	Example
<code>service_max_native_stack_size</code>	Windows only. The maximum size to which the Java VM's native stack can grow. Possible values are the valid Java VM maximum native stack size. The default value is the default for the Java VM.	<code>service_max_native_size=131072</code>
<code>service_min_heap_size</code>	Windows only. Specifies the minimum heap size for the adapter Java VM. Possible values are the valid Java VM heap sizes. The default value is the default Java VM heap size.	<code>service_min_heap_size=536870912</code>
<code>service_max_heap_size</code>	Windows only. Specifies the maximum heap size for the adapter Java VM. Possible values are any valid Java VM heap sizes. The default value is 536870912.	<code>service_max_heap_size=536870912</code>
<code>service_num_vm_args</code>	Windows only. The number of <code>service_vm_arg&lt;number&gt;</code> parameters specified. Possible values are the number of <code>service_vm_arg&lt;number&gt;</code> parameters. There is no default value.	<code>service_num_vm_args=1</code>
<code>service_vm_arg&lt;number&gt;</code>	Windows only. Specifies any additional arguments to the Java VM. For example, to get line numbers in any of the stack traces, set <code>service_vm_arg1=java.compiler=NONE</code> . If there is a list of arguments to specify, use multiple parameters as shown in the example by incrementing the last digit starting with 1. Be sure to set the <code>service_num_vm_args</code> correctly. Possible values are any valid Java VM arguments. There is no default value.	<code>service_vm_arg1=java.compiler=NONE</code> <code>service_vm_arg2=oai.adapter=.aq</code>
<code>service_jdk_version</code>	Windows only. The JDK version the adapter Java VM should use. The default value is 1.3.1.	<code>service_jdk_version=1.3.1</code>
<code>service_jdk_dll</code>	Windows only. The dll the adapter Java VM should use. The default value is <code>jvm.dll</code> .	<code>service_jdk_dll=jvm.dll</code>

## J. D. Edwards OneWorld Adapter Parameters

The following table lists the parameters specific to the J.D. Edwards OneWorld adapter.

Parameter	Description	Example
bridge_class	This indicates the entry class for the J. D. Edwards OneWorld adapter. Do not modify this value. A possible value is <code>com.actional.oai.Agent</code> . There is no default value.	<code>bridge_class=com.actional.oai.Agent</code>

## Starting the J. D. Edwards OneWorld Adapter

Start the J. D. Edwards OneWorld adapter using the `start` script in the directory named after the J.D. Edwards OneWorld adapter on Windows NT, UNIX, or HP.

On Windows NT or Windows 2000, start it from the Service window available from the Start menu.

1. Access the Services window from the Start menu:

On...	Choose...
Windows NT	Start > Settings > Control Panel > Services
Windows 2000	Start > Settings > Control Panel > Administrative Tools > Services

The Services window displays.

2. Select the *OracleHome9iASInterConnectAdapter-Application* service.
3. Start the service based on your operating system:

On...	Choose...
Windows NT	Choose Start.
Windows 2000	Right click the service and choose Start from the menu that displays.

---

## Supported J. D. Edwards Interfaces

J. D. Edwards OneWorld is a software application allowing organizations to maintain a single source for customer information. Using the J.D. Edwards OneWorld adapter, an organization can facilitate the sales and marketing forces by combining all customer information into one source. This source can be accessed using multiple channels. A field sales person can consult the Web, the company call center, or any number of networks to access information needed for a customer.

This chapter discusses the following topics:

- [Using the J. D. Edwards OneWorld Adapter](#)
- [Creating an Implemented Procedure](#)
- [Creating a Subscribed Event](#)

## Using the J. D. Edwards OneWorld Adapter

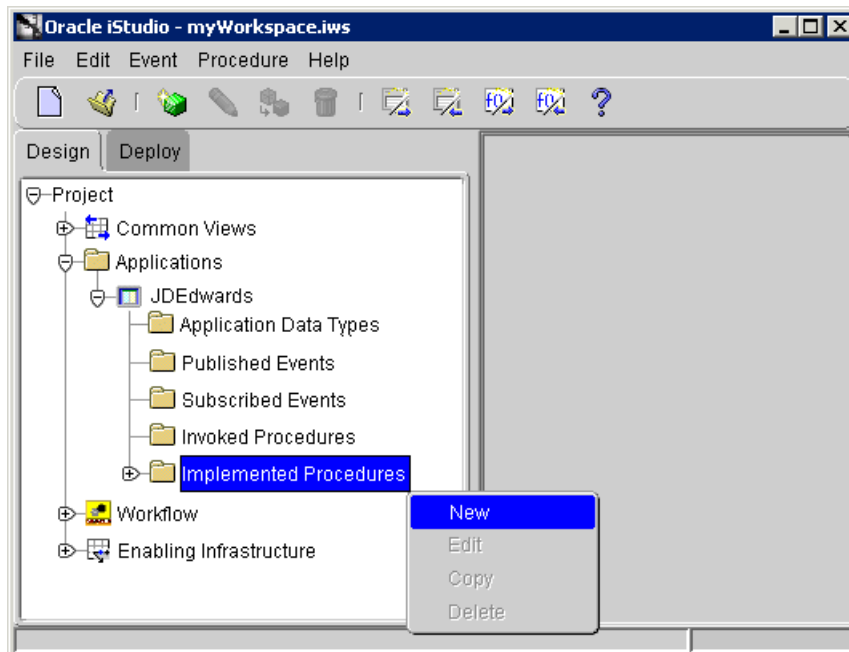
J.D. Edwards OneWorld adapter allows Oracle9iAS InterConnect users to browse the J.D. Edwards business functions, create events and procedures in iStudio, and execute the business functions.

### Creating an Implemented Procedure

To create an implemented procedure using iStudio:

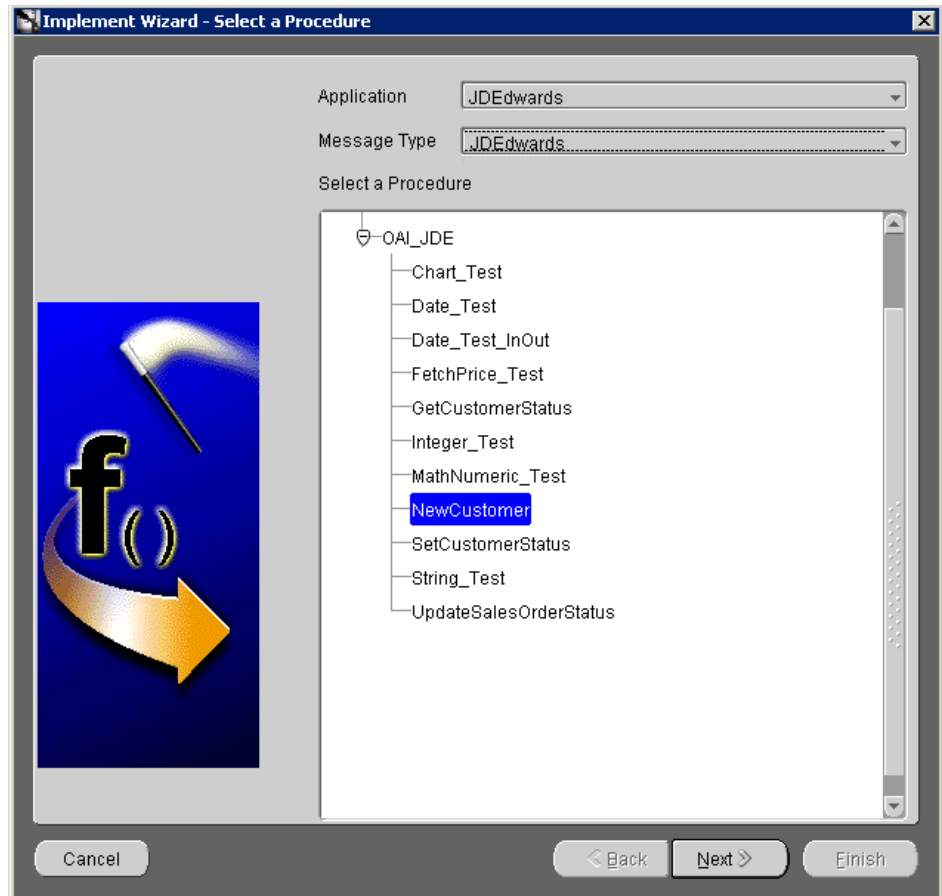
1. Start iStudio.
2. Open your project.
3. Expand your project.
4. Expand the Applications folder.
5. Right-click **Implemented Procedures** and select **New**.

*Figure 3–1 Creating an Implemented Procedure*



The Implement Wizard—Select a Procedure dialog displays.

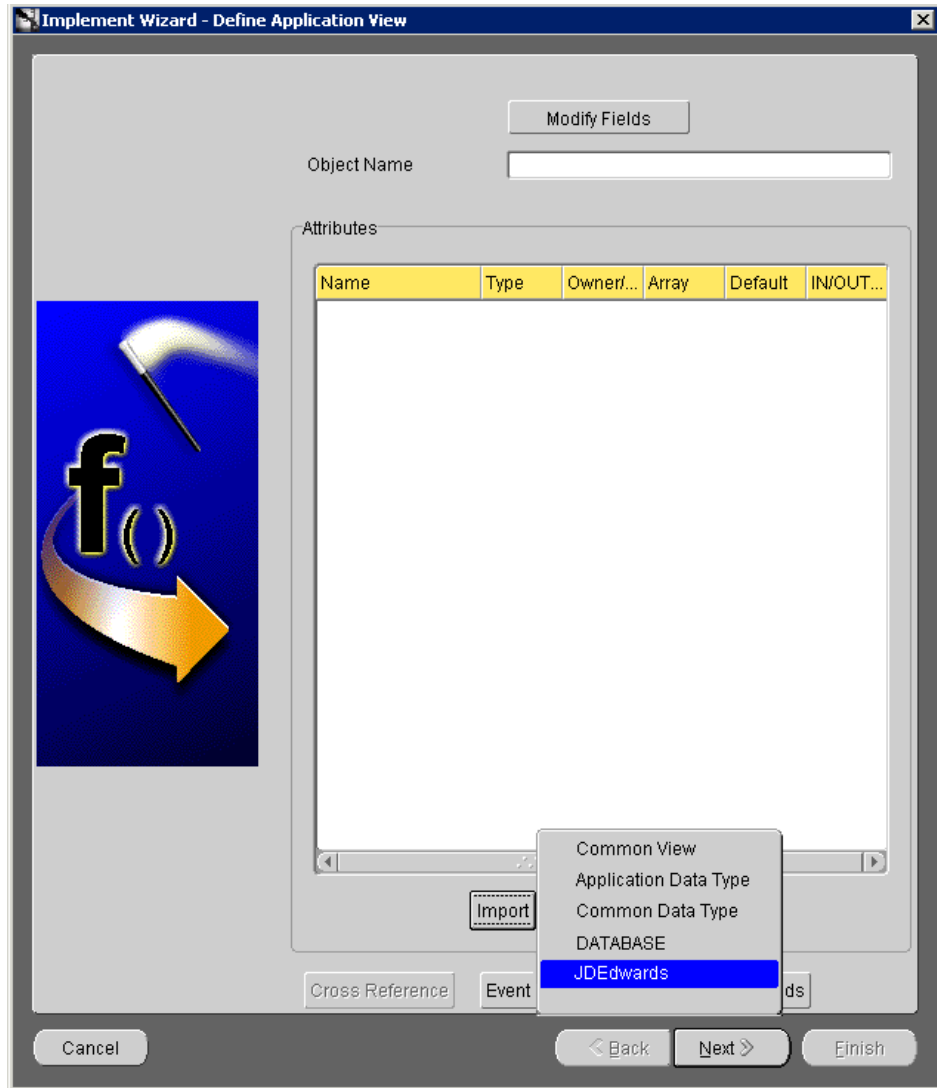
**Figure 3-2** *Selecting a Procedure*



6. Select the Application and Message Type from the dropdown menu.
7. Select a procedure and click **Next**.

The Implement Wizard—Define Application View dialog displays.

**Figure 3–3** *Defining the Application View - Importing J.D. Edwards*



8. Import attributes from J.D. Edwards.



If this is the initial login for this machine, you must log in to J.D. Edwards.

**See Also:** ["Importing Attributes from J.D. Edwards"](#) on page 3-6

If this machine has been previously logged in to J.D. Edwards, enter the password on the JDE Login dialog and click **OK**.

After logging in to J.D. Edwards and importing attributes, the Define Application View dialog is populated with the selected component and attributes.

**Figure 3-4** *Populated Define Application View*



9. Click **Next** to defining the mappings.

## Importing Attributes from J.D. Edwards

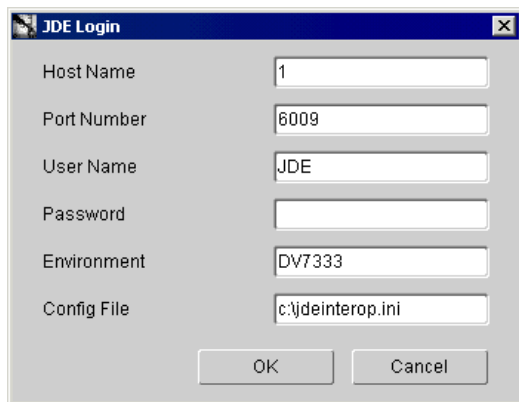
When you use iStudio to import attributes from J.D. Edwards, you must first log into J.D. Edwards OneWorld adapter. When logging into J.D. Edwards from iStudio, the login fields automatically populate, leaving the Password field the only field that requires input.

To import attributes from J.D. Edwards:

1. Click **Import** on the Define Application View dialog and select **J.D. Edwards**.

The JDE Login dialog displays.

**Figure 3–5 J. D. Edwards Login Dialog**



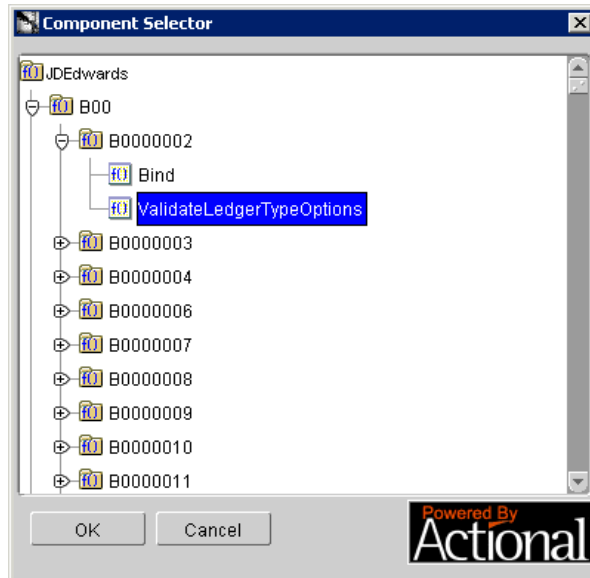
Host Name	1
Port Number	6009
User Name	JDE
Password	
Environment	DV7333
Config File	c:\jdeinterop.ini

The first time you log in from a new workstation, you are required to enter information in every field that is required for your setup. Every subsequent login from that workstation only requires a password to log in. For every iStudio session, only one login is required.

2. Enter information in the following fields:
  - Host Name—The host name.
  - Port Number—The port number.
  - User Name—The user name.
  - Password—The password for the system.
  - Environment—The environment.

- **Config File**—The directory containing the configuration file. This should point to `...\oai\9.0.2\JDE\jdeinterop.ini`.
3. Click **OK**.  
The Component Selector dialog displays.

**Figure 3–6** Component Selector Dialog



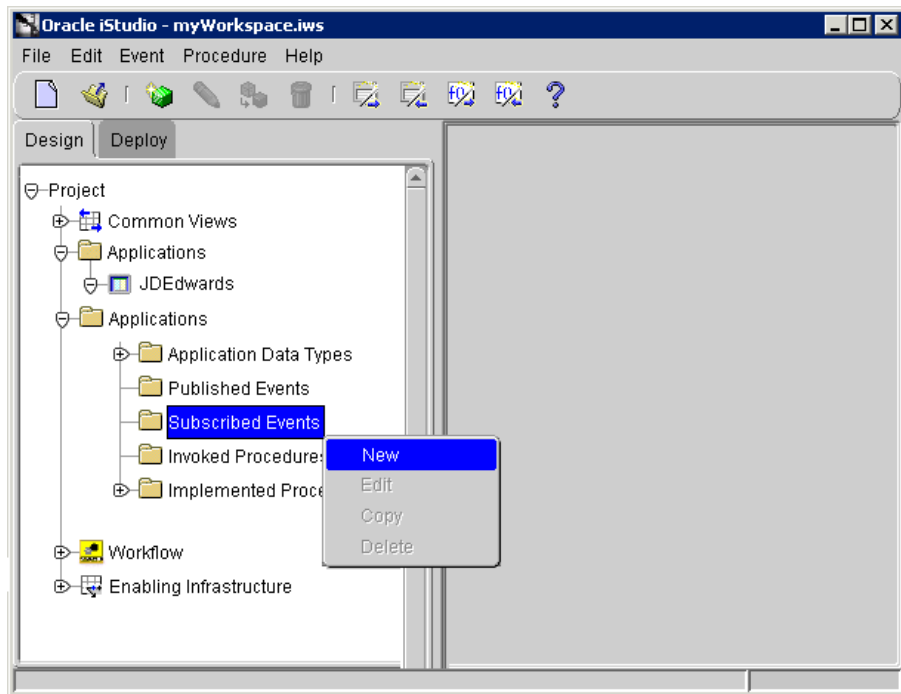
4. Expand the **JDEdwards** tree and select the node where your component is located.
5. Click **OK**.

## Creating a Subscribed Event

To create a Subscribed Event:

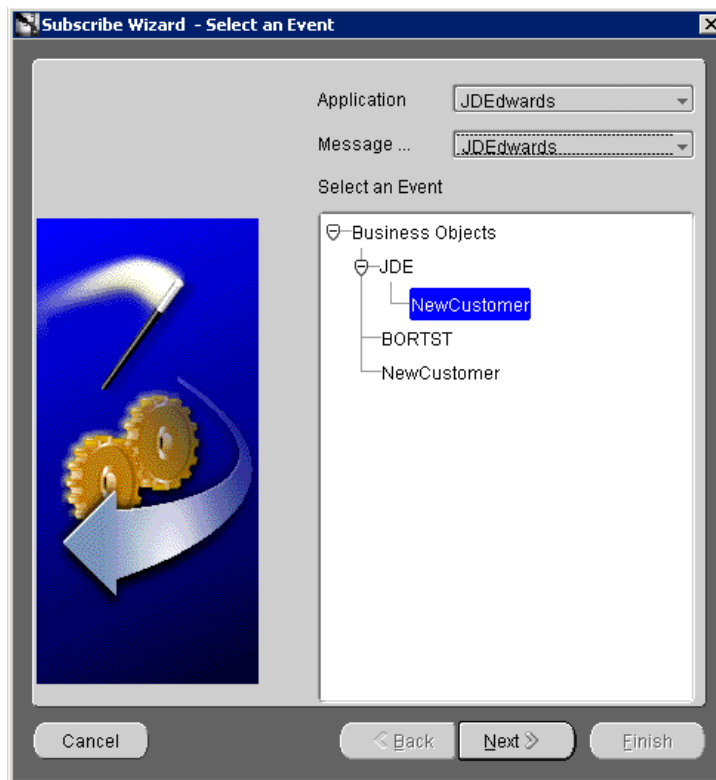
1. Start iStudio.
2. Open and expand your Project.
3. Expand the Applications folder.
4. Right-click **Subscribed Events** and select **New**.

*Figure 3–7 Creating a New Subscribed Event*



The Subscribe Wizard—Select an Event dialog displays.

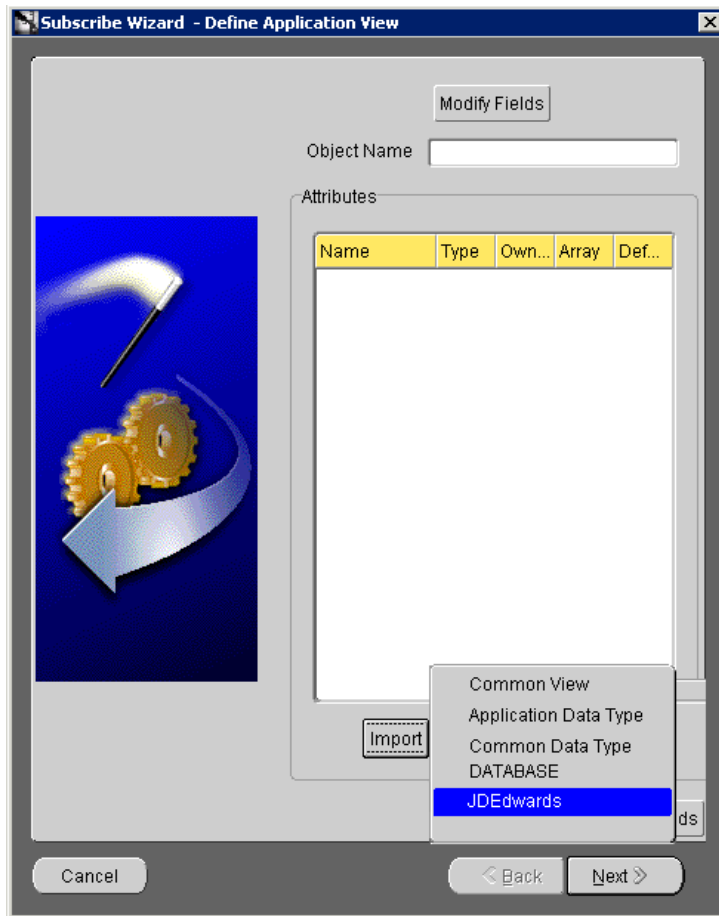
**Figure 3–8** *Selecting an Event Dialog*



5. Select the Application and Message Type from the dropdown menus.
6. Select an event dialog and click **Next**.

The Define Application View dialog displays.

**Figure 3–9** *Defining the Application View - Importing J. D. Edwards*



7. Import attributes from J.D. Edwards.

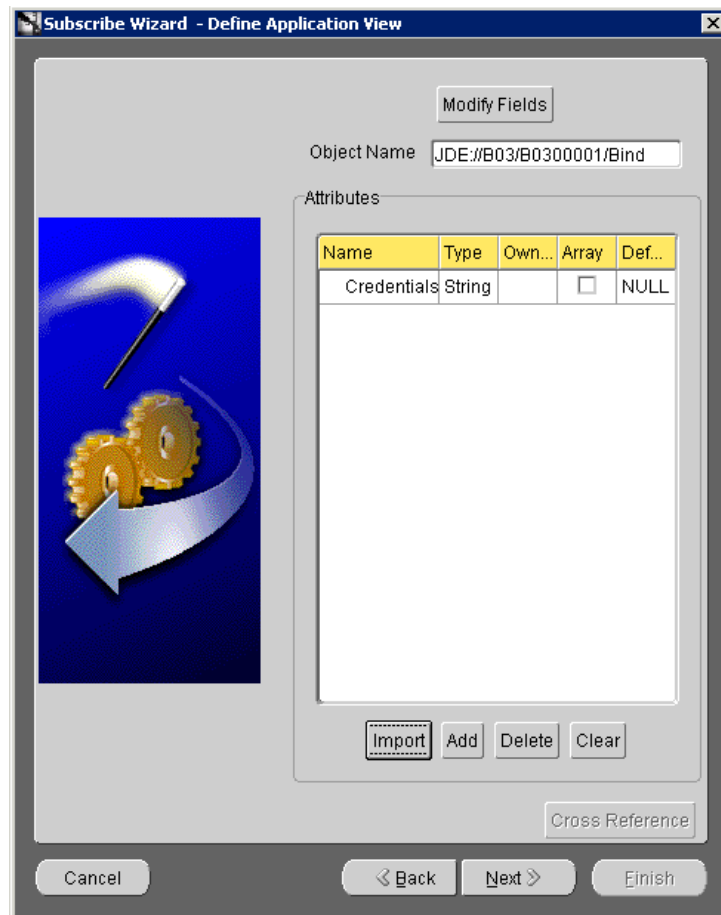
If this is the initial login for this machine, you must log in to J.D. Edwards.

**See Also:** ["Importing Attributes from J.D. Edwards"](#) on page 3-6

If this machine has been previously logged in to J.D. Edwards, enter the password on the JDE Login dialog and click **OK**.

After logging in to J.D. Edwards and importing attributes, the Define Application View dialog is populated with the selected component and attributes.

**Figure 3–10** Populated Subscribed Wizard - Define Application View



8. Click **Next**.
9. Click **New** to define the mappings, and click **Finish**.

The newly populated event displays in the right panel of iStudio.



---

---

## Using the Configuration Editor

This chapter describes how to use the Configuration Editor to configure the J.D. Edwards OneWorld adapter. The J.D. Edwards credentials are used to retrieve metadata about the J.D. Edwards source modules and their business functions. The default credentials are used at runtime to submit requests to the business functions.

This chapter discusses the following:

- [Configuration Editor](#)
- [Log into J.D. Edwards](#)

---

---

**Note:** Profiles and deployment are sensitive to the Master Key setting. If using a shared machine, before accessing the Configuration Editor ensure the Master Key is set to either that of User1 or create a new Master Key for your profiles. Refer to the *Oracle9iAS InterConnect Configuration Editor User's Guide* for more information on the Master Key.

---

---

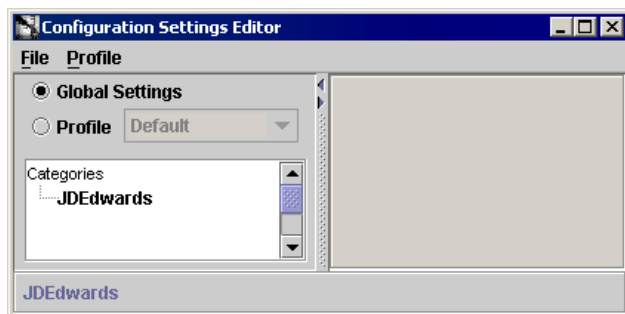
## Configuration Editor

To configure settings for the J.D. Edwards OneWorld adapter, access the J.D. Edwards Configuration Editor as follows:

1. Change directories to the `.../oai/9.0.2/config/configeditor` using a DOS prompt.
2. Type `configeditor` and click **Enter**.

The Configuration Settings Editor displays.

*Figure 4–1 Configuration Settings Editor Displaying J. D. Edwards*



3. Click **Profile**.
4. Select the **iStudio** profile from the dropdown list.

---

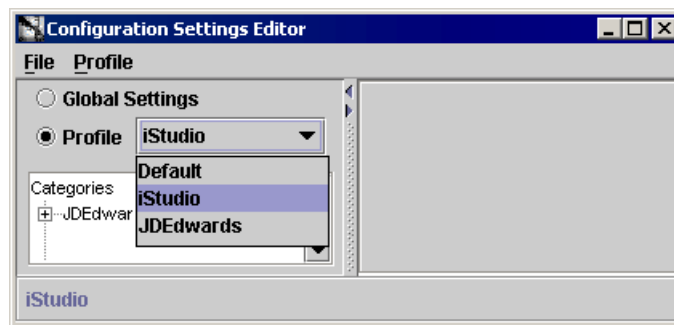
---

**Note:** Under some circumstances you may wish to run your adapter under a profile other than iStudio. This may be needed, for example, if you want to run two instances of the J. D. Edwards adapter on the same machine. You may want to have two instances of the same type of adapter if these instances need to connect to different backend system installations. To accomplish this, create a new profile using the configuration editor and fill in the settings for this new profile. The name of the new profile should be the same as the name of the application. For example, if your application is called APP2, create a profile called APP2. Now APP2 will use the settings in the profile called APP2 whenever it runs.

---

---

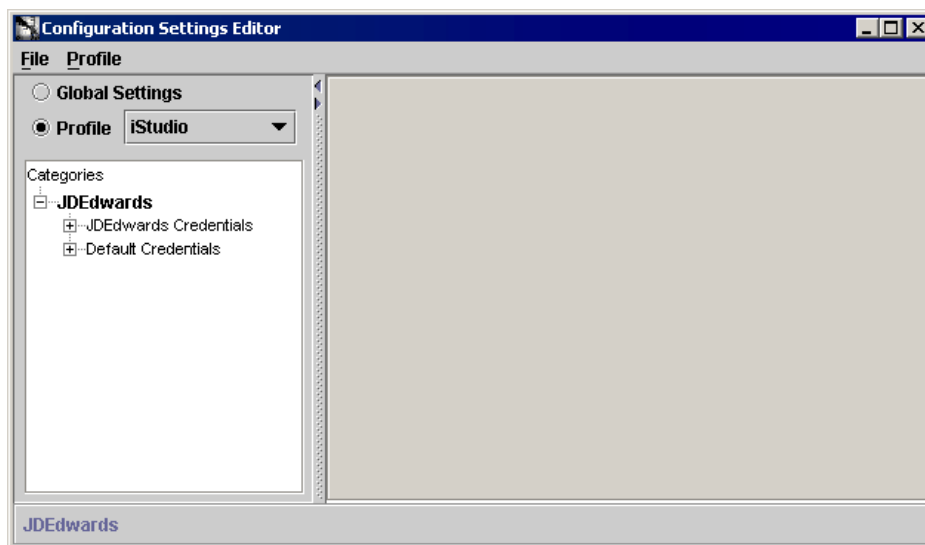
**Figure 4–2** Configuration Settings Editor Profile Menu



5. Double-click **J. D. Edwards** in the Categories pane to edit the configuration settings for the iStudio profile.

The Configuration Settings Editor dialog displays.

**Figure 4–3** Configuration Settings Editor: J.D. Edwards Configuration



## Log into J.D. Edwards

The login credentials under both the **JDEdwards Credentials** and **Default Credentials** branches are used by the J.D. Edwards OneWorld adapter. The JDEdwards credentials are used to retrieve metadata about the J.D. Edwards source modules and their business functions. The default credentials are used at runtime to submit requests to the business functions.

---

**Note:** Follow the same procedure to modify the **JDEdwards Credentials**.

---

## Default Credentials

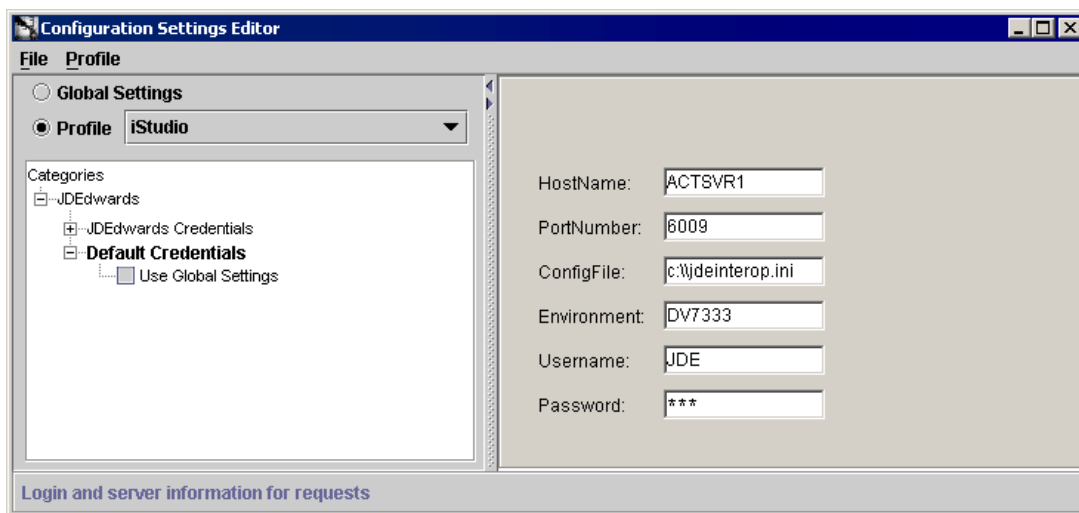
The default login to the J.D. Edwards group allows you to program your development application to automatically connect to J. D. Edwards servers. The default login to J.D. Edwards authenticates your runtime credentials.

From the Configuration Editor main menu:

1. Expand **Default Credentials** in the Categories pane.
2. Select **Use Global Settings** to edit the configuration settings for your profile.

The Configuration Settings Editor login property sheet displays in the right pane of the dialog.

**Figure 4–4 Configuration Settings Editor Login Dialog**



3. Enter the information in the following fields:
  - **HostName**—The name of the host server.
  - **PortNumber**—The port number that will exchange data.
  - **ConfigFile**—The path for the configuration file. This should include the path `... \oai\9.0.2\config\JDE\jdeinterop.ini` to the `jdeinterop.inifile`.
  - **Environment**—The environment variables.
  - **Username**—The username of the person that is accessing this information.
  - **Password**—The password for the user that is accessing this information.

When a connection is made to J.D. Edwards OneWorld, parameters are passed to the connection object (User, Password, Environment) and an instance of the J.D. Edwards OneWorld Application component is returned. These credentials are further defined by the name of the Enterprise/Application server, and the defined TCP/IP port the on which the service listens.

---

---

**Login Failure:** If a login fails, it is not possible to tell if it is because the credentials are wrong or there is a connection/socket/network/DBerror. If you receive login errors, check the credentials as well as the `jdeinterop.ini` file.

---

---

---

---

## Using the J. D. Edwards Adapter

This chapter provides an overview of the functionality of the J.D. Edwards OneWorld adapter. The following topics are discussed:

- [Control Flow Overview](#)
- [Business Functions](#)

## Control Flow Overview

The following topics discuss control flow.

### Design Time

When the J.D. Edwards OneWorld adapter loads, using the login credentials and system information from the Configuration Editor, one or more instances of the J.D. Edwards OneWorld Application business function are created and pooled. When you browse the namespace, there is a list of business functions. Clicking a business function displays its logical methods along with the method's signatures.

### Runtime

Instances of the J.D. Edwards OneWorld Application business function are created and pooled for each thread. When a method call is submitted to a business service the metadata for this is read using the J.D. Edwards OneWorld application business function (if it has not already been cached) and this makes a call to the appropriate method. It dynamically (at runtime) constructs the J.D. Edwards Interface layer as needed through which J.D. Edwards OneWorld adapter supports invocation and necessary data conversions.

The J.D. Edwards OneWorld adapter maps interface descriptions of the method signatures of the J.D. Edwards application, so that other applications can interact with these interface descriptions in a format that it can understand. It also adapts to system models of other applications and the J.D. Edwards application to provide end-to-end interaction.

The J.D. Edwards OneWorld adapter enables applications in the enterprise to interact with the J.D. Edwards application, by extending functionality from the J.D. Edwards application in the form of one or more of the following as applicable:

- Native data formats
- Procedures
- Methods
- Messages
- Properties
- Application interfaces



You can use the J.D. Edwards OneWorld adapter browser's Point of View functionality to view the elements of the J.D. Edwards application in its native format. The J.D. Edwards OneWorld adapter builds a description of application interfaces for other applications interacting with the J.D. Edwards application at runtime. It can create, delete, and invoke business objects as needed to perform computations in the J.D. Edwards application and it directly invokes methods. All calls into the J.D. Edwards OneWorld Application are synchronous calls. The J.D. Edwards OneWorld adapter transforms the data for the call into Java types.

### **Apartment Threading of J.D. Edwards OneWorld Business Functions**

The J.D. Edwards OneWorld Application business function and any business function instance that is obtained through it can only be used on the thread where they are created or obtained (essentially apartment threading). The J.D. Edwards OneWorld adapter's connection pooling framework manages a pool of these that can be utilized.

### **Connection Pooling**

Connection pooling improves performance of calls from the client system to the other system. It keeps connections to the server systems open and reuses them instead of closing them down after each call. The J.D. Edwards OneWorld adapter enables you to pool connections within particular logins, but still maintains critical control of the total number of connections across all pools.

Any new business function instance uses the thread on which it is created and the instance is destroyed after each operation. All J.D. Edwards calls to business functions are stateless. However, during the operation, the J.D. Edwards OneWorld adapter ensures that the business function is used on the correct thread.

## **Business Functions**

The J.D. Edwards OneWorld adapter only provides access to J.D. Edwards business functions. Metadata about business functions is read using a business function interface to find a list of business functions and associated data structures. Metadata is strongly typed in all cases for all business function methods.

When executing business functions, the J.D. Edwards OneWorld adapter dynamically builds and compiles C++ function wrapper classes, using a subset of the genjava output required to execute the business function.

All business function methods have the same calling convention, three parameters that are system derived, and a pointer to a data structure.

Business function data types are represented as internal to J.D. Edwards in the following manner:

**Table 5–1 Business Function Data Types**

<b>Data Type</b>	<b>Description</b>
char	Character string
int	Signed or unsigned short integer.
long	Signed or unsigned long integer.
MATH_NUMERIC	JD Edwards special implementation of floating point numbers, including currency values.
JDEDATE	JD Edwards special implementation of dates.
BYTE	A single unsigned character.
BOOL	An integer.
ID	An unsigned long integer.
HWND	Not supported.

## Handling String Values

The following describes how to configure certain string arguments as right justified (and left padded). J. D. Edwards OneWorld exposes two kinds of string values through its interoperability layer:

- char—A single character.
- maximum length string—A null terminated string with a maximum length.

When you view arguments of these types in the J.D. Edwards OneWorld adapter browser from a Point of View such as Automation Visual Basic, you see descriptions such as:

- char
- char array of max length <n>

The length <n> does not include the terminating null character.

You will also notice that J.D. Edwards uses Hungarian notation to name the arguments of these types in the business functions. For example, arguments of these types will begin with:

- c
- SZ

For nearly all arguments of the sz type, maximum length string or char array, J.D. Edwards expects a left justified value. For a street address line, which is of maximum length 40, J.D. Edwards expects:

```
"701 North Shoreline Blvd"
```

padded to length 40 with blanks. It is not necessary for you to enter the padding as the J.D. Edwards OneWorld adapter provides this for you. You only need to enter "701 North Shoreline Blvd" in your client code.

For some subset of values for this type, dialog expects values that are right justified with padding on the left. For example, for business functions in the B4200310 source module, the argument szBusinessUnit is of length 12. This argument represents a Plant, such as a production facility. For a plant number of 30, J.D. Edwards expects a value of:

```
"          30"
```

The following values are not valid:

```
"30"  
"30 "
```

To make it easier for you to submit these values, you can take advantage of a J.D. Edwards OneWorld adapter feature. It is possible for you to enter a list of these parameters so that they are automatically right justified and padded on the left with blanks. In this way, the following value would be valid for the szBusinessUnit value:

```
"30"
```

This value automatically formats correctly if you take advantage of this feature. You must create a text file with entries describing these parameters. This file, if you create it, is called:

```
%ACBDIR%\config\JDE\jdearglist.txt
```

where %ACBDIR% is the directory in which you installed the J.D. Edwards OneWorld adapter. If this file does not exist, or is empty, you will see an informational message in the J.D. Edwards OneWorld adapter log when you first load the J.D. Edwards OneWorld adapter.

The following is an example of the format for entries in the `jdearglist.txt` file:

```
<SourceModule>.<BusinessFunction>.<Argument>
```

For example:

```
B4200310.F4211FSBeginDoc.szBusinessUnit  
B4200310.F4211FSBeginDoc.szBusinessUnit
```

For a set of business functions belonging to the same business module, like-named arguments of the same type share some or all of the business functions. You can use the asterisk '\*' wildcard character instead of the business function name. For example:

```
B4200310.*.szBusinessUnit
```

## Using the MATH\_NUMERIC Type

The following describes the MATH\_NUMERIC type and details:

- How exponents are handled
- The maximum number of digits
- The maximum number of decimal digits

The MATH\_NUMERIC type is a numeric string type. To use it, enter parameter values of the following format:

```
<OptionalSign><IntegerAndFractionalPart><OptionalExponentPart>
```

Where

- `<OptionalSign>` can be '+' or '-'. '+' is the default.
- `<IntegerAndFractionalPart>` is a maximum of 32 significant digits, not counting the decimal symbol. The decimal symbol is locale-specific to the J.D. Edwards OneWorld installation—typically a period '.' or a comma ','. The digits may be all integer, all fraction, or part integer and part fraction but cannot exceed 32.
- `<OptionalExponentPart>` is in turn made up of:

'e' <OptionalSign><ExponentDigits>

where:

- \* <OptionalSign> can be '+' or '-'. '+' is the default.

- \* <ExponentDigits> are at most two digits. You are permitted values between 63 and -63 excluding zero.

Examples of valid MATH\_NUMERIC values include:

- 123.045
- 4089 (note there is no comma for thousands)
- -9084
- -230.75
- 0.010503
- 1.023e-10 which is equivalent to 0.0000000001023
- 0.097e5 or 0.097e+5 which is equivalent to 9700
- 1.0e-32 which is equivalent to 0.00000000000000000000000000000001

(This is valid because in this case the integral '0' is ignored, 32 significant fractional digits)

Examples of invalid MATH\_NUMERIC values include:

- 1034.0000000000000000000000000000001023—Too many significant digits.
- 1.023e-64—The exponent is too small.
- 0.00317e64—The exponent is too large.

Any non-numeric characters other than those appropriate for signs and decimal symbols result in an invalid value.

## Exponents

Exponents are provided by the J.D. Edwards MATH\_NUMERIC as a convenience for entering values. However, most values return without exponents (with all 32 significant digits visible).

### Invalid Values

Invalid values depend on the kind of value. A decimal fraction that is too small is interpreted as zero (all significant digits are lost).

An integer that has too many significant digits causes unexpected results. J.D. Edwards does not always raise an error condition in this case.

An exponent that is too large or too small returns as an invalid value.

### Precision for Operations

If an operation results in loss of precision, rounding occurs. For example:

```
1.9e-31 / 10.0 = 0.000000000000000000000000000000000000000002
1.9e-31 / 100.0 = 0.000000000000000000000000000000000000000000
```

In other cases, unpredictable results occur, as when a very large positive value is multiplied by another.

$1.01e32 * 2.053e32$  does not yield reliable results and does not raise an error.

For most foreseeable business scenarios, these ranges are not exceeded.

### Currency

When a J.D. Edwards business function expects a currency value, the business function always has a separate parameter for a four character currency code. It is not necessary to pass in this code unless you are using a currency other than the default configured for the OneWorld system.

## Typedef of MATH\_NUMERIC

```
struct tagMATH_NUMERIC {
char String[MAXLEN_MATH_NUMERIC+1];
    /* Just the digits -- no separators */
    char Sign; /* '-' if negative, 0x00 otherwise */
char EditCode; /* The Data Dictionary edit code to Format */
    /* for display */
short nDecimalPosition; /* # of digits from right end of string to */
    /* decimal point*/
short nLength; /* The number of digits in s */
WORD wFlags; /* Processing Flags */
char szCurrency[4]; /* The Currency Code */
short nCurrencyDecimals; /* The Number of Currency Decimals */
short nPrecision; /* The Data Dictionary Size */
};
```

## Typedef of JDEDATE

```
struct tagJDEDATE {  
    short nYear;  
    short nMonth;  
    short nDay;  
};
```





---

---

# Custom Package Creation and Installation

The J.D. Edwards OneWorld adapter requires a custom package to be installed on the J.D. Edwards OneWorld system. This chapter discusses the following topics:

- [Defining a Custom Package](#)
- [Custom Package Creation and Installation Prerequisites](#)
- [Creating a Custom Package](#)
- [Installing the Custom Package ACBREL1](#)
- [List of Modules](#)

## Defining a Custom Package

ACBREL1 is a custom package, or an automated software update in J.D Edwards' terms, that has business functions to extract metadata and custom functions to test the data types. A custom package is a post release deliverable that provides software changes for specific purposes, such as regulatory changes or enhancements. These custom packages are created for specific functionality. For example, ACBREL1 is created to extract metadata and to test the J.D. Edwards data types. When the ACBREL1 custom package is installed it updates selected modules within the J.D. Edwards OneWorld environment. To update, ACBREL1 objects must be merged into the appropriate OneWorld environment.

**See Also:** ["List of Modules"](#) on page 6-9

## Custom Package Creation and Installation Prerequisites

A custom package can be created and installed by the Configurable Networking Computing (CNC) specialists, OneWorld system administrators, or network/server administrators. A custom packages installation requires the following:

- A completely installed OneWorldXE (release B73 . 3 . 1 and above) environment with defined standard data sources and path codes.
- A valid J.D. Edwards license and product packaging tools.
- A J.D. Edwards documentation CD.
- A valid userID and password for Knowledge Garden, which is located on the J.D. Edwards site.

## Related Documentation

To create a custom package, the following reference documents are required:

- *Package Management*
- *Product Packaging Tips and Techniques*

These documents provide a more detailed explanation of the instructions in this chapter.

## Retrieving the Package Management Reference Document

The following steps explain how to retrieve the *Package Management* reference document from J.D. Edwards Documentation CD.

---

---

**Note:** In order to retrieve this document you must have a valid J.D. Edwards Documentation CD.

---

---

1. Browse the CD to locate and open the `howto.pdf`.
2. Click the main menu on the left hand side of the `howto.pdf`.
3. Locate and click **View books** in the main menu.
4. Click **Technical** in the main menu.
5. Click **Package Management** the main menu.

In the *Package Management* document the *Package Build* provides the information for packaging the source files and defining the custom package.

## Retrieving the Product Packing Tips and Techniques Reference Document

The detailed step-by-step instructions on how to create the product package and make it portable are described on the J.D.Edwards *Product Packaging Tips and Techniques* reference document. The following steps explain how to retrieve the document from J.D.Edwards Knowledge Garden.

---

---

**Note:** In order to download the document from J.D. Edwards Knowledge garden site, a valid userID and password for Knowledge Garden is required.

---

---

1. Go to <http://www.jdedwards.com> and click the Knowledge Garden Logon.  
To log on you must be a client with J.D. Edwards with a Knowledge Garden logon ID and password
2. Sign in with your Knowledge Garden user (logon) ID and password.
3. Click **Search** from the top menu.

A search criteria window appears with two panes. This window has a textbox where you can enter the search criteria and a group of checkboxes where you can select the documentation.

4. Select all the checkboxes on the left hand side of the screen.
5. Enter the OTM-01-0001 document ID in the search criteria textbox and click Search.  
Search results display in the right pane.
6. Click the *Product Packaging Tips and Techniques* reference document link.

## Creating a Custom Package

Creating a custom package involves packaging the source files and defining the custom package and making the custom package portable.

### Basic Steps for Packing the Source Files and Defining the Custom Package

This section provides the basic steps required to pack the source files and create the custom package. Review the *Package Management* reference document before completing the following steps.

**See Also:** ["Retrieving the Package Management Reference Document"](#) on page 6-3

---

---

**Note:** A compiled version of the source files is required to create a package. You retrieve the source files and the packaged binary files in the following directory:

```
<install_directory>\JDE\ESU
```

---

---

**See Also:** ["Installing the Custom Package ACBRELI"](#) on page 6-7

To pack the source files and create the custom package:

1. Run the package assembly:
  - In the package assembly utility, assign the new package a name, description, and path, and select the language.  
The Package Assembly utility is run from the Package and Deployment Tools. For step-by-step instructions on running the package assembly refer to the Package Build section in the Package Management reference document.

2. Enter the foundation location:
  - a. Change the foundation location from the default location displayed in the Foundation Component screen. The default foundation location is determined through the release associated with the package's path code.

This is normally the system directory at the same directory level as your path code.
  - b. Click Foundation on the Package Component Revision dialog and follow the *Package Build* section's instructions in the *Package Management* reference document.
3. The package build copies the help file from the location defined in Help Item Revisions to the package directory. Enter the help file location:

Click Help on the Package Component Revision dialog and follow the instructions in the Package Build section of the Package Management reference document.
4. Enter the database location:

Click Database on the Package Component Revision dialog and follow the instructions in the Package Build section of the Package Management reference document.
5. To add an object to the package, you must add the source files. To add objects to the package:
  - a. Click Objects on the Package Component Revision dialog to display the Object Component dialog.
  - b. Click Browse to examine the source files that you want to add to the package.
6. Add features to the package:
  - a. Click Features on the Package Component Revision dialog to display the Feature Component dialog.
  - b. Click Browse to view the features you want to select.
7. Activate an assembled package:

After the package is assembled, the status remains inactive. To make the status active run the product assembly:

Select Package Assembly from the Package and Deployment Tools menu and check the active/inactive box for the particular package.

## Basic Steps for Making the Custom Package Portable

This section explains how to make the package portable after the package creation process is completed. You require the *Product Packaging Tips and Techniques* reference document to complete the instructions in this section.

**See Also:** ["Retrieving the Product Packing Tips and Techniques Reference Document"](#) on page 6-3

The Product Packaging tool is required for the following steps.

1. The Product Packaging tool requires an out of the box (pristine) copy of the Data Dictionary for the Data Dictionary changes to be included on the CD. Create a (pristine) Data Dictionary by following the step-by-step instructions explained in section 1 of the *Implementation* section in the *Product Packaging Tips and Techniques* reference document.
2. The Product Packaging tool requires an out of the box (pristine) copy of the Control Tables for Control Table changes to be included on an update CD. Create a pristine Control Table set by following the step-by-step instructions in section 2 of the *Implementation* section in the *Product Packaging Tips and Techniques* document.
3. The Product Packaging process creates a new access database that is merged into existing environments during the installation of the update CD. This database is created in the TEMP directory on the workstation where the Product Package CD image is mastered. To copy tables into this database, an ODBC data source is created by following the step-by-step explained in section 3 of the *Implementation* section in the *Product Packaging Tips and Techniques* document.
4. Product packaging requires a set of tables called Change tables. Change tables are created by following the step-by-step instructions that are explained in section 4 of the *Implementation* section in the *Product Packaging Tips and Techniques* document.
5. The package build application must be set in mastering mode in order for an update package to be compressed. To set package build in mastering mode, follow the steps in section 5 of the *Implementation* section in the *Product Packaging Tips and Techniques* document.
6. Defining a CD is the process of creating the data that is placed on your update CD. A software master definition establishes a template and package to use, what build steps to follow, and the directory structure of the final master CD. This outline must be defined, validated, and finally mastered. To start this

process follow the instructions in section 7 of the *Implementation* section in the *Product Packaging Tips and Techniques* document.

7. Validating a CD is the process of validating the software master that was defined to make sure each build step is valid. To start this process follow the instructions in section 8 of the *Implementation* section in the *Product Packaging Tips and Techniques* document.
8. Mastering a CD is the process of executing the software master that was defined to create the update CD image on the workstation. To start this process follow the instructions explained in section 9 of the *Implementation* section in the *Product Packaging Tips and Techniques* document.
9. Check the existence of the `packagename.cab` and `packagename.exe` files in the `$TEMP\final` directory manually for the successful creation of the package.

## Installing the Custom Package ACBREL1

To install the custom package ACBREL1, the *Deployment Server Installation Guide* and the *Installation Workbench Guide* are required.

The following steps provide instructions for the installation process:

1. Log on as user JDE on the deployment server.
2. Insert the CD into the Deployment Server CD drive.
3. Create a new folder called ACBREL1 on the Deployment Server `..(root)/B7` folder.
4. Copy `ACBREL1.exe` and `ACBREL1.cab` from the ACBREL1 Custom Package CD that was created on the newly created ACBREL1 directory.
5. Execute `ACBREL1.exe` from the `../B7/ACBREL1` directory.

The J.D. Edwards Installation Manager automatically starts after the executable file expands.

---

---

**Note:** If the J.D. Edwards Installation Manager does not display, execute the `setup.exe` file located in the `..(root)/B7/ACBREL1` directory.

---

---

6. Click **Next**, then click **Finish** in the OneWorld Deployment Server Setup Type dialog.

When the installation finishes, a dialog informs you whether the installation was successful.

7. Log on to the JDEPLAN environment as user JDE on the deployment server.

---

---

**Note:** If the planner update Electronic Software Update (ESU) that includes SAR #4533357 has not been installed on your system. Select Software Updates from the System Installation Tools menu (GH9612). Input 02 for option 1 in the Processing Options panel. If the planner update ESU that includes SAR #4533357 has been installed on your system, select Application Software Update from the System Installation Tools menu (GH9612).

---

---

8. Double-click the ACBREL1 update on the Work with Application Update screen and click **Next**.
9. Double-click the environment(s) where you want the update installed.
10. If you want the software update to run in unattended mode, verify that Unattended Workbench is checked.
11. If you want the backup of the specs (so that the original specs can be restored) check **Backup** and then click **Next**.
12. Select the plan for the update you are installing on the Work with Installation **Plan** dialog and click **Select**.
13. After the installation is complete, check all the automatically generated PDFs for errors. If errors occur, consult the *J.D. Edwards Software Update Guide* for troubleshooting tips, or contact J.D. Edwards directly.
14. After installation has completed successfully, build a full package build for the environment(s) you applied to the ACBREL1 update. Consult J.D. Edwards documentation on how to build a full package build.



---

## List of Modules

The Custom Package ACBREL1 creates the following objects in J.D. Edwards:

- ACBCHAR01 - TEST CHAR TYPE 01
- ACBCUST - ACB CUSTOMER ID
- ACBDATE01 - TEST DATE TYPE 01
- ACBDEF - ACB FUNCTION TYPE DEFINITION
- ACBFCNT - ACB FUNCTION NAME LIST COUNT
- ACBFUNC - ACB FUNCTION NAME LIST
- ACBFUNCN - ACB FUNCTION NAME
- ACBINT01 - TEST INTEGER TYPE 01
- ACBLIB - LIBRARY
- ACBMATH01 - TEST MATH TYPE 01
- ACBNEWS - ACB NEW STATUS
- ACBORDER - ACB ORDER NUMBER
- ACBPRC - ACB ITEM PRICE
- ACBPROD - ACB PRODUCT ID
- ACBQTY - ACB ITEM QUANTITY
- ACBRES - ACB RESULT INDICATOR
- ACBSTAT - ACB STATUS
- ACBSTR01 - TEST STRING TYPE 01
- ACBTEST - ACB TEST SCREEN
- ACBTEST2 - ACB TEST SCREEN 2
- ACBTEST3 - ACB TEST SCREEN 3
- B5500900 - SUPPORT MODULE
- D5500900 - DATA STRUCTURE
- D5500900A - DATA STRUCTURE
- D5500900B - FETCH PRICE DATA STRUCTURE

- D5500900C - GET CUSTOMER STATUS DATA STRUCTURE
- D5500900D - SET CUSTOMER STATUS DATA STRUCTURE
- D5500900E - UPDATE SALES ORDER STATUS DATA STRUCTURE
- D5500900F - TEST INTEGER
- D5500900G - TEST STRING
- D5500900H - TEST DATE
- D5500900I - TEST CHAR
- D5500900J - TEST MATH NUMERIC
- D5500900K - TEST DATE 2

---

---

# Limitations

This chapter explains the limitations and workarounds when connecting to J.D. Edwards. The following topics are discussed:

- [Querying and Retrieving Lists of Records](#)
- [Controlling Iteration](#)

## Querying and Retrieving Lists of Records

The J.D. Edwards communication architecture is a single message, single reply architecture. You cannot return a list of messages or an array. The underlying code is C++, which calls with a pointer to a single structure, makes changes in the structure, and exits.

You cannot query and retrieve lists of records based on a search criteria using the J.D. Edwards OneWorld adapter due to a limitation with the J.D. Edwards business function architecture.

In J.D. Edwards, database connectivity is provided by using a set of proprietary (and complex) internal function calls. These calls mask the underlying database version by requiring very explicit and low-level calls to create lists of columns to retrieve or update, and create specialized structures for sorting or selection. The sets of APIs are not exposed through the Java (or any other) connectivity method, therefore, recordsets cannot be handled through business functions.

The J.D. Edwards database access APIs are not usable through the Java connector, therefore, managing lists of items (such as customers) can be challenging. It is relatively easy to create business functions in J.D. Edwards that handle a single record, or operate on a group of records, provided that all of the work is performed within the bounds of the J.D. Edwards toolset, but accessing lists of items outside the J.D. Edwards tools is more difficult.

To workaroud this limitation you can create a custom business function in J.D. Edwards that returns a list of record keys based on a query. You must segment the lists because JDENET (J.D. Edwards' internal proprietary messaging API) has a limitation on the message buffer size for managing large or effectively unbounded result sets. The client code must iterate (loop) through successive calls to the business function, until an indicator is returned stating that the list is complete.

### Controlling Iteration

All calls to J.D. Edwards business functions are stateless; therefore, the business function cannot maintain an open cursor and return more rows on request. Positioning information must be passed to the J.D. Edwards business function on each call.

The following is a list of techniques for controlling iteration:

- On the J.D. Edwards side, write the result set to a temporary storage area, such as a file, which returns an ID (such as a file name, or job number) that can be given on successive calls, along with the record number to position the cursor.

Successive calls position within the list are based on the passed-in record number.

---

---

**Note:** Calls through the J.D. Edwards OneWorld adapter can be load-balanced, however, they are eventually served by a single application server based on the credentials and business function being called. Therefore, if a temporary file was created on a server by one call, additional calls are served by the same server. For more information refer to *Object Configuration Mapping* in the *JDE CNC Guides*

---

---

- Position information (such as a primary key value) that can be passed back on the second and subsequent calls, and reissue the query based on the key as an additional parameter.

---

---

**Note:** Of the first two techniques, the preferred method is to use primary key values and reissue the query. It requires the smallest amount of code, and places the optimization and caching burden on the database.

---

---

- A list of primary keys (such as a cross reference) can be stored by the calling application. For example, if a customer record is created by a Customer Relationship Management (CRM) system, then added to J.D. Edwards using a business function call, the business function that adds a customer record sets the value for the AN8 field (short address number) and is visible in the return buffer. This number can then be written to a reference field on the original customer record, or stored into a custom cross-reference table.
- Most master records in J.D. Edwards have a concept of a lookup, or alternate key. This key can be used to store the key information from the calling system and business functions exist, or can be created to perform the lookup on the J.D. Edwards side. When parameters are passed to the business function to create a customer record, the long key value is set.

**See Also:** *Interoperability* topic in the J.D. Edwards help system



---

---

# Index

## A

---

acbrel 1  
installing, 6-7  
advanced queuing adapter  
installation, 2-2  
application  
starting associated with jdedwards, 2-12  
application parameter, 2-5

## B

---

business functions, 5-3  
handling string values, 5-4

## C

---

configuration, 2-4  
adapter ini parameter file, 2-6  
advanced queuing adapter parameters, 2-12  
agent connection parameters, 2-7  
directories, 2-5  
executable files, 2-4  
files, 2-5  
hub.ini, 2-6  
configuration editor, 4-2  
control flow  
overview, 5-2  
create  
custom package, 6-4  
implemented procedure, 3-2  
subscribe event, 3-8  
currency, 5-8  
custom package

create, 6-4  
creation and installation prerequisites, 6-2  
defining, 6-2  
installing acbrel 1, 6-7  
packing source files, defining custom  
package, 6-4  
related documentation, 6-2  
retrieving the package management reference  
document, 6-3  
retrieving the product packing tips and  
techniques reference documentation, 6-3  
customize  
jdeinterop.ini file, 1-3

## D

---

default credentials, 4-4  
design time, 5-2

## E

---

event  
creating a subscribed, 3-8  
exponents, 5-7

## I

---

installation, 2-2  
tasks, 2-2  
invalid values, 5-8  
iteration  
controlling, 7-2

## J

---

jdedate

- typedef, 5-9

jdedwards

- communication architecture, 1-2

- configuration, 2-4

- directories, 2-4

- installation, 2-2

- login, 3-6, 4-4

jdedwards adapter

- using, 3-2

## L

---

log in

- default credentials, 4-4

## M

---

math\_numeric

- typedef, 5-8

math\_numeric type

- using, 5-6

modules, 6-9

## P

---

platforms, 1-3

precision for operations, 5-8

preinstallation

- tasks, 2-2

procedure

- creating an implemented, 3-2

## R

---

records

- querying lists, retrieving lists, 7-2

runtime, 5-2

- apartment threading business functions, 5-3

- connection pooling, 5-3

## S

---

software requirements, 1-2