

## **JD Edwards EnterpriseOne Tools**

BI Publisher for JD Edwards EnterpriseOne Guide

Release 8.98 Update 4

**E14692-05**

December 2012

JD Edwards EnterpriseOne Tools BI Publisher for JD Edwards EnterpriseOne Guide, Release 8.98 Update 4  
E14692-05

Copyright © 2012, Oracle and/or its affiliates. All rights reserved.

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information on content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services.

---

---

# Contents

<b>Preface</b> .....	vii
Audience .....	vii
Documentation Accessibility .....	vii
Related Documents .....	vii
Conventions .....	viii
<b>1 Introduction to BI Publisher for JD Edwards EnterpriseOne</b>	
1.1 BI Publisher for JD Edwards EnterpriseOne Overview .....	1-1
1.1.1 Creating Embedded BI Publisher Reports .....	1-1
1.1.2 Creating Interactive BI Publisher Reports.....	1-1
1.2 BI Publisher for JD Edwards EnterpriseOne Implementation .....	1-2
<b>2 Understanding BI Publisher for JD Edwards EnterpriseOne</b>	
2.1 Introduction to BI Publisher for JD Edwards EnterpriseOne.....	2-1
2.1.1 Benefits of BI Publisher for JD Edwards EnterpriseOne.....	2-2
2.2 Embedded and Interactive BI Publisher Reporting .....	2-3
2.2.1 Embedded BI Publisher Reporting .....	2-3
2.2.2 Interactive BI Publisher Reporting .....	2-3
2.2.3 Embedded and Interactive BI Publisher Reporting Comparison.....	2-4
2.3 JD Edwards EnterpriseOne Embedded BI Publisher Process Overview.....	2-5
2.4 JD Edwards EnterpriseOne Interactive BI Publisher Report Process Overview .....	2-5
<b>3 Creating JD Edwards EnterpriseOne Embedded BI Publisher Objects</b>	
3.1 Understanding the BI Publisher Object Repository.....	3-1
3.1.1 Templates.....	3-1
3.1.2 Transformations.....	3-2
3.1.3 Translations and Localizations .....	3-2
3.2 Uploading, Updating, and Deleting JD Edwards BI Publisher Objects.....	3-3
3.2.1 Understanding BI Publisher Objects.....	3-3
3.2.1.1 Effective Dates.....	3-4
3.2.2 Updating BI Publisher Objects.....	3-5
3.2.3 Deleting BI Publisher Objects .....	3-5
3.2.4 Prerequisite .....	3-5
3.2.5 Forms Used to Upload, Update and Delete BI Publisher Objects .....	3-6
3.2.6 Creating BI Publisher Objects .....	3-6

3.2.7	Uploading BI Publisher Objects.....	3-7
3.2.8	Adding Translations and Localizations .....	3-8
3.2.9	Modifying BI Publisher Object Properties .....	3-8
3.2.10	Modifying Effective Dates.....	3-9
3.2.11	Modifying BI Publisher Objects.....	3-9
3.2.11.1	Revising BI Publisher Objects.....	3-9

## 4 Creating JD Edwards EnterpriseOne BI Publisher Report Definitions

4.1	Understanding JD Edwards EnterpriseOne BI Publisher Report Definitions .....	4-1
4.2	Creating BI Publisher Report Definitions.....	4-2
4.2.1	Understanding the Report Definition Process.....	4-2
4.2.2	Forms Used to Create BI Publisher Report Definitions .....	4-2
4.2.3	Defining Source of Data, Template Names, and Transformations.....	4-3
4.2.4	Defining Languages and Output Types.....	4-4
4.2.5	Defining Bursting and Delivery Options .....	4-6
4.2.6	Delivery Email Subject and Body.....	4-8
4.2.6.1	Report Definition Data Item for Subject and Body.....	4-10
4.2.6.2	Defining a Data Structure.....	4-12
4.2.6.3	Defining a Glossary Data Item .....	4-12
4.3	Modifying BI Publisher Report Definitions .....	4-13
4.4	Copying BI Publisher Report Definitions.....	4-14
4.5	Managing BI Publisher Objects in Object Management Workbench.....	4-14
4.5.1	Searching for BI Publisher Objects .....	4-15
4.6	Configuring BI Publisher Objects for Object Management Workbench.....	4-16

## 5 Submitting JD Edwards EnterpriseOne Report Definitions to BI Publisher

5.1	Understanding the Report Definition Submission Process .....	5-1
5.2	Submitting Report Definitions from BI Publisher Report Definitions (P95620).....	5-1
5.2.1	Submitting from BI Publisher Report Definitions .....	5-1
5.3	Submitting Report Definitions from Batch Versions (P95305).....	5-1
5.3.1	Associating Report Definitions with Batch Versions .....	5-2
5.3.2	Forms Used to Submit Report Definitions from Batch Versions.....	5-2
5.3.3	Submitting Batch Versions .....	5-3
5.3.4	Accessing Batch Version Advanced Option Overrides .....	5-3
5.4	Reviewing Report Definition Submission Details.....	5-5
5.4.1	Understanding Report Definition Jobs.....	5-5
5.4.2	Forms Used to View Report Definition Submission Details .....	5-6
5.4.3	Viewing Submission Details from Report Definition Jobs .....	5-6
5.4.4	Viewing Submission Details from Work With Submitted Jobs (P986110B).....	5-7

## 6 Managing JD Edwards EnterpriseOne Report Definition Output

6.1	Understanding Report Definition Output.....	6-1
6.2	Viewing Report Definition Output, Source, and Delivery Details .....	6-1
6.2.1	Example of Report Definition Output .....	6-2
6.2.2	Forms Used to View Report Definition Output .....	6-2
6.2.3	Viewing Report Definition Source .....	6-3

6.2.3.1	Example: Report Definition Source.....	6-3
6.2.4	Viewing Report Definition Output Details.....	6-4
6.2.5	Viewing Report Definition Output.....	6-8
6.2.6	Viewing Report Definition Output Delivery Details.....	6-8
6.2.7	Printing Report Definition Output.....	6-10
6.3	Setting Up Security for Report Definition Jobs and Output.....	6-10
6.3.1	Securing Report Definition Jobs.....	6-10
6.3.2	Securing Report Definition Output.....	6-10
6.4	Republishing Report Definition Output.....	6-11
6.5	Redelivering Report Definition Output.....	6-11
6.6	Archiving BI Publisher Report Output.....	6-12
6.6.1	Configuring the Enterprise Server jde.ini File.....	6-12
6.6.1.1	[UBE].....	6-12
6.6.1.2	Configuring UBE Settings with Server Manager.....	6-12
6.6.2	BI Publisher Archived Report Files.....	6-13
6.7	Deleting UBE and Report Definition Output.....	6-13
6.7.1	Deleting Report Definition Jobs.....	6-13
6.7.2	Deleting UBE Records.....	6-14
6.7.3	Purging Report Definition Output.....	6-14
6.7.4	Forms Used to Submit Job Master Deletion by Days Old Report.....	6-14
6.7.5	Setting Processing Options for Job Master Deletion by Days Old Report (R9861101).....	6-15
6.7.5.1	Defaults.....	6-15
6.7.5.2	Versions.....	6-15

## 7 Understanding the XML Output

7.1	Designing Batch Applications for BI Publisher.....	7-1
7.1.1	Page Headers and Page Footers.....	7-1
7.2	Interpreting XML Output.....	7-1
7.2.1	Section Tags.....	7-2
7.2.2	Item Tags.....	7-2
7.2.3	Special Characters.....	7-3
7.2.4	Formatted Numbers.....	7-3
7.2.5	Formatted Dates.....	7-3
7.2.6	Page Headers and Page Footers.....	7-4
7.2.7	Conditional Sections.....	7-4
7.2.8	Level Break Sections.....	7-5
7.2.9	Child Sections.....	7-5
7.2.10	Report Properties.....	7-5
7.3	Comparing XML Output Formats.....	7-6
7.3.1	Supported Object Types.....	7-6
7.3.2	General XML Format Differences.....	7-7
7.3.3	Properties Node.....	7-7
7.3.4	Column Headings Node.....	7-7
7.3.5	Page Headers Node.....	7-7
7.3.6	Report Details Node.....	7-7

## 8 Configuring Oracle BI Publisher Enterprise for Interactive Reporting

8.1	Implementing Interactive Reporting with Oracle BI Publisher Enterprise Edition 11g...	8-1
8.1.1	Manually Installing a JD Edwards EnterpriseOne Data Access Driver .....	8-2
8.1.2	Replace the Delivered xerces.jar and xalan.jar Files .....	8-2
8.1.3	Update the Oracle WebLogic PRE_CLASSPATH .....	8-3
8.2	Registering the JD Edwards EnterpriseOne Data Access Driver.....	8-3
8.3	Configuring the JD Edwards EnterpriseOne Data Access Driver .....	8-4
8.3.1	Understanding JD Edwards EnterpriseOne Data Access .....	8-5
8.3.1.1	JDBC Connection String .....	8-5
8.3.1.2	Oracle BI Publisher and JD Edwards EnterpriseOne Security .....	8-6
8.3.2	Configuring the JDBC Driver in Oracle Business Intelligence Publisher Enterprise	8-7
8.3.3	Testing the JDBC Driver Connection .....	8-9

## A JD Edwards EnterpriseOne BI Publisher Releases

A.1	JD Edwards XML Publisher 8.96 Templates .....	A-1
A.2	Migrating from 8.96 XML Publisher .....	A-2

## Glossary

## Index

---

---

# Preface

Welcome to the JD Edwards EnterpriseOne Tools BI Publisher for JD Edwards EnterpriseOne Guide.

## Audience

This guide is intended for reporting and analytics administrators and end users who are responsible for producing reports.

## Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc>.

### Access to Oracle Support

Oracle customers have access to electronic support through My Oracle Support. For information, visit <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info> or visit <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs> if you are hearing impaired.

## Related Documents

You can access related documents from the JD Edwards EnterpriseOne Release Documentation Overview pages on My Oracle Support. Access the main documentation overview page by searching for the document ID, which is 876932.1, or by using this link:

<https://support.oracle.com/CSP/main/article?cmd=show&type=NOT&id=876932.1>

To navigate to this page from the My Oracle Support home page, click the Knowledge tab, and then click the Tools and Training menu, JD Edwards EnterpriseOne, Welcome Center, Release Information Overview.

This guide contains references to server configuration settings that JD Edwards EnterpriseOne stores in configuration files (such as jde.ini, jas.ini, jdbj.ini, jdelog.properties, and so on). Beginning with the JD Edwards EnterpriseOne Tools Release 8.97, it is highly recommended that you only access and manage these settings for the supported server types using the Server Manager program. See the Server Manager Guide on My Oracle Support.

## Conventions

The following text conventions are used in this document:

<b>Convention</b>	<b>Meaning</b>
<b>Bold</b>	Indicates field values.
<i>Italics</i>	Indicates emphasis and JD Edwards EnterpriseOne or other book-length publication titles.
Monospace	Indicates a JD Edwards EnterpriseOne program, other code example, or URL.

---

# Introduction to BI Publisher for JD Edwards EnterpriseOne

This chapter contains the following topics:

- [Section 1.1, "BI Publisher for JD Edwards EnterpriseOne Overview"](#)
- [Section 1.2, "BI Publisher for JD Edwards EnterpriseOne Implementation"](#)

## 1.1 BI Publisher for JD Edwards EnterpriseOne Overview

This guide describes the tools and procedures for using Oracle Business Intelligence (BI) Publisher and common desktop tools to create customized reports for JD Edwards EnterpriseOne. Users create report templates with Microsoft Word, Microsoft Excel, and Adobe Acrobat that enable application data to be reused and reformatted into various layouts and output types. Depending on the template type, the available output types are PDF, Rich Text Format (RTF), Excel, XML, HTML, and eText.

JD Edwards EnterpriseOne provides two BI Publisher reporting options:

- **Embedded BI Publisher reporting.**  
Embedded BI Publisher reports are created, managed, and processed using the JD Edwards EnterpriseOne toolset. Embedded BI Publisher reports use data that is generated by JD Edwards EnterpriseOne batch applications.
- **Interactive BI Publisher reporting.**  
Interactive BI Publisher reports are created, managed, and processed using the Oracle BI Publisher Enterprise toolset. Interactive reports access JD Edwards EnterpriseOne data using the JD Edwards EnterpriseOne Data Access Driver.

### 1.1.1 Creating Embedded BI Publisher Reports

Chapters 3 through 7 in this guide describe how to use the embedded BI Publisher to create objects and generate data to produce and manage customized JD Edwards EnterpriseOne reports.

### 1.1.2 Creating Interactive BI Publisher Reports

Chapter 8 describes the steps to configure JD Edwards EnterpriseOne for interactive reporting with Oracle BI Publisher Enterprise.

## 1.2 BI Publisher for JD Edwards EnterpriseOne Implementation

This section provides an overview of the steps that are required to implement BI Publisher for JD Edwards EnterpriseOne.

In the planning phase of the implementation, take advantage of all JD Edwards sources of information, including the installation guides and troubleshooting information.

Follow these steps to implement the embedded BI Publisher for JD Edwards EnterpriseOne:

1. Install JD Edwards EnterpriseOne.  
*See JD Edwards EnterpriseOne Server Manager Guide and JD Edwards Reference Guide.*
2. Install one or more JD Edwards EnterpriseOne applications.  
*See JD Edwards EnterpriseOne installation documentation for your applications.*
3. Install Oracle's BI Publisher Desktop Components.  
*See Oracle Business Intelligence Publisher User's Guide.*

---

# Understanding BI Publisher for JD Edwards EnterpriseOne

This chapter contains the following topics:

- [Section 2.1, "Introduction to BI Publisher for JD Edwards EnterpriseOne"](#)
- [Section 2.2, "Embedded and Interactive BI Publisher Reporting"](#)
- [Section 2.3, "JD Edwards EnterpriseOne Embedded BI Publisher Process Overview"](#)
- [Section 2.4, "JD Edwards EnterpriseOne Interactive BI Publisher Report Process Overview"](#)

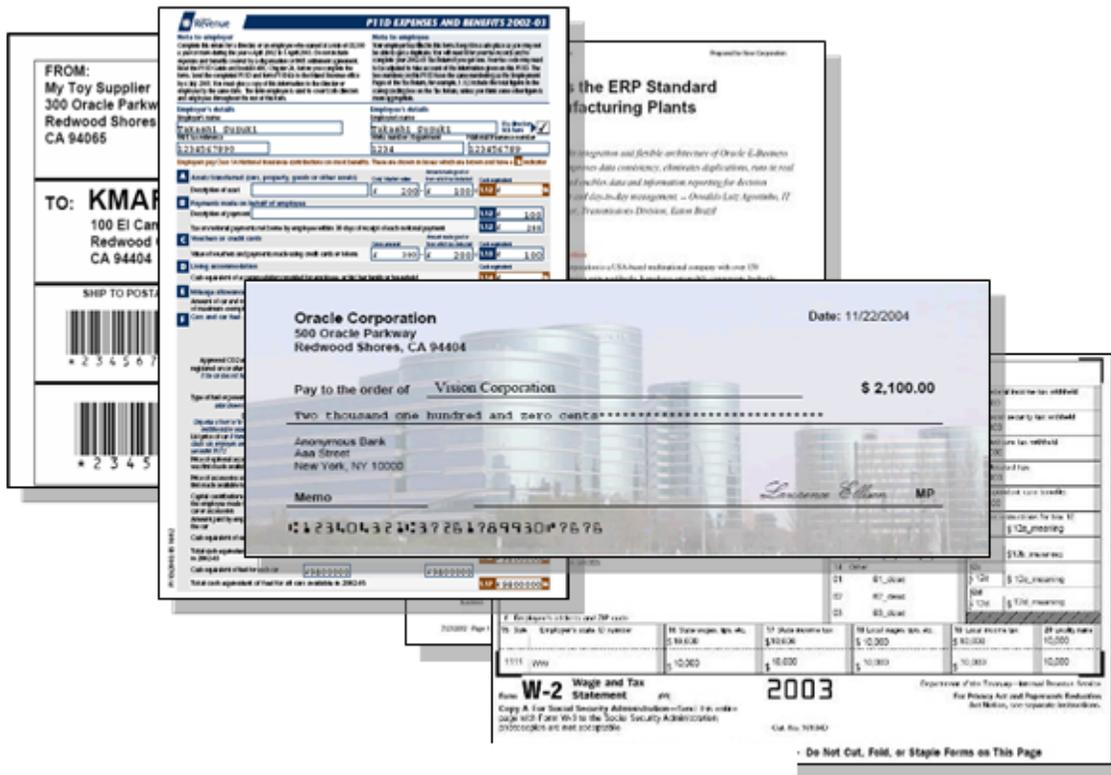
## 2.1 Introduction to BI Publisher for JD Edwards EnterpriseOne

JD Edwards EnterpriseOne has provided a reporting solution based on Oracle's BI Publisher since JD Edwards EnterpriseOne 8.96 Tools release. The solution was extended with JD Edwards EnterpriseOne 8.97 Tools release by integrating the embedded BI Publisher reports into JD Edwards EnterpriseOne life cycle management. The solution has been further extended in JD Edwards EnterpriseOne 8.98 Tools release to support interactive reports. Interactive reports are created, managed and processed in the Oracle BI Publisher Enterprise toolset. The embedded reports continue to be created, managed and processed using the JD Edwards EnterpriseOne toolset.

BI Publisher reduces the high cost of producing, customizing, and maintaining business documents. Built on the open XSLFO (Extended standard, it provides a full range of reporting capabilities, including charts, formulas, watermarks, and graphics. Using a set of familiar desktop tools, users can create and maintain their own report formats based on data extracts from JD Edwards EnterpriseOne reports.

The following example shows sample documents generated by BI Publisher:

Figure 2–1 BI Publisher sample documents



The flexibility of BI Publisher is a result of the separation of the report layout from the data. The data collection is still handled by JD Edwards EnterpriseOne; however, you can design and control how the report outputs are presented using template files. At runtime, BI Publisher merges your designed template files with the report data to create a variety of outputs to meet various business needs, including:

- Customer-ready PDF documents, such as financial statements, marketing materials, contracts, invoices, and purchase orders using colors, images, font styles, headers and footers, and many other formatting and design options.
- HTML output for optimum online viewing.
- Excel output to create a spreadsheet of your report data.
- Third-party provided PDF documents. You can download a PDF document (such as a government form) to use as a template for your report. At runtime, the data and template produce a completed form.
- Flat text files to exchange with business partners for electronic data interchange (EDI) and electronic file transfer (EFT) transmissions.

You can create batch applications using JD Edwards EnterpriseOne Report Design Aid to produce XML output that is compatible with BI Publisher.

### 2.1.1 Benefits of BI Publisher for JD Edwards EnterpriseOne

BI Publisher for JD Edwards EnterpriseOne enables you to:

- Create report layouts using familiar desktop applications.

- Format batch application output.
- Automatically distribute reports email.
- Leverage a single batch report to generate multiple different end-user reports.
- Define and manage report definitions.
- Burst a report to a variety of recipients based on data-driven metrics or specified distribution lists.

## 2.2 Embedded and Interactive BI Publisher Reporting

Embedded BI Publisher reports are created and managed using the JD Edwards EnterpriseOne toolset. In addition to the embedded reports, you have the option to create interactive BI Publisher reports using the Oracle BI Publisher Enterprise toolset.

### 2.2.1 Embedded BI Publisher Reporting

Embedded BI Publisher reporting is driven by a JD Edwards EnterpriseOne batch application. Use embedded BI Publisher reporting under these circumstances:

- An event in JD Edwards EnterpriseOne initiates the requirement for the report.  
For example, the creation of a new sales order could automatically launch the Pick Slip process. The Pick Slip process would be configured to produce a BI Publisher document that is shipped with the order. The document would be produced with no user interaction.
- The report needs to update, insert, or delete JD Edwards EnterpriseOne data.  
Examples of report processes that update JD Edwards EnterpriseOne data are: updating the product shipped date, advancing the sales order status, and updating the statement print date.
- The report needs to perform JD Edwards EnterpriseOne business logic.  
Examples of business logic include: determining the correct pricing for a product, computing an employee's salary or benefits, and calculating commissions for sales staff.

### 2.2.2 Interactive BI Publisher Reporting

Interactive BI Publisher reporting should be used under the following circumstances:

- The requested report is not part of a standard JD Edwards EnterpriseOne business process, or the requirements for the report change over time. The following are examples of these types of reports:
  - The CIO requests a list of current open Account Receivables.
  - A manager wants to review year-to-date budget amounts compared to actuals.
  - A manager requests a report showing current inventory levels.
  - Requests for other typical ad-hoc reports.
- The report only reads JD Edwards EnterpriseOne data.  
The following are examples of these types of reports:
  - A summary of activity for a particular customer.
  - The current General Ledger balances for a subset of accounts.

- A list of travel expenditures for a department.
- A list of active suppliers.

### 2.2.3 Embedded and Interactive BI Publisher Reporting Comparison

The following table summarizes the recommendations for using the embedded and interactive reporting capabilities of BI Publisher with JD Edwards EnterpriseOne.

Scenario	Embedded	Interactive Reporting
An event within JD Edwards EnterpriseOne triggers the report: <ul style="list-style-type: none"> <li>■ Pick Slips</li> <li>■ Invoice Print</li> <li>■ AP Checks</li> <li>■ W2</li> </ul>	Yes	No
A non-JD Edwards EnterpriseOne event triggers the report: <ul style="list-style-type: none"> <li>■ Open Account Receivables</li> <li>■ Budget to Actual</li> <li>■ Inventory levels</li> <li>■ Ad-hoc reports</li> </ul>	No	Yes
An update to the JD Edwards EnterpriseOne data is performed: <ul style="list-style-type: none"> <li>■ Update status codes</li> <li>■ Modify dates</li> <li>■ Adjust inventory quantities</li> <li>■ Create history records</li> </ul>	Yes	No
The report only reads JD Edwards EnterpriseOne data: <ul style="list-style-type: none"> <li>■ Customer activity</li> <li>■ General Ledger accounts</li> <li>■ Travel expenditures</li> <li>■ Summary of suppliers</li> </ul>	Yes	Yes
The report needs to perform business logic: <ul style="list-style-type: none"> <li>■ Calculate taxes</li> <li>■ Schedule delivery dates</li> <li>■ Calculate salaries and benefits calculations</li> <li>■ Calculate sales commission calculations</li> </ul>	Yes	No

## 2.3 JD Edwards EnterpriseOne Embedded BI Publisher Process Overview

Creating customized reports using the embedded BI Publisher involves these steps:

1. Use Oracle's BI Publisher Template Builder, Microsoft Excel, Microsoft Word, or Adobe Acrobat to create a blank template file.
2. Add an object to the BI Publisher Object Repository (P95600), using the template name for the object name.
3. Create a report definition in the JD Edwards BI Publisher Report Definition application (P95620), associating the source batch application (UBE) and the template.
4. Submit the report definition to generate an XML data file.
5. Download the XML data and design the template in the desktop application that you used to create the template.
6. Update the report definition in the BI Publisher Report Definitions application, defining criteria such as output types, bursting, and delivery options.
7. Upload the completed template into BI Publisher Object Repository.
8. Submit the report definition to BI Publisher.
9. Use Report Definition Jobs (P95630) to access the submission details, view details of the output, view the output delivery, and view the output.

## 2.4 JD Edwards EnterpriseOne Interactive BI Publisher Report Process Overview

To create interactive BI Publisher reports, you must first install and configure the JD Edwards Data Access Driver and Oracle Business Intelligence Publisher Enterprise.

After Oracle Business Intelligence Publisher Enterprise has been configured successfully, creating interactive reports involves these steps:

1. Use Oracle's BI Publisher Template Builder, Microsoft Excel, Microsoft Word, or Adobe Acrobat to create a blank template file.
2. Create a report in the Oracle BI Publisher Enterprise Console.
3. Create a SQL statement for the report.
4. Execute the query to create the report.

This guide contains additional information about installing and configuring the JD Edwards EnterpriseOne Data Access Driver and Oracle Business Intelligence Publisher Enterprise.

See [Chapter 8, "Configuring Oracle BI Publisher Enterprise for Interactive Reporting"](#).

The *Oracle Business Intelligence Publisher Report Designer's Guide 10.1.3.4* provides information about creating reports with Oracle Business Intelligence Publisher Enterprise.



---



---

## Creating JD Edwards EnterpriseOne Embedded BI Publisher Objects

This chapter contains the following topics:

- [Section 3.1, "Understanding the BI Publisher Object Repository"](#)
- [Section 3.2, "Uploading, Updating, and Deleting JD Edwards BI Publisher Objects"](#)

### 3.1 Understanding the BI Publisher Object Repository

The BI Publisher Object Repository (P95600) users to create, upload, update, copy, and delete BI Publisher objects. BI Publisher uses three types of objects:

- Templates
- Transformations
- Translations and localizations

#### 3.1.1 Templates

BI Publisher templates are used to format reports using the data that resides in the JD Edwards EnterpriseOne database. You can create templates using:

- Microsoft Word (RTF format only)
- Microsoft Excel (XLS or XSL)
- Adobe Acrobat (PDF)

This table describes how to create templates.

Template Type	Method
Microsoft Word	Design templates manually using Microsoft Word or Oracle's BI Publisher Template Builder to facilitate template development.
Microsoft Excel	Design templates manually using Microsoft Excel.
Adobe Acrobat	Design templates manually using the full version of Adobe Acrobat.

Oracle's BI Publisher Template Builder is an extension to Microsoft Word that simplifies the development of RTF templates. It automates many of the manual steps that you would otherwise have to perform. Use the Template Builder in conjunction with the *Oracle Business Intelligence Publisher User's Guide* to increase your productivity.

---

---

**Note:** The BI Publisher Template Builder for Word Add-in includes features to log in to and interact with Oracle Business Intelligence Publisher Enterprise. The following menu items work only with the Oracle BI Publisher Enterprise or Oracle BI Enterprise Edition implementations: Log In, Open, Upload, and Upload Template As. See the Template Builder help for more information.

---

---

The Template Builder is tightly integrated with Microsoft Word and enables you to perform the following functions. Instructions and tutorials for using the Template Builder are available from the readme and help files delivered with the tool.

- Insert data fields.
- Insert data-driven tables.
- Insert data-driven forms.
- Insert data-driven charts.
- Preview your templates with sample XML data.
- Browse and update the content of form fields.
- Extract boilerplate text into an XLIFF translation file and test translations.

Manual steps for performing these functions are covered in the *Oracle Business Intelligence Publisher User's Guide*. Instructions and tutorials for using the Template Builder are available from the readme and help files delivered with the tool.

See *Oracle Business Intelligence Publisher User's Guide*.

Excel templates must be created manually using Microsoft Excel. To create PDF templates, you must have the full version of Adobe Acrobat.

---

---

**Note:** For more information about creating templates specifically for JD Edwards EnterpriseOne, see "Creating XML Publisher Templates for JD Edwards EnterpriseOne" available on My Oracle Support.

---

---

### 3.1.2 Transformations

BI Publisher transformations are eXtensible Stylesheet Language (XSL) templates that are used to map data and transform an XML file into another XML file with a different format. This is beneficial when, for example, you use a template that was created using different field names and you need to map those field names to JD Edwards EnterpriseOne-specific field names.

### 3.1.3 Translations and Localizations

BI Publisher for JD Edwards EnterpriseOne introduces the concepts of localization and translation. Translations are XML files that are created to translate static text of the template into other languages. You can think of translations as a list of words on a template that are translated from one language, typically English, to another language at runtime. Translations are available for RTF templates only.

This is an example of an XML file that translates the static text of an English RTF template to German.

**Figure 3–1 German translation of English template**

```

- <xliiff version="1.0">
- <file source-language="EN-US" target-language="DE-DE" datatype="XDO" original="orphen.rtf" product-version="orphen.xlf" product-
  name="">
- <header>
- <prop-group name="ora_reconstruction">
  <prop prop-type="TemplateCode">n/a</prop>
  <prop prop-type="extractorVersion">5.6.1_115.16</prop>
</prop-group>
</header>
- <body>
- <trans-unit id="ab2120e2" maxbytes="4000" maxwidth="70" size-unit="char" translate="yes">
  <source>THANK YOU FOR YOUR BUSINESS!</source>
  <target>DANKE FÜR IHR GESCHÄFT!</target>
  <note>Text located: body</note>
</trans-unit>
- <trans-unit id="a115ca57" maxbytes="4000" maxwidth="22" size-unit="char" translate="yes">
  <source>SALES TAX</source>
  <target>VERKAUFSSTEUER</target>
  <note>Text located: body/table</note>
</trans-unit>
- <trans-unit id="643b3f93" maxbytes="4000" maxwidth="37" size-unit="char" translate="yes">
  <source>SHIPPING METHOD</source>
  <target>VERSCHIFFEN-METHODE</target>
  <note>Text located: body/table</note>

```

By contrast, a localization is a different version of an existing template that not only translates the static text from one language to another, but also includes or excludes specific data elements when compared to the base template. For example, a base template might have a data element that displays salary information. If displaying this information is illegal in a particular country, you can copy the base template and remove the salary data element.

## 3.2 Uploading, Updating, and Deleting JD Edwards BI Publisher Objects

This section provides overviews of uploading BI Publisher objects, updating BI Publisher objects, and deleting BI Publisher objects and discusses how to:

- Upload BI Publisher objects.
- Add translations and localizations.
- Modify BI Publisher object properties.
- Modify effective dates.
- Modify BI Publisher objects.
- Delete BI Publisher objects.

### 3.2.1 Understanding BI Publisher Objects

You must upload BI Publisher objects to JD Edwards EnterpriseOne before you can create report definitions for the desired output types. You use BI Publisher Object Repository (P95600) on either the web client or the Microsoft Windows client to upload templates, translations, localizations, and transformations into JD Edwards EnterpriseOne.

To upload BI Publisher objects, you must add a new BI Publisher object to JD Edwards EnterpriseOne. Name the object according to these recommended naming conventions:

- Use all capital letters.
- Do not begin the object name with a dash (-).
- Use a maximum of 50 characters.

Enter a brief but meaningful description of the object. This description should include the purpose of the object.

Identify the BI Publisher object as a specific object type. The following object types are available:

- eText.
- PDF.
- Rich text format (RTF).
- Extensible stylesheet language (XSL).
- Transformation.
- Translation.

After creating a new object in JD Edwards EnterpriseOne, you can search for and select the BI Publisher object that you need to upload. BI Publisher objects are stored in the JD Edwards EnterpriseOne database in the F95600 table.

The file types that you can upload for each BI Publisher object type are shown in this table:

BI Publisher Object	BI Publisher Object Type	File Type
Templates	TE	.rtf
-	TL	.xls
-	TP	.pdf
-	TR	.rtf
-	TS	.xsl
Translations	XL	.xml or .xlf
Transformations	XF	.xsl

The recommended maximum default BI Publisher object size is 50MB. You can modify the BI Publisher object size on the Windows client by changing the object size in the INTERACTIVE RUNTIME section of the jde.ini using the following definition:

```
[INTERACTIVE RUNTIME]
MaxFileUploadSize = 50
```

Oracle recommends that you use the Windows client to upload BI Publisher objects. However, if the web client is used to upload objects, you can modify the BI Publisher object size in the OWWEB section of the jas.ini using the following definition:

```
[OWWEB]
MaxFileUploadSize = 50
```

---

**Note:** You can increase or decrease the default maximum file upload size. The default maximum size is 50MB. If the BI Publisher object exceeds the maximum size definition, an error appears.

---

### 3.2.1.1 Effective Dates

Effective dates indicate when an object is available for use. If an object will be used indefinitely, you do not need to specify an end date. You can upload additional

templates with the same name and give them specific start and end dates as long as the effective dates do not overlap. The version without an end date becomes the default version when none of the other versions' effective dates correspond to the current date.

Effective dates enable a company to have multiple, slightly different versions of the same template. For example, to customize a newsletter with different logos for each month, you can design a master template and upload it to the BI Publisher Object Repository with a start date, but no end date. Then you can create twelve copies of the original template, add a monthly logo to each one, and upload them with the same name as the original template but with different start and end dates for each copy.

### 3.2.2 Updating BI Publisher Objects

You can update BI Publisher objects that have been uploaded to JD Edwards EnterpriseOne. First, download the BI Publisher object to your local machine. Then make the required modifications to the BI Publisher object. Upload the modified BI Publisher object to JD Edwards EnterpriseOne.

### 3.2.3 Deleting BI Publisher Objects

You can delete BI Publisher objects that have been uploaded to JD Edwards EnterpriseOne. From the BI Publisher Object Repository application, select the BI Publisher object. Click Delete.

When you select to delete an object, the system displays a Delete BI Publisher Object Warning form. Click OK to delete the object or click Cancel if you do not want to delete the object.

If the object is active when you attempt to delete it, the system displays a Delete Object Violation form. Click OK to close the form and return to the BI Publisher Object Repository. Select the object again and access Effective Dates from the row menu to remove the start and end dates. After removing the effective dates, select the object and click Delete to remove the reporting object.

When you delete a BI Publisher reporting object, the object record is deleted from the F95600 table. In addition, all associations to the deleted reporting object are also deleted. Associations reside in these tables:

- BI Publisher UBE Associations (F95610)
- BI Publisher Object Associations (F95611)

### 3.2.4 Prerequisite

Refer to the Getting Started chapter for a list of implementation steps that you need to perform.

See [Chapter 1, "Introduction to BI Publisher for JD Edwards EnterpriseOne"](#).

### 3.2.5 Forms Used to Upload, Update and Delete BI Publisher Objects

Form Name	FormID	Navigation	Usage
BI Publisher Object Repository	W95600E	On the EnterpriseOne Menu, select EnterpriseOne Menus, EnterpriseOne Life Cycle Tools, Report Management, BI Publisher, BI Publisher Repository.	Select a BI Publisher object.
Add BI Publisher Object to Repository	W95600B	Click Add on the BI Publisher Object Repository form.	Create BI Publisher objects.
Update BI Publisher Object in Repository	W95600B	Select a record on the BI Publisher Object Repository form, and click Select.	Modify object description, product code, or product system code for existing BI Publisher objects.
Update BI Publisher Object Effective Dates	W95600B	When uploading objects, the form appears after you click Load on the File Upload form. To update effective dates, select a record on the BI Publisher Object Repository form, and select Effective Dates on the row menu.	Modify or delete effective dates for existing BI objects.

### 3.2.6 Creating BI Publisher Objects

Access the Add BI Publisher Object to Repository form.

Figure 3–2 Add BI Publisher Object to Repository form

The screenshot shows a software window titled "Add BI Publisher Object to Repository". At the top, there are buttons for "OK", "Cancel", and "Tools". Below the title bar, the form contains several input fields and labels:

- Object Name: ADDRESS BOOK
- Object Description: Addresses
- Object Type: TR (with a dropdown arrow) and *RTF Template*
- Product Code: 55 (with a dropdown arrow) and *Reserved for Clients*
- Product System Code: 01 (with a dropdown arrow) and *Address Book*
- Object Region: US (with a dropdown arrow) and *United States*
- Object Language: EN (with a dropdown arrow) and *English*
- Localization:  Localization

Below these fields is a section titled "Effective Dates" containing two date input fields: "Start Date" and "End Date".

**Object Name**

Enter the name of the BI Publisher object.

**Object Description**

Enter a meaningful description of the object.

**Object Type**

Select a user-defined code (UDC) (H95 | XP) that indicates the object type of the reporting object.

**Product Code**

Select a product code from the 55–59 client reserved range.

**Product Code Reporting**

Select a product code that indicates where the reporting data resides.

**Object Region**

For templates, select a UDC (H95 | XC) that indicates the region for which the template was created. For translations, select the region in which the report will be presented. This field is not required if the object type is a transformation.

**Object Language**

For templates, select the language that is used in the template. For translations, select the language in which the report will be presented. This field is not required if the object type is a transformation.

**Localization**

Select this option if the object is a localization of an existing template. The localization object must have the same name and document type as the template upon which it is based.

**Start Date**

After clicking OK and uploading the object, add a start date to indicate when the object is available for use. If an object does not have a start date or if the start date is in the future, the object has an Inactive status.

**End Date**

Do not enter an end date if the object is to be used indefinitely. Enter an end date only if the object is to be used for a specific period of time.

---

---

**Note:** The Localization check box is available only when you are adding a localization object with the same name as an uploaded template. The Start Date, and End Date fields are not available until an object has been uploaded into the repository.

---

---

### 3.2.7 Uploading BI Publisher Objects

Click OK on the Add BI Publisher Object to Repository form to access the File Upload form.

**Figure 3–3 File Upload form**



1. Click Browse to locate the appropriate BI Publisher object.
2. Validate the object type and file extension.

---

**Note:** The system does not allow you to upload a file with an invalid extension.

---

3. On the File Upload form, click Load to upload the BI Publisher object from your local directory to JD Edwards EnterpriseOne.

### 3.2.8 Adding Translations and Localizations

Access the BI Publisher Object Repository form.

1. Find and select the template to which you want to add a translation or localized template.
2. From the Row menu, select either Add Translation or Add Localization.
3. On either the Add BI Publisher Translation to Repository form or the Add BI Publisher Localized Template to Repository form, complete the following fields and click OK:
  - Description
  - Product Code
  - Product System Code
  - Object Region
  - Object Language
4. On the File Upload form, locate the translation or localized template and click Load to upload the file from your local directory to JD Edwards EnterpriseOne.

---

**Note:** You must upload the correct file type or an error message will appear. Translations must be either .xml or .xlf files. Localizations must be .rtf files, the same as the templates on which they are based.

---

5. Enter a start date for the template if you want to make the template active, and an end date if the template is to be used for only a specific period of time.

### 3.2.9 Modifying BI Publisher Object Properties

Access the Update BI Publisher Object in Repository form.

1. Modify object information and click OK.
2. On the Upload File form, click No in answer to the question "Would you like to upload the file again?"

3. Click Close.

### 3.2.10 Modifying Effective Dates

Access the Update BI Publisher Object Effective Dates form.

1. Add, modify, or delete the start date.
2. Add, modify, or delete the end date.
3. Click OK.

### 3.2.11 Modifying BI Publisher Objects

Access the BI Publisher Object Repository form.

1. Select a BI Publisher object.
2. From the Row menu, select Download.
3. On the Microsoft Windows client, the file downloads automatically.
4. On the web client, select either Open or Save on the File Download form.

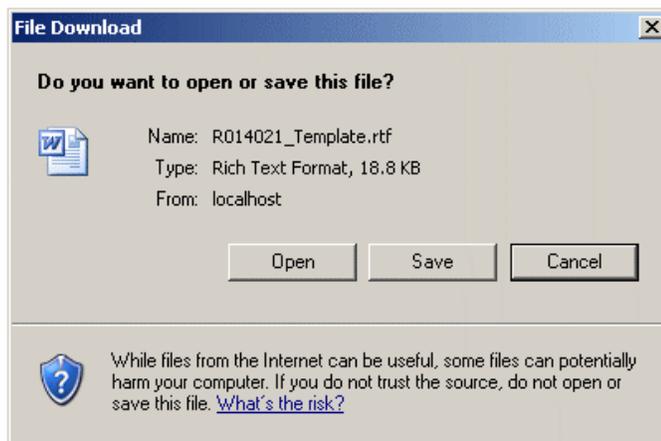
**Figure 3–4** File Download form on the web client

#### File Download

If file download does not start automatically, then [click here](#)

R014021\_Template.rtf

Close




---

**Note:** On the web client, when you select Download from the Row menu of the BI Publisher Object Repository form, the BI Publisher object is cached on your machine. As long as the File Download form remains open, you can continue to open and save the object. When you close the File Download form, the cache is cleared.

---

#### 3.2.11.1 Revising BI Publisher Objects

After the file downloads, you can revise the BI Publisher object.

See *Oracle Business Intelligence Publisher User's Guide*.

1. Modify the BI Publisher object using the appropriate editor, and save the object to your local machine.

2. Close the editor.
3. On the BI Publisher Object Repository form, select the object that you just modified, and click Select.
4. On the Update BI Publisher Object in Repository form, modify the object information and click OK.
5. On the Upload File form, click Yes in response to the question, "Would you like to upload the file again?"
6. Click the BI Publisher object name to upload the modified version.
7. Click Close.

---

---

# Creating JD Edwards EnterpriseOne BI Publisher Report Definitions

This chapter contains the following topics:

- [Section 4.1, "Understanding JD Edwards EnterpriseOne BI Publisher Report Definitions"](#)
- [Section 4.2, "Creating BI Publisher Report Definitions"](#)
- [Section 4.3, "Modifying BI Publisher Report Definitions"](#)
- [Section 4.4, "Copying BI Publisher Report Definitions"](#)
- [Section 4.5, "Managing BI Publisher Objects in Object Management Workbench"](#)
- [Section 4.6, "Configuring BI Publisher Objects for Object Management Workbench"](#)

## 4.1 Understanding JD Edwards EnterpriseOne BI Publisher Report Definitions

Report definitions specify the information that BI Publisher requires to process and deliver BI Publisher output. The details in a report definition include:

- The UBE and optional version to use as the data source.
- Associated templates, transformations, and localizations.
- Output types.
- Language options.
- Bursting options.
- Delivery options.

You use the BI Publisher Report Definitions application (P95620) to create, update, copy, and delete report definitions. After a report definition is created, it can be submitted to BI Publisher to produce the defined output types.

---

---

**Note:** Before creating a report definition, ensure that the necessary templates have been created and uploaded into the BI Publisher Object Repository (P95600).

---

---

See [Chapter 3, "Creating JD Edwards EnterpriseOne Embedded BI Publisher Objects"](#).

## 4.2 Creating BI Publisher Report Definitions

This section provides an overview of the report definition process and discusses how to:

- Define the source of data, template names, and transformations.
- Define languages and output types.
- Define bursting and delivery options.

### 4.2.1 Understanding the Report Definition Process

Although the BI Publisher Report Definitions application (P95620) is available on both the web client and the Microsoft Windows client, you must use the Microsoft Windows client to create a new report definition or to modify an existing report definition.

Adding a report definition is a three-step process. The first step consists of entering the required details for the report definition and defining the source of data and the templates to process. Additionally, if a template uses one or more transformations, they are defined in this step.

Second, you define the output types that a user can select when submitting the report definition. You also define the default output types and default languages for the XML output.

In the third step, you define the optional bursting field for the report definition, as well as the delivery options for the output. A report definition must comply with the following conditions to allow bursting to be configured:

- Templates of type TL (Microsoft Excel) and TS (XSL) cannot be used.
- A transformation cannot be used.
- Microsoft PowerPoint and XML output types cannot be used.

### 4.2.2 Forms Used to Create BI Publisher Report Definitions

Form Name	FormID	Navigation	Usage
BI Publisher Report Definitions	W95620A	On JD Edwards Solution Explorer EnterpriseOne Life Cycle Tools task view, select Report Management, BI Publisher, Report Definitions or enter <b>P95620</b> in the Fast Path field.	Select a BI Publisher report definition.
Add Report Definition	W95620C	Click Add on the BI Publisher Report Definitions form.	Add a report definition.
Add Report Definition-Languages and Output Types	W95620H	Click Next on the Add Report Definition form.	Define available output types, default output types, and default languages.

Form Name	FormID	Navigation	Usage
Add Report Definition-Bursting and Delivery	W95620J	Click Next on the Add Report Definitions-Language s and Output Types form.	Define an optional bursting field for the report definition, and delivery details for the output.
Update Report Definition	W95620C	On the BI Publisher Report Definitions form, select a record in the grid and then click Select.	Update a report definition.

### 4.2.3 Defining Source of Data, Template Names, and Transformations

Access the Add Report Definition form.

Figure 4–1 Add Report Definition form

The screenshot shows the 'Add Report Definition' form with the following fields and values:

- Report Definition: RD000001
- Description: One Line Per Address
- Product Code: 55 (Reserved for Clients)
- Product System Code: 01 (Address Book)
- Source Type: 1 (Batch Version)
- Source UBE: R014021 (One Line Per Address)
- Source Version: (empty)

At the bottom, there is a table with the following data:

Template Name	Template Description	Active Status	Transformation Name	Transformation Description	Active Status
ADDRESS BOOK	Addresses	ACTIVE			

**Report Definition**

Enter the name of the report definition, using a maximum of 10 characters. Oracle recommends that report definition names start with the letters *RD*.

**Description**

Enter a meaningful description for the report definition.

**Product Code**

Select a product code from the 55–59 client reserved range.

**Product System Code**

Select a product code that indicates where the reporting data resides.

**Source Type**

The source type (user-defined code (UDC) H95 | RY) appears automatically and is based on the source type of the data that the report definition uses.

**Source UBE**

Enter the name of the Report Design Aid (RDA) report template that is used to generate data for the report definition.

You can also use a subsystem report, such as R42520/XJDE0006, Print Pick Slips - Subsystem, as the source UBE and version for a report definition. Subsystem reports are batch processes that continually run independently of, but asynchronously with, JD Edwards EnterpriseOne applications. Subsystem reports offer many advantages, such as the elimination of startup time, one-time initialization of environment and specifications, and better utilization of the processor on the server.

For information about subsystem jobs, see "Working with Subsystem Jobs" in the *JD Edwards EnterpriseOne Tools Development Tools: Report Design Aid Guide*.

**Source Version**

(Optional) Enter the name of a version only if the report definition is limited to one version of the UBE. If a version is not specified, the user will have to select a version at submission time.

**Blind Submission**

(Optional) Leave this option blank to prompt the user with the available output, delivery, and language options when submitting the report definition to BI Publisher. If blind submission is selected, the user is not allowed to change the report definition options when submitting the report definition.

**Template Name**

Enter the name of the template that you want to associate with the report definition.

**Template Description**

The template description appears automatically and cannot be changed.

**Active Status**

Active Status is determined by the effective dates of the template. The status appears automatically and indicates whether the template is active or inactive.

**Transformation Name**

(Optional) Enter the name of the transformation that you want to associate with the report description.

**Transformation Description**

The transformation description appears automatically and cannot be changed.

**Active Status**

Active Status is determined by the effective dates of the transformation. The status appears automatically and indicates whether the transformation is active or inactive.

After entering the required report definition details, click Next to move to the Languages and Output Types form.

## 4.2.4 Defining Languages and Output Types

Access the Languages and Output Types form.

Figure 4-2 Report Definition-Languages and Output Types form

Report Definition: RD000001 One Line Per Address

Source Type: 1 Batch Version  Blind Submission

Source UBE: R014021 One Line Per Address

Source Version:

Output Types

Available

- PDF
- RTF
- XML
- HTML
- Excel
- eTEXT
- PowerPoint

Default

- PDF
- RTF
- XML
- HTML
- Excel
- eTEXT
- PowerPoint

Default Languages

\*ALL Active Languages  User Preference Language

Object Language	Language Description
AF	Afrikaans
EN	English
FR	French

**Note:** The templates, translations, and localizations that are attached to the report definition determine the languages that appear in the Object Language column.

**Available Output Types**

Select the output types to be available to users when they are submitting the report definition. The type of template attached to the report definition determines which output types are available.

**Default Output Types**

Select the default output types that will be selected when the report definition is submitted. Default output types are not enabled unless they are first defined as available output types. If the report definition is defined as blind submission, the default values indicate the type of output that will be produced when the report definition is submitted to BI Publisher.

---



---

**Note:** When you run a report definition, the system does not automatically generate a PDF.

---



---

**\*ALL Active Languages**

If this option is selected, all languages for the defined templates, localized templates, and translations that are active will process when a report definition is submitted to BI Publisher. Languages associated with the templates are displayed in the grid.

**User Preference Language**

If this option is selected, when a report definition is submitted to BI Publisher and the sign on user's language preference has been defined, the language preference for the sign on user is processed. A user's language preference is defined in the user's system profile. The User Preference Language option is not enabled if \*ALL Active Languages is selected.

**Object Language**

A code that indicates the language defined for a template, localization, or translation.

**Language Description**

A description of the language code assigned to a template, localization, or translation.

The output types available depend on the type of template that is used, as described in this table:

Template Type	Output Type
RTF	PDF, RTF, Excel, HTML
PDF	PDF
EFT/EDI (eText)	eText
Excel	Excel
XSL	XML

After defining the output types and languages, click Next to move to the Add Report Definition-Bursting and Delivery form.

## 4.2.5 Defining Bursting and Delivery Options

Access the Add Report Definitions-Bursting and Delivery form.

Figure 4-3 Add Report Definitions - Bursting and Delivery form

Data Driven Recipient	Data Recipient Language Preference
AddressNumber_ID1	

**Burst Report**

(Optional) Select this option if bursting will be performed for a report definition.

**Burst Field**

A report definition can burst on any level break section defined within the batch or report version that is used as its source of data. Typically, when you burst a report, you break on a level break header section. The field must contain the complete XPath as it appears in the XML data output that was created from the batch process.

**Printer**

Select this option to direct output to a printer.

**Address Book Number E-Mail**

Select this option to deliver output to the email address that is associated with a specific Address Book number. When a report definition is submitted to BI Publisher, all output produced is sent to the email address.

**Distribution List Type**

Select this option to deliver output to all members of the distribution list type who are associated with the provided Address Book number. If the distribution list type is not entered, the output is sent to the primary email address associated with the Address Book number.

**Consolidate Bursted Report for Address Number**

Select this option to deliver all the bursted output in a single report to the email address associated with the provided Address Book number.

**E-Mail Address**

Select this option to deliver output to a specific email address. When the report definition is submitted to BI Publisher and an email delivery address has been defined, all output produced is sent to the defined email address.

**Language**

Select the language in which you want the output delivered.

**Consolidate Bursted Report for E-Mail Address**

Select this option to deliver all the bursted output in a single report to the email address.

**Data Driven E-Mail**

Select this option to deliver the output to email recipients based on the data generated by the source batch version or report version.

**Distribution List Type**

Select this option to deliver output to all members of a distribution list and distribution list type that are associated with the data. If the distribution list type is not entered, the output is sent to the primary email address associated with the Address Book number.

**Data Item for Subject and Body**

(Optional) Specify a data dictionary glossary item for the subject and body of the delivery email.

**From E-Mail Address Override**

Specify an alternate email address if you want to override the from email address for the delivery email.

**Data Driven Recipient**

(Optional) To deliver the report definition output to recipients as determined by the data from a particular item within the XML output, specify the XPath field that contains the recipient data.

**Data Recipient Language Preference**

(Optional) Select this option if you want to deliver the report definition output to the recipients determined by the language preference within the XML output. The field must contain language data from the 01/LP or the H95/XL UDCs (User Defined Code).

After defining bursting and delivery details, click End to save the report definition and exit.

## 4.2.6 Delivery Email Subject and Body

If report definition output is sent to an email address, you can define a customized subject and body for the email message. To utilize this enhancement, a data dictionary glossary item (glossary group E) must be associated with the report definition.

Glossary group E designates error, warning, and information messages. If a message contains values that will be substituted by data items from a data structure, the message will have a placeholder for each text-substituted value. The placeholders are indicated by an ampersand (&) and a number in the message text.

The following example shows a custom data dictionary glossary item with text-substitution placeholders. This glossary item could be used as the email subject and body for a report definition that produces customer invoices.

**Figure 4–4 Data Dictionary Glossary Items form**

The screenshot shows the 'Glossary Items' application window. The title bar reads 'Glossary Items'. The menu bar includes 'OK', 'Can...', 'Dis...', 'Abo', 'Links', 'Window', 'OLE ...', and 'Internet'. Below the menu bar are three tabs: 'Item Specifications', 'Item Glossary' (which is selected), and 'Data Structure Template'. The 'Item Glossary' tab contains the following fields and text:

- Alias:** A text box containing '5508'.
- Subject:** A text box containing 'Your &1 Invoice'.
- Body:** A large text area containing the following text:
 

**Your &1 &2 invoice is attached to this email message. If you have questions about your account, please contact our Customer Service department as soon as possible.**

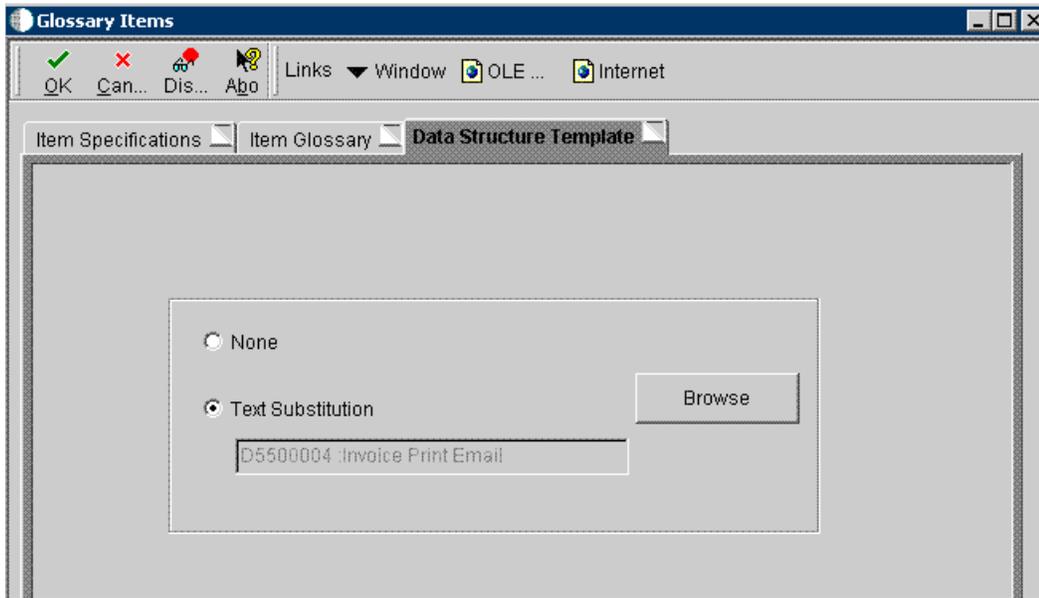
**Thank you.**

In this example, &1 could be the date of the invoice and &2 the type of invoice. Invoices can be produced on a daily, weekly, bi-weekly, semi-monthly, or monthly basis. When users submit the report definition, they can enter values for the &1 and &2 variables so that the subject of the email could be "Your 6/30/10 Invoice," and the body of the message could be "Your 6/30/10 monthly invoice is attached to this email message. If you have questions about your account, please contact our Customer Service department as soon as possible. Thank you."

#### **Data Structure Template**

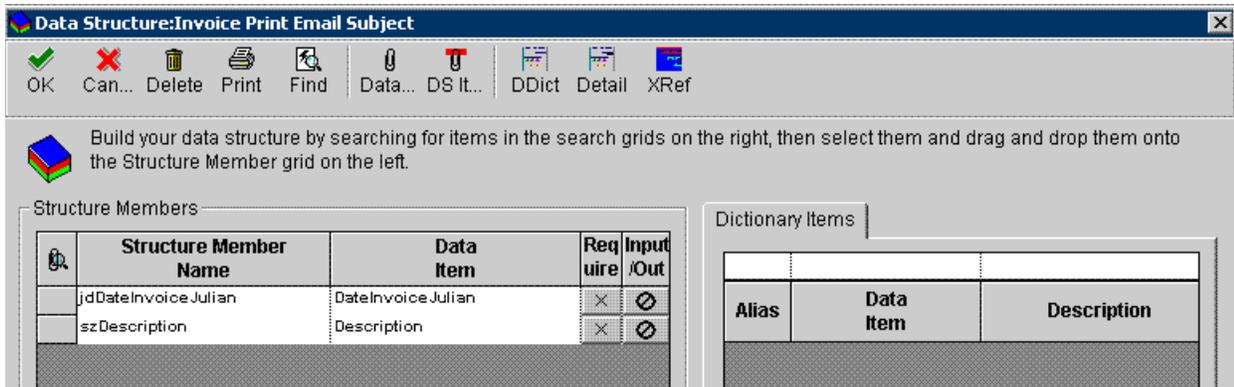
Text-substituted values are defined by data dictionary items in the data structure associated with a data dictionary glossary item. The Data Structure Template tab shows the name of the associated data structure. In the following example, a custom data structure template has been associated with the glossary item.

**Figure 4-5 Glossary Items form - Data Structure Template tab**



The data structure consists of a data item for each text-substituted value in the subject and body of the glossary data item. Because the message in this example has two text-substituted values, the associated data structure consists of two data items, as shown in the following example.

**Figure 4-6 Data structure with data items for text-substitution values**



**4.2.6.1 Report Definition Data Item for Subject and Body**

To provide the subject and text for an email message, enter the name of a glossary data item in the Data Item for Subject and Body field on the Bursting and Delivery form.

If a glossary data item with an associated text-substitution data structure has been defined for the report definition, you can click the Define Values button to enter specific text for the email message when the report definition is submitted. The text entered in the Structure Member Value field on the Structure Member Value Revisions form will be substituted for the variables in the subject and body of the email message. In the following example, 6/30/10 and Monthly are the values for the text-substituted variables.

Figure 4-7 Structure Member Value Revisions form

The screenshot shows a software window titled 'P95620 - [Structure Memeber Value Revisions]'. It has a menu bar with 'File', 'Edit', 'Preferences', 'Window', and 'Help'. Below the menu is a toolbar with icons for 'OK', 'Find', 'Can...', 'New...', 'Dis...', 'Abo', 'Links', 'Displ...', 'OLE ...', and 'Internet'. The main area contains three text input fields: 'Report Definition' with the value 'RD42675', 'Subject And Body Data Item' with '55CS', and 'Structure Name' with 'D5500004'. At the bottom is a table with the following data:

	Display Sequence	Data Item Alias	Structure Member Name	Structure Member Value
	0	IVD	jdDateInvoiceJulian	6/30/10
	1	DESC	szDescription	Monthly

4.2.6.1.1 Dynamic Text-Substitution

Another text-substitution option is to use an XPath (from the XML data output) for the structure member value. Using an XPath enables dynamic text-substitution to occur for text-substituted values in the email subject and body. For bursted reports, the XPath value could be different for each recipient, depending on which XPath was used.

For example, if you wanted the email subject line to dynamically include the invoice date each time the report definition was published, you would provide an XPath similar to the following for the Structure Member Value:

Figure 4-8 Structure Member Value Revisions form

This screenshot is identical to Figure 4-7, but the 'Structure Member Value' for the second row in the table is updated to an XPath expression: '/InvoiceDate\_ID1'.

	Display Sequence	Data Item Alias	Structure Member Name	Structure Member Value
	0	IVD	jdDateInvoiceJulian	/InvoiceDate_ID1
	1	DESC	szDescription	Monthly

In the same way, a data item could be included in the data structure for the invoice number. With dynamic text-substitution, the invoice number would change for each customer in a bursted report.

---



---

**Important:** The XPath must be preceded by a forward slash to be recognized as an XPath and not as a literal value. For example, if you want to dynamically substitute the customer invoice number, you would enter /CustomerInvoiceNumber\_ID1 for the Structure Member Value, assuming that is the XPath in your XML output.

---



---

#### 4.2.6.2 Defining a Data Structure

Access Object Management Workbench.

1. In Object Management Workbench, select a project.
2. Click the Add button.
3. On the Add EnterpriseOne Object to the Project form, select Data Structure, and then click OK.
4. On the Add Object form, enter the appropriate information, and then click OK.
5. In the Object Management Workbench project, select the data structure object and then click the Design button.
6. Click the Design Tools tab, and then click the Data Structure Design button.
7. On the Dictionary Items tab, use the QBE line to locate the data dictionary items that you want to include in the data structure.
8. To include data items in the data structure, drag them from the Dictionary Items tab to Structure Members.
9. When the data structure is complete, click OK.

#### 4.2.6.3 Defining a Glossary Data Item

Access Object Management Workbench.

1. In Object Management Workbench, select a project.
2. Click the Add button.
3. On the Add EnterpriseOne Object to the Project form, select Data Item, and then click OK.

The Data Dictionary Item Type message box appears.

4. Click Yes to add a glossary data item.
5. On the Item Specifications tab, complete the fields as follows:

Field	Description
Alias	Enter a name that identifies the glossary item.
Glossary Group	Enter E for the glossary group. Glossary group E is used for error messages, warning messages, and information messages.
Product Code	Use product codes 55-59 for custom data items.
Product Reporting Code	Use product reporting codes 55-59 for custom data items.

Field	Description
Description	The description is used as the email subject when the report definition is submitted. You can use text-substituted variables in the description.
Error Level	Enter 3 to define the glossary item as an information message.

6. Click the Item Glossary tab.
7. Enter the body of the message in the text area, inserting text-substitution variables as required.
8. Click the Data Structure Template tab.
9. Click the Text Substitution option.
10. Click the Browse button to locate and select the data structure with the data items that will be used for the text-substituted variables.
11. Click OK to save the glossary item.

---

**Note:** You could also use the Work With Data Dictionary Items or the Error Messages selection on menu GH951 to add a type E glossary data item.

---

### 4.3 Modifying BI Publisher Report Definitions

Access the BI Publisher Report Definitions form on the Microsoft Windows client.

1. On the BI Publisher Report Definitions form, select a report definition in the grid and then click Select.
2. On the Update Report Definition form, you can modify these options:
  - Description
  - Product code
  - Product system code
  - Source UBE
  - Version
  - Template
3. To modify output or language options, select Output/Languages from the Form menu.
4. On the Update Report Definition-Languages and Output types form, add options, modify options, or delete options, and then click OK.
5. To modify bursting or delivery options, select Burst/Delivery from the Form menu.
6. On the Update Report Definition-Bursting and Delivery form, add options, modify options, or delete options, and then click OK.
7. On the Update Report Definition form, click OK to save the report definition modifications.

---

---

**Note:** Modifying the Source UBE, Version, or Template fields could invalidate the languages, output types, bursting, and delivery options that were defined previously for the report definition. If you modify any of these values, the Languages and Output Types form and the Bursting and Delivery form will appear automatically, enabling you to make any necessary changes.

---

---

## 4.4 Copying BI Publisher Report Definitions

Copying and modifying a report definition might be more efficient than creating a new, but similar, report definition. The copy feature is available only in the Microsoft Windows version of the BI Publisher Report Definitions application.

To copy a report definition:

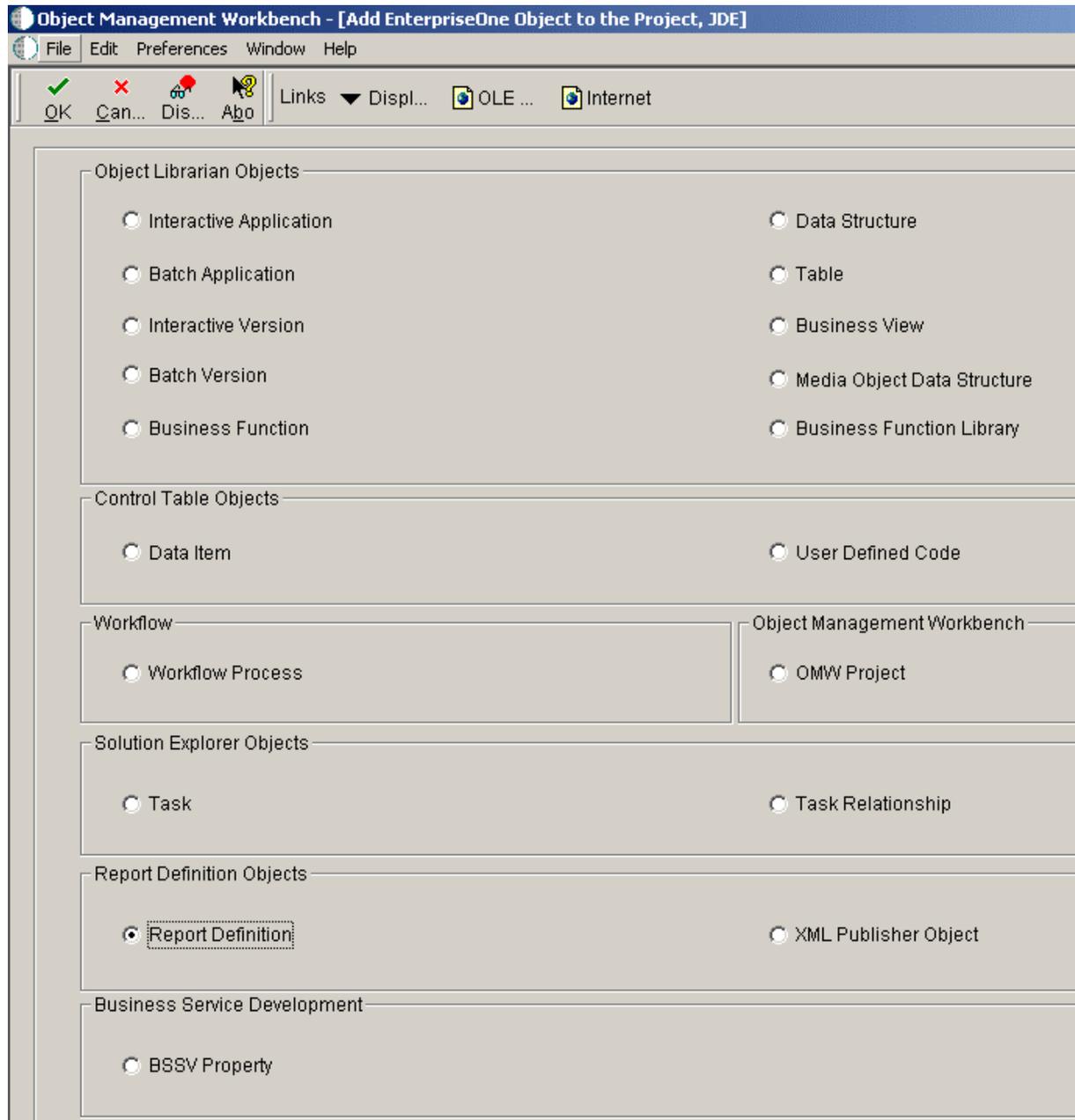
1. Enter **P95620** in the Fast Path field.
2. On the BI Publisher Report Definitions form, select a report definition and then click Copy.
3. On the Copy Report Definition form, enter a name and description for the new report definition.
4. If the report definition will be submitted without user input, click Blind Submission.
5. After completing the required information, click OK.

## 4.5 Managing BI Publisher Objects in Object Management Workbench

When you access the BI Publisher Repository (P95600) or BI Publisher Report Definitions (P95620) applications from JD Edwards Solution Explorer, the objects that you create are added to your default project in Object Management Workbench (OMW). As non-Object Librarian objects, templates and report definitions are not available to users in other environments until they are advanced through project statuses and, finally, transferred to a data source. To advance the status of an object, you must move it from your default project to a project.

In addition to being available in JD Edwards Solution Explorer, you can also access the BI Publisher Repository and BI Publisher Report Definitions applications through OMW. To add new objects within OMW, select an OMW project and then click Add. Select either Report Definition or BI Publisher Object, as this form illustrates:

Figure 4–9 Object Management Workbench - Add Enterprise Object to the Project form



When you click OK, either the BI Publisher Repository application or the BI Publisher Report Definition application launches, depending on the type of object that you are creating.

Once the objects exist in OMW, you can modify them by clicking the Design button, which will launch the appropriate application. You can also use OMW to copy or delete templates and report definitions, as you would with other OMW objects.

See the *JD Edwards EnterpriseOne Tools Object Management Workbench Guide*.

#### 4.5.1 Searching for BI Publisher Objects

To search for BI Publisher objects or report definitions within OMW, in the Category field select either **XML Publisher Objects** or **Report Definitions**. After selecting the

category, the Search Type field enables you to search by object name, description, or system code.

When you perform an Advanced Search within OMW, BI Publisher Objects are type **XMLP** and report definitions are type **RPDF**.

## 4.6 Configuring BI Publisher Objects for Object Management Workbench

To perform actions on BI Publisher objects and report definitions within OMW projects, you must configure object transfer activity rules. For each object type on which you want to perform an action, you must define this information:

- Project statuses at which users can add, design, and delete objects.
- Status changes at which objects will be transferred.
- Project statuses at which object tokens are released.

See the *JD Edwards EnterpriseOne Tools Object Management Workbench Guide*.

---

---

# Submitting JD Edwards EnterpriseOne Report Definitions to BI Publisher

This chapter contains the following topics:

- [Section 5.1, "Understanding the Report Definition Submission Process"](#)
- [Section 5.2, "Submitting Report Definitions from BI Publisher Report Definitions \(P95620\)"](#)
- [Section 5.3, "Submitting Report Definitions from Batch Versions \(P95305\)"](#)
- [Section 5.4, "Reviewing Report Definition Submission Details"](#)

## 5.1 Understanding the Report Definition Submission Process

After a report definition has been created, it must be submitted to BI Publisher to produce the selected output types. You can submit report definitions from the BI Publisher Report Definitions application (P95620). You can also submit a report definition from the Batch Versions application (P98305) by selecting a batch version that is associated with a report definition.

## 5.2 Submitting Report Definitions from BI Publisher Report Definitions (P95620)

You can use BI Publisher Report Definitions (P95620) to submit report definitions to BI Publisher.

### 5.2.1 Submitting from BI Publisher Report Definitions

Select Report Definitions (P95620) from the Report Management menu (GH9111).

1. On the BI Publisher Report Definitions form, find and select the desired report definition.
2. From the Row menu, select Submit.
3. If available, change the output, language, or delivery options.
4. Click OK to submit the report definition to BI Publisher.

## 5.3 Submitting Report Definitions from Batch Versions (P95305)

This section discusses submitting report definitions from the Batch Versions application and describes how to:

- Associate report definitions with batch versions.
- Submit batch versions.
- Access Batch Version Advanced Option Overrides.

You can associate a report definition with a batch version for user convenience. By your doing so, users can submit report definitions from the Batch Versions application by submitting a batch version that is associated with a report definition. To associate a report definition with a batch version, the version must first be checked out. You can check out the version either in Object Management Workbench (OMW) or through the Batch Versions application (P98305) on the Microsoft Windows client.

Some batch versions, such as R42520/XJDE0006, Print Pick Slips - Subsystem, are configured as subsystem jobs. Subsystem jobs offer many advantages, such as the elimination of startup time, one-time initialization of environment and specifications, and better utilization of the processor on the server. You can set up a subsystem report and version to produce BI Publisher output by creating a template and adding a report definition in the Report Definitions application (P95620), with the subsystem report as the source UBE and version.

See [Section 3.1.1, "Templates"](#)

See [Section 4.2.3, "Defining Source of Data, Template Names, and Transformations"](#)

After the report definition has been defined, you can associate it with the batch version if you want the version to produce only BI Publisher output.

For information about subsystem jobs, see "Working with Subsystem Jobs" in the *JD Edwards EnterpriseOne Tools Development Tools: Report Design Aid Guide*.

### 5.3.1 Associating Report Definitions with Batch Versions

Select Batch Versions from the Report Management menu (GH9111).

1. On the Work With Batch Versions form, locate the checked-out version that you want to associate with a report definition.
2. Select the version, and from the Row menu select Version Detail.
3. From the Form menu select Report Definition.
4. Enter the name of a report definition in the Report Definition field, and click OK.
5. Promote the batch version through the OMW project life cycle to make it available to users in other environments.

### 5.3.2 Forms Used to Submit Report Definitions from Batch Versions

Form Name	FormID	Navigation	Usage
Work With Batch Versions - Available Versions	W98305WA	EnterpriseOne Menu, Submit Job	Submit batch versions.
Version Prompting	W98305D	On the Work With Batch Versions form, select a version.	Enter optional data selection or data sequencing criteria. Submit batch versions.

Form Name	FormID	Navigation	Usage
Advanced Version Prompting	W98305I	From the Form menu on the Version Prompting form, select Advanced.	Select options to submit the batch version with the default report definition or to prompt for the report definition.
Batch Versions - Submitted Job Search	W986110BA	On the Form menu on the Work With Batch Versions - Available Versions form, select Submitted Jobs.	View submission details by accessing the Work With Servers form and selecting a server to view submitted jobs.

### 5.3.3 Submitting Batch Versions

Access the Work With Batch Versions - Available Versions form.

Figure 5-1 Work With Batch Versions - Available Versions form

**Work With Batch Versions - Available Versions**

Select Find Add Copy Delete Close Row Form Tools

Batch Application  One Line Per Address

Read Only Report (Y/N)

**Records 1 - 3**

<input type="checkbox"/>	Version	Version Title	User
<input type="checkbox"/>	XJDE0001	One Line Per Address	DEMO
<input type="checkbox"/>	XJDEDOC1	Screen Shot - 1 Address per Line	DEMO
<input type="checkbox"/>	ZJDE0001	One Line Per Address	DEMO

1. Enter the name of a report template in the Batch Application field and click Find.
2. Select a batch version and click Select.
3. On the Version Prompting form, click Submit.

### 5.3.4 Accessing Batch Version Advanced Option Overrides

Access the Advanced Version Prompting form.

Figure 5–2 Advanced Version Prompting form

**Batch Versions - Advanced Version Prompting**

OK Cancel Tools

---

Override Location
  Submit Version Specifications Only

Logging (JDE.log)
  Submit with Report Definition

Tracing (JDEDEBUG.log)
  Prompt for Report Definition

Override Job Queue

UBE Logging Level

If a batch version has a report definition associated with it, the Advanced Version Prompting form provides options to submit the report definition when the batch version is submitted. If you select the Submit with Report Definition option, you can also select the option to prompt for the report definition.

---

**Note:** Processing options associated with the Batch Versions application also determine whether the Advanced Version Prompting options can be selected.

---

#### Override Location

Select to define a different location in which the batch version processes. You must have permissions for this option. When you submit the batch version for processing, you can select a new location from a list of available data sources on the JDE Data Source form. Data sources include the enterprise servers available on the network and the local workstation.

#### Logging (JDE.log)

Select to enable logging for processing of the batch job on the server. To enable logging on the workstation, you must modify the output setting in the workstation jde.ini. However, note that this output setting affects all JD Edwards EnterpriseOne logging. You can select this option without selecting the Tracing option.

#### Tracing (JDEDEBUG.log)

Select to enable tracing for the processing of the batch job on the server. You cannot select this option without selecting the Logging option. The system selects the Logging option for you when you select the Tracing option.

#### Override Job Queue

Select to submit the batch job to another available queue by overriding the job queue that is defined for batch versions in both the jde.ini and the specifications. You must have permissions for this option.

#### UBE Logging Level

Enter a value from 0–6 to indicate the level of detail to be captured in the logs. This option is used in partnership with the logging options. When you select a high value to receive more technical information, you also receive all of the information for the lower values. For example, when you enter a value of 3 (object level messages), you

also receive information for 2 (section level messages), 1 (informative messages), and 0 (error messages).

**Submit Version Specifications Only**

Select to submit version specifications to the server without processing the batch version locally or on the server. The associated report template specifications must already reside on the server to use this feature.

**Submit with Report Definition**

Select to submit the batch version with a report definition. This option is available only if a report definition has been associated with the version.

**Prompt for Report Definition**

Select to prompt the user to select a report definition other than the default to be used for the submission. All batch applications that use this batch version as their source of data and have either no source version specified or this specific version specified will be available for submission.

## 5.4 Reviewing Report Definition Submission Details

This section provides an overview of the Report Definition Jobs application (P95630) and discusses how to:

- View submission details from Report Definition Jobs (P95630)
- View submission details from Work With Submitted Jobs (P986110B)

### 5.4.1 Understanding Report Definition Jobs

After a report definition has been submitted, you can access the submission details and report definition output from Report Definitions Jobs (P95630). The report definition output repository contains the report definition output and details about the report definition submission, such as:

- UBE execution host.
- UBE server job number.
- Report definition execution host.
- Report definition job number.
- Report definition name and description.
- Job status and job status description.
- Origination host.
- User ID.
- Submission date and time.
- Source type.
- Source UBE.
- Source version.

Job status codes are updated as the report definition job progresses. This table shows the available status values for report definition jobs.

Job Status Codes	Definition
D	BI Publishing Done
E	Data Extraction Failure
P	Data Extraction
X	BI Publishing
FD	Delivery Error
FL	Language Unavailable
FO	Output Type Invalid
FR	Report Definition Invalid
FS	System Error (During BI Publishing Processing)
FT	Template or Transformation Unavailable
FX	BI Publishing Error

The Report Definitions Jobs application is also accessible from selections from row menus within the Work With Submitted Jobs application (P986110B). Work With Submitted Jobs is available from the View Job Status selection on the EnterpriseOne Menu or from Batch Versions (P98305).

### 5.4.2 Forms Used to View Report Definition Submission Details

Form Name	FormID	Navigation	Usage
Report Definition Job Control Search	W95630B	On the EnterpriseOne Menu, select EnterpriseOne Menus, EnterpriseOne Life Cycle Tools, Report Management, Report Definition Jobs.	Select a report definition job.
Report Definition Output Repository	W95630E	Select a job control record on Report Definition Job Control Search, and click Select.	Select a report definition output record to view submission details, view the output, access the delivery details for the output, and print the output if the output is type PDF.
Work With Servers	W986116A	On the EnterpriseOne Menu, select View Job Status (P986116).	Find the server that processed a submitted report definition.
Submitted Job Search	W986110B	Select a record on Work With Servers, and click Select.	Select a report definition to view submission details.

### 5.4.3 Viewing Submission Details from Report Definition Jobs

Access the Report Definition Job Control Search form.

Figure 5–3 Report Definition Job Control Search form

**Report Definition Jobs - Report Definition Job Control Search** i ?

Select Find Delete Close Form Row Tools

UBE Execution Host: DENMLSAN101  
 UBE Server Job Number: \*  
 User ID: JDE

Records 1 - 2 Customize Grid

<input type="checkbox"/>	UBE Execution Host	UBE Server Job Number	Report Definition Execution Host	Report Definition Job Number	Report Definition	Report Definition Description
<input type="checkbox"/>	DENMLSAN101	10	DENMLSAN101	12	RDR014021	One Line Per A
<input type="checkbox"/>	DENMLSAN101	11	DENMLSAN101	13	RDR014021	One Line Per A

1. Select a job control record and then click Select.
2. On the Report Definition Output Repository form, view the submission details for the report definition job.

**Note:** In addition to the report definition submission details, the Report Definition Output Repository enables you to view the created output, republish the report definition, print the output, access the delivery details for the output, and redeliver the output.

See [Chapter 6, "Managing JD Edwards EnterpriseOne Report Definition Output"](#).

### 5.4.4 Viewing Submission Details from Work With Submitted Jobs (P986110B)

Access the Submitted Job Search form.

Figure 5–4 Submitted Job Search form

**View Job Status - Submitted Job Search** i |

Select Find Delete Close Row Tools

Host: DENMLSAN101  
 User ID: JDE  
 Job Queue: \*

Records 1 - 10 Customize Grid

<input type="checkbox"/>	Submit Type	Queue	Queue Type	Status	Description	Priority	Type	User	Job Details
<input type="checkbox"/>	BV	QBATCH	01	D	Done	5	UBE	JDE	R0006P_XJDE0001_1
<input checked="" type="checkbox"/>	RD	QBATCH	01	D	Done	5	UBE	JDE	R014021_XJDE0001_2
<input type="checkbox"/>	RD	QBATCH	01	D	Done	5	UBE	JDE	R014021_XJDE0001_3

You can access the Report Definition Output Repository from the Submitted Job Search form. The Submit Type column indicates whether a job was submitted as a batch version (BV) or a report definition (RD).

1. Select a report definition (RD) job, and from the Row menu select View RD Jobs.
2. On the Report Definition Output Repository form, view the submission details for the report definition job.

---

---

**Note:** Additional row menu selections enable you to view BI Publisher output, view report definition output, view report definition source, and republish the report definition.

---

---

See [Chapter 6, "Managing JD Edwards EnterpriseOne Report Definition Output"](#).

---

---

# Managing JD Edwards EnterpriseOne Report Definition Output

This chapter contains the following topics:

- [Section 6.1, "Understanding Report Definition Output"](#)
- [Section 6.2, "Viewing Report Definition Output, Source, and Delivery Details"](#)
- [Section 6.3, "Setting Up Security for Report Definition Jobs and Output"](#)
- [Section 6.4, "Republishing Report Definition Output"](#)
- [Section 6.5, "Redelivering Report Definition Output"](#)
- [Section 6.6, "Archiving BI Publisher Report Output"](#)
- [Section 6.7, "Deleting UBE and Report Definition Output"](#)

## 6.1 Understanding Report Definition Output

After a report definition has been submitted to BI Publisher, you can view the output in the Report Definition Jobs application (P95630). In addition to viewing the submission details, you can view the output, access the delivery details for the output, redeliver the output, view the source, republish a report definition, and delete the output. Because the report definition output is archived as an XML file, the output can be republished without running the batch version (UBE) a second time. You can also print the output from the Report Definition Output Repository if the output type is PDF.

Additionally, from selections from row menus in the Work With Submitted Jobs (P986110B) Submitted Job Search form, you can view BI Publisher output, view the report definition source, republish a report definition, and access the Report Definition Output Repository to view the report definition jobs and output.

## 6.2 Viewing Report Definition Output, Source, and Delivery Details

Selecting a report definition job on the Report Definition Job Control Search form provides access to the output repository for the report definition job. The Report Definition Output Repository form displays details about the output types that were created for the report definition. These details include the source UBE, version, template name, transformation name, object language, and burst indicator. For each output record in the grid, you can view the output, access the delivery details for the output, and print the output if it is a PDF output type.

Selecting an individual record in the grid displays that record's output details on the Report Definition Output Details form. The same information appears on the Report

Definition Output Repository form. The only difference is that the Report Definition Output Repository displays the details in grid columns for all the output records, whereas the Report Definition Output Detail form displays the information for one output record only.

The Report Definition Output Delivery Details form displays information such as delivery status, Address Book number for the delivery, and the delivery location. You can redeliver any of the output from this form, although for security reasons you can redeliver the output only to the original recipients.

## 6.2.1 Example of Report Definition Output

This is an example of report definition output.

**Figure 6–1 Report definition output – Address Book**

**ORACLE**  
**JD EDWARDS ENTERPRISEONE**

One Line Per Address  
2006/4/27  
8:41:43

Address Number	1130
Name	Secrets
Phone Number	610 111-2222
Address	1130 Private Lane
City	Allentown
State	PA
Postal Code	18101
Country	United States

Address Number	3004
Name	Pacific Company, The
Phone Number	602 629-3004
Address	2005 Market Street, Suite 200
City	Philadelphia
State	PA
Postal Code	19103
Country	All Countries

## 6.2.2 Forms Used to View Report Definition Output

Form Name	FormID	Navigation	Usage
Report Definition Job Control Search	W95630B	On the EnterpriseOne Menu, select EnterpriseOne Menus, EnterpriseOne Life Cycle Tools, Report Management, Report Definition Jobs.	Select a report definition job that was submitted to BI Publisher.

Form Name	FormID	Navigation	Usage
Report Definition Output Repository	W95630E	Select a job control record on Report Definition Job Control Search and click Select, or select View RD Output from the Row menu.	Select a report definition job to view details of the output, view the output, access the delivery details for the output, and print the output if the output is type PDF.
Report Definition Output Delivery Details	W95630C	Select a record on the Report Definition Job Control Search form and select View Delivery from the Row menu, or select a record on the Report Definition Output Repository form and select View Delivery from the Row menu.	View delivery details, view the created output, and redeliver the output.
Report Definition Output Details	W95630F	Select a record on the Report Definition Job Control Search form and click Select.	View the output details for one output record.
Work With Servers	W986116A	EnterpriseOne Menu, View Job Status (P986116).	Find the server that processed a submitted report definition.
Submitted Job Search	W986110B	Select a record on the Work With Servers form and click Select.	Select a report definition to view submission details. Selections from row menus enable you to view BI Publisher output, view report definition jobs, view report definition output, view the report definition source, or republish a report definition.

### 6.2.3 Viewing Report Definition Source

The report definition source is the XML output that the batch version (source UBE) created when the report definition was submitted.

Access the Report Definition Job Control Search form.

1. Select a report definition output record.
2. From the Row menu, select View RD Source.

#### 6.2.3.1 Example: Report Definition Source

This is an example of the source that a batch version creates when a report definition is submitted.

**Figure 6–2 Report Definition source**

```

- <Purchase_Order_Print_S42>
  <SysPageNumber>1</SysPageNumber>
  <Line_Number_ID1>1.000</Line_Number_ID1>
  <Change_Order_ID3>0</Change_Order_ID3>
  <DescriptionLine1_ID37>69691</DescriptionLine1_ID37>
  <UnitsTransactionQty_ID25>10.0000</UnitsTransactionQty_ID25>
  <UoM___As_Input_ID35>EA</UoM___As_Input_ID35>
  <PurchasingUnitPrice_ID23>96.9100</PurchasingUnitPrice_ID23>
  <UnitOfMeasurePurchas_ID33>EA</UnitOfMeasurePurchas_ID33>
  <Extended_Price_ID27>969.10</Extended_Price_ID27>
  <Requested_ID17>2005-06-10</Requested_ID17>
  <OrderSuffix_ID29>000</OrderSuffix_ID29>
  <AddressNumberShipTo_ID31>26969</AddressNumberShipTo_ID31>
</Purchase_Order_Print_S42>
<UITMItemNumber_ID2>69691</UITMItemNumber_ID2>
<Tolerance_ID44>0.00</Tolerance_ID44>
<__ID34>-</__ID34>
<Lot_Potency_Range_ID8>Lot Potency Range</Lot_Potency_Range_ID8>
<__ID7>-</__ID7>
<Lot_Grade_Range_ID5>Lot Grade Range</Lot_Grade_Range_ID5>
<PurchaseLineCode01_ID49>Category Code 01</PurchaseLineCode01_ID49>
<PurchaseLineCode03_ID55>Category Code 03</PurchaseLineCode03_ID55>
<Description___Purchase_Line_Co_ID53>.</Description___Purchase_Line_Co_ID53>
<Description___Purchase_Line_Co_ID59>.</Description___Purchase_Line_Co_ID59>
<PurchaseLineCode02_ID51>Category Code 02</PurchaseLineCode02_ID51>
<PurchaseLineCode04_ID57>Category Code 04</PurchaseLineCode04_ID57>
<Description___Purchase_Line_Co_ID54>.</Description___Purchase_Line_Co_ID54>
<Description___Purchase_Line_Co_ID60>.</Description___Purchase_Line_Co_ID60>
</On_Line_Number_S47>
</On_Order_Suffix_S58>
<Total_Order___ID12>Total Order</Total_Order___ID12>
<Order_Extended_Price_ID13>969.10</Order_Extended_Price_ID13>

```

## 6.2.4 Viewing Report Definition Output Details

Access the Report Definition Output Repository form.

Figure 6–3 Report Definition Output Repository form

**Report Definition Output Repository** i ?

Select Find Close Form Row Tools

Report Definition Execution Host

Report Definition Job Number

User ID

Records 1 - 2 Customize Grid

	Report Definition Execution Host	Report Definition Job Number	Report Definition	Report Definition Description	Source Type	Source UBE
<input checked="" type="radio"/>	DENMLSAN101	20	RD000001	One Line Per Address	1	R014021
<input type="radio"/>	DENMLSAN101	20	RD000001	One Line Per Address	1	R014021

1. Select a report definition output record.
2. Review the output details in the grid, or click Select to view the details on the Report Definition Output Details form.

**Report Definition Execution Host**

The host machine where the report definition was submitted to BI Publisher.

**Report Definition Job Number**

The job number that identifies a report definition submission to BI Publisher.

**Report Definition**

The name of the report definition that was submitted to BI Publisher.

**Report Definition Description**

The description of the report definition.

**Source Type**

A code that describes the Source of Data type for the report definition.

**Source UBE**

The source UBE for the report definition that was submitted to BI Publisher.

**Source Version**

The batch version that was used for the report definition that was submitted to BI Publisher.

**UBE Exe Host**

The name of the server that processed the batch version.

**UBE Job Number**

The number assigned to the batch version job on the server.

**Template Name**

The name of the template associated with a report definition output record.

**Transformation Name**

The name of the transformation associated with a report definition output record.

**Object Language**

A code that indicates the language assigned to a report definition output record.

**Output Type**

The BI Publisher output type (UDC H95 | OT) of the report definition output record. Output types depend on the type of template that is attached to the report definition. The available output types are ETEXT, EXCEL, HTML, PDF, PPT, RTF, and XML.

**Machine Key Submitted**

The name of the workstation that submitted the report definition.

**User ID**

The user ID of the user who submitted the report definition.

**Date/Time Submitted**

The date and time that the report definition was submitted.

**Burst Indicator**

Indicates whether bursting is to be performed for a report definition.

This is an example of the output details for one report definition job.

Figure 6-4 Report Definition Output Details form

### Report Definition Jobs - Report Definition Output Details

OK Cancel Tools

#### Report Definition Output Details

RD Execution Host	DENMLSAN101	
RD Job Number	20	
Report Definition	RD000001	<i>One Line Per Address</i>
Source Type	1	<i>Batch Version</i>
Source UBE	R014021	<i>One Line Per Address</i>
Source Version	ZJDE0001	<i>One Line Per Address</i>
Template Name	ADDRESS_BOOK_RTF_FORM	
Transformation Name		
Object Language	EN	<i>English</i>
Output Type	PDF	<i>Portable Document Format</i>
Burst Indicator		
Burst Field		

#### Audit Information

Machine Key Submitted	DENMLSAN101
User ID	JDE
Date/Time Submitted	09/24/2007 20:54:14 UTC+

**Report Definition Execution Host**

The host machine where the report definition was submitted to BI Publisher.

**Report Definition Job Number**

The job number that identifies a report definition submission to BI Publisher.

**Report Definition**

The name of the report definition that was submitted to BI Publisher.

**Report Definition Description**

The description of the report definition.

**Source Type**

A code that describes the Source of Data type for the report definition.

**Source UBE**

The source UBE for the report definition that was submitted to BI Publisher.

**Source Version**

The batch version that was used for the report definition submission to BI Publisher.

**UBE Exe Host**

The name of the server that processed the batch version.

**UBE Job Number**

The number assigned to the batch version job on the server.

**Template Name**

The name of the template associated with a report definition output record.

**Transformation Name**

The name of the transformation associated with a report definition output record.

**Object Language**

A code that indicates the language assigned to a report definition output record.

**Output Type**

The BI Publisher output type (user-defined code (UDC) H95 | OT) of the report definition output record. Output types depend on the type of template that is attached to the report definition. The available output types are ETEXT, EXCEL, HTML, PDF, PPT, RTF, and XML.

**Machine Key Submitted**

The name of the workstation that submitted the report definition.

**User ID**

The user ID of the user who submitted the report definition.

**Date/Time Submitted**

The date and time that the report definition was submitted.

**Burst Indicator**

Indicates whether bursting is to be performed for a report definition.

## 6.2.5 Viewing Report Definition Output

The templates and source UBE in a report definition determine the output that is produced when the report definition is submitted to BI Publisher. You might want to view the output to verify that the format and data are correct, or to see the final results before republishing a report definition.

Access the Report Definition Output Repository form.

1. Select an output record in the grid.
2. From the Row menu, select View Output.

## 6.2.6 Viewing Report Definition Output Delivery Details

Access Report Definition Output Delivery Details.

Figure 6–5 Report Definition Output Delivery Details form

**Report Definitions - Report Definition Output Delivery Details**

Select Find Close Row Tools

Report Definition Execution Host DENMLSAN101

Report Definition Job Number 20

Template Name \*

Transformation Name \*

Object Language \*

Output Type \*

Submission User ID JDE

Records 1 - 2 Customize Grid

<input type="checkbox"/>	Report Definition Execution Host	Report Definition Job Number	Report Definition	Report Definition Description	Delivery Status	Status Description
<input type="checkbox"/>	DENMLSAN101	20	RD000001	One Line Per Address	S	Submitted
<input type="checkbox"/>	DENMLSAN101	20	RD000001	One Line Per Address	S	Submitted

The Report Definition Output Delivery Details form displays information about the delivery status and delivery location for each output type. The delivery details are determined by the delivery options in the report definition at the time of submission. Delivery status, delivery type, and delivery location appear for all output records. The delivery Address Book number and email addresses appear only if they were included in the report definition when it was submitted.

To view the delivery details, access the Report Definition Output Delivery Details form. The grid columns display the delivery details for each output record.

This table explains some of the output delivery details that appear for the report definition output.

#### Delivery Status

Report definition delivery status. Valid values are:

**A:** Address failure (Address Book value, associated email address, or static email address).

**C:** Communication Failure.

**S:** Submitted (to printer or email server).

#### Output Type

The BI Publisher output type (UDC H95 | OT) of the report definition output record. Output types depend on the type of template that is attached to the report definition. The available output types are ETEXT, EXCEL, HTML, PDF, PPT, RTE, and XML.

#### Delivery Type

The type of email delivery that was performed for a report definition. Valid values are:

**A:** The email address assigned to the Address Book number provided.

**E:** The email address provided.

**D:** The email address within the defined Data Driven fields.

**Delivery AB Number**

When a report definition is submitted to BI Publisher and a delivery Address Book number has been defined, all output produced will be emailed to the email address associated with the Address Book number.

**Delivery Location**

The email address where the report definition output was sent.

## 6.2.7 Printing Report Definition Output

Access the Report Definition Output Repository form.

1. Select an output record in the grid with an output type of **PDF**.
2. From the Row menu, select Print Output.

## 6.3 Setting Up Security for Report Definition Jobs and Output

Because report definition output could contain personal or confidential information, you might want to enable security features to:

- Allow users to view only the report definition jobs that they submit.
- Restrict users from viewing certain types of report definition output.

### 6.3.1 Securing Report Definition Jobs

If you have restricted all users to access only their individually-submitted jobs in the Work With Submitted Jobs application (P986110B), you should enable the same security features for report definition jobs in the Report Definition Jobs application (P95630).

The sign-on user ID is the default value for the User ID field on the Report Definition Job Control Search form. If you disable the User ID field for input, the sign-on user ID value cannot be changed and is passed to other forms within the application. Consequently, if you disable the User ID field on all of the forms, users are limited to viewing the output only from the jobs that they submit.

To restrict users to viewing only their report definition jobs (as well as the output from those jobs), use the JD Edwards EnterpriseOne Form Design Aid (FDA) to disable the User ID field on the following forms in the Report Definition Jobs:

- Report Definition Job Control Search (W95630B)
- Report Definition Output Repository (W95630E)
- Report Definition Output Delivery Details (W95630C)

### 6.3.2 Securing Report Definition Output

If you have concerns about the information that might appear in certain types of report definition output, you can enable exit security that restricts the output that users can view. For example, if you do not want users to view the source for a report definition that creates payroll checks, you can create exit security to prevent users from accessing the report definition source. Likewise, you can use exit security to secure users from viewing or printing report definition output.

See the *JD Edwards EnterpriseOne Tools Security Administration Guide*.

This table shows the types of output that you can secure on forms within the Report Definition Jobs application (P95630):

Report Definition Jobs Forms	Exits from Row Menus
Report Definition Job Control Search	View RD Source
Report Definition Output Repository	View Output, Print Output
Report Output Delivery Details	View Output

This table shows the output types (available from row menu exits) that you can secure on forms within the Work With Submitted Jobs application (P986110B):

Work With Submitted Jobs Forms	Exits from Row Menus
Submitted Job Search	BI Publisher, View RD Source

## 6.4 Republishing Report Definition Output

Republishing enables you to reuse the XML output that was generated when a report definition was originally submitted to BI Publisher. The XML result set is archived in the repository and can be used to republish any report definition that uses the same UBE and version. Republishing enables you to create the report definition output without having to run the batch version a second time.

When you republish, you can override the original processing location and submit the job to a different server. Additionally, if other report definitions use the same UBE and version, you can select a different report definition. If a report definition is not selected, the republishing job uses the original report definition.

Access the Report Definition Job Control Search form.

1. Select a report definition job.
2. From the Row menu, select Republish.
3. In the Publish Report Definition Prompt message form, select the values you want from the following options:
  - Override Location
  - Prompt for Report Definition

## 6.5 Redelivering Report Definition Output

Report definition output can be delivered as part of the submission process, or it can be printed or redelivered from the Report Definition Output Delivery Details form. The form displays the delivery status and delivery location for each output type. For security reasons, output can be redelivered only to the original recipient. This feature is especially important for individual payroll records and other personal documents that a report definition might produce.

Access the Report Definition Output Delivery Details form.

1. Select a report definition output record in the grid.
2. From the Row menu, select Redeliver.

## 6.6 Archiving BI Publisher Report Output

BI Publisher report output can be preserved in the file system for archiving purposes. Some companies are legally required to retain output files, and others require searchable output files to comply with review procedures or industry auditing requirements. Note that only the report outputs are archived, not the intermediate XML format files.

### 6.6.1 Configuring the Enterprise Server jde.ini File

To enable archiving, you must configure the following settings in the [UBE] section of the enterprise server jde.ini file.

#### 6.6.1.1 [UBE]

Setting	Value	Purpose
BipSaveOutputOnFs=	0 or 1	A value of 1 enables the archiving of report definition output. The default value is 0 (zero).
BipOutputDirectory=	Any valid path to an output directory on the enterprise server operating system.	Defines the archive location for the BI Publisher report output. The path will vary, depending on your platform. Examples are: Windows NT: e:\bipoutput\ UNIX: /u02/ddp/bipoutput IBM i (always on IFS): /jde/bioutput

If the `BipSaveOutputOnFs` value is 0, archiving is turned off and report output is saved only in the F96531 table, not in the file system. Consequently, the report output is not searchable because PDF output is stored in a compressed blob field in the table.

Archiving is turned on when the `BipSaveOutputOnFs` value is 1. BI Publisher output is saved in the F96531 table in the database, and the physical files are preserved in the output location specified by the `BipOutputDirectory` setting. Contrary to the report output in F96531, the physical report files in the archive are searchable.

#### 6.6.1.2 Configuring UBE Settings with Server Manager

Follow these steps to modify the [UBE] section of the enterprise server jde.ini file:

1. Access and log in to Server Manager.
2. On the Server Manager main page, select the enterprise server that you want to set up for archiving.
3. On the enterprise server page, locate the Configuration menu in the lower left corner.
4. Select Batch Processing.
5. On the Batch Processing page, scroll down to the BI Publisher Save Output on File System setting and set it to **1** to enable archiving.
6. Locate the BI Publisher Output Directory setting and enter the path for the output directory.
7. Log out of Server Manager.

See *JD Edwards EnterpriseOne Tools Release 8.98 Server Manager Guide* available on My Oracle Support (Doc ID: 747330.1).

## 6.6.2 BI Publisher Archived Report Files

The archived report output files follow a naming convention that enables them to be easily identified in the archive. The report output file names begin with the report name and version. The template language and the batch and report definition job numbers are then appended, followed by other system-generated information. The output files look similar to these examples:

- Regular report definition output file: R014021\_XJDE0001\_EN\_110\_31\_xmlp\_1231890467234.pdf
- Bursted report definition output file: R014021\_XJDE0001\_EN\_111\_32\_JDE\_1224\_4524\_14\_1231890577\_EN-US\_ADDRBOOK1\_2.rtf

The file names in these examples are the actual reports that were generated when a report definition was submitted. The important segments in the examples are as follows:

Segment	Description
R014021	The report name
XJDE0001	The report version
EN	The language of the template that was used for the output
110 or 111	JD Edwards EnterpriseOne job number, as shown in P986110 - Work With Submitted Jobs
31 or 32	Report definition job number, as shown in P95630 - Report Definition Jobs
_xmlp or _JDE	_xmlp indicates a report that was not bursted, whereas _JDE indicates a bursted report
.pdf or .rtf	The file extension that indicates the output format.

Maintenance of the archived files is the responsibility of the enterprise. Normal JD Edwards EnterpriseOne maintenance processes will not affect the report output files in the archive.

## 6.7 Deleting UBE and Report Definition Output

You can delete the UBE output and the report definition output for an individual report definition job or you can use processing options for R9861101 to purge all report definition output that is older than a specified number of days.

---

**Note:** BI Publisher report output files that have been preserved in an archive will not be affected by these procedures.

---

See [Chapter 6.6, "Archiving BI Publisher Report Output"](#).

### 6.7.1 Deleting Report Definition Jobs

If you choose to delete the output for a particular report definition job, you must select the job itself, not the individual output records. Deleting some of the output records

would alter the output, giving the impression that the report definition created output that was incomplete. For this reason you must delete the report definition job, thereby removing all of the associated records from the repository.

Access the Report Definition Job Control Search form.

1. Find and select the report definition job that you want to delete.
2. Click Delete.
3. Click OK in response to the question, "Are you sure that you want to delete the selected item?"

### 6.7.2 Deleting UBE Records

The UBE records for each report definition job include the UBE job, UBE logs, and the XML data output. You use the Work With Submitted Jobs application (P986110B) to delete this output for a specific report definition job.

Access the Submitted Job Search form.

1. Select a report definition job.
2. Click Delete.
3. Click OK in response to the question, "Are you sure that you want to delete the selected item?"

### 6.7.3 Purging Report Definition Output

Oracle recommends that you periodically purge report definition output. The Job Master Deletion By Days Old report (R9861101) includes processing options that enable you to specify whether to delete UBE records, report definition records, or both UBE and report definition records. Running the UBE in Proof Mode generates a report that shows the records that will be deleted if you run the UBE in Final Mode. Running the UBE in Final Mode deletes the selected records.

This table shows the output that will be purged for UBE and Report Definition records that are selected for deletion.

Record Type	Output Deleted
UBE records	Job Record, UBE Logs, UBE XML Data Output
Report Definition records	RD Job Record, RD Output, RD Delivery details

### 6.7.4 Forms Used to Submit Job Master Deletion by Days Old Report

Form Name	FormID	Navigation	Usage
Work With Batch Versions - Available Versions	W98305A	In Solution Explorer, select Report Management, Report Management Advanced and Technical Operations, Job Master Deletion by Days Old Report.	Submit a report listing obsolete UBE and report definition jobs submitted to servers and delete obsolete records.

## 6.7.5 Setting Processing Options for Job Master Deletion by Days Old Report (R9861101)

Use these processing options to set up the defaults and versions for the report.

### 6.7.5.1 Defaults

Use this processing option to set up the data source, number of days to query for, type of processing, and record type.

#### 1. Data Source

Specify the name that identifies the data source.

#### 2. Days Old

Specify the number of days to query for old records.

#### 3. Control Mode

Select an option that specifies the type of processing. Valid values are:

- 1 = Proof Mode
- 2 = Final Mode

#### 4. Delete Mode

Specify the records to delete. Valid values are:

- 1 = Delete both UBE and Report Definition records.
- 2 = Delete UBE records only.
- 3 = Delete Report Definition records only.

### 6.7.5.2 Versions

#### 1. Job Control Cleanup Status (R9861102)

You can select one of the following versions. The default version is XJDE0001.

XJDE0001 = Jobs With All Status.

XJDE0002 = Jobs With Done Status.

XJDE0003 = Jobs With Error Status.

XJDE0004 = Jobs With Wait Status (UBE only).



---

---

## Understanding the XML Output

This chapter contains the following topics:

- [Section 7.1, "Designing Batch Applications for BI Publisher"](#)
- [Section 7.2, "Interpreting XML Output"](#)
- [Section 7.3, "Comparing XML Output Formats"](#)

### 7.1 Designing Batch Applications for BI Publisher

When you design batch applications and batch versions for BI Publisher for JD Edwards EnterpriseOne, consider how you use page headers and page footers.

#### 7.1.1 Page Headers and Page Footers

Do not include page headers and page footers in report templates or batch versions that are designed for BI Publisher. Page headers and page footers should be included in the BI Publisher template.

Report developers commonly use page headers as both a page header and a level break header. This combination usage of page headers and level break headers does not work well with BI Publisher.

### 7.2 Interpreting XML Output

You generate XML output by running a batch version or a report definition with a batch version. You use the XML output when creating BI Publisher templates with the Microsoft Word Template Builder.

See *Oracle Business Intelligence Publisher User's Guide*.

The XML output contains information from the batch version specifications. Some of the kinds of information that you should consider when you review your XML output are:

- Section tags.
- Item tags.
- Formatted numbers.
- Formatted dates.
- Page headers and page footers.
- Conditional sections.

- Special characters.
- Level break sections.
- Child sections.
- Report properties.

## 7.2.1 Section Tags

Section tags are determined by the name that is specified in JD Edwards EnterpriseOne Report Design Aid (RDA). The name that you see in the XML output is the same as the name found in the Description field of the Section Properties form.

Because multiple sections in an RDA report template can be named the same, the section tag in the XML output is appended with a section ID. The section ID begins with the letter *S* to indicate that it is a section from the report. The section ID is assigned by RDA to ensure that each report section is unique. The section ID is retrieved from the report specifications and cannot be modified.

This is an example of the sections that are included in the XML output for a batch version of the Purchase Order Print report:

```
- <R43500>
  + <Properties>
  + <Purchase_Order_Print_S42_Column_Headings Language="">
  + <PageHeaders>
  + <Purchase_Order_Print_S42_Group>
</R43500>
```

The first node of the XML output displays the name of the Purchase Order Print report template, R43500. All other nodes are nested under this report template node:

- The first node under the report template name is the Properties node.
- The second node is the Column Headings Language node.  
This node appears only when the batch version includes columnar sections.
- The third node is the PageHeaders node.
- The fourth node is a section that groups all associated level break headers and level break footers with the report detail section.

## 7.2.2 Item Tags

Item tags that you see in the XML output are determined by the data dictionary name of the fields used in the RDA report template. An exception to this rule is that if the data dictionary name has been overridden in RDA, then the override is used as the item tag.

Because multiple data items in an RDA report template can be named the same, the item tag in the XML output is appended with an object ID. The object ID begins with the letters *ID* to indicate that it is a field from the report. The object ID is assigned by RDA to ensure that each report object is unique. The object ID is retrieved from the report specifications and cannot be modified.

This is an example of an individual section that is included in the XML output for a batch version of the Purchase Order Print report:

```
- <Purchase_Order_Print_S42_Group>
  - <On_Ship_To_S45>
    <PageHeaderNumber1</PageHeaderNumber>
```

```

    <DocumentOrderInvoiceE_ID135>Order Number</DocumentOrderInvoiceE_ID135>
    <OrderSuffix_ID140>000</OrderSuffix_ID140>
    <OrderType_ID138>OD</OrderType_ID138>
    <DocumentOrderInvoiceE_ID136>1</DocumentOrderInvoiceE_ID136>
    <Order_Revision_Notes_S65/>
+ <On_Order_Suffix_S58>
    <Total_Order__ID12>Total Order</Total_Order__ID12>
    <Order_Extended_Price_ID13>969.10</Order_Extended_Price_ID13>
+ <Taxes_S49>
+ <Grand_Total_S50>
+ <Buyer_S51>
</On_Ship_To_S45>

```

In this XML output example:

- The <On\_Ship\_To\_S45> node is a level break header.  
The page header number indicates where the data resides in the RDA report.
- The <On\_Order\_Suffix\_S58> is also a level break.
- The information that resides directly under the <On\_Order\_Suffix\_S58> node is level break footer information.  
The level break footer information is at the same level as the level break header information because both level break sections break on the same field.
- The <Taxes\_S49>, <Grand\_Total\_S50>, and <Buyer\_S51> nodes are all conditional sections that are called by the level break footer.

When the node of a section is expanded, you can view the item tags of the individual records. Each piece of data is enclosed within descriptive item tags that include the object ID.

## 7.2.3 Special Characters

Many special characters that are used in RDA and included in EnterpriseOne data are replaced in the XML output. For example:

- '<' is replaced with "&lt;"
- '>' is replaced with "&gt;"
- '&' is replaced with "&amp;"
- apostrophes are replaced with "&apos;"
- quotes are replaced with "&quot;"

Other special characters that are prohibited, as defined in the XML 1.0 specifications, are replaced with an underscore.

## 7.2.4 Formatted Numbers

Numbers are formatted in the BI Publisher output using JAVA BigDecimal String format. This formatting consists of an optional sign ("-") followed by a sequence of decimal digits. These digits can be followed by a fraction and additionally by an exponent, for example, -123456.3455e-3.

## 7.2.5 Formatted Dates

Dates are formatted in Canonical format.



```

    <Order_Extended_Price_ID13>969.10</Order_Extended_Price_ID13>
  - <Taxes_S49>
    <PageHeaderNumber1</PageHeaderNumber>
    <Tax_Group_Summary_ID2>Tax Group Summary</Tax_Group_Summary_ID2>
    <DescriptionTaxArea_ID3>CA</DescriptionTaxArea_ID3>
    <Tax_Rate_1_ID4>10.5000</Tax_Rate_1_ID4>
    <__ID5>%</__ID5>
    <Taxable_Amount_ID9>969.10</Taxable_Amount_ID9>
    <Tax_Amount_ID11>101.76</Tax_Amount_ID11>
  </Taxes_S49>
  + <Grand_Total_S50>
  + <Buyer_S51>
</On_Ship_To_S45>

```

The <Taxes\_S49> node is a conditional section called by the level break footer section. The <Grand\_Total\_S50> and <Buyer\_S51> nodes are also conditional sections called by the level break footer section.

## 7.2.8 Level Break Sections

Level break sections are used in RDA to group large amounts of data into manageable groups. Level break headers typically display a descriptive heading for the group while the level break footer typically displays an aggregate. These level break sections appear in the XML output as children of the section on which they were created.

## 7.2.9 Child Sections

Child sections appear in the XML output as children of the section from which they are called.

## 7.2.10 Report Properties

The following report properties are included in the XML output for auditing and informational purposes:

Property	Description
Version	The object name of the submitted batch version.
Title	The title of the submitted batch version.
Machine	The name of the machine where the batch version was submitted.
Host	The name of the machine where the submitted batch version was processed.
Environment	The name of the environment where the batch version specifications reside.
User	The name of the user who submitted the batch version.
Role	The role of the user who submitted the batch version.
Language	The language in which the batch version was processed.
Company	The name of the company for which data is reported in the submitted batch version.
Release	The JD Edwards EnterpriseOne release that was used to process the batch version.

Property	Description
Date	The date that the batch version was processed.
Time	The time that the batch version was processed.

This is an example of report properties that are included in the XML output for a batch version of the Purchase Order Print report:

```
- <Properties>
<Version>JENTEST1</Version>
<Title>Purchase Order Print</Title>
<Machine>JEHOOD-LAP1</Machine>
<Host>JEHOOD-LAP1</Host>
<Environment>STGAWSC1</Environment>
<User>JDE</User>
<Role>*ALL</Role>
<Company>Oracle - JD Edwards</Company>
<OneWorldRelease>E90</OneWorldRelease>
<Date>2007-09-20</Date>
<Time>12:57:02</Time>
</Properties>
```

## 7.3 Comparing XML Output Formats

Some differences exist between the XML output file that is generated for 8.96 XML Publisher and the output file that is generated in subsequent releases.

The differences between the 8.96 XML output and the XML output in subsequent releases include:

- Supported object types
- General XML format differences
- Properties node
- Column headings node
- Page headers node
- Report details node

### 7.3.1 Supported Object Types

The following RDA object types that were not supported in 8.96 XML Publisher are supported in subsequent releases:

- Report constants
- Column headers
- System date
- System time
- Page number
- Total pages

## 7.3.2 General XML Format Differences

These are some of the differences that affect the entire XML output:

- Empty tags have been removed.
- Hidden fields and sections appear in the output.
- Utimes are formatted in canonical format.
- Constant text appears in the output.
- Conditional sections are displayed as children of the parent section.

Conditional sections were not displayed as children of the parent section in 8.96, rather, they appeared at the same level as the detail report section.

Because of the differences between the XML output file that is generated for 8.96 XML Publisher and the output file that is generated in subsequent releases, Oracle recommends that you create new templates if you are migrating to subsequent releases.

See [Appendix A, "JD Edwards EnterpriseOne BI Publisher Releases"](#).

## 7.3.3 Properties Node

The Properties node of the XML output includes property information regarding the submitted batch version. The Properties node of the XML output in subsequent releases differs from the 8.96 XML output in that it:

- Includes the version name as a separate tag.
- Includes the host machine name.

## 7.3.4 Column Headings Node

Each columnar section of the submitted batch version is preceded with a Column Headings Language node. This node lists all column headings included in the columnar section. The column headings are represented by beginning and ending tags that include the ID assigned to each heading. In between the beginning and ending tags is the column heading as it appears in the batch version.

This section does not exist in the 8.96 XML Publisher output.

## 7.3.5 Page Headers Node

The Page Header nodes of the XML output include page information from the submitted batch version. The Page Header nodes of the XML output in subsequent releases differ from the 8.96 XML Publisher output in these ways:

- Includes the page header number.
- Includes the report name.
- Includes the date that the batch version was submitted.
- Includes the page number.

## 7.3.6 Report Details Node

The report detail node of the XML output in subsequent releases differs from the 8.96 XML Publisher output in these ways:

- The report detail node includes the page header number where the associated data appears.
- Conditional sections are nested under the parent section.

---

---

# Configuring Oracle BI Publisher Enterprise for Interactive Reporting

This chapter contains the following topics:

- [Section 8.1, "Implementing Interactive Reporting with Oracle BI Publisher Enterprise Edition 11g"](#)
- [Section 8.2, "Registering the JD Edwards EnterpriseOne Data Access Driver"](#)
- [Section 8.3, "Configuring the JD Edwards EnterpriseOne Data Access Driver"](#)

## 8.1 Implementing Interactive Reporting with Oracle BI Publisher Enterprise Edition 11g

To implement JD Edwards EnterpriseOne interactive reporting with Oracle Business Intelligence Publisher (OBIEE-BIP), you must install Oracle BI Publisher Enterprise (OBIEE), install the JD Edwards EnterpriseOne Data Access Driver, register the JD Edwards Data Access Driver, and then configure a JDBC Driver in OBIEE-BIP.

Follow these steps to integrate the JD Edwards EnterpriseOne Data Access Driver with OBIEE-BIP.

1. Install Oracle Business Intelligence Enterprise Edition 11g.

---

---

**Note:** WebLogic is part of the OBIEE installation process.

---

---

See *Oracle® Fusion Middleware Installation Guide for Oracle Business Intelligence 11g Release 1 (11.1.1.5)*.

2. Install the JD Edwards EnterpriseOne Data Access Driver.

---

---

**Important!** When you install OBIEE, you might be prompted to install a JD Edwards EnterpriseOne Data Access Driver. If you do not see this option, or after installing the available driver you wish to update the driver with the latest version, follow the manual install steps provided below.

---

---

---

---

**Note:** The Data Access Driver can be downloaded from the JD Edwards EnterpriseOne and World Update Center on My Oracle Support.

---

---

3. Register the JD Edwards EnterpriseOne Data Access Driver in Server Manager.
4. Configure the JDBC Driver (Data Access Driver) in OBIEE-BIP.

### 8.1.1 Manually Installing a JD Edwards EnterpriseOne Data Access Driver

If the JD Edwards EnterpriseOne Data Access Driver was not installed as part of the OBIEE installation process or you wish to install a newer version of the driver, you can install the driver manually.

To install the JD Edwards Data Access Driver manually:

1. Unzip the tools-specific Data Access Driver par file archive (for example, 8.98.n.n-Data-Access-Driver\_06\_70.par). Extract all files from the DADriver\_EAR.jar file.
2. On the server where OBIEE-BIP is installed, locate the home directory for OBIEE.
3. To install the Data Access Driver, copy ALL of the files from the extracted DADriver\_EAR.jar into the <OBIEE\_home>/user\_projects/domains/<bifoundation\_domain>/lib folder:
  - avalon-framework-cvs-20020806.jar
  - castor.jar
  - commons-codec-1.3.jar
  - commons-httpclient-3.0.jar
  - commons-logging.jar
  - e1dadriver.jar
  - fop.jar
  - images.jar
  - j2ee.jar
  - jmxremote.jar
  - jmxremote\_optional.jar
  - jmxri.jar
  - OWResource.jar
  - rtf2fo.jar
  - xalan.jar
  - xerces.jar
  - xmlparserv2.jar

### 8.1.2 Replace the Delivered xerces.jar and xalan.jar Files

The following steps must be performed for Oracle BI Publisher installations that use the Sun Java HotSpot(TM) 64-Bit Server VM. The combination of the JD Edwards delivered xerces.jar and xalan.jar combined with Sun Java HotSpot(TM) 64-Bit Server VM results in a failed Oracle WebLogic Enterprise Manager (EM) service if the xerces.jar and xalan.jar files are not replaced.

To manually replace the xerces.jar file:

1. Locate the Oracle 11g WebLogic xercesImpl-2.9.1.jar common file (for example, <orcl\_BI>/oracle/Middleware/oracle\_common/modules/oracle.owasp\_11.1.1/).
2. Copy the xercesImpl-2.9.1.jar file into the <orcl\_BI>/user\_projects/domains/<bifoundation\_domain>/lib directory.
3. Delete the xalan.jar and xerces.jar files found in the <orcl\_BI>/user\_projects/domains/<bifoundation\_domain>/lib directory.
4. Restart the WebLogic Administrative Console BIP services.

### 8.1.3 Update the Oracle WebLogic PRE\_CLASSPATH

For Oracle BI Publisher 11g 64-bit installations, you must update the Oracle WebLogic PRE\_CLASSPATH as described in the My Oracle Support Document ID 1327145.1.

Review the document, and then follow steps 2 through 4 in the document to update the PRE\_CLASSPATH with the location of the Server Manager Agent folder containing the DAD instance configuration files (for example, Z:\JDE\_HOME\targets\DENPBDS2\_DAD\config).

## 8.2 Registering the JD Edwards EnterpriseOne Data Access Driver

Use Server Manager to register OBIEE and configure the JAS.INI, JDBJ.INI, and JDELOG.PROPERTIES files.

---



---

**Note:** Server Manager registering is for all OBIEE releases.

---



---

To register the JD Edwards Data Access Driver:

1. Log in to Server Manager.
2. On the Managed Instance home page, click the Create New Managed Instance button.
3. On the Create/Register A Managed Instance page, select EnterpriseOne Data Access Driver, and then click Continue.
4. In the Usage Type field, select the 11g BI Publisher option from the drop-down list.

---



---

**Note:** The Software Component box will not display when the Usage Type selected is 11g BI Publisher.

---



---

5. On the Create/Register a Managed Instance form, complete these fields:
  - Install Name.
  - Install Location.

Ensure that you modify the Install Location to match the machine where OBIEE-BIP is installed.

**Important!** Ensure that the path to the BI Publisher installation is correct. If the path is not correct, the registration will fail.

The default value is:

```
<WLS_HOME>\user_
projects\domains\<BIPDOMAIN>\servers\<BIPMANAGEDSERVER>
```

The modified value might be similar to the following:

```
/slot/ems7332/oracle/BIP11g/user_projects/domains/bifoundation_
domain/servers/bi_server1
```

6. Click Continue to proceed.

The following example shows the configuration that enables the data access driver to work within OBIEE-BIP.

**Figure 8–1 Server Manager configuration for data access driver**

The screenshot shows the 'Create/Register A Managed Instance' configuration page. At the top, a progress bar indicates the current step is 'Instance Properties'. Below the progress bar, the following configuration items are listed:

- Primary Security Server: NONE
- System Datasource Name: (empty)
- Database Type: Oracle Database
- Database Name: (empty)
- Database Server Name: (empty)
- Database TCP/IP Port: 0
- Physical Database: (empty)
- Object Owner: (empty)
- Supports Large Objects (LOBS):
- Unicode Database:
- Outgoing JENET Port: 6080
- Incoming JENET Port: 6080
- Bootstrap User: (empty)
- Bootstrap User Password: (empty)
- Bootstrap Role: \*ALL
- Bootstrap Environment: (empty)

7. Click the Create Instance button to complete the Data Access Driver registration.
8. After you complete the registration, the browser is redirected to the EnterpriseOne Data Access Driver home page.

### 8.3 Configuring the JD Edwards EnterpriseOne Data Access Driver

After the JD Edwards EnterpriseOne Data Access Driver has been registered in Server Manager, you must configure the driver in OBIEE-BIP 11g.

## 8.3.1 Understanding JD Edwards EnterpriseOne Data Access

Creating JD Edwards interactive reports with Oracle BI Publisher Enterprise requires access to JD Edwards EnterpriseOne data. After Oracle BI Publisher Enterprise and the Data Access Driver have been installed, you configure BI Publisher to use a JDBC driver to access the JD Edwards EnterpriseOne database.

### 8.3.1.1 JDBC Connection String

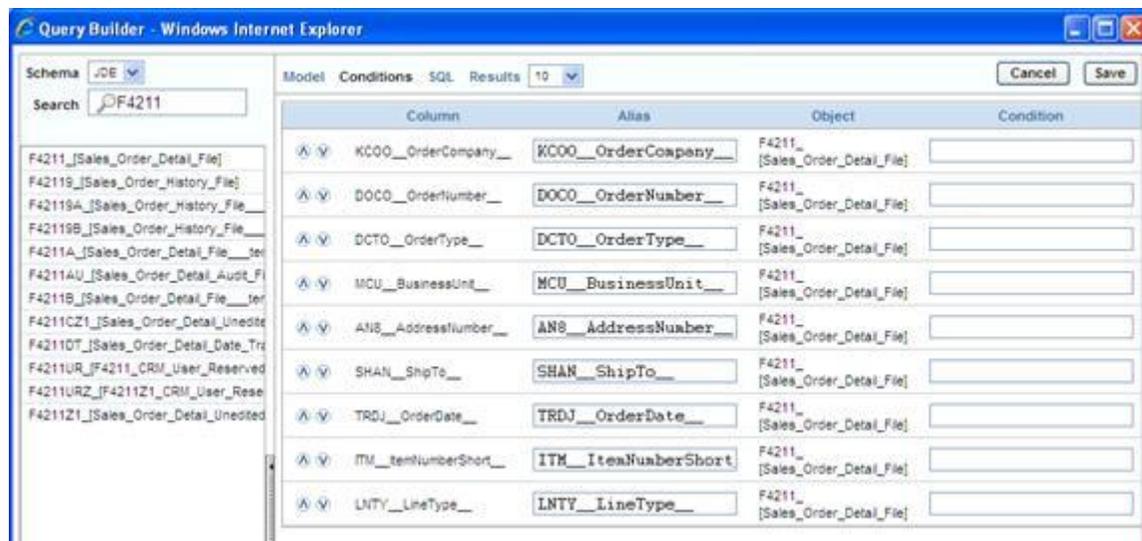
Setting up JD Edwards EnterpriseOne data access is accomplished by defining a JDBC driver connection string in Oracle BI Publisher. The connection string has a predefined format, but you can append information to enable these additional features:

- Specify the JD Edwards EnterpriseOne role when connecting to the database.
- Retrieve table descriptions along with table IDs.
- Retrieve column descriptions from the data dictionary.
- Retrieve the associated user defined code (UDC) description for data fields with associated UDCs.

Providing table descriptions, column descriptions, and UDC descriptions simplifies the process of building queries in BI Publisher. Without these descriptions, a report developer must understand JD Edwards EnterpriseOne table schema when designing reports.

The following example shows how table descriptions and column descriptions appear when you use the BI Publisher Query Builder.

**Figure 8–2 Table descriptions and column descriptions in Query Builder**



**8.3.1.1.1 Specifying the JD Edwards EnterpriseOne Role** You can add information to the end of the connection string that defines the JD Edwards EnterpriseOne role that will be used when connecting to the database; for example, *enterprisone.role=\*ALL*. If the role is not specified in the connection string, the role defined as the bootstrap role will be used. Use Server Manager to locate the bootstrap role.

**8.3.1.1.2 Retrieving Table Descriptions** You can add information to the end of the connection string that enables the JDBC driver to retrieve table descriptions in addition to table names. To display table descriptions, add *TDSC=1* to the connection

string. If the value is 0 or the TDSC tag is not in the connection string, table descriptions will not be retrieved from the database.

**8.3.1.1.3 Retrieving Column Descriptions** You can add information to the end of the connection string that enables the JDBC driver to retrieve column descriptions in addition to column names. The column description is the long column name from the data dictionary. Additionally, the column description is retrieved in the language of the user who is building the query in BI Publisher.

To display the column description, add *CDSC=1* to the connection string. If the CDSC value is 0 or the CDSC tag is not in the connection string, column descriptions will not be retrieved from the database.

**8.3.1.1.4 Retrieving UDC Descriptions** You can add information to the end of the connection string that enables the JDBC driver to retrieve the UDC description for table columns that have an associated UDC. (Each table column is based on a data dictionary item, which could have a UDC assigned to it.)

The UDC description is retrieved in the language of the user who is building the query in BI Publisher. Without the UDC description, the report developer must know which table columns have UDCs associated with them. The report developer can override the column name while designing the report.

To display UDC descriptions, add *UDSC=1* to the connection string. If the value is 0 or the UDSC tag is not in the connection string, UDC descriptions will not be retrieved from the database.

---

---

**Note:** Use a semi-colon between tags when appending them to the connection string. For example, the following connection string shows the appended information for retrieving the table description, column description, and UDC description:

```
jdbc:oracle:enterpriseone://DV812;TDSC=1;CDSC=1;UDSC=1;
```

---

---

### 8.3.1.2 Oracle BI Publisher and JD Edwards EnterpriseOne Security

To create an interactive BI Publisher report, a user must be able to sign on to both Oracle BI Publisher and to the JD Edwards EnterpriseOne database. The connection string for the data source, along with the JD Edwards EnterpriseOne JDBC Driver configuration, specifies the database that BI Publisher will access when creating and running interactive reports.

At the time that the JDBC driver is configured, it is highly recommended that you select the Use Proxy Authentication option for the data source. Using proxy authentication assumes that the user IDs in Oracle BI Publisher and JD Edwards EnterpriseOne are the same, either by duplication or by using Lightweight Directory Access Protocol (LDAP).

With proxy authentication, JD Edwards EnterpriseOne authentication security is enforced against the user name and password in the data source, not the BI Publisher user name that is using the data source. The BI Publisher user name is automatically appended to the database connection string and passed in when the data source is selected. Because the JD Edwards user ID and the BI Publisher user name must be the same, all row and column security assigned to the JD Edwards EnterpriseOne user or its roles is applied to this appended user when connecting to the database using the database connection string. It should also be noted that in this configuration, the user name and password of the JDBC data source in BI Publisher must match the Bootstrap user and password of the JD Edwards EnterpriseOne JDBC driver.

Without proxy authentication, the BI Publisher and JD Edwards user IDs can be different. In this configuration, EnterpriseOne authentication, row, and column security is enforced against the data source user ID and password, not the BI Publisher user that is using the data source. Consequently, any BI Publisher user could have access to the JD Edwards EnterpriseOne database, limited only by the security applied to the data source user ID. Restricting data access would require that BI Publisher users have separate data sources with unique user names and passwords. The data sources would need to be configured with different levels of access to the database, and each BI Publisher user would be required to use the appropriate data source when connecting to the database. In this configuration, the user name and password of the JDBC data source in BI Publisher does not need to match the Bootstrap user and password of the JD Edwards EnterpriseOne JDBC driver.

In summary, the Use Proxy Authentication option is recommended when configuring the BI Publisher JDBC Connection because it enforces database security without having to create multiple data sources.

### 8.3.2 Configuring the JDBC Driver in Oracle Business Intelligence Publisher Enterprise

To configure the JDBC Driver in Oracle BI Publisher:

1. Start the Oracle BI Publisher server.
2. Sign in to Oracle BI Publisher Enterprise Console.
3. Click the Administration tab.
4. Under the Data Sources section, select JDBC Connection.  
The Data Sources form appears, with the JDBC tab selected.
5. Click Add Data Source.
6. Enter the following values on the Add Data Source form:

---



---

**Note:** When configuring the JDBC Driver with the Use Proxy Authentication option, it is required that you use the EnterpriseOne bootstrap user and password (as found in the `jdbj.ini`) for the BI data-source user.

---



---

Field	Value
Data Source Name	<user defined> Example: <i>E1_DAD</i>
Driver Type	<i>Other</i>
Database Driver Class	<i>com.jdedwards.jdbc.driver.JDBCdriver</i>

Field	Value
Connection String	<p><i>jdbc:oracle:enterpriseone://&lt;JD Edwards environment&gt;;</i></p> <p>Example: <i>jdbc: oracle:enterpriseone://DV812;</i></p> <p><b>Note:</b> You can append information to the end of the connection string that defines the JD Edwards EnterpriseOne role that will be used when connecting to the database, for example: <i>enterpriseone.role = *ALL</i>. If the role is not defined in the connection string, the role that is defined as the Bootstrap Role will be used. Locate the Bootstrap Role by using Server Manager.</p> <p><b>Note:</b> You can append information to the end of the connection string to display table, column, or UDC descriptions if desired. <i>TDSC=1</i> displays the table name description, <i>CDSC=1</i> displays the column description, and <i>UDSC=1</i> displays the UDC description.</p> <p>Example of connection string without display options: <i>jdbc:oracle:enterpriseone://DV812;</i></p> <p>Example of connection string with display options: <i>jdbc:oracle:enterpriseone://DV812;TDSC=1;CDSC=1;UDSC=1;</i></p>
User Name	<p>Enter the JD Edwards EnterpriseOne user ID that you will connect to the database with.</p> <p><b>Note:</b> The user name is case-sensitive.</p>
Password	<p>Enter the password for the user ID.</p> <p><b>Note:</b> The password is case-sensitive.</p>
Use Proxy Authentication	<i>Selected or Cleared</i>

---

**Note:** Using proxy authentication assumes that the user IDs in Oracle BI Publisher and JD Edwards EnterpriseOne are the same, either by duplication or by using Lightweight Directory Access Protocol (LDAP).

---

This example shows the completed form:

Figure 8–3 Adding the JDBC Driver as an Oracle BI Publisher data source

Administration > JDBC > Add Data Source

Add Data Source

Apply Cancel

**General**

✓ TIP Please make sure to install the required JDBC driver classes.  
 ✓ TIP With Oracle Fusion Middleware Security Model, select the Use System User checkbox to use the BI System User for your BI Server Database Connection.

\* Data Source Name: TEST

\* Driver Type: Other

\* Database Driver Class: com.jdedwards.jdbc.driver.JDBCdriver  
 (Example: oracle.jdbc.OracleDriver)

\* Connection String: jdbc:oracle:enterpriseone://DV812;TDSC=1;CDSC=1;UDSC=1

Use System User:

\* Username: JDE

Password: ●●●●

Use Proxy Authentication

Test Connection

7. Click Apply.
8. The new JDBC data source should appear in the Data Sources list.

This example shows the completed form:

Figure 8–4 Data Sources list displays newly added data source

Administration > JDBC

Data Sources

JDBC JNDI File LDAP OLAP

Add Data Source

Data Source Name	Connection String	Delete
<a href="#">demo</a>	jdbc:oracle:thin:@HOST:PORT:SID	
<a href="#">Oracle BI EE</a>	jdbc:oraclebi://DNVMTQA02:9703/	
<a href="#">REGCONN</a>	jdbc:oracle:enterpriseone://JDV9007	
<a href="#">REGCONN-DESC</a>	jdbc:oracle:enterpriseone://JDV9007;TDSC=1;CDSC=1;UDSC=1	
<a href="#">TEST</a>	jdbc:oracle:enterpriseone://DV812;TDSC=1;CDSC=1;UDSC=1	

See "Using the Java Database Connectivity Driver" in the *JD Edwards EnterpriseOne Tools Interoperability Guide*.

### 8.3.3 Testing the JDBC Driver Connection

After configuring the JDBC Driver, it is recommended that you test the data source to verify that it is able to communicate and authenticate with the driver. Testing the data source requires that the Enterprise Server/Security Server services are up and running. Additionally, ensure that Oracle BI Publisher is up and running.

To test the JDBC Driver connection:

1. Start the Oracle BI Publisher server.
2. Sign in to Oracle BI Publisher Enterprise Console using an administrative account.
3. Click the Administration tab.
4. Under the Data Sources section, select JDBC Connection.
5. Select the data source that you want to test.
6. Click the Test Connection button to initiate the test.
7. If the data source is able to communicate and authenticate with the Data Access Driver, the "Connection established successfully" message appears.

---

---

**Note:** It is also possible to test the connection when adding the data source.

---

---

---

---

# JD Edwards EnterpriseOne BI Publisher Releases

This appendix contains the following topics:

- [Section A.1, "JD Edwards XML Publisher 8.96 Templates"](#)
- [Section A.2, "Migrating from 8.96 XML Publisher"](#)

JD Edwards EnterpriseOne has provided a reporting solution based on Oracle's BI Publisher since the JD Edwards EnterpriseOne Tools 8.96 release. The solution was extended with the JD Edwards EnterpriseOne Tools 8.97 release by integrating the embedded BI Publisher reports into the JD Edwards EnterpriseOne lifecycle management. The solution has been further extended in JD Edwards EnterpriseOne Tools 8.98 to support interactive reporting.

---

---

**Note:** In the JD Edwards EnterpriseOne Tools 8.96 and 8.97 releases, BI Publisher was called XML Publisher.

---

---

## A.1 JD Edwards XML Publisher 8.96 Templates

As a JD Edwards EnterpriseOne customer, you fall into one of the following categories regarding BI Publisher:

1. You are new to BI Publisher, in that you are a new JD Edwards EnterpriseOne customer or an existing JD Edwards EnterpriseOne customer who has not previously used BI Publisher.
2. You are currently using the 8.96 release of XML Publisher and are migrating to one of these releases:
  - JD Edwards EnterpriseOne Tools release 8.97
  - JD Edwards EnterpriseOne Tools release 8.98
3. You are currently using the 8.96 release of XML Publisher but do not want to migrate to either 8.97 or 8.98 at this time.

Oracle strongly recommends that you migrate to the 8.97 release of XML Publisher or to the 8.98 release of BI Publisher for JD Edwards EnterpriseOne. Migrating to future releases of BI Publisher will become increasingly more difficult if you continue to create objects using the 8.96 XML Publisher.

If you decide to continue using the 8.96 XML Publisher, you should consult *Oracle Business Intelligence Publisher User's Guide*.

## A.2 Migrating from 8.96 XML Publisher

The way that XML output is generated in 8.97 XML Publisher and 8.98 BI Publisher for EnterpriseOne is different from how the XML output was generated in the 8.96 release.

See [Chapter 7, "Understanding the XML Output"](#).

Because of this difference, if you are currently using 8.96 XML Publisher, Oracle recommends that you migrate to either 8.97 XML Publisher or 8.98 BI Publisher for JD Edwards EnterpriseOne.

After upgrading to either 8.97 or 8.98, follow these steps to convert your 8.96 XML Publisher templates:

1. From the BI Publisher Object Repository (P95600), download your 8.96 XML Publisher templates.
2. Create a report definition for each template.
3. Run each report definition to generate new XML output files.
4. Open each existing template.
5. When using Template Builder, from the Data menu, select Load XML Data.
6. Define each required field from the XML output to the template.

For all other methods of creating and modifying templates, make the necessary modifications to the templates to recognize the 8.97 XML Publisher or 8.98 BI Publisher document tags.

---

---

# Glossary

## **Accessor Methods/Assessors**

Java methods to “get” and “set” the elements of a value object or other source file.

## **activity rule**

The criteria by which an object progresses from one given point to the next in a flow.

## **add mode**

A condition of a form that enables users to input data.

## **Advanced Planning Agent (APAg)**

A JD Edwards EnterpriseOne tool that can be used to extract, transform, and load enterprise data. APAg supports access to data sources in the form of relational databases, flat file format, and other data or message encoding, such as XML.

## **application server**

Software that provides the business logic for an application program in a distributed environment. The servers can be Oracle Application Server (OAS) or WebSphere Application Server (WAS).

## **Auto Commit Transaction**

A database connection through which all database operations are immediately written to the database.

## **batch processing**

A process of transferring records from a third-party system to JD Edwards EnterpriseOne.

In JD Edwards EnterpriseOne Financial Management, batch processing enables you to transfer invoices and vouchers that are entered in a system other than JD Edwards EnterpriseOne to JD Edwards EnterpriseOne Accounts Receivable and JD Edwards EnterpriseOne Accounts Payable, respectively. In addition, you can transfer address book information, including customer and supplier records, to JD Edwards EnterpriseOne.

## **batch server**

A server that is designated for running batch processing requests. A batch server typically does not contain a database nor does it run interactive applications.

**batch-of-one**

A transaction method that enables a client application to perform work on a client workstation, then submit the work all at once to a server application for further processing. As a batch process is running on the server, the client application can continue performing other tasks.

**best practices**

Non-mandatory guidelines that help the developer make better design decisions.

**BPEL**

Abbreviation for Business Process Execution Language, a standard web services orchestration language, which enables you to assemble discrete services into an end-to-end process flow.

**BPEL PM**

Abbreviation for Business Process Execution Language Process Manager, a comprehensive infrastructure for creating, deploying, and managing BPEL business processes.

**Build Configuration File**

Configurable settings in a text file that are used by a build program to generate ANT scripts. ANT is a software tool used for automating build processes. These scripts build published business services.

**build engineer**

An actor that is responsible for building, mastering, and packaging artifacts. Some build engineers are responsible for building application artifacts, and some are responsible for building foundation artifacts.

**Build Program**

A WIN32 executable that reads build configuration files and generates an ANT script for building published business services.

**business analyst**

An actor that determines if and why an EnterpriseOne business service needs to be developed.

**business function**

A named set of user-created, reusable business rules and logs that can be called through event rules. Business functions can run a transaction or a subset of a transaction (check inventory, issue work orders, and so on). Business functions also contain the application programming interfaces (APIs) that enable them to be called from a form, a database trigger, or a non-JD Edwards EnterpriseOne application. Business functions can be combined with other business functions, forms, event rules, and other components to make up an application. Business functions can be created through event rules or third-generation languages, such as C. Examples of business functions include Credit Check and Item Availability.

**business function event rule**

See named event rule (NER).

**business service**

EnterpriseOne business logic written in Java. A business service is a collection of one or more artifacts. Unless specified otherwise, a business service implies both a published business service and business service.

**business service artifacts**

Source files, descriptors, and so on that are managed for business service development and are needed for the business service build process.

**business service class method**

A method that accesses resources provided by the business service framework.

**business service configuration files**

Configuration files include, but are not limited to, interop.ini, JDBj.ini, and jdelog.properties.

**business service cross reference**

A key and value data pair used during orchestration. Collectively refers to both the code and the key cross reference in the WSG/XPI based system.

**business service cross-reference utilities**

Utility services installed in a BPEL/ESB environment that are used to access JD Edwards EnterpriseOne orchestration cross-reference data.

**business service development environment**

A framework needed by an integration developer to develop and manage business services.

**business services development tool**

Otherwise known as JDeveloper.

**business service EnterpriseOne object**

A collection of artifacts managed by EnterpriseOne LCM tools. Named and represented within EnterpriseOne LCM similarly to other EnterpriseOne objects like tables, views, forms, and so on.

**business service framework**

Parts of the business service foundation that are specifically for supporting business service development.

**business service payload**

An object that is passed between an enterprise server and a business services server. The business service payload contains the input to the business service when passed to the business services server. The business service payload contains the results from the business service when passed to the Enterprise Server. In the case of notifications, the return business service payload contains the acknowledgement.

**business service property**

Key value data pairs used to control the behavior or functionality of business services.

**Business Service Property Admin Tool**

An EnterpriseOne application for developers and administrators to manage business service property records.

**business service property business service group**

A classification for business service property at the business service level. This is generally a business service name. A business service level contains one or more business service property groups. Each business service property group may contain zero or more business service property records.

**business service property key**

A unique name that identifies the business service property globally in the system.

**business service property utilities**

A utility API used in business service development to access EnterpriseOne business service property data.

**business service property value**

A value for a business service property.

**business service repository**

A source management system, for example ClearCase, where business service artifacts and build files are stored. Or, a physical directory in network.

**business services server**

The physical machine where the business services are located. Business services are run on an application server instance.

**business services source file or business service class**

One type of business service artifact. A text file with the .java file type written to be compiled by a Java compiler.

**business service value object template**

The structural representation of a business service value object used in a C-business function.

**Business Service Value Object Template Utility**

A utility used to create a business service value object template from a business service value object.

**business services server artifact**

The object to be deployed to the business services server.

**business view**

A means for selecting specific columns from one or more JD Edwards EnterpriseOne application tables whose data is used in an application or report. A business view does not select specific rows, nor does it contain any actual data. It is strictly a view through which you can manipulate data.

**central objects merge**

A process that blends a customer's modifications to the objects in a current release with objects in a new release.

**central server**

A server that has been designated to contain the originally installed version of the software (central objects) for deployment to client computers. In a typical JD Edwards EnterpriseOne installation, the software is loaded on to one machine—the central

server. Then, copies of the software are pushed out or downloaded to various workstations attached to it. That way, if the software is altered or corrupted through its use on workstations, an original set of objects (central objects) is always available on the central server.

**charts**

Tables of information in JD Edwards EnterpriseOne that appear on forms in the software.

**check-in repository**

A repository for developers to check in and check out business service artifacts. There are multiple check-in repositories. Each can be used for a different purpose (for example, development, production, testing, and so on).

**checksum**

A fixed-size datum computed from an arbitrary block of digital data for the purpose of detecting accidental errors that may have been introduced during its transmission or storage. JD Edwards EnterpriseOne uses the checksum to verify the integrity of packages that have been downloaded by recomputing the checksum of the downloaded package and comparing it with the checksum of the original package. The procedure that yields the checksum from the data is called a checksum function or checksum algorithm. JD Edwards EnterpriseOne uses the MD5 and STA-1 checksum algorithms.

**connector**

Component-based interoperability model that enables third-party applications and JD Edwards EnterpriseOne to share logic and data. The JD Edwards EnterpriseOne connector architecture includes Java and COM connectors.

**Control Table Workbench**

An application that, during the Installation Workbench processing, runs the batch applications for the planned merges that update the data dictionary, user-defined codes, menus, and user override tables.

**control tables merge**

A process that blends a customer's modifications to the control tables with the data that accompanies a new release.

**correlation data**

The data used to tie HTTP responses with requests that consist of business service name and method.

**credentials**

A valid set of JD Edwards EnterpriseOne username/password/environment/role, EnterpriseOne session, or EnterpriseOne token.

**cross-reference utility services**

Utility services installed in a BPEL/ESB environment that access EnterpriseOne cross-reference data.

**database credentials**

A valid database username/password.

**database server**

A server in a local area network that maintains a database and performs searches for client computers.

**Data Source Workbench**

An application that, during the Installation Workbench process, copies all data sources that are defined in the installation plan from the Data Source Master and Table and Data Source Sizing tables in the Planner data source to the system-release number data source. It also updates the Data Source Plan detail record to reflect completion.

**deployment artifacts**

Artifacts that are needed for the deployment process, such as servers, ports, and such.

**deployment server**

A server that is used to install, maintain, and distribute software to one or more enterprise servers and client workstations.

**direct connect**

A transaction method in which a client application communicates interactively and directly with a server application.

See also batch-of-one and store-and-forward.

**Do Not Translate (DNT)**

A type of data source that must exist on the iSeries because of BLOB restrictions.

**embedded application server instance**

An OC4J instance started by and running wholly within JDeveloper.

**edit code**

A code that indicates how a specific value for a report or a form should appear or be formatted. The default edit codes that pertain to reporting require particular attention because they account for a substantial amount of information.

**edit mode**

A condition of a form that enables users to change data.

**edit rule**

A method used for formatting and validating user entries against a predefined rule or set of rules.

**Electronic Data Interchange (EDI)**

An interoperability model that enables paperless computer-to-computer exchange of business transactions between JD Edwards EnterpriseOne and third-party systems. Companies that use EDI must have translator software to convert data from the EDI standard format to the formats of their computer systems.

**embedded event rule**

An event rule that is specific to a particular table or application. Examples include form-to-form calls, hiding a field based on a processing option value, and calling a business function. Contrast with the business function event rule.

**Employee Work Center**

A central location for sending and receiving all JD Edwards EnterpriseOne messages (system and user generated), regardless of the originating application or user. Each user has a mailbox that contains workflow and other messages, including Active Messages.

**enterprise server**

A server that contains the database and the logic for JD Edwards EnterpriseOne.

**Enterprise Service Bus (ESB)**

Middleware infrastructure products or technologies based on web services standards that enable a service-oriented architecture using an event-driven and XML-based messaging framework (the bus).

**EnterpriseOne administrator**

An actor responsible for the EnterpriseOne administration system.

**EnterpriseOne credentials**

A user ID, password, environment, and role used to validate a user of EnterpriseOne.

**EnterpriseOne development client**

Historically called “fat client,” a collection of installed EnterpriseOne components required to develop EnterpriseOne artifacts, including the Microsoft Windows client and design tools.

**EnterpriseOne extension**

A JDeveloper component (plug-in) specific to EnterpriseOne. A JDeveloper wizard is a specific example of an extension.

**EnterpriseOne object**

A reusable piece of code that is used to build applications. Object types include tables, forms, business functions, data dictionary items, batch processes, business views, event rules, versions, data structures, and media objects.

**EnterpriseOne process**

A software process that enables JD Edwards EnterpriseOne clients and servers to handle processing requests and run transactions. A client runs one process, and servers can have multiple instances of a process. JD Edwards EnterpriseOne processes can also be dedicated to specific tasks (for example, workflow messages and data replication) to ensure that critical processes don't have to wait if the server is particularly busy.

**EnterpriseOne resource**

Any EnterpriseOne table, metadata, business function, dictionary information, or other information restricted to authorized users.

**Environment Workbench**

An application that, during the Installation Workbench process, copies the environment information and Object Configuration Manager tables for each environment from the Planner data source to the system-release number data source. It also updates the Environment Plan detail record to reflect completion.

**escalation monitor**

A batch process that monitors pending requests or activities and restarts or forwards them to the next step or user after they have been inactive for a specified amount of time.

**event rule**

A logic statement that instructs the system to perform one or more operations based on an activity that can occur in a specific application, such as entering a form or exiting a field.

**explicit transaction**

Transaction used by a business service developer to explicitly control the type (auto or manual) and the scope of transaction boundaries within a business service.

**exposed method or value object**

Published business service source files or parts of published business service source files that are part of the published interface. These are part of the contract with the customer.

**fast path**

A command prompt that enables the user to move quickly among menus and applications by using specific commands.

**file server**

A server that stores files to be accessed by other computers on the network. Unlike a disk server, which appears to the user as a remote disk drive, a file server is a sophisticated device that not only stores files, but also manages them and maintains order as network users request files and make changes to these files.

**final mode**

The report processing mode of a processing mode of a program that updates or creates data records.

**foundation**

A framework that must be accessible for execution of business services at runtime. This includes, but is not limited to, the Java Connector and JDBj.

**FTP server**

A server that responds to requests for files via file transfer protocol.

**HTTP Adapter**

A generic set of services that are used to do the basic HTTP operations, such as GET, POST, PUT, DELETE, TRACE, HEAD, and OPTIONS with the provided URL.

**instantiate**

A Java term meaning "to create." When a class is instantiated, a new instance is created.

**integration developer**

The user of the system who develops, runs, and debugs the EnterpriseOne business services. The integration developer uses the EnterpriseOne business services to develop these components.

**integration point (IP)**

The business logic in previous implementations of EnterpriseOne that exposes a document level interface. This type of logic used to be called XBPs. In EnterpriseOne 8.11, IPs are implemented in Web Services Gateway powered by webMethods.

**integration server**

A server that facilitates interaction between diverse operating systems and applications across internal and external networked computer systems.

**integrity test**

A process used to supplement a company's internal balancing procedures by locating and reporting balancing problems and data inconsistencies.

**interface table**

See Z table.

**internal method or value object**

Business service source files or parts of business service source files that are not part of the published interface. These could be private or protected methods. These could be value objects not used in published methods.

**interoperability model**

A method for third-party systems to connect to or access JD Edwards EnterpriseOne.

**in-your-face error**

In JD Edwards EnterpriseOne, a form-level property which, when enabled, causes the text of application errors to appear on the form.

**jargon**

An alternative data dictionary item description that JD Edwards EnterpriseOne appears based on the product code of the current object.

**Java application server**

A component-based server that resides in the middle-tier of a server-centric architecture. This server provides middleware services for security and state maintenance, along with data access and persistence.

**JDBNET**

A database driver that enables heterogeneous servers to access each other's data.

**JDEBASE Database Middleware**

A JD Edwards EnterpriseOne proprietary database middleware package that provides platform-independent APIs, along with client-to-server access.

**JDECallObject**

An API used by business functions to invoke other business functions.

**jde.ini**

A JD Edwards EnterpriseOne file (or member for iSeries) that provides the runtime settings required for JD Edwards EnterpriseOne initialization. Specific versions of the file or member must reside on every machine running JD Edwards EnterpriseOne. This includes workstations and servers.

**JDEIPC**

Communications programming tools used by server code to regulate access to the same data in multiprocess environments, communicate and coordinate between processes, and create new processes.

**jde.log**

The main diagnostic log file of JD Edwards EnterpriseOne. This file is always located in the root directory on the primary drive and contains status and error messages from the startup and operation of JD Edwards EnterpriseOne.

**JDENET**

A JD Edwards EnterpriseOne proprietary communications middleware package. This package is a peer-to-peer, message-based, socket-based, multiprocess communications middleware solution. It handles client-to-server and server-to-server communications for all JD Edwards EnterpriseOne supported platforms.

**JDeveloper Project**

An artifact that JDeveloper uses to categorize and compile source files.

**JDeveloper Workspace**

An artifact that JDeveloper uses to organize project files. It contains one or more project files.

**JMS Queue**

A Java Messaging service queue used for point-to-point messaging.

**listener service**

A listener that listens for XML messages over HTTP.

**local repository**

A developer's local development environment that is used to store business service artifacts.

**Location Workbench**

An application that, during the Installation Workbench process, copies all locations that are defined in the installation plan from the Location Master table in the Planner data source to the system data source.

**logic server**

A server in a distributed network that provides the business logic for an application program. In a typical configuration, pristine objects are replicated on to the logic server from the central server. The logic server, in conjunction with workstations, actually performs the processing required when JD Edwards EnterpriseOne software runs.

**MailMerge Workbench**

An application that merges Microsoft Word 6.0 (or higher) word-processing documents with JD Edwards EnterpriseOne records to automatically print business documents. You can use MailMerge Workbench to print documents, such as form letters about verification of employment.

**Manual Commit transaction**

A database connection where all database operations delay writing to the database until a call to commit is made.

**master business function (MBF)**

An interactive master file that serves as a central location for adding, changing, and updating information in a database. Master business functions pass information between data entry forms and the appropriate tables. These master functions provide a common set of functions that contain all of the necessary default and editing rules for related programs. MBFs contain logic that ensures the integrity of adding, updating, and deleting information from databases.

**master table**

See published table.

**media storage object**

Files that use one of the following naming conventions that are not organized into table format: Gxxx, xxxGT, or GTxxx.

**message center**

A central location for sending and receiving all JD Edwards EnterpriseOne messages (system and user generated), regardless of the originating application or user.

**messaging adapter**

An interoperability model that enables third-party systems to connect to JD Edwards EnterpriseOne to exchange information through the use of messaging queues.

**messaging server**

A server that handles messages that are sent for use by other programs using a messaging API. Messaging servers typically employ a middleware program to perform their functions.

**Monitoring Application**

An EnterpriseOne tool provided for an administrator to get statistical information for various EnterpriseOne servers, reset statistics, and set notifications.

**named event rule (NER)**

Encapsulated, reusable business logic created using event rules, rather than C programming. NERs are also called business function event rules. NERs can be reused in multiple places by multiple programs. This modularity lends itself to streamlining, reusability of code, and less work.

**Object Configuration Manager (OCM)**

In JD Edwards EnterpriseOne, the object request broker and control center for the runtime environment. OCM keeps track of the runtime locations for business functions, data, and batch applications. When one of these objects is called, OCM directs access to it using defaults and overrides for a given environment and user.

**Object Librarian**

A repository of all versions, applications, and business functions reusable in building applications. Object Librarian provides check-out and check-in capabilities for developers, and it controls the creation, modification, and use of JD Edwards EnterpriseOne objects. Object Librarian supports multiple environments (such as

production and development) and enables objects to be easily moved from one environment to another.

### **Object Librarian merge**

A process that blends any modifications to the Object Librarian in a previous release into the Object Librarian in a new release.

### **Open Data Access (ODA)**

An interoperability model that enables you to use SQL statements to extract JD Edwards EnterpriseOne data for summarization and report generation.

### **Output Stream Access (OSA)**

An interoperability model that enables you to set up an interface for JD Edwards EnterpriseOne to pass data to another software package, such as Microsoft Excel, for processing.

### **package**

JD Edwards EnterpriseOne objects are installed to workstations in packages from the deployment server. A package can be compared to a bill of material or kit that indicates the necessary objects for that workstation and where on the deployment server the installation program can find them. It is point-in-time snapshot of the central objects on the deployment server.

### **package build**

A software application that facilitates the deployment of software changes and new applications to existing users. Additionally, in JD Edwards EnterpriseOne, a package build can be a compiled version of the software. When you upgrade your version of the ERP software, for example, you are said to take a package build.

Consider the following context: “Also, do not transfer business functions into the production path code until you are ready to deploy, because a global build of business functions done during a package build will automatically include the new functions.” The process of creating a package build is often referred to, as it is in this example, simply as “a package build.”

### **package location**

The directory structure location for the package and its set of replicated objects. This is usually \\deployment server\release\path\_code\package\package name. The subdirectories under this path are where the replicated objects for the package are placed. This is also referred to as where the package is built or stored.

### **Package Workbench**

An application that, during the Installation Workbench process, transfers the package information tables from the Planner data source to the system-release number data source. It also updates the Package Plan detail record to reflect completion.

### **Pathcode Directory**

The specific portion of the file system on the EnterpriseOne development client where EnterpriseOne development artifacts are stored.

### **patterns**

General repeatable solutions to a commonly occurring problem in software design. For business service development, the focus is on the object relationships and interactions.

For orchestrations, the focus is on the integration patterns (for example, synchronous and asynchronous request/response, publish, notify, and receive/reply).

**print server**

The interface between a printer and a network that enables network clients to connect to the printer and send their print jobs to it. A print server can be a computer, separate hardware device, or even hardware that resides inside of the printer itself.

**pristine environment**

A JD Edwards EnterpriseOne environment used to test unaltered objects with JD Edwards EnterpriseOne demonstration data or for training classes. You must have this environment so that you can compare pristine objects that you modify.

**processing option**

A data structure that enables users to supply parameters that regulate the running of a batch program or report. For example, you can use processing options to specify default values for certain fields, to determine how information appears or is printed, to specify date ranges, to supply runtime values that regulate program execution, and so on.

**production environment**

A JD Edwards EnterpriseOne environment in which users operate EnterpriseOne software.

**Production Published Business Services Web Service**

Published business services web service deployed to a production application server.

**program temporary fix (PTF)**

A representation of changes to JD Edwards EnterpriseOne software that your organization receives on magnetic tapes or disks.

**project**

In JD Edwards EnterpriseOne, a virtual container for objects being developed in Object Management Workbench.

**promotion path**

The designated path for advancing objects or projects in a workflow. The following is the normal promotion cycle (path):

11>21>26>28>38>01

In this path, 11 equals new project pending review, 21 equals programming, 26 equals QA test/review, 28 equals QA test/review complete, 38 equals in production, 01 equals complete. During the normal project promotion cycle, developers check objects out of and into the development path code and then promote them to the prototype path code. The objects are then moved to the productions path code before declaring them complete.

**proxy server**

A server that acts as a barrier between a workstation and the internet so that the enterprise can ensure security, administrative control, and caching service.

**published business service**

EnterpriseOne service level logic and interface. A classification of a published business service indicating the intention to be exposed to external (non-EnterpriseOne) systems.

**published business service identification information**

Information about a published business service used to determine relevant authorization records. Published business services + method name, published business services, or \*ALL.

**published business service web service**

Published business services components packaged as J2EE Web Service (namely, a J2EE EAR file that contains business service classes, business service foundation, configuration files, and web service artifacts).

**published table**

Also called a master table, this is the central copy to be replicated to other machines. Residing on the publisher machine, the F98DRPUB table identifies all of the published tables and their associated publishers in the enterprise.

**publisher**

The server that is responsible for the published table. The F98DRPUB table identifies all of the published tables and their associated publishers in the enterprise.

**QBE**

An abbreviation for query by example. In JD Edwards EnterpriseOne, the QBE line is the top line on a detail area that is used for filtering data.

**real-time event**

A message triggered from EnterpriseOne application logic that is intended for external systems to consume.

**refresh**

A function used to modify JD Edwards EnterpriseOne software, or subset of it, such as a table or business data, so that it functions at a new release or cumulative update level.

**replication server**

A server that is responsible for replicating central objects to client machines.

**rules**

Mandatory guidelines that are not enforced by tooling, but must be followed in order to accomplish the desired results and to meet specified standards.

**secure by default**

A security model that assumes that a user does not have permission to execute an object unless there is a specific record indicating such permissions.

**Secure Socket Layer (SSL)**

A security protocol that provides communication privacy. SSL enables client and server applications to communicate in a way that is designed to prevent eavesdropping, tampering, and message forgery.

**selection**

Found on JD Edwards EnterpriseOne menus, a selection represents functions that you can access from a menu. To make a selection, type the associated number in the Selection field and press Enter.

**serialize**

The process of converting an object or data into a format for storage or transmission across a network connection link with the ability to reconstruct the original data or objects when needed.

**Server Workbench**

An application that, during the Installation Workbench process, copies the server configuration files from the Planner data source to the system-release number data source. The application also updates the Server Plan detail record to reflect completion.

**SOA**

Abbreviation for Service Oriented Architecture.

**softcoding**

A coding technique that enables an administrator to manipulate site-specific variables that affect the execution of a given process.

**source repository**

A repository for HTTP adapter and listener service development environment artifacts.

**Specification merge**

A merge that comprises three merges: Object Librarian merge, Versions List merge, and Central Objects merge. The merges blend customer modifications with data that accompanies a new release.

**specification**

A complete description of a JD Edwards EnterpriseOne object. Each object has its own specification, or name, which is used to build applications.

**Specification Table Merge Workbench**

An application that, during the Installation Workbench process, runs the batch applications that update the specification tables.

**SSL Certificate**

A special message signed by a certificate authority that contains the name of a user and that user's public key in such a way that anyone can "verify" that the message was signed by no one other than the certification authority and thereby develop trust in the user's public key.

**store-and-forward**

The mode of processing that enables users who are disconnected from a server to enter transactions and then later connect to the server to upload those transactions.

**subscriber table**

Table F98DRSUB, which is stored on the publisher server with the F98DRPUB table and identifies all of the subscriber machines for each published table.

**super class**

An inheritance concept of the Java language where a class is an instance of something, but is also more specific. "Tree" might be the super class of "Oak" and "Elm," for example.

**table access management (TAM)**

The JD Edwards EnterpriseOne component that handles the storage and retrieval of use-defined data. TAM stores information, such as data dictionary definitions; application and report specifications; event rules; table definitions; business function input parameters and library information; and data structure definitions for running applications, reports, and business functions.

**Table Conversion Workbench**

An interoperability model that enables the exchange of information between JD Edwards EnterpriseOne and third-party systems using non-JD Edwards EnterpriseOne tables.

**table conversion**

An interoperability model that enables the exchange of information between JD Edwards EnterpriseOne and third-party systems using non-JD Edwards EnterpriseOne tables.

**table event rules**

Logic that is attached to database triggers that runs whenever the action specified by the trigger occurs against the table. Although JD Edwards EnterpriseOne enables event rules to be attached to application events, this functionality is application specific. Table event rules provide embedded logic at the table level.

**terminal server**

A server that enables terminals, microcomputers, and other devices to connect to a network or host computer or to devices attached to that particular computer.

**transaction processing (TP) monitor**

A monitor that controls data transfer between local and remote terminals and the applications that originated them. TP monitors also protect data integrity in the distributed environment and may include programs that validate data and format terminal screens.

**transaction processing method**

A method related to the management of a manual commit transaction boundary (for example, start, commit, rollback, and cancel).

**transaction set**

An electronic business transaction (electronic data interchange standard document) made up of segments.

**trigger**

One of several events specific to data dictionary items. You can attach logic to a data dictionary item that the system processes automatically when the event occurs.

**triggering event**

A specific workflow event that requires special action or has defined consequences or resulting actions.

**user identification information**

User ID, role, or \*public.

**User Overrides merge**

Adds new user override records into a customer's user override table.

**value object**

A specific type of source file that holds input or output data, much like a data structure passes data. Value objects can be exposed (used in a published business service) or internal, and input or output. They are comprised of simple and complex elements and accessories to those elements.

**versioning a published business service**

Adding additional functionality/interfaces to the published business services without modifying the existing functionality/interfaces.

**Versions List merge**

The Versions List merge preserves any non-XJDE and non-ZJDE version specifications for objects that are valid in the new release, as well as their processing options data.

**visual assist**

Forms that can be invoked from a control via a trigger to assist the user in determining what data belongs in the control.

**vocabulary override**

An alternate description for a data dictionary item that appears on a specific JD Edwards EnterpriseOne form or report.

**web application server**

A web server that enables web applications to exchange data with the back-end systems and databases used in eBusiness transactions.

**web server**

A server that sends information as requested by a browser, using the TCP/IP set of protocols. A web server can do more than just coordination of requests from browsers; it can do anything a normal server can do, such as house applications or data. Any computer can be turned into a web server by installing server software and connecting the machine to the internet.

**Web Service Description Language (WSDL)**

An XML format for describing network services.

**Web Service Inspection Language (WSIL)**

An XML format for assisting in the inspection of a site for available services and a set of rules for how inspection-related information should be made.

**web service softcoding record**

An XML document that contains values that are used to configure a web service proxy. This document identifies the endpoint and conditionally includes security information.

**web service softcoding template**

An XML document that provides the structure for a soft coded record.

**Where clause**

The portion of a database operation that specifies which records the database operation will affect.

**Windows terminal server**

A multiuser server that enables terminals and minimally configured computers to display Windows applications even if they are not capable of running Windows software themselves. All client processing is performed centrally at the Windows terminal server and only display, keystroke, and mouse commands are transmitted over the network to the client terminal device.

**wizard**

A type of JDeveloper extension used to walk the user through a series of steps.

**workbench**

A program that enables users to access a group of related programs from a single entry point. Typically, the programs that you access from a workbench are used to complete a large business process. For example, you use the JD Edwards EnterpriseOne Payroll Cycle Workbench (P07210) to access all of the programs that the system uses to process payroll, print payments, create payroll reports, create journal entries, and update payroll history. Examples of JD Edwards EnterpriseOne workbenches include Service Management Workbench (P90CD020), Line Scheduling Workbench (P3153), Planning Workbench (P13700), Auditor's Workbench (P09E115), and Payroll Cycle Workbench.

**workflow**

The automation of a business process, in whole or in part, during which documents, information, or tasks are passed from one participant to another for action, according to a set of procedural rules.

**workgroup server**

A server that usually contains subsets of data replicated from a master network server. A workgroup server does not perform application or batch processing.

**XAPI events**

A service that uses system calls to capture JD Edwards EnterpriseOne transactions as they occur and then calls third-party software, end users, and other JD Edwards EnterpriseOne systems that have requested notification when the specified transactions occur to return a response.

**XML CallObject**

An interoperability capability that enables you to call business functions.

**XML Dispatch**

An interoperability capability that provides a single point of entry for all XML documents coming into JD Edwards EnterpriseOne for responses.

**XML List**

An interoperability capability that enables you to request and receive JD Edwards EnterpriseOne database information in chunks.

**XML Service**

An interoperability capability that enables you to request events from one JD Edwards EnterpriseOne system and receive a response from another JD Edwards EnterpriseOne system.

**XML Transaction**

An interoperability capability that enables you to use a predefined transaction type to send information to or request information from JD Edwards EnterpriseOne. XML transaction uses interface table functionality.

**XML Transaction Service (XTS)**

Transforms an XML document that is not in the JD Edwards EnterpriseOne format into an XML document that can be processed by JD Edwards EnterpriseOne. XTS then transforms the response back to the request originator XML format.

**Z event**

A service that uses interface table functionality to capture JD Edwards EnterpriseOne transactions and provide notification to third-party software, end users, and other JD Edwards EnterpriseOne systems that have requested to be notified when certain transactions occur.

**Z table**

A working table where non-JD Edwards EnterpriseOne information can be stored and then processed into JD Edwards EnterpriseOne. Z tables also can be used to retrieve JD Edwards EnterpriseOne data. Z tables are also known as interface tables.

**Z transaction**

Third-party data that is properly formatted in interface tables for updating to the JD Edwards EnterpriseOne database.



## A

---

associations  
location, 3-5

## B

---

batch applications  
designing for BI Publisher, 7-1  
formatting considerations, 7-1  
batch versions  
advanced option overrides, 5-3  
submitting, 5-3  
BI Publisher  
benefits of, 2-2  
embedded reporting process overview, 2-5  
getting started, 1-1  
interactive reporting process overview, 2-5  
BI Publisher Object Repository  
overview, 3-1  
BI Publisher Object Repository (P95600), 3-3  
BI Publisher objects  
creating, 3-6  
default file size, 3-4  
deleting, 3-5  
location, 3-4  
modifying, 3-9  
modifying effective dates, 3-9  
modifying properties, 3-8  
revising, 3-9  
types, 3-4  
understanding, 3-3  
updating, 3-5  
uploading, 3-7  
uploading and updating, 3-3  
BI Publisher report definitions  
creating, 4-2  
BI Publisher reporting  
embedded, 2-1  
implementing interactive reporting, 8-1  
interactive, 2-1

## C

---

child sections, 7-5  
conditional sections, 7-4

connection string  
for JDBC driver, 8-5  
creating BI Publisher objects  
updating objects, 3-1

## D

---

Data Access Driver  
configuring in Oracle BI Publisher, 8-4  
installing manually, 8-2  
registering in Server Manager, 8-3

## E

---

effective dates, modifying for BI Publisher  
objects, 3-9

## F

---

formatted numbers, 7-3

## I

---

Interactive BI Publisher reporting  
implementing, 8-1  
item tags, 7-2

## J

---

JDBC connection string, 8-5  
JDBC driver  
configuration steps, 8-7  
installing manually, 8-2  
overview, 8-4  
registering in Server Manager, 8-3  
JDBC driver connection  
testing, 8-9  
JDE Data Source form, 5-4  
JDEDEBUG.log  
enabling, 5-4  
jde.ini  
enabling logging, 5-4  
JDE.log  
enabling, 5-4

## **L**

---

level break sections, 7-5

logs

defining the level of detail to capture, 5-5

enabling logging, 5-4

enabling tracing, 5-4

## **O**

---

Overview, 1-1, 2-1

## **P**

---

page footers, 7-4

page headers, 7-1, 7-4

prerequisite, 3-5

properties, modifying for BI Publisher objects, 3-8

proxy authentication, 8-6

## **R**

---

report definition jobs, 5-5

Report definition output example, 6-2

report definition, Batch Version submission

process, 5-1

report definition, understanding the submission

process, 5-1

report definitions

submitting to BI Publisher, 5-1

report properties, 7-5

## **S**

---

section tags, 7-2

security

for interactive reporting with Oracle BI

Publisher, 8-6

special characters, 7-3

## **T**

---

templates

output types, 4-6

understanding, 3-1

testing a JDBC driver connection, 8-9

transformations

understanding, 3-2

translations

understanding, 3-2

## **U**

---

Use Proxy Authentication option, 8-6