JD Edwards EnterpriseOne Tools

BI Publisher for JD Edwards EnterpriseOne Guide

9.2

JD Edwards EnterpriseOne Tools
BI Publisher for JD Edwards EnterpriseOne Guide

9.2

Part Number: E53612-07

Contents

Preface

Welcome to the JD Edwards EnterpriseOne documentation.

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 that have purchased support 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 Information

For additional information about JD Edwards EnterpriseOne applications, features, content, and training, visit the JD Edwards EnterpriseOne pages on the JD Edwards Resource Library located at:

`http://learnjde.com`

Conventions

The following text conventions are used in this document:

Convention	Meaning
Bold	Boldface type indicates graphical user interface elements associated with an action or terms defined in text or the glossary.
Italics	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
`Monospace`	Monospace type indicates commands within a paragraph, URLs, code examples, text that appears on a screen, or text that you enter.
> Oracle by Example	Indicates a link to an Oracle by Example (OBE). OBEs provide hands-on, step- by-step instructions, including screen captures that guide you through a process using your own environment. Access to OBEs requires a valid Oracle account.

1 Introduction to BI Publisher for JD Edwards EnterpriseOne

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.
- BI Publisher reporting with JD Edwards Data Access Drivers.

  You can also create ad-hoc, dynamic reports with BI Publisher using JD Edwards EnterpriseOne Data Access Driver. The JD Edwards EnterpriseOne Data Access Driver is a read-only JDBC driver, which is used to access JD Edwards EnterpriseOne data for BI Publisher reports.

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.

Creating BI Publisher Reports with JD Edwards Data Access Driver

Chapter 8 describes the steps to configure JD Edwards EnterpriseOne for creating BI Publisher reports with JD Edwards Data Access Driver.

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 Tools 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.

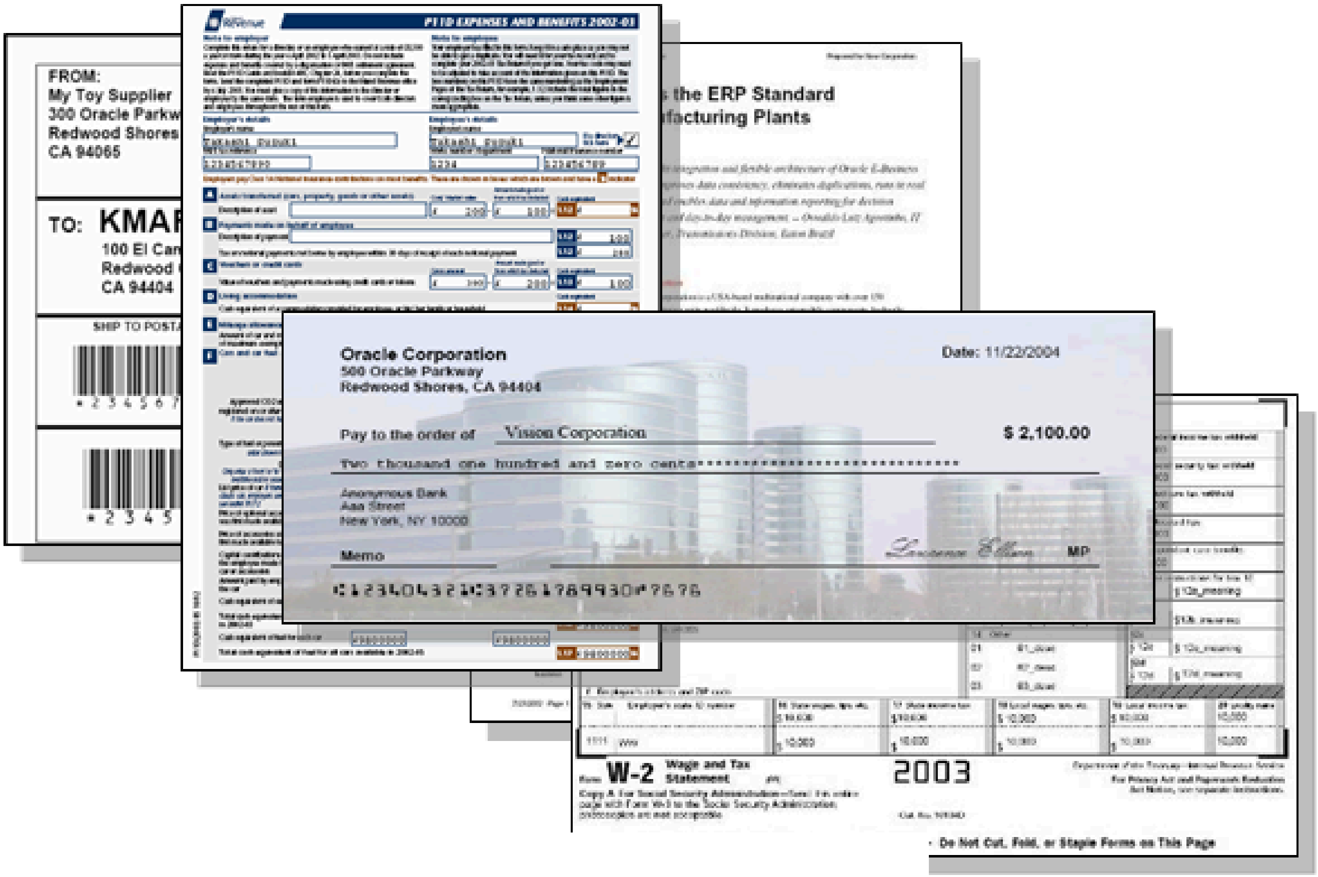
2 Understanding BI Publisher for JD Edwards EnterpriseOne

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 was further extended in JD Edwards EnterpriseOne 8.98 Tools release to support creating, managing, and processing reports in Oracle BI Publisher with JD Edwards Data Access Driver using the Oracle BI Publisher 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:



FROM:
My Toy Supplier
300 Oracle Parkw
Redwood Shores
CA 94065

TO: KMAR
100 El Can
Redwood
CA 94404

SHIP TO POSTA

P11D EXPENSES AND BENEFITS 2002-03

the ERP Standard
facturing Plants

Oracle Corporation
500 Oracle Parkway
Redwood Shores, CA 94404

Date: 11/22/2004

Pay to the order of Vision Corporation $ 2,100.00

Two thousand one hundred and zero cents******************************

Anonymous Bank
Aaa Street
New York, NY 10000

Memo

Lawrence Ellison MP

Form W-2 Wage and Tax Statement 2003

Copy A For Social Security Administration

Do Not Cut, Fold, or Staple Forms on This Page

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.

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.

Reporting with BI Publisher

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 BI Publisher reports using the Oracle BI Publisher Enterprise toolset.

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.

BI Publisher Reporting with JD Edwards Data Access Driver

BI Publisher reporting with JD Edwards Data Access Driver 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.

BI Publisher Reporting Comparison

The following table summarizes the recommendations for using embedded BI Publisher reporting and BI Publisher reporting with JD Edwards Data Access Driver.

Scenario	Embedded	Data Access Driver
An event within JD Edwards EnterpriseOne triggers the report: • Pick Slips • Invoice Print • AP Checks • W2	Yes	No
A non-JD Edwards EnterpriseOne event triggers the report: • Open Account Receivables • Budget to Actual • Inventory levels • Ad-hoc reports	No	Yes
An update to the JD Edwards EnterpriseOne data is performed: • Update status codes • Modify dates • Adjust inventory quantities • Create history records	Yes	No

Scenario	Embedded	Data Access Driver
The report only reads JD Edwards EnterpriseOne data: • Customer activity • General Ledger accounts • Travel expenditures • Summary of suppliers	Yes	Yes
The report needs to perform business logic: • Calculate taxes • Schedule delivery dates • Calculate salaries and benefits calculations • Calculate sales commission calculations	Yes	No

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.

BI Publisher Reporting with JD Edwards Data Access Driver Overview

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

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 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 *Creating Oracle BI Publisher Reports with JD Edwards Data Access Driver*

See the *Oracle® Fusion Middleware Report Designer's Guide for Oracle Business Intelligence Publisher* for information about creating reports with Oracle Business Intelligence Publisher Enterprise.

3 Creating JD Edwards EnterpriseOne Embedded BI Publisher Objects

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

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® Fusion Middleware Report Designer's Guide for Oracle Business Intelligence Publisher* 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. The following menu items work only with Oracle BI Publisher 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® Fusion Middleware Report Designer's Guide for Oracle Business Intelligence Publisher* . Instructions and tutorials for using the Template Builder are available from the readme and help files delivered with the tool.

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.

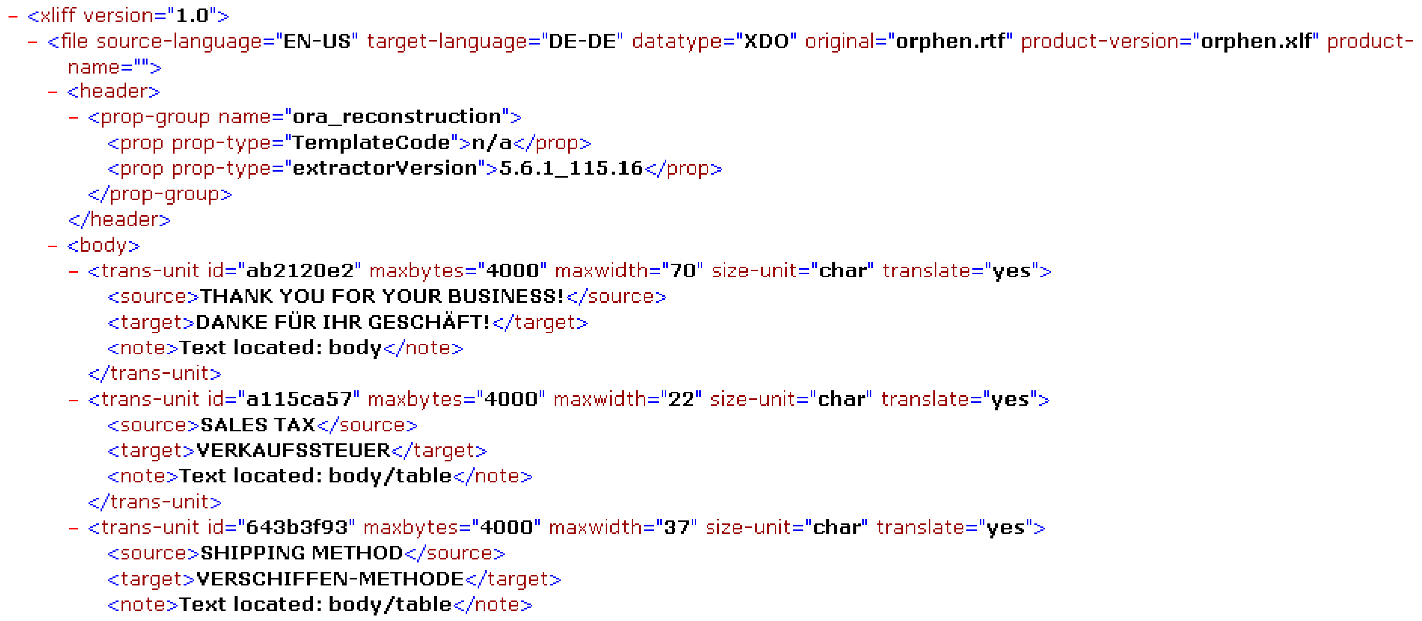
Transformations

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.

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.



```
- <xliff 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.

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.

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 the following recommended naming conventions.

The template name can be a maximum of 100 characters and should be formatted as TPwwwxxxxyyzz, such as TP743005TR1 where:

- TP = Template for JD Edwards EnterpriseOne Embedded BI Publisher
- www = System code
- xxxx = Three to four characters to identify the associated UBE. For example, use 3005 if the source UBE is R743005
- yy = Template type, such as TR for a .rtf template
- zz = A sequentially assigned number to identify the different outputs associated with the same UBE. For example, TP743005TR1 and TP743005TR2 could produce different .rtf outputs, whereas TP74.005TS1 could produce .xsl output.

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

BI Publisher Object	BI Publisher Object Type	File Type
-	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.

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.

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.

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)

Prerequisite

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

See *Introduction to BI Publisher for JD Edwards EnterpriseOne*.

Forms Used to Upload, Update and Delete BI Publisher Objects

Form Name	FormID	Navigation	Usage
BI Publisher Object Repository	W95600E	From the Navigator 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.

Form Name	FormID	Navigation	Usage
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.

Creating BI Publisher Objects

Access the Add BI Publisher Object to Repository form.

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.

Uploading BI Publisher Objects

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

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.

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.

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.

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.

> **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.

Revising BI Publisher Objects

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

See *Oracle® Fusion Middleware Report Designer's Guide for Oracle Business Intelligence Publisher* .

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.

4 Creating JD Edwards EnterpriseOne BI Publisher Report Definitions

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 *Creating JD Edwards EnterpriseOne Embedded BI Publisher Objects*.

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.

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.

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 **P95620** in the Fast Path field. 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.
Add Report Definition-Bursting and Delivery	W95620J	Click Next on the Add Report Definitions-Languages 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. On the Update Report Definition form, from the Form menu select Output/Languages or Burst/Delivery to update the desired options.	Update a report definition.

Report Definition Naming Conventions

Report definition names can be a maximum of 10 characters and should be formatted as RDwwwxxxxyy, such as RD743005A where:

RD = Report definition template

www = Three characters for the system code

xxxx = Three to four characters to identify the associated UBE. For example, use 3005 if the source UBE is R743005)

yy = One to two sequential characters to identify the different report definitions associated with the same UBE, for example RD743005A and RD753005B.

Defining Source of Data, Template Names, and Transformations

Access the Add Report Definition form.

Report Definition
Enter the name of the report definition, using a maximum of 10 characters and should be formatted as RDwwwxxxxyy, such as RD743005A. 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 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.

Defining Languages and Output Types

Access the Languages and Output Types form.

> **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.

Defining Bursting and Delivery Options

The Update Report Definition - Bursting and Delivery form contains the following two tabs:

- Bursting/Delivery
- Override Output File Name

Access the Bursting/Delivery tab in the Update Report Definition - Bursting and Delivery form.

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 static printer, which is the default printer assigned to a combination of a user/role, a report, a version, a host, and an environment in the Printers application (P98616). Selecting this option directs all report definition PDF output to the default 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 Delivery

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

Distribution List Type for Email

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 Driven Printer Mapping

Select this option to enable a report definition to send output to specific printers based on data generated by the source batch version or report version.

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 or printers 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.

Access the Override Output File Name tab in the Update Report Definition - Bursting and Delivery form:

Override Output File Name

Select this option to customize the name of the output file from a report definition.

See *Customizing Report Definition Output File Name*

Data Driven Delivery

In addition to the options to send report definition output to static printers and specific email addresses, you can direct output to recipients and printers based on an XPath field from the XML data. For bursted reports, the XPath value could be different for each recipient, depending on which XPath is used.

Three different combinations are possible with data driven delivery:

- Data driven email only.
- Data driven printers only.
- Both data driven email and printers.

The system handles email to address book number and distribution list types differently than other recipient values. Data driven delivery for address book numbers and distribution lists relies on existing JD Edwards EnterpriseOne address book functionality. As a result, you only need to enter an XML field that contains address book numbers (such as <AddressNumber_ID1>) as the data driven recipient for report definition output.

In the case of data driven delivery to printers, you must first create a relationship between an appropriate XML field (such as <Company_ID5>, <CostCenter_ID7>, etc.) and a printer delivery mapping name. The XML field is also known as a delivery tag. Once the mapping name has been created, you associate different XML field values with the desired printer delivery name and assign a printer device to each field value as required.

To use both data driven email and data driven printer delivery for the same report definition submission, you must create a custom report variable (RV) in Report Design Aid and populate it with either an address book number (for email delivery) or an appropriate XML field (such as <Company_ID5>, <CostCenter_ID7>, etc.) for printer delivery. You would then enter the custom delivery tag as the delivery tag in the Report Definitions Printer Mapping application (P95621).

See *Combined Data Driven Email and Data Driven Printing*

Data Driven Email

If you select the Data Driven Delivery option for report definition output, you can direct output to address book numbers contained in the XML field (such as AddressNumber_ID1) that you enter in the Data Driven Recipient grid on the Bursting and Delivery form. Additionally, if you specify a burst field, many different reports can be created from one report definition submission.

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. You enter the complete XPath of the tag that defines the level break section (as it appears in the XML data output) in the Burst Field on the Bursting and Delivery form.

The level break section can contain an email recipient in one of the following formats:

- An address book number for an individual or a distribution list.
- An SMTP email address.
- An SMTP distribution list.

Data driven email delivery requires that email addresses be set up correctly in the Address Book application (P01012).

See *"Adding Electronic Address Information to Who's Who Records" in the JD Edwards EnterpriseOne Applications Address Book Implementation Guide*

For external email, JD Edwards EnterpriseOne provides an integrated mail system that is able to send messages to external email addresses. The system uses the Simple Mail Transfer Protocol (SMTP) to do this. SMTP is a TCP/IP protocol for sending messages from one computer to another on a network. SMTP is used on the internet to route messages.

In addition to having address book email addresses, data driven email delivery to external email addresses requires changes to the JDEMAIL section of the JDE.ini file, so that JD Edwards EnterpriseOne can communicate with the mail server.

See *"Setting Up External Mail Access" in the JD Edwards EnterpriseOne Tools Workflow Tools Guide*

Defining Data Driven Email Recipients

Access the Report Definition application (P95620).

1. Select a report definition in the grid, and then click the Select button.

   > **Note:** If you are adding a new report definition, you can define a data driven email recipient on the Bursting and Delivery form, as described below.

2. On the Update Report Definition form, from the Form menu, select Burst/Delivery.
3. (Optional) If you want to burst the report, enter an XPath from the XML output in the Burst Field.
4. In the Delivery section, click the Data Driven Delivery checkbox.
5. (Optional) Enter a distribution list type if you want to send

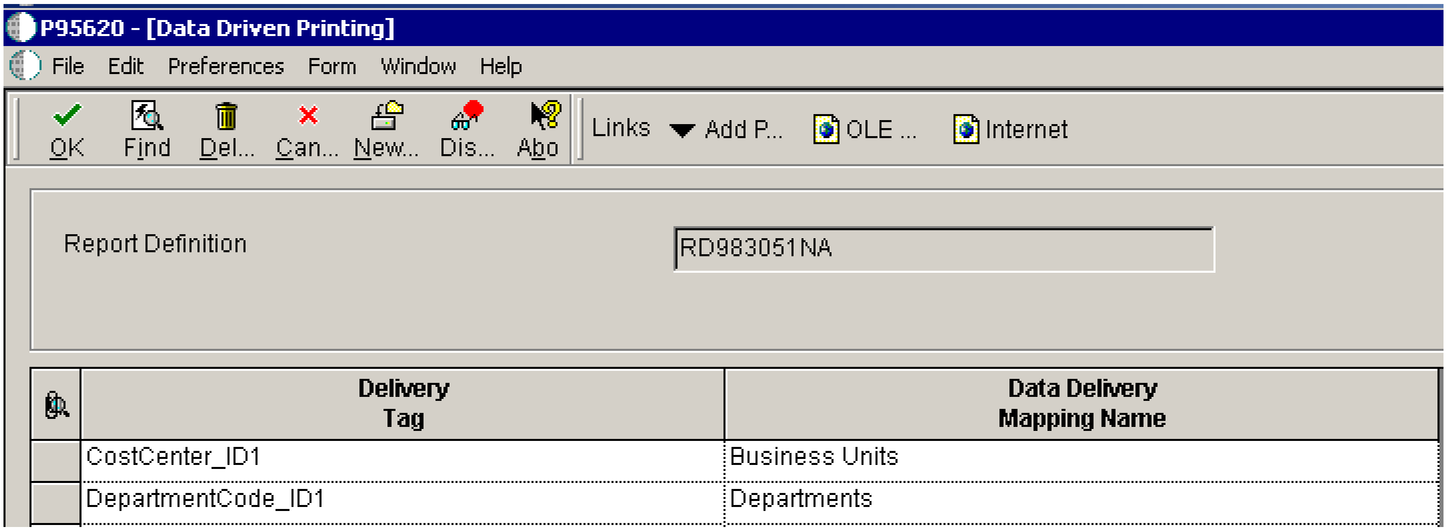
   the output to a distribution list.
6. In the Data Driven Recipient column, enter an XPath from the XML output that contains the recipient data.
7. Click OK to save the data driven email information.

Data Driven Printers

If you select the Data Driven Delivery option for report definition output, you can direct the output to specific printers based on an XPath field (a delivery XML tag) from the XML data. For bursted reports, the XPath field value could be different for each recipient, depending on which XPath is used.

Clicking the Data Driven Printer Mapping button launches the Data Driven Printing form. The value in the Delivery Tag column is passed into the Data Driven Printing form from the data driven recipient that you select on the Bursting and Delivery form. You can change the delivery tag on the Data Driven Printing form if necessary, or enter a new delivery tag from the XML data

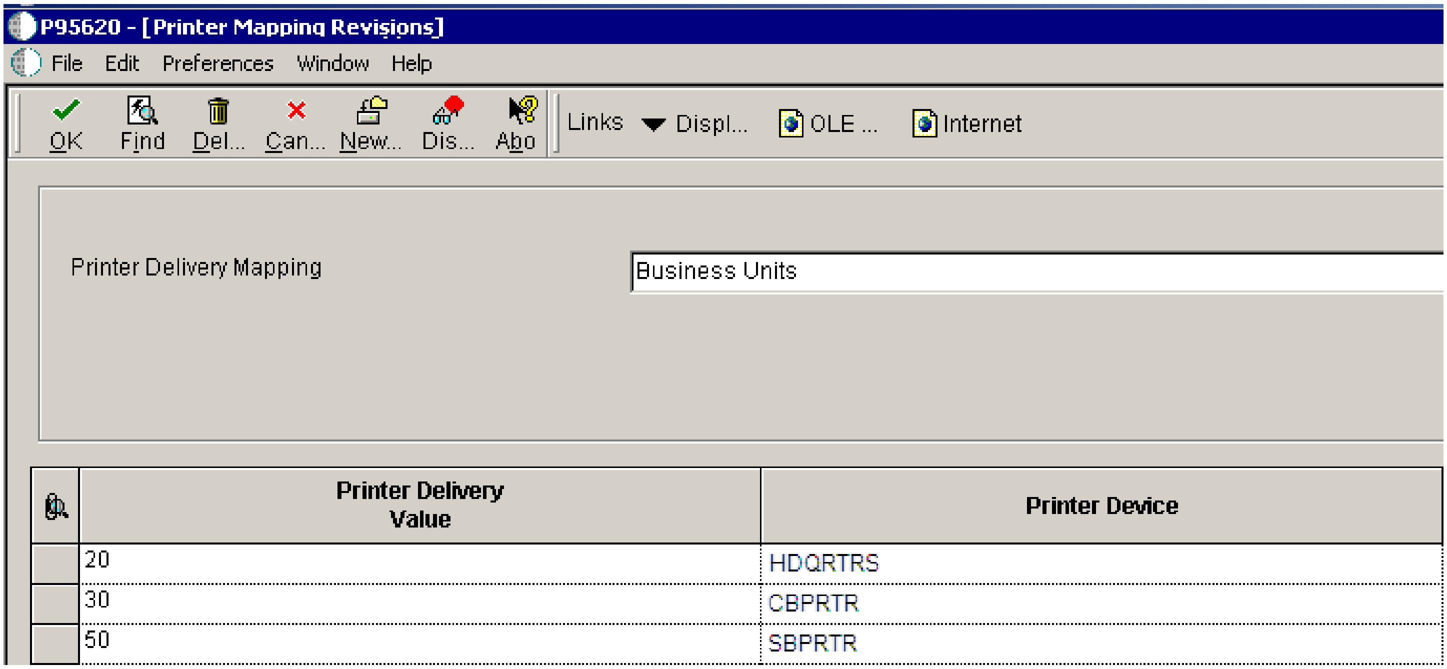
You use the Data Driven Printing form to associate a delivery tag with a printer delivery mapping name, as shown in the following example:


P95620 - [Data Driven Printing]
File Edit Preferences Form Window Help
OK Find Del... Can... New... Dis... Abo
Links Add P... OLE ... Internet

Report Definition RD983051NA

Delivery Tag	Data Delivery Mapping Name
CostCenter_ID1	Business Units
DepartmentCode_ID1	Departments

After the delivery tag has been assigned to a data delivery mapping name, you can associate printer delivery values and their respective printers to the data delivery mapping name. To do this, select Add Printer Mapping from the Form menu on the Data Driven Printing form to access the Printer Mapping Revisions form. You can enter a printer in the Printer Device column, or you can use the visual assist to select a printer name.



Printer Delivery Mapping: Business Units

Printer Delivery Value	Printer Device
20	HDQRTRS
30	CBPRTR
50	SBPRTR

You can also add, modify, or delete report definition mapping records in the Report Definition Printer Mappings application (P95621).

Printer Delivery Tables

The Report Definition Printer Mappings application (P95621) uses two tables for the printer mapping records. The tables define the relationship between the data and the printer devices.

- XMLP Report Definition Data Map (F95627)

  The F95627 table contains the XPath field from the report definition source XML and the printer mapping name for each record in the table. For example, the XPath field value CostCenter_ID1 could be mapped to the printer delivery mapping name *Business Units*.

Data Driven XPath	Printer Delivery Mapping Name
CostCenter_ID1	Business Units
DepartmentCode_ID1	Departments

- XMLP Report Definition Printer Map (F95628)

  The F95628 table contains the printer mapping name, printer delivery value, and printer device name. For example, the printer delivery mapping name BusinessUnits could be mapped to the values in the CostCenter_ID1 XPath, such as *30*, *20*, *50*, and so on. Each value can then be assigned to a specific printer device, as the following example illustrates.

Printer Delivery Mapping Name	Printer Delivery Value	Printer Device
Business Units	20	HDQRTRS
Business Units	30	CBPRTR
Business Units	50	SBPRTR
Departments	7392	APPRINTER
Departments	7373	POPRINTER

Forms Used to Define Printer Mappings

Form Name	FormID	Navigation	Usage
Add Report Definition-Bursting and Delivery	W95620J	When adding a new report definition, click Next on the Add Report Definitions - Languages and Output Types form. To update an existing report definition, on the Update Report Definition form, select a record in the grid, and then from the Form menu, select Burst/ Delivery.	Define printer delivery details for report definition output.
Data Driven Printing	W95620M	On the Report Definition - Bursting and Delivery form, select a record in the Data Driven Recipient grid (optional), click the Data Driven Delivery option, and then click the Data Driven Printer Mapping button.	Define a printer mapping.
Data Driven Printer Mapping Search & Select	W95621D	On the Data Driven Printing form, click the visual assist for the Printer Delivery Mapping Name column.	Select a pre-defined printer as the Printer Delivery Mapping Name.

Form Name	FormID	Navigation	Usage
Printer Mapping Revisions	W95621F	On the Report Definition - Bursting and Delivery form, click the Data Driven Printer Mapping button. Alternatively, on JD Edwards Solution Explorer EnterpriseOne Life Cycle Tools task view, select Batch Processing Setup. Report Definition Printer Mappings, or enter **P95621** in the Fast Path field. On the Work With Report Definition Printer Mappings form, click Add to add a new printer mapping or Find to display the existing printer mapping records. To modify an existing printer mapping, select it in the grid and then click Select.	Add or modify a printer mapping record.
Work With Report Definition Printer Mappings	W95621A	On JD Edwards Solution Explorer EnterpriseOne Life Cycle Tools task view, select Batch Processing Setup. Report Definition Printer Mappings, or enter **P95621** in the Fast Path field.	View report definition printer mappings. Delete report definition printer mappings.

Defining a Data Driven Printer Mapping Record

Access the Report Definition application (P95620).

1. Select a report definition in the grid, and then click the Select button.

   > **Note:** If you are adding a new report definition, you can define a data driven printer mapping record on the Bursting and Delivery form, as described below.

2. On the Update Report Definition form, from the Form menu, select Burst/Delivery.
3. (Optional) On the Bursting and Delivery form, in the Data Driven Recipient column, enter an XPath from the XML output that contains the recipient data.

   > **Note:** If there is more than one data driven recipient XPath shown in the grid, ensure that you select the correct value before you define the data driven printer record. The tag for the selected value is passed into the Data Driven Printing form. If necessary, the value passed into the form can be changed.

4. In the Delivery section, click the Data Driven Delivery checkbox.
5. Click the Data Driven Printer Mapping button.
6. On the Data Driven Printing form, enter the XPath field in the Delivery Tag column if the value was not passed in from the Bursting and Delivery form.
7. Enter a printer delivery mapping name, or use the visual assist to select a mapping name.

   If an appropriate mapping name does not exist, follow the next steps to add a new printer delivery mapping name.

8. To add a new printer delivery mapping name, select Add Printer Mapping from the Form menu.
9. On the Printer Mapping Revisions form, enter a name in the Printer Delivery Mapping field.
10. Enter a value in the Printer Delivery Value field.

    The printer delivery value must be a valid value from the XPath field that you specified as the delivery tag.

    > **Note:** The printer delivery value can be any valid business view column or variable value in the XML source that you associate with a printer device to indicate where you want the output to be sent. You must enter the value exactly as it is shown in the XML source. For example, if the XML shows 00001 for Company, you must enter 00001 for the printer delivery value, not 1. Some values might have leading spaces in the XML source, as in the following example:
    >
    > ```
    > <CostCenter_1>       50</CostCenter_1>
    > ```
    >
    > Note that there are leading spaces. When you enter *50* as the printer delivery value on the Printer Mapping Revisions form, you must include the leading spaces. However, you can remove the leading spaces in the XML source by using the **ltrim** function in Report Design Aid event rules. (To locate the ltrim function, select a value in the Expression Manager, and then expand the Text folder in the Advanced Functions list). The following example shows an assignment using ltrim to remove leading spaces from the MCU value:
    >
    > ```
    >                                                                 RV DeliveryTag
    > = ltrim[BC Business Unit (F42565)(MCU)
    > ```
    >
    > As a result, the XML source will show the business unit value without leading spaces, as shown in the following example:
    >
    > ```
    > <CostCenter_1>50</CostCenter_1>
    > ```
    >
    > In this case, you can enter the value *50* as the printer delivery value without any leading spaces.

11. Enter a printer device, or use the visual assist to select a printer.
12. Click OK to save the data driven printer mapping record and return to the Bursting and Delivery form.

Revising a Printer Mapping Record

Access the Report Definition application (P95620).

> **Note:** Alternatively, you can access the Report Definition Printer Mappings application by entering P95621 in the Fast Path field.

1. Select a report definition in the grid, and then click the Select button.
2. On the Update Report Definition form, from the Form menu, select Burst/Delivery.
3. Click the Data Driven Printer Mapping button.
4. On the Data Driven Printing form, modify the appropriate printer delivery mapping record.
5. If you need to add, modify, or delete printer delivery values, from the Form menu select Add Printer Mappings.
6. On the Printer Mapping Revisions form, enter a a name in the Printer Delivery Mapping field, and then click Find.
7. Modify the printer delivery value or printer device as necessary, and then click OK.
8. Click OK to save the data driven printer mapping record and return to the Bursting and Delivery form.

Deleting a Printer Mapping Record

Access the Report Definition application (P95620).

> **Note:** Alternatively, you can access the Report Definition Printer Mappings application by entering P95621 in the Fast Path field.

1. Select a report definition in the grid, and then click the Select button.
2. On the Update Report Definition form, from the Form menu, select Burst/Delivery.
3. Click the Data Driven Printer Mapping button.
4. On the Data Driven Printing form, delete the appropriate printer delivery mapping record.

   > **Note:** If you attempt to delete a printer mapping record that is used by a report definition, the system displays a message to advise you that the mapping record cannot be deleted.

5. Click OK to return to the Bursting and Delivery form.

Combined Data Driven Email and Data Driven Printing

To utilize both data driven email and data driven printing in a single report definition submission, you must create a customized delivery tag for either the email or printer delivery values. The customized delivery tag could be a report variable (RV) that you populate with delivery tag values using event rule logic in the Report Design Aid.

For example, if you want to send specific outputs as email to certain Address Book numbers, and specific outputs to different printers based on the data, you would populate the RV variable in the source UBE with the delivery values. In this way, the delivery tag has the necessary information to deliver the output appropriately.

Consider the following scenario: A company groups its customers by business unit and sends them statements on a monthly basis. Customers can receive an electronic statement by email or a printed statement sent by regular mail. The customer record has a field that denotes the email preference. If the email preference is not selected in the customer record or if an email address has not been specified, a printed statement is considered the preferred delivery method.

After creating a report variable for the delivery tag, the logic for this hypothetical scenario would be similar to the following:

```
If EmailAddressValue is <blank>
  RV DeliveryTag = BC Business Unit (F4201)(MCU)
Else
RV DeliveryTag = [BC Address Number (F4201)(AN8)]
End If
```

For printer delivery, the report variable would be mapped to a data delivery mapping name in the Report Definition Printer Mappings application (P95621); business unit values from the report variable would then be assigned to individual printer devices. As a result, the statements for customers who want a printed copy would be sent to the printer assigned to their business unit.

No mapping is required for email delivery, because the Address Book application (P01012) contains logic to deliver the output to the email address associated with each address number.

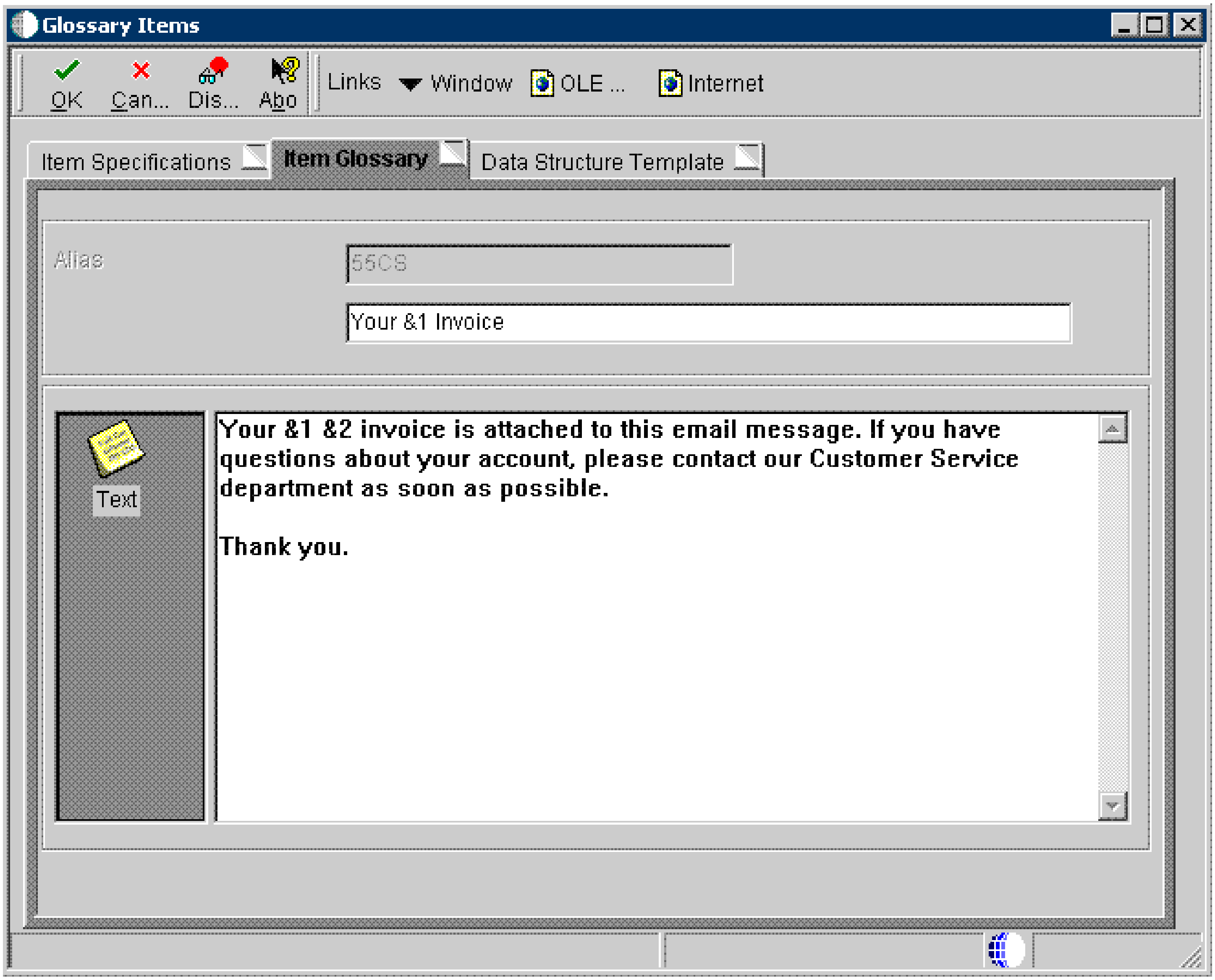
For information about using report variables and event rules in the Report Design Aid, see *"Working with Event Rules" in the JD Edwards EnterpriseOne Tools Report Design Aid Guide*

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.



Alias 55CS

Your &1 Invoice

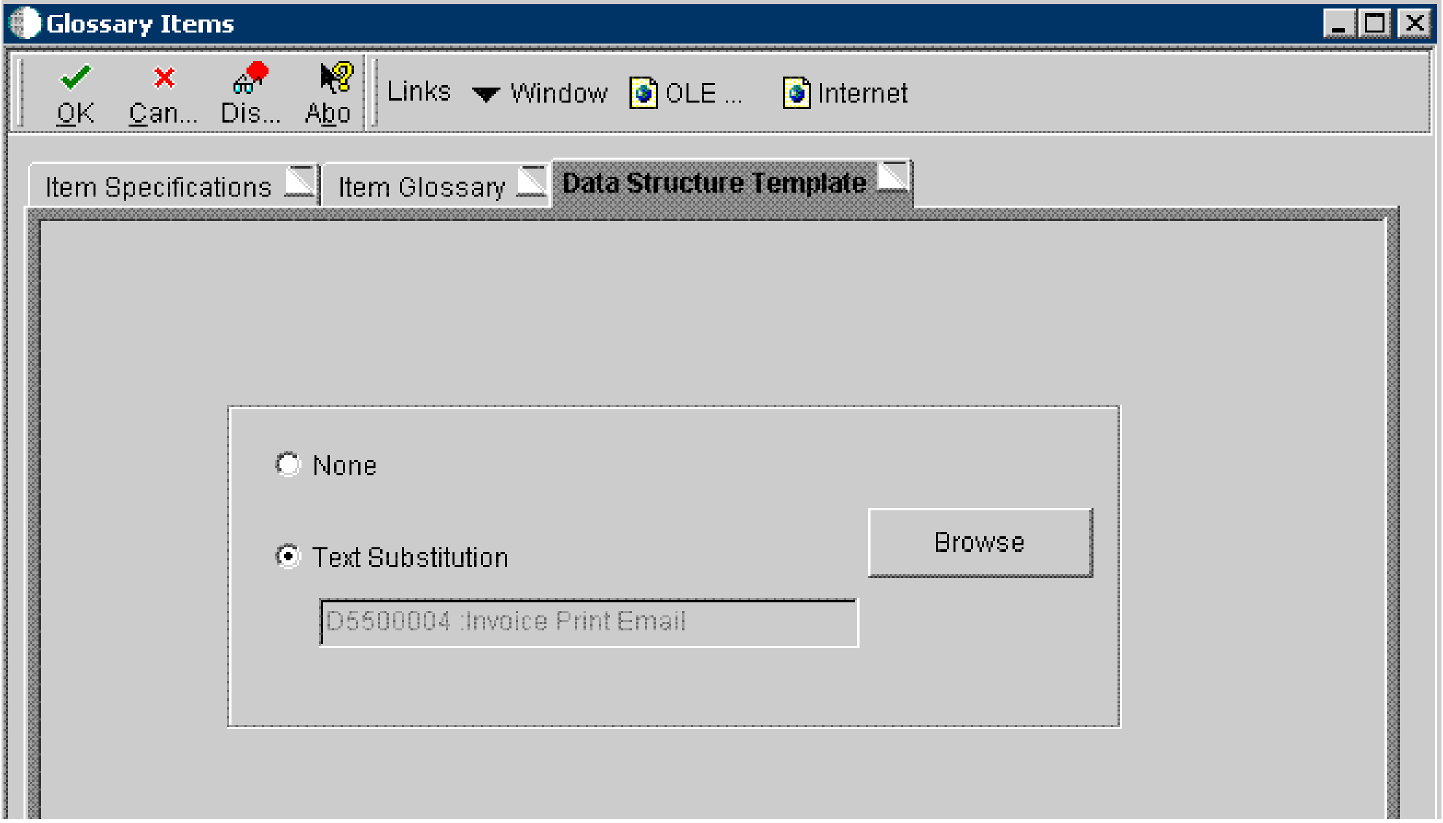
Your &1 &2 invoice is attached to this email message. If you have questions about your account, please contact our Customer Service department as soon as possible.

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.

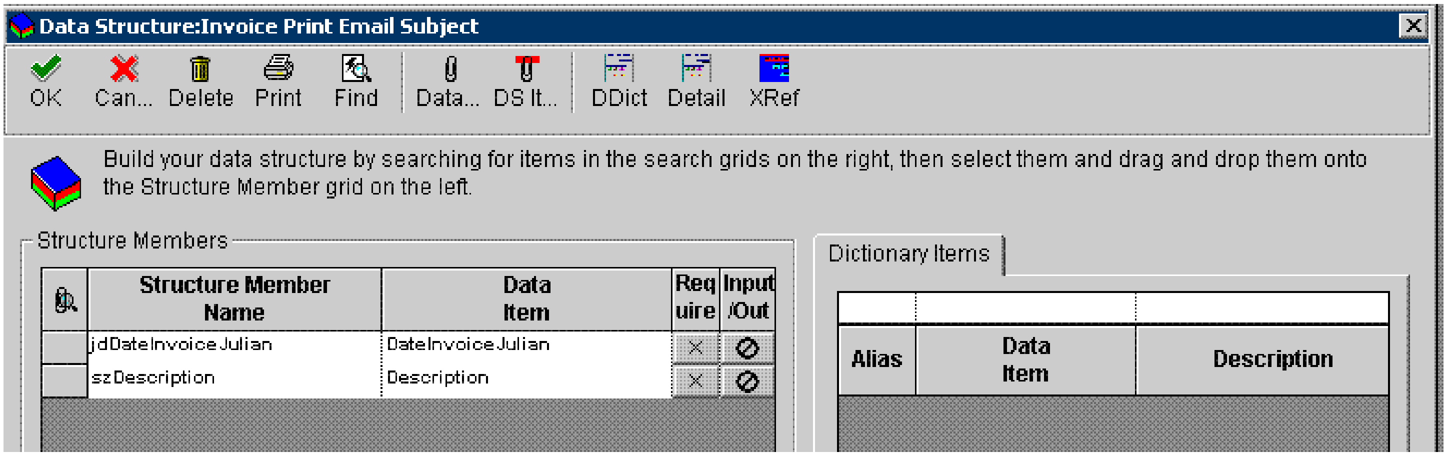


○ None

● Text Substitution

D5500004 :Invoice Print Email

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.



Build your data structure by searching for items in the search grids on the right, then select them and drag and drop them onto the Structure Member grid on the left.

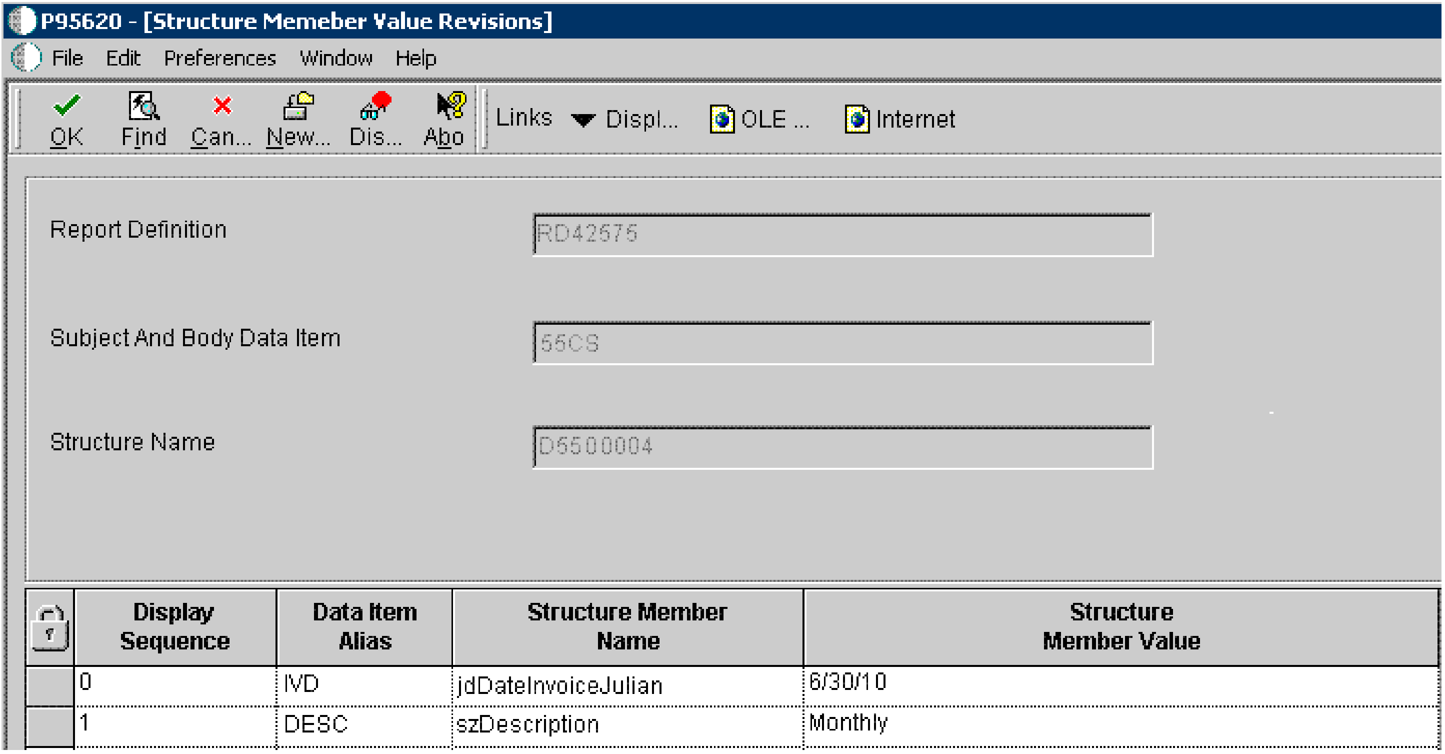
	Structure Member Name	Data Item	Require	Input /Out
	jdDateInvoiceJulian	DateInvoiceJulian	×	⊘
	szDescription	Description	×	⊘

Alias	Data Item	Description

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.


P95620 - [Structure Memeber Value Revisions]
File Edit Preferences Window Help
OK Find Can... New... Dis... Abo Links Displ... OLE ... Internet

Report Definition: RD42575

Subject And Body Data Item: 55CS

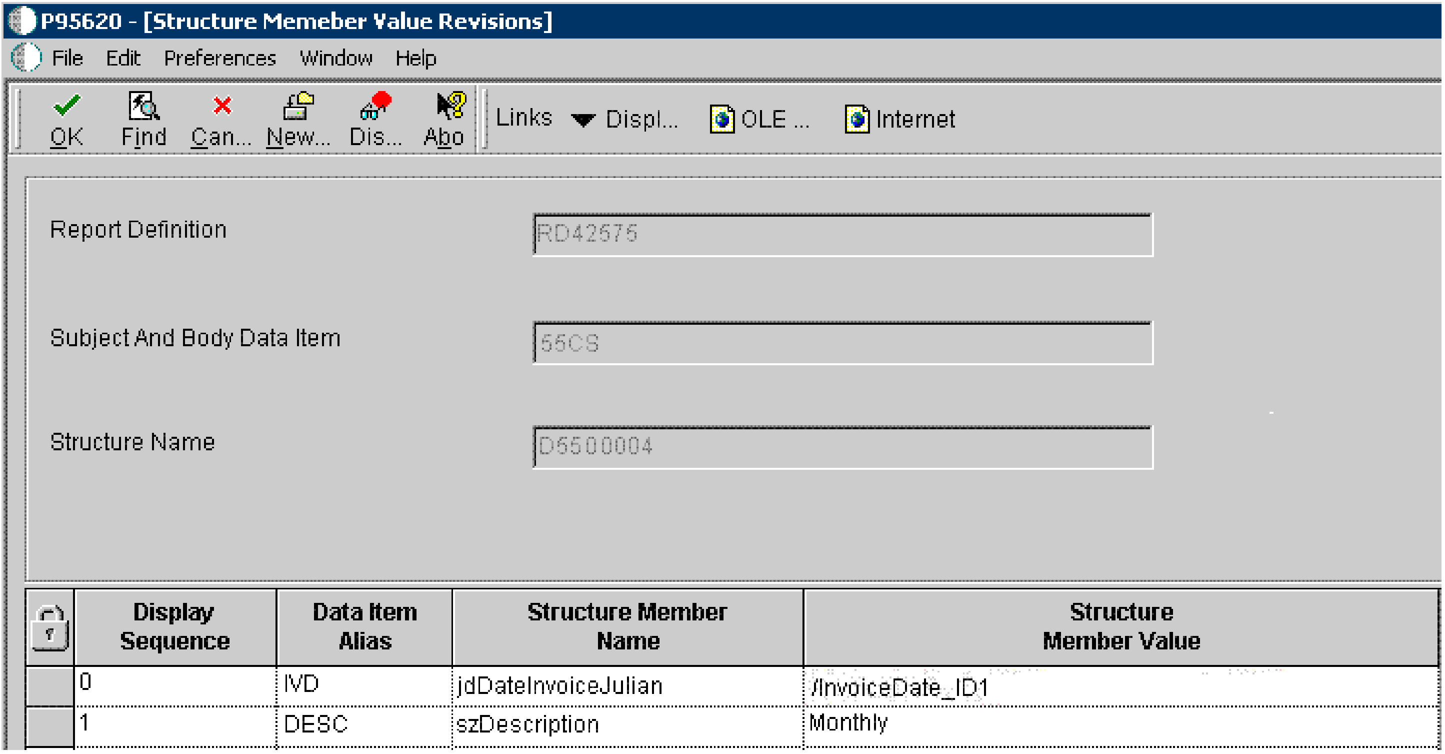
Structure Name: D5500004

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

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:



Report Definition	RD42575
Subject And Body Data Item	55CS
Structure Name	D5500004

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.

> **Note:** 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.

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.

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.
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.

Configuring Report Definition Jobs

When all the available BI Publisher Kernel(s) are busy processing the Report Definition jobs, and there is further influx of job load, the waiting jobs may end in error because the BI Publisher kernel(s) could not respond to the request in the stipulated wait time. To resolve this limitation, the system now uses a polling methodology to determine priority of jobs to be processed to ensure that no jobs end in error due to a system time out.

This section discusses how to configure report definition jobs and direct them to specific kernels or servers for processing by BI Publisher.

Configuring the Embedded BI Publisher Server Load

You can prioritize and queue BI Publisher jobs for optimum results by configuring Percentage of large and small job kernels.

The large report definition jobs will run on the large job kernels and the small report definition jobs will run on small job kernels based on the maximum size defined for the small job kernels in Embedded BI Publisher Server Load Configuration application (P95645).

However, a small report definition job can run on a large job kernel if all the small job kernels are busy and a large job kernel is available. But, a large report definition job will never run on a small job kernel.

To set up the percentage of large and small job kernels:

1. Access the Embedded BI Publisher Server Load Configuration application (P95645).
2. Click Add.
3. Enter the Server Name. Alternatively, you can use the visual assist to select the server name. The Logical Data Source field gets populated automatically.
4. Enter the value in the Percentage of Large Job Kernels field. For example, enter 70.
5. The value 30 is automatically populated in the Percentage of Small Job Kernels field.

> **Note:** The system will not allow the total value of Percentage of Large Job Kernels and Percentage of Small Job Kernels to exceed or deceed 100. For example, if you enter a value <X> in the Percentage of Large Job Kernels field, the value <100-X> is automatically populated in the Percentage of Small Job Kernels field and vice versa. If you set the Percentage of Large Job Kernels to 0, and Percentage of Small Job Kernels to 100, and run a large report definition job, the job status will remain as WX (Waiting for XML Publishing) since there are no large job kernels available for publishing. If you set the percentage of Large Job Kernels to 100, and Percentage of Small Job Kernels to 0, and run a small report definition job, the job will be completed by the large job kernel.

6. Enter the required value in the Max size for Small Job Kernels in MB field. The default value is 10 MB.
7. Click OK.

> **Note:** You must restart the server every time you modify the Embedded BI Publisher Server Load Configuration to make the changes effective.

Overriding Report Definition Load Configuration

A report definition job runs on a large or a small job kernel depending on its XML size defined in the Embedded BI Publisher Server Load Configuration application (P95645).

If for a particular business reason, the system administrator wants to override a specific job always to a large job kernel or a small job kernel, he/she can achieve that by overriding the jobs using the Override Report Definition Load Configuration application (P95646).

Access the Override Report Definition Load Configuration application (P95646).

1. Select the Report Definition you want to modify and click Select.
2. Select the Job Type to LJ for large jobs or SJ for small job from the drop-down menu.
3. Click OK.

Configuring Embedded BI Publisher Server Group

To achieve the scalability, you can define a server group with multiple servers having the same server map data source and same port number.

In a defined server group, all the BI publisher kernels running on all the hosts will use a polling process to query for the next available job once they complete the assigned BI publisher job. When a particular kernel is assigned by the system to run a particular job, the job status is updated to X (Publishing).

To configure a server group, access the Embedded BI Publisher Server Group Configuration application (P95647).

1. Click Add.
2. In the Embedded BI Publisher Server Group Configuration form, enter an appropriate name in the field Embedded BI Publisher Server Group.

   > **Note:** Make sure that the Embedded BI Publisher Server Group name is unique. If you enter an existing Embedded BI Publisher Server Group name, an error message is displayed.

3. Enter the Server Name as required. The Logical Data Source field populates automatically.

   > **Note:** Make sure that the server map data source and port number is same for all the servers.

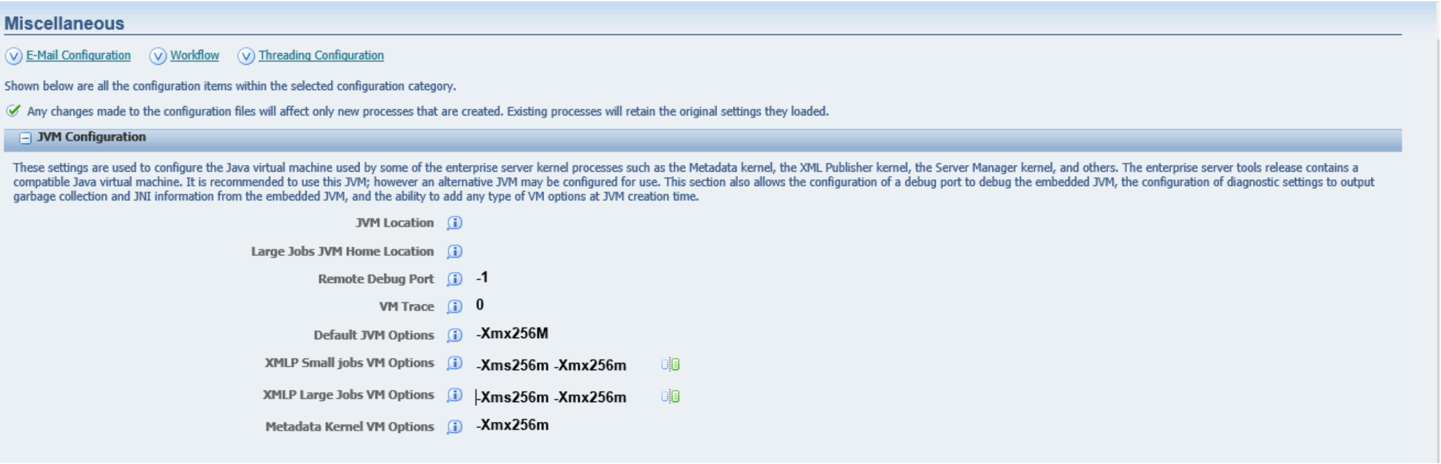
4. Click OK.

Configuring the Enterprise Server jde.ini File

You must configure the following settings in the [JDE JVM] section of the enterprise server ini file.

Setting	Value	Purpose
OutProcessJVMHome	The parameter value should point to the JAVA_HOME and not the java executable path. For example, if java home is present at C:\Program Files\Java\jre6\bin\java.exe, you should set the value as C:\Program Files\Java\jre6 path. If this path is missing , the system will use the InProcessJVMHome location.	Define the JAVA_HOME location.

The server manager screen is as shown below.



Shown below are all the configuration items within the selected configuration category.

✓ Any changes made to the configuration files will affect only new processes that are created. Existing processes will retain the original settings they loaded.

These settings are used to configure the Java virtual machine used by some of the enterprise server kernel processes such as the Metadata kernel, the XML Publisher kernel, the Server Manager kernel, and others. The enterprise server tools release contains a compatible Java virtual machine. It is recommended to use this JVM; however an alternative JVM may be configured for use. This section also allows the configuration of a debug port to debug the embedded JVM, the configuration of diagnostic settings to output garbage collection and JNI information from the embedded JVM, and the ability to add any type of VM options at JVM creation time.

JVM Location	
Large Jobs JVM Home Location	
Remote Debug Port	-1
VM Trace	0
Default JVM Options	-Xmx256M
XMLP Small jobs VM Options	-Xms256m -Xmx256m
XMLP Large Jobs VM Options	-Xms256m -Xmx256m
Metadata Kernel VM Options	-Xmx256m

XMLP Large Jobs VM options

There is a separate INI entry to adjust the JVM options of the large job kernels since, the memory requirement of large job kernels is different from the small job kernels. Initially, this may be set same as the small job kernels but, depending on the customer's large job load, they can increase the JVM memory values.

Defining Report Definition Output

This section contains the following topics:

- Customizing Report Definition Output File Name
- Defining Report Definition Output Path
- Creating Hierarchy for the Report Definition Output

Customizing the report definition output file name and the file path enables you to maintain and locate the output files easily.

For example, you can name the report definition output file as CustomerEmailAddress and specify the output file path if you using the Microsoft Windows operating system as C: /JDE/RDOutputStore. The output file with the name CustomerEmailAddress.pdf suggests that this PDF file contains email addresses of all customers.

Different paths and file names can be specified for different report definitions. This enables users to store the output on different locations.

You can also save the output files in the folder that contains the structural hierarchy.

See *Creating Hierarchy for the Report Definition Output*

Customizing Report Definition Output File Name

The customized file name is reflected in the Report Definition Output Repository (W95630E) form and Work With Submitted Jobs form of the Use Report Definition Jobs (P95630) application. If the output delivery option is selected as email, the customized output file name is reflected in the email as well.

If the archiving option is turned on, the report definition output is saved with the customized file name in the defined location.

See *Defining Report Definition Output Path*

Access the Report Definition application (P95620).

1. Select the report definition you want in the grid, and then click the Select button.
2. On the Update Report Definition form, from the Form menu, select Burst/Delivery.
3. In the Update Report Definition - Bursting and Delivery form, click the Override Output File Name tab.

> **Note:** If you do not use the Override Output File Name option, the output files will be saved with the default file name in the following format: *<RD_NAME>_<RD_LANG>_<UBE_JOBNUMBER>_<RD_JOBNUMBER>_<SEQ_NO>.<OUTPUT_TYPE>* See *BI Publisher Archived Report Files*

4. Select the option Override Output File Name. The Data Item for Output File Name field is enabled
5. In the Data Item for Output File Name field, enter the data dictionary item that has template substitution data structure defined. The Define File Name button is enabled.

   See *Delivery Email Subject and Body*
6. Click Define File Name. The Output Structure Definition Value Revisions form displays.
7. If you have several data structures defined for the data item you selected, you can enter several values in the Structure Member Value field and define the sequence in which you want the names to be displayed.
8. The Data Item Alias and Structure Member Name fields are populated automatically.
9. Enter any one of the following values or a combination of the following values in the Structure Member Value field:

   - Static customized file name - Enter any static text you want to be the name of the report definition output file. For example, test.
   - Dynamic customized file name - Enter the name of the field you want to be the name of the report definition output file. **Note.** Prefix the name of the field with '/' character. This prefix indicates that the text is an XML field.

     If the field is entered, the value of the field will be accessed from the XML source file of the report definition output and displayed as the name of the output file.

     This is helpful since you do not have to change the file name every time you run a report definition. For example, if you are running the address book number report, you can include the AddressNumber field as part of the file name. The report definition gets generated with different address number as the output file name every time you run this report definition.

     If you choose the data item with several data structures, you can define the name of the output file in a sequence.

10. Click OK.

> **Note:** If there is more than one entry in the Structure Member Value field with the same value, and if you want to archive the files in the same location, the duplicate file name might be generated based on the static and dynamic values of the Structure Member Value field. The dynamic value is taken from the source XML. If there is more than one similar static value in the Structure Member Value filed, and if duplicate files are archived at the same location, an email will be sent to the user informing that the file is archived with duplicate name when the report definition output is generated. The file is saved with the same name appended by (1) or (2) or (<any number>) depending on the number of duplications in the specified path. Consider the following scenarios:
>
> a. A static file name called Sample.pdf exists, and you want to create another output file with the same name in the same location. If you do so, an email will be sent to you informing that the file is archived with duplicate file name and the file is created with the name Sample (1).pdf in the specified path. You can access the Output Structure Definition Value Revisions form and correct the file name to avoid the duplication if you want.
>
> b. Consider the value of the Structure Member Value field is a data structure with two parameters. The value of first parameter is a static value called Sample. The value of the second parameter is a dynamic value called /AddressNumber.
>
>   Now, the file generated for each output will be: Sample_[AN8 from xml].pdf

Defining Report Definition Output Path

See *Configuring the Enterprise Server jde.ini File*

If you use the enterprise server jde.ini to define the path for archiving, all the report definitions will be saved in same path defined in enterprise server jde.ini file.

You can choose to define a customized output path for a specific report definition output using the Report Definition Server Mapping application (P95641).

> **Note:** If the path is not defined for a report definition in P95641, the output file is stored in the default path defined in the enterprise server jde.ini.

To define output path for the report definition, access the Report Definition Server Mapping application (P95641).

1. Click Add.
2. On the Report Definition Archive Paths form, complete the following fields:

Field	Description
Report Definition	Enter the report definition for which you want to define the output path. You can use the visual assist to select a report definition.
Server Name	Enter the name of the server.
User/ Role	Enter user/role.

Field	Description
Path	Enter the path where you want the output file to be saved. Note. If you enter a folder name in the path that does not exist, the folder structure is created and the output is stored at the specified path. For example, if you enter C: /JDE/RDOutputStore in the Path field and if the folder RDOutputStore does not exists, then the folder with the name RDOutputStore is created and the output file is stored in the path specified. If you enter an invalid path, the report definition job will end in FD status. You can access P95630 application and select View Delivery option and then, select Re-Deliver Output option to correct the path and save the file again.
Object Status	Select the following status as required. ○ AV: If the object status is active, the report definition output is saved in the path defined above. ○ NA: If the object status is not active, the report definition output is saved in the path defined in the enterprise server jde.ini.
Status Description	Status description populates automatically to Active if the object status is AV and Not Active if the Object status is NA.

3. Click OK.

Creating Hierarchy for the Report Definition Output

You can choose to create hierarchy of folders with report definition name, report definition language and report definition job number for the report definition output.

To create the report definition output in folder hierarchy:

1. Access Embedded BI Publisher Configuration application (P95642).
2. Select the Archive Report Definition in Hierarchical Structure checkbox.

   If Archive Report Definition in Hierarchical Structure is selected, the output file is saved in hierarchical structure in the path defined in Report Definition Server Mapping application (P95641). If the path is not defined in Report Definition Server Mapping application (P95641), the output file is archived in hierarchical structure in the path defined in enterprise server jde.ini.
3. Click Save.

Note: *Click here to view a recording of this feature.*

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.

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.

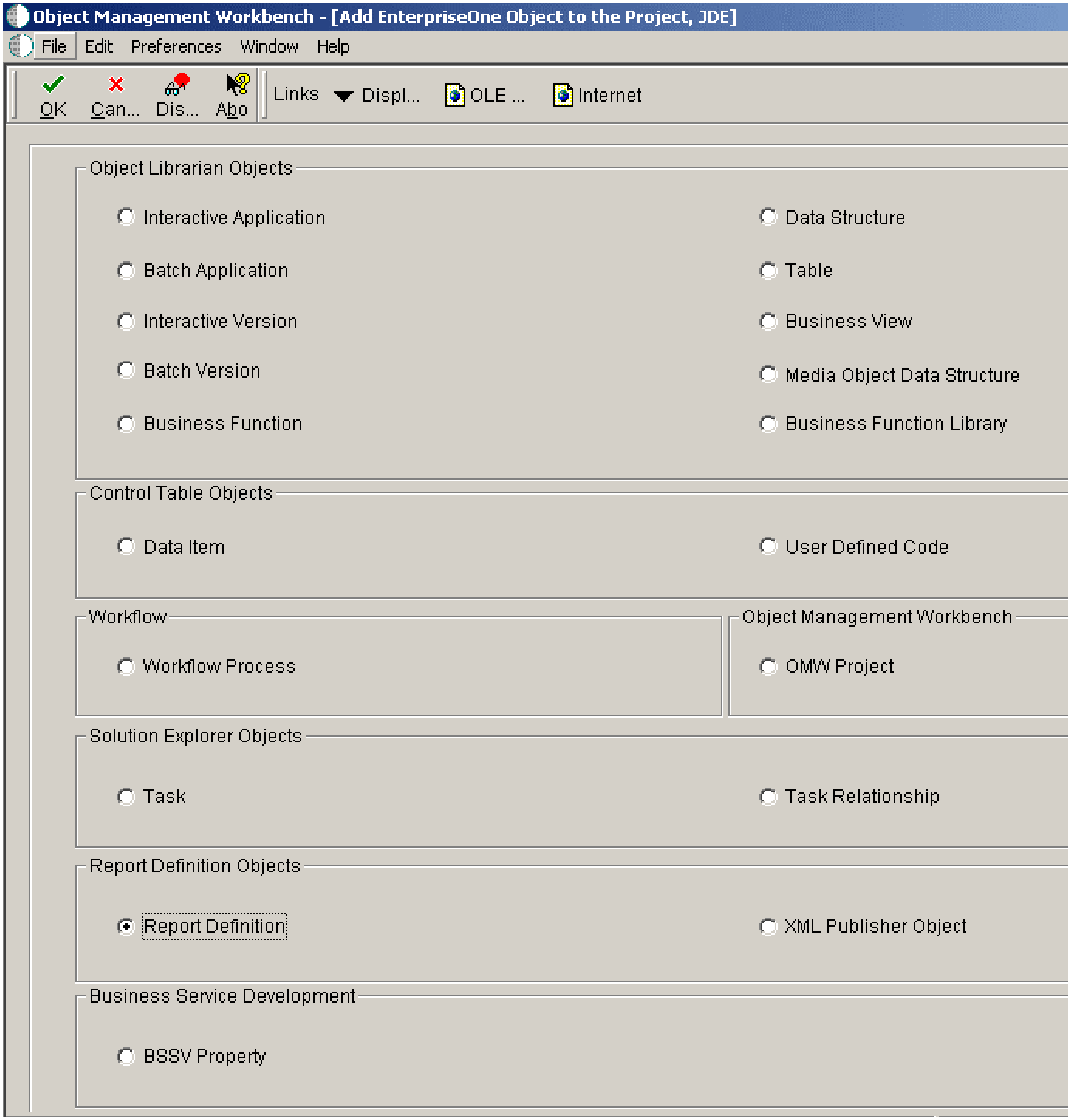
To copy a report definition:

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.

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:


Object Management Workbench - [Add EnterpriseOne Object to the Project, JDE]
File
Edit
Preferences
Window
Help
OK
Can...
Dis...
Abo
Links
Displ...
OLE ...
Internet

Object Librarian Objects

○ Interactive Application
○ Batch Application
○ Interactive Version
○ Batch Version
○ Business Function
○ Data Structure
○ Table
○ Business View
○ Media Object Data Structure
○ Business Function Library

Control Table Objects

○ Data Item
○ User Defined Code

Workflow

○ Workflow Process

Object Management Workbench

○ OMW Project

Solution Explorer Objects

○ Task
○ Task Relationship

Report Definition Objects

● Report Definition
○ XML Publisher Object

Business Service Development

○ BSSV Property

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* .

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**.

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* .

5 Submitting JD Edwards EnterpriseOne Report Definitions to BI Publisher

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.

For a report definition job that is set up with printer delivery, you can choose various print options at runtime, such as portrait, landscape, simplex, duplex, tumble, number of copies, and so on.

If blind submission is set for a report definition job, print options can be set beforehand for the default printer in the Printers application (P98616).

See "Selecting a Print Style Option" in the *JD Edwards EnterpriseOne Tools Report Printing Administration Technologies Guide* .

Submitting Report Definitions from BI Publisher Report Definitions (P95620)

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

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 type, delivery, and language options as needed.
4. If the report definition output is sent to a printer (and the report definition is not defined as blind submission), the Printer selection form appears.

   Change the printer selection and print options (such as simplex, duplex, tumble) as needed.
5. Click OK to submit the report definition to BI Publisher.

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 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.

By associating a report definition with a batch version, if the output is sent to a printer you can choose various print options at runtime, such as portrait, landscape, simplex, duplex, tumble, number of copies, and so on.

See "Selecting a Print Style Option" in the *JD Edwards EnterpriseOne Tools Report Printing Administration Technologies Guide* .

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 *Templates*

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 Report Design Aid Guide* .

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.

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.
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.

Submitting Batch Versions

Access the Work With Batch Versions - Available Versions form.

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.
4. If the report definition output is sent to a printer (and the report definition is not defined as blind submission), the Printer selection form appears.

   Change the printer selection and print options (such as simplex, duplex, or tumble) as needed.
5. Click OK to submit the report definition to BI Publisher.

Accessing Batch Version Advanced Option Overrides

Access the Advanced Version Prompting form.

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.

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)

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

Job Status Codes	Definition
FX	BI Publishing Error
WX	Waiting for XML Publishing

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).

Forms Used to View Report Definition Submission Details

Form Name	FormID	Navigation	Usage
Report Definition Job Control Search	W95630B	From the Navigator 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.
Submitted Job Search	W986110B	Select a record, then from the Row menu select View RD Jobs.	Select a report definition to view submission details.

Viewing Submission Details from Report Definition Jobs

Access the Report Definition Job Control Search form.

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 *Managing JD Edwards EnterpriseOne Report Definition Output*.

Viewing Submission Details from Work With Submitted Jobs (P986110B)

Access the Submitted Job Search form.

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 *Managing JD Edwards EnterpriseOne Report Definition Output*

6 Managing JD Edwards EnterpriseOne Report Definition Output

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.

Viewing Report Definition Output, Source, and Delivery Details

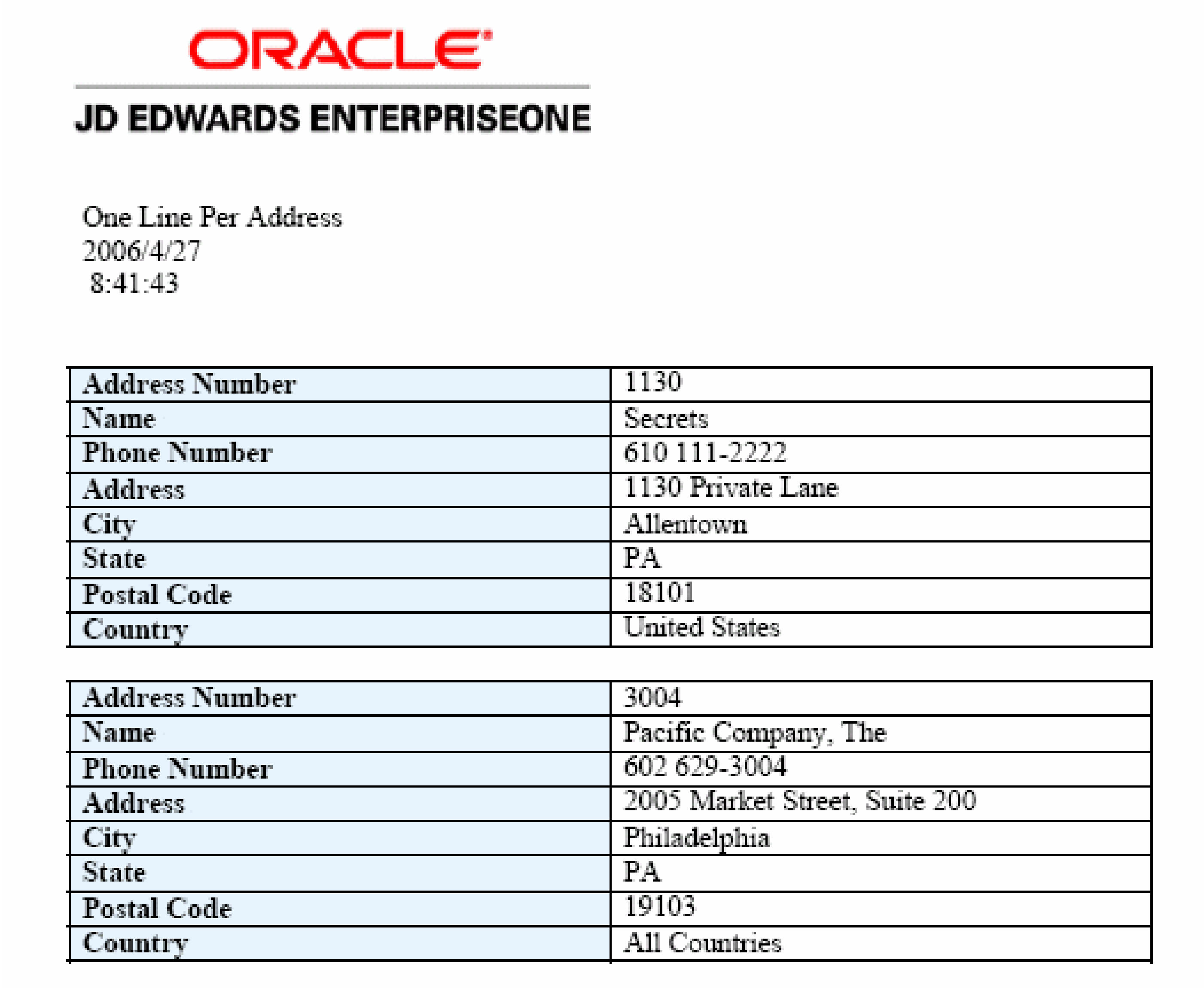
Selecting a report definition job on the Report Definition Job Control Search form provides access to the output repository for the report definition job. The Report Definition Output Repository form displays details about the output types that were created for the report definition. These details include 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.

Example of Report Definition Output

This is an example of report definition output.



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

Forms Used to View Report Definition Output

Form Name	FormID	Navigation	Usage
Report Definition Job Control Search	W95630B	From the Navigator 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.
Submitted Job Search	W986110B	Select a record, and then from the Row menu select View RD Jobs.	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.

Viewing Report Definition Source

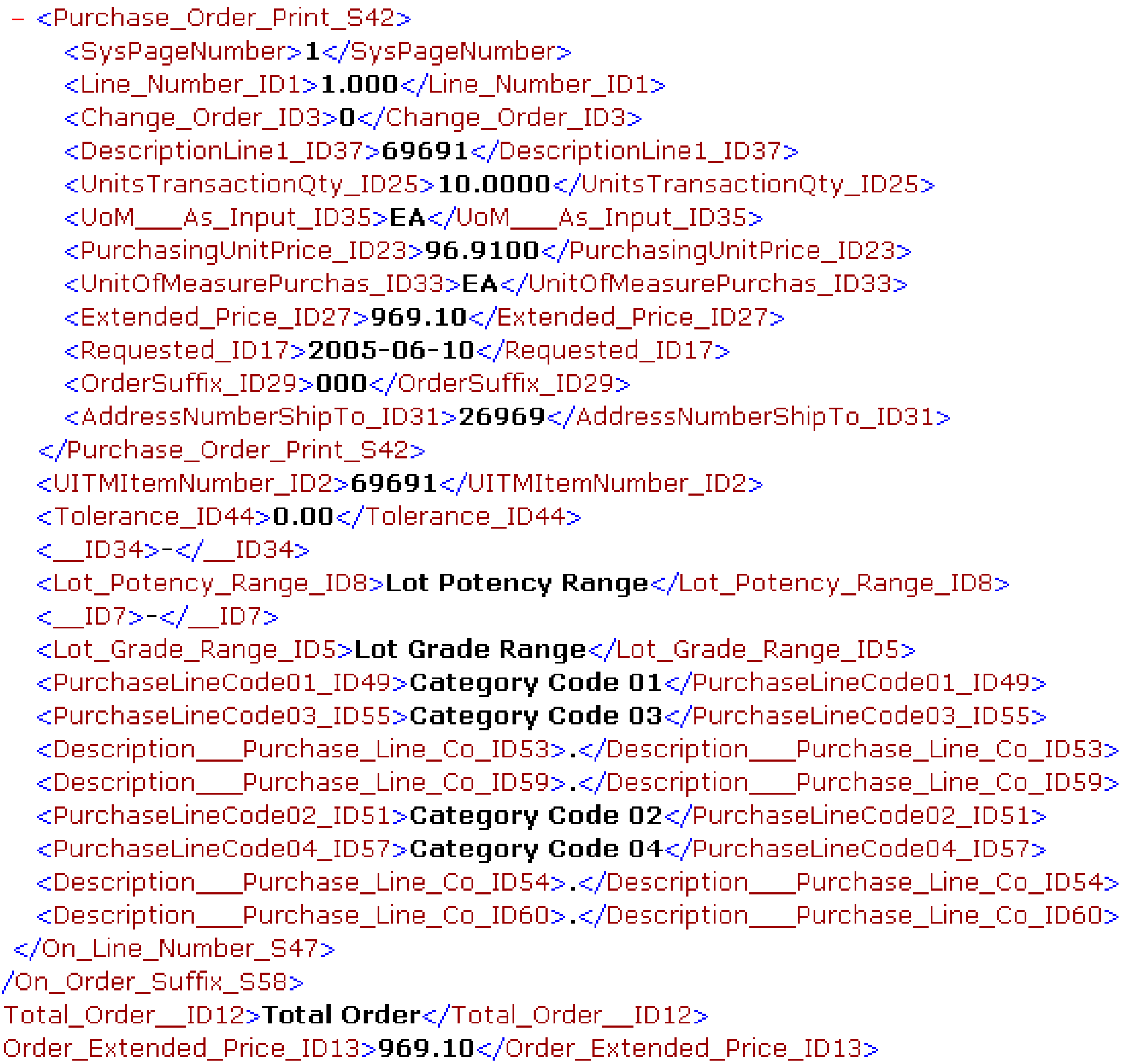
The report definition source is the XML output that the batch version (source UBE) created when the report definition was submitted.

Access the Report Definition Job Control Search form.

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

Example: Report Definition Source

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



```
  - <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>
```

Viewing Report Definition Output Details

Access the Report Definition Output Repository form.

1. Select a report definition output record.

2. Review the output details in the grid.

Report Definition Name

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

Job #

The number assigned to the job.

Report Definition #

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

Object Language

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

User

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

Machine Key Submitted

The name of the workstation that submitted the report definition.

Date/Time Job Submitted

The date and time that the report definition was submitted.

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.

Output Bursted

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

UBE Host

The name of the server that processed the batch version.

Report Definition Host

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

Report

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

Version

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

Report Definition

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

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.

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.

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. Click the View Output icon, or from the Row menu select View Output.

Viewing Report Definition Output Delivery Details

Access Report Definition Output Delivery Details.

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, RTF, and XML.

Delivery Type
The type of delivery that was performed for a report definition (UDC H95 |RD). Valid values are:

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

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

E: The email address provided.

F: The override for the From email address.

P The static printer for the output.

Z: The data-driven printer for the output.

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.

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. Click the Print icon, or from the Row menu select Print Output.

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.

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)

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

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

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.

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.

You can archive the report definition output files in hierarchical folder structure using the Embedded BI Publisher Configuration application (P95642).

See *Creating Hierarchy for the Report Definition Output*

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.

[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.

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 Server Manager Guide* .

BI Publisher Archived Report Files

The archived report output files follow a default naming convention if you have not customized the output file name using the Update Report Definition - Bursting and Delivery application.

If you have customized the output file name using the Update Report Definition - Bursting and Delivery application, the output files are saved with the customized file names in the archived folder.

See 4.2.9.1 Customizing Report Definition Output File Name

The default 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: RDT13546_EN_3657_3759_1.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

Segment	Description
.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.

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 *Archiving BI Publisher Report Output*

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?"

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?"

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

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.

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.

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:

a. 1 = Proof Mode
b. 2 = Final Mode

4. Delete Mode

Specify the records to delete. Valid values are:

a. 1 = Delete both UBE and Report Definition records.
b. 2 = Delete UBE records only.
c. 3 = Delete Report Definition records only.

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).

7 Understanding the XML Output

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.

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.

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.

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.

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.

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.

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.

Formatted Dates

Dates are formatted in Canonical format.

Page Headers and Page Footers

Information that is contained in page headers and page footers of RDA reports is grouped together at the top of the XML output. This ensures that the headers and footers do not break up the report data.

Design your reports in RDA without including page headers and page footers. You should design the page headers and page footers in the BI Publisher template.

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

```
- <PageHeaders>
- <Page_Header_S43>
     <PageHeaderNumber>1</PageHeaderNumber>
  <R43500_ID1>R43500</R43500_ID1>
  <Variable_000004_ID4>2007-09-20</Variable_000004_ID4>
  <szCompanyDescription_ALPH_ID17>06969</szCompanyDescription_ALPH_ID17>
  <Page__ID7>Page</Page__ID7>
  <Variable_000006_ID6>1</Variable_000006_ID6>
   <szReportDescription_ALPH_ID18>Purchase Order Print</szReportDescription_ALPH_
    ID18>
  </Page_Header_S43>
```

There are individual page header nodes nested under the PageHeaders node, one for each page of the report where a **reprint page header** was explicitly called by the report logic. By default, 8.97 and subsequent releases will have only one page header. In this example, the PageHeaderNumber is the same as the Variable_000006_ID6. The PageHeaderNumber is the sequential page number generated by the system. The variable page number is the page number printed in the page header of the report.

Conditional Sections

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

This is an example of conditional sections that are 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>
      <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.

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.

Child Sections

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

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.

Property	Description
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.
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>
```

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

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

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 *JD Edwards EnterpriseOne BI Publisher Releases*.

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.

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.

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.

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.

Enabling the QR Code to the PDF Output (Release 9.2.6.3)

You can generate the PDF output with QR code enabled for any field in the XML file.

Mapping the QR Code in your RTF Template

You can use the QR code or the PDF417 2D barcode type in the RTF templates. When you create an RTF template, use the QR code or the PDF417 functions to specify the barcode type. These functions do not require external fonts.

Use the following QR code syntax:

```
<?qrcode: <DATA>; <SIZE>[; <CHARSET>]?>
```

In the above syntax:

- DATA – Data to be encoded in the QR code format.
- SIZE – QR code size dimension in points(pt).
- CHARSET – (Optional) Character set for encoding the data. The Default is UTF8.

Use the following PDF417 syntax:

```
<?pdf417: <DATA>[; <XSCALE>[; <COLUMNS>[; <ROWS>[; <CHARSET>]]]]?>
```

In the above syntax:

- DATA – Data to be encoded in the PDF417 format.
- XSCALE – Point(pt) per PDF417 module width. The Default is 1 (1pt per module).
- COLUMNS – The number of columns to be used in the generated PDF417 symbol. The Default value is -1.
- ROWS – The number of rows to be used in the generated PDF417 symbol. The Default value is -1.
- CHARSET – (Optional) Character set to be used to encode data with the Byte compaction mode. Specify CHARSET only if the data contains non-Latin-1(ISO-8859-1) characters.

You can enable the QR code in an RTF template for any field from the XML file. Make sure that you have installed BI Publisher as a plug-in for Microsoft Word. From the BI Publisher menu, click Sample XML to load your XML file into the RTF template and right-click the field for which you want to enable the QR code. Select the Properties tab and click the Advanced tab. Here, you can add the QR code command for the field by using the QR code syntax explained earlier. Click OK.

Generating QR-Coded PDF Output

To generate the QR-coded PDF output:

- Use the fast path to access BI Publisher Object Repository (P95600) and click Add to upload your QR-Coded RTF template into the repository.
- Use the fast path to access BI Publisher Report Definitions (P95620) and create a report definition by choosing the uploaded RTF template and the respective UBE|Version.
- Submit the report definition.
- After the report definition job is complete, download the PDF output and you can see the QR code for the designated fields.

8 Creating Oracle BI Publisher Reports with JD Edwards Data Access Driver

JD Edwards Data Access Reporting with Oracle BI Publisher 11g Overview

JD Edwards EnterpriseOne Data Access Driver enables you to use Oracle Business Intelligence (BI) Publisher Enterprise (OBIEE-BIP) to produce BI Publisher reports using JD Edwards data.

This chapter provides an overview of installing and configuring Oracle Business Intelligence (BI) Publisher 11g and JD Edwards Data Access Driver. The process to implement JD Edwards EnterpriseOne reporting using the Data Access Driver is as follows:

- Install Oracle BI Publisher.
- Install the JD Edwards EnterpriseOne Data Access Driver.

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

- Configure a JDBC Driver in Oracle BI Publisher.

Installing Oracle BI Publisher

To use the Data Access Driver with Oracle BI Publisher you only need to install Oracle BI Publisher. You do not need to install Oracle Business Intelligence Enterprise Edition or Oracle Real-Time Decisions.

Because BI Publisher installs WebLogic, you do not need to install it beforehand. The default domain in WebLogic functions for BI Publisher, so you do not need to create an additional domain unless you prefer to keep the domains separate.

Prerequisites

Before you complete the tasks in this section, you must:

- Install and configure a supported database.

  For the latest information about supported databases, visit the Oracle Fusion Middleware Certification document at:

  http://www.oracle.com/technetwork/middleware/ias/downloads/fusion-certification-100350.html

  Look for product area "Oracle Fusion Middleware 11g Release 1 Certifications" and then access "System Requirements and Supported Platforms for:

  - Oracle Business Intelligence Suite Enterprise Edition 11gR1 (11.1.1.3.0-11.1.1.6.0) (xls).
  - Oracle Business Intelligence Suite Enterprise Edition 11gR1 (11.1.1.9.0) (xls).

- Download the Oracle Repository Utility (RCU) from Oracle Technology Network (OTN), Oracle Business Intelligence (11.1.1.x) Downloads page:

  https://www.oracle.com/middleware/technologies/bi-enterprise-edition-downloads.html

  The RCU version must match the BI Publisher version.

- Download Oracle BI Publisher from the Oracle Software Delivery:

  https://www.edelivery.oracle.com

  Select Oracle Business Intelligence for the Product Pack field and choose the appropriate platform.

- Install JD Edwards EnterpriseOne Applications Release 9.1.

Installing Oracle BI Publisher

If you do not plan to change any of the default settings, such as ports and security logins, then follow the *Oracle® Fusion Middleware Quick Installation Guide for Oracle Business Intelligence* to perform the installation. If you do plan to change some of the default settings, then follow the *Installation Guide for Oracle Business Intelligence. .* Please note that although the Oracle BI installation package includes the following products, with all the technology required to install, configure, and run them, you only need to install Oracle Business Intelligence Publisher:

- Oracle Business Intelligence Publisher
- Oracle Business Intelligence Enterprise Edition (Oracle BI Answers, Oracle BI Interactive Dashboards, Oracle BI Delivers, Oracle BI Administration Tool, Oracle BI Add-in for Microsoft Office, and Oracle BI Publisher) (optional)
- Oracle Real-Time Decisions (optional)
- Essbase Suite (optional)

Oracle BI Publisher 11g (11.1.1.7) is packaged with WebLogic Server 10.3.5.0 and Oracle JDK 1.6.0_35.

Oracle BI Publisher 11g (11.1.1.9) is packaged with WebLogic Server 10.3.6.0 and Oracle JDK 1.7.0_80.

> **Note:** For BI Publisher reporting using the Data Access Driver, you only need to install Oracle BI Publisher.

The following steps provide brief guidance for the installation. Review these steps first and then use the *Oracle® Fusion Middleware Quick Installation Guide for Oracle Business Intelligence* , or if you plan to change some of the default settings, refer to the *Installation Guide for Oracle Business Intelligence.* for more detailed steps.

> **Note:** A new installation of BI Publisher is recommended even if you are currently on a prior release. This allows you to do some parallel testing before switching to the latest release.

To install Oracle BI Publisher:

1. Launch the Oracle Repository Utility (RCU) and create the BI Publisher Schemas. (BIPLATFORM and MDS).

   For more information, see the *Installation Guide for Oracle Business Intelligence.* guide.
2. Launch the Oracle Business Intelligence Installer after the schemas are created.

   > **CAUTION:**
   > - Do not install the software using a user ID that contains a special character.
   > - BI Publisher and the HTML server must be within the same firewall in order to have two-way web service and http communication.
   > - If you have anti-virus software running on your installing machine, the installation process may take longer.
3. Select the appropriate installation type:

   - "Simple Install" installs Oracle Business Intelligence components with the default settings on a single computer in the minimum number of steps. The Managed Server is not installed. This installation is targeted for demonstration, evaluation and proof of concept use cases.
   - "Enterprise Install" installs Oracle Business Intelligence components that you select and associates them with the Middleware, Oracle data source, and WebLogic Server homes that you specify. The enterprise installation type creates a separate Managed Server, bi_server1, along with the Admin Server. You should also use this installation type if you have a previous version of BI Publisher installed.

     This installation type is recommended for a production environment.
   - "Software Only Install" installs software binary files in an existing Middleware home for later configuration.
4. When the Oracle Business Intelligence Component Installation screen appears, select only the Business Intelligence Publisher option:

   > **Note:** For detailed instructions on how to install Oracle BI Publisher, see the *Oracle® Fusion Middleware Quick Installation Guide for Oracle Business Intelligence .*
5. When the installation and configuration finish, sign onto the WebLogic Admin Console and BI Server to make sure the servers are up and running by using the following URLs:

   http://host:port/console (WebLogic Admin Console)

   http://host:9704/xmlpserver (BI Publisher login page, port 9704 is the default port for "Enterprise Install" type)
6. Review the Certifications for any updates or additional requirements.

Installing the JD Edwards EnterpriseOne Data Access Driver

The Data Access Driver can be downloaded from the JD Edwards EnterpriseOne and World *Update Center*.

To install the JD Edwards EnterpriseOne Data Access Driver:

1. Unzip the tools-specific Data Access Driver par file archive (for example, 9.1.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_BIP_home>/user_projects/domains/<bifoundation_domain>/lib folder:
    - avalon-framework-cvs-20020806.jar
    - castor.jar (releases prior to 9.2.6 only)
    - commons-codec.jar
    - commons-codec-2.6.jar
    - commons-logging.jar
    - e1dadriver.jar
    - fop.jar
    - httpclient.jar
    - httpcore.jar
    - httpmime.jar
    - images.jar
    - j2ee.jar
    - jakarta.activation.jar (Tools Releases 9.2.6 and greater)
    - jakarta.xml.bind-api.jar (Tools Releases 9.2.6 and greater)
    - jaxb-core.jar (Tools Releases 9.2.6 and greater)
    - jaxb-impl.jar (Tools Releases 9.2.6 and greater)
    - jmxremote.jar
    - jmxremote_optional.jar
    - jmxri.jar
    - OWResource.jar
    - rtf2fo.jar
    - xalan.jar
    - xerces.jar
    - xmlparserv2.jar

Replacing 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.

Updating 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).

Migrating WebLogic Users to a New Version of Oracle BI Publisher

If you are upgrading your version of Oracle BI Publisher from a previous version, after the installation, you must migrate the WebLogic users from an existing version to the new server.

To migrate users to a new version of Oracle BI Publisher:

1. Export the user information from the existing BI Server.
   a. Log on to the BI Publisher WebLogic Admin Console.
   b. Select Security Realm.
   c. Select Export from the Migration tab.
   d. Enter the location for the exported files.
   e. Click Save.
2. Import the user information to the new server.
   a. Log on to the BI Publisher WebLogic Admin Console.
   b. Select Security Realm.
   c. Select Import from the Migration tab.
   d. Enter the location of the exported files.
   e. Click Save.

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:

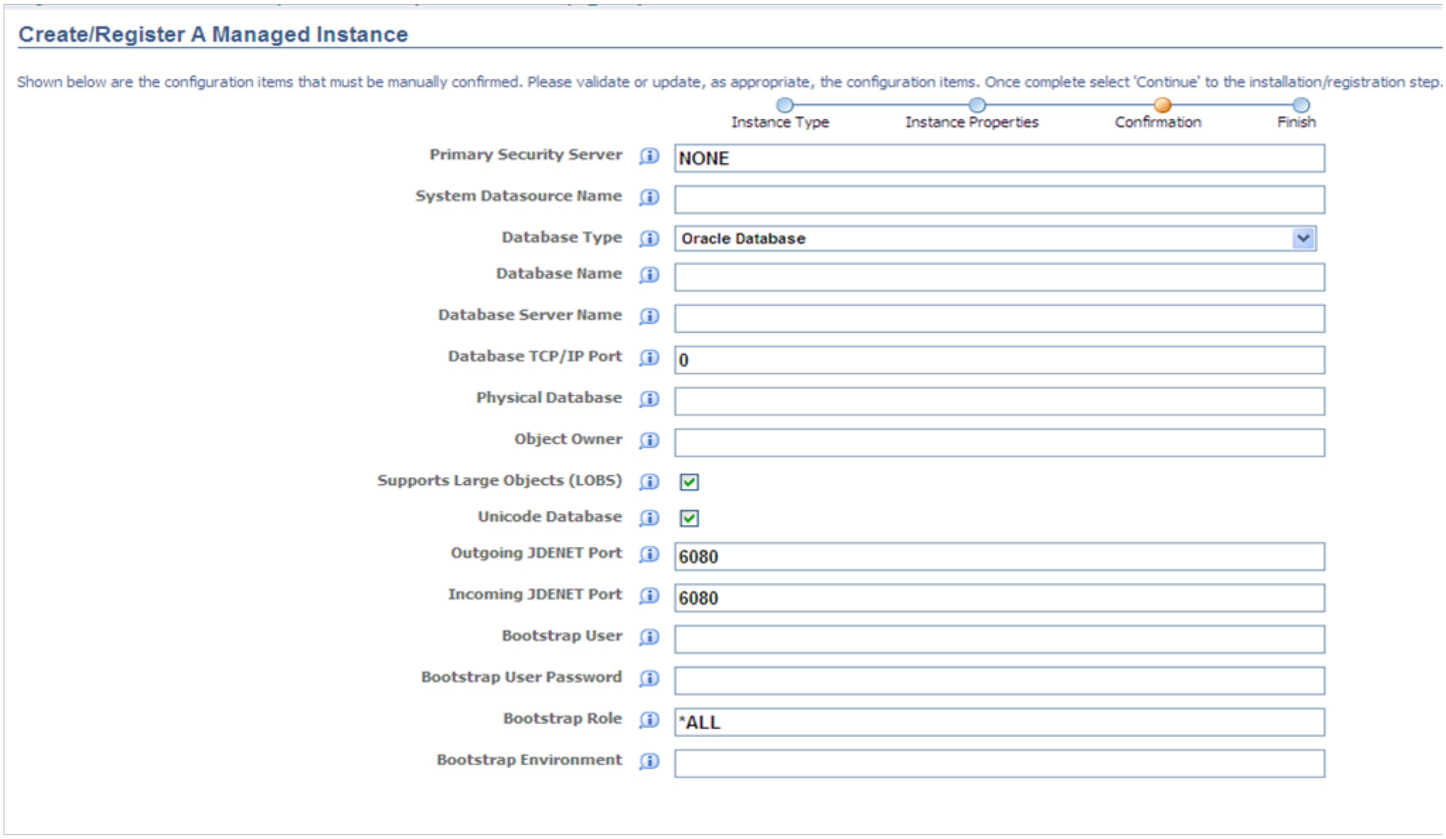
   - Install Name.
   - Install Location.

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

   > **Note:** 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.



Create/Register A Managed Instance

Shown below are the configuration items that must be manually confirmed. Please validate or update, as appropriate, the configuration items. Once complete select 'Continue' to the installation/registration step.

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

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.

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.

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.

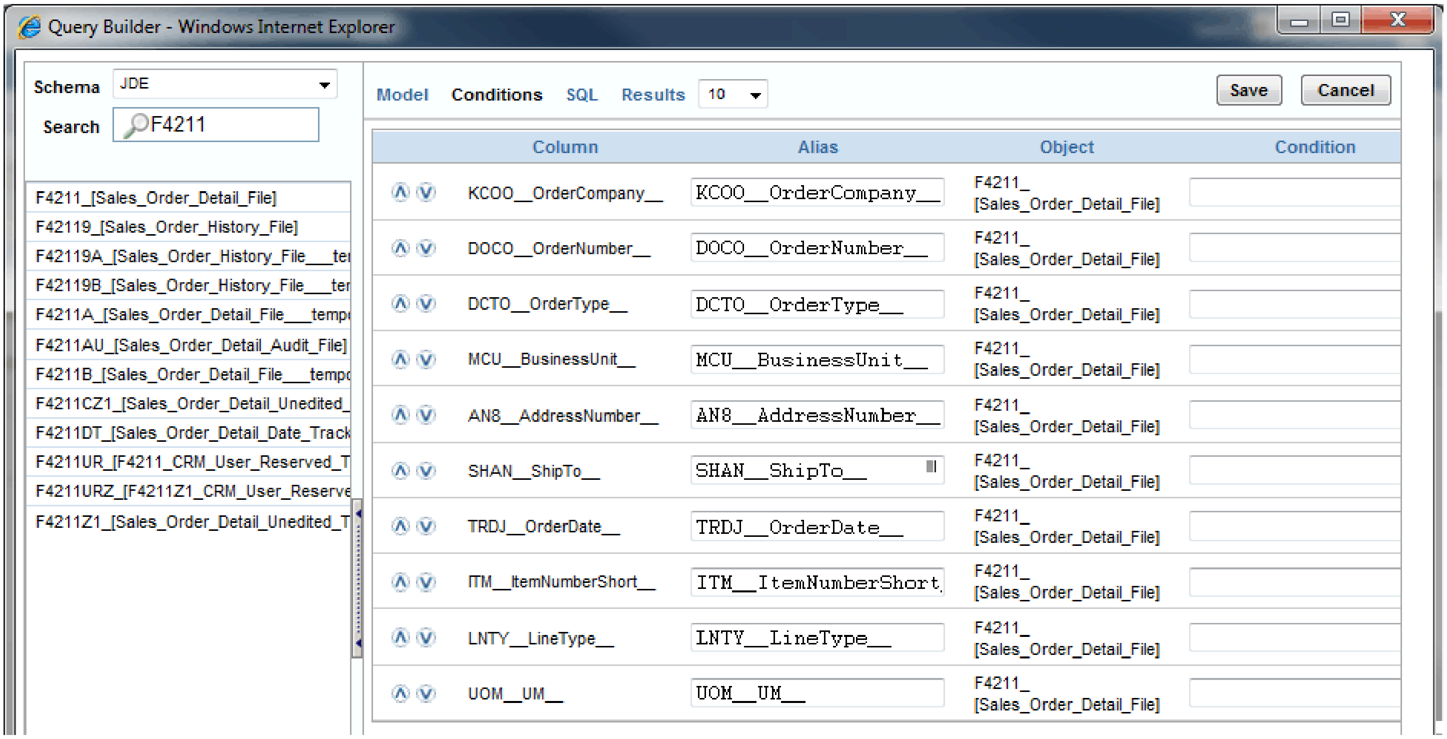
JDBC Connection String

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.



F4211

F4211_[Sales_Order_Detail_File]
F42119_[Sales_Order_History_File]
F42119A_[Sales_Order_History_File___ter
F42119B_[Sales_Order_History_File___ter
F4211A_[Sales_Order_Detail_File___temp
F4211AU_[Sales_Order_Detail_Audit_File]
F4211B_[Sales_Order_Detail_File___temp
F4211CZ1_[Sales_Order_Detail_Unedited_
F4211DT_[Sales_Order_Detail_Date_Track
F4211UR_[F4211_CRM_User_Reserved_T
F4211URZ_[F4211Z1_CRM_User_Reserve
F4211Z1_[Sales_Order_Detail_Unedited_T

Column	Alias	Object	Condition
KCOO__OrderCompany__	KCOO__OrderCompany__	F4211_ [Sales_Order_Detail_File]	
DOCO__OrderNumber__	DOCO__OrderNumber__	F4211_ [Sales_Order_Detail_File]	
DCTO__OrderType__	DCTO__OrderType__	F4211_ [Sales_Order_Detail_File]	
MCU__BusinessUnit__	MCU__BusinessUnit__	F4211_ [Sales_Order_Detail_File]	
AN8__AddressNumber__	AN8__AddressNumber__	F4211_ [Sales_Order_Detail_File]	
SHAN__ShipTo__	SHAN__ShipTo__	F4211_ [Sales_Order_Detail_File]	
TRDJ__OrderDate__	TRDJ__OrderDate__	F4211_ [Sales_Order_Detail_File]	
ITM__ItemNumberShort__	ITM__ItemNumberShort_	F4211_ [Sales_Order_Detail_File]	
LNTY__LineType__	LNTY__LineType__	F4211_ [Sales_Order_Detail_File]	
UOM__UM__	UOM__UM__	F4211_ [Sales_Order_Detail_File]	

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, *enterpriseone.role=SYSADMIN*. If the role is not specified in the connection string, the *ALL role will be used.

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.

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.

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://JDV911;TDSC=1;CDSC=1;UDSC=1;*

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.

Configuring the JDBC Driver in Oracle Business Intelligence Publisher

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: *E1_DAD*
Driver Type	*Other*
Database Driver Class	*com.jdedwards.jdbc.driver.JDBCDriver*
Connection String	*jdbc:oracle:enterpriseone://<JD Edwards environment>;* Example: *jdbc: oracle:enterpriseone://JDV911;* **Note**: 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: *enterpriseone.role = *ALL*. 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. **Note**: You can append information to the end of the connection string to display table, column, or UDC descriptions if desired. *TDSC=1* displays the table name description, *CDSC=1* displays the column description, and *UDSC=1* displays the UDC description.

Field	Value

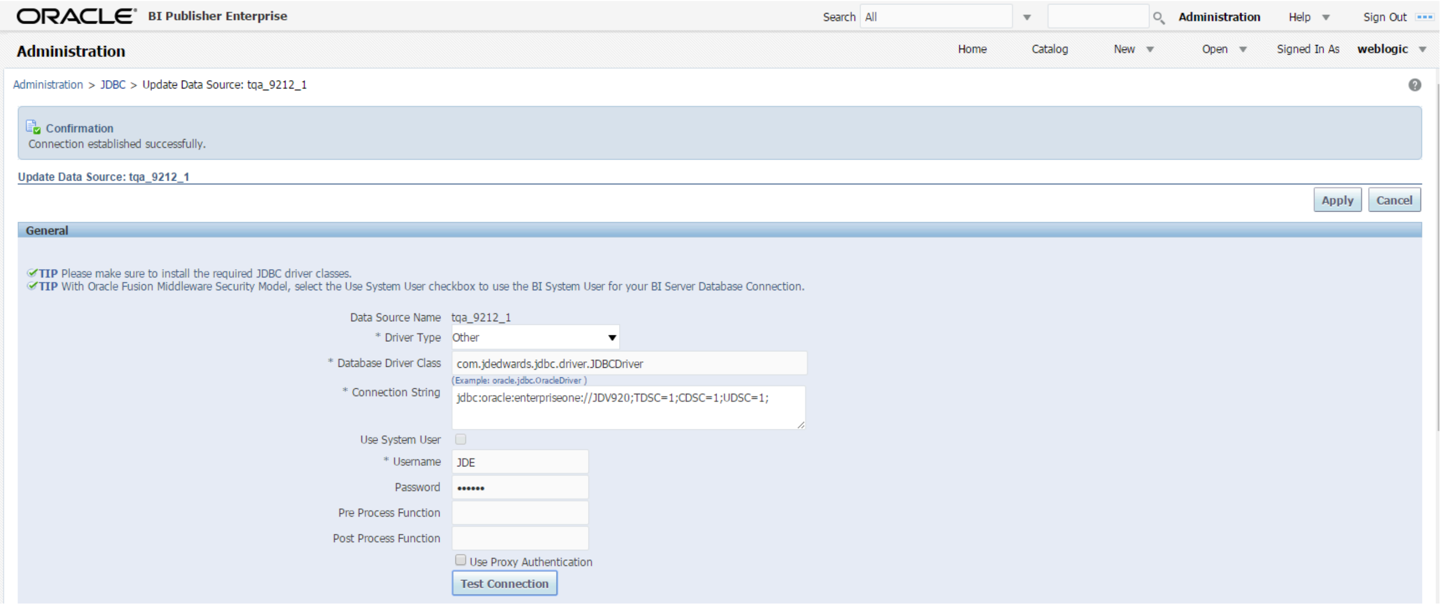
	Example of connection string without display options:
	jdbc:oracle:enterpriseone://JDV911;
	Example of connection string with display options:
	jdbc:oracle:enterpriseone://JDV911;TDSC=1;CDSC=1;UDSC=1;

User Name	Enter the JD Edwards EnterpriseOne user ID that you will connect to the database with. **Note:** The user name is case-sensitive.
Password	Enter the password for the user ID. **Note:** The password is case-sensitive.
Use Proxy Authentication	*Selected* or *Cleared*

Field	Value

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:



Confirmation
Connection established successfully.

Update Data Source: tqa_9212_1

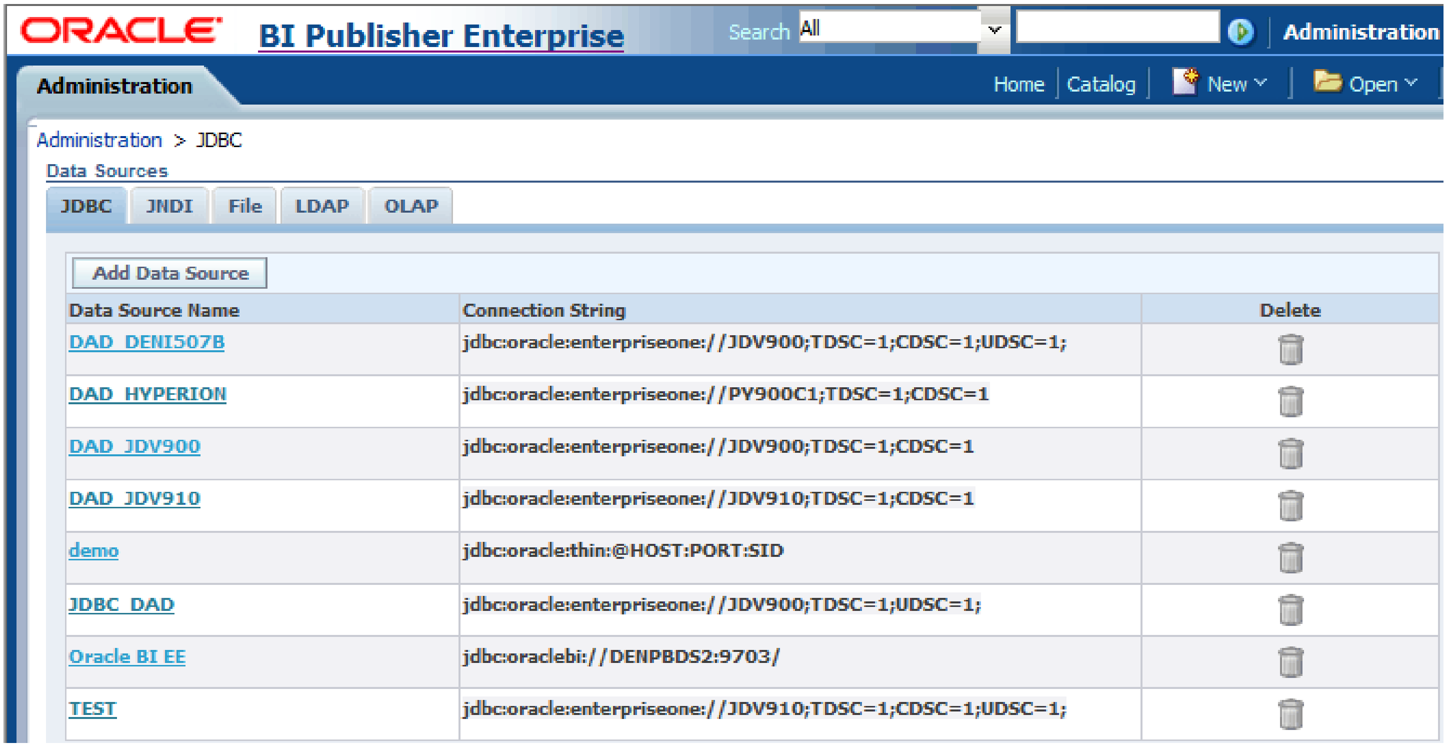
TIP Please make sure to install the required JDBC driver classes.
TIP With Oracle Fusion Middleware Security Model, select the Use System User checkbox to use the BI System User for your BI Server Database Connection.

Data Source Name	tqa_9212_1
* Driver Type	Other
* Database Driver Class	com.jdedwards.jdbc.driver.JDBCDriver (Example: oracle.jdbc.OracleDriver)
* Connection String	jdbc:oracle:enterpriseone://JDV920;TDSC=1;CDSC=1;UDSC=1;
Use System User	□
* Username	JDE
Password	••••••
Pre Process Function	
Post Process Function	
	□ Use Proxy Authentication

7. Click Apply.

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

   This example shows the completed form:



Data Source Name	Connection String	Delete
DAD_DENI507B	jdbc:oracle:enterpriseone://JDV900;TDSC=1;CDSC=1;UDSC=1;	🗑
DAD_HYPERION	jdbc:oracle:enterpriseone://PY900C1;TDSC=1;CDSC=1	🗑
DAD_JDV900	jdbc:oracle:enterpriseone://JDV900;TDSC=1;CDSC=1	🗑
DAD_JDV910	jdbc:oracle:enterpriseone://JDV910;TDSC=1;CDSC=1	🗑
demo	jdbc:oracle:thin:@HOST:PORT:SID	🗑
JDBC_DAD	jdbc:oracle:enterpriseone://JDV900;TDSC=1;UDSC=1;	🗑
Oracle BI EE	jdbc:oraclebi://DENPBDS2:9703/	🗑
TEST	jdbc:oracle:enterpriseone://JDV910;TDSC=1;CDSC=1;UDSC=1;	🗑

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

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 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.

9 Creating Oracle BI Publisher Reports with JD Edwards Data Access Driver for Oracle 12c

JD Edwards Data Access Reporting with Oracle BI Publisher 12c Overview

JD Edwards EnterpriseOne Data Access Driver enables you to use Oracle Business Intelligence (BI) Publisher Enterprise (OBIEE-BIP) to produce BI Publisher reports using JD Edwards data.

This chapter provides an overview of installing and configuring Oracle Business Intelligence (BI) Publisher 12c and JD Edwards Data Access Driver. The process to implement JD Edwards EnterpriseOne reporting using the Data Access Driver is as follows:

- Install Oracle BI Publisher.
- Install the JD Edwards EnterpriseOne Data Access Driver.

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

- Configure a JDBC Driver in Oracle BI Publisher.

Installing Oracle BI Publisher

To use the Data Access Driver with Oracle BI Publisher, you only need to install Oracle BI Publisher. You do not need to install Oracle Business Intelligence Enterprise Edition or Oracle Real-Time Decisions.

You need to install WebLogic before installing BI Publisher 12c. The default domain in WebLogic functions for BI Publisher, so you do not need to create an additional domain unless you prefer to keep the domains separate.

Prerequisites

Before you complete the tasks in this section, you must:

- Install and configure a supported database.

  For the latest information about supported databases, visit the Oracle Fusion Middleware Certification document at:

  http://www.oracle.com/technetwork/middleware/ias/downloads/fusion-certification-100350.html

  Look for product area "Oracle Fusion Middleware 12c Certifications" and then access "System Requirements and Supported Platforms" for:

  - System Requirements and Supported Platforms for Oracle Fusion Middleware 12c (12.2.1.1.0) (xls)

- Download and install the WebLogic Server (WLS) from Oracle Technology Network (OTN), Oracle Business Intelligence 12c (12.2.1.1.0) Downloads page:

  `http://www.oracle.com/technetwork/middleware/bi-enterprise-edition/downloads/business-intelligence-3046226.html`
- Download Oracle BI Publisher from the Oracle Software Delivery:
  https://www.edelivery.oracle.com
  Select Oracle Business Intelligence for the Product Pack field and choose the appropriate platform.
- Install JD Edwards EnterpriseOne Applications Release 9.2.

Installing Oracle BI Publisher

Use the information provided in the *Installing and Configuring Oracle Business Intelligence* guide to perform the installation. Note that although the Oracle BI installation package includes the following products, with all the technology required to install, configure, and run them, you only need to install Oracle Business Intelligence Publisher:

- Oracle Business Intelligence Publisher
- Oracle Business Intelligence Enterprise Edition (Oracle BI Answers, Oracle BI Interactive Dashboards, Oracle BI Delivers, Oracle BI Administration Tool, Oracle BI Add-in for Microsoft Office, and Oracle BI Publisher) (optional)
- Oracle Real-Time Decisions (optional)
- Essbase Suite (optional)
- WebLogic Server

> **Note:** For BI Publisher reporting using the Data Access Driver, you only need to install Oracle BI Publisher.

The following steps provide guidance for the installation. Also see *Installing and Configuring Oracle Business Intelligence* .

> **Note:** A new installation of BI Publisher is recommended even if you are currently on a prior release. This enables you to perform some parallel testing before switching to the latest release.

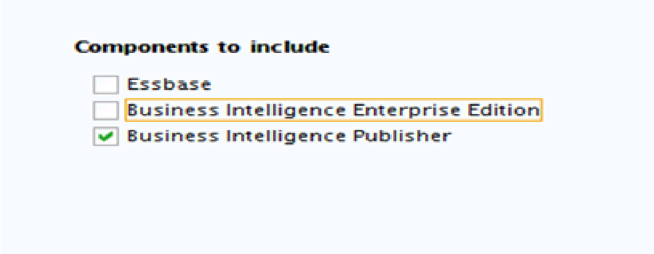
To install Oracle BI Publisher:

1. Launch the Oracle Business Intelligence Installer to install and create the BI Publisher Schemas (BIPLATFORM and MDS).
   For more information, see the *Installing and Configuring Oracle Business Intelligence* guide.

   > **CAUTION:**
   > - Do not install the software using a user ID that contains a special character.
   > - BI Publisher and the HTML server must be within the same firewall in order to have two-way web service and http communication.
   > - If you have anti-virus software running on your installing machine, the installation process may take longer.

2. Select the appropriate installation type:
    - "Simple Install" installs Oracle Business Intelligence components with the default settings on a single computer in the minimum number of steps. The Managed Server is not installed. This installation is targeted for demonstration, evaluation, and proof of concept use cases.
    - "Enterprise Install" installs Oracle Business Intelligence components that you select and associates them with the Middleware, Oracle data source, and WebLogic Server homes that you specify. The enterprise installation type creates a separate Managed Server, bi_server1, along with the Admin Server. You should also use this installation type if you have a previous version of BI Publisher installed.

      This installation type is recommended for a production environment.
    - "Software Only Install" installs software binary files in an existing Middleware home for later configuration.
3. When the Oracle Business Intelligence Component Installation screen appears, select only the Business Intelligence Publisher option:



Components to include

□ Essbase
□ Business Intelligence Enterprise Edition
☑ Business Intelligence Publisher

> **Note:** For detailed instructions on how to install Oracle BI Publisher, see the *Installing and Configuring Oracle Business Intelligence* guide.

4. When the installation and configuration finish, sign onto the WebLogic Admin Console and BI Server to make sure the servers are up and running by using the following URLs:

   http://host:port/console (WebLogic Admin Console)

   http://host:9502/xmlpserver (BI Publisher login page, port 9502 is the default port for "Enterprise Install" type)
5. Review the Certifications for any updates or additional requirements.

Installing the JD Edwards EnterpriseOne Data Access Driver

The Data Access Driver can be downloaded from the JD Edwards EnterpriseOne and World *Update Center*.

To install the JD Edwards EnterpriseOne Data Access Driver:

1. Unzip the tools-specific Data Access Driver par file archive (for example, 9.2.n.n-Data-Access-Driver_06_70.par). Extract all files from the DADriver_EAR.jar file.
2. On the server on which 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_BIP_home>/user_projects/domains/<bifoundation_domain>/lib folder:
    - avalon-framework-cvs-20020806.jar

- castor.jar (releases prior to 9.2.6 only)
- commons-codec.jar
- commons-codec-2.6.jar
- commons-logging.jar
- e1dadriver.jar
- fop.jar
- httpclient.jar
- httpcore.jar
- httpmime.jar
- images.jar
- j2ee.jar
- jakarta.activation.jar (Tools Releases 9.2.6 and greater)
- jakarta.xml.bind-api.jar (Tools Releases 9.2.6 and greater)
- jaxb-core.jar (Tools Releases 9.2.6 and greater)
- jaxb-impl.jar (Tools Releases 9.2.6 and greater)
- jmxremote.jar
- jmxremote_optional.jar
- jmxri.jar
- OWResource.jar
- rtf2fo.jar
- xalan.jar
- xerces.jar
- xmlparserv2.jar

Replacing 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 12c WebLogic xercesImpl.jar common file (for example, <orcl_BI>/oracle/Middleware/oracle_common/modules/oracle.owasp_12.2.1.2/).
2. Copy the xercesImpl.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.

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 12c BI Publisher option from the drop-down list.

   > **Note:** The Software Component box will not be displayed when the Usage Type selected is 12c BI Publisher.

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

   - Install Name
   - Install Location

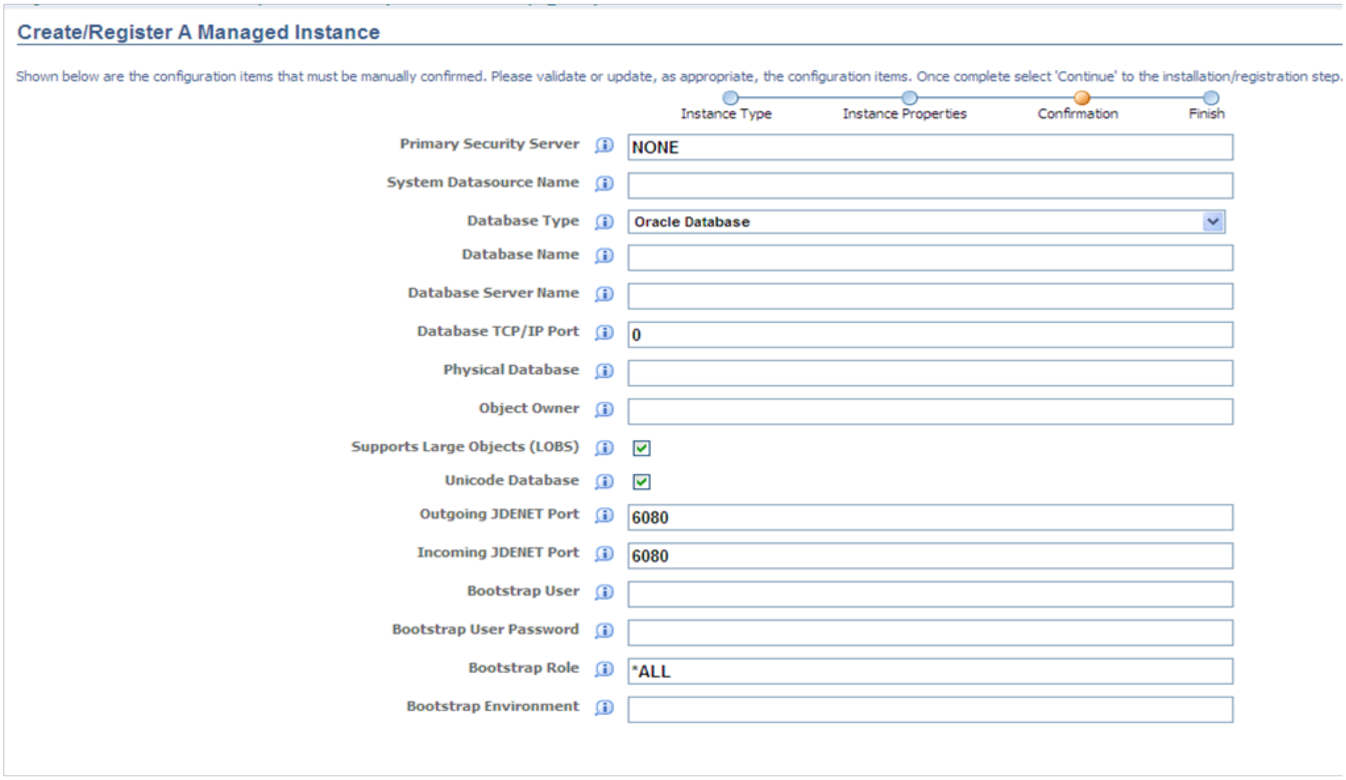
   Ensure that you modify the Install Location to match the machine on which OBIEE-BIP is installed.

   > **Note:** 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/BIP12c/user_projects/domains/bifoundation_domain/servers/bi_server1*

6. Click Continue to proceed.

7. Enter appropriate values in the relevant fields.

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



Create/Register A Managed Instance

Shown below are the configuration items that must be manually confirmed. Please validate or update, as appropriate, the configuration items. Once complete select 'Continue' to the installation/registration step.

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

8. Click the Create Instance button to complete the Data Access Driver registration.

   After you complete the registration, the browser is redirected to the EnterpriseOne Data Access Driver home page.

Adding Classpath in the WebLogic Console

To add classpath in the WebLogic Console:

1. Log on to the WebLogic Console.
2. In the Domain Structure section, expand the Environment node.
3. Expand the Server and click the server on which the Business Intelligence Publisher is installed.
4. Click Lock & Edit.
5. Click the Server Start tab.

6. Enter the location of the DAD jar files and the configuration files in the Class Path field.

   > **Note:** The path for the DAD jar files and the configuration files must be separated by a colon for Linux machines and a semicolon for Windows machine. For example, /slot/ems7051/oracle/jde_agent_DAD/SCFHA/targets/BIP21c_den60203jems/config:/slot/ems7051/oracle/Middleware/user_projects/domains/bi/lib.

7. Click Save, and then click Activate Changes.
8. Restart the BI Publisher server.

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 12c.

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.

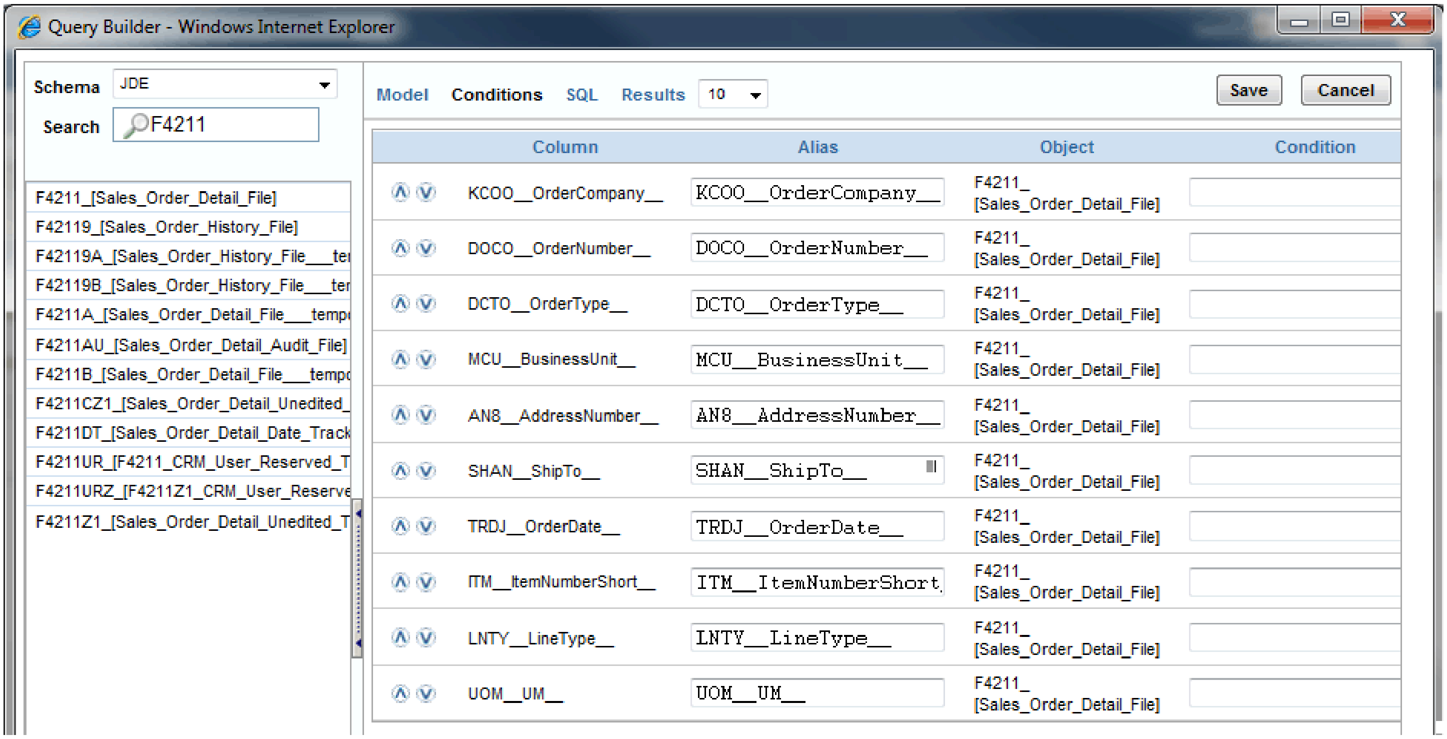
JDBC Connection String

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.



Schema JDE

Search F4211

Results 10

F4211_[Sales_Order_Detail_File]
F42119_[Sales_Order_History_File]
F42119A_[Sales_Order_History_File___ter
F42119B_[Sales_Order_History_File___ter
F4211A_[Sales_Order_Detail_File___tempo
F4211AU_[Sales_Order_Detail_Audit_File]
F4211B_[Sales_Order_Detail_File___tempo
F4211CZ1_[Sales_Order_Detail_Unedited_
F4211DT_[Sales_Order_Detail_Date_Track
F4211UR_[F4211_CRM_User_Reserved_T
F4211URZ_[F4211Z1_CRM_User_Reserve
F4211Z1_[Sales_Order_Detail_Unedited_T

Column	Alias	Object	Condition
KCOO__OrderCompany__	KCOO__OrderCompany__	F4211_ [Sales_Order_Detail_File]	
DOCO__OrderNumber__	DOCO__OrderNumber__	F4211_ [Sales_Order_Detail_File]	
DCTO__OrderType__	DCTO__OrderType__	F4211_ [Sales_Order_Detail_File]	
MCU__BusinessUnit__	MCU__BusinessUnit__	F4211_ [Sales_Order_Detail_File]	
AN8__AddressNumber__	AN8__AddressNumber__	F4211_ [Sales_Order_Detail_File]	
SHAN__ShipTo__	SHAN__ShipTo__	F4211_ [Sales_Order_Detail_File]	
TRDJ__OrderDate__	TRDJ__OrderDate__	F4211_ [Sales_Order_Detail_File]	
ITM__ItemNumberShort__	ITM__ItemNumberShort	F4211_ [Sales_Order_Detail_File]	
LNTY__LineType__	LNTY__LineType__	F4211_ [Sales_Order_Detail_File]	
UOM__UM__	UOM__UM__	F4211_ [Sales_Order_Detail_File]	

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, *enterpriseone.role=SYSADMIN*. If the role is not specified in the connection string, the *ALL role will be used.

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.

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.

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 semicolon 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://JDV920;TDSC=1;CDSC=1;UDSC=1;*

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.

Configuring the JDBC Driver in Oracle Business Intelligence Publisher

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: *E1_DAD*
Driver Type	*Other*
Database Driver Class	*com.jdedwards.jdbc.driver.JDBCDriver*
Connection String	*jdbc:oracle:enterpriseone://<JD Edwards environment>;* Example: *jdbc: oracle:enterpriseone://JDV920;* **Note**: 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: *enterpriseone.role = *ALL*. 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. **Note**: You can append information to the end of the connection string to display table, column, or UDC descriptions if desired. *TDSC=1* displays the table name description, *CDSC=1* displays the column description, and *UDSC=1* displays the UDC description.

Field	Value

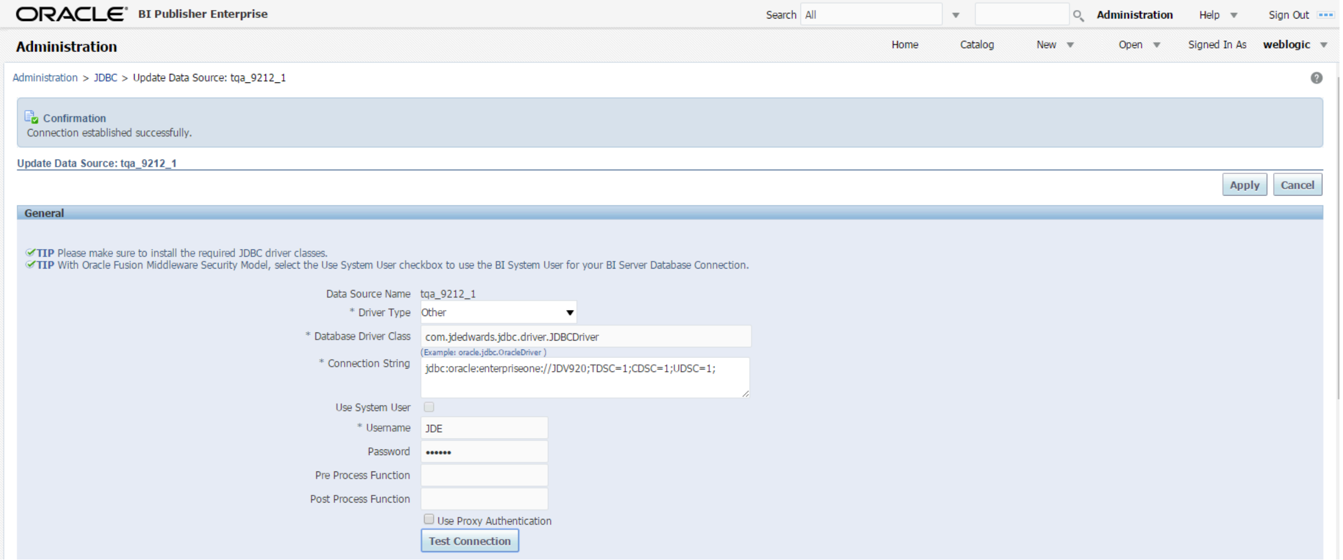
	Example of connection string without display options:
	jdbc:oracle:enterpriseone://JDV920;
	Example of connection string with display options:
	jdbc:oracle:enterpriseone://JDV920;TDSC=1;CDSC=1;UDSC=1;

User Name	Enter the JD Edwards EnterpriseOne user ID that you will connect to the database with. **Note:** The user name is case-sensitive.
Password	Enter the password for the user ID. **Note:** The password is case-sensitive.
Use Proxy Authentication	*Selected* or *Cleared*

Field	Value

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:



Confirmation
Connection established successfully.

Update Data Source: tqa_9212_1

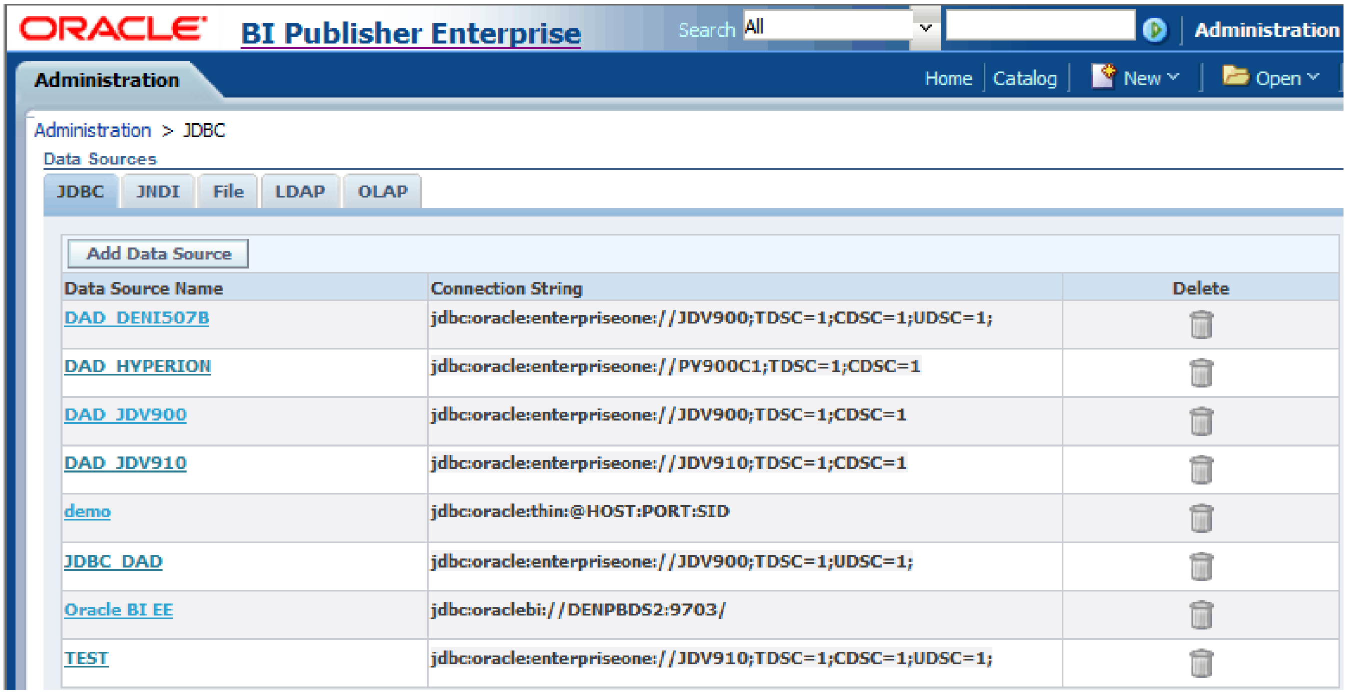
TIP Please make sure to install the required JDBC driver classes.
TIP With Oracle Fusion Middleware Security Model, select the Use System User checkbox to use the BI System User for your BI Server Database Connection.

Data Source Name	tqa_9212_1
* Driver Type	Other
* Database Driver Class	com.jdedwards.jdbc.driver.JDBCDriver (Example: oracle.jdbc.OracleDriver)
* Connection String	jdbc:oracle:enterpriseone://JDV920;TDSC=1;CDSC=1;UDSC=1;
Use System User	☐
* Username	JDE
Password	••••••
Pre Process Function	
Post Process Function	
	☐ Use Proxy Authentication

7. Click Apply.

   The new JDBC data source should appear in the Data Sources list.

   This example shows the completed form:



Data Source Name	Connection String	Delete
DAD_DENI507B	jdbc:oracle:enterpriseone://JDV900;TDSC=1;CDSC=1;UDSC=1;	🗑
DAD_HYPERION	jdbc:oracle:enterpriseone://PY900C1;TDSC=1;CDSC=1	🗑
DAD_JDV900	jdbc:oracle:enterpriseone://JDV900;TDSC=1;CDSC=1	🗑
DAD_JDV910	jdbc:oracle:enterpriseone://JDV910;TDSC=1;CDSC=1	🗑
demo	jdbc:oracle:thin:@HOST:PORT:SID	🗑
JDBC_DAD	jdbc:oracle:enterpriseone://JDV900;TDSC=1;UDSC=1;	🗑
Oracle BI EE	jdbc:oraclebi://DENPBDS2:9703/	🗑
TEST	jdbc:oracle:enterpriseone://JDV910;TDSC=1;CDSC=1;UDSC=1;	🗑

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 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.

   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.

10 Appendix A - JD Edwards EnterpriseOne BI Publisher Releases

JD Edwards XML Publisher 8.96 Templates

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 was further extended in JD Edwards EnterpriseOne Tools 8.98 to support creating BI Publisher reports with JD Edwards Data Access Driver.

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

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 9.1 at this time.

    Oracle strongly recommends that you migrate to 9.1 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* .

Migrating from 8.96 XML Publisher

The way that XML output is generated in 9.1 BI Publisher for EnterpriseOne is different from how the XML output was generated in the 8.96 release.

See *Understanding the XML Output*.

Because of this difference, if you are currently using 8.96 XML Publisher, Oracle recommends that you migrate to 9.1 BI Publisher for JD Edwards EnterpriseOne.

After upgrading to 9.1 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 9.1 BI Publisher document tags.

11 Appendix B - Data Access Driver Troubleshooting

SQL Exceptions and Error Messages

When errors occur, the JDBC driver throws SQLExceptions. When this happens, check the log files for additional information. The log locations are specified in the jdelog.properties file. If the logs do not help to resolve the issue, enable the debug logs for more information.

It is especially helpful to inspect entire exception stack traces, because traces include exception messages, class names, and line numbers. Exception stack traces cause exceptions that result in SQLExceptions.

When you evaluate a series of exceptions in a trace, you should concentrate on the first exception, since it is often the cause of subsequent exceptions.

Following are some exception examples, with recommendations on how to resolve them.

No Suitable Driver

Exception: java.sql.SQLException: No suitable driver.

Cause: The JD Edwards EnterpriseOne JDBC drivers use the native database JDBC drivers to access physical data. If the class path does not include the necessary drivers, the JDBC drivers throw this exception on any attempt to read physical data.

Recovery: For the Type 4 JDBC driver, contact your system administrator and ensure that all of the applicable JDBC drivers are included in the same class path as the Data Access Driver.

Data Source for F0010, TBLE Not Found

Exception: com.jdedwards.services.objectlookup.DataSourceNotFoundException: Data source for F0010, TBLE not found. (With a cause message in parenthesis)

Cause: This exception indicates that the JDBC driver cannot access its system tables in ERP mode. Table F0010 is the first system table that the JDBC driver attempts to access. Be sure to check the cause message that is attached to the exception message. The exception trace usually includes a direct cause as well.

Recovery: Check the cause exception and follow the recovery instructions listed for those exceptions. If none apply, contact your system administrator and verify that the [JDBj-BOOTSTRAP DATA SOURCE] section of the jdbj.ini file references a valid data source. The JDBj-BOOTSTRAP DATA SOURCE section describes the location of ERP system tables like F0010.

Receiving "[SECURITY_ERROR] A Security Error Occurred"

These errors occur when you create models in Oracle BI Publisher Enterprise using JD Edwards EnterpriseOne tables.

Data Access Driver (DAD) logs contain the following messages:

```
Sign on: valid = false
  Sign on: status = 17
  Sign on: error message ID = 348
                        com.jdedwards.database.base.JDBException: [SECURITY_ERROR] A
                        security error occurred.
com.jdedwards.database.base.JDBException: [SECURITY_ERROR] A
security error occurred.
com.jdedwards.database.base.JDBException: [SECURITY_ERROR] A
security error occurred.
Cause exception com.jdedwards.database.base.JDBException:
[SECURITY_ERROR] A security error occurred.
JDESpec error code = SPEC_REP_LOAD_FAILED
 com.peoplesoft.pt.e1.base.metadata.dataAccess.JDESpecException:
Spec repository implementation could not be loaded.
Spec repository implementation could not be loaded.
 com.peoplesoft.pt.e1.base.metadata.dataAccess.JDESpecException:
Spec repository implementation could not be loaded.
For package <PD114> in data source <Central Objects - PD812>,
failed to find the package manifest. Aborting the discovery
process, this package is not functional.
 com.jdedwards.database.services.packagelookup.PackageLookupServic
eException: Spec repository implementation could not be loaded.
com.jdedwards.base.spec.SpecException: [SECURITY_ERROR] A
security error occurred.
com.jdedwards.database.base.JDBException: [SECURITY_ERROR] A
security error occurred.
com.jdedwards.base.spec.SpecException: [SECURITY_ERROR] A
security error occurred. com.jdedwards.base.spec.SpecException:
SECURITY_ERROR] A security error occurred.
ROJDBC - getColumnsInfo  specMap.getSpec() is causing exception
for table=F4101 com.jdedwards.base.spec.SpecException:
[SECURITY_ERROR] A security error occurred.
```

Cause: This error occurs when the security token allocated for a Data Access Driver has expired. DAD starts its connection to EnterpriseOne when the BI Publisher Enterprise instance is started. At that time, it requests and obtains a security token from the Security Kernel running on the Enterprise Server. This token is used to grant DAD access to EnterpriseOne tables. The security token has a lifetime, and after it expires transactions between DAD and EnterpriseOne tables will fail.

Solution: See document 885414.1 on My Oracle Support.

Invalid SQL Queries Cause Database or Query Timeout Errors in BI Publisher Enterprise

Invalid SQL queries can cause the following errors.

Example 1

The Data Access Driver shows the following error messages:

```
08 Dec 2009 09:34:56,531 [SEVERE] - [JDBJ] SQLSTATE = 42000
SQLMessage = SQLMessage = ORA-01795: maximum number of expressions in a list
                        is 1000
08 Dec 2009 09:34:56,531 [SEVERE] - [JDBJ] Query failed to
execute within the specified timeout of 10000 ms. Timeout the
Query
08 Dec 2009 09:34:56,532 [SEVERE] - [BASE]
com.jdedwards.database.base.JDBException: [QUERY_TIMED_OUT] Query
timed out. Operation is cancelled.
08 Dec 2009 09:34:56,532 [SEVERE] - [BASE]
com.jdedwards.database.base.JDBException: [QUERY_TIMED_OUT] Query
timed out. Operation is cancelled.
com.jdedwards.database.base.JDBException:
```

Solution: Rewrite the SQL query used in BI Publisher Enterprise to avoid using more than 1000 parameters in a WHERE clause expression.

Example 2

The Data Access Driver shows the following error messages:

```
com.jdedwards.database.base.JDBException: [QUERY_TIMED_OUT]
Query timed out. Operation is cancelled.
```

Solution: Increase the connection timeout parameter in jdbj.ini.

```
[JDBj-CONNECTION POOL]
connectionTimeout=1800000
cleanPoolInterval=1800000
```

Example 3

The Data Access Driver shows the following error messages:

```
[SQL_EXCEPTION_OCCURRED] An SQL exception occurred:
Resultset timeout java.sql.SQLException
```

Solution: Increase the resultset timeout parameter in jdbj.ini.

```
[JDBj-RUNTIME PROPERTIES]
msSQLQueryTimeout=1800000
resultSetTimeout=1800000
transactionTimeout=1800000
```

Query Builder Does Not Load JD Edwards EnterpriseOne Tables

Error: Query Builder throws the error "Could not load schema information. Please make sure that this connection is active." This error occurs even though the JDBC data source test connection is successful. The error occurs with or without Proxy Authentication in the JDBC Data Source.

Cause: The e1root log represents that the log in credentials are incorrect in [JDBj-BOOTSTRAP SESSION] section of jdbj.ini, as shown below.

```
06 Dec 2011 17:47:58,000 [SEVERE] - [JDBJ] Sign on: valid =
false
Sign on: status = 2
Sign on: error message ID = 331
```

Solution: To resolve this issue, ensure that you entered valid user, password, role and environment parameters in the [JDBj-BOOTSTRAP SESSION] section. There should not be any blanks.

For example:

```
[JDBj-BOOTSTRAP SESSION]
role=*ALL
user=JDE
password=xxxxxxxx
environment=JPD900
```

Additionally:

- Use Server Manager to enter the password, so that it will be encrypted properly. If you enter the password directly in jdbj.ini, it will not be encrypted. An unencrypted password will cause issues in Query Builder.
- The JDBC data source connection string should use the same environment as the one in the [JDBj-BOOTSTRAP SESSION] section. Otherwise the test connection will fail.
- After making changes to the jdbj or jas.ini files, restart the BI server to make the changes effective.

12 Glossary

batch versions

Variations of a report template that contain changes to the way the report is processed or displayed. When you submit a report, you must use a batch version.

connection mode

A term that applies only to the JDBC drivers and provides an indication of the type of additional filtering and processing that the JD Edwards EnterpriseOne data that you are accessing requires. Application code designates a connection mode when it establishes each new connection.

connection properties

Properties that applications pass to the JDBC drivers when establishing a new connection in order to configure a particular connection type. The concept of connection properties is a standard JDBC mechanism, but each driver defines its own set of recognized connection properties.

connection URL

A string that identifies a particular data source to which to connect. The concept of a connection URL is a standard JDBC mechanism, but each driver defines its own URL syntax.

driver class name

A string that identifies the primary class for a JDBC driver. You must register this class name with the JDBC driver manager before using it. This is a standard JDBC concept, but each driver defines its own driver class name.

driver manager

The JDBC class that manages multiple registered JDBC drivers and dispatches connection initialization requests to them. The Java driver manager class is java.sql.DriverManager.

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.

Index