ORACLE
PeopleSoft

PeopleTools 8.58: BI Publisher for PeopleSoft

May 2020

ORACLE

PeopleTools 8.58: BI Publisher for PeopleSoft
Copyright © 1988, 2020, Oracle and/or its affiliates.

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

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

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

U.S. GOVERNMENT END USERS: Oracle programs (including any operating system, integrated software, any programs embedded, installed or activated on delivered hardware, and modifications of such programs) and Oracle computer documentation or other Oracle data delivered to or accessed by U.S. Government end users are "commercial computer software" or “commercial computer software documentation” pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, reproduction, duplication, release, display, disclosure, modification, preparation of derivative works, and/or adaptation of i) Oracle programs (including any operating system, integrated software, any programs embedded, installed or activated on delivered hardware, and modifications of such programs), ii) Oracle computer documentation and/or iii) other Oracle data, is subject to the rights and limitations specified in the license contained in the applicable contract. The terms governing the U.S. Government’s use of Oracle cloud services are defined by the applicable contract for such services. No other rights are granted to the U.S. Government.

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

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

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

This software or hardware and documentation may provide access to or information about content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services unless otherwise set forth in an applicable agreement between you and Oracle. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services, except as set forth in an applicable agreement between you and Oracle.

The business names used in this documentation are fictitious, and are not intended to identify any real companies currently or previously in existence.

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.

Contents

Copyright © 1988, 2020, Oracle and/or its affiliates.

Copyright © 1988, 2020, Oracle and/or its affiliates.

Copyright © 1988, 2020, Oracle and/or its affiliates.

Copyright © 1988, 2020, Oracle and/or its affiliates.

Preface

Understanding the PeopleSoft Online Help and PeopleBooks

The PeopleSoft Online Help is a website that enables you to view all help content for PeopleSoft applications and PeopleTools. The help provides standard navigation and full-text searching, as well as context-sensitive online help for PeopleSoft users.

Hosted PeopleSoft Online Help

You can access the hosted PeopleSoft Online Help on the Oracle Help Center. The hosted PeopleSoft Online Help is updated on a regular schedule, ensuring that you have access to the most current documentation. This reduces the need to view separate documentation posts for application maintenance on My Oracle Support. The hosted PeopleSoft Online Help is available in English only.

To configure the context-sensitive help for your PeopleSoft applications to use the Oracle Help Center, see Configuring Context-Sensitive Help Using the Hosted Online Help Website.

Locally Installed Help

If you're setting up an on-premise PeopleSoft environment, and your organization has firewall restrictions that prevent you from using the hosted PeopleSoft Online Help, you can install the online help locally. See Configuring Context-Sensitive Help Using a Locally Installed Online Help Website.

Downloadable PeopleBook PDF Files

You can access downloadable PDF versions of the help content in the traditional PeopleBook format on the Oracle Help Center. The content in the PeopleBook PDFs is the same as the content in the PeopleSoft Online Help, but it has a different structure and it does not include the interactive navigation features that are available in the online help.

Common Help Documentation

Common help documentation contains information that applies to multiple applications. The two main types of common help are:

- Application Fundamentals
- Using PeopleSoft Applications

Most product families provide a set of application fundamentals help topics that discuss essential information about the setup and design of your system. This information applies to many or all applications in the PeopleSoft product family. Whether you are implementing a single application, some combination of applications within the product family, or the entire product family, you should be familiar with the contents of the appropriate application fundamentals help. They provide the starting points for fundamental implementation tasks.

Copyright © 1988, 2020, Oracle and/or its affiliates.

In addition, the *PeopleTools: Applications User's Guide* introduces you to the various elements of the PeopleSoft Pure Internet Architecture. It also explains how to use the navigational hierarchy, components, and pages to perform basic functions as you navigate through the system. While your application or implementation may differ, the topics in this user's guide provide general information about using PeopleSoft applications.

Field and Control Definitions

PeopleSoft documentation includes definitions for most fields and controls that appear on application pages. These definitions describe how to use a field or control, where populated values come from, the effects of selecting certain values, and so on. If a field or control is not defined, then it either requires no additional explanation or is documented in a common elements section earlier in the documentation. For example, the Date field rarely requires additional explanation and may not be defined in the documentation for some pages.

Typographical Conventions

The following table describes the typographical conventions that are used in the online help.

Typographical Convention	***Description***
Key+Key	Indicates a key combination action. For example, a plus sign (+) between keys means that you must hold down the first key while you press the second key. For Alt+W, hold down the Alt key while you press the W key.
. . . (ellipses)	Indicate that the preceding item or series can be repeated any number of times in PeopleCode syntax.
{ } (curly braces)	Indicate a choice between two options in PeopleCode syntax. Options are separated by a pipe (|).
[ ] (square brackets)	Indicate optional items in PeopleCode syntax.
& (ampersand)	When placed before a parameter in PeopleCode syntax, an ampersand indicates that the parameter is an already instantiated object. Ampersands also precede all PeopleCode variables.
⇒	This continuation character has been inserted at the end of a line of code that has been wrapped at the page margin. The code should be viewed or entered as a single, continuous line of code without the continuation character.

ISO Country and Currency Codes

PeopleSoft Online Help topics use International Organization for Standardization (ISO) country and currency codes to identify country-specific information and monetary amounts.

ISO country codes may appear as country identifiers, and ISO currency codes may appear as currency identifiers in your PeopleSoft documentation. Reference to an ISO country code in your documentation

Copyright © 1988, 2020, Oracle and/or its affiliates.

does not imply that your application includes every ISO country code. The following example is a country-specific heading: "(FRA) Hiring an Employee."

The PeopleSoft Currency Code table (CURRENCY_CD_TBL) contains sample currency code data. The Currency Code table is based on ISO Standard 4217, "Codes for the representation of currencies," and also relies on ISO country codes in the Country table (COUNTRY_TBL). The navigation to the pages where you maintain currency code and country information depends on which PeopleSoft applications you are using. To access the pages for maintaining the Currency Code and Country tables, consult the online help for your applications for more information.

Region and Industry Identifiers

Information that applies only to a specific region or industry is preceded by a standard identifier in parentheses. This identifier typically appears at the beginning of a section heading, but it may also appear at the beginning of a note or other text.

Example of a region-specific heading: "(Latin America) Setting Up Depreciation"

Region Identifiers

Regions are identified by the region name. The following region identifiers may appear in the PeopleSoft Online Help:

- Asia Pacific
- Europe
- Latin America
- North America

Industry Identifiers

Industries are identified by the industry name or by an abbreviation for that industry. The following industry identifiers may appear in the PeopleSoft Online Help:

- USF (U.S. Federal)
- E&G (Education and Government)

Translations and Embedded Help

PeopleSoft 9.2 software applications include translated embedded help. With the 9.2 release, PeopleSoft aligns with the other Oracle applications by focusing our translation efforts on embedded help. We are not planning to translate our traditional online help and PeopleBooks documentation. Instead we offer very direct translated help at crucial spots within our application through our embedded help widgets. Additionally, we have a one-to-one mapping of application and help translations, meaning that the software and embedded help translation footprint is identical—something we were never able to accomplish in the past.

Copyright © 1988, 2020, Oracle and/or its affiliates.

Using and Managing the PeopleSoft Online Help

Select About This Help in the left navigation panel on any page in the PeopleSoft Online Help to see information on the following topics:

- Using the PeopleSoft Online Help
- Managing Hosted online help
- Managing locally installed PeopleSoft Online Help

Understanding BI Publisher for PeopleSoft Enterprise

Business Intelligence (BI) Publisher for PeopleSoft Enterprise is a template-based reporting solution that separates the data extraction process from the report layout and allows the reuse of extracted application data into multiple report layouts. BI Publisher uses select features from Oracle Business Intelligence Publisher (BI Publisher) that have been integrated into PeopleTools.

PeopleTools Related Links

PeopleTools 8.58 Home Page

PeopleTools Elasticsearch Home Page

"PeopleTools Product/Feature PeopleBook Index" (PeopleTools 8.58: Getting Started with PeopleTools)

PeopleSoft Hosted Online Help

PeopleSoft Information Portal

PeopleSoft Spotlight Series

PeopleSoft Training and Certification | Oracle University

My Oracle Support

Oracle Help Center

Contact Us

Send your suggestions to psoft-infodev_us@oracle.com. Please include the applications update image or PeopleTools release that you're using.

Copyright © 1988, 2020, Oracle and/or its affiliates.

Follow Us

Facebook.

YouTube

Twitter@PeopleSoft_Info.

PeopleSoft Blogs

LinkedIn

Copyright © 1988, 2020, Oracle and/or its affiliates.

Chapter 1

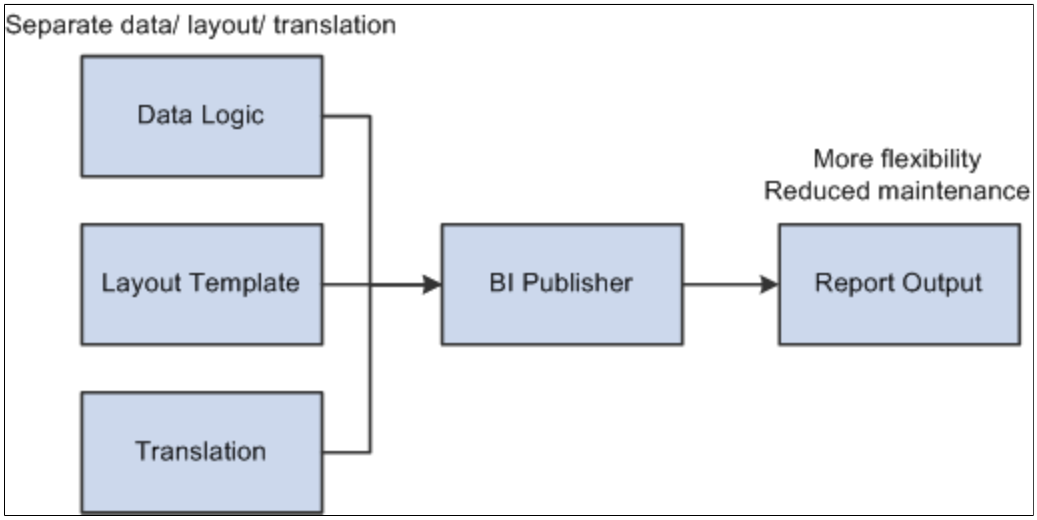
Getting Started with BI Publisher

BI Publisher Overview

Oracle Business Intelligence Publisher (BI Publisher, formerly XML Publisher) is an enterprise reporting solution that streamlines report and form generation. A primary feature of Oracle's BI Publisher product is the separation of the data extraction process from the report layout. BI Publisher enables you to design and create report layout templates with the more common desktop applications of Microsoft Word and Adobe Acrobat, and renders XML data based on those templates. With a single template, it can generate reports in many formats (PDF, RTF, Excel, HTML, and so on) in many languages. This approach to reporting can dramatically reduce report maintenance, enabling power business users to adjust report templates without involvement of IT resources.

Image: BI Publisher concept

The following diagram illustrates the concept of BI Publisher.

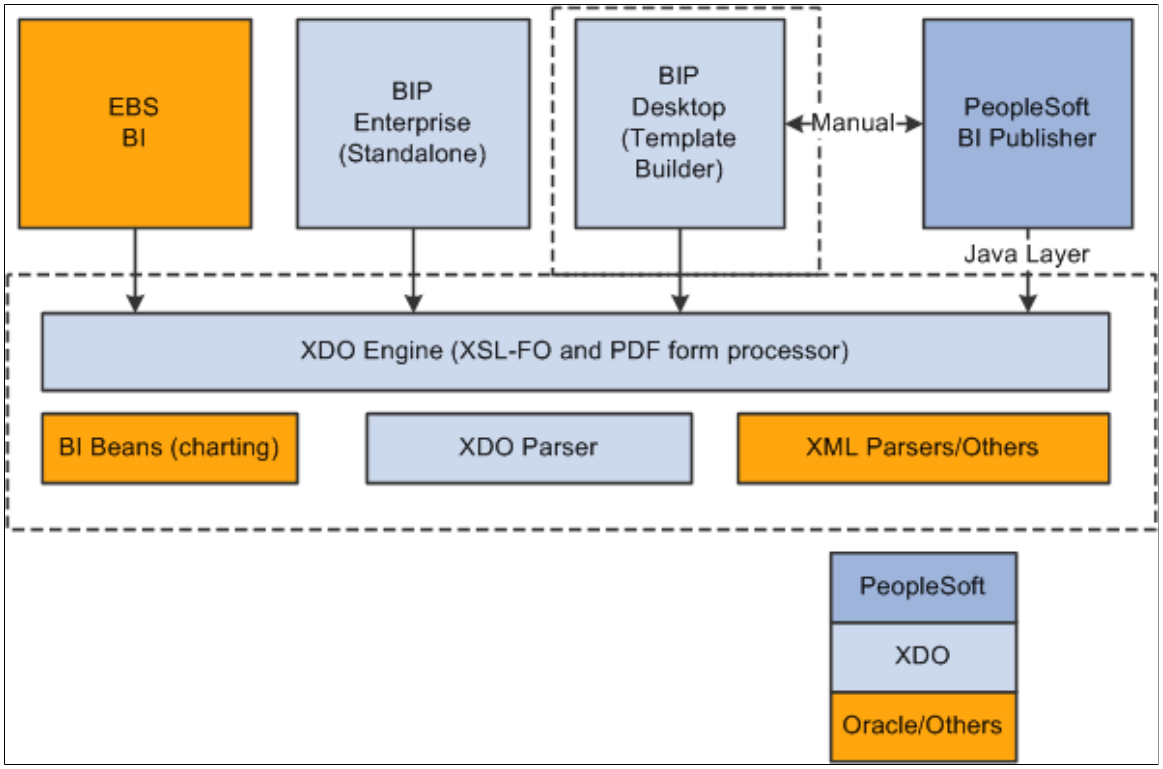


Select features of Oracle's BI Publisher product have been integrated into and enhanced for use with PeopleTools.

Copyright © 1988, 2020, Oracle and/or its affiliates.

Image: Integration with BI Publisher

This diagram illustrates the integration of PeopleSoft applications with BI Publisher.



PeopleSoft BI Publisher has a direct Java integration to the XDO Engine and XDO Parser. The BI Publisher Desktop requires installation and can be downloaded from a PeopleSoft Pure Internet Architecture page.

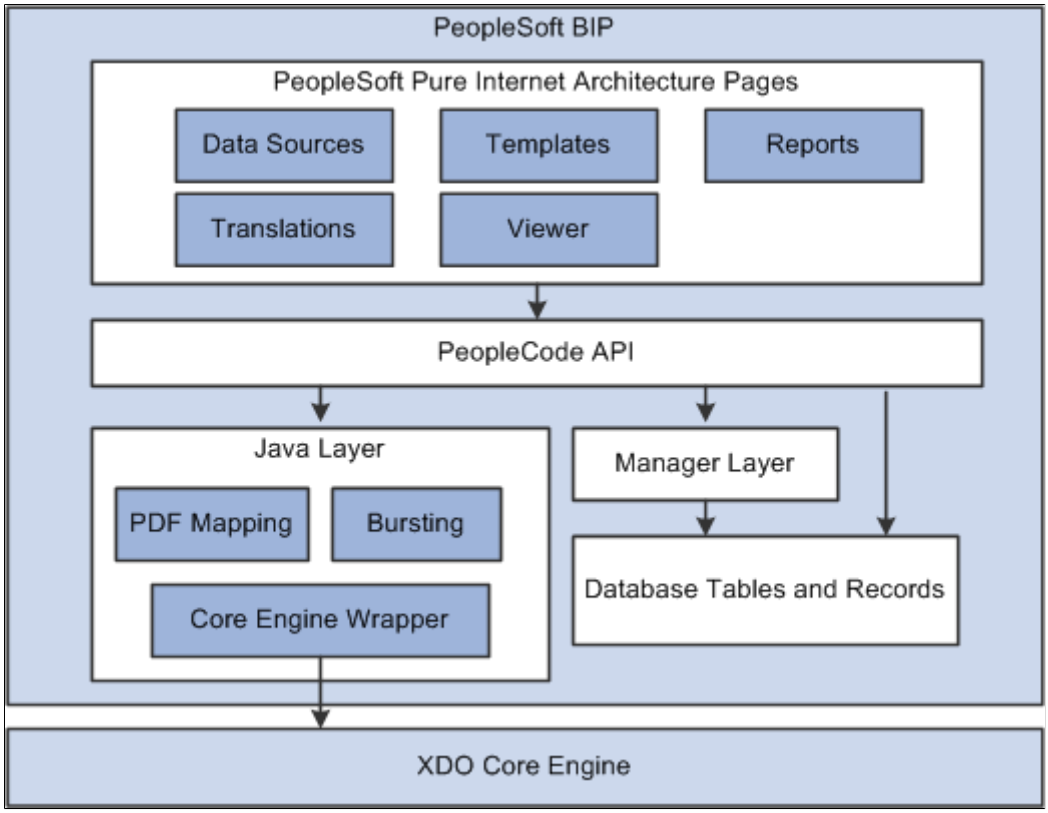
Note: Not all BI Publisher features are available through the PeopleSoft implementation.

PeopleSoft Query as well as any PeopleTools based applications providing XML data are available to BI Publisher as a data source. BI Publisher for PeopleSoft Enterprise provides an environment for managing templates, data sources, reports, translations, and content components. It also offers an electronic bursting capability to produce reports according to a user-defined criteria and secure the reports using an application's security join table. A set of PeopleCode BI Publisher classes for runtime report generation is also provided.

Copyright © 1988, 2020, Oracle and/or its affiliates.

Image: BI Components

This diagram illustrates the BI components within the PeopleSoft system.



Oracle provides a set of PeopleSoft Pure Internet Architecture pages for defining data sources, report definitions, templates, and translations and for running and viewing reports. Oracle also provides a set of PeopleCode application program interfaces (APIs) that wrap the Oracle XDO engine APIs. These APIs are used by the BI Publisher PeopleSoft Pure Internet Architecture pages and are available for advanced report developers to use for custom applications and batch processes.

BI Publisher Phases

BI Publisher implementation includes the following phases:

- Set up BI Publisher.
- Create and register data sources.
- Create and upload report templates.
- Define BI Publisher reports.
- Run, locate, and view BI Publisher reports.

Set Up BI Publisher

To prepare your system for using BI Publisher, perform the following steps:

Copyright © 1988, 2020, Oracle and/or its affiliates.

Step	*Reference*
1. Define BI Publisher settings.	See Setting Up BI Publisher.
2. Set up BI Publisher permission list security.	See Assigning BIP Permissions to Users.
3. Set up Report Manager.	See "Understanding Report Manager" (PeopleTools 8.58: Process Scheduler).
4. Define report categories, including Report Definition Editor security.	See Setting Up Report Categories.
5. Download design plug-ins to facilitate the offline template design activities.	See Working with Template Design Helpers.

Create and Register Data Sources

To create and register data sources, perform the following steps:

Step	*Reference*
1. Identify or create the source of your report data. Data sources can be PS Query, Connected Queries, XML files, or Composite Query . **Note:** Rowset and XML Doc object data sources have been deprecated. Rowset and XML Doc object data sources created in previous releases will continue to be supported. To generate XML files from XML Doc or rowsets, refer to PeopleCode API documentation. See "Understanding BI Publisher and the BI Publisher Classes" (PeopleTools 8.58: PeopleCode API Reference).	See "Creating New Queries" (PeopleTools 8.58: Query), "Running Queries" (PeopleTools 8.58: Query), andCreating Data Sources.
2. Register schema and sample data files for BI Publisher data sources. For PS Query, you can automatically generate schema file and sample data. **Note:** Schema is no longer used for bursting starting in PeopleTools 8.50. It is still available for backwards compatibility. Schema is also used with XSLT Mapper for data transforms.	See Registering Data Sources.

Create and Upload Report Templates

To create and upload templates, perform the following steps:

Copyright © 1988, 2020, Oracle and/or its affiliates.

Step	*Reference*
1. Create and upload schema and sample data.	See Creating Data Sources.
2. Download sample data from the appropriate data source to facilitate template design.	See Defining Reports.
3. Use either Microsoft Word or Adobe Acrobat to develop and maintain custom report formats.	See Understanding Report Template Types.
4. (Optional) Create and maintain reusable sub-template definitions.	See Maintaining Sub-Templates.
5. (Optional) Register translation XLIFF files for report templates and Content Library sub-templates.	See Maintaining Template Translations.

Define BI Publisher Reports

To create and maintain report definitions, perform the following step:

Step	*Reference*
1. Define reports by associating data sources with layout template files.	See Defining Reports.

Run, Locate, and View BI Publisher Reports

You can run BI Publisher reports online or in batch through the Process Scheduler. For query-based reports, pages are available for running the reports both online and in batch. To run BI Publisher reports, perform the following steps:

Step	*Reference*
1. Schedule Query-based BI Publisher reports.	See Scheduling Reports in Query Report Scheduler.
2. Schedule other BI Publisher reports. **Note:** You will need to create an application engine program using BI PeopleCode APIs.	See Running Reports Using PeopleCode.
3. View Query-based BI Publisher reports online in real time.	See Running Reports in Query Report Viewer.
4. View other BI Publisher reports online in real time.	See Running Reports Using PeopleCode.
5. Locate BI Publisher reports using enhanced search criteria.	See Searching the BI Publisher Report Repository, "Search Operator Values" (PeopleTools 8.58: PeopleCode API Reference).
6. View BI Publisher reports in the Report Manager.	See "Viewing Reports" (PeopleTools 8.58: Process Scheduler).

Copyright © 1988, 2020, Oracle and/or its affiliates.

Copyright © 1988, 2020, Oracle and/or its affiliates.

Chapter 2

Setting Up BI Publisher

Understanding BI Publisher Setup

Before using BI Publisher, there are some set up tasks necessary to set up the environment and facilitate template design. This table lists the categories for the set up tasks:

Set Up	*Description*
Properties	Properties for BI Publisher can be set at four levels. System and global properties should be configured as part of the initial set up for BI Publisher.
Security	Security is defined for creating and editing report definitions.
Template Design	Template Builder is an extension to Microsoft Word that simplifies the development of RTF templates. Template Builder can be downloaded from PeopleSoft Pure Internet Architecture page or Oracle Technical Network (OTN).

BI Publisher Properties

There are two types of properties used in BI Publisher:

System Properties

System level properties are set in the xdo.cfg file. System level properties include:

- xslt-parser
- xslt-scalable
- system-cachepage-size
- system-temp-dir
- fonts

Non-system Properties

Non-system or functional properties are set in PeopleSoft Pure Internet Architecture.

Property Definition Levels

There are four levels where properties are defined, this table lists the levels:

Copyright © 1988, 2020, Oracle and/or its affiliates.

Level	*Description*	*Location*
System properties and fonts	System properties and fonts.	xdo.cfg file
Global properties	Global properties are shared by all reports and override the xdo engine default property values.	Reporting Tools > BI Publisher > Setup > Global Properties
Report properties	Properties are defined in the report definition and override global properties for a single report.	Reporting Tools > BI Publisher > Report Definition > Properties See Setting Report Properties.
Runtime properties	Override report properties.	Runtime properties are set at runtime through ReportDefn. SetRuntimeProperties PeopleCode API.

Defining System Properties

This section provides an overview of the xdo.cfg file and discusses how to:

- Set system temp directory.
- Set application server or process scheduler domain-specific xdo.cfg file.

Understanding xdo.cfg File

BI Publisher system properties settings are defined in the xdo.cfg file. The default xdo.cfg file is located in the $PS_HOME/appserver directory, which is shared by all application server and process scheduler domains by default.

Note: In PeopleTools 8.4x, the xdo.cfg file is used to define all types of properties (system and non-system). In PeopleTools 8.50 and later releases the file should be used for system properties and fonts only. The result is unpredictable if the same property is defined in xdo.cfg and other levels.

This is an example of the xdo.cfg file:

```
<config version="1.0.0" xmlns="http://xmlns.oracle.com/oxp/config/">

    <properties>
        <!-- System level properties -->
        <property name="xslt-xdoparser">true</property>
        <property name="xslt-scalable">true</property>
        <property name="system-cachepage-size">50</property>
        <property name="system-temp-dir"></property>
    </properties>


    <!--<fonts>-->
        <!--<font family="3 of 9 Barcode" style="normal" weight="normal">-->
            <!--<truetype path="C:\WINNT\Fonts\3of9.ttf" />-->
        <!--</font>-->
    <!--</fonts>-->
```

 Copyright © 1988, 2020, Oracle and/or its affiliates.

```
</config>
```

See *Report Designer's Guide for Oracle Business Intelligence Publisher,* “Introduction to Designing Reports,” About Setting Run-Time Properties.”

Setting System Temp Directory

By default, the `system-temp-dir` property is not set. This property must be set to point to a temp folder on the server. Note that temporary files created in that directory could grow very large in size depending on the size of your reports, so you need to choose your`system-temp-dir` for optimum system performance.

Setting Application Server or Process Scheduler Domain-Specific xdo.cfg File

You can also specify an application server or process scheduler domain-specific xdo.cfg file. To do this, you need to change the application server or process scheduler configuration file to update the`JavaVM Options -Dxdo.ConfigFile` setting. For example, to specify a separate xdo.cfg file for the application server domain P8538041, change the [PS_CFG_HOME]/appserv/P8538041/psappsrv.cfg file as indicated in the following code samples and put the new xdo.cfg into the [PS_CFG_HOME]/appserv/P8538041directory.

Original line in psappsrv.cfg:

```
JavaVM Options=-Xrs -Dxdo.ConfigFile=%PS_HOME%/appserv/xdo.cfg
```

New line in psappsrv.cfg:

```
JavaVM Options=-Xrs -Dxdo.ConfigFile==%PS_CFG_HOME%/appserv/P8538041/xdo.cfg
```

In the preceding code sample, P8538041 is the application server domain name.

If you change the content of xdo.cfg, you don't need to restart the application server or the process scheduler domain that uses it. It refreshes automatically the next time you run it. But if you change the application server or process scheduler configuration file, you need to restart the affected domain.

Setting Up BI Publisher

This section discusses how to:

- Set up report categories.
- Define global properties.
- Work with template design helpers.

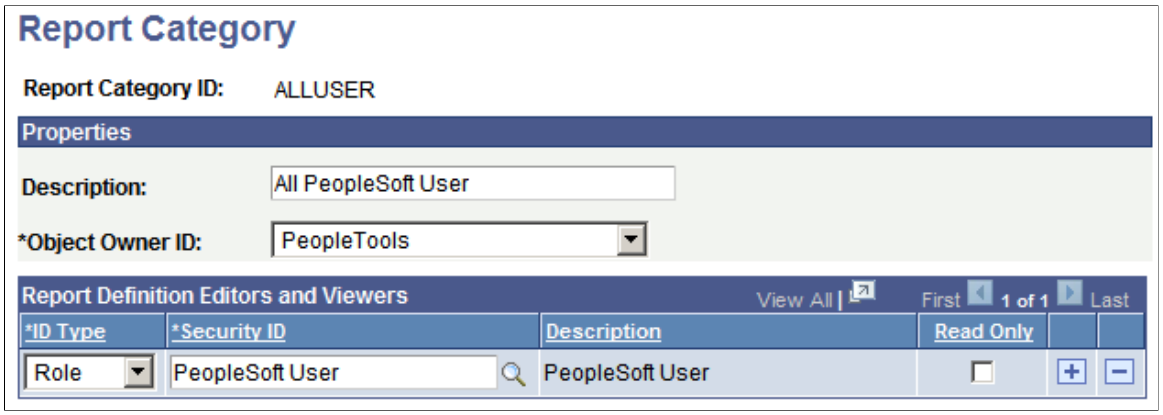
Setting Up Report Categories

Access the Report Category page (Select Reporting Tools > BI Publisher > Setup > Report Category.)

Copyright © 1988, 2020, Oracle and/or its affiliates.

Image: Report Category page

This example illustrates the fields and controls on the Report Category page. You can find definitions for the fields and controls later on this page.



Report Category is a required attribute on all report definitions and Content Library sub-templates. By assigning a report category, you are actually applying row level security to the data on those components.

Report Category ID Enter a report category ID to define a grouping that enables users to control who can access and edit report definitions and Content Library sub-templates.

Description (Optional) Enter descriptive text that provides more detail about the report category.

Object Owner ID Indicate which product, feature, or application owns this report category.

ID Type Select an ID type of either *Role* or *User ID* to grant authorization to.

Security ID Select the authorized editor's security ID based on the ID type.

Description A read-only field that indicates the related display value for the security ID.

Read Only (Optional) Select to indicate that the designated ID is only able to view the report definitions under this category and not update them.

Note: The PeopleCode BI Publisher classes also respect report category settings and read-only access rights.

Defining Global Properties

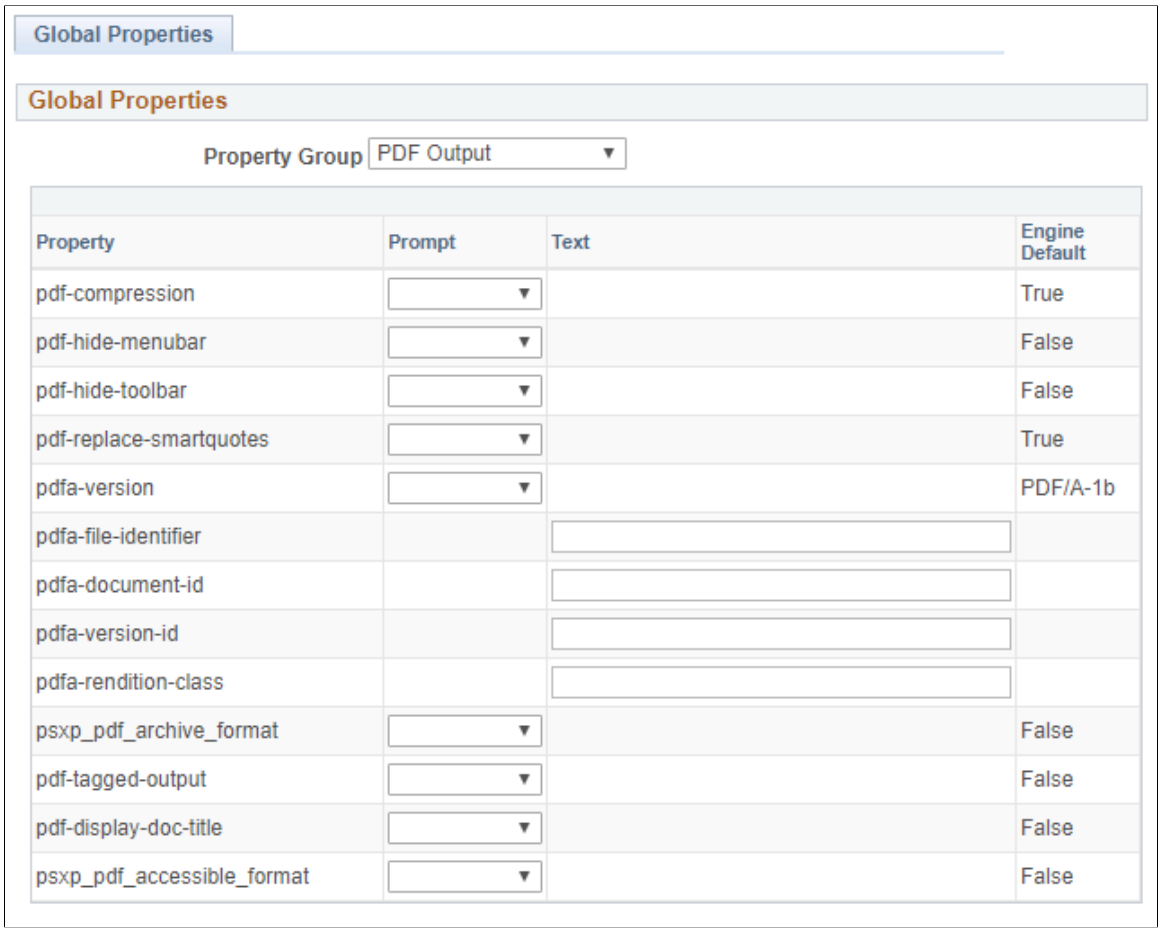
Access the Global Properties page (select Reporting Tools > BI Publisher > Setup > Global Properties.)

Note: To change the property setting at the global level and at the report-definition level, you require the BIP Report Developer role.

Copyright © 1988, 2020, Oracle and/or its affiliates.

Image: Global Properties page

This example illustrates the fields and controls on the Global Properties page. You can find definitions for the fields and controls later on this page.



Property Group

Select the property group. The options are:

- FO Processing
- HTML Output
- PDF Digital Signature
- PDF Output
- PDF Security
- PDF Submit Settings
- PDF Template
- PeopleTools Settings
- Printer Properties
- RTF Output
- RTF Template

Copyright © 1988, 2020, Oracle and/or its affiliates.

- XLS Output

Note: This section provides documentation for the PeopleTools-specific properties and any relevant core BIP properties that are associated with them.
For details on all the core BIP properties, see *Fusion Middleware Report Designer's Guide for Oracle Business Intelligence Publisher,* "Setting Report Processing and Output Document Properties."

Property All properties available for the property group selected appear.

Prompt Select the value for the property.

Default Displays the default value for the property.

Generating PDF/A Output

BI Publisher supports the generation of reports in the ISO standard PDF/A format for long-term preservation and archival of documents.

The properties to configure PDF/A output are located under the PDF Output property group.

psxp_pdf_archive_format Set this property to *True* to enable ISO standard PDF/A output. The default value is *False*.

pdfa-version This property allows you to specify which of the available PDF/A standards to use.

The valid values are:

- PDF/A-1a
- PDF/A-1b
- PDF/A-2a
- PDF/A-2b

The default value is *PDF/A-1b*. The PDF/A-2b standard supports the PDF/A-1b features, preserves transparency, uses compressed objects and XRef streams to make the PDF file size smaller, and allows for the use of PDF version higher than 1.4 (1.7 is used). The PDF/A-1a and PDF/A-2a standards adhere to all the requirements of the respective PDF/A specification including those pertaining to accessible PDF.

The following properties are used to specify the values of the corresponding XMP metadata in the PDF/A file:

pdfa-file-identifier One or more valid file identifiers set in the xmpMM:Identifier field of the metadata dictionary. To specify more than one identifier, separate values with a comma (,). When this property

 Copyright © 1988, 2020, Oracle and/or its affiliates.

	is not specified, an automatically generated file identifier is used.
pdfa-document-id	Valid document ID. The value is set in the xmpMM:DocumentID field of the metadata dictionary.
pdfa-version-id	Valid version ID. The value is set in the xmpMM:VersionID field of the metadata dictionary.
pdfa-rendition-class	Valid rendition class. The value is set in the xmpMM:RenditionClass field of the metadata dictionary.

To generate PDF/A output:

- The report template type must be RTF or XSL.
- The PDF/A standard requires that all fonts be embedded. By default, all fonts used in the template are automatically substituted with the delivered Albany font, unless font mapping is configured in the xdo.cfg file, in which case, the mapped fonts will be embedded in the resulting PDF. If the Albany font is missing from the JVM font directory and font mapping is not configured, the Adobe Base 14 font Helvetica is used as a non-embedded font and the PDF fails to be PDF/A compliant.
- PDF/A output is not supported when using the PDFMerger utility class, the External Attachment feature, or a Digital Signature. The resulting PDF file may no longer be PDF/A compliant.
- The PDF/A standard does not allow for encryption. Generating PDF/A format implicitly sets the property pdf-security to *False*.

Note: It is recommended to set these properties at the report definition level.

Generating Accessible PDF and PDF/UA Output

BI Publisher supports the generation of accessible PDF output. Accessible PDF output is a structured document that includes a document title and the PDF tags. The paragraph, link, image, table, and list elements are tagged in accessible PDF Documents.

The properties to configure accessible PDF output are located under the PDF Output property group.

pdf-tagged-output	This property ensures that PDF content is tagged appropriately. The default value is *False*.
pdf-display-doc-title	This property ensures that the PDF contains a document title. The default value is *False*.

The following property, under the PDF Security property group, controls whether text access for screen reader devices is allowed for encrypted PDF reports when the pdf-encryption-level property is set higher than one:

pdf-enable-accessibility	This property is applicable when pdf-encryption-level is set to one or higher. When set to *True*, text access for screen reader devices is enabled. The default value is *True*.

BI Publisher also supports the generation of reports in the ISO standard PDF/UA-1 (ISO 14289-1:2014) format. The property to configure PDF/UA output is located under the PDF Output property group, and

Copyright © 1988, 2020, Oracle and/or its affiliates.

when set to *True*, implicitly sets the properties pdf-tagged-output, pdf-display-doc-title and pdf-enable-accessibility to *True*. It is described here:

psxp_pdf_accessible_format Set this property to *True* to generate the PDF in ISO standard PDF/UA-1 format. The default value is *False*.

To generate Accessible PDF or PDF/UA output:

- The report template type must be RTF or XSL.
- The PDF/UA-1 standard requires that all fonts be embedded. By default, all fonts used in the template are automatically substituted with the delivered Albany font, unless font mapping is configured in the xdo.cfg file. If configured, the mapped fonts are embedded in the resulting PDF. If the Albany font is missing from the JVM font directory and font mapping is not configured, the Adobe Base 14 font Helvetica will be used as a non-embedded font and the PDF fails to be PDF/UA-1 compliant.
- See *Fusion Middleware Report Designer's Guide for Oracle Business Intelligence Publisher*, "Generating Accessible PDF Output," Requirements and Limitations, for design-time requirements when creating a template for an accessible PDF report.

Note: The properties psxp_pdf_archive_format and psxp_pdf_accessible_format are mutually exclusive. When either of these properties are set to *True* on the Global Properties page or on the Report Definition page, the other one is automatically set to *False*, if previously set. If both properties are set through the ReportDefn API method SetRuntimeProperties(), psxp_pdf_accessible_format takes precedence.

Note: It is recommended to set these properties at the report definition level.

PeopleTools Settings

The properties in PeopleTools Settings control report attributes that are specific to PeopleSoft implementation of BI Publisher.

psxp_pdf_optimized This property controls whether or not the core engine uses the "optimized" PDF Form Processing feature. This increases the efficiency and performance of PDF-template based reports, while disabling certain features. Valid values are:

- True.

  Enables core engine optimization for PDF-based reports. The optimized engine will provide better performance, while disabling certain PDF-template specific features such as repeated fields and editable fields.

  Note: Full path mapping can be used.

- False

  Uses the unoptimized engine (same as BIP server), which will enable repeated fields and editable fields in a PDF template.

  Allows using digital signature on PDF reports and creating PDF reports that can be updated.

 Copyright © 1988, 2020, Oracle and/or its affiliates.

Note: Full path mapping is not supported.

Note: The default setting for the psxp_pdf_optimized property is True at the global level, but you can override this setting at the report-definition level if required.

Note: The default behavior of PeopleSoft BIP PDF Form Processing engine that existed since 8.48.02 release is to enable multiple document output meaning that the PDF template is used to generate as many documents in the output as there are instances of the high level repeating node in the XML data. This uses the "performance optimized" engine introduced by the Oracle Core BIP product team specifically for PeopleSoft use. By design, the "performance optimized" engine does not support editable fields. The editable fields feature was eliminated in order to meet the necessary performance goals of the "performance optimized" engine. At the present time, to get the editable fields feature, the user must fall-back to the non-performance optimized engine by setting psxp_pdf_optimized=false.
Also by design, the "non-performance optimized" engine does not, and never did support producing multiple documents based on the PDF template.

See the table at the end of this section for a summary of the behavior differences when setting this property.

psxp_usedefaultoutdestination

This property is used to indicate that default processing directory is exposed to the OutDestination property even if this value has not been previously set. The default value is *False*.

- True

  A basic tools directory is exposed to the user, without showing an additional RptInst directory. This is the behavior in pre-8.50 BI Publisher.

  If this property is set to True and the user does not set value for OutDestination at runtime, then Tools will create an output file <Domain>\files\XMLP\123456789\RptInst \MyReport.HTM where 123456789 is for a directory name being generated with a random name. In this example the OutDestination property will return the value: <Domain> \files\XMLP\123456789.

  Some directories will not be cleaned up after processing is done and the report is delivered into Report Manager. Any empty directories that are left after the BI reports are delivered to the Report Manager will be cleaned up when the regularly scheduled Application Engine process PRCSYSPURGE runs. You can also run the Application Engine program PSXP_DIRCLN to clean up the directories.

Copyright © 1988, 2020, Oracle and/or its affiliates.

- False

  This is the default value. Querying the OutDestination property without previously setting it at runtime, will cause it to return blank. After the reports have been delivered to the report repository, the temporary files and directories used for processing will be deleted.

psxp_debug

This property controls whether or not to leave temporary files on the application server or the process scheduler server for debugging purpose. It is recommended to set this property at the report definition level to debug a specific report. Valid values are:

- True

  Temporary files will not be deleted from application server or process scheduler server for debugging purpose.

- False

  Temporary files are deleted from application server or process scheduler server.

Note: If this property is set to true, remember to change it back to False when debugging is completed.

---

psxp_nocdatafields

This property is used to indicate if character fields should not be wrapped in CDATA xml sections. By default, Query and Connected Query use CDATA xml sections for all character fields.

Use the Text field to indicate any fields that should not be wrapped in CDATA.

See Understanding Rich Text Editor Data in BI Reporting.

psxp_excel_outputformat

This property is used to indicate the Microsoft Excel output format of reports, that is whether you want to view the reports in the .xls or the .xlsx format. The valid values of this property are:

- XLS-HTML

  The report can be viewed in Microsoft Excel versions 2002 and higher.

- XLS-MHTML

  The report can be viewed in Microsoft Excel versions 2002 and higher.

- XLSX

  This is the default value. The report can be viewed in Microsoft Excel versions 2007 and higher.

Copyright © 1988, 2020, Oracle and/or its affiliates.

Note: You can override the global property setting for a specific report by changing the property setting at the report-definition level. For example, if you set the property to *XLS-MHTML* at the global level, you can override it for a specific report by choosing *XLSX* at the report-definition level.

psxp_number_format_src

This property allows you to modify the behavior regarding the personalization of number values in a report run online through a PIA page. It does not have any effect on scheduled reports.

The valid values are:

- *Browser.* (Default.)

  Formatting of a number value will be based on the PeopleSoft Regional Settings. The effective settings are determined after considering the user-specified overrides, browser-locale defaults, and system defaults, in that order.

- *LangCode.*

  Formatting of a number value will be based on the language code of the current user session. This is the behavior in PeopleTools 8.55 and earlier releases.

Note: This property does not have any effect on reports run through the process scheduler, as number formatting in a scheduled report is always be driven by the language code of the process run control. See Applying PeopleSoft Regional Settings Personalizations to BI Publisher Reports

psxp_ext_email_appclass

Enter the name of an application class to retrieve a list of external email addresses to which to send reports.

See Emailing BI Publisher Reports to External Users

psxp_cq_report_viewer

This property allows you to view BI Publisher reports based on a Connected Query data source, from the Query Report Viewer page. The default value is *False* at the global level, but you can override this setting at the report definition level if required. At the report definition level, this property is only visible for those report definitions that are associated with a Connected Query data source.

Note: If this property is set to *True*, a message appears to warn you that viewing reports based on a Connected Query data source, rather than scheduling them, may slow down the application server, and thus affect other online users.

psxp_expand_entityref

This property eliminates XML entity expansion (XML Bomb). The default value is False.

Copyright © 1988, 2020, Oracle and/or its affiliates.

Only a user with the Security Administrator role can see and modify this property setting on the Global Properties page. You cannot override this property setting at the report-definition level.

Note: PeopleSoft recommends that you do not change the default value of this property.

psxp_private_report_access

This property allows the sharing of a private report, so that it may be run by other users that also have a private query sharing the same name as the private query data source associated with the report. Note that report category permissions are enforced, so that only those users who have at least read-only permissions to the report can share the report.

The valid values are:

- Owner.(Default)

  Only the owner of the private Query data source can access and run this report.

- Shared.

  Users that have a private query that shares the same name as the private query data source associated with the report may also run the report, provided they have permissions to access the report based on its report category definition.

  These users also have read-only access to the report definition.

This table shows a summary of the differences in behavior when setting the psxp_pdf_optimized property:

 Copyright © 1988, 2020, Oracle and/or its affiliates.

psxp_pdf_optimized = true	*psxp_pdf_optimized = false*
1. Editable Fields NOT supported. 2. Repeated Fields NOT supported. 3. Full Path mapping supported. 4. Repeated document generation for each instance of high level repeating node in XML:. <?xml version="1.0"?> <root> <employee_details> repeating high l⇒ evel node <employee_id>1247</employee_id> <employee_name>Lesnar, Brock</emp⇒ loyee_name> … </employee_details> <employee_details> <employee_id>1248</employee_id> <employee_name>Couture, Randy</em⇒ ployee_name> … </employee_details> ... </root>	1. Editable Fields supported. 2. Repeated Fields supported. 3. Full Path mapping NOT supported. 4. Single document output for single high level node in XML: <?xml version="1.0"?> <root> <employee_details> single high leve⇒ l node <employee_info> <employee_id>3256</employee_id⇒ > <employee_name>Carano, Gina</e⇒ mployee_name> … </employee_info> <employee_address> <Street_Address>24 Park Avenue⇒ </Street_Address> <City>Richmond</City> <State>Virginia</State> … </employee_address> … </employee_details> </root> 5. Digital signature in PDF reports supported. 6. Supports PDFs that can be updated.

Printer Properties

The printer properties are described here:

PDF2X_COPIES

This property is used to indicate the number of copies to be printed. This property is translated to PCL number of copies command (<esc>&l#X) on the output. The default value is *1*.

Valid value for this property is an integer greater than 0.

PDF2X_MEDIA_SIZE_NAME

This property is used to specify the media size. When this property is set, media size in PDF is ignored. This property's value is translated to corresponding value of PCL page size command (<esc>&l#A).

Note: For PostScript output, the RTF template must be in the media size that you want for the final report output.

Valid values:

- A3

Copyright © 1988, 2020, Oracle and/or its affiliates.

- A4
- Com-10
- Executive
- International - b5
- International - c5
- International - dl
- Ledger
- Legal
- Letter
- Monarch

Media is selected based on page's media size.

PDF2X_PAGE_ORIENTATION

This property is used to specify the page orientation. This property is valid only when the pdf2x-media-size-name property is specified. The default page orientation is portrait.

Valid values:

- Landscape
- Portrait
- Reverse-landscape
- Reverse-portrait

PDF2X_PAGE_RANGES

This property is used to specify a range of pages to be printed. For example, 1, 3–5. The default value is all pages.

PDF2X_SIDES

This property is used to specify whether printing should be done on one side of paper only or both sides, and if it is both sides, which layout to use. This property does not work unless the printer has duplex unit.

By default, no values are set. A printer determines the sides for printing.

Valid values:

- Duplex
- One-sided
- Tumble

PDF2X_PAGE_WIDTH

This property is used to specify the width of a print media in points (1/72 inches). This property is valid only when the pdf2x-page-height property is also set. When this property

Copyright © 1988, 2020, Oracle and/or its affiliates.

	is set, media size in PDF is ignored. This property's value is translated to the corresponding value of PCL page size command (<esc>&l#A). The default value is the PDF page's width
PDF2X_PAGE_HEIGHT	This property is used to specify the height of a print media in points. This property is valid only when it is used in conjunction with the pdf2x-page-width property. When this property is set, media size in PDF is ignored. This property's value is translated to the corresponding value of PCL page size command (<esc>&l#A). The default value is the PDF page's height.
PDF2X_MEDIA_TRAY	Use this property to specify the media tray or bin for PCL and PostScript compatible printers. Valid values are: • *auto:* (Default.) Feed paper from the default paper tray as defined in the printer properties. • *manual:* Manual feed paper to the printer. • *tray1:* Feed paper from lower tray. • *tray2:* Feed paper from the printer-specified tray. • *tray3:* Feed from optional paper source. For output to PostScript printers, *auto* and *manual* values are supported for this property. For output to PCL printers, all values are supported. When this property is not specified, the printer selects the appropriate tray based on page size. This property is ignored if the pdf2x-media-size-name property is specified.
PDF2X_SHEET_COLLATE	This property is used to specify whether or not the media sheets of each copy of each printed document in a job are to be in sequence, when multiple copies of the document are specified by the pdf2x-copies property. Printer determines collation when this property is not specified and number of copies is greater than 1.
PDF2X_PS_CID_FONT _CONVTYPE	When converting to Postscript and if the PDF has embedded CID Font Type 2, this property determines whether to convert the font to Postscript Type0+Type42 fonts or Type0+Type3 fonts. Valid values are 3 and 42. The default value is 42.

Copyright © 1988, 2020, Oracle and/or its affiliates.

Note: Type42 font normally renders faster and produces better quality output. However, because of Postscript's internal limitation, an error may occur when the font has large number of glyphs (greater than 32768). Type3 font does not have the limitation, but it is slower on most printers. Type3 font may still hit memory limitation when font has large number of glyphs.

PCL_COLOR_SUPPORTED

This property is used to specify whether a PCL printer supports color prints. The default value is false.

Valid values:

- True

  When this property is set to true, the output will use PCL 5c, color PCL, commands. Text, vector, and raster graphics data are generated in RGB color. The output is printable only by color PCL printer supporting PCL 5c.

- False

  When this property is not set or set to false, the output will use PCL 5 commands only and color information is translated to black and white.

Note: Most PCL printers do not support PCL 5c unless these are color printers. Unlike Postscript printers which automatically translate color information to grayscale if printer only supports black and white, PCL 5c commands are not accepted by black and white PCL printers and sending PCL 5c file to black and white printer may produce garbage output.

PCL_EDGE_TO_EDGE _SUPPORTED

Unlike Postscript printers which allow use of the full extent of the media by default, PCL printers imposes some margins around printable area (see the Printable Area section of PCL 5 Printer Language Technical Reference Manual). However, some PCL printers support elimination of PCL margins around printable area by using the XEDGETOEDGE PJL (Printer Job Language) command. When this property is set to true, the XEDGETOEDGE PJL command is set to on. Use this property with pcl-page-scale=1.0 to get the original 100% size page output. This property works only on PCL printers that support XEDGETOEDGE PJL command, not on all the PCL printers.

Valid values are true or false. The default value is false.

PCL_X_ADJUSTMENT

Adjust the x position of objects on a page. This property is translated to the Left Offset Registration PCL command (<esc>&l#U). This is helpful in aligning position of text on pre-printed form, typically used for check printing.

The adjustment is in points.

Copyright © 1988, 2020, Oracle and/or its affiliates.

The default value is 0.

PCL_Y_ADJUSTMENT

Adjust the y position of objects on a page. This property is translated to the Top Offset Registration PCL command (<esc>&l#Z). This is helpful in aligning position of text on pre-printed form, typically used for check printing.

The adjustment is in points.

The default value is 0.

PDF Digital Signature

PDF Digital Signature properties are described here.

Note: The psxp_signature_appclass, psxp_signature_mapfields, and psxp_signature_digitalID properties are specific to the PeopleSoft implementation of BI Publisher.

signature-enable

Set this property to true to enable digital signature for PDF report.

The default value is False.

signature-field-name

This property reflects the digital signature field name. This property applies to PDF templates only.

If your report is based on a PDF template, you can enter a field in the PDF template where the digital signature will be placed.

For more information on designating a field for a signature in a PDF template, see *Report Designer's Guide for Oracle Business Intelligence Publisher, "Creating PDF Templates,* " Adding or Designating a Field for a Digital Signature."

signature-field-location

This property is applicable to RTF or PDF layout templates. The default value is top-left.

Valid values:

- Top-center
- Top-left
- Top-right

Note: If you set this property, do not set the X and Y coordinates or width and height properties.

signature-field-pos-x

This property is applicable to RTF or PDF layout templates. The default value is 0.

Using the left edge of the document as the zero point of the X axis, enter the position in points that you want the digital signature to be placed from the left. For example, if you want

Copyright © 1988, 2020, Oracle and/or its affiliates.

	the digital signature to be placed horizontally in the middle of an 8.5 inch by 11 inch document (which is 612 points in width and 792 points in height), enter 306.
signature-field-pos-y	This property is applicable to RTF or PDF layout templates. The default value is 0. Using the bottom edge of the document as the zero point of the Y axis, enter the position in points that you want the digital signature to be placed from the bottom. For example, if you want the digital signature to be placed vertically in the middle of an 8.5 inch by 11 inch document (which is 612 points in width and 792 points in height), enter 396.
signature-field-width	Enter in points (72 points equal one inch) the desired width of the inserted digital signature field. The default value is 0. This property should be set only if you set the X and Y coordinates properties.
signature-field-height	Enter in points (72 points equal one inch) the desired height of the inserted digital signature field. The default value is 0. This property should be set only if you set the X and Y coordinates properties.
psxp_signature_appclass	This property sets which application class should be called by BI Publisher to map a generated PDF document to a signing authority. The application class should be created by application development implementing the IPT_PDFSIGNATURE_INT:IPDFSignature interface. For example, APPS_PDFSIGNATURE_UNITTEST:PDFSignatureId. You must set this property if you do not specify a value for the psxp_signature_digitalID property. **Note:** This property is specific to the PeopleSoft implementation of BI Publisher.
psxp_signature_mapfields	A list of fields to be used by supplying data values to the application class. Values of these fields could be used to figure out the required signer ID. You can separate field values by using comma, semicolon, colon, pipe (|), or white space. This property is optional. **Note:** This property is specific to the PeopleSoft implementation of BI Publisher.
psxp_signature_digitalID	If you do not specify a value for the psxp_signature_appclass property, you must specify the digital ID of the signer directly.

Copyright © 1988, 2020, Oracle and/or its affiliates.

If you specify the digital ID of the signer, do not set the psxp_signature_appclass property.

If the psxp_signature_appclass property and the psxp_signature_digitalID property are set, the psxp_signature_digitalID property is ignored.

Note: This property is specific to the PeopleSoft implementation of BI Publisher.

Note: PeopleSoft recommends that all properties of the PDF Digital Signature property group are set at the report-definition level.

PDF Submit Settings

The properties of PDF Submit Settings are described here:

psxp_submit_enable

This property defines whether a PDF report can be updated. The default value is False.

At runtime, BI Publisher embeds Acrobat JavaScript into the template to create a Save or Submit button. The Save or Submit button may not be visible in a template, but appears on the PDF report.

If you set the value of this property to False, all other properties in the PDF Submit Settings property group are ignored.

psxp_submit_field_name

This property allows you to designate an existing field in the PDF template as the Save or Submit button.

If you set this property, BI Publisher will not automatically create a Save or Submit button. Instead, BI Publisher will use the designated field for the Save or Submit button.

psxp_submit_location

This property specifies the location of the Submit button. The default value is bottom-right.

Valid values:

- Bottom-left
- Bottom-right
- Top-left
- Top-right

psxp_submit_caption

This property reflects the message catalog entry and should be set in the following format:

message_set,message_number

The default value is 235,250.

Copyright © 1988, 2020, Oracle and/or its affiliates.

psxp_submit_tooltip	This property reflects the message catalog entry and should be set in the following format: message_set,message_number The default value is 235,251.
psxp_submit_width	Enter in points (72 points equal one inch) the desired width of the Submit button. The default value is 90.
psxp_submit_height	Enter in points (72 points equal one inch) the desired height of the Submit button. The default value is 18.
psxp_submit_appclass	This property specifies the application class name that saves submitted data to the database, and the application class name should be entered in the following format: Application_Package:Application_class For example, PDF_SUBMIT_TEST:PDFCONFIRM
psxp_submit_use_HTTPS	This property sets the network protocol (HTTP or HTTPS) that will be used while submitting a PDF report. The default value is True. Valid values: • True • False
psxp_submit_service_operation_alias	BI Publisher provides a default IB Service operation to be used for reports that can be updated. The default service operation is intended to be used mainly for self-service operations. If a report, which can be updated, is to be used for other purposes, you should provide an alternate IB service operation. The default value is PSXP_PDF_SUBMIT. The psxp_submit_service_operation_alias and the psxp_submit_service_operation_version properties enable you to specify an IB service operation.
psxp_submit_service_operation_version	Enter a value in the following format: v<VERSION_NUMBER> For example, v1.

Editing PDF Output

In previous releases the ability to edit PDF output was defined on the Report Definition Output page using the *PDF report output may be edited* check box. Starting with PeopleTools 8.50, this is now configured via properties set either on the Global Properties page for all reports or on the Report Properties page for a specific report.

 Copyright © 1988, 2020, Oracle and/or its affiliates.

To allow editing of PDF reports, the properties for the following property groups should be set as indicated:

PDF Security

Property	***Value***
pdf-security	True
pdf-encryption-level	2
pdf-changes-allowed	2 or higher

PDF Template

Property	***Value***
all-field-readonly	False

PeopleTools Settings

Property	***Value***
psxp_pdf_optimized	False

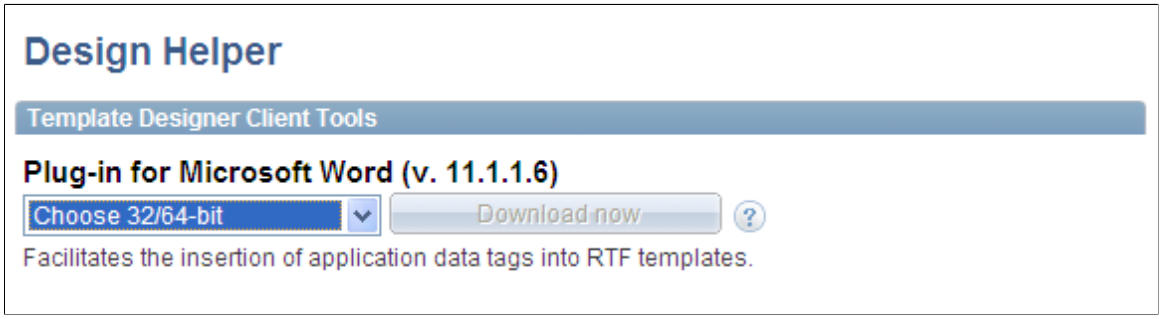
Note: It is recommended to set these properties at the report definition level.

Working with Template Design Helpers

Access the Design Helper page (Select Reporting Tools > BI Publisher > Setup > Design Helper.)

Image: Design Helper page

This example illustrates the fields and controls on the Design Helper page.



During template creation, a design helper facilitates the insertion of application data tag placeholders into your templates. A schema or sample data file is required for using a design helper. If you use a sample data file, you can preview your template offline during the design process.

Copyright © 1988, 2020, Oracle and/or its affiliates.

Two template design helpers are available: one for designing RTF reports (MS Word Template Builder) and one for the PDF template that gets embedded as a JavaScript inside the PDF template itself when you upload the template to PeopleTools.

This page enables users to download a plug-in for Microsoft Word to facilitate offline RTF template design. Two versions of the plug-in are available: one for 32–bit Word and another for 64–bit Word. Select a version that applies from the list to enable the download button.

The plug-in is downloaded as a zip file. Based on your selection, the zip file contains either BIPublisherDesktop32.exe or BIPublisherDesktop64.exe that should be run to install the plug-in.

Included in the BI Publisher desktop plug-in setup, multiple directories are created that contain documentation, samples, tutorial, and demos. The program is also added to the Start menu.

See Using PDF Templates.

Assigning BIP Permissions to Users

BI Publisher menu access is permission-list driven. This table describes the delivered roles and permission lists for that provide access to BI Publisher components and features.

Role	*Permission List*	*Description*
XMLP Report Developer	PTPT2600	Users can access all BI Publisher components, including setup capability on the advanced feature Report Definition Bursting page.
XMLP Power User	PTPT2500	Users can run reports based on query data sources using Query Report Viewer and Query Report Scheduler. Users can also access report definitions and the Content Library. However, access to the report definition bursting information is read-only. This role does not have access to the Report definition properties. **Note:** This role and permission list are supported for backwards compatibility. Assign users other roles and permission lists described in this table.
XMLP Report User	PTPT5300	Users can run and schedule reports. Users can also open BI Publisher report definitions in read-only mode.
NA	PTPT1000	Users can access the BI Publisher Report Repository

This table describes the component access provided by each of the BI Publisher roles and permission lists.

Component	*XMLP Report Developer (PTPT2600)*	*XMLP Power User (PTPT2500)*	*XMLP Report User (PTPT5300)*	*PeopleSoft User (PTPT1000)*
Report Category	Yes	No	No	No
Design Helper	Yes	Yes	No	No
Global Properties	Yes	No	No	No

 Copyright © 1988, 2020, Oracle and/or its affiliates.

Component	***XMLP Report Developer (PTPT2600)***	***XMLP Power User (PTPT2500)***	***XMLP Report User (PTPT5300)***	***PeopleSoft User (PTPT1000)***
Data Source	Yes	No	Read-only	No
Report Definition	Yes	Yes **Note:** Display-only access for bursting. **Note:** Report properties page is not available.	Read-only	No
Content Library	Yes	Yes	Read-only	No
Template Translations	Yes	No	Read-only	No
Query Report Viewer	Yes	Yes	Yes	No
Query Report Scheduler	Yes	Yes	Yes	No
Report Repository	Yes	Yes	Yes	Yes

Copyright © 1988, 2020, Oracle and/or its affiliates.

Copyright © 1988, 2020, Oracle and/or its affiliates.

Chapter 3

Creating and Registering Data Sources

Creating Data Sources

This section provides an overview of data generation and discusses how to create schema and sample data.

Understanding Data Generation

In BI Publisher, the data extraction is separate for the data presentation. Sample data can be used to design your RTF template and map your PDF templates. The data schema file is an XML Schema Definition (XSD) file that defines the structure and elements in the extracted XML. Data schema was used in previous releases for the bursting feature and is still available for backwards compatibility and bursting. If PeopleSoft queries are used for data extraction, the system will generate the schema; for all other data sources, you must create the data schema using tools outside of the PeopleSoft system. Schemas are also used for data transformation.

See Using Data Transform.

BI Publisher can register PS/Query, Connected Query, XML, and Composite Query files as a data source, but you can generate XML data using any means including PS/Query, SQR, Application Engine, PeopleCode, File Layout, and so forth.

For RTF template-based reports, design your data source XML structure using groupings that resemble the groupings needed for the output report. This improves runtime performance by preventing unnecessary grouping by the formatting engine.

Creating Schema and Sample Data

Use sample data source information for developing your RTF report templates, defining bursting, and mapping your PDF templates.

Storing the sample data file in PeopleTools provides a means to:

- Insert form field tags in RTF templates
- Conduct PDF mapping.
- Choose the bursting field during design time.
- Preview the template.

Sample Data File

Requirements for the structure of XML sample data file include:

- Must consist of a root node with one repeating group.

Copyright © 1988, 2020, Oracle and/or its affiliates.

Textual elements in this repeating group are candidates for bursting.

- Elements should have textual content.

  Element should not be empty.

- All expected elements must be included.

  All text elements should contain default values. All defined elements can be used for mapping.

This is an example of a sample XML file used as a data source:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
   <taxroot>
      <Box_Year>2005</Box_Year>
      <Box_Form>W2</Box_Form>
      <EE_SSN>111-11-1111</EE_SSN>
      <ER_EIN>ER_111111111</ER_EIN>
      <Employee>
          <EE_FirstName>Incheol</EE_FirstName>
          <EE_LastName>Kang</EE_LastName>
          <EE_Address1>500 Oracle Parkway</EE_Address1>
          <EE_Address2>Redwood Shores</EE_Address2>
          <EE_Address3>CA 94065</EE_Address3>
      </Employee>
      <Employer>
          <ER_Name>Oracle USA</ER_Name>
          <ER_Address1>500 Oracle Parkway</ER_Address1>
          <ER_Address2>Redwood Shores</ER_Address2>
          <ER_Address3>CA 94065</ER_Address3>
      </Employer>
      <Tax>
          <Fed_Wages_COR>20000</Fed_Wages_COR>
          <Fed_Tax_COR>20000</Fed_Tax_COR>
         <RETIRE_EE_PRV>Y</RETIRE_EE_PRV>
      </Tax>
   </taxroot>
   </root>
```

Note: In this example, the elements *Box_Form*, *Box_Year*, *EE_SSN* and *ER_EIN* are available as burst candidates. All of the elements will be available for mapping.

The actual data file may contain repeated instances of the high level repeating group, as shown in this example:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
   <taxroot>
      <Box_Year>2005</Box_Year>
      <Box_Form>W2</Box_Form>
      <EE_SSN>111-11-1111</EE_SSN>
      <ER_EIN>ER_111111111</ER_EIN>
      <Employee>
          <EE_FirstName>Incheol</EE_FirstName>
          <EE_LastName>Kang</EE_LastName>
          <EE_Address1>500 Oracle Parkway</EE_Address1>
          <EE_Address2>Redwood Shores</EE_Address2>
          <EE_Address3>CA 94065</EE_Address3>
      </Employee>
      <Employer>
          <ER_Name>Oracle USA</ER_Name>
          <ER_Address1>500 Oracle Parkway</ER_Address1>
          <ER_Address2>Redwood Shores</ER_Address2>
          <ER_Address3>CA 94065</ER_Address3>
```

 Copyright © 1988, 2020, Oracle and/or its affiliates.

```
      </Employer>
      <Tax>
          <Fed_Wages_COR>20000</Fed_Wages_COR>
          <Fed_Tax_COR>20000</Fed_Tax_COR>
         <RETIRE_EE_PRV>Y</RETIRE_EE_PRV>
      </Tax>
   </taxroot>
   <taxroot>
       <Box_Year>2005</Box_Year>
       <Box_Form>W2</Box_Form>
       <EE_SSN>2222-22-2222</EE_SSN>
       <ER_EIN>ER_222222222</ER_EIN>
       <Employee>
           <EE_FirstName>Chang</EE_FirstName>
           <EE_LastName>Yu</EE_LastName>
           <EE_Address1>500 Oracle Parkway</EE_Address1>
           <EE_Address2>Redwood Shores</EE_Address2>
           <EE_Address3>CA 94065</EE_Address3>
       </Employee>
       <Employer>
            <ER_Name>Oracle USA</ER_Name>
            <ER_Address1>500 Oracle Parkway</ER_Address1>
            <ER_Address2>Redwood Shores</ER_Address2>
            <ER_Address3>CA 94065</ER_Address3>
       </Employer>
       <Tax>
            <Fed_Wages_COR>10000</Fed_Wages_COR>
            <Fed_Tax_COR>10000</Fed_Tax_COR>
            <RETIRE_EE_PRV>Y</RETIRE_EE_PRV>
       </Tax>
     </taxroot>
   </root>
```

Schema File

This is the sample schema for the XML file shown previously:

```
<?xml version="1.0"?>
<xs:schema id="root" targetNamespace="http://tempuri.org/example_xml.xsd"
xmlns:mstns="http://tempuri.org/example_xml.xsd"
xmlns="http://tempuri.org/example_xml.xsd"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:msdata="urn:schemas-microsoft-com:xml-msdata"
attributeFormDefault="qualified" elementFormDefault=
"qualified">
  <xs:element name="root" msdata:IsDataSet="true"
    msdata:EnforceConstraints="False">
    <xs:complexType>
      <xs:choice maxOccurs="unbounded">
        <xs:element name="taxroot">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="Box_Year" type="xs:string" minOccurs="0"/>
              <xs:element name="Box_Form" type="xs:string" minOccurs="0"/>
              <xs:element name="EE_SSN" type="xs:string" minOccurs="0"/>
              <xs:element name="ER_EIN" type="xs:string" minOccurs="0"/>
              <xs:element name="Employee" minOccurs="0" maxOccurs="unbounded">
                <xs:complexType>
                  <xs:sequence>
                    <xs:element name="EE_FirstName" type="xs:string"
                     minOccurs="0"/>
                    <xs:element name="EE_LastName" type="xs:string"
                     minOccurs="0"/>
                    <xs:element name="EE_Address1" type="xs:string"
                     minOccurs="0"/>
                    <xs:element name="EE_Address2" type="xs:string"
                     minOccurs="0"/>
                    <xs:element name="EE_Address3" type="xs:string"
                     minOccurs="0"/>
                  </xs:sequence>
```

Copyright © 1988, 2020, Oracle and/or its affiliates.

```
                </xs:complexType>
              </xs:element>
              <xs:element name="Employer" minOccurs="0" maxOccurs="unbounded">
                <xs:complexType>
                  <xs:sequence>
                    <xs:element name="ER_Name" type="xs:string" minOccurs="0"/>
                    <xs:element name="ER_Address1" type="xs:string"
                     minOccurs="0"/>
                    <xs:element name="ER_Address2" type="xs:string"
                     minOccurs="0"/>
                    <xs:element name="ER_Address3" type="xs:string"
                     minOccurs="0"/>
                  </xs:sequence>
                </xs:complexType>
              </xs:element>
              <xs:element name="Tax" minOccurs="0" maxOccurs="unbounded">
                <xs:complexType>
                  <xs:sequence>
                    <xs:element name="Fed_Wages_COR" type="xs:string"
                     minOccurs="0"/>
                    <xs:element name="Fed_Tax_COR" type="xs:string"
                      minOccurs="0"/>
                    <xs:element name="RETIRE_EE_PRV" type="xs:string"
                     minOccurs="0"/>
                  </xs:sequence>
                </xs:complexType>
              </xs:element>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:choice>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

Note: Schema and sample data are generated for the PeopleSoft Query data source.

Registering Data Sources

This section provides an overview of data source registration and discusses how to register data sources.

Understanding Data Source Registration

A data source registers the schema and sample data design files. The extracted application fields from the data source files are placed into the template files to create the final report.

The data source can be PS Query, Connected Query, Composite Query, or XML files.

Note: PeopleSoft queries with in tree prompts are not allowed as a data source.

For PS Query data sources, BI Publisher runs RunToFile() to obtain the XML data for a report. The RunToFile function uses the language code specified in a user's profile to obtain the data. If you want to run a report that requires to be translated, you must create a user and in the user profile select the language code for which the report has to be translated. Use this user to run the report for translation. For example, if you want a report to be translated into Spanish, you need to select the language code for Spanish in the user profile, and use this user to run the report.

Benefits of data source registration include the ability to:

Copyright © 1988, 2020, Oracle and/or its affiliates.

1. Reuse previously registered data sources with multiple report definitions.
2. Take advantage of built-in bursting features.

   See Setting Bursting Options.

Note: When creating a report definition, you can select a PS Query, Connected Query, or Composite Query data source that has not yet been registered and that data source is registered automatically when you save the report definition. However, all other types of data sources must be registered before they can be associated with a report definition.

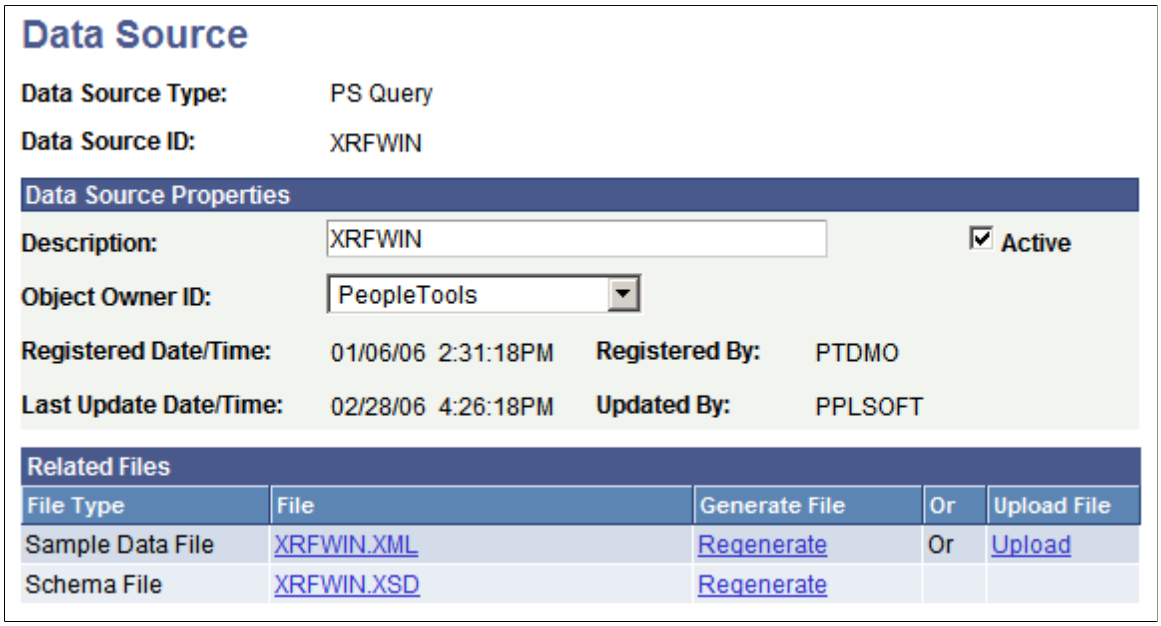
Note: You can check the format of an XML output file by opening it using Microsoft Internet Explorer (IE). IE opens the file and alerts you to any problems, such as unclosed tags.

Registering Data Sources

Access the Data Source page (Select Reporting Tools > BI Publisher > Data Source.)

Image: Data Source page

This example illustrates the fields and controls on the Data Source page. You can find definitions for the fields and controls later on this page.



Data Source Type Select *PS Query*, *Connected Query*, or *XML File*, or *Composite Query*.

Data Source ID Select or enter the data source ID.

When adding a new data source of type PS Query, Connected Query, or Composite Query, the Data Source ID will always be the name of the PS Query, Connected Query, or Composite Query definition that you select.

Copyright © 1988, 2020, Oracle and/or its affiliates.

	For other data source types, this field accepts free-form text entry. Enter an ID that indicates what the data is, because you want to easily identify your data sources when defining reports.
Description	(Optional) Enter descriptive text that provides more detail about the data source. The description is automatically supplied by default to the data source ID.
Object Owner ID	(Optional) Indicate which product, feature, or application owns this data source. This field is used to extract and package production data source and report registrations and their supporting files.
Registered Date/Time	This is a read-only field maintained by the system that indicates the date that the initial data source registration was made.
Last Update Date/Time	This is a read-only field maintained by the system that indicates the date that the last update to the data source was made.
Active	Select to indicate that this is an active data source. Only active data sources can be chosen when creating a new report definition. Only reports with active data sources can be processed.
Registered By	This is a read-only field maintained by the system that indicates the user ID of the operator who initially registered the data source.
Updated By	This is a read-only field maintained by the system that indicates the user ID of the operator who last updated the data source.

Related Files

The sample data file is an XML file with sample data that is used for data mapping, template preview, and determining burst fields. Preview action is available within a desktop template designer or from within the report definition page. For PS Query, Connected Query, and Composite Query data source types the sample data file can be system-generated or uploaded. For XML file data source type, the sample data file must be uploaded.

Note: Prior to PeopleTools 8.50, the sample data field was only used for data mapping and preview and the schema file was used required for bursting. In PeopleTools 8.50, the sample data file is used for bursting, as well as data mapping and preview.

File	(Optional) Click the file name links to view the XML and XSD files after you have generated, regenerated, or uploaded them.
Last Update Date/Time	(Optional) This is a read-only field maintained by the system that indicates the date that the last update to the related file was made.

Copyright © 1988, 2020, Oracle and/or its affiliates.

Generate/Regenerate

(Optional) Click the Generate link for PS Query, Connected Query, or Composite Query data sources to generate the related sample data .

When the related files have been initially generated for PS Query, Connected Query, or Composite Query data sources, click the Regenerate link to regenerate them in case the underlying query has changed.

Upload

(Optional) Click the Upload link for XML file data sources to bring the related sample data and schema files into the database.

You can also upload a sample data file for PS Query, Connected Query, or Composite Query if you would prefer to use a sample data file with more realistic data.

Note: A validation is run against the schema XSD file that is uploaded to a data source, alerting the developer if problems occurred while the system was using their schema.

Copyright © 1988, 2020, Oracle and/or its affiliates.

Copyright © 1988, 2020, Oracle and/or its affiliates.

Chapter 4

Creating Report Templates

Understanding Report Template Types

Template design involves the construction of a report layout in a template file and is dependent upon what the core Oracle BI Publisher engines accept for processing.

The nature of the data plays a role in the selection of a template.

The following table lists and describes supported template types and provides guidelines for you to consider:

Template Type	***Description***
PDF Template	Reports are pre-rendered PDF forms that are populated with data at runtime. Starting in PeopleTools 8.50 nested structures are supported. Nested structures should not be used for any reports that need to be backwards compatible. This template type is suitable when you have existing PDF forms that you need to use to generate reports, such as government forms. Generally, using this template type is faster than using RTF templates because no runtime rendering is involved. Use PDF templates when: • You already have PDF templates that you must use (for example, government forms). • You have simple form-based reporting requirements with no complex formatting, that is, no charting, dynamic tables, dynamic repeated fields, and so forth.
RTF Template	Reports are full rendered, which means that the actual output is generated at runtime using XSLFO technology. Report designers have full control of output formatting and can incorporate charts, dynamic tables, conditional formatting, and so forth. Reports generation is generally slower than PDF-based reports because they involve real-time output rendering.

Copyright © 1988, 2020, Oracle and/or its affiliates.

Template Type	*Description*
eText	eText templates are RTF-based templates that are used to generate flat-file text output that can be transmitted to a bank or other customer for Electronic Funds Transfer (EFT) or Electronic Data Interchange (EDI). Because the output is intended for electronic communication, these templates must follow specific format instructions for data placement. **Note:** XML file and Connected Query are the recommended data sources for eText templates because the requirements for eText templates are very specific. XML produced by PS Query data sources lacks the required structure for eText templates and is therefore not available. See *Fusion Middleware Report Designer's Guide for Oracle Business Intelligence Publisher*, "Creating eText Templates."
XSL Templates	For more complex design requirements, a number of XSL and XSL-FO elements are supported for use with your XSL templates. See *Fusion Middleware Report Designer's Guide for Oracle Business Intelligence Publisher*, "Supported XSL-FO Elements."

Note: Sample report templates are bundled with the BI Publisher Desktop, and available in <Installation Directory>\BI Publisher Desktop\samples.

Using RTF Templates

RTF templates support most XSL functionality and can be designed with robust processing and formatting code.

This section discusses how to:

- Create RTF templates.
- Incorporate sub-templates.
- Include images.
- Change default template font.
- Use drilling URL in RTF template.
- Incorporating data created with Rich Text Editor (RTE) into template.
- Embedding PCL code into template.
- Using two dimensional (2D) barcode functions.
- Handle sensitive data in RTF templates.

Copyright © 1988, 2020, Oracle and/or its affiliates.

Creating RTF Templates

To create an RTF template using Microsoft Word:

1. Download the delivered BI Publisher Template Builder plug-in for offline template design on the Reporting Tools, BI Publisher, Setup, Design Helper page to facilitate the insertion of application data tags into your RTF templates.

   The BI Publisher Template Builder is an extension to Microsoft Word that simplifies the development of RTF templates. While the Template Builder is not required to create RTF templates, it provides many automated functions that may increase your productivity.

   Note: You can choose to automatically view the Word Template Builder Tutorial File, Template Builder for Word Tutorial.doc, upon installing the plug-in. This document offers a quick and informative tutorial of the Template Builder.
   The Template Builder for Word Tutorial.doc is located in the \Template Builder for Word\doc directory of the folder where Oracle BI Publisher Desktop, BI Publisher Template Builder plug-in, was installed.
   Sample report templates are available in <Installation Directory>\BI Publisher Desktop\samples.

   ---

2. Download the XML sample data file by clicking the Sample Data link on the Reporting Tools, BI Publisher, Report Definition page for a specified query.

3. Load the sample data into the document by selecting Data > Load XML Data from the Microsoft Word Template Builder tool bar menu.

4. Design your template in the RTF document.

   By using the downloaded XML sample data, you can insert the data field tags into your template rather than manually typing XSL-formatted tags.

   You can preview the template output with the sample XML data from the Oracle BI Publisher menu using Preview Template or select Preview from the Template Builder toolbar.

5. Upload the completed template into the report definition by clicking the Upload button on the Reporting Tools, BI Publisher, Report Definition, Template page.

Note: Your data source XML structure should be designed to be as close as possible to the groupings used for in the actual report template structure; this improves runtime performance by preventing unnecessary XSL transformation. This is particularly applicable for reports with complex data structures and very large file sizes.

See *Fusion Middleware Report Designer's Guide for Oracle Business Intelligence Publisher*, “Creating RTF Templates.”

Refer to Word Template Builder Tutorial.

See Working with Template Design Helpers, Creating Report Definitions.

Incorporating Sub-Templates

When designing a template, you can incorporate one or more sub-templates into your primary template.

You must use specific syntax to:

Copyright © 1988, 2020, Oracle and/or its affiliates.

- Create sub-templates.
- Import sub-templates.
- Call sub-templates.

See Maintaining Sub-Templates.

Creating Sub-Templates

Within a single sub-template file, multiple sub-template components can be available. Start and end template indicators must exist to distinguish these various components.

```
<?template:component_name?>
<?end template?>
```

For example, syntax of a sub-template file containing two components could be:

```
<?template:peoplesoft?>
Pleasanton Campus
500 Oracle Lane
Pleasanton, CA  94488
<?end template?>

<?template:logo2x.5?>
 Oracle_Logo
<?end template?>
```

where `<?template:peoplesoft?>` is the start template indicator of the component *peoplesoft* and`<?template:logo2x.5?>` is the start template indicator of the component *logo2x.5*. Each`<?end template?>` tag indicates the end of its respective component.

Importing Sub-Templates

To import a sub-template file that is stored in the Content Library, place the following syntax at the top of the primary template file:

```
<?import:psxmlp://sub-template_NAME?>
```

where `sub-template_NAME` is the registered sub-template ID in the Content Library.

For example:

```
<?import:psxmlp://STDHEADER?>
```

This syntax must be in `Normal` text.

Note: The sub-template reference is defined only in the RTF template. The sub-template must be defined in Content Library; however, the relationship to templates using the sub-template is not defined in the database. Developers must be aware of the sub-template relationships when modifying the RTF sub-template.

See Maintaining Sub-Templates.

Copyright © 1988, 2020, Oracle and/or its affiliates.

Calling Sub-Templates

Place the following syntax in the primary template file in the location where the desired text or XSL instructions from the sub-template file should appear:

```
<?call-template:peoplesoft?>
```

In the preceding sample code `peoplesoft` is the name of the component that you want to use in the sub-template file.

Note: Primary templates calling nonexistent or inactive sub-templates cause an error message to be issued indicating the reason for the problem. This error information is incorporated into Process Scheduler error handling as well as into online viewing or previewing of the report.

See Running BI Publisher PeopleSoft Query Reports.

Testing a Sub-Template in Microsoft Word

You should test your template and sub-template using Template Builder before uploading to PeopleTools to make your sub-template is accessible to your template on the file system.

Use the following syntax when importing:

```
<?import:file:C:///Template_Directory/subtemplate_file.rtf?>
```

Notice the triple slashes and the use of the actual file name instead of template ID.

When your design is complete, you can change the import statement back to make the sub-template available to the main template in PeopleTools environment:

Note: In order to successfully import the sub-template file in Word, select BI Publisher menu, Options, Preview tab, and open the Properties table. Search for and set the property xdk-secure-io-mode to *False*. This property is set to *True* by default, as a security precaution.

Including Images

BI Publisher supports a number of methods for including images in your reports:

- Inserting images.
- Importing images.
- Rendering image field data (BLOB).

Inserting Images

To directly insert a .jpg, .gif, or .png image file into a template:

1. Select Insert > Picture > From File while the template is open in Microsoft Word.
2. Select the desired .jpg, .gif, or .png file to insert into the template.
3. Save the template.

Copyright © 1988, 2020, Oracle and/or its affiliates.

Note: Oracle recommends that you use the Microsoft Word *Insert* menu option to insert the image, because the additional properties that you need to set for the RTF template to correctly generate reports with those images are automatically set by means of this method. Additionally, dragging and dropping an image onto a template creates a link to the local machine being used and may cause problems when the report is generated.

Importing Images

To import an image from a sub-template file:

1. Embed the .jpg, .gif, or .png into the sub-template file.

   For example,

   ```
   <?template:logo2x.5?>
    Oracle_Logo
   <?end template?>
   ```

   where `Oracle_Logo` is the actual .jpg, .gif, or .png.

2. Import the sub-template file that includes the image by including the following syntax at the top of the primary template file:

   ```
   <?import:psxmlp://sub-template_NAME?>
   ```

   In this code sample, `sub-template_NAME` is the registered sub-template ID in the Content Library.

3. Add the calling details in the primary template at the appropriate location using the following syntax:

   ```
   <?call-template:logo2x.5?>
   ```

   In this code sample, `logo2x.5` is the name of the component that contains the image in the sub-template file.

See Incorporating Sub-Templates.

Rendering Image Field Data (BLOB)

BI Publisher supports the rendering of BLOB (Base64 encoded) data fields in an XML file.

To render an image at runtime, add the following syntax in a Form Field Help Text box :

```
<fo:instream-foreign-object content-type="image/jpg">
<xsl:value-of select="IMAGE_ELEMENT"/>
</fo:instream-foreign-object>
```

where `image/jpg` is the MIME type of the image (other MIME types are image/gif and image/png) and `IMAGE_ELEMENT` is the element containing BLOB (Base64 encoded) data in the XML file.

Also, you can specify height and width attributes for the image to set its size in the published report. You can specify height and width attributes in inches, pixels, centimeters, or by percentage. BI Publisher scales the image to fit the size that you define. The following examples illustrate the attributes for height and width of an image:

Image size attributes in inches:

```
<fo:instream-foreign-object content-type="image/jpg" height="3 in" width="4 in">
```


Copyright © 1988, 2020, Oracle and/or its affiliates.

Image size attributes in pixels:

```
<fo:instream-foreign-object content-type="image/jpg"
height="300 px" width="400 px">
```

Image size attributes in centimeters:

```
<fo:instream-foreign-object content-type="image/jpg"
height="3 cm" width="4 cm">
```

Image size attributes as a percentage of the original dimensions:

```
<fo:instream-foreign-object content-type="image/jpg"
height="300%" width="300%">
```

Note: BI Publisher supports rendering of images in all output formats—PDF, HTML, RTF, and XLS.

Query and Connected Query data sources containing image fields generate the binary base64 encoded data string for these image fields in the generated XML file provided the query's Image Fields property is set correctly.

If you are using a Query or Connected Query data source, follow these steps for the underlying query containing the image field:

1. Open a query definition (Reporting Tools, Query, Query Manager).
2. Click the Properties link.
3. In the Image Fields group box, select Image Data.
4. Click OK, and save the query.

See "Viewing and Editing Query Properties" (PeopleTools 8.58: Query), Setting Image Fields.

Changing Default Template Font

The output report from RTF template uses template-level default fonts for empty report spaces and empty table cells. If the default font size does not match the font height used in a template, a final report could look different from user expectations. In this case, the user can change the template default font either in design time or runtime:

- Design time

  Set the xdo.cfg for the font. For example, set the default font for a specific report to be Helvetica, size 8: `<property name="rtf-output-default-font">Helvetica:8</property>`

- Runtime

  Use PeopleCode to set the font. For example, set the default font for a specific report to be Times New Roman with height 10:

  ```
  &asPropName = CreateArrayRept("", 0);
  &asPropValue = CreateArrayRept("", 0);
  &asPropName.Push(""rtf-output-default-font");
  &asPropValue.Push(""Times New Roman:10");
  &oRptDefn.SetRuntimeProperties(&asPropName, &asPropValue);
  ```

Copyright © 1988, 2020, Oracle and/or its affiliates.

Using Drilling URL in RTF Template

Drilling URLs are supported in BI Publisher reports with a data source of PS Query or Connected Query.

Note: Drilling URLs are supported only in RTF templates.

To use a drilling URL in a BI Publisher report:

1. Create the query with the drilling URL defined as a field.

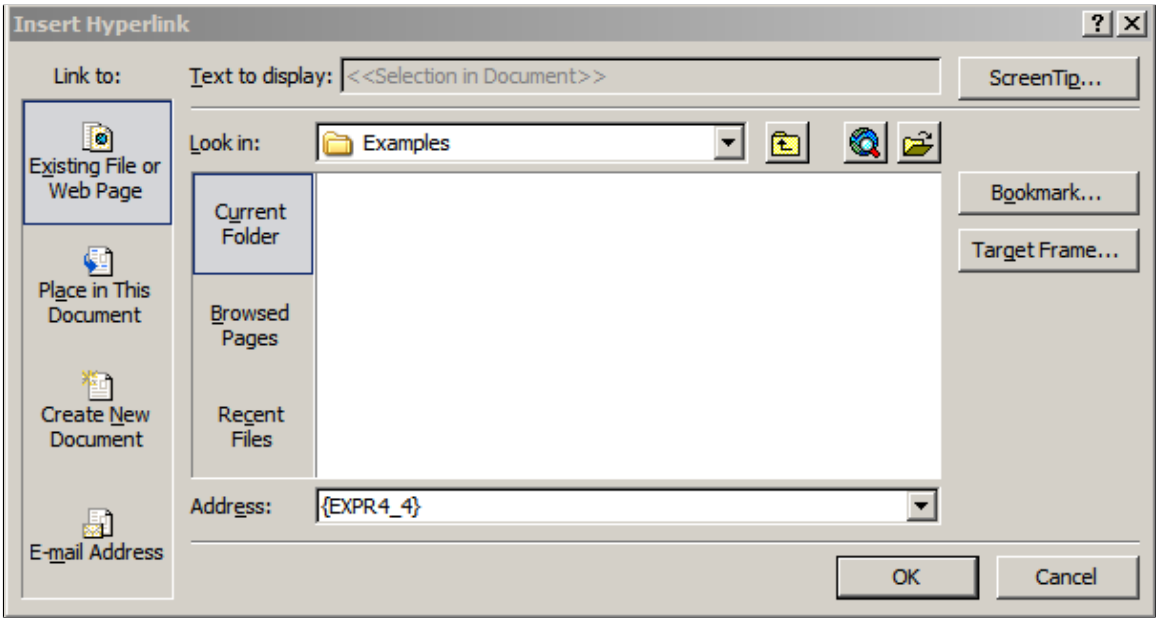
   See "Defining Query Drilling URLs" (PeopleTools 8.58: Query)

2. Create an RTF template.
3. In the RTF template map one or more fields to the fields that contain drilling URL.
   - Highlight the field where you want to place the drilling URL.
   - Select Insert (from the Word menu), Hyperlink or use Ctrl+K.
   - In the Insert Hyperlink dialog box, enter the URL link in Address field.

     Each URL link should be defined as {URL_FIELD}, where URL_FIELD is the unique field name for the expression that contains a specific drilling URL.

     Image: Insert Hyperlink dialog box

     This example illustrates the fields and controls on the Insert Hyperlink dialog box.



   - Use the Target Frame push button to select how this URL link will be opened, either in the same window or in the new window

Note: The URL value does not need to be added to the report, as long as the unique field name (data file that contains the URL) is mapped to a field in the report.

Copyright © 1988, 2020, Oracle and/or its affiliates.

4. If the BI Publisher report is run in Process Scheduler using an application engine program, you will need to add additional code to identify the process instance for the application engine program before processing the report. The process instance can be retrieved from the state record PSXPQRYRPT_AET. This call is needed to set a drilling URL during Query or Connected Query execution.

```
&ProcessInstance=PSXPQRYRPT_AET.PROCESS_INSTANCE;
&oRptDefn.ProcessInstance = &ProcessInstance;
&oRptDefn.ProcessReport("", "", %Date, "");
```

5. When you click the drilling URL in the report depending on the drilling URL type, one of the following occurs:

   - For Query URL, the Query results are displayed.
   - For Component URL, the appropriate PeopleSoft page is opened.
   - For External URL, the external page is opened.

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

Incorporating Data Created with Rich Text Editor (RTE) into Template

The data entered in an RTE (Rich Text Enabled) long edit field is stored in the PeopleSoft database as formatted HTML data. BI Publisher reports are capable of displaying this HTML formatted data in the output report without requiring any special conversion, as had been required in previous PeopleTools releases. The Oracle BI Publisher core engine natively supports the use of HTML formatted data fields in an RTF report template.

See Understanding Rich Text Editor Data in BI Reporting.

Embedding PCL Code into Template

You can embed custom Printer Command Language (PCL) code to enable font selection for secure check printing (that is, to invoke the MICR font used for bank account codes and the check's secure signature fonts). To embed PCL commands in the file that is printed, use the BI Publisher commands described below in your RTF template in the specific position on the page where you want the PCL commands to render.

RTF templates can be created with PCL hardcoded lines, which means that a user can directly specify the exact PCL command to be included at a specific visual position on a PCL page.

In the template, include the following syntax at the position where you want to print the text.

```
<pcl><control><esc/>(pcl command)</control>(text or data field)<sp/>
(text or data field)</pcl>
```

where:

<*pcl*><*/pcl*>: indicates the start and end of the custom PCL command sequence and the text to print using the custom command.

<*control*><*/control*>: indicates the start and end of the PCL sequence.

<*esc/*>: include <esc/> between <control> and </control> to escape character (ASCII 0x1b) in the output.

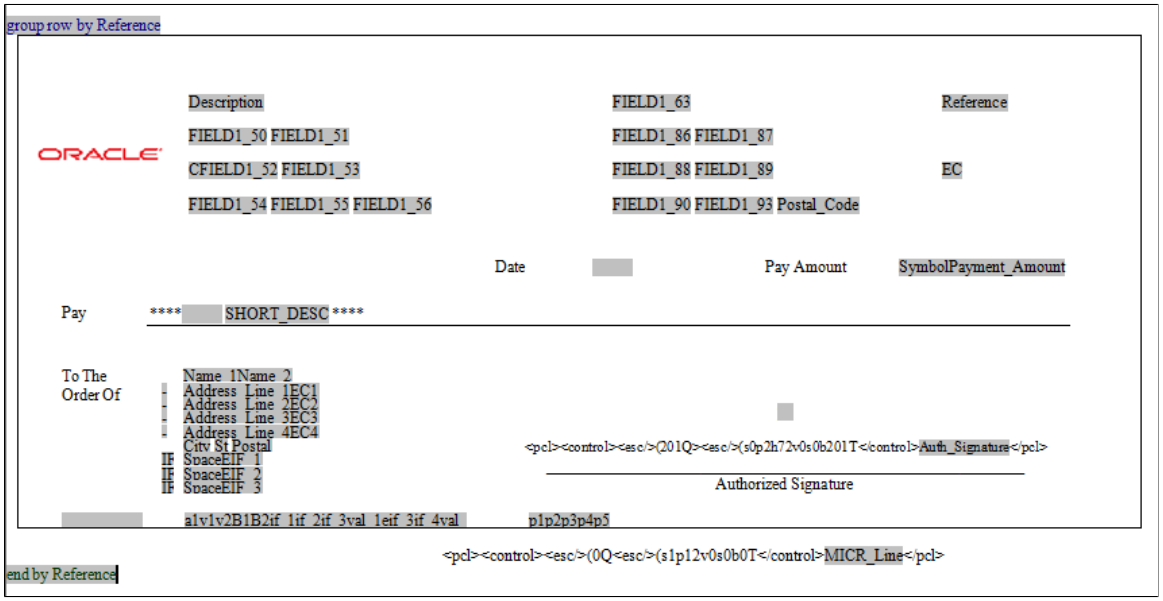
Copyright © 1988, 2020, Oracle and/or its affiliates.

<*sp/*>: inserts a space.

This example illustrates the PCL command sequence in a RTF template.

Image: RTF template

RTF template with embedded PCL code



For PCL printing, a report follows the following process:

- Creating the report

  For PCL outputs, during the report creation process, the RTF template will be populated with the XML data and then a PDF report will be created. In the resulting PDF, the PCL command sequence specified in the RTF template will still be displayed as regular text with the XML data replaced in the fields.

  Note: Only the PDF report with the embedded PCL command sequence is stored in Report Manager. The PCL file is removed from the server unless the psxp_debug property is enabled. However, if a password is required to open or modify the PCL file, the PCL file will be removed even when the psxp_debug property is enabled.

- Printing the PDF report

  When the PCL file is sent to a printer where the MICR fonts and the Troy Signatures are installed, the PCL command sequence is replaced with bank account code and signature.

Using Two Dimensional (2D) Barcode Functions

You can use QR code or PDF417 2D barcode type to create barcodes in RTF templates. When you create an RTF template, use the qrcode or pdf417 functions to specify the barcode type. These functions do not require external fonts.

 Copyright © 1988, 2020, Oracle and/or its affiliates.

QR Code Syntax

In the template, include the following qrcode syntax:

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

where:

DATA: indicates the data to be encoded in the QR code format.

SIZE: indicates the QR code size dimension in points (pt).

CHARSET (Optional): indicates the character set for encoding the data. Default is *UTF8*.

pdf417 Syntax

In the template, include the following pdf417 syntax:

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

where:

DATA: indicates the data to be encoded in the PDF417 format.

XSCALE: indicates the point (pt) per PDF417 module width. Default value is *1* (1pt per module).

COLUMNS: indicates the number of columns to be used in the generated PDF417 symbol. Default value is *-1*.

ROWS: indicates the number of rows to be used in the generated PDF417 symbol. Default value is *-1*.

CHARSET (Optional): indicates the 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.

Handling Sensitive Data in RTF Templates

A BI Publisher report that uses query, connected query, or composite query as a data source may include data fields containing sensitive information, which may be masked either with an asterisk or with another masking character. When you use a formatting function, such as `format-number()` or `format-date()`, on such a field, it may generate an error message or produce a blank value.

This happens because masking characters invalidate the canonical form or representation of the data, which is typically a requirement for most BI Publisher functions.

PeopleSoft provides the `$MaskedFields` parameter to the BI Publisher core engine at runtime. This parameter contains a semi-colon separated list of all masked fields that have been detected in the underlying data source.

With this information, a report developer can use conditional logic in the RTF template to check whether a certain field is masked, before applying any formatting function on its value. Instead of using a formatting function, the report developer can customize the masked value. For example, replacing asterisks with the text *PRIVATE*.

For example, if the data source contains the masked fields BIRTHDATE and ADDRESS, the `$MaskedFields` runtime parameter will contain the following text:

```
BIRTHDATE;ADDRESS;
```

Copyright © 1988, 2020, Oracle and/or its affiliates.

Before using this parameter, you must declare it under a form field at the top of the reports primary RTF template, as shown in the following example:

```
<xsl:param name="MaskedFields" xdofo:ctx="begin"/>
```

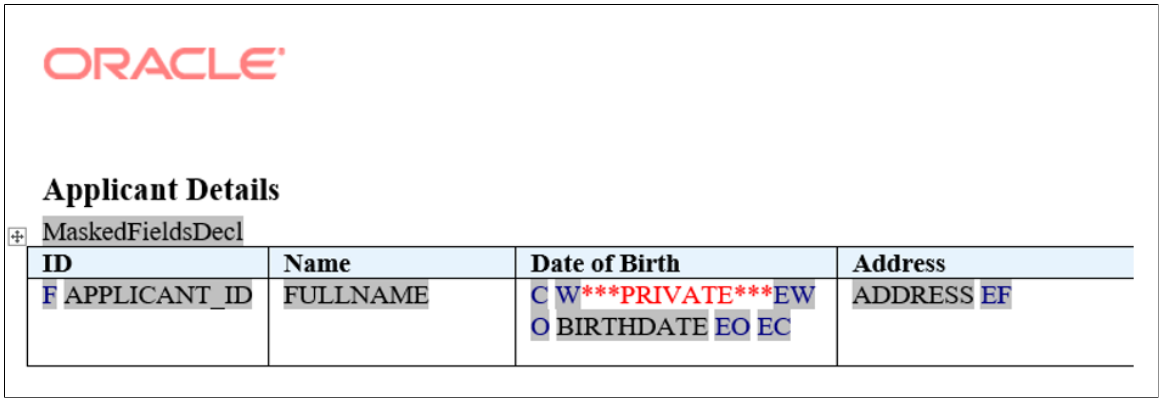
Example to Format a Masked Data Field

This example shows how to use conditional logic in RTF templates to either format a potentially masked data field using a function, or to display a customized text, for example *PRIVATE*, in its place.

Instead of formatting or customizing the masked data field, you may continue to use the un-formatted data value. Check how the ADDRESS field is processed in this example.

Image: Formatting masked data field in RTF templates

This example illustrates the formatting of masked data field in RTF templates.



The following table shows the entries made in the template shown in the above figure:

Default Text	*Form Field Entry*	*Description*
MaskedFieldsDecl	<xsl:param name="MaskedFields" xdofo:ctx="begin"/>	Runtime parameter declaration.
F	<?for-each:row?>	Opening for-each loop for the row element.
APPLICANT_ID	<?APPLICANT_ID?>	Placeholder for APPLICANT_ID element.
FULLNAME	<?FULLNAME?>	Placeholder for FULLNAME element.
C	<?choose:?>	Opens the choose statement.

Copyright © 1988, 2020, Oracle and/or its affiliates.

Default Text	*Form Field Entry*	*Description*
W	<?when: contains($MaskedFields, 'BIRTHDATE;')?>	Tests if BIRTHDATE element is contained in list of masked fields. If it does, the boilerplate text ***PRIVATE*** that follows, is displayed. **Note:** Include trailing semi-colon after the masked field.
EW	<?end when?>	Ends when condition test.
O	<?otherwise:?>	If the previous condition tests are not true, the form field/text that follows this is displayed.
BIRTHDATE	<?format-date:BIRTHDATE;'LONG'?>	Displays the BIRTHDATE element as a formatted date.
EO	<?end otherwise?>	Ends the otherwise statement.
EC	<?end choose?>	Ends the choose statement.
ADDRESS	<?ADDRESS?>	Placeholder for ADDRESS element.
EF	<?end for-each?>	Closing tag for the for-each loop.

This is one example to show how conditional logic may be applied in RTF templates.

For other possible ways to use conditional logic, see *Fusion Middleware Report Designer's Guide for Oracle Business Intelligence Publisher*, “Creating RTF Templates,” Using Conditional Formatting.

Masked Data Fields in a Connected Query

A connected query can generate hierarchical XML.

When a BI Publisher report uses a connected query data source, the `$MaskedFields` runtime parameter will contain a semi-colon separated list of all masked fields in their full XPath form.

This example shows an XML generated from a connected query data source, and the corresponding value of `$MaskedFields` for such a BI Publisher report using this data source.

Copyright © 1988, 2020, Oracle and/or its affiliates.

Image: Sample XML Data

This example illustrates a sample XML Data.

```
<?xml version="1.0" encoding="UTF-8"?>
- <ConnectedQuery xsi:noNamespaceSchemaLocation="" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" name="CQMASKTST">
  - <MASKTESTPARENT>
    - <A.PRCSTYPE>
      - <![CDATA[
            Application Engine
        ]]>
      </A.PRCSTYPE>
    - <A.PRCSNAME>
      - <![CDATA[
            CLEANATT84
        ]]>
      </A.PRCSNAME>
      <A.MESSAGE_SET_NBR>*****</A.MESSAGE_SET_NBR>
      <A.MESSAGE_NBR>34</A.MESSAGE_NBR>
    - <MASKTESTCHILD1>
        <A.MESSAGE_SET_NBR>137</A.MESSAGE_SET_NBR>
      - <A.DESCRSHORT>
        - <![CDATA[
              **********
          ]]>
        </A.DESCRSHORT>
      </MASKTESTCHILD1>
    - <MASKTESTCHILD2>
        <A.MESSAGE_SET_NBR>137</A.MESSAGE_SET_NBR>
        <A.MESSAGE_NBR>*****</A.MESSAGE_NBR>
      - <A.MESSAGE_TEXT>
        - <![CDATA[
              File Attachment Status
          ]]>
        </A.MESSAGE_TEXT>
      </MASKTESTCHILD2>
```

For this XML data, the `$MaskedFields` runtime parameter value will be:

```
MASKTESTPARENT/A.MESSAGE_SET_NBR;MASKTESTPARENT/MASKTESTCHILD1/A.DESCRSHORT;
MASKTESTPARENT/MASKTESTCHILD2/A.MESSAGE_NBR;
```

Related Links

"Using Query Administration" (PeopleTools 8.58: Query)

Using PDF Templates

This section discusses how to:

- Work with PDF templates.
- Create PDF templates.
- Map data tags.

See Associating Templates.

Working with PDF Templates

PDF templates do not require an external plug-in for offline template design. A mapping feature for XML data element tags is enabled when the PDF template file is uploaded to the Reporting Tools, BI Publisher, Report Definition, Template page. The BI Publisher PDF mapping functionality enables you to match existing form fields in a PDF template with sample data field tags.

You only need to do mapping, if the form field names in the PDF template do not match the tag names in the XML data. This is usually the case when you are using a third-party PDF template (such as government form) and when it is not easy to customize the tag names in XML data to match the PDF form fields.

 Copyright © 1988, 2020, Oracle and/or its affiliates.

Observe the following guidelines when working with PDF templates regardless of whether you are mapping PDF template fields or tags:

- The PDF document must allow editing.

  Check the Security setting in the File, Document Properties, Summary page.

- Files must be Adobe Acrobat 5.0-compatible.

  If you are using a later version of Adobe Acrobat, select File (or Document — depending on the version of Adobe) > Reduce File Size and select the *Acrobat 5.0 and later* value in theMake Compatible with: option.

- Files must have form fields for the placement of application data, with each field tag being unique.

  If no form fields exist, use the Adobe Professional version to add field tags. If duplicate tags or non-unique tags are in forms obtained from third parties, use Adobe Professional to update the tags.

- Files should not have embedded JavaScript.

  BI Publisher removes it during the course of the `Map Enablement` function when theGenerate button on theReporting Tools > BI Publisher > Report Definition > Template page is selected.

Processing PDF Templates

The Oracle BI Publisher Core Engine adheres to the following rules when processing PDF templates:

- The search for the tag that matches the field name starts at the end of the XML file.
- The first match of the tag is used for the field value.
- If the tag is not found, the Oracle BI Publisher Core Engine looks at the map file (if provided).

  This means that even if the form is mapped, when a tag is in the XML data that matches the PDF template form field tag, it has priority over the map for placing the data.

Using Full Path Mapping

The PeopleSoft implementation for PDF mapping supports full path mapping. Full path mapping is not supported in the BIP Server. Full path mapping should only be used when you have no control on the structure and names of your XML data tags.

Note: Full path mapping is available only for PeopleTools 8.50 and later. It is not backward compatible.

Creating PDF Templates

To create a PDF template without mapping tags using Adobe Acrobat:

1. Design your template in the PDF document as documented in *Fusion Middleware Report Designer's Guide for Oracle Business Intelligence Publisher*.

   Be sure that the PDF template field names match the XML data tags.

Copyright © 1988, 2020, Oracle and/or its affiliates.

2. Upload the completed template into the Report Definition by clicking the Upload button on theReporting Tools > BI Publisher > Report Definition > Template page.

Note: BI Publisher supports Adobe Acrobat 5.0 (PDF specification version 1.4). If you are using a later version of Adobe Acrobat, use the File > Reduce File Size option to save your file as Adobe Acrobat 5.0-compatible.

See *Fusion Middleware Report Designer's Guide for Oracle Business Intelligence Publisher*, "Creating PDF Templates."

Mapping Data Tags

Third parties most often supply PDF templates in which the form fields already exist inside the form template. For the XML data element tags to know where they should print within the PDF template, a mapping is required between the field elements from the data source and the form field elements in the PDF template. Once a PDF form with editable form fields is mapped to the XML sample data fields, the template is ready for use by BI Publisher.

Prior to being able to perform this mapping, some BI Publisher-specific pre-processing of the file is required. This processing requires the existence of an open sample data and the report's data source. Adobe Standard or above and version 6 or above is required for the template mapping. In the event that the PDF form does not have form fields, the form field and tags can be inserted using the Designer or Professional versions of Adobe. The form field tags can then be mapped to the sample data tags.

To create a PDF template by mapping data element tags using Adobe Acrobat:

1. Upload the PDF template file to be mapped by clicking the template file Upload button on theReporting Tools > BI Publisher > Report Definition > Template page.
2. If you are using full path mapping, select the Full Path Mapping check box.
3. Generate the file to be mapped by clicking the map file Generate button.

   Generate creates a map-enabled PDF, with the following naming convention. The plug-in enables you to access the data tags by embedding a JavaScript plug-in inside the PDF template.

   - A dash and the letter *m* added at the end of the file. For example, if the original file is *template.pdf,* the mapped file will be *template-m.pdf* if path mapping is not selected.
   - A dash and the letter *mfp* added at the end of the file. For example, if the original file is *template.pdf,* the mapped file will be *template-mfp.pdf* if path mapping is not selected.

4. Visually map the data tags to the form's field tags.

   The mapping exercise is performed offline within the Adobe Acrobat application.

5. Save the file.

   The generated file name indicates the type of mapping, as previously defined in step 2.

6. Upload the mapped PDF file on the Reporting Tools > BI Publisher > Report Definition > Template page by selecting the map fileUpload button.

 Copyright © 1988, 2020, Oracle and/or its affiliates.

When uploaded to the server, the mapping information is stored in the database along with the PDF form template.

Note: If the PDF template’s field names are the same as the data source's data tag names, then no mapping or uploading of a map file is required.

Note: PDF file security has to allow editing and saving for the mapping to be completed. The ability to perform these functions depends on the Adobe version that you are working with.

Note: If no map file exists for your PDF file, selecting the Preview button on theReporting Tools > BI Publisher > Report Definition > Template page will not show any data because the form fields names do not match XML data tag names.

BI Publisher and Adobe

BI Publisher provides the following features within Adobe:

- A visual indication of the PDF form fields that have been mapped.

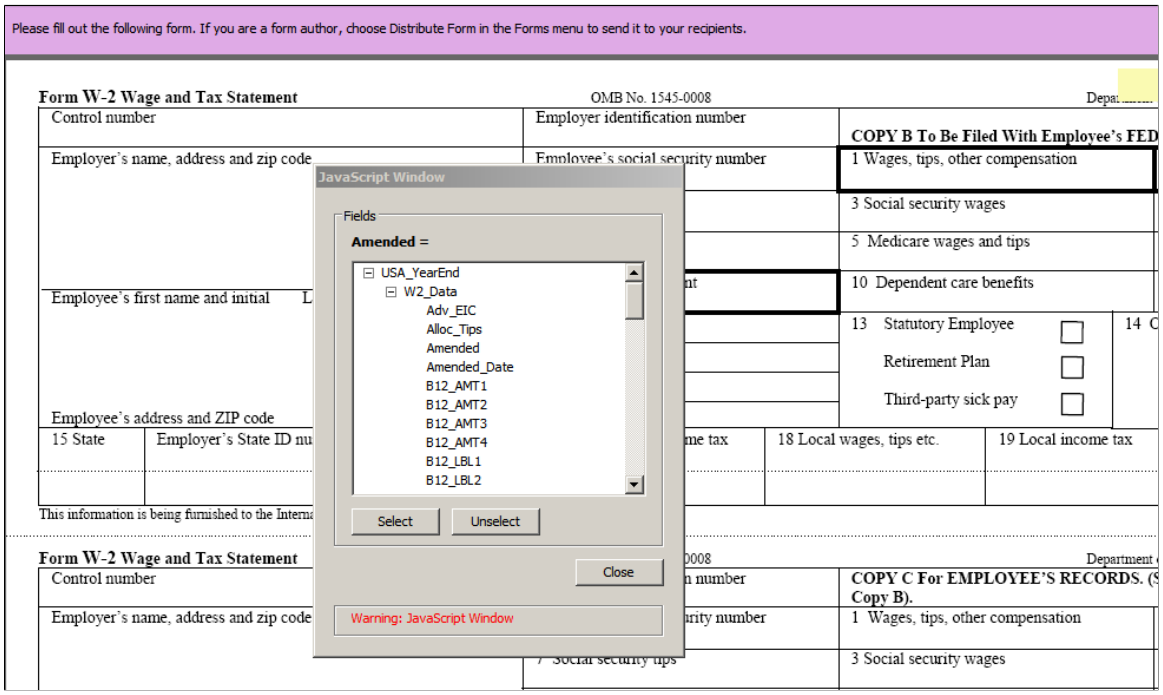
  A dark blue outline appears around the mapped form fields.

- Display of the mapped field tag name when the cursor hovers over the PDF form field.
- A pop-up dialog box containing an XML tag list that you can select from to insert the field tag when you click the PDF form field.
- Preparation of the PDF form for uploading to the report definition when you save the file locally by doing a File > Save.

Copyright © 1988, 2020, Oracle and/or its affiliates.

Image: Sample PDF file mapping

This example illustrates the mapping of form fields on a PDF file.



Creating Submittable PDF Reports

BI Publisher for PeopleSoft enables you to develop PDF reports that can be edited and allows the changes to be saved in a database. BI Publisher uses an embedded button to submit the changes to the database.

Note: Only reports generated from PDF templates can be used as submittable reports.

The submittable PDF report functionality is available for delivered self-service reports and for customer-created reports and is mainly used with government delivered PDF templates.

To implement submittable PDF reports you:

- Create a PDF template.
- Create a custom application class that sets properties, validates data, and saves validated data to the database.
- Set properties in the report definition.

Prerequisites for Developing and Using Submittable PDF Reports

To develop and use submittable reports, PeopleSoft Integration Broker must be configured during development and in the production environment.

PeopleSoft provides an activity guide that walks you through the steps of performing a basic configuration of PeopleSoft Integration Broker.

Copyright © 1988, 2020, Oracle and/or its affiliates.

PeopleSoft also provides the Integration Network which provides centralized access the pages to configure Integration Broker, as well as features and configuration settings for developing, managing, monitoring and administering integrations.

If you are setting up Integration Broker solely to meet the prerequisite for creating submittable reports, use the Integration Broker Configuration activity guide. If you are developing and administering integrations, consider using the Integration Network.

See "Accessing and Navigating the Integration Broker Configuration Activity Guide" (PeopleTools 8.58: Integration Broker Administration) and "Accessing the Integration Network" (PeopleTools 8.58: Integration Broker Administration) for more information.

Viewing Requirements for Submittable PDF Reports

The submittable PDF report functionality uses Adobe Reader plug-ins.

If a user's browser supports Adobe Reader plug-ins, when a user runs a submittable PDF report, he or she views, updates, and submits report changes in the browser in the same PeopleSoft session.

If a user's browser does not support Adobe Reader plug-ins, the report is downloaded to the user's machine for viewing and updating in the Adobe Reader stand-alone product. To submit his or her report updates to the system, the user clicks the Submit button on the report. However, since the user has modified the report outside of an active PeopleSoft session, he or she must re-authenticate his or her PeopleSoft credentials. As a result, when the user clicks the Submit button he or she must enter their PeopleSoft user ID and password for the report data to be submitted to the system.

For working with downloaded submittable PDF reports, Adobe Acrobat Reader version 10 or above should be installed on client machines. Further, if the stand-alone Adobe Reader is used to open submittable reports in an HTTPS environment you need to install a trusted root certificate.

Browser Support for Adobe Reader Plug-Ins	***Behavior***
Yes.	1. The user clicks a button or link to run a submittable PDF report. 2. The report opens in the browser. 3. The user enters and updates information in the report. 4. The user clicks the Submit button. 5. The changes are submitted.

Copyright © 1988, 2020, Oracle and/or its affiliates.

Browser Support for Adobe Reader Plug-Ins	*Behavior*
No.	1. The user clicks a button or link to run a submittable PDF report. 2. The report is downloaded to the user's machine. 3. The user opens the PDF with Adobe Reader. 4. The user enters and updates information in the report. 5. The user clicks the Submit button. 6. A dialog box appears and the user must enter his or her PeopleSoft user ID and password to re-authenticate. 7. If the authentication is successful, the changes are submitted.

Supported Mobile Devices for Submittable PDF Reports

You can submit interactive reports on mobile devices that support Adobe JavaScript. At the time of this writing, only Microsoft Windows devices support Adobe JavaScript.

Opening submittable PDF reports requires Windows-based devices. You can schedule interactive PDF reports on most other devices.

Submittable PDF Report Processing

The submittable report data is submitted to the application server using PeopleSoft Integration Broker REST services. All service components are provided by PeopleSoft and there is no service set up or configuration required, other than to have Integration Broker configured and running. The default service used for processing is *PSXP_PDF_SUBMIT REST* and the service operation used for processing is *PSXP_PDF_SUBMIT_POST*. You can override the default service operation in the PDF Submit Settings property group in the BI Publisher Global Properties or in the Report Definition – Properties page for a report.

Data submitted using submittable PDF report processing is submitted to the application server in XML Forms Data Format (XFDF). The following example show a sample of this format:

All key data required to uniquely identify a row while inserting PDF form data into application database table(s) should exist in the PDF template as fields. It also should exist in the XML data source (Query/ Connected Query or XML file).

Development Considerations for Submittable PDF Report Templates

Consider the following points while developing templates for submittable PDF reports:

- Use Acrobat Professional to create and update templates.
- Templates can use the PDF mapping feature, however the Full Path mapping feature is not supported.
- Templates can use embedded Adobe JavaScript for field validation, field calculations, and so on.
- All functionality listed in the Document Restriction Summary of a PDF template must be allowed.

 Copyright © 1988, 2020, Oracle and/or its affiliates.

To view the document restrictions for a PDF:

1. Open the PDF template.
2. Right-click and select Document Properties.
3. Click the Security tab.

   The Document Restriction Summary appears.
4. All options must be set to *Allowed.*

Note: If a PDF template was created with restrictions, for example it was provided by a government entity, it must be updated using Adobe Acrobat, re-saved and uploaded again to the report definition. In addition, the PDF's associated mapping file must also be regenerated, re-mapped and uploaded again.

- If there are fields that you do not want users to use to submit data, define those fields as read-only in the form editor view of the PDF template. The manner to access the form editor depends on the version of Adobe Acrobat Professional that you are using. The following information is based on using Adobe Acrobat Professional XI.

  To define fields as read-only:

  1. Open Adobe Acrobat Professional.
  2. Select File >Create >Create Form.

     The Create Form dialog box appears.
  3. Choose the appropriate source: From Scratch or Template *or* From Existing Document, and click the Launch button.

     The template appears in the form editor.
  4. Right-click a field to define as read-only and select Properties.
  5. On the General tab, select the Read-Only box.
  6. Select File >Save to save the changes.

Using Submit Buttons for Submittable PDF Reports

At run time, BI Publisher inserts embedded Adobe JavaScript into the PDF template to create a Submit/Save button, as well as some additional hidden edit boxes. The Submit/Save button is not be visible in the template but appears in the report presented to the end user.

Alternatively, you have the ability to insert a custom Submit button into the template to enable precise control on the location and appearance of the button. Use the properties in the PDF Submit Settings group on the Global Properties page or the Report Definition – Properties page to customize the button. The report developer should also record the name of the custom Submit button in the report definition properties. See Defining Global Properties for descriptions of the properties that you can set.

Copyright © 1988, 2020, Oracle and/or its affiliates.

In addition, to ensure that the Submit button is invisible if a report is created as a static report, you must define the following settings for this button in the Adobe Acrobat Text Field Properties box:

- In General Tab, in Form Field drop-down list, select *Hidden*.
- In Appearance Tab, for the Border Color and the Fill Color options, select *No Color*.

Securing Submittable PDF Reports

By default inbound/user-submitted PDFs must be sent from systems using basic authentication and SSL. This parameter is set using the Req Verification drop-down list on the Service Operation — General page of the *PSXP_PDF_SUBMIT_POST* service operation. To access the page select PeopleTools >Integration Broker >Integration Set Up >Service Operations. If SSL is used for user authentication, web server SSL should be used as well. The system administrator can change the authentication method on the service operation to basic authentication only, if preferred. However, keep in mind that SSL provides additional security in the form of login data encryption. If SSL is not used for authentication the report developer should record the fact in the report definition properties

See "Validating Security on Inbound Integrations" (PeopleTools 8.58: Integration Broker Administration) for more information about settings.

In addition, the *PSXP_PDF_SUBMIT_POST* service operation is secured by PeopleTools permission lists.

See "Securing Service Operations with Permission Lists" (PeopleTools 8.58: Integration Broker Administration) for more information.

If the BI Publisher IB Handler is using the HTTPS protocol, the web site used with PeopleSoft software should have SSL feature properly configured.

Performance Considerations for Submittable PDF Reports

Submittable PDF reports are intended to be mostly used by self-service applications running on the application server. The number of application servers required depends on the number of users simultaneously running submittable reports, but should not be less than three (3) as follows:

- One (1) application server for processing.
- One (1) application server for PeopleTools debugging as needed.
- One (1) application server for Integration Broker.

Performance and scalability depends on:

- The number of users running submittable reports at the same time.
- The number of application servers being used.
- The complexity of reports.

Setting Up Submittable PDF Reports

To set up a submittable PDF report:

1. Access the Report Definition page. (Reporting Tools >BI Publisher >Report Definition and click the Properties tab.)

 Copyright © 1988, 2020, Oracle and/or its affiliates.

Select PDF Submit Settings property group and enter values for the properties.

You must set the psxp_submit_enable property to *True* to enable the feature.

The other properties in the property group enable you to:

- Set the size, caption and location of the Submit button or Save button.
- Select the network protocol to use during report submission.
- Override the default service operations to use for processing.

See Defining Global Properties for a description of properties associated with the PDF Submit Settings property group.

2. Select the PDF Security property group and set the following properties and values:

Property	***Value***
pdf-security	*False.*
pdf-no-changing-the-document	*False.*
pdf-changes-allowed	*2.*

3. Select the PDF Template property group and set the following properties and values:

Property	***Value***
all-field-readonly	*False.*
all-field-readonly-asis	*True.* This property is required if some fields are read-only and some fields can be updated, in this case 'all-field-readonly' is ignored)

4. Select the PeopleTools Settings property group and set the folopsxp_pdf_optimized property to *False.*

Developing Application Classes for Submittable PDF Reports

To implement submittable reports you need to create a custom application class that:

- Sets properties mentioned in this section.
- Provides data validation.
- Saves data to the database.

The application class must implement the delivered *IPT_PDFSUBMIT_INT* interface.

Upon completion you must register the application class name in the report definition properties.

Copyright © 1988, 2020, Oracle and/or its affiliates.

To register the application class:

1. Access the Report Definition – Properties page. (Select Reporting Tools, BI Publisher, Report Definition.)
2. Click the Properties page.
3. From the Property Group drop-down list, select *PDF Submit Settings.*

   Property settings for submittable PDF reports appear in the property settings grid.
4. In the psxp_submit_appclass field, enter the name of the custom application class.

Adding Confirmation Pages to Submittable PDF Reports

You have the option to include a confirmation page with submittable PDF reports. A confirmation page is a report that the system merges with the main report into a single output.

The page can let the user know the report was successfully submitted or if an error occurred.

Use the following properties to display an informational page when users submit submittable reports:

Property	***Values***	***Description***
ShowSuccessPage	*True:* Show confirmation page. *False:* Do not show confirmation page.	Use this property to display a confirmation page when users submit a submittable report and the report is successfully saved to the database. In the case of user data input validation errors, a confirmation page with error details always appears.
ConfirmationReportName	*True:* Returns the name of the confirmation page based on PDF_ SUBMIT_CONFIRMATION data source provided by Oracle. *False:* The default confirmation report name, PDFSUBMCONF, is used.	This property provides a name for the report when the success page is shown.

If including a confirmation or information page with a report, use the following properties to describe the condition of the report, for example, if the report was successfully submitted or has validation problems, as wells as display information, error messages, and other content.

Property	***Values***	***Description***
MsgSuccess	Property string get	To be used if data was successfully submitted. Informational translated messages to be displayed to the end user

 Copyright © 1988, 2020, Oracle and/or its affiliates.

Property	*Values*	*Description*
MsgAttention	Property string get	To be used if submitted data has validation problems. Informational translated messages to be displayed to the end user
MsgConclusion	Property string get	To be used to provide user instructions on the confirmation page

Making Submittable PDF Reports Static

Use the SetStaticPDFReport property to make a submittable PDF report non-submittable or static after a preset period of time.

This property enables you to deliver a single PDF template when both a submittable and static report are required.

The values for the SetStaticPDFReport are:

- *True:* Static report.
- *False:* Submittable report.

Using Excel Templates

With Excel based templates, you can generate spreadsheet reports by keeping the initial formatting and layout of the template intact.

It allows you to design spreadsheet reports with greater ease and flexibility. Excel templates are mostly used to generate financial reports that present substantial numerical data, along with any relevant calculations or aggregation of that data.

Working with Excel Templates

Excel template allows you to split hierarchical data across multiple worksheets and dynamically naming those sheets, creating sheets of data with master-detail relationships. Most importantly, you can use Excel functions and formulas.

To create an Excel template:

1. Download the delivered BI Publisher Template Builder plug-in for offline template design on the Reporting Tools, BI Publisher, Setup, Design Helper page to facilitate the insertion of application data tags into your Excel templates.
2. Download the XML sample data file by clicking the Sample Data link on the Reporting Tools, BI Publisher, Report Definition page for a specified query.
3. Load the sample data into the document by selecting Data > Load XML Data from the Microsoft Word Template Builder tool bar menu.

Copyright © 1988, 2020, Oracle and/or its affiliates.

4. Design your template in Excel.

Note: Save the template using the format Excel 97-2003 Workbook (.xls) as this is the only Excel format currently supported for Excel templates.

For more details on excel templates,

See *Report Designer's Guide for Oracle Business Intelligence Publisher,* “Creating Excel Templates.”

Limitation

Template translate file (XLIFF) cannot be created with an Excel template type. If multiple versions of the template are required in different languages, separate templates will need to be created for each of them.

Copyright © 1988, 2020, Oracle and/or its affiliates.

Chapter 5

Defining Report Definitions

Creating Report Definitions

This section provides an overview of report definitions and discusses how to:

- Define reports.
- Associate templates.
- Use data transform.
- Set output options.
- Set report properties.
- Set security options.
- Set bursting options.

Understanding Report Definitions

Report definitions associate a data source with template files. A data source registers the schema and sample data design files. The extracted application fields from the data source files are placed into the template files to create the final report.

A report can include multiple templates. A template is used to associate different layout formats as required by different countries and regions or as required by different channels (web posting, printer, fax, and so on).

The defined output options from the report definition are reflected on the output type and format prompts on the Process Scheduler request page when the application process that runs the report is assigned the process type of BI Publisher. Security settings for a report definition determine who can view the report when it has been run.

Report properties can be set to control formatting of the report.

With the advanced bursting feature, report generation results in separate output files when bursted reports are run through Process Scheduler.

Report definition access is based on user permission list security and roles. For example, bursting is read-only for BI Publisher power users, because only developers can set up bursting, and the page only appears when settings exist.

BI Publisher power users can start to define a report to download the sample data files to create their templates.

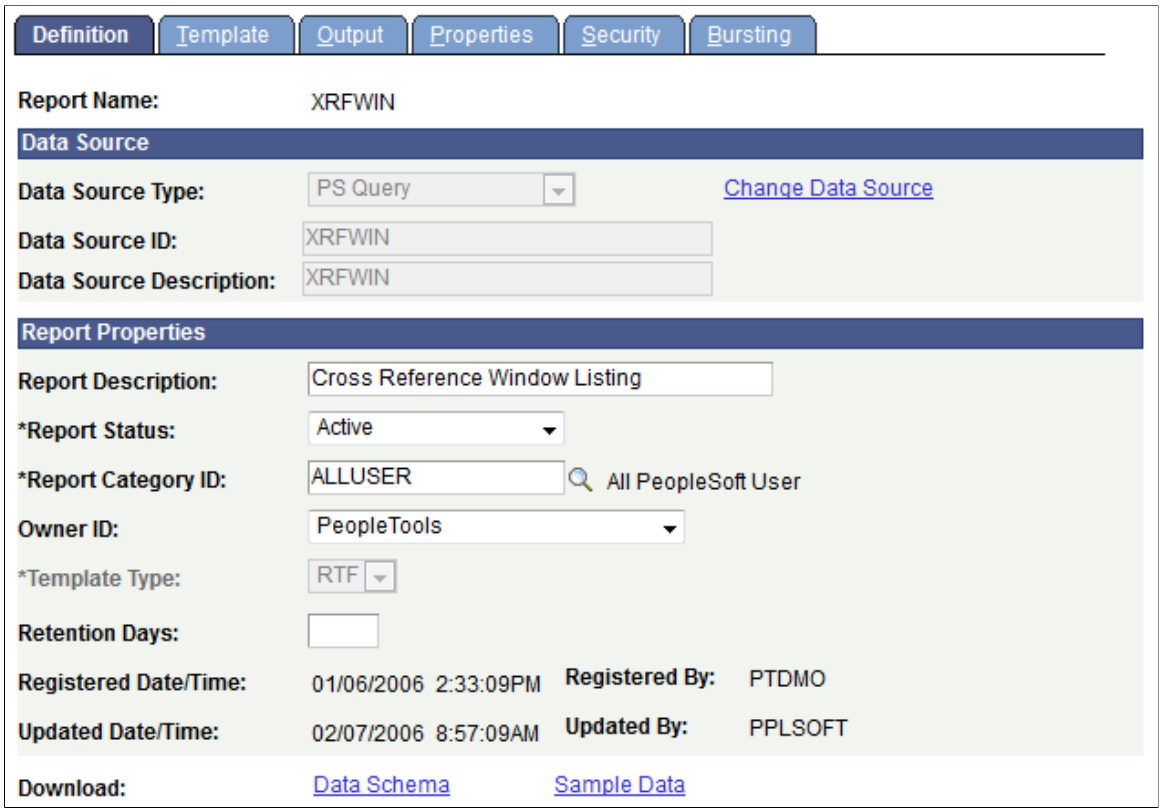
Copyright © 1988, 2020, Oracle and/or its affiliates.

Defining Reports

Access the Definition page (Reporting Tools > BI Publisher > Report Definition > Definition.)

Image: Report Definition – Definition page

This example illustrates the fields and controls on the Report Definition – Definition page. You can find definitions for the fields and controls later on this page.



Report Name

Enter a report name.

The report name must be unique, and it must not contain any special characters. If you enter spaces in the report name, the system replaces them with underscores.

Data Source Type

Select *Connected Query*, *PS Query*, *Rowset*, *XML Doc*, *XML File*, or *Composite Query*.

Note: For BI Publisher power users, the data source types available are *PS Query*, *Connected Query*, and *Composite Query* only.
Rowset and XMLDoc are deprecated in PeopleTools 8.50. If the data source was defined in a previous release, it will be available. You can not create a new data source for rowset or XmlDoc.

See Registering Data Sources.

Change Data Source

This control appears on existing report definitions for users with the XMLP Report Developer role.

Copyright © 1988, 2020, Oracle and/or its affiliates.

When you click the link the Data Source Type and Data Source ID fields become editable and you can define new values.

Warning! Changing the data source may make the report unusable.

If you change the data source, you may need to change other dependant parts of the report definition, such as the report template, report-level properties, bursting values, and so on. After you change the data source, be sure to review the entire report definition to verify that the report values match the newly selected report Data Source.

Before changing the data source for a report, you may want to make a copy of the report and test the impact of changing the data source. See Copying Report Definitions.

Cancel Data Source Changes

This control appears after you click the Change Data Source link.

Click the link to revert the report back to the initial state.

Important! You are able to revert back to the initial state only prior to saving the page.

Data Source ID

Select the data source ID.

You can choose from data source IDs that are based on previously registered data sources. You can select queries regardless of whether they have been previously registered as data sources. For queries, the lookup table respects the public, private, and query access group security for the current user ID.

When you save a report definition with an unregistered query data source, the query is systematically registered as a data source. The query has no object owner ID, but that value can be entered manually on the Data Source page, if required.

Data Source Description

This is a read-only field that reflects the value that was entered when the data source was registered.

For unregistered query data sources, this field reflects the query description.

Report Description

(Optional) Enter descriptive text that provides more detail about the report.

If this field is left blank, the report name appears by default.

Report Status

Select *Active*, *In Progress*, or *Inactive.*

Setting the report status allows work in progress as well as retirement of report definitions. Active reports must have at

Copyright © 1988, 2020, Oracle and/or its affiliates.

least one active template. Only active reports can be selected at runtime and run to success.

Report Category ID

Select a report category ID.

This is a grouping mechanism for reports that provides row-level security for editing report definitions per the rights defined on the report category setup table.

See Setting Up Report Categories.

Object Owner ID

(Optional) Indicate which product, feature, or application owns this report.

Note: The default value that appears here is based on the Object Owner ID setting in the Report Category component (PSXPSETUPRPTCAT).

Template Type

Select *PDF*, *RTF*, *ETX*, *XLS (Excel)* or *XSL*.

Note: ETX is only available if the data source is XML file.

Only one template type is allowed per report.

The template file extension that you can upload on the Template page is controlled by this value. This value also controls which report templates appear on the Translation component (PSXPTMPLTRNS), because only RTF templates are translatable.

Retention Days

(Optional) Enter a value to set the option to purge the reports from the Report Repository and archive the data to the Report Archive table.

The value that you enter overrides the system setting for retaining reports. The maximum value that you can enter is *9999* days. If you don't select a value, the value from thePeopleTools > Process Scheduler > System Settings page applies.

Only BI Publisher report developers or power users with permission list PTPT2600 or PTPT2500 can set this value.

See "Maintaining Reports" (PeopleTools 8.58: Process Scheduler).

Registered Date/Time

This is a read-only field maintained by the system that indicates the date that the initial report definition was registered.

Updated Date/Time

This is a read-only field maintained by the system that indicates the date that the last update to the report definition was made.

Copyright © 1988, 2020, Oracle and/or its affiliates.

Registered By This is a read-only field maintained by the system that indicates the user ID of the operator who initially registered the report definition.

Updated By This is a read-only field maintained by the system that indicates the user ID of the operator who last updated the report definition.

Download Click Data Schema to detach the schema file orSample Data to detach the data file.

Detaching the files enables the user to view the data elements prior to finalizing the report definition.

These links appear if the related files exist on the registered data source. For PS Query data sources, both links always appear regardless of whether the data source is registered because these files are system-generated.

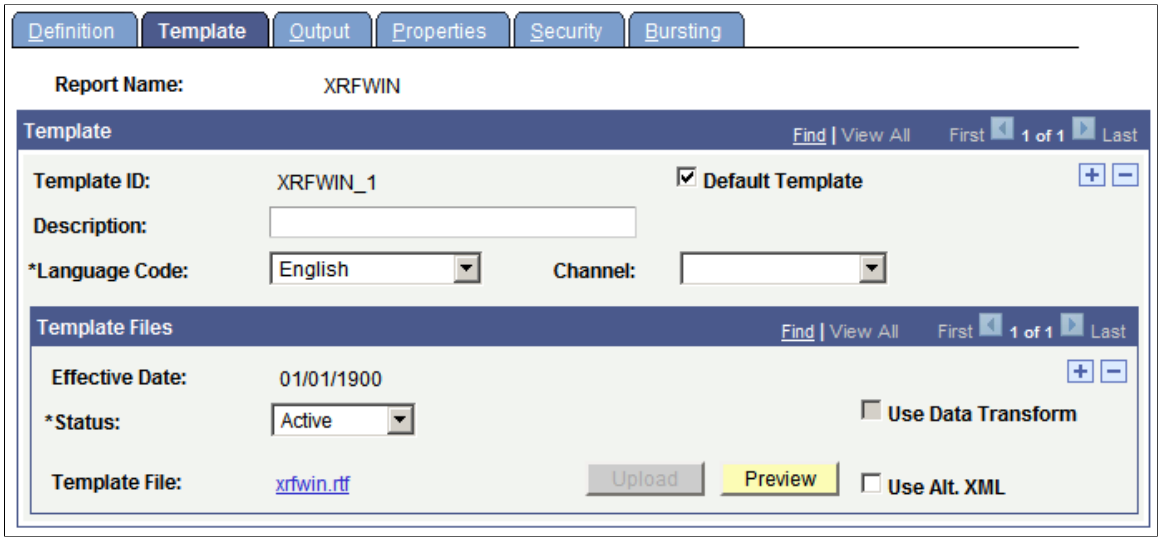
See Registering Data Sources.

Associating Templates

Access the Template page (Reporting Tools > BI Publisher > Report Definition > Template.)

Image: Report Definition-Template page (RTF template)

This example illustrates the fields and controls on the Report Definition-Template page (RTF template). You can find definitions for the fields and controls later on this page.



The Template group box on the Template page refers to a particular template layout, because one report definition can associate multiple template layouts differentiated by language code or channel.

Template ID Enter a template ID that uniquely identifies this template.

The default template ID is a system-generated ID based on the report name. You can edit this ID when you first add a

Copyright © 1988, 2020, Oracle and/or its affiliates.

template to the report definition, but it must be unique across all templates in the system, not just within the current report definition.

Description (Optional) Enter descriptive text that provides more detail about the template and identifies its use.

Entering a meaningful description helps the user select the proper template at runtime. For example, indicate a unique layout or channel.

Language Code Select a language code for the template.

The default value reflects the default template language.

Default Template Indicate whether this is the default template.

You can select only one template as the default template. The first template that you add to the report definition is automatically selected as the default. You can change this selection as necessary.

Default templates are automatically used at runtime if no other value is supplied.

Channel (Optional) Select the distribution channel for the template.

The Channel attribute supports the need to identify different layout formats as required by the various distribution mechanisms. For example, a printout may require a different template layout than an email or a web posting. Leaving the channel blank would indicate that this particular template does not have a format that is specifically suited to just one channel.

These values are for information only and do not trigger a particular Process Scheduler distribution mechanism. Developers can drive a template choice based on channel through the PeopleCode BI Publisher classes.

Adding Template Files

Within each template layout defined previously is one or more effective-dated versions of the template. For example, you can have a new government form for each year. In the Template Files group box, you attach effective-dated files that are the actual report templates.

Effective Date Select an effective date for the template file in order to maintain new versions or versions specific to a particular time period. For example, a new file could be uploaded to reflect a new format, effective for reports as of the new date.

The default date for a newly added template file is the current system date. The user can change the data per effective-dating logic with Update, Update/Display, and Correction modes.

Copyright © 1988, 2020, Oracle and/or its affiliates.

	See "Using Effective Dates" (PeopleTools 8.58: Applications User's Guide).
Status	Select a status of *In Progress Active,* or *Inactive* for the template file. This field indicates the usability of the template file. Runtime selection logic for a template file uses this field in conjunction with the Effective Date field to determine which template file to use. At least one file must be active to save a report definition.
Template File	When you upload the template, the template name appears as a link. Click this link to download the template file to your local computer for updating the field or tag assignments.
Upload	Click to attach a template file to the template. The file extension is checked against the template type value on the Definition page and a warning is issued if no match is found. When you save the report definition, this button becomes disabled. To re-upload a new version of the template, you must either delete and add it again in correction mode or add a new effective-dated row.
Preview	Click to preview the report using the current template file based upon the sample data file that was registered with the data source. The Preview button is not enabled when no sample data file is registered with the data source. The preview tab title depends on the default output type as follows: • PDF output uses the template name with a system-generated number. • HTML output uses the title property from the word template. To change the title property in MS Word, select File, Properties, Summary.
Use Data Transform	Select to specify a data transform program to be applied to this report definition. Once you save the report definition, this field will be Display Only. See Using Data Transform.
Use Alt. XML (Use alternate XML)	Select to use an alternate XML file for previewing. When you click the Preview button, a dialog box appears, where you can select the file.

Copyright © 1988, 2020, Oracle and/or its affiliates.

Note: The preview button uses the sample XML data file to generate report output. Sometimes, if the sample data does not match the real data, you may find discrepancies between preview and real report outputs. This is specifically true when the report template uses sample data in variables and conditional formatting. Creating your own sample file with real data makes the report look more realistic. This sample file can also be used to preview reports using template builder.

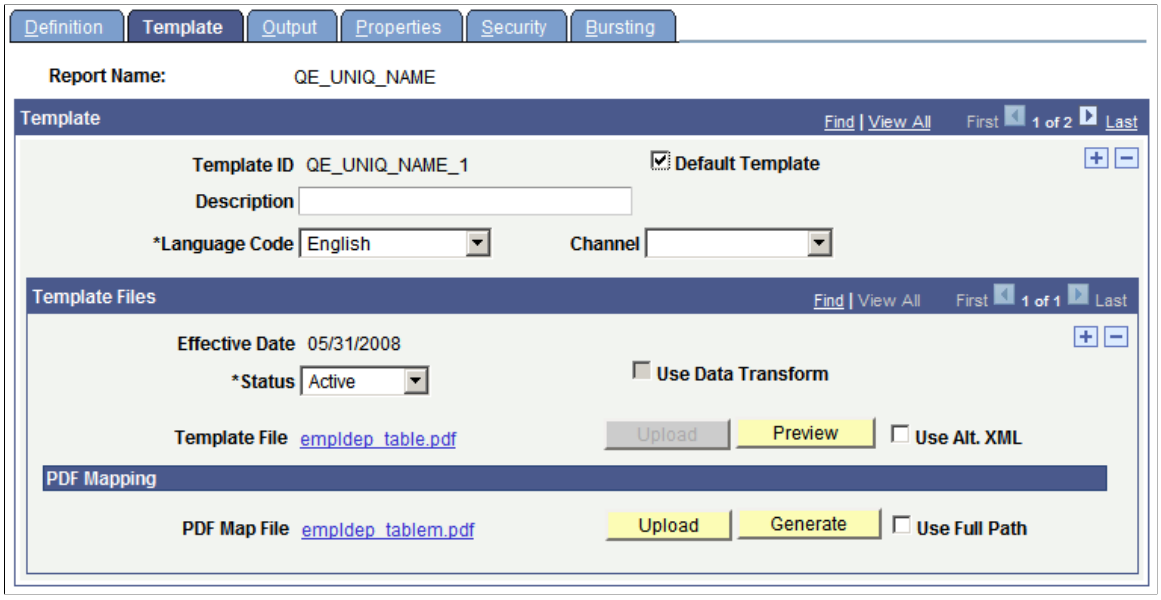
See Mapping Data Tags.

Mapping PDF Template Files

For PDF files, a mapping is sometimes required between the field elements from the data source and the form field elements on the PDF template in order for the XML data element tags to print in the correct place within the PDF template. This is often true for third-party PDF templates, for which the form fields already exist inside the form template. However, if you create PDF form fields and XML tag names that are the same, no mapping is necessary.

Image: Template page for PDF mapping

This example illustrates the fields and controls on the Template page for PDF mapping. You can find definitions for the fields and controls later on this page.



The following fields appear on the Template page for PDF templates files:

Map File

When you upload the mapped PDF file, the file name appears as a link. Click this link to open or download the file to your local computer.

If changes are required in the map file, you can make the changes and upload the revised file without creating a new effective-dated row.

Generate

Click to generate the PDF map file.

The system uses the uploaded PDF template file and the sample XML data associated with the data source definition to generate

Copyright © 1988, 2020, Oracle and/or its affiliates.

a PDF template embedded with a Visual JavaScript plug-in used for mapping.

Any changes made to XML tag names and structure after the template is defined or mapped, require you to redefine or remap the template.

Note: PDF file security must allow altering and saving for the mapping to be completed. This depends on the version of Adobe with which you are working.
When working with PDF map files, some indication of mapping file should be included in the file name to distinguish the mapping file from the unmapped template file. By default, the generated mapping file name is the name of the template file followed by a dash and either an *m* for map file or *mfp* for full path mapping.

Upload	Click to upload the PDF map file when the tags have been mapped.
Full Path Mapping	Select this check box if your XML data has elements with the same name at different levels. For instance, ADDRESS is used at the company level and also at the employee level.

This is an example of XML file that requires full path mapping:

```
<PayChecks>
  <PayCheck>
    <EmpNo>00001</EmpNo>
    <CompanyInfo>
      <Address>1 Company st. CA 00001</Address>
      <Description>Company Info</Description>
    </CompanyInfo>
    <EmployeeInfo>
      <Address>1 Employee st. CA 00001</Address>
      <Description>Employee Info</Description>
      <Salary>50000</Salary>
      <Vacation>12</Vacation>
      ......
    </EmployeeInfo>
  </PayCheck>
  <PayCheck>
    ......
  </PayCheck>
  <PayCheck>
    ......
  </PayCheck>
</PayChecks>
```

The JavaScript plug-in will use the full path for address data elements instead of the element name. So it will use *PayCheck.Employee.Address* to map to the employee' address form field, and use *PayCheck.Company.Address* to map to the company's address field.

See Mapping Data Tags.

Related Links

Using Digital Signature in PDF Reports
Creating Submittable PDF Reports

Copyright © 1988, 2020, Oracle and/or its affiliates.

Using Data Transform

Data transform allows you to create a common template and then use the same template with different data sources. Each report template contains pre-defined placeholders (XML element tags) that are used for the mapping between the layout and the xml data file. These XML element tag names must be used in order to get the proper report output. When you create a new report definition that uses a common template, you will need to create a new xsl program or use an existing xsl program that will map the data fields in the original schema to the data fields for your new report.

When using Query or Connected Query as a data source, the data schema is auto-generated by the system. These system generated tags may not match the XML element tags defined in the common template. In order to generate the standardized or common report properly, the XML data will need to be transformed to match the tags defined in the common template. To do this, an XSLT transform process will need to be applied to the incoming xml data source ahead of the standard BIP template processing.

You can manually create an xsl program or select to use the Oracle XSL Mapper directly from a link on the page. In order to use Oracle XSL Mapper, the system must be properly configured.

Launching Oracle XSL Mapper from Report Definition Template Page

To use Oracle XSL Mapper from the Report Definition Template page you must:

- Install Oracle JDeveloper. If you are using Oracle JDeveloper 11g

  See "Prerequisites for Developing Transforms Using Oracle XSL Mapper" (PeopleTools 8.58: Integration Broker).

- If you are using Oracle JDeveloper 11g, you need to install JDeveloper extensions – Oracle SOA Composite Editor from Oracle Fusion Middleware Update Center.

  In JDeveloper, select Help, Check for Updates.

- Set environment variables on client machine for JDEV_HOME and JDEV_MAPPER_CLASSPATH.

  - JDEV_HOME should be set to the directory where JDeveloper was installed. For example: D:\Oracle\JDeveloper.

  - JDEV_MAPPER_CLASSPATH for JDeveloper 11.1.1.x should be set to:

    <JDEV_HOME>\jdev\extensions\oracle.bpm.mapper.jar;<JDEV_HOME>\integration\lib\bpm-ide-common.jar;<JDEV_HOME>\ide\lib\javatools.jar;<JDEV_HOME>\jdev\lib\xmleditor.jar;<JDEV_HOME>\modules\oracle.xdk_11.1.0\xmlparserv2.jar;<JDEV_HOME>\modules\oracle.xmlef_11.1.1\xmlef.jar;<JDEV_HOME>\modules\oracle.javatools_11.1.1\javatools-nodeps.jar

- Assign the role *XMLP Service User* to the Default User defined for the default local node.

  See "Configuring Nodes" (PeopleTools 8.58: Integration Broker Administration), "Setting Roles" (PeopleTools 8.58: Security Administration).

Note: You must reboot your machine after adding the environment variables.

 Copyright © 1988, 2020, Oracle and/or its affiliates.

Defining the Transform

To define the transform program:

1. Access the Template page in the report definition.
2. Select the Use Data Transform check box.
3. Click Upload and select the common template.
4. Click Upload and select the data schema associated with the common template.
5. Click Save.

   Note: Both the data schema and template must be uploaded for the Mapping Tool link to appear when the page is saved.

6. Click the Mapping Tool link, if you want to create an XSLT program using XSL Mapper. If you already have an xsl program defined, you can skip this step and upload the transform file.

   Note: The first time you access this link, the dialog box will be presented to open with Java(TM) Web Start Launcher (default). Select the check box *Do this automatically for files like this from now on* and click OK.

7. Oracle XSL Mapper will open with the source and target schema.
8. Map the source fields to the target.

   See "Mapping Records and Fields" (PeopleTools 8.58: Integration Broker).

9. Click Save.

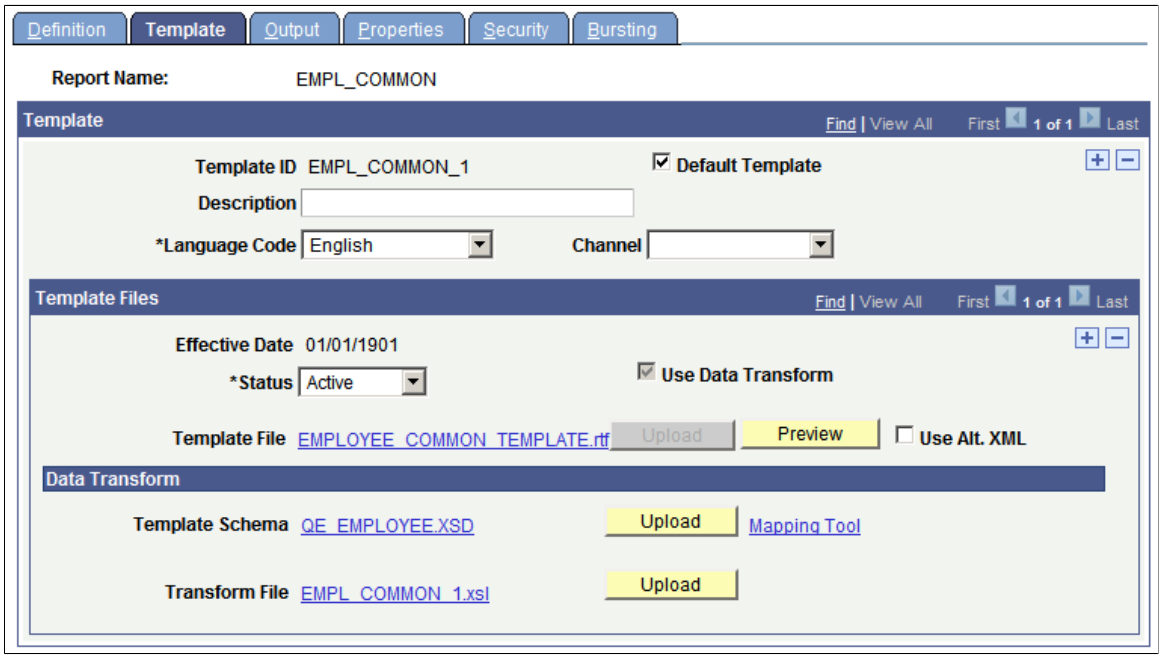
   Note: The xsl file is saved using the template ID name.

10. Close JDeveloper.
11. On the Report Definition page, click the Correct History button.
12. Open the Report Definition.
13. On the template page, click Preview and verify your report format.
14. Set the Report and Template Statuses to Active.
15. Click Save.

Copyright © 1988, 2020, Oracle and/or its affiliates.

Image: Template defined to use data transform

This example illustrates the Template page defined to use data transform.



Transform Example

In this example, a common template is created for Employee. The tags generated from the query that will use this template, do not match the common template:

Common Template Tags	*Query Schema Tags*
EMPLID	EMPLID
NAME	QE_EMPLOYEE_NAME
FIRST_NAME	QE_FIRST_NAME
LAST_NAME	QE_LAST_NAME
SOC_SEC_NBR	QE_SOC_SEC_NBR
JOBCODE	QE_JOBCODE
DEPTID	DEPTID

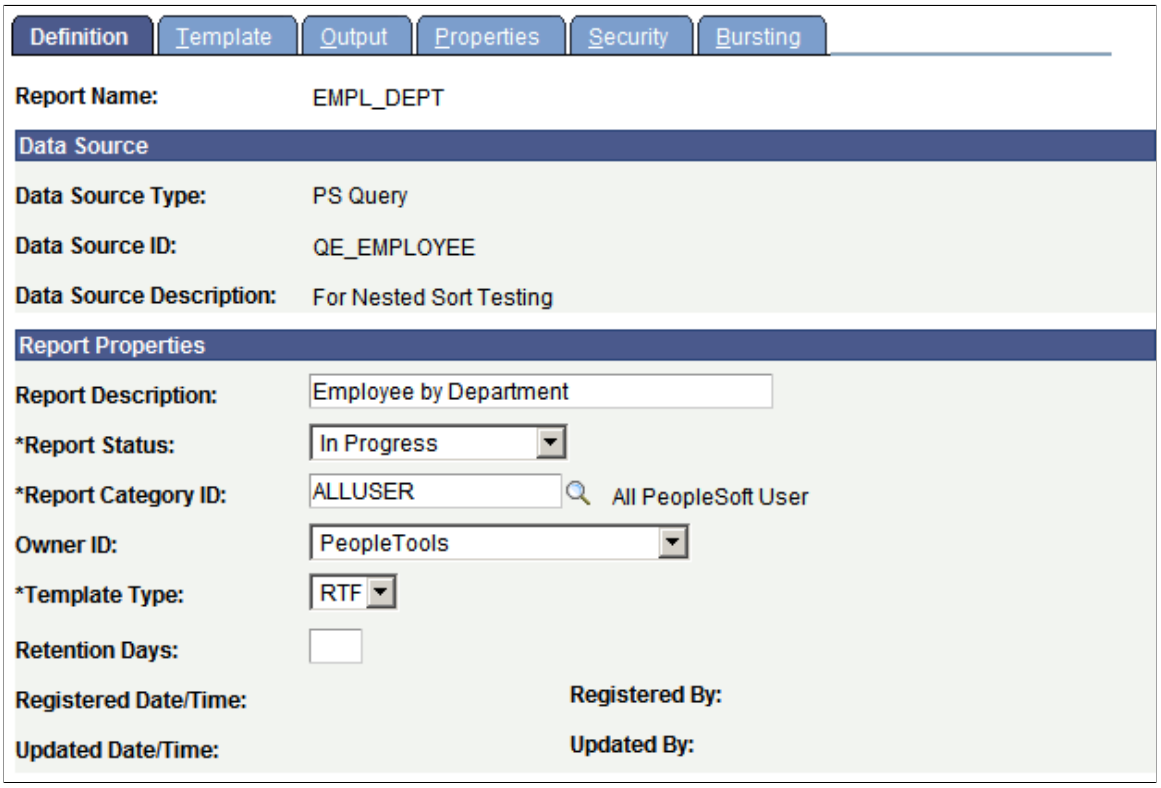
To create the report with the query data source using the common template:

1. Select PeopleTools, BI Publisher, Report Definition.
2. Select Add a New Value and enter the report name, then click Add.
3. Enter a report description and category. The default template type is RTF.

Copyright © 1988, 2020, Oracle and/or its affiliates.

Image: Report definition page specifying RTF template

This example illustrates the Report Definition page, which is defined to use the RTF template.

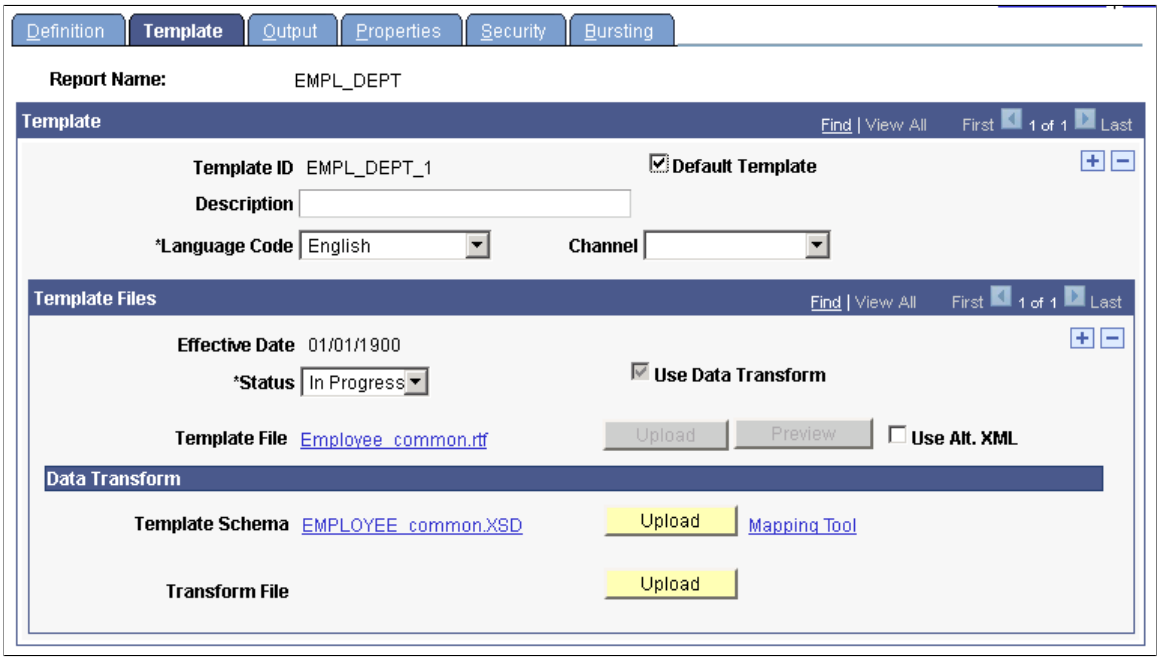


4. Select the Template page, enter the effective date and select Use Transform.
5. Upload the template file.
6. Upload the template schema.
7. Save the page.

Copyright © 1988, 2020, Oracle and/or its affiliates.

Image: Mapping Tool link is displayed

This example illustrates the Mapping Tool link on the Report Template page.



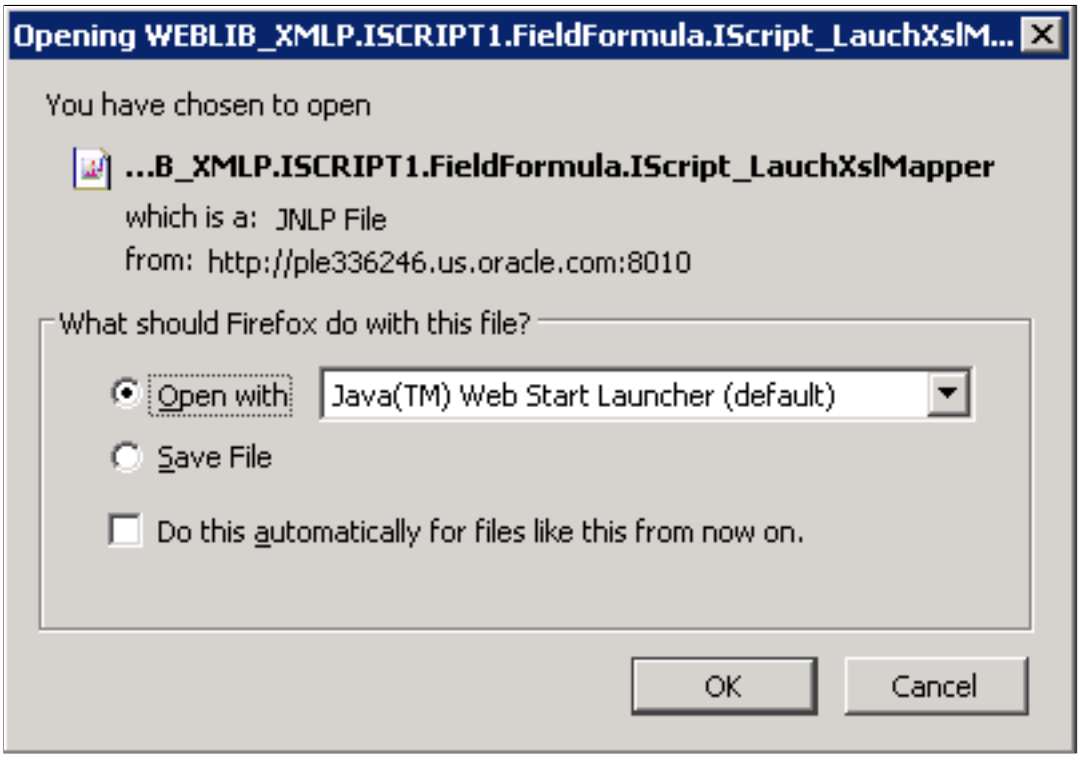
8. Click the Mapping Tool link.

   XSL Mapper will open and you can map the fields. Depending on your environment setup, you may get prompted to open a JNLP file. (JNLP : Java Network Launching Protocol)

Copyright © 1988, 2020, Oracle and/or its affiliates.

Image: Java Network Launching Protocol

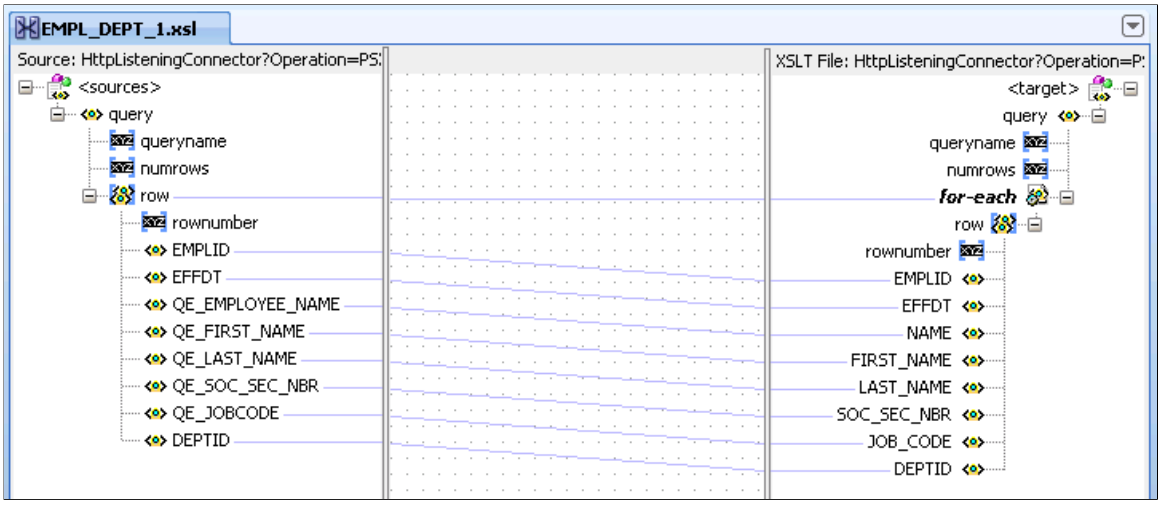
This example illustrates opening a Java Network Launching Protocol (JNLP) file.



9. When XSL mapper opens, you make the necessary mappings between the source and target fields by dragging the field from the source to the target pane in the mapper.

Image: XSL mapping

This example illustrates mapping between the source and the target fields.



10. Select File, Save in XSL Mapper when you have completed your mapping.

Copyright © 1988, 2020, Oracle and/or its affiliates.

Note: This will automatically save the xslt file to the PeopleSoft system. The file will not appear on the page until you click Save again, or reopen the Report Definition.

11. If you want to save the xslt file locally, then select File, Save As and supply the file path.

Note: When you use Save As to a local file, the file is not saved to the PeopleSoft system. You will need to manually upload the xslt file on the Report Definition.

12. After saving the file, you can exit XSL Mapper.
13. On the Report Definition page, click the Correct History button.
14. Open the Report Definition.
15. On the template page, click Preview and verify your report format.
16. Set the Report and Template Statuses to Active.
17. Click Save.

Determining When to Use PDF Mapping Versus Data Transform

If you are using a PDF template, keep the following in mind when determining which method to use for your report template:

- PDF mapping

  Used to change template fields to match data file field names. This method is more efficient than data transform as it requires significantly less processing.

- Data Transform

  Used to change XML data file field names to match the template field names. This method should be used when a common template is required to run in different organizational units.

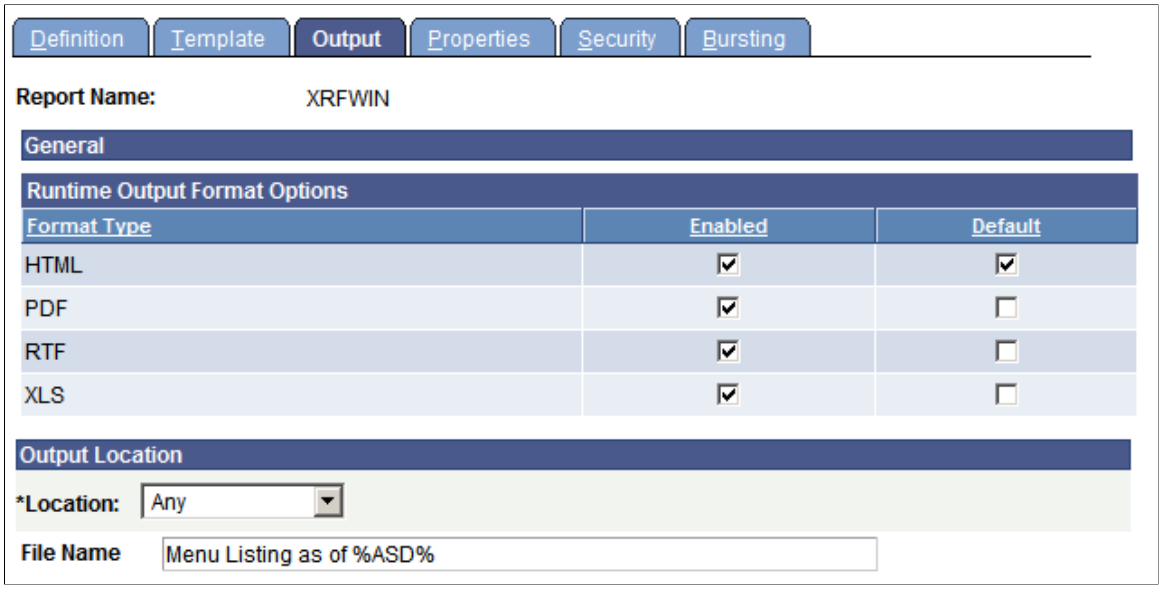
Setting Output Options

Access the Output page (Select Reporting Tools > BI Publisher > Report Definition > Output.)

Copyright © 1988, 2020, Oracle and/or its affiliates.

Image: Report Definition-Output page (RTF template)

This example illustrates the fields and controls on the Report Definition-Output page (RTF template). You can find definitions for the fields and controls later on this page.



Format Type Dynamically lists the available output formats based on the template type.

Enabled Select specific values to limit the output choices for the user at runtime.

Default Select a default format type.

This value appears at runtime on the prompt or run control page. It specifies the output format that the system uses if no other value is fed into the BI Publisher engine.

Location Select one of the following locations:

- *Any* indicates that the user can select the output location at runtime.
- *Email* indicates that the output goes to email.

  Note: The users defined in the distribution list must have a valid email address defined in the user profile. If Allow viewer assignment at report runtime is selected, you can enter additional email addresses at runtime.
- *File* writes the output to the file that you indicate in theOutput Destination field.
- *Printer* indicates that the output goes directly to a printer.

  Specify the printer destination for the output in the Printer field. This field is available only when the output location that you select is *Printer*.

Copyright © 1988, 2020, Oracle and/or its affiliates.

Printer is a valid selection only when PDF output format is enabled.

- *Web* indicates that the output goes to a web report repository that is accessible by the Report Manager.

  Select the folder for the output from the Report Manager Folder Name lookup. This field is available only when the output location that you select is *Web*.

  This is the default location used at runtime if no location is selected.

- *Window* indicates the output will be posted, like Web output, to the report repository and then streamed to the browser window, the same way scheduled query runs to Window.

  Note: Window output is supported for scheduled and non-bursted reports only. Users building a custom process request page should check for the bursting field name (BurstFieldName) in the ReportDefn class before issuing a process request.

File Name

Specify a file name template that gets translated at runtime to a physical file name. This field accepts a combination of output variables and plain text.

Output variables are enclosed within percent signs (%) and used as part of the descriptive report name visible in the Report Manager or on the BIP Report Search page. The following variables are supported.

- *%ASD%* inserts the as of date.
- *%RID%* inserts the report ID.
- *%BTV%* inserts the burst field value.
- *%LAN%* inserts the report translation language.
- *%field%* where field is the name of a field from the XML data that lies below the first repeating field. For example, if you want the employee ID value to appear in the file name, you would use %EMPLID%.

  Note: %field% variable is only supported for Report Definition where bursting is enabled, and only burst key candidates on the bursting page are eligible.

See Setting Bursting Options

 Copyright © 1988, 2020, Oracle and/or its affiliates.

For example, if you have a report CERTIFICATE that is burst by STUDENT_ID, you can use the file name to provide more details:

- If no file name is specified, the report description will use the report name, such as CERTIFICATE[2916]-CERTIFICATE.HTM.
- If a file name of *LOCATION %TRAINING_LOC% %END _DT%* is specified, the report description will include the variables, such as CERTIFICATE[2916]-LOCATION BOSTON 2009–03–13.HTM.
- If a file name of *%STUDENT_NAME%* is specified, the report description will include the variables, such as CERTIFICATE[2916]-LEE,JAMES.HTM.

Note: If you leave the File Name field blank, the system uses the report ID as the file name. For bursted report, burst value can be used as file name if set programmatically through the ReportDefn class property UseBurstValueAsOutputFileName. The ReportFileName can also be set programmatically as a property of the ReportDefn class. If a ReportFileName is set either in PeopleCode or on the page, it overrides the UseBurstValueAsOutputFileName property.

Note: The BI Publisher report definition output options are reflected in the output type and output format prompts on the Process Scheduler Request page only when the application process that runs the report is assigned the process type of BI Publisher.

Output Format Options

The output options are based on the template type as shown in this table:

Template Type	***Output Options***
RTF	.pdf, .html, .rtf, .xls (html)
PDF	.pdf
E-Text	.txt
XSL	.pdf, .html, .rtf, .xls (html)

Printing BI Publisher Report Output

PeopleSoft applications support batch printing BI Publisher reports directly from a server using several output formats such as HTML, PCL, PDF, PS, RTF and XLS.

Bursted reports are sent to a single printer, but as multiple print jobs

Copyright © 1988, 2020, Oracle and/or its affiliates.

Users can print BI Publisher reports from process scheduler, query report scheduler, or PeopleCode.

Note the following points when printing reports in Microsoft Windows:

- BI Publisher functionality uses the Copy DOS command to print reports from Microsoft Windows:

  ```
  copy  file-name  printer-name
  ```

- For printing in Microsoft Windows the printer must be shared in a windows print server and the printer name has to be specified in UNC format:

  ```
  \\server-name\printer-name
  ```

Note the following points when printing reports from Unix, GNU, and Linux environments:

- BI Publisher functionality uses the “LP” command in Unix, GNU, and Linux environments:

  ```
  lp -c -d  printer-name  file-name
  ```

- For Unix and GNU environments, the printer can be local or in a network, and it doesn’t necessarily need to be in a print server.

See Setting Up BI PublisherRunning Reports Using PeopleCodeRunning Reports in Process SchedulerUsing RTF Templates

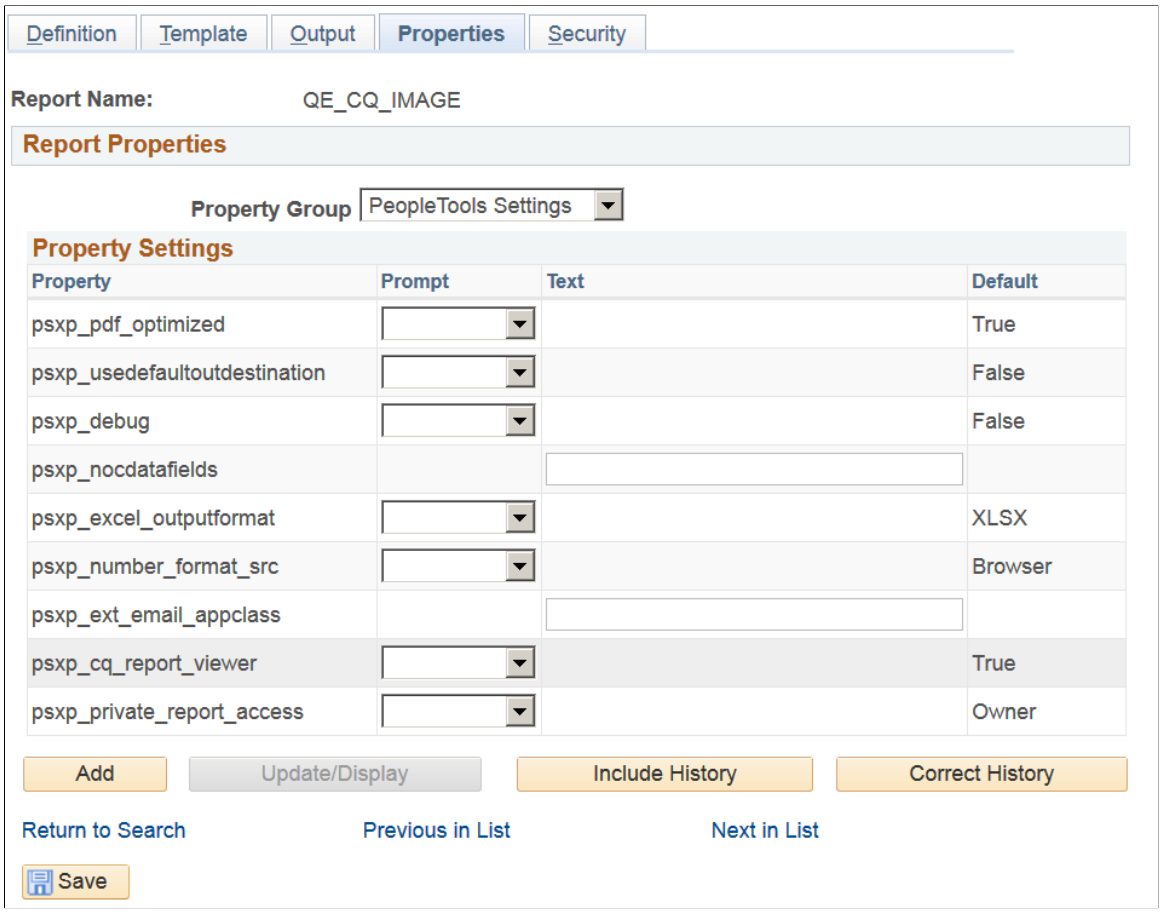
Setting Report Properties

Access the Properties page (Select Reporting Tools > BI Publisher > Report Definition > Properties.)

 Copyright © 1988, 2020, Oracle and/or its affiliates.

Image: Report Properties page

This example illustrates the fields and controls on the Report Properties page. You can find definitions for the fields and controls later on this page.



Properties defined in the report definition will override the global properties for this report.

See Defining Global Properties for information about the fields and controls that appear on this page.

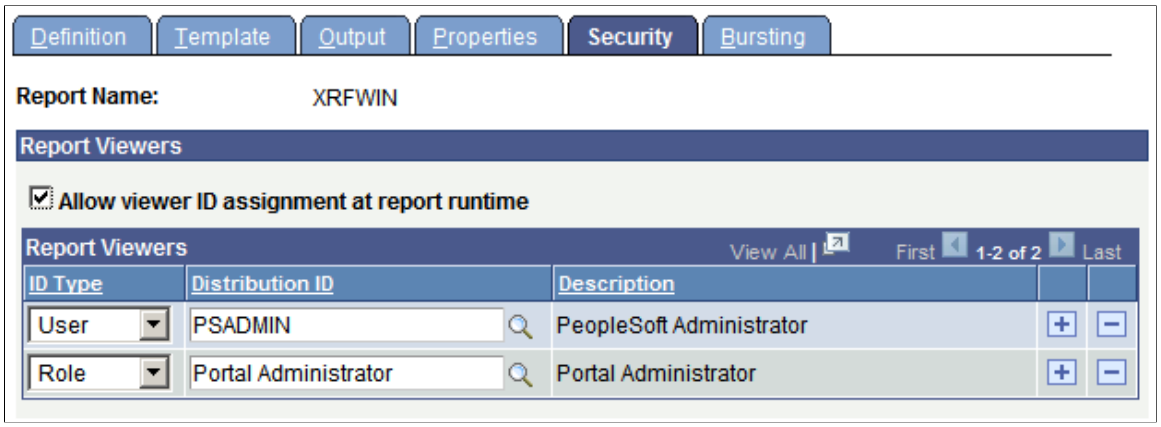
Setting Security Options

Access the Security page (Select Reporting Tools > BI Publisher > Report Definition > Security.)

Copyright © 1988, 2020, Oracle and/or its affiliates.

Image: Report Definition-Security page

This example illustrates the fields and controls on the Report Definition-Security page. You can find definitions for the fields and controls later on this page.



The Security page captures attributes regarding who can view web-posted output in the Report Manager repository and through the BIP Report Search page.

Allow viewer ID assignment at report runtime

Select to indicate that the report requestor can add users or roles to the standard distribution list on the Process Scheduler Request page, under the Distribution Detail link. The users and roles added in the report definition security tab will appear grayed out in the Distribution Detail page to disallow modifying the distribution items coming from the report definition.

When there are no distribution items in the report definition or in the run control tables, the current user is added automatically.

Note: If you are using security join tables to limit report distribution, leave this check box cleared. If you add a user or role at runtime, the associated users will be able to view all bursted reports for that report instance.

ID Type

Select an ID type of either *Role* or *User ID*.

Distribution ID

Select a corresponding distribution ID based on the ID type.

Description

Displays the related description of the distribution ID.

Note: The users and roles defined on this page can view all bursted reports. If you are using security join tables to limit report distribution, do not enter any roles or users on this page.

Setting Bursting Options

Bursting is an optional advanced feature that is available only when reports are run through Process Scheduler. It is not intended for real-time online viewing. It is typically used when you are repeating the generation of a templated report layout many times for multiple like sets of data, for example, generating a batch run on vendor purchase orders or customer invoices. With bursting, you can generate individual

Copyright © 1988, 2020, Oracle and/or its affiliates.

report files resulting in separate secured output, for example, generating a file for each vendor, customer or employee.

Setting up bursting requires thorough knowledge and understanding of data values and schema structures. You could possibly make entries on the Bursting page that would cause the report to fail at runtime. When you generate a bursted report, the system creates separate document files for each unique data value for a specified field tag.

Note: This Burst by field tag must be from the highest level repeating group (node) in the XML data. For bursting to work, only one high-level repeating group should be in the XML source.

Because bursting is an advanced feature, PeopleTools delivers permission list security that is intended for BI Publisher report developers (PTPT2600). When users are assigned a role with this permission list, they have access to setup entries on the Bursting page. A view-only permission list (PTPT2500) option also exists for BI Publisher power users that provides view-only access to the bursting information. The Bursting page appears for the power user only when bursting instructions exist for the report.

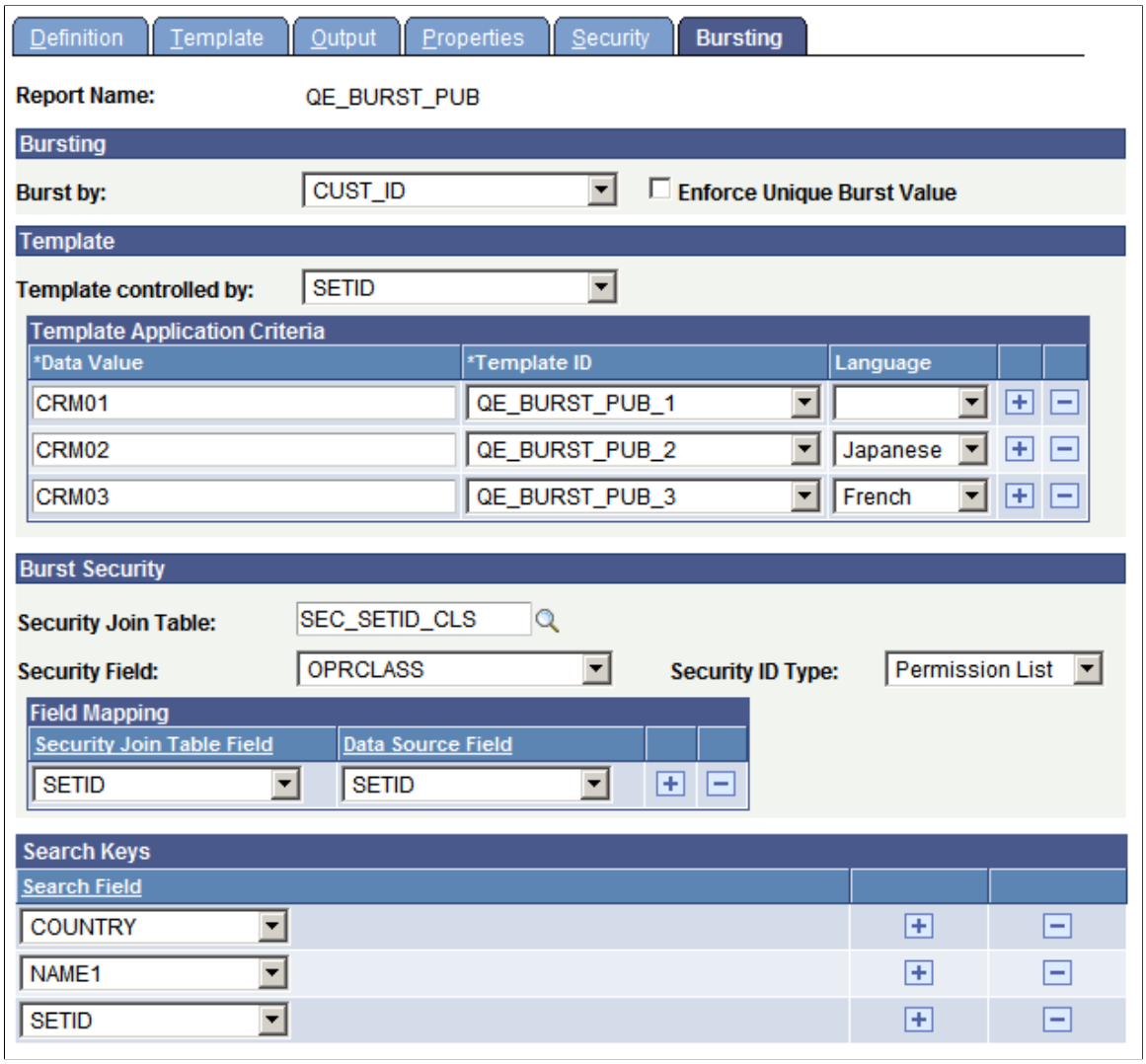
Note: In previous versions of BI Publisher, schemas were necessary for bursting. For backwards compatibility, you can still register and use schemas to define bursting.

Access the Bursting page (Select Reporting Tools > BI Publisher > Report Definition > Bursting.)

Copyright © 1988, 2020, Oracle and/or its affiliates.

Image: Report Definition-Bursting page

This example illustrates the fields and controls on the Report Definition-Bursting page. You can find definitions for the fields and controls later on this page.



Burst by Select a burst by field to enable report bursting.

All subsequent bursting features are disabled until you select this value The values in the drop-down list box are the children from the highest-repeating level (group node) in the XML schema associated with the data source that is assigned to the report definition.

When you select a burst field, the report generates multiple files at runtime with a separate report instance file generated each time a unique value appears for the Burst by data tag. For example, this could be one report file for each employee when you are bursting by EmplID or one report for each department (that includes multiple occurrences of the report, one for each employee) when you are bursting by DeptID.

Copyright © 1988, 2020, Oracle and/or its affiliates.

Enforce Unique Burst Value

Select this check box to indicate that the Burst by field contains unique values. If a non unique value is found, the report will not be published and an error will be logged. It is recommended to use unique bursting values.

If this check box is cleared, bursted files with the same Burst by field will be combined in one report.

Note: Prior to 8.5x, unique burst value was not enforced. Non unique burst value will produce unpredictable results including incorrect search.

Template Assignment for Bursting (Optional)

This feature dynamically drives the template assignment at runtime based upon the data value of a designated schema tag. You can assign a language code to apply a specific template translation as well. This means that the various bursted report occurrences in one batch run can each have an appropriately assigned template and translation. For example, you can print Canadian paychecks in English or French depending upon the employee's preference.

You should select a template ID for each data value that requires a special template.

At runtime, the process looks for the specified template and language. If the language does not exist, then it applies the base untranslated template. If the process encounters a data value that is not assigned on the report definition, then it assigns the template ID that is entered on the run control. If the system captures no template ID selection at runtime, then it applies the default template of the report definition.

Template controlled by

Select the schema tag value from the first child level to indicate the field with the template translation preference.

Data Value

Enter a row for each data value that requires a specific template or template translation.

Template ID

Select the template ID to apply when the data value specified previously is found in the XML data.

These drop-down list box values are dynamically determined by those already defined on the report.

Language

(Optional) Select a language code for the desired translation of the template when the specified data value is found in the XML data.

The language choices in the drop-down list box reflect the complete list of available languages and are not limited by the existing registered Translation XLIFF files.

See Maintaining Template Translations.

Security for Bursting (Optional)

When a report is set up to be bursted, the report designer can also designate how the generated documents are secured when they are posted to the Report Manager. At runtime, the system uses this information

Copyright © 1988, 2020, Oracle and/or its affiliates.

to determine who can view each bursted report instance. You can use bursting security to supplement or replace the basic report viewer security by role or user ID. Otherwise, the system limits access to each report instance based on preexisting system security definitions.

The system automatically limits access to each report instance based on the Burst by field. For example, if the report is burst by employee ID, only the users designated with access to each employee ID can view the output file.

The report designer must provide the record name of the security join table and designate the common fields to join with the bursting field. The system performs the join and determines who can view the report instances. This matching allows the Report Manager's posting process to dynamically identify the user IDs or roles that are assigned viewing rights for each report instance.

Note: If a user has the role ReportDistAdministrator, that user can view all bursted reports, regardless of security join table.

Security Join Table	Select the record name for the table that stores either a user ID or a permission list assigned to a data value in the XML data.
Security Field	Select the field from the Security Join table that stores the user ID or permission list to secure on.
Security ID Type	Select either *User ID* or *Permission List* to indicate what type value is in theSecurity Field.
Security Join Table Field	Select the field from the Security Join table that joins with the data tag to identify the proper row from which to find the value in the in the Security Field that is used to secure the bursted file.
Data Source Field	Select the data tag that stores the values that determines the security assignment. This may require more than one tag, because they must be first-child level tags. For example, they could be employee, customer, department ID, or a set ID/vendor ID combination, and so on.

Search Keys (Optional)

When report results are burst into separate files, you should be able to locate the desired individual report from the Report Manager repository. Delivered search keys include Burst By, Report Definition Name, and Generated On Date. You can define additional search keys to provide even more specific granularity.

At report runtime, the report posting program uses this information to store the key names defined here along with the specific data values for each burst report. From the BI Publisher Report Search page, users can use these configurable search fields to locate a specific report occurrence. For example, if the pay advice report runs regularly and posts numerous report files for self-service access, and as an employee you want to locate a particular dated advise, you would not want to browse through all the advise files to locate the one you want to see. By assigning the pay period as a Search Field in the report definition, the user can enter a date to search for the correct advise.

Search Field	Select an additional field to search on from within the BI Publisher Report Search page.

 Copyright © 1988, 2020, Oracle and/or its affiliates.

The drop-down list box values are taken from the children from the highest repeating level (group node) in the XML schema. Make sure that these values are unique per burst value.

At design time, you can select as many search fields as are required. However, at search time, the BI Publisher Report search page allows only two search criteria in addition to the Burst by value.

An API is provided to facilitate finding bursted BI Publisher reports in the Report Manager repository. When reports are burst into multiple separate files and posted in the Report Manager, the configurable search keys with their values are available as search keys in addition to Report Name, Burst By, Date, and Process Instance ID.

Note: The search feature uses Integration Broker functionality. The service operation PSXP_RATTR is used to insert BIP report metadata for searching. This service operation must be active with a local-to-local routing.

See "Search Operator Values" (PeopleTools 8.58: PeopleCode API Reference).

Assigning Report Viewers at Runtime

There are three settings in the report definition that determine how web reports are distributed at runtime:

1. Report Viewer List on the report definition security page.

   Assign users and roles allowed to view the reports regardless of whether the report is bursted or not.

2. Security Join Table on the report definition bursting page.

   Assign users that can view individual bursted report files based on security join tables. These users are combined with the users and roles defined on the security page.

   Note: When security join tables are used, and the Allow viewer ID assignment at report runtime check box is selected, any users, roles or email addresses added at runtime will see all bursted reports. If roles or users are defined on the security page or at runtime, they can view all bursted reports ignoring the security join table.

   ---

3. Allow viewer ID assignment at report runtime check box on the report definition page.

   Allows the users running the report the ability to modify (add or remove) additional roles, users or email addresses on the runtime report distribution page.

This table describes how viewers are selected for non-bursted reports based on the report definition security settings.

Copyright © 1988, 2020, Oracle and/or its affiliates.

Report viewers assigned	*Allow viewer ID assignment at report runtime*	*Viewers*
Yes	No	Reports are distributed to all roles and users defined on the security page. Runtime overrides are not allowed.
Yes	Yes	Reports distributed to all users and roles defined on the security page. Runtime overrides are allowed.
No	Yes	Distribution list is assigned at runtime on the Process Scheduler distribution detail page. By default the requester is added.
No	No	No reports posted to Report Repository. Runtime overrides are not allowed.

This table describes how viewers are selected for bursted reports based on the combination report definition settings.

Report viewers assigned	*Security join table implemented*	*Allow viewer ID assignment at report runtime*	*Viewers*
Yes	No	No	All bursted reports are distributed to all roles and users defined on the security page. Runtime overrides are not allowed.
Yes	Yes	No	All users and roles defined on the security page will see all bursted reports. Users defined from the security join table will see only the bursted reports based on their join criteria. Runtime overrides are not allowed.
Yes	Yes	Yes	All users and roles identified at runtime will see all bursted reports. Users defined from the security join table will see only the bursted reports based on their join criteria. Runtime overrides are allowed. Any users, roles or email addresses entered on the Process Scheduler distribution detail page will see all bursted reports.

 Copyright © 1988, 2020, Oracle and/or its affiliates.

Report viewers assigned	*Security join table implemented*	*Allow viewer ID assignment at report runtime*	*Viewers*
Yes	No	Yes	All bursted reports are distributed to all roles and users defined on the security page. All users, roles or email addresses identified at runtime will see all bursted reports. Runtime overrides are allowed
No	Yes	No	Users defined from the security join table will see only the bursted reports based on their join criteria. Runtime overrides are not allowed.
No	No	Yes	Assign distribution at runtime. By default requester is added. Any users, roles or email addresses entered on the Process Scheduler distribution detail page will see all bursted reports.
No	Yes	Yes	All users and roles identified at runtime will see all bursted reports. Runtime overrides are allowed. Users defined from the security join table will see only the bursted reports based on their join criteria.
No	No	No	No reports posted to Report Repository. Runtime overrides are not allowed.

Maintaining Sub-Templates

This section provides an overview of sub-templates and discusses how to maintain sub-templates.

Understanding Sub-Templates

You may have text, images, or logic in your templates that you want to reuse across many report templates. Examples include company headquarter address information or standard legal language. Rather than replicate this text, code in every template, or both, you can store sub-template files that include the reusable content. These sub-template files are referenced with standard XSL commands in the primary template file. Sub-template functionality is available for use only with primary RTF and XSL templates.

Sub-templates are secondary RTF or XSL templates that are imported by primary RTF or XSL report templates. The primary template accesses the sub-template through the XSL import style sheet feature.

Copyright © 1988, 2020, Oracle and/or its affiliates.

You can import any XSL style sheets or other RTF or XSL templates using standard XSL import and call functions. PeopleTools simplified sub-template syntax is also supported.

Primary templates calling nonexistent or inactive sub-templates causes an error message to be issued indicating the reason for the problem. This error information is incorporated into Process Scheduler error handling as well as into online viewing or previewing of the report.

The sub-template files are independently stored and are not registered in association with a data source or primary template. This being the case, if any form fields exist inside the sub-template, the report in which the sub-template is placed must have a related data source that supplies those fields, or the data must be passed in as runtime parameters.

The Content Library is a component provided for the registration of reusable sub-template files. The metadata is similar to that of primary template files and includes the sub-template ID, sub-template description, language, object owner ID, report category, effective date, and status. As with Report Definition security, sub-template editor registration security is applied through report categories. Because Report Category secures the data in the component, you can assign select users read-only access for a report category. These users can browse, view, and download sub-template files but not add them. This facilitates the offline design of primary templates for users who can access the library of existing sub-templates but who can’t alter them.

Sub-template names are not exposed to the end user at either report design time or runtime. The complete template (primary and sub-templates) is systematically assembled by the BI Publisher engine during report generation. The same occurs during online previewing as long as the sub-template file exists.

Note: No method is available for viewing which report templates include which sub-templates. This means that users must be careful about changing, deleting, or inactivating sub-templates.

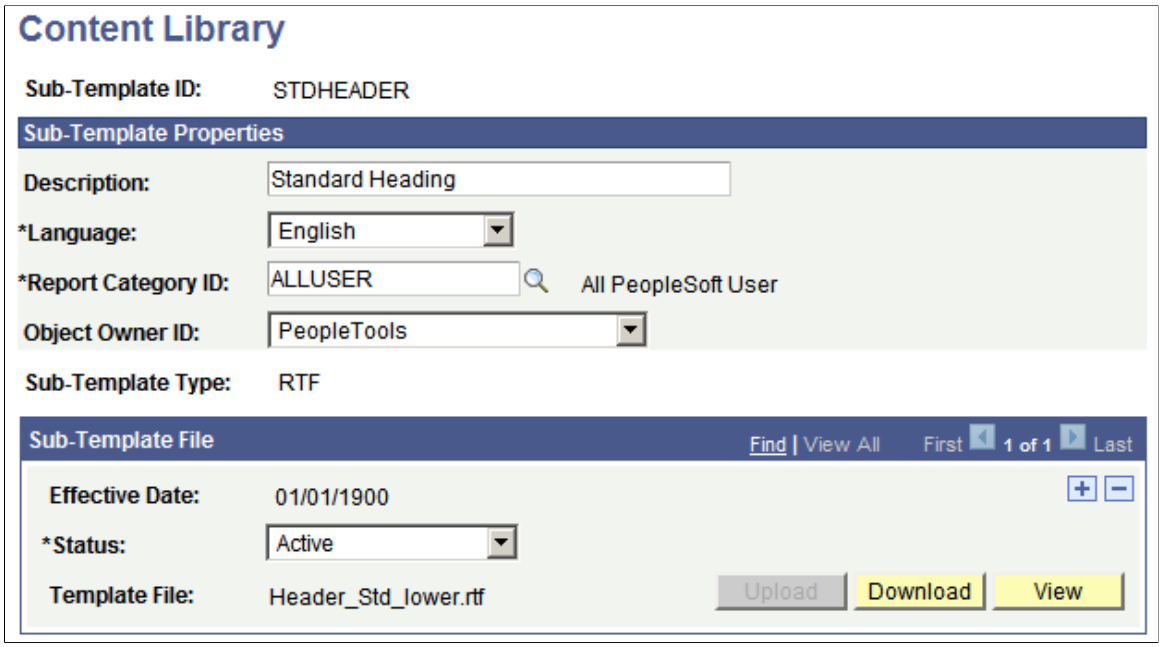
Maintaining Sub-Templates

Access the Content Library page (Select Reporting Tools > BI Publisher > Content Library.)

 Copyright © 1988, 2020, Oracle and/or its affiliates.

Image: Content Library page

This example illustrates the fields and controls on the Content Library page. You can find definitions for the fields and controls later on this page.



Sub-Template ID Enter a unique sub-template ID.

Description (Optional) Enter descriptive text that provides more detail about the sub-template and identifies its use.

Language Select a language code for the sub-template.

The default value reflects the users base language.

Report Category ID Select a report category ID.

This is a grouping mechanism that provides row-level security for editing sub-templates per the rights defined on the report category setup table.

See Setting Up Report Categories.

Object Owner ID (Optional) Indicate which product, feature, or application owns this sub-template.

Use this field to extract and package production data source and report registrations and their supporting files.

Sub-Template Type Select *RTF* or *XSL*.

Effective Date Select an effective date for the sub-template file in order to maintain new versions or versions that are specific to a particular time period. For example, a new file could be uploaded to reflect a new format or new legal language for reports, and the new sub-template is automatically used as of the new effective date.

Copyright © 1988, 2020, Oracle and/or its affiliates.

	The default date for a newly added sub-template file is the current system date. This effective date has no correlation with the effective date of the primary template. The as of date on the Query Report Viewer, Query Report Scheduler, or Run Control page determines which effective-dated templates and sub-templates are run.
Status	Select a status of *In Progress, Active,* or *Inactive* for the sub-template file. This field indicates the usability of the sub-template file. Runtime selection logic for a sub-template file uses this field in conjunction with the Effective Date field to determine which sub-template file to use at runtime. At least one file must be active to save a sub-template in the Content Library.
Template File	Displays the name of the sub-template file.
Upload	Click to attach an actual effective-dated sub-template file. When you save the sub-template, this button becomes disabled. To re-upload a new version of the sub-template, you must delete and add it again.
Download	Click to download the sub-template to your local computer for updating.
View	Click to view the contents of the sub-template.

Maintaining Template Translations

This section provides an overview of template translations and discusses how to:

- Search template translations.
- Maintain template translations.

Understanding Template Translations

The Template Translation component interacts with both report definition templates and Content Library sub-templates. Template translation files can be created only when a report's template type is RTF. Template Translation is a separate component with no row-level security, because the target user is different from the report developer, requestors, or viewers.

The Template Translation feature is based upon standard Localization Interchange File Format (XLIFF) .xlf file processing. Each report template or sub-template file can have related translation XLIFF files. These XLIFF files include translation units for each content element to be translated. The translatable units include all the fixed verbiage of the template excluding any values supplied by the data source. The Template Translations page includes an action button that generates a translatable file that must then be manually edited with the appropriately translated values. When the translation exercise is complete, the XLIFF file is uploaded and integrated into the BI Publisher translation system.

 Copyright © 1988, 2020, Oracle and/or its affiliates.

The Template Translation Search page provides advanced search capabilities to facilitate the location and management of template translations. Using this search page, you can determine whether a particular translation exists. The search can be focused by template or report, thus handling both Report Definition templates and Content Library sub-templates. You can also search based on target language.

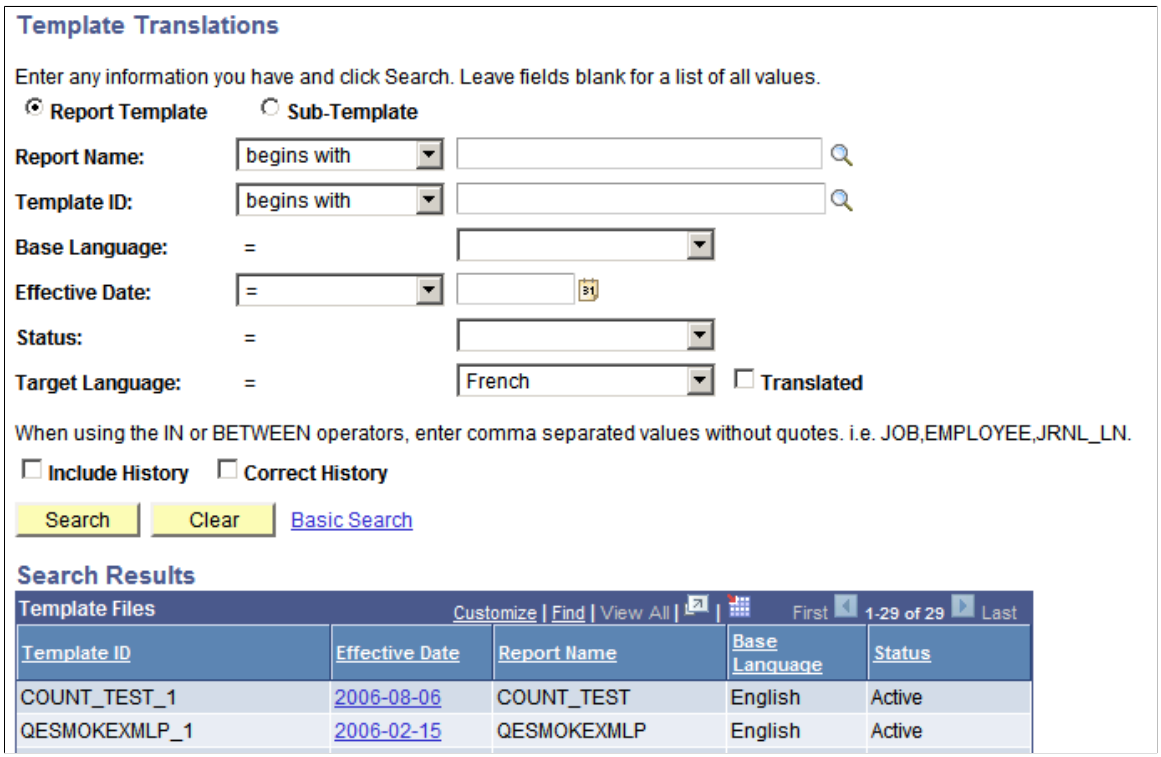
Note: A template must exist before it can be translated.
Template translations are not available for template types other than RTF. For a PDF report, multiple PDF templates must be registered to the report, one for each locale or language as required.

Searching Template Translations

Access the Template Translations Search page (Select Reporting Tools > BI Publisher > Translations.)

Image: Template Translations Search page

This example illustrates the fields and controls on the Template Translations search page.



To search for a template translation:

1. Select either the Report Template orSub-template option, depending on whether you want to search the Report Definition templates or the Content Library sub-templates.

   The subsequent search prompts vary depending upon this choice. For example, the Report Name drop-down list box appears only ifReport Template is selected.

2. Select your search criteria and click the Search button.

   The Translated check box appears only when you have selected a value in theTarget Language field. When selected, this check box enables you to search for templates that have already been translated into the selected target language. If this check box is cleared, you are searching for templates that have not yet been translated into the target language.

Copyright © 1988, 2020, Oracle and/or its affiliates.

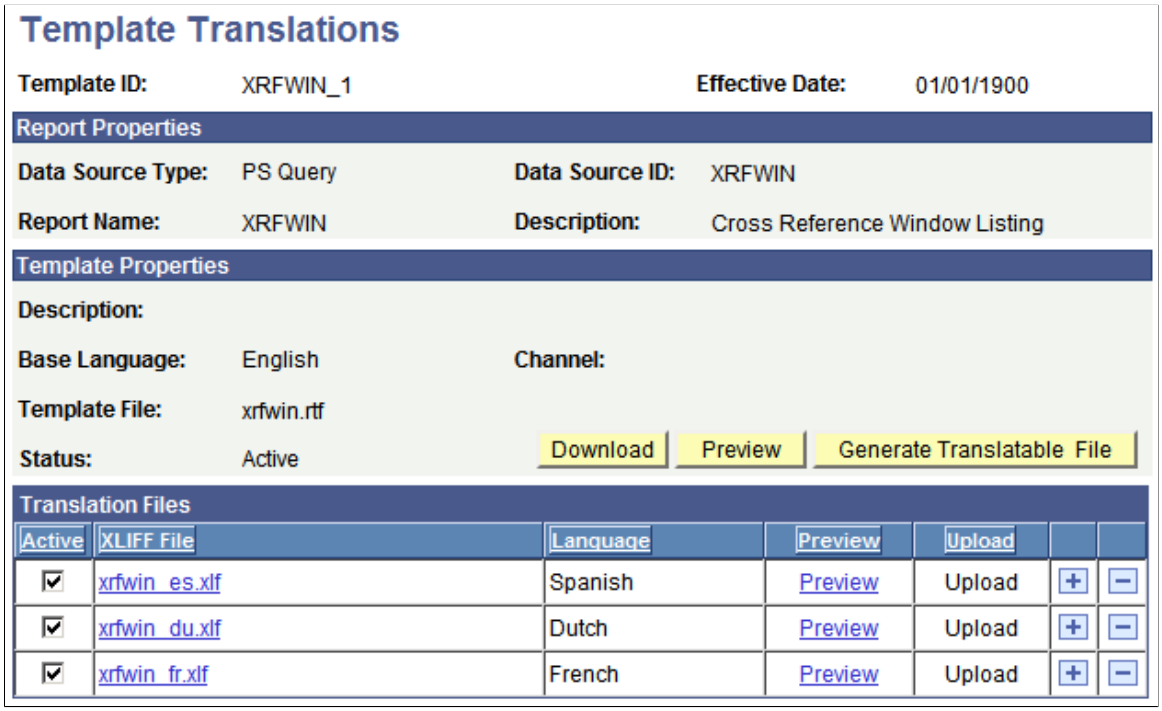
3. When your search results appear, select the effective date of the template for which you want to maintain translations.

Maintaining Template Translations

Access the Template Translations page (Select Reporting Tools > BI Publisher > Translations.)

Image: Template Translations page

This example illustrates the fields and controls on the Template Translations page. You can find definitions for the fields and controls later on this page.



Template ID/Sub-Template ID Displays the unique template ID or sub-template ID.

Effective Date Displays the effective date as registered for the template under the Report Definition component or for the sub-template under the Content Library component.

Note: The translation inherits the same date and cannot be changed.

Report Properties

When the file to be translated is a report template, basic metadata about the report appears. This information does not appear when the file selected is a Content Library sub-template.

Data Source Type Displays the report's corresponding data source type of *PS Query*, *Rowset*, *XML Doc*, *XML File*, *Connected Query*, or *Composite Query*.

Data Source ID Displays the report's data source ID.

Copyright © 1988, 2020, Oracle and/or its affiliates.

Report Name	Displays the report's name.
Description	Displays the report's description.

Template Properties/Sub-Template Properties

The Template Properties/Sub-Template Properties group box displays basic metadata about the base-language template file that has been selected for translation.

Description	Displays the template's description.
Base Language	Displays the base language of the template.
Channel	Displays the distribution channel for the template.
Template File	Displays the name of the template file.
Status	Displays a status of *In Progress, Active,* or *Inactive* for the template file.
Download	Click to open or save the base template file.
Preview	For report templates, click to preview the report template with sample data from the sample data file that was registered with the data source. The Preview button is not enabled when no sample data file is registered with the data source. For sub-templates, click the View button to view the sub-template file.
Generate Translatable File	Click to generate an .xlf file, which includes all translatable units extracted from the fixed text of the selected template or sub-template file. This file must be saved locally and then manually translated.

Translatable Files

The generated translatable XLIFF file includes the template’s static headings and body text that require translation into another language. At the top of the file, the `<source-language>` tag indicates the base language value. You must update the`<target-language>` tag to the language that you are translating into. Initially the`<source-language>` and`<target-language>` values are the same. Prior to uploading the translated file into the database, you must edit the`<target-language>` tag to the translated language code. The value must be the two-character ISO language code.

For example, `fr` equals French,`jp` equals Japanese, and so on. The file won't load if the file type isn’t .xlf or if the`<source-language>` equals the`<target-language>` and an error message appears.

In the `<body>` section of the file, each`<trans-unit id>` tag contains both a`<source>` tag and a`<target>` tag. The`<source>` tag contains the text in the base language. The corresponding`<target>` tag contains the translate fixed text.

Copyright © 1988, 2020, Oracle and/or its affiliates.

No naming restriction is placed on XLIFF files; however, you should keep them close to the template file name and include the language. For example, for a French translation of the XRFWIN template, you could use XRFWIN_FR.xlf.

This code is an example of a translated XLIFF file:

```
  <?xml version="1.0" encoding="utf-8" ?>
- <xliff version="1.0">
 - <file source-language="en-US" target-language="fr-FR" datatype="XDO"
     original="orphen.rtf" product-version="orphen.xlf" product name="">
     <header />
   - <body>
     - <trans-unit id="" maxbytes="4000" maxwidth="15"
         size-unit="char" translate="yes">
         <source>Total</source>
         <target>Totale</target>
         <note>Text located: body/table</note>
       </trans-unit>
     - <trans-unit id="" maxbytes="4000" maxwidth="22"
         size-unit="char" translate="yes">
         <source>Seq Name/</source>
         <target>Nom de Seq/</target>
         <note>Text located: body/table/table header</note>
       </trans-unit>
```

Translation Files

You maintain the translated XLIFF files for your templates in the Translation Files grid.

Active

When it is uploaded, the translated template must be *Active* to make that language translation available at runtime.

The file is *Active* by default.

XLIFF File

Click the name of the uploaded translation file to open or save the file.

This action opens a new window that displays the file per the user's browser and OS settings and allows for updating and reloading the file.

Language

Displays the language into which the file was translated.

During the upload of the translated file, the system determines the language from the `<target-language>` tag and automatically updates the template translation metadata.

Preview

Select to display a translated version of the report in a new window.

This link is active only if the report's data source has a sample data file. No link is available for sub-templates, because no report context is available to preview.

Upload

Select to browse and upload the translation file.

Copyright © 1988, 2020, Oracle and/or its affiliates.

Chapter 6

Copying Report Definitions

Copying Report Definitions

This topic describes copying BI Publisher report definitions.

Understanding Copying Report Definitions

BI Publisher for PeopleSoft simplifies report development by enabling report developers to copy BI Publisher reports.

For a report developer to copy a BI Publisher report definition, he or she must have the *XMLP Report Developer* role with permission list *PTPT2600* assigned his or her user profile.

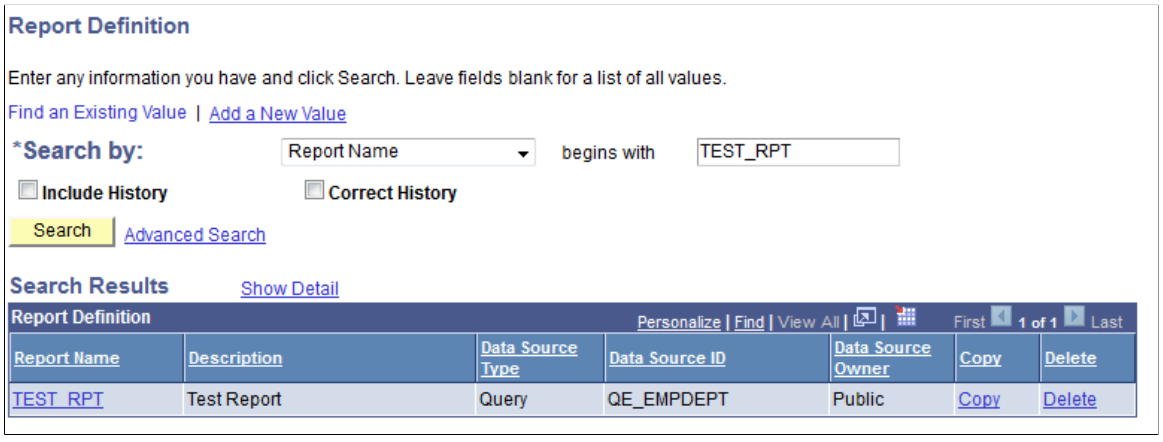
Copying BI Publisher Report Definitions

To copy a BI Publisher report definition, you use the Report Definition search functionality to search for and select the definition to copy. From the search results you access the Copy Report Definition page (PSXPRPTCOPY) to define a name for the copied report.

To access the Report Definition search page, select Reporting Tools >BI Publisher >Report Definition.

Image: Report Definition search page

This example illustrates the fields and controls on the Report Definition Search page.



After you enter criteria to search for a report definition to copy, the results appear in the Report Definition grid.

In the Report Definition grid, locate the row that displays the name of the report to copy and click the Copy link located on the right side of the row.

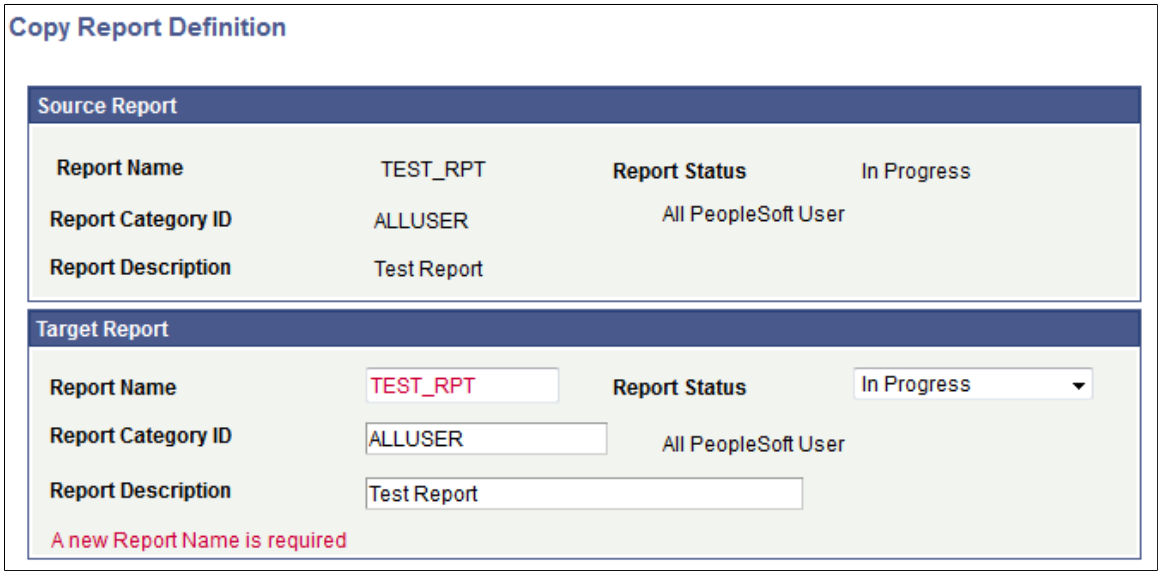
Copyright © 1988, 2020, Oracle and/or its affiliates.

Note: The Copy and Delete links appear in the Report Definition results grid only if the XMLP Report Developer role is assigned to your user profile.

When you click the Copy link in the grid, the Copy Report Definition page appears.

Image: Copy Report Definition page

This example illustrates the fields and controls on the Copy Report Definition page. You can find definitions for the fields and controls later on this page.



The Source Report section of the page displays the key metadata fields defined in the report you have selected to copy. The Target Report section of the page enables you to select different values for these metadata fields for the copied report.

By default the values from the original/source report appear in the Target Report section of the page. The only field you are required to change is the Report Name field located in the Target Report section of the page.

Clicking the Save button saves your report copy to the database. Clicking Cancel button cancel the report copy operation.

The following table describes the fields and controls on the Copy Report Definition page:

Report Name	Name of the report.
Report Status	Status of the report. Select *Active, In Progress,* or *Inactive.*
Report Category ID	Select a report category ID. This is a grouping mechanism for reports that provides row-level security for editing report definitions per the rights defined on the report category setup table.
Report Description	(Optional.) Enter descriptive text that provides more detail about the report.

Copyright © 1988, 2020, Oracle and/or its affiliates.

The report status of the copied report definition is set to *In Progress* by default. To make your copied report definition available to other users change the value in the Report Status field to *Active*. Having the Report Status field value set to the value of *In Progress* allows you to create a back-up copy of the existing report without exposing it to other users.

Copyright © 1988, 2020, Oracle and/or its affiliates.

Copyright © 1988, 2020, Oracle and/or its affiliates.

Chapter 7

Running, Locating, and Viewing BI Publisher Reports

Running BI Publisher PeopleSoft Query Reports

You can view and schedule query-based reports with BI Publisher. You can run custom reports as well as query-based reports batch through the Process Scheduler or online using PeopleCode APIs.

You can view and schedule query-based reports with BI Publisher-delivered PeopleSoft Internet Architecture pages.

This section discusses how to:

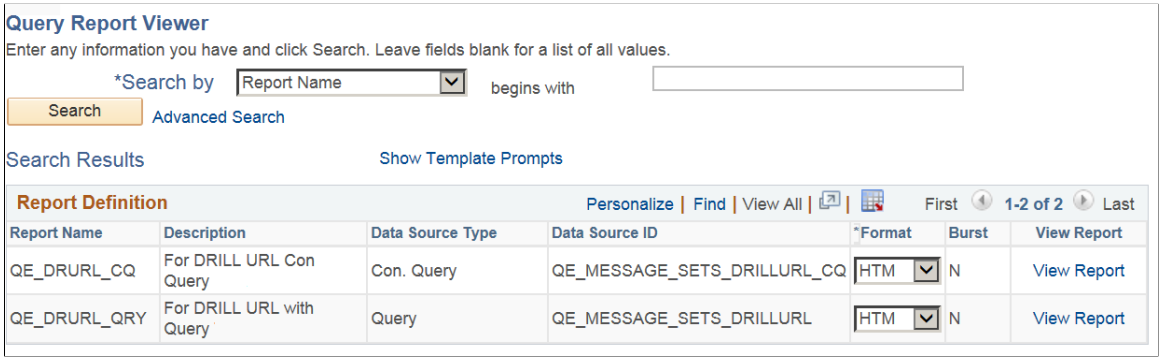
- Run reports in Query Report Viewer.
- Schedule reports in Query Report Scheduler.

Running Reports in Query Report Viewer

Access the Query Report Viewer page (Select Reporting Tools > BI Publisher > Query Report Viewer.)

Image: Query Report Viewer page

This example illustrates the fields and controls on the Query Report Viewer page. You can find definitions for the fields and controls later on this page.



The Query Report Viewer allows selection and online viewing of those reports that have a data source type of PeopleSoft Query, Connected Query, or Composite Query. Existing Query security applies so that each user has access to run only the reports to which he or she has qualified Query access to the data source.

Note: Connected Query based reports cannot be viewed by default. To view the reports, set the psxp_cq_report_viewer property to *True* either on the Global Properties page or on the Properties tab of the Report Definition.

Copyright © 1988, 2020, Oracle and/or its affiliates.

See Setting Up BI Publisher

Show Template Prompts	Click to expand the Report Definition Search Results grid to include the template ID and as of date template prompts.
Report Name	Displays the name of the report.
Description	Displays the report's description.
Data Source Type	Displays the type of data source.
Data Source ID	Displays the report's data source ID.
Template ID	Select from the templates associated with the report definition.
As Of Date	Select the as of date for the template version that you want to view.
Format	Select from the output format choices associated with the report definition.
Burst	Indicates whether the report definition includes bursting instructions. **Note:** Bursted reports are listed, but you can't run them from the Query Report Viewer component. Bursted reports must be run from the Query Report Scheduler component.
View Report	Click to view the report online. When Query runtime parameters exist, the parameters are displayed. A new window opens displaying the report results according to the runtime inputs. You can save the report results locally by using the browser's Save functionality. **Note:** This link is disabled for bursted reports.

Note: The data and template translation language choice is automatically supplied to the user's session language.

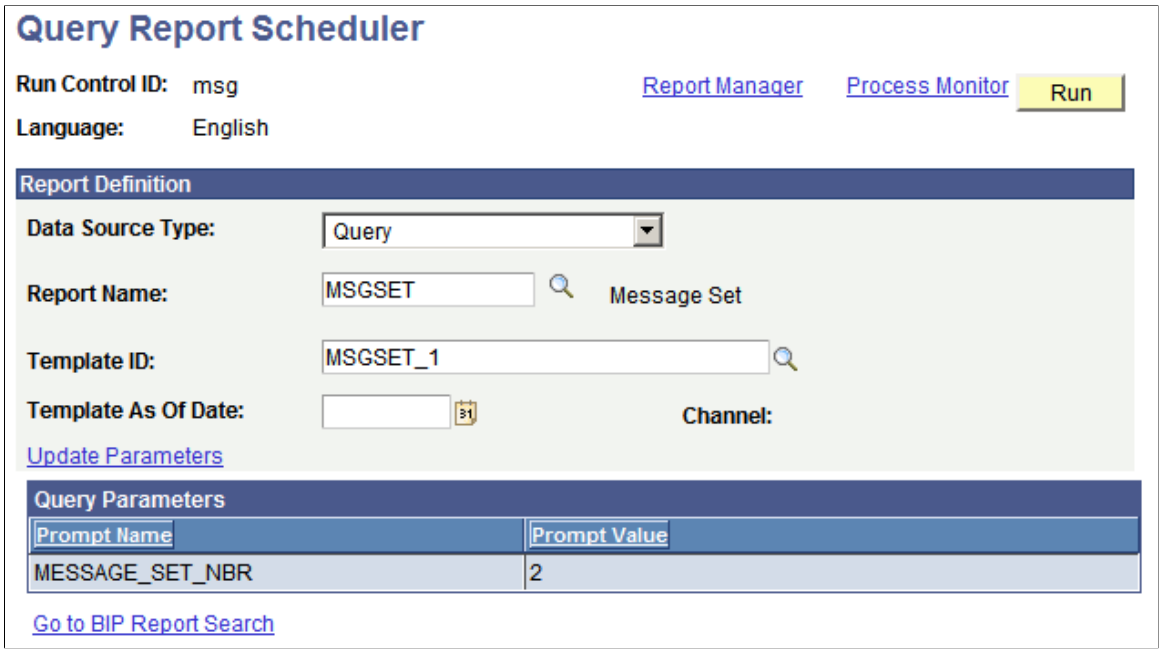
Scheduling Reports in Query Report Scheduler

Access the Query Report Scheduler page (Select Reporting Tools > BI Publisher > Query Report Scheduler.)

Copyright © 1988, 2020, Oracle and/or its affiliates.

Image: Query Report Scheduler page

This example illustrates the fields and controls on the Query Report Scheduler page. You can find definitions for the fields and controls later on this page.



Query Report Scheduler uses the existing Process Scheduler functionality to:

- Select runtime parameters for query-based, connected query-based, or composite query-based reports.
- Monitor the report process request.
- Post and secure the results to either the Report Manager, a printer, or the Process Scheduler file directory.

Note: If a query is run through Reporting Tools > Query > Schedule Query the BI Publisher-related prompts do not appear. Only the basic table-formatted query results are generated.

Run Control ID	Enter a run control ID.
Language	Indicates the language of the run control. The report selects data and template translations based upon the language code of the run control. The user sets this value on the My System Profile > General Profile Information page in theMy Preferred Language for Reports and Email field. The language appears in the Query Report Scheduler Search Results so that you are informed of the language selection criteria.
Data Source Type	Select either *Query*, *Connected Query*, or *Composite Query*.
Report Name	Select the name of the Query, Connected Query, or Composite Query-based report that you want to schedule. The drop-down list box values are based on previously registered report definitions. Existing Query security applies so

Copyright © 1988, 2020, Oracle and/or its affiliates.

	that each user has access to run only reports to which they have Query access.
Burst Field Name	Displays the value set in the Burst by field of the Report Definition, Bursting page. This field appears for bursted reports only.
Dynamic Template	Displays either *Active* or *Inactive,* depending on whether criteria exists to dynamically select the template, language, or both based upon a data value that is set in theTemplate group box of the Report Definition, Bursting page. This field appears for bursted reports only.
Bursting Security	Displays either *Active* or *Inactive,* depending on whether criteria exists to assign unique bursting security that is set in theBurst Security group box of the Report Definition, Bursting page. This field appears for bursted reports only.
Template ID	Select from the templates associated with the report definition.
Template As Of Date	(Optional) Select the as of date for the template version to use for the report. The system looks at the report definition for templates that are active as of this date.
Channel	Indicates the distribution channel of the template.
Update Parameters	Click to update the Query, Connected Query, or Composite Query runtime prompt values. After runtime values are entered, they are saved with the run control ID.
Report Manager	Click to go to the Report Manager to check the progress of your process request and to view the report content immediately after the output file is posted. The final output file is posted to the Report Manager repository for web access by authorized users. See "Viewing Reports" (PeopleTools 8.58: Process Scheduler).
Process Monitor	Click to go to the Process Monitor to check the progress of your request. See "Understanding Process Monitor" (PeopleTools 8.58: Process Scheduler).
Run	Click to access the Process Scheduler Request page. The Process Scheduler Request page enables you to specify variables, such as where a process runs and in what format the process output is generated.

Copyright © 1988, 2020, Oracle and/or its affiliates.

	The values for output type and output format appear by default per the report definition and can be changed if the report definition allows it. Distribution options are also active, allowing updates to viewers, only as allowed in the report definition.
Go to BIP Report Search	Click to access the BIP Report Search page. See Searching the BI Publisher Report Repository.

See "Submitting Process Requests" (PeopleTools 8.58: Process Scheduler) and "Scheduling Process Requests" (PeopleTools 8.58: Process Scheduler).

Running Reports in Process Scheduler

This section discusses how to:

- Use the Process Scheduler Request page.
- Create the run control page.
- Create a process definition.
- Monitor requests.

Using the Process Scheduler Request Page

At runtime, the Process Scheduler Request page appears after you click the Run button on the run control page. This page includes operator-selectable choices of output type and output format. Output type choices reflect the location values from the report definition. Output format choices reflect the output format values from the report definition.

Because values for output location, output format, and viewer security are associated with each report definition, these values should be passed to the Process Scheduler Request page. These values are passed automatically only when the process definition type is *XML Publisher*, and the process name is the same as the BIP Report Definition name that it is meant to execute.

Note: In order to execute BI Publisher reports with Db2 for z/OS databases you will need to configure a Process Scheduler on a PeopleTools certified Windows or other UNIX batch server. The PeopleTools Process Scheduler on z/OS cannot execute BI Publisher reports.

Printing PDF Reports in PCL and PS Formats

BI Publisher reports can be printed to PCL (Printer Command Language) and PS (PostScript) compatible printers. The Process Scheduler Request page allows you to select PCL and PS as output formats when you select printer as output type; however only a PDF version of the report is stored in the Report Manager.

See Embedding PCL Code into Template

Copyright © 1988, 2020, Oracle and/or its affiliates.

Creating the Run Control Page

You need to create a custom run page that contains the prompts required by your report definition.

Your run control page should be a combination of the PeopleSoft PeopleTools-supplied run control sub-page and the application-specific section for runtime parameters for the data extraction program. It should include report, template name, language, and as of date. Depending upon your application design, these values could be systematically deduced from user preferences, come from program defaults, or come from the operator input selection.

See "Running Processes from PeopleSoft Applications" (PeopleTools 8.58: Process Scheduler).

Creating a Process Definition

A process type of *XML Publisher* is delivered as system data and is available on thePeopleTools > Process Scheduler > Process Definition page. An application process that is defined and assigned the *XML Publisher* process type, is required to have the same name as the BI Publisher Report that it is intended to execute. The Application Engine program that extracts the data must also have the same name as the process name. Effectively, the BI Publisher report name, Process Scheduler process name, and the Application Engine program name must all be the same. This ensures that entries for output type and format on the runtime Process Scheduler Request page will reflect the definitional metadata under the BI Publisher report definition.

See "Defining Process Definitions" (PeopleTools 8.58: Process Scheduler).

Monitoring Requests

The Process Scheduler processes BI Publisher-based reports. You can define multiple related activities as separate processes. For example, generation of the XML data, the BI Publisher merging of that data with the template and creating the final output, and the subsequent postprocessing to send related emails. Each process appears separately in the Process Monitor. Error messages indicate whether the problem is on the data extraction or the BI Publisher portion of the report request.

Running Reports Using PeopleCode

This section provides an overview of PeopleCode BI Publisher classes and discusses how to:

- Run reports using PeopleCode.
- Choose a template.
- Pass parameters.
- Use time zones in BI Publisher reports.
- Burst reports.
- Customize printed report output.
- Distribute reports.

Copyright © 1988, 2020, Oracle and/or its affiliates.

- Search for reports.

Understanding PeopleCode BI Publisher Classes

All report runtime functionality is available using the PeopleCode BI Publisher classes. The classes and methods that you use to define custom reports respect the report category security assigned to the report definition. Users with read-only access to report definitions cannot edit them.

Runtime classes are available to call and pass in XML data and a choice of report template to the BI Publisher core engine to generate the output in a desired format. For online viewing, a function is available to pass the output back to the browser. Processing a report through the Process Scheduler posts BI Publisher output entries to the web, the Report Manager, or both according to the existing processes. When processes are categorized under the XML Publisher process type, the capability to establish output destination, format, and authorized viewer choices from the related report definition is enhanced. A search method is also available for accessing reports in the report repository.

You can call the PeopleCode from a page for online processing, or you can create an application engine program to run the report in batch.

See Creating Report Definitions.

See "Creating, Opening, and Renaming Programs" (PeopleTools 8.58: Application Engine)

See "Understanding BI Publisher and the BI Publisher Classes" (PeopleTools 8.58: PeopleCode API Reference).

Running Reports Using PeopleCode

The BI Publisher classes enable you to access the runtime portions of the BI publishing process programmatically, that is, after the templates and reports have been created. The BI Publisher classes are not built-in classes, like rowset, field, record, and so on. They are application classes. Before you can use these classes in your PeopleCode program, you must import them to your program.

Your import statements should look like this:

```
import PSXP_RPTDEFNMANAGER:*;
```

See "ReportDefn Class Constructor" (PeopleTools 8.58: PeopleCode API Reference).

Example: Publish a Report Based on PS Query

This is a code snippet example for publishing a report based on PS Query:

```
import PSXP_RPTDEFNMANAGER:*;

/* get report definition object */
&oRptDefn = create PSXP_RPTDEFNMANAGER:ReportDefn (&sRptDefn);
&oRptDefn.Get();
/* fill query runtime prompt record */
&rcdQryPrompts = &oRptDefn.GetPSQueryPromptRecord();
If Not &rcdQryPrompts = Null Then
  &oRptDefn.SetPSQueryPromptRecord(&rcdQryPrompts);
End-If;
/*generate report*/
&oRptDefn.ProcessReport (&sTmpltID, &sLangCd, &AsOfDate, &sOutFormat);
/*publish report */
&oRptDefn.Publish(&sPrcsServerName,"",&sFolder, &prcsInstId);
```

Copyright © 1988, 2020, Oracle and/or its affiliates.

Example: Print a Report Based on XMLFile

This is a code snippet example for printing a report based on XMLFile:

```
import PSXP_RPTDEFNMANAGER:*;

/* get report definition object */
&oRptDefn = create PSXP_RPTDEFNMANAGER:ReportDefn (&sRptDefn);
&oRptDefn.Get();
/* pass XMLFile to the report definition */
&oRptDefn.SetRuntimeDataXMLFile(&sXMLFilePath);
/*generate report*/
&oRptDefn.ProcessReport (&sTmpltID, &sLangCd, &AsOfDate, &sOutFormat);
/*print report */
&oRptDefn.PrintOutput(&sDestinationPath);
```

Choosing a Template

Because report definition information is available from the PeopleCode BI Publisher classes, you can incorporate prompts on runtime pages to select reports and templates. You must pass in XML data and a choice of report template to the BI Publisher core engine to generate the output in a desired format.

You can retrieve a particular template file or expose a choice of templates at runtime. Only active reports and templates are retrieved. An as of date is also required to coordinate with the template file's effective date. If not supplied, the as of date is assumed to be the system date. At runtime, the template as of date is used to select the appropriate active effective-dated template and sub-template that is current as of that date.

A PeopleCode class is available to retrieve a report's template IDs based on a channel value, although it is not exposed on a PeopleSoft PeopleTools-delivered Pure Internet Architecture page. You can also incorporate template administration functionality directly into your application pages. This functionality includes creating the definitions and storing the related files, as well as querying to find the templates associated with a report definition.

Passing Parameters

The system may need to pass runtime parameters into the BI Publisher core engine. Numbers and text are sent as strings with single quotes. By default, PeopleTools(through the ReportDefn class ProcessReport method) always passes the following parameters/tags:

```
<?$ReportID?>
<?$ReportTitle?>
<?$RunDate?>
<?$RunTime?>
```

These tags can be included in the template layout wherever they are needed, they are especially useful for report headers. Before inserting these parameters into the template (or sub-template), the following declarations must be entered under a form field at the top of the report's primary template; one for each parameter called:

```
<xsl:param name="ReportID" xdofo:ctx="begin"/>
<xsl:param name="ReportTitle" xdofo:ctx="begin"/>
<xsl:param name="RunDate" xdofo:ctx="begin"/>
<xsl:param name="RunTime" xdofo:ctx="begin"/>
```

These tags can be included in the template layout wherever they are needed. These parameters are especially useful for report headers.

 Copyright © 1988, 2020, Oracle and/or its affiliates.

The PeopleSoft-delivered BI Publisher report XRFWIN demonstrates the usage of these values in a report calling a sub-template for a header.

For the standard parameter passage of report ID and report description, the translation of report descriptions may become important for report headers. BI Publisher includes PeopleSoft-related language tables for the data source, and report and template tables that support the report's data language values for the description fields.

Bursting Reports

The ReportDefn class ProcessReport method has code built in to process a single report request to create multiple output files per the bursting instructions defined on the report definition. Bursting always occurs at runtime if a burst value is stored in the report definition's burst field value.

See Setting Bursting Options.

Customizing Printed Report Output

The PeopleSoft application supports batch printing BI Publisher reports directly from a server using PDF output format. Printers with Postscript level 3 interpreter natively support printing PDF format. You can also print to PCL and Postscript printers by specifying the type of printer in the ReportDefn.SetPrinterType() class method. The PeopleSoft application provides PeopleCode support for sending PDF files directly to a specified printer, and it also provides customization capability for inserting conversion logic from PDF to different printer formats.

To send a BI Publisher report to a printer and to specify a printer type, use the PrintOutput and the SetPrinterType methods after the ProcessReport method as shown in this example:

```
&MyReportDefn.ProcessReport("myTemplate", "", %Date, "PDF");
&MyReportDefn.SetPrinterType("PCL" or "PS")
&MyReportDefn.PrintOutput(&PrinterPath);
```

If you want to insert conversion logic from PDF to a different printer format (other than PCL and PS) before an output file is sent to a printer, create a batch file named `psxprint.bat` on Microsoft Windows or `psxprint.sh` on Unix under the Process Scheduler server home directory %PS_HOME %\appserv\prcs\%domain_name% and write a call to an external conversion program in this batch file.

In the batch file, you can use the following variables, which the `ReportDefn.PrintOutput()` method replaces with actual data at report runtime:

Variable	*Description*
`%RPTOUTDIR%`	Full path to the report output directory.
`%REPORTFILE%`	Full path to the report output file.
`%DESTPRINTER%`	Full path to the destination printer.

See Setting Output Options.

Related Links

"Understanding BI Publisher and the BI Publisher Classes" (PeopleTools 8.58: PeopleCode API Reference)

Copyright © 1988, 2020, Oracle and/or its affiliates.

Distributing Reports

PeopleCode options are available for posting your generated report to a file server, printing it, or publishing it to the Report Manager with appropriate security.

For online viewing, a method is available for passing the output back to the browser. No report results are persisted, but the user viewing the results can use the browser's Save As feature to retain the report file locally.

When the output type is *Printer,* the output format is limited to PDF. A printer location must be specified, and the printer must be capable of printing PDF output. If the output file is large, adequate memory must be available on the print server.

Distribute To IDs are those defined in the Report Definition, Security page. Distribution functionality within the Process Scheduler is enhanced to assign values systematically per the BI Publisher report definition. The Report Definition, Security page provides choices for selecting a Report Manager folder as well as the ability to assign viewing rights to additional roles or user IDs at runtime if allowed by the report definition.

When the report definition has the Allow viewer ID assignment at report runtime check box selected, the report requestor can add or delete IDs. If no viewers are assigned, by default the requestor’s ID is added systematically.

Searching for Reports

A search method is available for accessing reports in the Report Manager repository. The PeopleCode uses additional search keys based on the report definition’s additional metadata.

See Setting Bursting Options, Searching the BI Publisher Report Repository.

Using Time Zones in BI Publisher Reports

When displaying date time values, BI Publisher takes into consideration the time zone of the user running the report. The time zone is retrieved from the user's Personalizations settings (My Personalizations > Regional Settings > Local Time Zone). Personalized time zone display is dependent on the following conditions:

- The report template must be either *RTF* or *XSL*.
- The datetime element in the XML file must include the UTC offset, for example, 2008-07-28T09:00:00-0700.

  Note: A Query data source includes the offset for datetime fields.

- The time zone must be mapped to a default IANA time zone under the Time Zone Mapping page. To access the page select PeopleTools >Utilities >International >Time Zone Mapping.

  BI Publisher uses the associated IANA time zone when processing the report. All delivered PeopleSoft time zones are mapped to a default IANA time zone. If custom time zones have been implemented in your environment and have not been mapped to a default IANA time zone, BI

 Copyright © 1988, 2020, Oracle and/or its affiliates.

Publisher will not recognize these custom time zones and will display the time in UTC. The following warning will be logged:

```
Warning: No corresponding IANA time zone defined for user time zone XYZ.
```

See *PeopleTools: Global Technology,* “Setting and Maintaining Time Zones” for more information.

- The datetime field in the template can be formatted using an Oracle abstract format mask that displays the time zone. For example:

  ```
  <?format-date:STARTDATETIME;'SHORT_TIME_TZ'?>
  ```

  Oracle abstract format masks are listed in *Fusion Middleware Report Designer's Guide for Oracle Business Intelligence Publisher* “Creating an RTF Template,” Using Oracle Abstract Format Masks.

  Alternatively, you may apply the PeopleSoft regional settings formatting mask such as $PsftDateTime or $PsftDateTimeTZ. See Applying PeopleSoft Regional Settings Personalizations to BI Publisher Reports

Locating and Viewing BI Publisher Reports

This section discusses how to search the BI Publisher report repository.

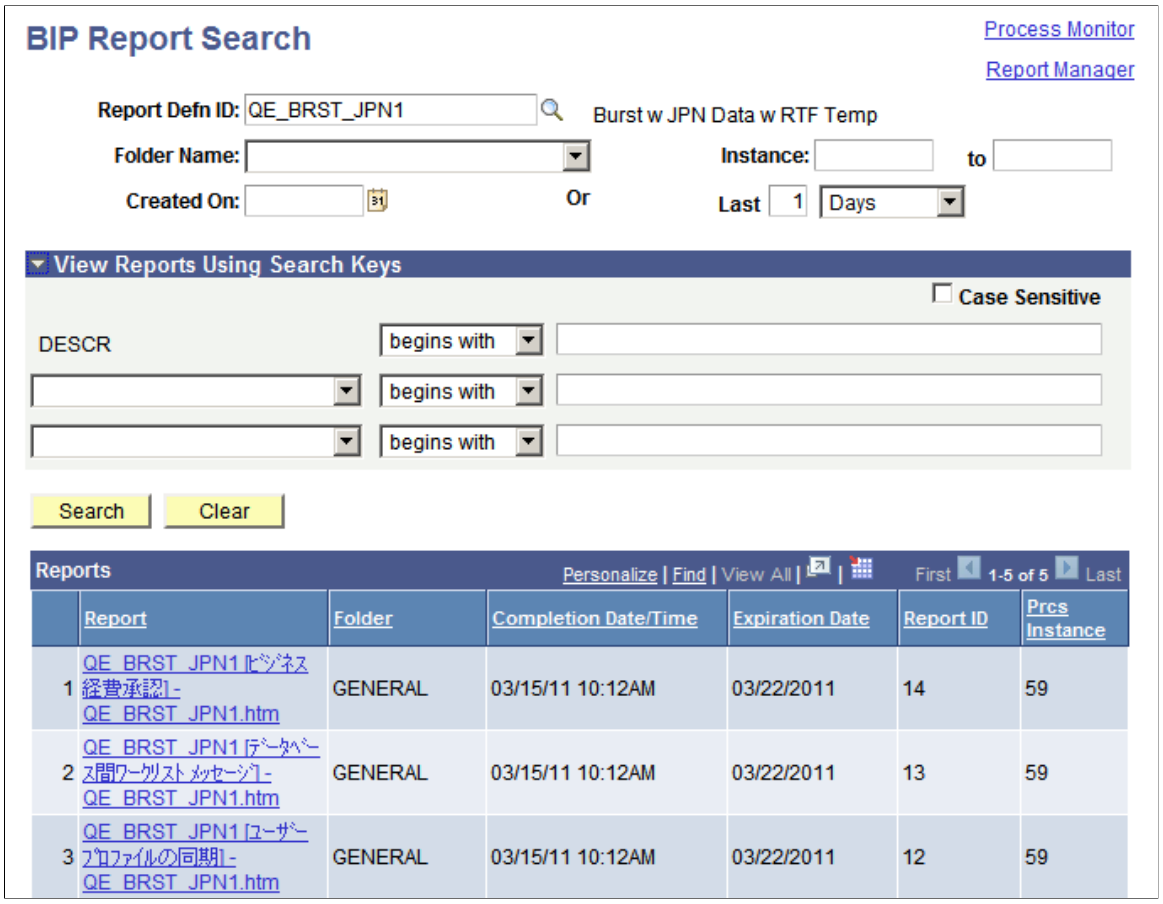
Searching the BI Publisher Report Repository

Access the BI Report Search page (Select Reporting Tools > BI Publisher > BIP Report Search.)

Copyright © 1988, 2020, Oracle and/or its affiliates.

Image: BIP Report Search page

This example illustrates the fields and controls on the BIP Report Search page. You can find definitions for the fields and controls later on this page.



Enter criteria to filter the reports to list. BI Publisher Report Search ignores criteria for fields that are blank.

Report Definition ID (Optional) Select the name of the report definition to search on. If you want to use additional search keys, you must select the Report Definition ID.

Folder Name (Optional) Select a specific folder to list only the reports that are contained in that folder.

Created On (Optional) Use the calendar, or enter a specific date to list only reports that are created on that date.

Instance and**to** (Optional) Enter a range of process instances. Leave the to field blank to list all instances after the number that you enter in theInstance field.

Last (Optional) Use to display only those reports that were created in the last number of days, hours, or minutes. For example, to list only those reports that were created within the last two hours, enter *2* and select *Hours*.

Copyright © 1988, 2020, Oracle and/or its affiliates.

Report Manager	Click to go to the Report Manager. See "Understanding Report Manager" (PeopleTools 8.58: Process Scheduler).
Process Monitor	Click to go to the Process Monitor. See "Understanding Process Monitor" (PeopleTools 8.58: Process Scheduler).

See "Viewing Reports" (PeopleTools 8.58: Process Scheduler).

Viewing Reports Using Additional Search Keys

Additional search keys are available based on the Report Definition ID.

Users can also search by the following criteria for bursted reports:

- A specific value in the Burst By field.

  This is a read-only field that appears automatically when a value is set in the Burst by field of theReport Definition > Bursting page.

- Up to two additional values in the predefined bursting Search Key fields.

  These drop-down list boxes display the values set in the Search Keys region of theReport Definition > Bursting page.

To view bursted reports using the additional search keys:

1. Select the Case Sensitive check box to perform a case-sensitive search.
2. For the Burst By field, select a search operator.
3. Enter a value to search on.
4. For the additional Search Key fields, select the search field name, search operator and search value.

See Setting Bursting Options.

Copyright © 1988, 2020, Oracle and/or its affiliates.

Copyright © 1988, 2020, Oracle and/or its affiliates.

Chapter 8

Creating Reports that Include Rich Text Editor Data

Understanding Rich Text Editor Data in BI Reporting

The rich text editor control extends the capability of a long edit box. It allows for rich formatting of text content, including common structural treatments such as lists, formatting treatments such as bold and italic text, and so forth.

The data entered in an RTE (Rich Text Enabled) long edit field is stored in the PeopleSoft database as formatted HTML data. BI Publisher reports are now capable of displaying this HTML formatted data in the output report without requiring any special conversion, as had been required in previous PeopleTools releases. The Oracle BI Publisher core engine now natively supports the use of HTML formatted data fields in an RTF report template.

Requirements for Reports Using RTE Fields

In order to create reports that include RTE data fields:

1. User data must be input using the RTE Tool supplied by PeopleSoft.
2. Report needs to use an RTF template.
3. The following form field syntax should be applied for all RTE formatted data fields in the RTF template:

   ```
   <?html2fo:elementname?>
   ```

   where `elementname` is the XML element that contains the RTE field data.
4. The RTE formatted data field should be enclosed in a CDATA section in the XML file.

   In an XML file generated from Query/Connected data source, the character fields are always enclosed in CDATA.
5. Images are not supported in the HTML data.

The following example illustrates how RTE fields can be used in a BI Publisher report.

Query generated XML file containing an RTE field—QEPC_RTE_DATA:

```
<?xml version='1.0'?>
<query numrows="1" queryname="RTE_TEST" xmlns:xsi="http://www.w3.org/2001/XMLSchema⇒
-instance" xsi:noNamespaceSchemaLocation="">
<row rownumber="1">
<QEPC_SPORT><![CDATA[TEST]]></QEPC_SPORT>
<QEPC_RTE_DATA><![CDATA[<h1><span style="font-family:georgia,serif">RTE TEST DATA</⇒
span></h1><em><strong>This is Bold and Italicized Text</strong></em>.<br />This is ⇒
a <a href="http://www.oracle.com/technetwork/documentation/psftent-090284.html">hyp⇒
```

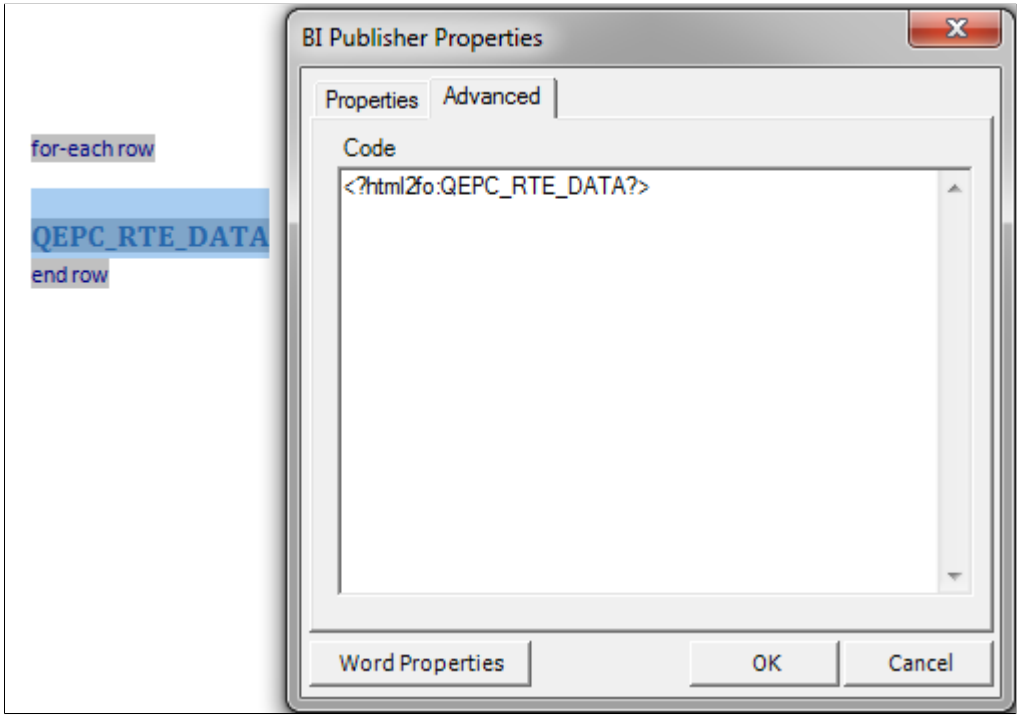
Copyright © 1988, 2020, Oracle and/or its affiliates.

```
erlink </a>to the PeopleSoft Documentation Site.<br />The following is a bulleted l⇒
ist:<ul> <li><span style="color:#0000FF">Item 1 in blue</span></li> <li><u><span st⇒
yle="color:#FF0000">Item 2 in red and underlined</span></u></li></ul>The following ⇒
is a numbered list:<ol> <li>Item 1 with special chars &pound;, &eacute;, &euro;</li⇒
> <li><s>Item 2 striked out</s></li></ol>This is a table with 2 columns and 2 rows:⇒
<table border="1" cellpadding="1" cellspacing="1" style="width:100%"> <caption>&nbs⇒
p;</caption> <tbody> <tr> <td style="text-align: center;">Column 1</td> <td style="⇒
text-align: center;">Column 2</td> </tr> <tr> <td style="text-align: center;">Row 2⇒
</td> <td style="text-align: center;">Row 2</td> </tr> </tbody></table>]]></QEPC_RT⇒
E_DATA>
</row>
</query>
```

Example of required syntax entered in form field for a RTE field in the RTF template:

Image: BI Publisher Properties dialog box

This example illustrates the required syntax in form field for a RTE field.

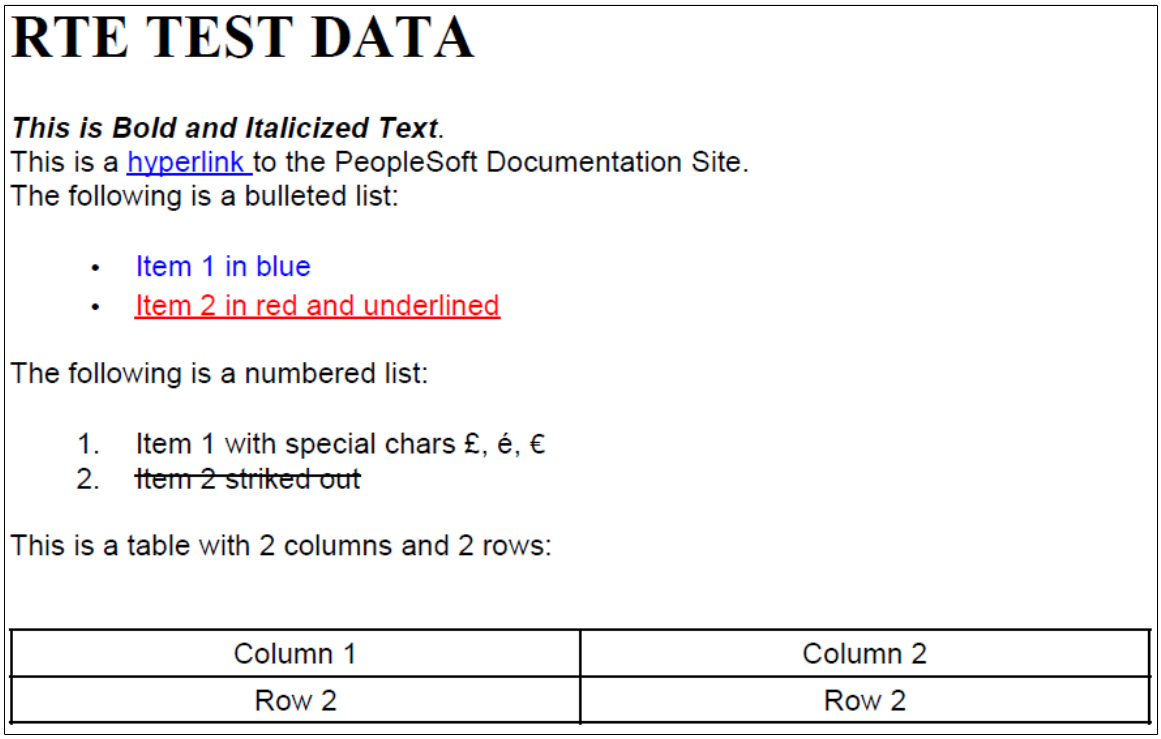


The following example shows a PDF output displaying RTE data:

Copyright © 1988, 2020, Oracle and/or its affiliates.

Image: PDF output

This example illustrates RTE data in a PDF output.



RTE TEST DATA

This is Bold and Italicized Text.
This is a hyperlink to the PeopleSoft Documentation Site.
The following is a bulleted list:

- Item 1 in blue
- Item 2 in red and underlined

The following is a numbered list:

1. Item 1 with special chars £, é, €
2. ~~Item 2 striked out~~

This is a table with 2 columns and 2 rows:

Column 1	Column 2
Row 2	Row 2

Configuring RTE on Page for BI Reporting

In order to use rich text in a BI Report, the rich text must be entered from a Rich Text long edit box on a PeopleSoft page.

PeopleSoft delivers a configuration file PT_RTE_CFG_PTXP that hides the Image buttons in the RTE toolbar.

To enable rich text editor functionality for the long edit box:

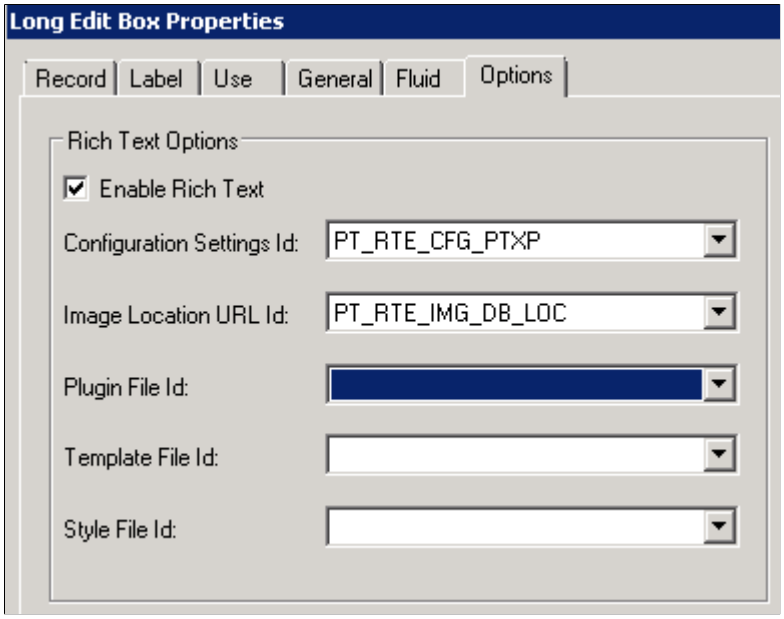
1. Add the long edit box to a page.
2. Double-click the control.

   Alternatively, you can highlight the control and select Page Field Properties from the Edit menu or the pop-up menu, which is activated by right clicking the control.
3. Access the Options tab and select Enable Rich Text Editor.
4. Select the Configuration Setting Id *PT_RTE_CFG_PTXP* or any other configuration file that hides the Image buttons.

Copyright © 1988, 2020, Oracle and/or its affiliates.

Image: Long Edit Box Properties dialog box

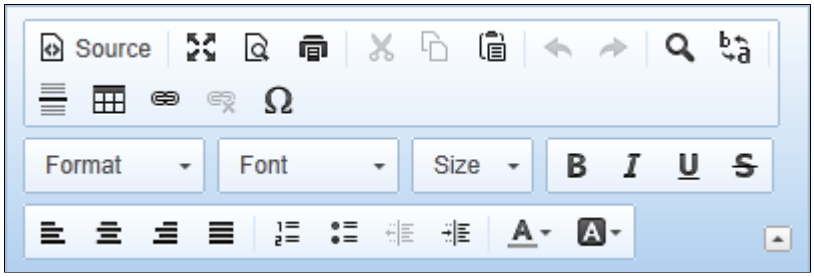
This example illustrates enabling rich text in the Long Edit Box Properties dialog box.



The toolbar displayed on the page will not allow Images.

Image: Rich text editor toolbar

This example illustrates the Rich text editor toolbar.



See "Creating Custom Configurations" (PeopleTools 8.58: Application Designer Developer's Guide).

Using CDATA in an XML File Generated Through XML File Layout

Utilizing BI Publisher's built-in support for Rich Text data (using html2fo function in form field) requires the use of CDATA sections for RTE data elements in an XML file. However, when using XML File Layout to create the XML file, there is currently no native support for creating these CDATA sections. As a result, the CDATA sections should be added manually prior to writing the XML file.

The following example illustrates how to manually add the CDATA sections.

Consider the following record QE_APPL_RESUME, having character fields IDENTIFIER and QE_RESUME_TEXT, the latter containing RTE data.

```
select * from PS_QE_APPL_RESUME;
IDENTIFIER    QE_RESUME_TEXT
```

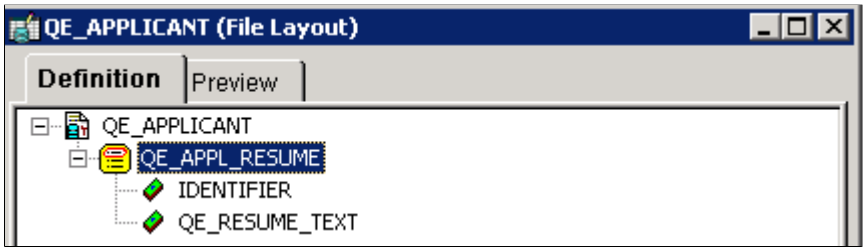
 Copyright © 1988, 2020, Oracle and/or its affiliates.

```
------------ ------------------------------------------------------------
12231        <p><strong>RTE Text</strong><br />containing <,>,", and &</p>
```

The following screenshot shows XML File Layout defined for this record.

Image: XML File Layout

This example illustrates the XML File Layout defined for the sample record.



The following PeopleCode is used to generate an XML file using the WriteRecord() method in XML File Layout:

```
Local Record &Rec;
Local File &fXML;
Local SQL &SQL;

&fXML = GetFile("c:\temp\QE_APPLICANT1.xml", "W", "UTF-8", %FilePath_Absolute);

If &fXML.IsOpen Then
   If &fXML.SetFileLayout(FileLayout.QE_APPLICANT) Then
      &Rec = CreateRecord(Record.QE_APPL_RESUME);
      &SQL = CreateSQL("%Selectall(:1)", &Rec);
      While &SQL.Fetch(&Rec)
         &fXML.WriteRecord(&Rec);
         &fXML.WriteLine("</QE_APPL_RESUME>");
      End-While;
   End-If;
End-If;

&fXML.Close();
```

The following is a snippet from the generated XML file:

```
<QE_APPL_RESUME>
<IDENTIFIER>12231</IDENTIFIER>
<QE_RESUME_TEXT>&lt;p&gt;&lt;strong&gt;RTE Text&lt;
/strong&gt;&lt;br /&gt;containing &lt;,&gt;,&quot;, and &amp;&lt;/p&gt;
</QE_RESUME_TEXT>
</QE_APPL_RESUME>
```

BI Publisher's built-in support for Rich Text data requires that RTE text must *not* be escaped with entity reference characters, but rather the RTE text must be in a CDATA block for the BIP Reporting Engine to be able to process it.

The way around this is to use WriteString() method instead of WriteRecord() and manually write the CDATA section for each individual field that contains RTE data:

The following snippet illustrates the modified PeopleCode:

```
Local Record &Rec;
Local File &fXML;
Local SQL &SQL;

&fXML = GetFile("c:\temp\QE_APPLICANT2.xml", "W", "UTF-8", %FilePath_Absolute);

If &fXML.IsOpen Then
```

Copyright © 1988, 2020, Oracle and/or its affiliates.

```
   If &fXML.SetFileLayout(FileLayout.QE_APPLICANT) Then
      &Rec = CreateRecord(Record.QE_APPL_RESUME);
      &SQL = CreateSQL("%Selectall(:1)", &Rec);
      While &SQL.Fetch(&Rec)
         rem &fXML.WriteRecord(&Rec);
         &fXML.WriteLine("<QE_APPL_RESUME>");
         &fXML.WriteString("<IDENTIFIER>");
         &fXML.WriteString(&Rec.IDENTIFIER.Value);
         &fXML.WriteString("</IDENTIFIER>");
         &fXML.WriteString("<QE_RESUME_TEXT><![CDATA[");
         &fXML.WriteString(&Rec.QE_RESUME_TEXT.Value);
         &fXML.WriteString("]]></QE_RESUME_TEXT>");
         &fXML.WriteLine("</QE_APPL_RESUME>");
      End-While;
   End-If;
End-If;

&fXML.Close();
```

It is essentially the same as the original code, but uses WriteString() to individually write out all fields from the record that contains the RTE field. This is necessary to be able to write the CDATA section string around the RTE field contents.

The following snippet is from the generated XML file:

```
<QE_APPL_RESUME>
<IDENTIFIER>12231</IDENTIFIER><QE_RESUME_TEXT>
<![CDATA[<p><strong>RTE Text</strong><br />containing <,>," and &</p>]]>
</QE_RESUME_TEXT></QE_APPL_RESUME>
<QE_APPL_RESUME>
```

 Copyright © 1988, 2020, Oracle and/or its affiliates.

Chapter 9

Attaching Digital Signature to PDF Reports

Understanding Digital Signature in BI Publisher Reports

BI Publisher for PeopleSoft enables you to add a digital signature to PDF reports so that you can verify the authenticity of the documents or reports you send and receive. Also, the receiver of the documents or reports can verify the authenticity of the documents or reports.

In order to attach a digital signature to a report, BI Publisher for PeopleSoft accesses your digital ID file from a central, secure location and at runtime signs the PDF report with the digital ID. The digital signature verifies the signer's identity and ensures that the document was not altered after it was created and signed.

BI Publisher for PeopleSoft supports attaching digital signatures only to PDF reports that are based on PDF or RTF templates.

Using Digital Signature in PDF Reports

A digital signature can be applied to PDF reports that are based on RTF and PDF templates.

This section discusses:

- Prerequisites for using digital signatures in PDF reports.
- Limitations in using digital signatures in PDF reports.
- Applying a digital signature to a PDF report.

Prerequisites

Before a user can use digital signatures on PDF reports in BI Publisher for PeopleSoft, the following prerequisites must be completed:

- Obtain a digital ID from a public certificate authority or from a private/internal certificate authority (if for internal use only).
- Store the digital ID inside a PFX file.
- Upload the PFX file and its related password to the PeopleSoft keystore.

Limitations

The following limitations are applicable when using digital signatures on PDF reports in BI Publisher for PeopleSoft:

Copyright © 1988, 2020, Oracle and/or its affiliates.

- Only a single digital ID can be used to sign a PDF document.
- Digital signature is enabled at the report level. Therefore, multiple templates assigned to the same report share the digital signature properties.

Attaching a Digital Signature to a Report

To apply a digital signature to a report:

1. Access the Report Definition page. (Reporting Tools >BI Publisher >Report Definition and click the Properties tab.)

   A user must have the XMLP Report Developer role to access the PDF Digital Signature property group.

   See Defining Global Properties for a description of the properties associated with the PDF Digital Signature property group.

2. Enable a digital certificate for the report.

3. Specify the position to place the digital signature on the report. You can specify the position of the digital signature by entering values for the properties related to location.

   - If your report is based on a PDF template, you can designate a field in which the digital signature should be placed.

     For information on designating a field in a PDF template for a digital signature, see *Report Designer's Guide for Oracle Business Intelligence Publisher,* "Creating PDF Templates," Adding or Designating a Field for a Digital Signature."

     Note: If you use a pre-existing field in the PDF template to hold the digital signature, the PDF template should be able to handle fields that can be updated. In this case, the PeopleTools psxp_pdf_optimized property is set to False for the report.
     For details of the psxp_pdf_optimized property, see Defining Global Properties and for information on updatable fields in PDF template, see Creating Submittable PDF Reports.

   - If you set the location of digital signature by entering values for the coordinates/width/height properties, the signature appears on the last page of the report and the coordinates will be calculated from the lower right corner of the report page.

4. Optionally, specify the name of PeopleSoft application class that will obtain the digital certificate ID necessary for retrieving the PFX file and it's password from the keystore.

   The optional application class must implement the IPT_PDFSIGNATURE_INT:IPDFSignature interface class, and the application class should be used for mapping specific report(s) to the correct signer's digital certificate. For details of the IPT_PDFSIGNATURE_INT:IPDFSignature interface class, see "Understanding the IPDFSignature Interface Class" (PeopleTools 8.58: PeopleCode API Reference).

 Copyright © 1988, 2020, Oracle and/or its affiliates.

Note: If the signer of a report should be set dynamically at report run-time, you must specify a PeopleSoft application class based on the IPT_PDFSIGNATURE_INT:IPDFSignature interface class. For example, a report may require different signers based on the report data. If the signer of a report is known at report design time, you must specify the digital ID of the report signer in the psxp_signature_digitalID property.

The digital signature functionality is tightly connected to the security functionality for external PFX certificates. For more information on security functionality, see "Using the External Digital Certificates Page" (PeopleTools 8.58: Security Administration) and "Understanding Applying Digital Signatures to PDF Report Output" (PeopleTools 8.58: Security Administration).

5. Specify a set of key fields used for mapping digital signature ID based on report field values.

At runtime, digital signature for bursted reports will either be attached individually for each report or a single signature for all bursted reports based on the definition in the application class.

Copyright © 1988, 2020, Oracle and/or its affiliates.

Copyright © 1988, 2020, Oracle and/or its affiliates.

Chapter 10

Emailing BI Publisher Reports to External Users

Emailing BI Publisher Reports to External Users

This topic discusses how to email non-bursted and bursted reports to external user email addresses.

Understanding Emailing BI Publisher Reports to External Users

There may be situations where you want to distribute reports using email to recipients that do not have PeopleSoft user profiles. PeopleTools enables you to distribute bursted and non-bursted BI Publisher reports to external users whose email addresses are stored in the database, file server, or other location.

A report developer creates an application class to retrieve the stored distribution lists, and then specifies the application class in the report definition.

Email Report Delivery

When you run and email a report to external users, the report is delivered as an email attachment in the output format defined in the report definition.

The email message that delivers the report contains the following default subject line text and default body text:.

Report Type	***Default Email Subject Text***	***Default Body Text***
Non-bursted	*BI Publisher <REPORT_NAME> Report* For example: `BI Publisher MONTHLY_SALES Report`	*BI Publisher Report <REPORT_NAME> Report Process ID: <PROCESS_ID_NUMBER>* For example: `BI Publisher MONTHLY_SALES Report Process ID: 391`
Bursted	*BI Publisher <REPORT_NAME> Report <BURST_VALUE>* For example: `BI Publisher MONTHLY_SALES Report 2500`	*BI Publisher Report <REPORT_NAME> <BURST_VALUE> Process ID: <PROCESS_ID_NUMBER>* For example: `BI Publisher MONTHLY_SALES Report 2500 Process ID: 391`

Copyright © 1988, 2020, Oracle and/or its affiliates.

Prerequisites to Email BI Publisher Reports to External Users

An SMTP server is required that allows sending email to external domains.

Developing Application Classes to Email BI Publisher Reports to External Users

As described previously, the email address that you use to send a report can reside on the database, a file server, or other source.

You need to develop an application class to retrieve the email addresses to which to send a report from storage. The application class that you develop should implement the IPT_EXT_BIP_EMAIL_INT: BIP_EMAIL_DATA application interface.

BI Publisher expects the application class to provide the following:

- An array of email addresses.
- An error string, if any.

The application class that you create is then defined as a report property in the report definition.

For more information about developing this application interface, see the *PeopleTools 8.55: PeopleCode API Reference* documentation.

Defining Report Definitions to Email BI Publisher Reports to External Users

To set up a report to email to external users you must ensure two parameters are set/defined in the report definition:

- The report output location must be set to email.
- An application class must be defined to retrieve the email addresses to which to send the report.

Setting the Report Output Location

To set the report output location to email:

1. Select Reporting Tools >BI Publisher >Report Definition.

   The Report Definition-Definitions page appears.

2. Click the Output tab.

   The Report Definition-Output page appears.

3. From the Location drop-down list, select Email.
4. Click the Save button.

Specifying Application Classes to Retrieve External Email Addresses

To specify an application class to retrieve external email addresses for report distribution:

1. Select Reporting Tools >BI Publisher >Report Definition.

Copyright © 1988, 2020, Oracle and/or its affiliates.

The Report Definition-Definitions page appears.

2. Click the Properties tab.

   The Report Definition-Properties page appears.

3. From the Property Group drop-down list, select PeopleTools Settings.

4. In the Text field for the psxp_ext_email_appclass property, enter the name of the application class that you've created to retrieve the external email addresses.

   The format to enter the application class name is:

   ```
   <PACKAGE_NAME>:<CLASS_NAME>
   ```

5. Click the Save button.

Copyright © 1988, 2020, Oracle and/or its affiliates.

Copyright © 1988, 2020, Oracle and/or its affiliates.

Chapter 11

Including External Attachments with BI Publisher Reports

Including External Attachments with BI Publisher Reports

This topic describes including external attachments with BI Publisher reports.

Understanding Including External Attachments with BI Publisher Reports

PeopleTools enables you to include attachments from external sources with BI Publisher PDF report output.

The source attachments are converted to PDF format and then merged with the main report. The report is then displayed to the end users or delivered to Report Manager.

To convert attachments to PDF output, BI Publisher utilizes Oracle's Outside In Technology (OIT), a product that enables the transformation and control of more than 600 file formats, including documents from office suites, specialty formats, and legacy files

This feature is available for any BI Publisher report with PDF output. The following template types are supported: RTF templates, PDF templates, and XSL templates.

Note: This feature is not supported on z/OS and Linux platforms.

Staging External Attachments for PDF Conversion and Merging with PDF Report Output

You must create a unique directory with a unique file path on a file system that the application server or process scheduler server can access and then stage the file to convert to PDF and merge with a BI Publisher report in the report output directory. The directory path should be unique for the current user and the current process.

Attachments are passed to the system in a string array and are included in the report in the order in which they are positioned in the array.

Setting Fonts and Other External Attachment Properties

You can set the fonts that are used to render text in external attachments and other converted attachment properties using the pspdfexport.cfg file.

The pspdfexport.cfg file is located in the <PS_HOME>\appserv directory.

To specify the font to use to render text, locate the #SCCOPT_FONTDIRECTORY section in the file and uncomment the *fontdirectory* property and specify the location of the fonts to use.

Copyright © 1988, 2020, Oracle and/or its affiliates.

The following example shows the #SCCOPT_FONTDIRECTORY section of the property file:

```
#SCCOPT_FONTDIRECTORY
#This option specifies the directory on the fonts that are to be used
#when rendering text
#This is a required option on unix - if not set in Windows, the fallback
#will be to use the default windows font directory.
#
#fontdirectory C:/WINDOWS/Fonts
```

See Oracle's *Outside In Technology* documentation for information on the properties in the pspdfexport.cfg file.

Passing Attachment Directories to BI Publisher

Passing external attachments to BI Publisher for converting to PDF and merging with the main report must be done prior to calling the BIP ProcessReport method.

To pass external attachments and files to BI Publisher, use the following PeopleCode:

```
&oRptDefn.setPDFConversionFiles(&inputFiles)
```

In this PeopleCode, `oRptDefn` is a report object already instantiated and `&inputFiles` is a string array representing paths to the external documents to include in the report as PDF files.

&inputFiles Array

For bursted and non-bursted reports provide an array where each array member specifies an absolute path to the single file or an absolute path to the directory that contains files to be converted.

By default, BI Publisher places the main report as the first document in the PDF report output, followed by the external attachments. The external attachments appear in the generated report in the order in which you define them in the array.

In all cases, the name of the merged output file (main report and external attachments) is the name of the BI Publisher report.

There are three variables that you can use in the &inputFiles array:

%REPORTFILE%

Use this variable to specify where the main report is positioned in the output.

This variable can be positioned only as the first or the last member of the &inputFiles array. If this rule is not followed the report file will be used as the first member of the array

If this placeholder is missing from the array it is added as the first array member during the processing.

%ALLFILES%

Use this variable to include all files in the directory. (This variable resolves to *.*)

It processes only files in the specified directory, child directories are ignored. Converted files appear in alphabetical order.

This variable is valid for bursted and non-bursted reports.

 Copyright © 1988, 2020, Oracle and/or its affiliates.

%BTV%	This variable denotes a burst value. This variable is valid only for bursted reports. If used for a non-bursted report, the array member is ignored.

Example: String Array of Fully-Qualified Paths

This example represents a string array of fully-qualified paths to the external documents to merge with the main report. In this example, the documents in the report output will appear in the order of the array members.

```
&inputFiles.Push("D:\ExternalFiles\sdk\samplefiles\microsoft-word.doc");
&inputFiles.Push("D:\Temp\Rownumber.htm");
&inputFiles.Push("D:\ExternalFiles\sdk\samplefiles\jpeg-bitmap.jpg");
&inputFiles.Push("D:\ExternalFiles\sdk\samplefiles\microsoft-excel.xls");
&inputFiles.Push("%REPORTFILE%");
/* in this example the main BI Publisher report is presented as the last */
/* document in the PDF output */
```

Example: String Array of Single Directories

This example represents an array of a single directory. All files contained in the directory will be converted to PDF documents and merged with the main report.

In this example the attachments in the report output will appear in alphabetical order. To achieve a specific order, rename the documents in the folder accordingly.

```
/* the main BIP report is presented as the first document in */
/* the PDF output */
&inputFiles.Push("%REPORTFILE%");
&inputFiles.Push("D:\Temp\ExternalFiles\%ALLFILES%");
/* Placeholder %ALLFILES% tells BI Publisher to use all documents contained */
/* in the folder D:\Temp\ExternalFiles */
```

Example: String Array of Multiple Directories (Bursted Reports)

This example shows an array of multiple directories named in accordance with the bursted value.

```
&inputFiles.Push("D:\Temp\ExternalFiles\%BTV%\%ALLFILES%");
```

In this case BI Publisher expects that directory *D:\Temp\ExternalFiles* has a set of sub-directories named as burst values. Each of the directories contains the documents for a specific bursted value.

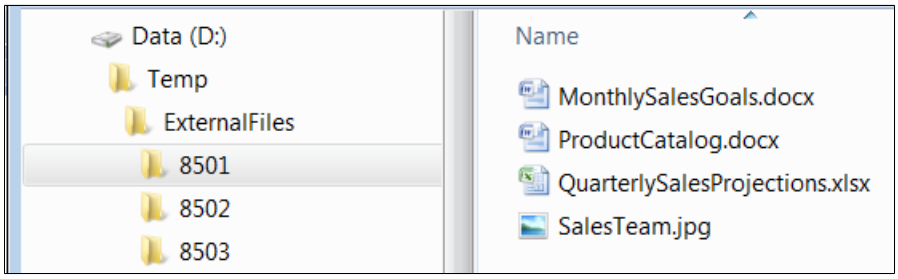
For each bursted report, the documents in the BI Publisher output will appear in alphabetical order. Note that the main report will appear first in the output report if there is no *%REPORTFILE%* placeholder in the &inputFiles array.

The following example shows the Microsoft Windows directory structure where a BI Publisher report is bursted by SALES_ID *8501*. In the example, the directory for *SALES_ID* is *D:\Temp\ExternalFiles\8501*.

Copyright © 1988, 2020, Oracle and/or its affiliates.

Image: Directory structure for bursted reports

This example illustrates a directory structure for a bursted report. All of the documents in the



Example: String Array of Multiple Directories and Multiple Static Files (Bursted Reports)

This example shows a string array that contains multiple directories named in accordance with the bursted value as well multiple static files to add to the bursted report.

This example also illustrates having a header page that separates a report from the converted PDF-formated attachments with a common text for each bursted report.

```
&inputFiles.Push("D:\Temp\ExteralFiles\attachment_header.rtf");
&inputFiles.Push("D:\Temp\ExternalFiles\%BTV%\%ALLFILES%");
```

To attach multiple static files to each bursted instance use the following syntax:

```
&inputFiles.Push("D:\Temp\ExternalFiles\COMMON_FILES\%ALLFILES%");
&inputFiles.Push("D:\Temp\ExternalFiles\%BTV%\%ALLFILES%");
```

In the previous example, the directory *D:\Temp\ ExternalFiles\COMMON_FILES* is a container for commonly used files for each bursted report instance.

Unicode Support for External Document Paths and Filenames

Unicode characters are supported in the external document paths and filenames.

This includes fully-qualified paths as well as directory paths containing files with Unicode characters in their names.

The following example shows Unicode characters in a full-qualified path:

```
&inputFiles.Push("D:\ExternalFiles\sdk\VEŘEJNÉ\microsoft-中word.doc");
```

The following example shows a directory path to a filename that contains Unicode characters:

```
&inputFiles.Push("D:\Temp\ExternalFiles\%ALLFILES%");
```

where *D:\Temp\ExternalFiles* contains a file such as *русский_язык_test.txt.*

Application servers and Process Scheduler servers running on Unix or Linux platforms require the character set portion of the shell locale to be one that supports Unicode, such as UTF-8.

Setting PDF Attachment Conversion Timeout

You can set a timeout value for attachment conversion and processing programmatically in PeopleCode or in BI Publisher's Global Properties page.

 Copyright © 1988, 2020, Oracle and/or its affiliates.

Understanding Setting PDF Attachment Conversion Timeout

When you set a timeout for PDF conversion, if the PDF conversion process is not finished within the specified time, the process is terminated, an error displays, and details are logged in the log file.

For bursted BI Publisher reports the conversion timeout value is used for report attachments for each bursted report instance

The PeopleCode runtime method overrides the value defined in the Global Settings and can be used in cases where relatively large files or a large numbers of files are expected. It also makes sense to increase the timeout interval at runtime in cases when a report is running via process scheduler (PRCS) and execution time is not a priority.

Note: Any programmatic setting made takes precedence over the setting on the Global Properties page.

Setting PDF Attachment Conversion Timeout Programmatically

To set the PDF attachment conversion timeout programmatically, use the following PeopleCode:

```
&oRptDefn.setPDFConversionTimeOut(&oitTimeOut);
```

Where `oitTimeOut` is a timeout value in seconds.

Setting PDF Attachment Conversion Timeout in the Global Properties

You must have the Report Developer role to set the timeout property in the Global Properties.

To set the PDF attachment conversion timeout in the Global Properties:

1. Select Reporting Tools >BI Publisher >Setup >Global Properties.

   The Global Properties page appears.

2. From the Property Group drop-down list, select *PeopleTools Settings*.

3. For the psxp_pdfconversion_timeout property, enter a value (in seconds) in the Text column. The default value is *20* seconds.

4. Click the *Save* button.

Example: Implementation Example

This PeopleCode example illustrates code for creating a report with several attachments merged into the report.

```
import PSXP_RPTDEFNMANAGER:*;

Local PSXP_RPTDEFNMANAGER:ReportDefn &oRptDefn;
Local string &sRptDefn;
Local string &sTemplateId;
Local date &dAsOfDate;
Local Record &rcdQryPrompts;

try
   &sRptDefn = PSXPRPTSRCH_VW.REPORT_DEFN_ID.Value;
   &sTemplateId = PSXPQRYVIEW_WRK.TMPLDEFN_ID.Value;
   &dAsOfDate = PSXPQRYVIEW_WRK.ASOFDATE.Value;

   /* create report defn object */
```

Copyright © 1988, 2020, Oracle and/or its affiliates.

```
   &oRptDefn = create PSXP_RPTDEFNMANAGER:ReportDefn(&sRptDefn);
   &oRptDefn.Get();


   /* output format */
   &sOutputFormat = &oRptDefn.GetOutDestFormatString(Value
   (PSXPQRYVIEW_WRK.OUTDESTFORMAT.Value));

   /* fill prompt record */
   &rcdQryPrompts = &oRptDefn.GetPSQueryPromptRecord();
   If Not &rcdQryPrompts = Null Then
      If Not Prompt(&oRptDefn.GetDatasource().Name, "", &rcdQryPrompts)
    Then
         Exit;
      End-If;
      &oRptDefn.SetPSQueryPromptRecord(&rcdQryPrompts);
   End-If;

   /* If we uncomment 2 statements below output file is generated  */
  /* in specified location. It should be cleaned up later by Apps */
   rem &oRptDefn.OutDestination = "d:\Temp\BIPoutputdir\" | UuidGen();
   rem CreateDirectory(&oRptDefn.OutDestination, %FilePath_Absolute);

/* Creating an array of docs to convert. In this case actual report*/
 /* came as the last one */
 If (&oRptDefn.ID = "COUNT_TEST" Or  /* Testing specific reports */
        &oRptDefn.ID = "A_QE_UPGRADE") And
        &sOutputFormat = "PDF" Then
     Local array of string &inputFiles;
     &inputFiles = CreateArrayRept("", 0);

     &inputFiles.Push("D:\ExternalFiles\sdk\samplefiles\
    microsoft-word.doc");
     &inputFiles.Push("D:\Temp\Rownumber.htm");
     &inputFiles.Push("D:\ExternalFiles\sdk\samplefiles\
    jpeg-bitmap.jpg");
     &inputFiles.Push("D:\ExternalFiles\sdk\samplefiles\
    microsoft-excel.xls");
     &inputFiles.Push("%REPORTFILE%");

    /*setting runtime timeout for conversion process */
     Local number &oitTimeOut = 15;
     &oRptDefn.setPDFConversionFiles(&inputFiles);
     &oRptDefn.setPDFConversionTimeOut(&oitTimeOut);
  End-If;

 &oRptDefn.ProcessReport(&sTemplateId, %Language_User, &dAsOfDate,
  &sOutputFormat);

  CommitWork();
  /* display the output */
  &oRptDefn.DisplayOutput();

 catch Exception &Err
   If Not &oRptDefn = Null Then
      &oRptDefn.Close();
    End-If;

    Error MsgGet(&Err.MessageSetNumber, &Err.MessageNumber, &Err.ToString());
end try;
```

 Copyright © 1988, 2020, Oracle and/or its affiliates.

Chapter 12

Applying PeopleSoft Regional Settings Personalizations to BI Publisher Reports

Applying PeopleSoft Regional Settings Personalizations to BI Publisher Reports

This topic discusses how to:

- Applying PeopleSoft regional settings personalizations to online reports in PIA.
- Personalize reports run through process scheduler.

Understanding Applying PeopleSoft Regional Settings Personalizations to BI Publisher Reports

When displaying date, date-time, time or number values in a report, you may format these values according to the PeopleSoft Regional Settings formatting options.

Parameters

To apply PeopleSoft regional settings personalizations to BI Publisher reports, PeopleSoft provides the following parameters to the BI Publisher core engine at runtime:

- $PsftDateTime
- $PsftDateTimeTZ
- $PsftDate
- $PsftTime

These parameters contain the PeopleSoft specific formatting mask for PeopleSoft date-time, date, and time fields. You may declare and use these parameters in the template in conjunction with the format-date() function. For example:

- <?format-date:QEQ_HIST_DT;$PsftDate?>
- <?format-date:DTTM_STAMP;$PsftDateTime?>
- <?format-date:DTTM_STAMP;$PsftDateTimeTZ?>

Prior to using these parameters, you must declared them under a form field at the top of the reports primary template, one for each of the parameters being used as shown in the following example:

```
<xsl:param name="PsftDate" xdofo:ctx="begin"/>
<xsl:param name="PsftDateTime" xdofo:ctx="begin"/>
<xsl:param name="PsftDateTimeTZ" xdofo:ctx="begin"/>
```

Copyright © 1988, 2020, Oracle and/or its affiliates.

```
<xsl:param name="PsftTime" xdofo:ctx="begin"/>
```

Time Value Formats

To format time values, you must prefix an arbitrary date value to the time before you can successfully format it. This is necessary to ensure that the date-time value passed to the format-date() function is in the required canonical format. The following example shows prefixing a date component to the runtime value and formatting it using the $PsftTime parameter:

```
<?if:RUNTIME!=' ' and RUNTIME!=''?>
<?format-date:concat('1900-01-01T',RUNTIME);$PsftTime?>
<?end if?>
```

Number Value Formats

Formatting a number value does not require the use of a parameter. It only requires that an Oracle number format mask, including the grouping separator (G) and decimal separator (D), be specified in the format-number() function, as shown in the following example:

```
<?format-number:AMOUNT;'999G999D99'?>
```

See *Report Designers Guide for Oracle Business Intelligence Publisher,* “Formatting Numbers” for more information.

Formatting Masks

The following are the PeopleSoft regional settings formatting options taken into consideration for the PeopleSoft formatting masks:

- Time Separator
- Date Separator
- Local Time Zone
- Date Format
- Digit Group Separator
- Decimal Separator
- Time Format

Dependencies

Note the following dependencies for applying PeopleSoft regional settings personalizations to BI Publisher reports:

- The report template type must be RTF or XSL.
- The format-date() function requires a date-time value in the canonical format. For example:

  ```
  Date:  2008-07-28
  Date-Time:  2008-07-28T09:00:00-0700
  Time:  1900-01-01T05:55:00
  ```

  Data from the *T* onwards is optional. If not included, BI Publisher assumes it represents 12:00 AM UTC.

 Copyright © 1988, 2020, Oracle and/or its affiliates.

For example:

```
2008-07-28T00:00:00-0000
```

Applying PeopleSoft Regional Settings Personalizations to Online Reports in PIA

This section discusses:

- Applying regional settings personalizations to online reports in PIA.
- Using the session language code to determine number formatting.
- An example of applying PeopleSoft regional settings personalizations to BI Publisher reports using the en-us locale
- An example of applying PeopleSoft regional settings personalizations to BI Publisher reports using the fr locale.

Applying PeopleSoft Regional Settings Personalizations to Online Reports in PIA

When a BI Publisher report is run online using the PeopleCode APIs, or via the Query Report Viewer PIA page, the personalization settings are determined based on the following three levels:

- User-specific.
- Locale-specific.
- System-wide.

The user-specific settings are those that are specified by overriding any of the default values displayed under your My Preferences - Regional Settings page, and have the highest level of precedence.

If no overrides have been defined for a user, the personalization settings are driven by the locale that is set in the browser, assuming that personalization options have been configured for this specific locale in the PeopleSoft system. See "Defining Locale-Specific Formatting" (PeopleTools 8.58: Global Technology) for more information.

If no locale-specific personalization settings have been defined for the particular locale in use, the system-wide defaults will then be applied. The system-wide defaults are the default values displayed under your My Preferences – Regional Settings page prior to specifying any overrides.

Using the Session Language Code to Determine Number Formatting

The PeopleTools Settings property group features a psxp_number_format_src property that lets you modify the behavior regarding the personalization of number values. It effectively allows you to revert to the method used in PeopleTools 8.55 and earlier releases. In PeopleTools 8.55 and earlier releases, you can use the language code of the current user session to determine the decimal and digit group separators used when formatting a number with the format-number() function.

Note that this property does not have any effect on reports run through the process scheduler, as number formatting in a scheduled report will *always* be driven by the language code of the process run control. For more information see section *Personalizing Reports Run Through Process Scheduler* later in this topic for more information.

Copyright © 1988, 2020, Oracle and/or its affiliates.

By default, the psxp_number_format_src property is set to *Browser,* and the formatting of number values is based on the PeopleSoft Regional Settings. You can set the property to *LangCode* to put into effect the older behavior.

The property is located in the PeopleTools Property Group on the Global Properties page or on the Report Definition page - Properties page. See Setting Up BI Publisher for more information.

You can also set the property using the ReportDefn class method, SetRuntimeProperties, in the PeopleCode API. See "SetRuntimeProperties" (PeopleTools 8.58: PeopleCode API Reference)

Example: Applying PeopleSoft Regional Settings Personalizations Using the en-us Locale

The following examples illustrate reports with PeopleSoft formatted date-time, number, date and time values, when run in a browser using the *en-us* locale

Note that the examples in this section reflect the regional settings as delivered for the en-us locale, for example: the date format is MMDDYY, the time format uses the 12 hour clock with AM/PM designator, the number has a comma for the digit group separator, and a decimal for the decimal separator.

Image: BI Publisher Report QEQ_LOCALEQUERY with PeopleSoft Regional Settings (en-us Locale)

The following example illustrates BI Publisher Report, QEQ_LOCALEQUERY, formatted with PeopleSoft regional settings using the *en-us* locale.

QEQ_LOCALEQUERY

DTTM_STAMP	QEQ_HIST_AMT	QEQ_HIST_DT
02/11/2005 03:16:46 PM PST	-88,569.90	03/05/2008
02/11/2005 03:16:46 PM PST	-88,569.90	03/05/2008
02/11/2005 03:16:46 PM PST	-88,569.90	04/09/2009
02/11/2005 03:16:46 PM PST	-88,569.90	04/09/2009

Image: BI Publisher Report QE_QRY_PRCS_TIME with PeopleSoft Regional Settings (en-us Locale)

The following example illustrates BI Publisher Report, QE_QRY_PRCS_TIME, formatted with PeopleSoft regional settings using the *en-us* locale.

QE_QRY_PRCS_TIME

SCHEDULENAME	RUNTIME
QE_nVision_SameDrillDn_3Times	03:33:00 AM
QE_QUERY_STRESS	04:05:47 PM
QE_QUERY_STRESS	04:06:00 AM

Example: Applying PeopleSoft Regional Settings Personalizations Using the fr Locale

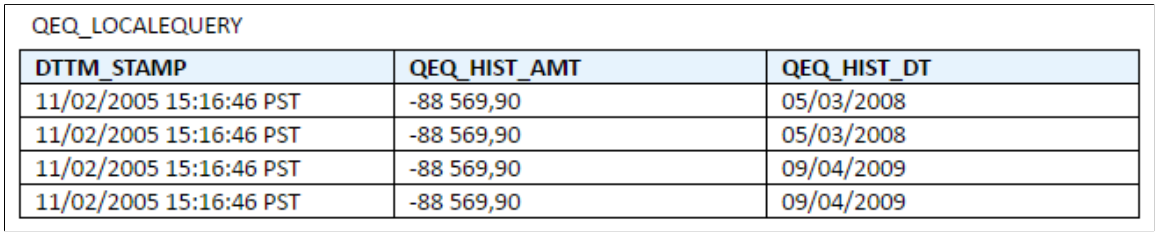
The following examples illustrate Reports with PeopleSoft-formatted date-time, number, date and time values, when run in a browser using the *fr* locale

 Copyright © 1988, 2020, Oracle and/or its affiliates.

Note that the examples in this section reflect the regional settings as delivered, for the fr locale, for example: the date format is DDMMYY, the time format uses the 24 hour clock, and the number has a space for the digit group separator, and a comma for the decimal separator.

Image: BI Publisher Report QEQ_LOCALEQUERY with PeopleSoft Regional Settings (fr Locale)

The following example illustrates BI Publisher Report, QEQ_LOCALEQUERY, formatted with PeopleSoft regional settings using the *fr* locale.

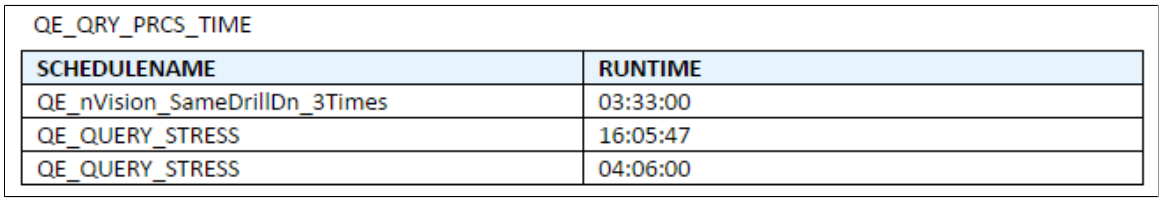


QEQ_LOCALEQUERY

DTTM_STAMP	QEQ_HIST_AMT	QEQ_HIST_DT
11/02/2005 15:16:46 PST	-88 569,90	05/03/2008
11/02/2005 15:16:46 PST	-88 569,90	05/03/2008
11/02/2005 15:16:46 PST	-88 569,90	09/04/2009
11/02/2005 15:16:46 PST	-88 569,90	09/04/2009

Image: BI Publisher Report QE_QRY_PRCS_TIME with PeopleSoft Regional Settings (en-us Locale)

The following example illustrates BI Publisher Report, QE_QRY_PRCS_TIME, formatted with PeopleSoft regional settings using the *fr* locale.



QE_QRY_PRCS_TIME

SCHEDULENAME	RUNTIME
QE_nVision_SameDrillDn_3Times	03:33:00
QE_QUERY_STRESS	16:05:47
QE_QUERY_STRESS	04:06:00

Applying PeopleSoft Regional Settings Personalizations to Reports Run Through Process Scheduler

When reports are run through the Process Scheduler, there is no browser locale to consider, and personalization depends only on the user-specified overrides or system defaults from the My Preferences - Regional Settings page. This is determined specifically for the user who scheduled the report.

Note however that for a scheduled report, the regional settings only apply to the date, date-time and time values. For the purpose of maintaining backwards compatibility, a number value in a scheduled report will always be determined by the language code that is associated with the process run control. This is meant to avoid impacting already-existing behavior regarding the formatting of number values in a scheduled report. Essentially, the language code is used to obtain the associated ISO locale code from the PSLANGUAGES table (for example, ENG=en, FRA=fr, etc.), which is then used by the BIP engine while processing the report. This determines the appropriate locale-specific grouping separator (G) and decimal separator (D) to be used when a number is formatted with the format-number() function.

For example, when running with language code *FRA,* the outcome of this form-field syntax:

```
<?format-number:AMOUNT;'999G999D99'?>
```

where `AMOUNT = 1234.56` will be `1 234,56.`

Copyright © 1988, 2020, Oracle and/or its affiliates.

 Copyright © 1988, 2020, Oracle and/or its affiliates.

Appendix A

Securing BI Publisher

BI Publisher Security

BI Publisher security can be separated into three categories:

- Defining reports.
- Running reports.
- Viewing reports.

When you are defining Query-based reports, Query security determines which queries you can access and select from to create your BI Publisher report definitions. Security for editing and viewing report definitions is controlled by the Report Category ID attribute, which is set on theReporting Tools > BI Publisher > Setup > Report Category page.

Security for running and viewing BI Publisher reports is controlled by setting options in a number of places. This table illustrates where security can be set:

Activity	***Security Settings***	***Query-based reports (Non-Bursted)***	***Query-based reports (Bursted)***	***Non-Query-based reports (Non-Bursted)***	***Non-Query-based reports (Bursted)***
Running Reports	Query Security	X	X	NA	NA
Running Reports	Application Security	X	X	X	X
Running Reports	Process Scheduler Security	X	X	X	X
Viewing Report Definitions	Report Definition > Security page	X	X	X	X
Viewing Report Definitions	Report Definition > Bursting page	NA	X	NA	X

Application security and Process Scheduler security determine who can run reports. BI Publisher does not provide additional security beyond what Oracle currently provides. That means that the component security of the data extraction program drives access control to the associated reports. For processes, process security prevails and for queries, query security prevails. When you are running a Query-based report, the requester's row-level security to the underlying data source always applies.

Copyright © 1988, 2020, Oracle and/or its affiliates.

Query-based reports viewed online in real time from the Query Report Viewer respect query access groups for the user's primary permission list. For non-Query-based reports viewed online in real time, security is controlled by the application.

When you are viewing a report that was run through either the Query Report Scheduler or the Process Scheduler, security is controlled by both the Distribution ID field on theReport Definition > Security page and, when theAllow viewer ID assignment at report runtime check box is selected, by those IDs selected at runtime on theProcess Scheduler Request > Distribution Detail page. Additional viewing security can also be defined for bursted reports on theReport Definition > Bursting page.

If no viewers are designated on the Report Definition > Security page and theAllow viewer ID assignment at report runtime is selected, then the report requestor's ID is applied as a viewer by default at runtime. This applies to bursted reports as well.

See Setting Up Report Categories, Creating Report Definitions, "Scheduling Process Requests" (PeopleTools 8.58: Process Scheduler).

 Copyright © 1988, 2020, Oracle and/or its affiliates.

Appendix B

Migrating BIP Definitions

BIP Definitions Overview

To facilitate the movement of reports and templates from development to test and then to production, BI Publisher (BIP) objects are available as managed objects that can be placed into projects for migration from database to database. To facilitate the location of report-related objects, these items can be identified based on object owner ID.

Migrating BIP Definitions Using ADS

PeopleSoft provides Application Data Set (ADS) definitions to migrate Business Intelligence Publisher definitions from one environment to another as a project file.

For information on using Data Set Designer and other functionality of data sets, see Lifecycle Management Guide.

The following definition types can be added to projects in Application Designer:

- BIP Data Source Definitions
- BIP File Definitions
- BIP Report Definitions
- BIP Template Definitions

If the data source for the BIP report is PS Query, Connected Query, or Composite Query, then the query, connected query, or composite query definition should also be included in the project.

Note: Because BI Publisher is based on managed objects, all updates to your data need to be performed using the PeopleSoft BI Publisher Pure Internet Architecture pages, PeopleSoft Application Designer, or BI Publisher PeopleCode APIs.

See "Understanding Related Language Tables" (PeopleTools 8.58: Global Technology), "PeopleSoft Pure Internet Architecture Fundamentals" (PeopleTools 8.58: Portal Technology).

ADS Tables for BIP Definitions Migration

The following ADS definition tables can be used for migrating BIP Definitions.

ADS Definition	*Description*
BIP_DEFINITION	Table for BIP definitions.

Copyright © 1988, 2020, Oracle and/or its affiliates.

ADS Definition	*Description*
BIP_TEMPLATE	Table for BIP templates.
BIP_DATA_SOURCE	Table for BIP data sources.
BIP_FILE_DEFINITION	Table for BIP file definitions.

Tables for BI Publisher Definitions

The BIP_DEFINITION uses the following tables.

Table	*Description*
PSXPRPTDEFN	Stores BIP Report Definitions.
PSXPRPTDEFNDEL	Stores BIP Report Definitions for report deletions.
PSXPRPTDEFNLNG	Stores BIP Report Definitions related language information.
PSXPRPTOUTFMT	Stores BIP Report output formats.
PSXPRPTPROP	Stores BIP Report properties.
PSXPRPTSCOPEFLD	Stores BIP Report burst security field mapping information.
PSXPRPTSRCHKEYS	Stores BIP Report Definition search keys.
PSXPRPTTMPL	Stores BIP Report template relationships.
PSXPRPTTMPLCTRL	Stores BIP Report template control tables.
PSXPRPTVIEWER	Stores BIP Report viewer tables.

Tables for BI Publisher Template Definitions

The BIP_TEMPLATE uses the following tables.

Table	*Description*
PSXPTMPLDEFN	Stores BIP Template Definitions.
PSXPTMPLDEFNDEL	Stores BIP Template Definition for template deletions.
PSXPTMPLDEFNLNG	Stores BIP Template Definition related language information.
PSXPTMPLFILEDEF	Stores BIP Template file definitions.
PSXPTMPLTRINFO	Stores BIP Template translation information.

 Copyright © 1988, 2020, Oracle and/or its affiliates.

Tables for BI Publisher Data Source Definitions

The BIP_DATA_SOURCE uses the following tables.

Table	*Description*
PSXPDATASRC	Stores BIP Data Source definitions.
PSXPDATASRCDEL	Stores BIP Data Source deletions.
PSXPDATASRCLNG	Stores BIP Data Source related language information.
PSXPSCHEMAFLMN	Stores BIP Data Source sample schema file information/data.
PSXPSMPLDTMN	Stores BIP Data sample XML data file information/data

Tables for BI Publisher File Definitions

The BIP_FILE_DEFINITION uses the following tables.

Table	*Description*
PSFILEDEFN	Stores BIP File definitions.
PSFILEDATA	Stores BIP File data information.
PSFILEDEL	Stores BIP Data File deletions.

Migrating BI Publisher-Translated Languages

BIP template translation uses related XLIFF files (one for each language) that contain not only specific translation pairs but the whole template definition. This is a standard for using XLIFF translation methodology.

Because the translation is tied to the template definition, the translation file (BIP Template Definition) as well as the specific XLIFF files (BIP File Definition) should be included in the project.

Note: If the template file is not copied with the language files, the correct translation file cannot be used when you run the report. Run a SYSAUDIT for Audit BI Publisher Integrity and delete any orphaned definitions.

See "BI Publisher Integrity" (PeopleTools 8.58: Data Management).

Cleaning Up BI Publisher Metadata

This section describes how to clean up BI Publisher metadata by running the PSXPCLEAN Application Engine program.

Copyright © 1988, 2020, Oracle and/or its affiliates.

Understanding the PSXPCLEAN Application Engine Program

To ensure the integrity of the BI Publisher files, run the Application Engine program PSXPCLEAN.

This Application Engine program finds:

- Un-referenced objects in PSFILEDEFN.
- Template definitions and template translations for which file objects are missing.
- Inconsistencies between PSFILEDEFN and PSFILEDATA.
- Temporary BI Publisher files.

Understanding Temporary BI Publisher Files

When a user runs a BI Publisher report, the system creates temporary files in the database (with a *.vazip* extension) and delivers the reports to the user's browser. After the report is rendered on the user's browser, the system deletes the temporary files in the database.

However, in some cases the user's browser might be set to block pop-ups. In these cases the system creates the temp files in the database, but the process of rendering the reports and then removing the temp files cannot complete due to pop-ups. As a result temp files can accumulate in the database, wasting space.

Running the PSXPCLEAN Application Engine Program

The PSXPCLEAN application engine program is delivered in *Report and Delete* mode. To run the program in *Report Only* mode, open the application engine program PSXPCLEAN in Application Designer and remove the comment in the following statement in PSXPCLEAN:Main:Start PeopleCode action:

```
rem PSXPFILECLN_AET.REPORT_ONLY_FLAG = "Y"
```

You can schedule and run PSXPCLEAN using the PeopleTools, Process Scheduler, System Process Request page. You should run this program on a regular basis to keep template metadata consistent.

 Copyright © 1988, 2020, Oracle and/or its affiliates.

Appendix C

Deleting Report Definitions

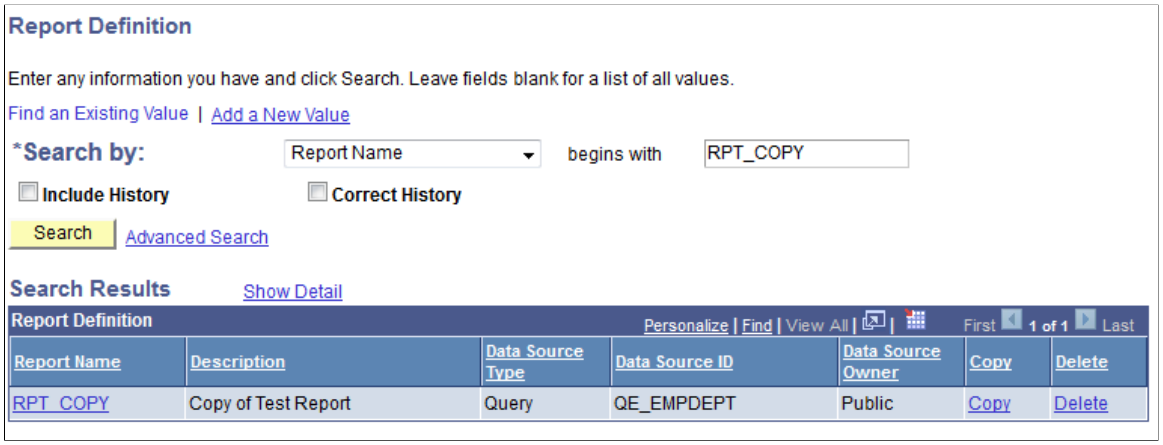
Deleting Report Definitions

For a report developer to delete a BI Publisher report definition, he or she must have the *XMLP Report Developer* role with permission list *PTPT2600* assigned his or her user profile.

To access the Report Definition search page, select Reporting Tools >BI Publisher >Report Definition.

Image: Report Definition Search page

This example illustrates the fields and controls on the Report Definition Search page.



After you enter criteria to search for a report definition to delete, the results appear in the Report Definition grid.

In the Report Definition grid, locate the row that displays the name of the report to delete and click the Delete link located on the right side of the row.

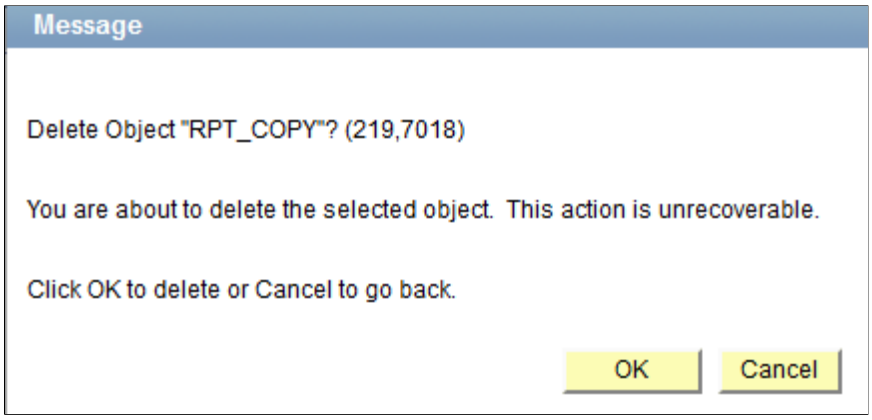
Note: The Copy and Delete links appear in the Report Definition results grid only if the XMLP Report Developer role is assigned to your user profile.

When you click the Delete link in the grid a delete action confirmation message appears.

Copyright © 1988, 2020, Oracle and/or its affiliates.

Image: Report definition delete confirmation message

This example illustrates the action confirmation message that appears when you click the Delete button in the Report Definition results grid.



When the message box appears, perform one of these actions:

- Click the OK button to confirm the deletion of the report definition.
- Click the Cancel button to cancel the action.

Copyright © 1988, 2020, Oracle and/or its affiliates.