



BI Publisher Sub-Templates

PT 8.59

Randall Groncki

Introduction

BI Publisher Sub-Templates are reusable code blocks for RTF templates.

Sub-Templates allow us to create a commonly used code block such as headers with logos and other information instead of recreating that header on every report. When an item in that header changes, we only need to change the header sub-template instead of every report that uses that common header.

Additionally, we can conditionally call sub-templates as different headers for a common document to apply appropriate branding. Think of a State University system with eighteen schools on a common PeopleSoft HCM instance. Instead of 18 versions of a common letter, we can have one letter using the appropriate branding as per the school. If the letter changes, one template needs updated and tested, not eighteen.

We can also use sub-templates as conditional, reusable blocks in our reports. Think of an address block or a benefits block. That block can be used every time or conditionally included on each individual report depending on the data in that report.

BI Publisher can invoke both RTF and XSL sub-templates. This document will only explore RTF sub-templates.

Creating Sub-Templates

* See Appendix A for correcting a bug in the Oracle BI Publisher Desktop directly affecting sub-template testing.

RTF Sub-Templates are RTF documents with one or more sub-templates defined with the `<?template:xxxx?>` `<?end template?>` tags.

There are a few rules to follow when creating sub-templates:

- The template XSL is plain text, not embedded fields.
- The tags themselves must have the "Normal" styling in the word document
- Everything between the template tags is consumed by the calling document
- An RTF document with sub-templates can have multiple sub-templates
- Each template name must be unique within that sub-template document.

```
<?template:pt3_stdhdr?>  
  Header text and objects here  
<?end template?>
```

- Sub-Templates can call other sub-templates

This is a simple example of a sub-template for a standard header for our fictional company:

<?template:pt3_stdhdr?> Big, Generic Conglomerate Holding Corp Std Header for All Companies Located everywhere in the world <?end template?>	Page 1 of 1 1/26/2022
---	------------------------------

This sub-template has the company logo and address, page numbers and print date. Each report using this sub-template will have the same header. When the logo or anything else in this header changes, then just this sub-template needs updated, not every report calling it.

This is an example of a sub-templet file with multiple sub-templates.

<?template:pt3_stdhdr?> Big, Generic Conglomerate Holding Corp Std Header for All Companies Located everywhere in the world <?end template?>	Page 1 of 1 1/26/2022
<?template:pt3_hdr_GBI?> Global Business Institute GBI Located everywhere in the world <?end template?>	Page 1 of 1 1/26/2022
<?template:pt3_hdr_KAM?> Aust Manufacturing Industries KAM Located everywhere in the world <?end template?>	Page 1 of 1 1/26/2022
<?template:pt3_hdr_KF1?> Business Institute - France KF1 French Division <?end template?>	Page 1 of 1 1/26/2022

Calling Sub-Templates in a Report

Using sub-templates in a template is a two-step process:

- Importing the sub-template document
- Calling the specific sub-template

This is simple employee training report using a sub-template as a common header.

for-each

<?import:file:C:///psreports/pt3_stdhdr.rtf?>
<?call-template: pt3_stdhdr?>

Training Status

Employee:

fld_EMPLID

fld_NAME

Company:

fld_COMPANY

fld_COMPANY_DESCR

Location:

fld_LOCATION

fld_LOCATION_DESCR

Job:

fld_JOBCODE

fld_JOB_DESCR

Course	TITLE	Start Date	Session	Grade
F fld_COURSE	fld_COURSE_TITLE	START_DT	Sessn	GRADE E

page breakend

All import and call commands are plain text XSL that must be in the documents “Normal” style.

Import template

Importing a template is different depending on whether you are testing the report in the BI Publisher Desktop or invoking the report from within the PeopleSoft application.

For local testing, the import statement must point to the local directory and file where the sub-template is stored.

```
<?import:file:C:///psreports/pt3_stdhdr.rtf?>
```

Notice the three slashes “///” after the drive letter. This is calling a sub-template file “pt3_stdhdr.rtf” that resides on the local machine’s C:/psreports directory.

The import statement changes when called from within the PeopleSoft application.

```
<?import:psxmlp://PT3_STDHDR?>
```

The import is looking for the same the same template within PeopleSoft's BI Publisher Sub-Template repository.

Calling a template

Call the specific sub-template from within the imported file. The names must be an exact match.

This is an example of an unconditional sub-template call:

```
<?call-template:pt3_stdhdr?>
```

Conditionally calling sub-templates

BI Publisher's XSL allows us to call different sub-templates based on data within the report.

If Statement

The if statement tests one Boolean condition. If the result is true, then all commands within the construct before then `<?end if?>` are executed.

```
<?if:[Boolean condition]?>  
    [code]  
    [code]  
    [...]  
<?end if?>
```

Here is an example of an "If" statement.

```
<?if:fld_FIRST_CHILD_FLAG="Y"?>  
<?call-template:trn_sesn_tbl?>  
<?end if?>
```

It evaluates the value of the `fld_FIRST_CHILD_FLAG` field. If the value = "Y", then the "trn_sessn_tbl" sub-template is called.

Choose/When/Otherwise Statement

This construct works much the same as an “Evaluate” statement in PeopleCode. We can test for multiple conditions instead of just one.

The first condition to evaluate as true ends the statement.

```
<?choose:?>
<?when:[Boolean condition]?>
    [code]
<?end when?>
<?when:[Boolean condition]?>
    [code]
<?end when?>
<?otherwise:?>
    [code]
<?end otherwise?>
<?end choose?>
```

We can add as many <?when?> blocks as we like. The first true condition executes its code and ends the <?choose?> block.

The <?otherwise?> block executes if none of the <?when?> statements evaluated true.

This below example is a report that is using different branding in its headers depending on the value of the “fld_COMPANY” value. There are specific templates for each of three companies (GBI, KAM & KF1). Any other value in that field falls through to the <?otherwise?> block and uses the generic, standard header for the report

```
<?choose:?>
<?when:fld_COMPANY ='GBI'?>
<?call-template:pt3_hdr_GBI?>
<?end when?>
<?when:fld_COMPANY ='KAM'?>
<?call-template:pt3_hdr_KAM?>
<?end when?>
<?when:fld_COMPANY ='KF1'?>
<?call-template:pt3_hdr_KF1?>
<?end when?>
<?otherwise:?>
<?call-template:pt3_stdhdr?>
<?end otherwise?>
<?end choose?>
```

Here is an example of an rtf report template conditionally calling sub-templates:

- The header sub-template is chosen based on the value of the fld_COMPANY
- The body detail section of the report is selected based on fld_FIRST_CHILD_FLAG field, which tells whether this employee has class session detail in the TRAINING table.
- Notice that two different sub-templates are called

```
for-each
<?import:file:C:///psreports/pt3_stdhdr_mult.rtf?>

<?choose:?>
<?when:fld_COMPANY='GBI'?>
<?call-template:pt3_hdr_GBI?>
<?end when?>
<?when:fld_COMPANY='KAM'?>
<?call-template:pt3_hdr_KAM?>
<?end when?>
<?when:fld_COMPANY='KF1'?>
<?call-template:pt3_hdr_KF1?>
<?end when?>
<?otherwise:?>
<?call-template:pt3_stdhdr?><?end otherwise?><?end choose?>
```

Training Status

Employee:	fld_EMPLID	fld_NAME
Company:	fld_COMPANY	fld_COMPANY_DESCR
Location:	fld_LOCATION	fld_LOCATION_DESCR
Job:	fld_JOBCODE	fld_JOB_DESCR

Child Rows fld_FIRST_CHILD_FLAG

```
<?import:file:C:///psreports/pt3_rpt_sections.rtf?>
<?if:fld_FIRST_CHILD_FLAG="Y"?>
<?call-template:trn_sesn_tbl?>
<?end if?>
<?if:fld_FIRST_CHILD_FLAG="N"?>|
<?call-template:trn_sesn_none?>
<?end if?>
```

page breakend

Margins and Layout

A best practice is to use the same margins and page layout in the sub-templates as the reports calling them. This is not required, but odd formatting can result if these are different.

Sub-Templates in the PeopleSoft application

Sub-Template Content Library

Navigation: Reporting Tools > BI Publisher > Maintain BIP Sub-Templates

Upload your sub-templates to the PeopleTools BI Publisher Content Library.

[< Content Library](#)

Content Library

New Window

Content Library

Sub-Template ID PT3_STD_1

Sub-Template Properties

DescriptionStd Header

*LanguageEnglish

*Report Category IDALLUSERAll PeopleSoft User

Owner IDPeopleTools

Sub-Template TypeRTF

Sub-Template File

1 of 1

View All

Effective Date01/01/2022

*StatusActive

Template Filept3_stdhdr.rtf

UploadDownloadView

Add

Update/Display

Include History

Correct History

Return to Search

Previous in List

Next in List

Save

Take note of the sub-template status is Active and the effective date is correct.

The Sub-Template ID field at the top of the page is the value for the import statement for any template using this sub-template.

BI Pub Report Definition

Before uploading your main RTF template to PeopleTools, remember to change the import statements to point to PeopleTools BI Publisher Content Library for the sub-template rather than a location on the local machines drives.

Change `<?import:file:C:///psreports/pt3_stdhdr.rtf?>`

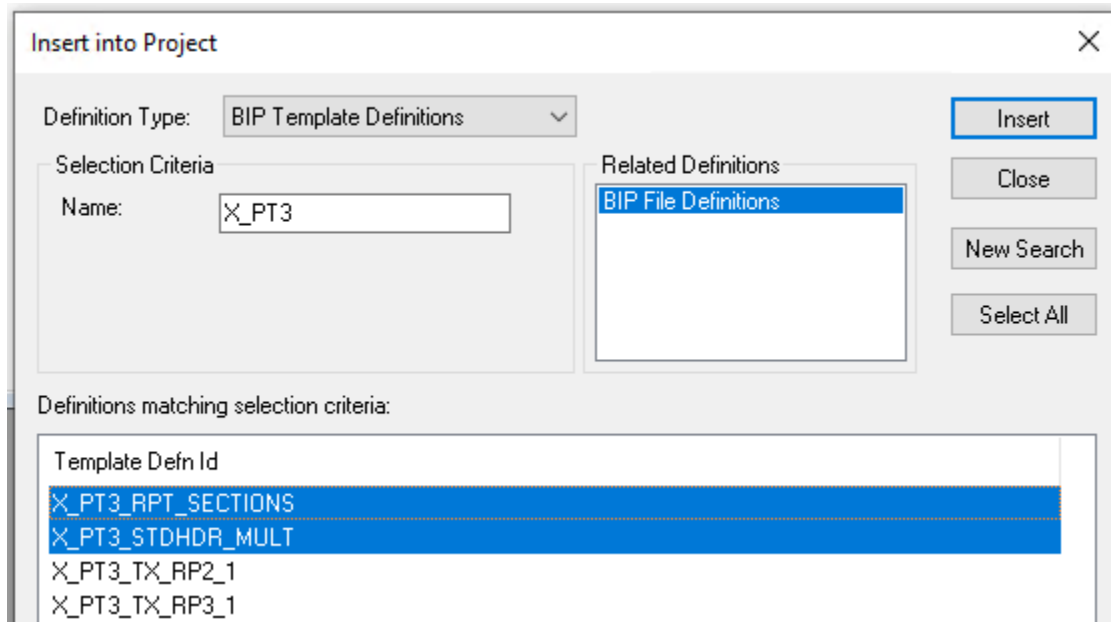
To `<?import:psxmlp://PT3_STDHDR?>`

The target of the call [PT3_STDHDR] is the key created for the sub-template in the BI Publisher Content Library in the step above.

After all import statements have been converted, define and upload the report template as a normal RTF template.

PeopleTools Project: Adding BI Pub Sub-Template Definitions

Add your sub-templates to your PeopleTools project using the “BIP Template Definitions” Definition Type. The sub-templates will appear in the results list along with any regular templates that match your criteria.



Make sure to include the “BIP File Definitions” in the “Related Definitions” box. You need the sub-template definition and the uploaded file.

Appendix A

As of version 12.2.1.4.0 of Oracle BI Publisher Desktop, there is a bug in the 64-bit version causing an error when testing sub-templates (Oracle Support Doc ID 2247221.1).

```
oracle.xdo.XDOException: java.util.EmptyStackException
```

As per Oracle Support, the fix for this is to edit the xdo.cfg file on your local machine and add the following line to the <properties> section:

```
<property name="xdk-secure-io-mode">false</property>
```

Here is the complete <properties> section from my local xdo.cfg file.

```
<!-- Properties -->
<properties>
  <!-- System level properties -->

  <!-- PLEASE SELECT A VALID TEMPFILE DIRECTORY!!! -->
  <property name="system-temp-dir">c:/Temp</property>

  <!-- PDF compression -->
  <property name="pdf-compression">true</property>

  <!-- PDF security -->
  <property name="pdf-security">false</property>
  <property name="pdf-open-password">user</property>
  <property name="pdf-permissions-password">owner</property>
  <property name="pdf-no-printing">true</property>
  <property name="pdf-no-changing-the-
document">true</property>

  <!-- check Boxes -->
  <property name="rtf-checkbox-
glyph">Wingdings;254;168</property>

  <!-- sub-template build and test -->
  <property name="xdk-secure-io-mode">false</property>

</properties>
```

This is the location of the xdo.cfg file on my testing machine:

```
C:\Program Files\Oracle\BI Publisher\BI Publisher Desktop\Template Builder for
Word\config\xdo.cfg
```