



Creating an XML File with PeopleTools using the File Layouts

PT 8.50+

Randall Groncki

Introduction

Several tools in PeopleSoft use and XML file as a data source. Prominent among these is the BI Publisher reporting tool. We can use PeopleTools to quickly generate the data, enrich and shape it before sending it on to the receiving utility.

This document will discuss the File Layout method of creating an XML File using PeopleTools and PeopleCode.

The process in brief:

1. Create a File Layout object using records
2. Create and load a rowset
3. Enrich the data in that rowset
4. Copy the Rowset to the File Layout object
5. Write the file

For demonstration purposes, our premise is creating an XML file for a BI Publisher report. However, there are other situations requiring XML Files.

XML File Creation

Since PeopleTools version 8.50, BI Publisher reports can only use XML Files, Queries and Connected Queries as data sources for reports. Prior versions of the tool also included options for XMLDoc & Rowsets Objects as data sources. Though no new data sources may be defined as type Rowset or XMLDoc, backward compatibility allows BI Publisher reports created in earlier tools versions using these objects to continue functioning.

Since all versions use XML Files as a data source, this demonstration will focus on generating XML Files using the Rowset Object

Premise for all XML File Creation Examples.

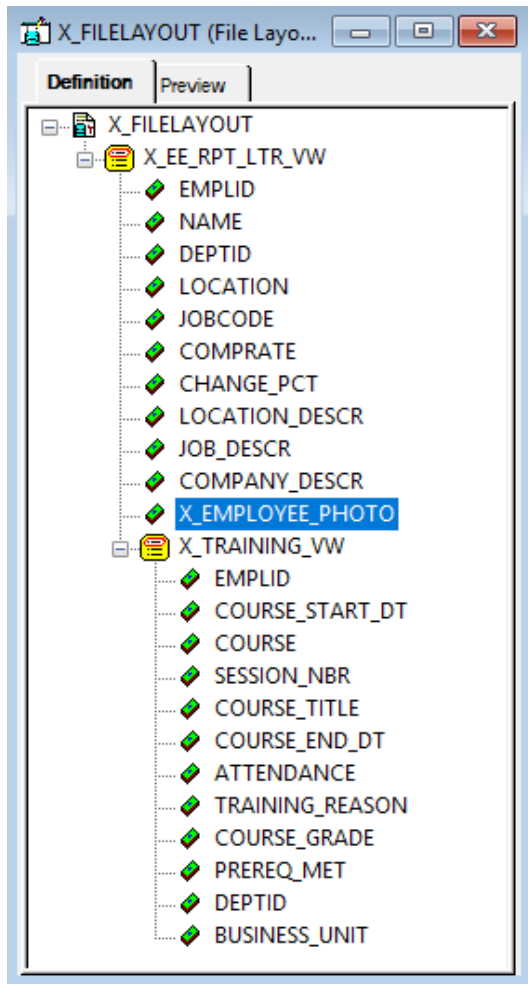
In order to demonstrate the differences multiple XML File creation techniques, all demonstrations will use the same report premise: A letter to all employees summarizing their current job and showing all training taken through the company. This is a complex data set with a parent/child relationship. Each employee will get a separate page in the report. For each employee, there will be zero or more training entries. See Appendix A for the class populating this data structure. The class returns this Rowset object:

Root
Employee Record: X_EE_RPT_LTR_VW (Parent Record)
• EMPLID
• NAME
• DEPTID
• LOCATION
• JOBCODE
• COMPRATE
• CHANGE_PCT
• LOCATION_DESCR
• JOB_DESCR
• COMPANY_DESCR
• X_EMPLOYEE_PHOTO
Training Record: X_TRAINING_VW (Child Record)
• EMPLID
• COURSE_START_DT
• COURSE
• SESSION_NBR
• COURSE_TITLE
• COURSE_END_DT
• ATTENDANCE
• TRAINING_REASON
• COURSE_GRADE
• PREREQ_MET
• DEPTID
• BUSINESS_UNIT

*Note: Per MOS Doc ID: Doc ID 962712.1:

PSQuery and rowset data sources may not be the most efficient way to generate the XML file that is used as input to BI Publisher, (XML Publisher). Consider using SQR or other mechanisms to generate the XML file

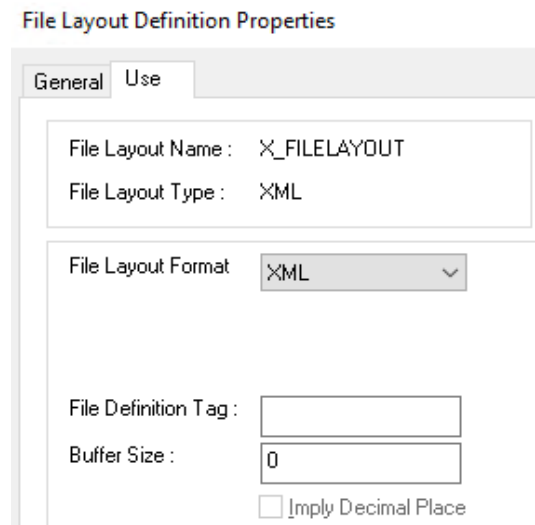
File Layout method of creating an XML File



PeopleSoft's File Layout Object allows us to quickly create XML files. Given the right conditions, this is the fastest method to create an XML File.

Create a record structure reflecting your final XML file structure, then, add those records accordingly to the file layout object. Remember to correctly record the Parent/Child relationships of the XML file on the file layout.

On the "Use" tab of the Properties box, define the file layout type as "XML".



The "File Definition Tag" field allows you to define the high-level node name of the XML file. Leaving this blank will result in a high-level node name of "<start>".

The File Layout object is assigned to the file during after the GetFile statement. Then, the file object inherits an internal Rowset Object reachable through PeopleCode.

In the example below, the data Rowset Object is copied directly into the File Layout Rowset Object since they are intentionally identical. Then, the Rowset can be dumped to file with the "WriteRowset" method of the File Object. Since the File Layout Type is defined as XML, PeopleSoft creates an XML File.

See Appendix B for the XML File created by this code.

```
import X_BI_PUB_PCODE:LoadTestData;

Local X_BI_PUB_PCODE:LoadTestData &LoadTestData = create
X_BI_PUB_PCODE:LoadTestData();
Local File &XML_File;
Local Rowset &RS_Employee, &FILEROWSET;
Local string &XML_Filename_path, &Str_MyXml;
Local Field &Fld_DescrLong = GetRecord().DESCRLONG;

&RS_Employee = &LoadTestData.LoadTestDataSet();

/* output files */

&XML_File = GetFile("FILELAYOUT.xml", "W", "UTF8");
&XML_File.SetFileLayout(FileLayout.X_FILELAYOUT);

&FILEROWSET = &XML_File.CreateRowset();

&RS_Employee.CopyTo(&FILEROWSET, Record.X_EE_RPT_LTR_VW, Record.X_EE_RPT_LTR_VW,
Record.X_TRAINING_VW, Record.X_TRAINING_VW);

&XML_File.WriteRowset(&FILEROWSET, True);
&XML_Filename_path = &XML_File.Name;
&XML_File.Close();

&XML_File = GetFile(&XML_Filename_path, "R", %FilePath_Absolute);
&Str_MyXml = &XML_File.GetString();
&XML_File.Close();
```

XML File Generation Considerations

Concurrency: Generating multiple versions of the same report simultaneously

PeopleSoft by its very nature may have multiple, different users attempting to create the same report at the same time. Ensure that your code accounts for this probability by making the file names unique per each instance of the report. One idea to accomplish this is to append the User or Employee ID to the file creating a unique file name.

Another method is to append the current date/time stamp to the filename to force uniqueness.

Schema Files

For PeopleTools version 8.48 & 8.49, BI Publisher requires XML Schema files for the correct mapping of tags to a PDF Template. You could include these for RTF (MSWord) templates, but they were not necessary.

There are several online tools which generate an Schema (XSD) from a file or xml string.

Starting with PeopleTools 8.50, BI Publisher does not require a schema file for PDF Template mapping. Only a sample data file is required.

Sample Files for defining BI Publisher Data Sources.

You must provide a Sample XML Data file when creating a new BI Publisher report. A good idea is to execute a version of the XML File generation code using a sample or representative employee. Another good idea is to, for the sample run, modify your code to put the XML Tag names as values in the XML File. This will aid in mapping and formatting: do you have the right field in the right place on your report?

House Keeping

Depending on data policies of the implementations site, it may be a good idea to delete the BI Publisher Source files from the files directory after the report documents have been created. The easiest way to delete the file is to re-open the file after the BI Publisher code is complete, then use the "delete()" method instead of the "close()" method. This will remove the file from the servers.

Appendix A

Data set class for data source creation. This function populates and returns a complex RowSet object.

```

/*****
/** GronkWare, Inc
/** Randy Groncki 20168-04-16
/** BI Publisher
/** XML File Generation Examples
/** Contact: Randy.Groncki@GronkWare.com
*****/

class LoadTestData
    method LoadTestData();
    method LoadTestDataSet() Returns Rowset;

private

    method LoadEmployeeImage(&Emplid As string) Returns string;

end-class;

method LoadTestData

end-method;

method LoadTestDataSet
    /+ Returns Rowset +/

    Local Rowset &RS_Training, &RS_Employees;
    Local Record &JOB_REC, &LOCATION_TBL_REC, &COMPANY_TBL_REC, &JOB_CODE_TBL_REC;
    Local integer &i;

    /* create records */
    &JOB_REC = CreateRecord(Record.JOB);
    &LOCATION_TBL_REC = CreateRecord(Record.LOCATION_TBL);
    &COMPANY_TBL_REC = CreateRecord(Record.COMPANY_TBL);
    &JOB_CODE_TBL_REC = CreateRecord(Record.JOB_CODE_TBL);

    /* create rowsets */
    &RS_Training = CreateRowset(Record.X_TRAINING_VW); /* child rowset */
    &RS_Employees = CreateRowset(Record.X_EE_RPT_LTR_VW, &RS_Training); /* parent
rowset */

    /* Fill Parent */
    &RS_Employees.Fill("where emplid like 'KU00%' and exists (select 'x' from
ps_training t where t.emplid = fill.emplid)");

    /* Loop through parent rowset for processing on each row */
    For &i = 1 To &RS_Employees.ActiveRowCount

        /* Fill child rowset */
        &RS_Training = &RS_Employees(&i).GetRowset(Scroll.X_TRAINING_VW);
        &RS_Training.Fill("where emplid = :1",
&RS_Employees(&i).X_EE_RPT_LTR_VW.EMPLID.Value);

        /* Get job row for linking other data */
        &JOB_REC.EMPLID.Value = &RS_Employees(&i).X_EE_RPT_LTR_VW.EMPLID.Value;
        &JOB_REC.EMPL_RCD.Value = 0;
        /* get the current effdt & effseq for the EEs job row */
        SQLExec("select %dateout(j.effdt), j.effseq from ps_job j where j.emplid = :1
and j.empl_rcd = :2 and j.effdt = (select max(j2.effdt) from ps_job j2 where
j2.emplid = j.emplid and j2.empl_rcd = j.empl_rcd and j2.effdt <= %datein(:3)) and
j.effseq = (select max(j3.effseq) from ps_job j3 where j3.emplid = j.emplid and
j3.empl_rcd = j.empl_rcd and j3.effdt = j.effdt)", &JOB_REC.EMPLID.Value,
&JOB_REC.EMPL_RCD.Value, %Date, &JOB_REC.EFFDT.Value, &JOB_REC.EFFSEQ.Value);
        &JOB_REC.SelectByKey();
    
```

Data set function for data source creation (Continued)

```
    /* retrieve specific location data for description in report */
    &LOCATION_TBL_REC.SETID.Value = &JOB_REC.SETID_LOCATION.Value;
    &LOCATION_TBL_REC.LOCATION.Value = &JOB_REC.LOCATION.Value;
    &LOCATION_TBL_REC.SelectByKeyEffDt(%Date);
    &RS_Employees(&i).X_EE_RPT_LTR_VW.LOCATION_DESCR.Value =
&LOCATION_TBL_REC.DESCR.Value;

    /* retrieve specific company data for description in report */
    &COMPANY_TBL_REC.COMPANY.Value = &JOB_REC.COMPANY.Value;
    &COMPANY_TBL_REC.SelectByKeyEffDt(%Date);
    &RS_Employees(&i).X_EE_RPT_LTR_VW.COMPANY_DESCR.Value =
&COMPANY_TBL_REC.DESCR.Value;

    /* retrieve specific jobcode data for description in report */
    &JOBCODE_TBL_REC.SETID.Value = &JOB_REC.SETID_JOBCODE.Value;
    &JOBCODE_TBL_REC.JOBCODE.Value = &JOB_REC.JOBCODE.Value;
    &JOBCODE_TBL_REC.SelectByKeyEffDt(%Date);
    &RS_Employees(&i).X_EE_RPT_LTR_VW.JOB_DESCR.Value = &JOBCODE_TBL_REC.DESCR.Value;

    /* get employee image */
    &RS_Employees(&i).X_EE_RPT_LTR_VW.X_EMPLOYEE_PHOTO.Value =
%This.LoadEmployeeImage(&JOB_REC.EMPLID.Value);

End-For;

Return &RS_Employees;
end-method;

method LoadEmployeeImage
    /* &Emplid as String */
    /* Returns String */

    Local File &Image_File;
    Local string &Base64String, &NewFileName, &FQ_Filename_path;
    Local integer &retcode;

    &NewFileName = %UserId | %Datetime | ".jpg";
    &Image_File = GetFile(&NewFileName, "W");
    &FQ_Filename_path = &Image_File.Name;
    &Image_File.Close();

    &retcode = GetAttachment("record://X_EPHOTO_VW", &Emplid, &FQ_Filename_path);

    If &retcode < 2 Then
        &Image_File = GetFile(&FQ_Filename_path, "R", %FilePath_Absolute);
        &Base64String = &Image_File.GetBase64StringFromBinary();
        &Image_File.Close();
    End-If;

    /* delete file */
    &Image_File = GetFile(&FQ_Filename_path, "R", %FilePath_Absolute);
    &Image_File.Delete();

    Return &Base64String;
end-method;
```

Appendix B

File Layout XML File Example

```
<?xml version='1.0'?>
<Employee_Training_Letter>
<X_EE_RPT_LTR_VW>
<EMPLID>KU0007</EMPLID>
<NAME>Betty Locherty</NAME>
<DEPTID>13000</DEPTID>
<LOCATION>KUDE00</LOCATION>
<JOBCODE>420050</JOBCODE>
<COMPRATE>5769.230769</COMPRATE>
<CHANGE_PCT>0.000</CHANGE_PCT>
<LOCATION_DESCR>Delaware Operations</LOCATION_DESCR>
<JOB_DESCR>Finance Director</JOB_DESCR>
<COMPANY_DESCR>Global Business Institute</COMPANY_DESCR>
<X_EMPLOYEE_PHOTO> </X_EMPLOYEE_PHOTO>
<X_TRAINING_VW>
<EMPLID>KU0007</EMPLID>
<COURSE_START_DT>05/05/2002</COURSE_START_DT>
<COURSE> </COURSE>
<SESSION_NBR> </SESSION_NBR>
<TITLE>Organizational behaviour</TITLE>
<COURSE_END_DT>05/05/2002</COURSE_END_DT>
<ATTENDANCE>C</ATTENDANCE>
<TRAINING_REASON> </TRAINING_REASON>
<COURSE_GRADE> </COURSE_GRADE>
<PREREQ_MET>N</PREREQ_MET>
<DEPTID> </DEPTID>
<BUSINESS_UNIT> </BUSINESS_UNIT>
</X_TRAINING_VW>
<X_TRAINING_VW>
<EMPLID>KU0007</EMPLID>
<COURSE_START_DT>10/22/2010</COURSE_START_DT>
<COURSE>K001</COURSE>
<SESSION_NBR>0018</SESSION_NBR>
<TITLE>Time Management</TITLE>
<COURSE_END_DT>10/22/2010</COURSE_END_DT>
<ATTENDANCE>E</ATTENDANCE>
<TRAINING_REASON> </TRAINING_REASON>
<COURSE_GRADE> </COURSE_GRADE>
<PREREQ_MET>N</PREREQ_MET>
<DEPTID>13000</DEPTID>
<BUSINESS_UNIT>US005</BUSINESS_UNIT>
</X_TRAINING_VW>
</X_EE_RPT_LTR_VW>
</Employee_Training_Letter>
```