Sending Images to BI Publisher using PeopleTools

PT 8.59
Randall Groncki

Introduction

Using PeopleTools, we can generate BI Publisher reports containing images from our PeopleSoft environment. For example, an employee report listing with the employee's photo or an asset report containing the item pictures.

To accomplish this, we need to use several technologies previously covered:

- Create an XML File with PeopleTools using the Rowset Method
- Invoke BI Publisher with PeopleCode
- Using Design Time Images in PeopleTools
- User Uploaded Images in PeopleTools
- File Image Utilities PeopleSoft forgot

This is not something we can accomplish using a PSQuery data source and the generic BI Publisher App Engine. The above subjects are outside the scope of this document. See the included links for detail instructions, downloadable objects and code for each of these subjects

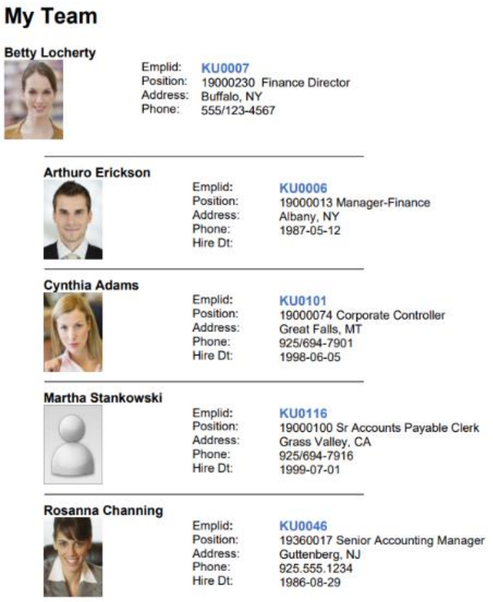
Demo Example

Our example is a Manager's Team Member report. This report (Example right) shows the basic information about a manager's direct reports:

- Name
- Job Info
- Basic Contact Info
- Photo

We pull each photo from the database as we create the report.

In the case an employee does not have a photo, a "Dummy" silhouette is displayed for UI Consistency.



My Team

Betty Locherty
Emplid: KU0007
Position: 19000230 Finance Director
Address: Buffalo, NY
Phone: 555/123-4567

Arthuro Erickson
Emplid: KU0006
Position: 19000013 Manager-Finance
Address: Albany, NY
Phone: 1987-05-12
Hire Dt:

Cynthia Adams
Emplid: KU0101
Position: 19000074 Corporate Controller
Address: Great Falls, MT
Phone: 925/694-7901
Hire Dt: 1998-06-05

Martha Stankowski
Emplid: KU0116
Position: 19000100 Sr Accounts Payable Clerk
Address: Grass Valley, CA
Phone: 925/694-7916
Hire Dt: 1999-07-01

Rosanna Channing
Emplid: KU0046
Position: 19360017 Senior Accounting Manager
Address: Guttenberg, NJ
Phone: 925.555.1234
Hire Dt: 1986-08-29

Approach Overview: How are we going to do this?

PeopleTools

- Create views of our data for easier data capture
- Create a Parent/Child Rowset using these view Record Definitions to hold the data
- Send each employee photo to the file server and read it back using the File Object’s Base64 method
- Place that photo’s Base64 string into a Long Field on the view records
- Use the Rowset Method to create an XML String
- Send that XML String to a file
- Use BI Publisher Delivered App Packages to generate the report

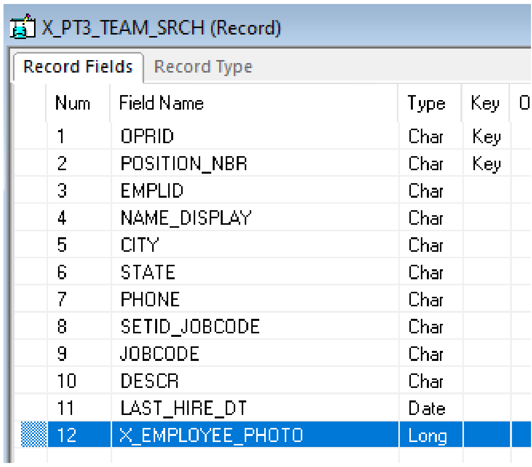
BI Publisher Template

- Use an RTF template for this report
- Use the fo:instream-foreign-object function to transform the Base64 string into an image
- Use conditional areas to deal with missing photos

Record Definitions

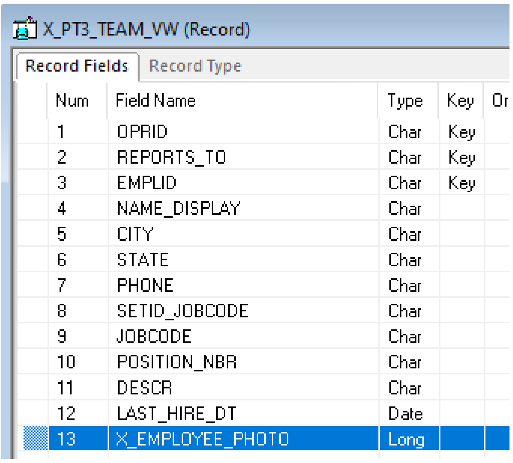
We need to send an extremely long string containing the Base64 encoded photo from PeopleTools to BI Publisher.

X_EMPLOYEE_PHOTO is a type LONG field added to both the parent and child records of the data structure. This is the field that will hold the encoded photo of each employee.



X_PT3_TEAM_SRCH (Record)

Num	Field Name	Type	Key	O
1	OPRID	Char	Key	
2	POSITION_NBR	Char	Key	
3	EMPLID	Char		
4	NAME_DISPLAY	Char		
5	CITY	Char		
6	STATE	Char		
7	PHONE	Char		
8	SETID_JOBCODE	Char		
9	JOBCODE	Char		
10	DESCR	Char		
11	LAST_HIRE_DT	Date		
12	X_EMPLOYEE_PHOTO	Long		



X_PT3_TEAM_VW (Record)

Num	Field Name	Type	Key	Or
1	OPRID	Char	Key	
2	REPORTS_TO	Char	Key	
3	EMPLID	Char	Key	
4	NAME_DISPLAY	Char		
5	CITY	Char		
6	STATE	Char		
7	PHONE	Char		
8	SETID_JOBCODE	Char		
9	JOBCODE	Char		
10	POSITION_NBR	Char		
11	DESCR	Char		
12	LAST_HIRE_DT	Date		
13	X_EMPLOYEE_PHOTO	Long		

Structure

X_PT3_TEAM_SRCH

- Parent Record
- Contains the Manager’s data and photo
- One record per report

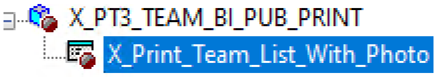
X_PT3_TEAM_VW

- Child Record
- Team Member reporting to Manager in Parent Record
- One row per each employee reporting to manager

```
   /* create rowsets */
   &RS_X_PT3_TEAM_VW = CreateRowset(Record.X_PT3_TEAM_VW); /* child rowset */
   &RS_X_PT3_TEAM_SRCH = CreateRowset(Record.X_PT3_TEAM_SRCH, &RS_X_PT3_TEAM_VW);
/* parent rowset */
```

Data Load App Package

I've created a custom App Package/App Class to contain all the code to load the data for our demonstration.



X_PT3_TEAM_BI_PUB_PRINT

X_Print_Team_List_With_Photo

The X_Print_Team_List_With_Photo() class contains all the methods and properties referenced in this doc. All the objects used in this demonstration are available on PeopleToolsTechTips.com and github.com/PeopleToolsTechTips

The core of this App Package is the LoadEmployeeImage() method. This method:

- Uses a view to convert the EMPL_PHOTO table to an attachment record
- Uses the GetAttachment() function to move the image to a file
- Opens that file and uses the file object's GetBase64StringFromBinary() function to read it back as a Base64 string
- Returns the Base64 string for load into the X_EMPLOYEE_PHOTO field on each record

```
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();

   <* this makes the xml file unnecessarily large with a repeated default image
   If None(&Base64String) Then
      &Base64String = %This.LoadDummyImage();
   End-If;                           *>

   Return &Base64String;
end-method;
```

Create the XML File

After the Rowset is loaded with the manager header and all the reports in the child Rowset, convert that rowset to an XML string using the delivered PSXP_XMLGEN:RowSetDS class.

Write the resulting string to a file.

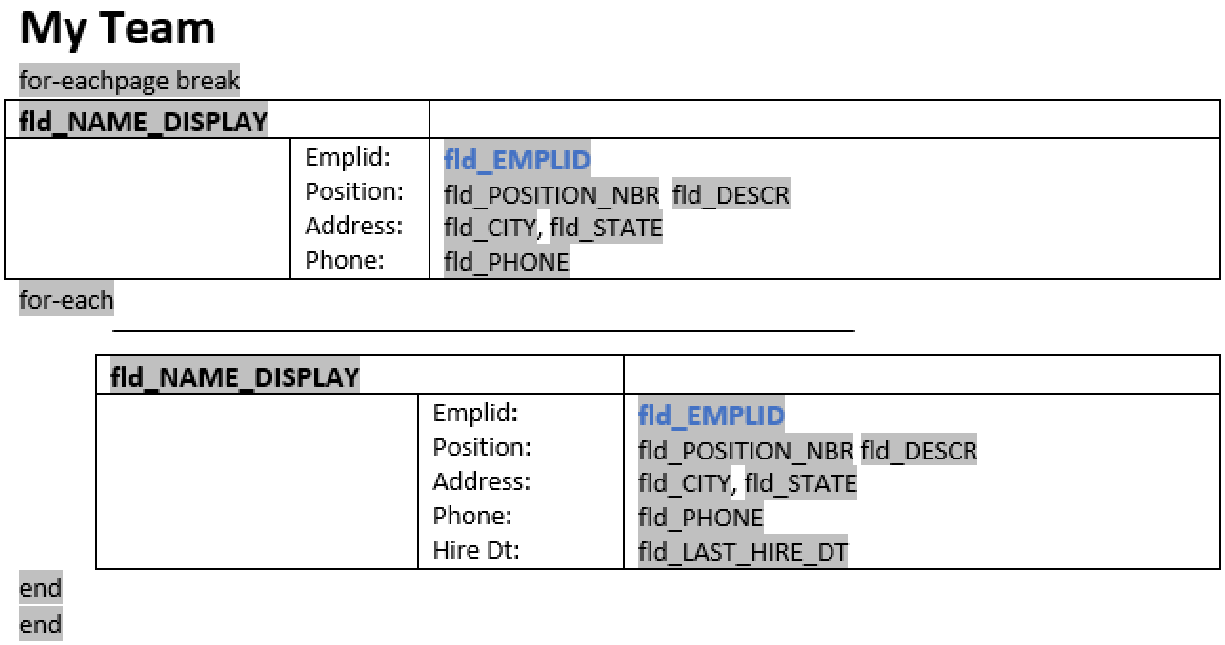
```
&oXML_GENERATOR = create PSXP_XMLGEN:RowSetDS();
&my_xml = &oXML_GENERATOR.getXMLData(&RS_Team_List, "");

&Str_Filename = "Team_Listing_" | %UserId | ".xml";
&oXML_File = GetFile(&Str_Filename, "W", "UTF8");
&oXML_File.WriteLine(&my_xml);

/* save file name and path for publishing */
&XML_Filename_path = &oXML_File.Name;
&oXML_File.Close();
```

BI Publisher Template

Use an RTF template for this report.



My Team

for-eachpage break

fld_NAME_DISPLAY		
	Emplid: Position: Address: Phone:	**fld_EMPLID** fld_POSITION_NBR fld_DESCR fld_CITY, fld_STATE fld_PHONE

for-each

fld_NAME_DISPLAY		
	Emplid: Position: Address: Phone: Hire Dt:	**fld_EMPLID** fld_POSITION_NBR fld_DESCR fld_CITY, fld_STATE fld_PHONE fld_LAST_HIRE_DT

end

end

The images will be placed in the 1st box on of the grid.

fo:instream-foreign-object

The fo:instream-foreign-object() function enables us to insert an Base64 encoded image from our XML File into our BI Publisher RTF Template.

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

Image Type Options

- content-type="image/jpg"
- content-type="image/png"
- content-type="image/gif"

Image Size Options

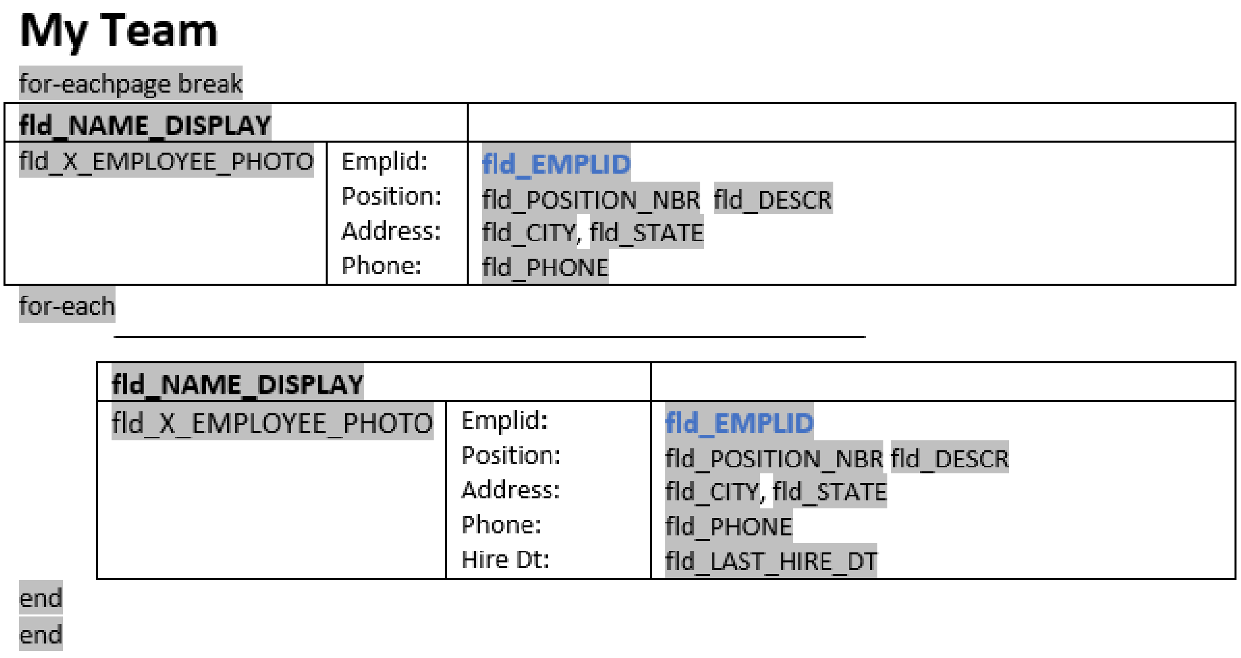
Height and Width parameters are optional. If not specified, the image will render at its natural size on your document.

Other sizing options include:

- px – pixels
- cm – centimeters
- % - percentage of original dimensions

Image Field designation

Insert the field containing the Base64 image into the report.



My Team

for-eachpage break

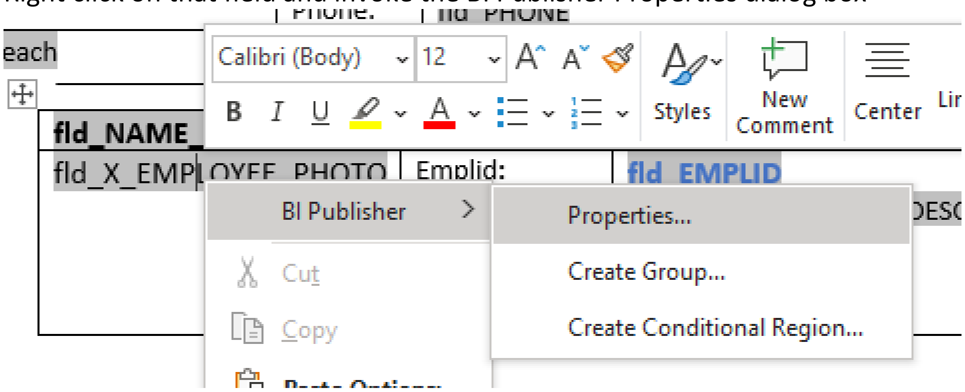
fld_NAME_DISPLAY		
fld_X_EMPLOYEE_PHOTO	Emplid: Position: Address: Phone:	**fld_EMPLID** fld_POSITION_NBR fld_DESCR fld_CITY, fld_STATE fld_PHONE

for-each

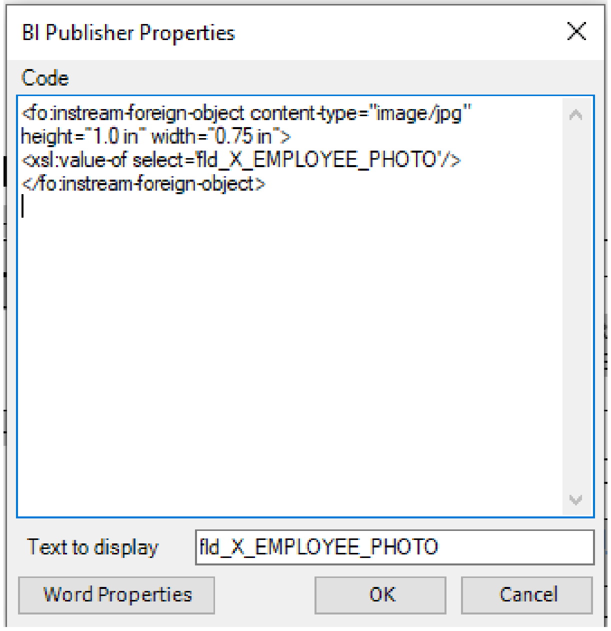
fld_NAME_DISPLAY		
fld_X_EMPLOYEE_PHOTO	Emplid: Position: Address: Phone: Hire Dt:	**fld_EMPLID** fld_POSITION_NBR fld_DESCR fld_CITY, fld_STATE fld_PHONE fld_LAST_HIRE_DT

end
end

Right click on that field and invoke the BI Publisher Properties dialog box



Paste in the fo:instream-foreign-object function and edit to specifications



```
<fo:instream-foreign-object content-type="image/jpg"
height="1.0 in" width="0.75 in">
<xsl:value-of select='fld_X_EMPLOYEE_PHOTO'/>
</fo:instream-foreign-object>
```

Text to display: fld_X_EMPLOYEE_PHOTO

Edit the fo:instream-foreign-object syntax

```
<fo:instream-foreign-object content-type="image/jpg" height=”1.0 in” width=”0.75
in”>
<xsl:value-of select='fld_X_EMPLOYEE_PHOTO'/>
</fo:instream-foreign-object>
```

Ensure the Select is pointed at the field containing the Base64 encoded image and the sizing information is correct.

Missing Photos

Not all employees will have a photo. Missing photos create a problem in a BI Publisher document:

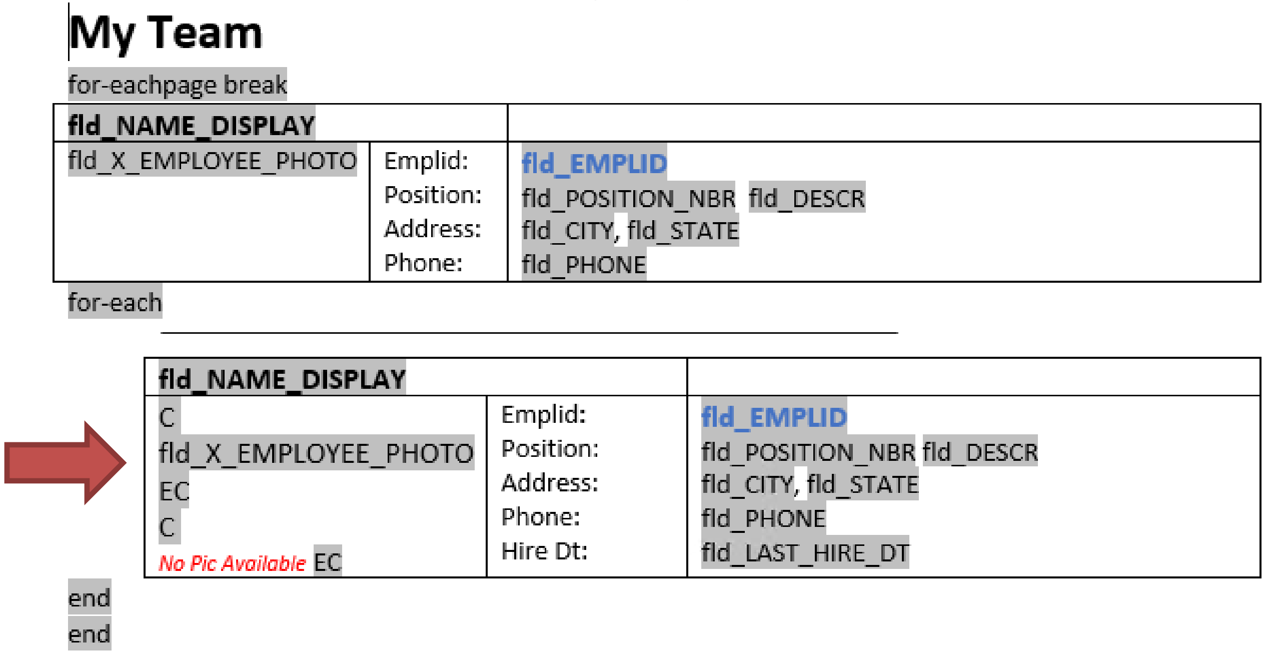
- The user is visually expecting a photo in the UX
- Missing photos are sized overly large and badly in BI Publisher.

There are several ways to handle missing photos:

- When generating the XML File, insert a generic photo in place of the missing photo.
  - Easiest option to deal with in the RTF Template
  - Creates an unnecessarily large XML file with the default image defined multiple times
  - Taxes system resources more heavily than necessary
- RTF Template Conditional Area – No Photo
  - Create a conditional area to only show the photo if populated
- RTF Template Conditional Area – Default Photo
  - Create a conditional area to show an alternate fixed default photo if no photo is populated in the data

RTF Template – No Photo

Create a conditional area that tests for a string in the photo field



My Team

for-eachpage break

fld_NAME_DISPLAY		
fld_X_EMPLOYEE_PHOTO	Emplid: Position: Address: Phone:	**fld_EMPLID** fld_POSITION_NBR fld_DESCR fld_CITY, fld_STATE fld_PHONE

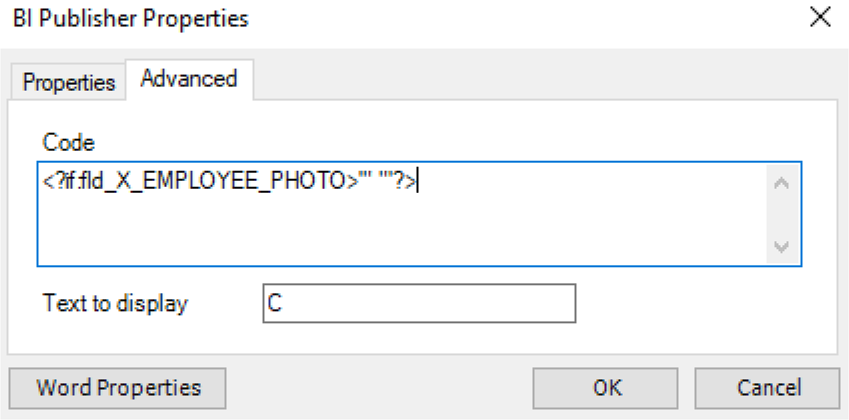
for-each

fld_NAME_DISPLAY		
C fld_X_EMPLOYEE_PHOTO EC C *No Pic Available* EC	Emplid: Position: Address: Phone: Hire Dt:	**fld_EMPLID** fld_POSITION_NBR fld_DESCR fld_CITY, fld_STATE fld_PHONE fld_LAST_HIRE_DT

end

end

The XSL in the condition tag looks checks to see if the field is greater than a space. If so, show the photo.

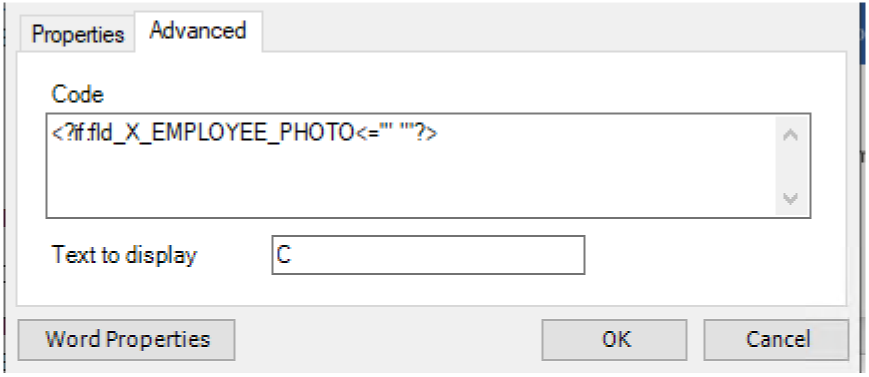


Code

```
<?if:fld_X_EMPLOYEE_PHOTO>"' '"?>
```

Text to display: C

The next conditional area checks if the field is less than or equal to a space. If so, show the “No Photo Available” verbiage.



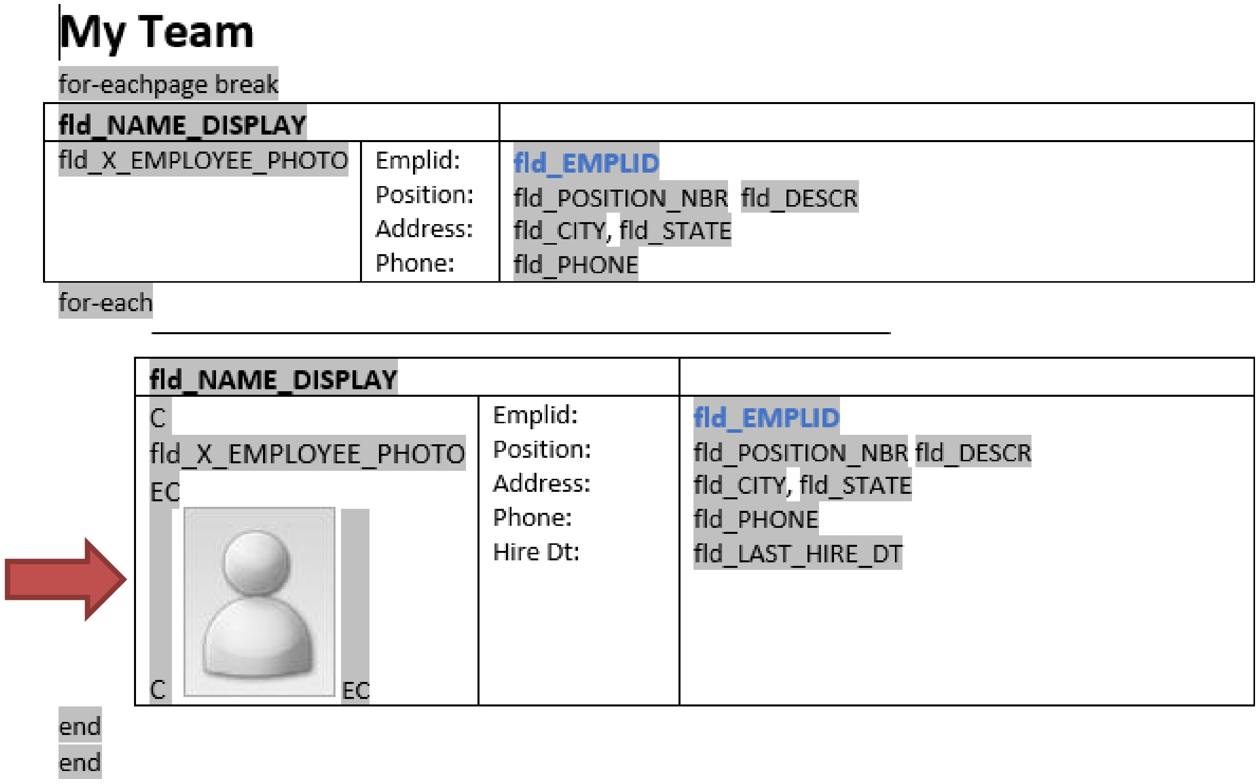
Code

```
<?if:fld_X_EMPLOYEE_PHOTO<="' '"?>
```

Text to display: C

RTF Template – Default Image

The RTF template showing the default image is much the same as the template showing the “No Photo Available” string. Instead of the text, a default image is inserted into the template. This saves us from sending that same image to BI Publisher many times for all the cases where an image is unavailable.



My Team

for-eachpage break

fld_NAME_DISPLAY		
fld_X_EMPLOYEE_PHOTO	Emplid: Position: Address: Phone:	**fld_EMPLID** fld_POSITION_NBR fld_DESCR fld_CITY, fld_STATE fld_PHONE

for-each

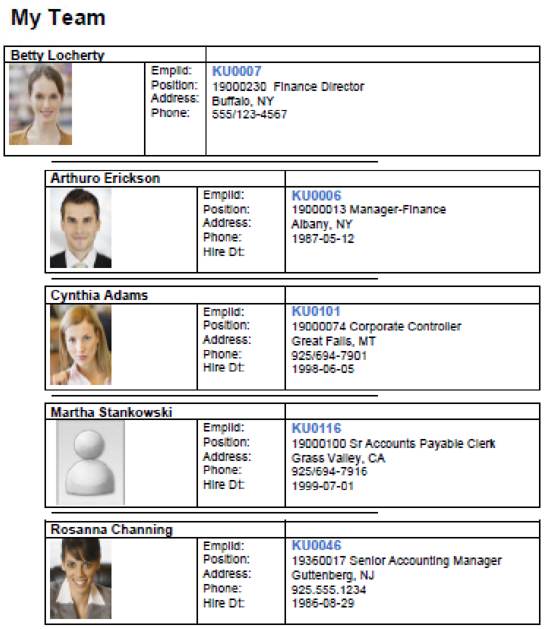
fld_NAME_DISPLAY		
C fld_X_EMPLOYEE_PHOTO EC C EC	Emplid: Position: Address: Phone: Hire Dt:	**fld_EMPLID** fld_POSITION_NBR fld_DESCR fld_CITY, fld_STATE fld_PHONE fld_LAST_HIRE_DT

end

end

Report Examples

Default Photo



My Team

Betty Locherty		
	Emplid: Position: Address: Phone:	KU0007 19000230 Finance Director Buffalo, NY 555/123-4567

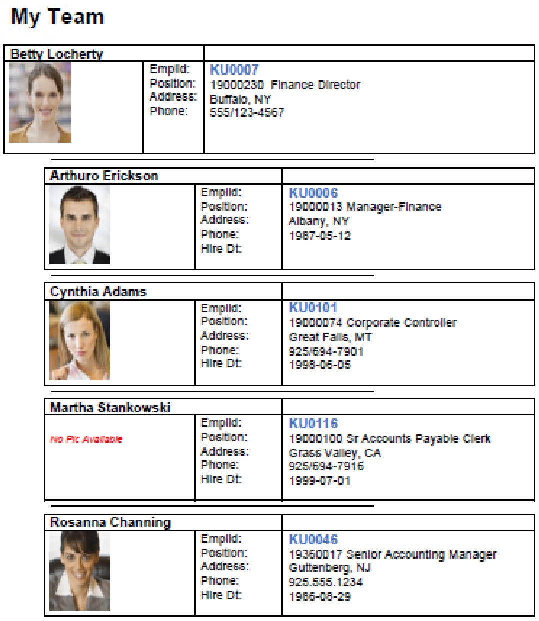
Arthuro Erickson		
	Emplid: Position: Address: Phone: Hire Dt:	KU0006 19000013 Manager-Finance Albany, NY 1987-05-12

Cynthia Adams		
	Emplid: Position: Address: Phone: Hire Dt:	KU0101 19000074 Corporate Controller Great Falls, MT 925/694-7901 1998-06-05

Martha Stankowski		
	Emplid: Position: Address: Phone: Hire Dt:	KU0116 19000100 Sr Accounts Payable Clerk Grass Valley, CA 925/694-7916 1999-07-01

Rosanna Channing		
	Emplid: Position: Address: Phone: Hire Dt:	KU0046 19360017 Senior Accounting Manager Guttenberg, NJ 925.555.1234 1986-08-29

Missing Photo Message



My Team

Betty Locherty		
	Emplid: Position: Address: Phone:	KU0007 19000230 Finance Director Buffalo, NY 555/123-4567

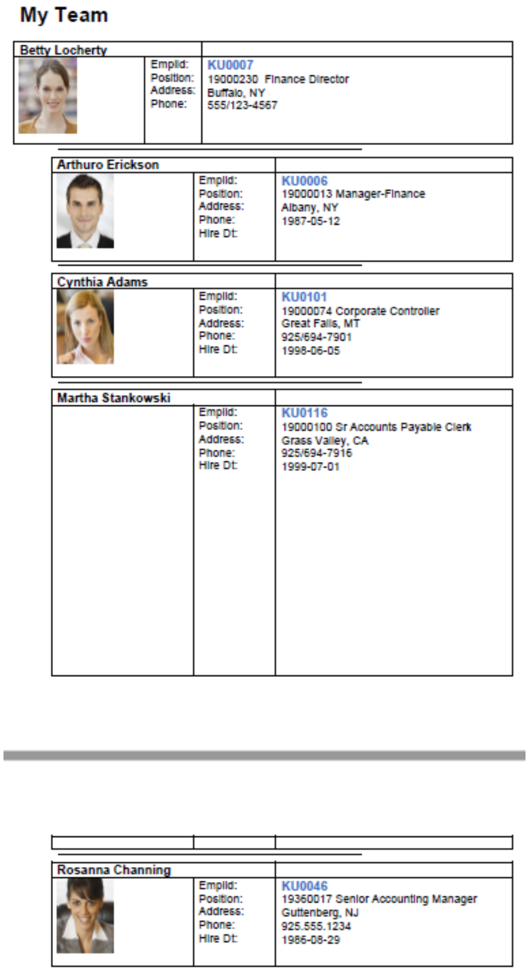
Arthuro Erickson		
	Emplid: Position: Address: Phone: Hire Dt:	KU0006 19000013 Manager-Finance Albany, NY 1987-05-12

Cynthia Adams		
	Emplid: Position: Address: Phone: Hire Dt:	KU0101 19000074 Corporate Controller Great Falls, MT 925/694-7901 1998-06-05

Martha Stankowski		
No Pic Available	Emplid: Position: Address: Phone: Hire Dt:	KU0116 19000100 Sr Accounts Payable Clerk Grass Valley, CA 925/694-7916 1999-07-01

Rosanna Channing		
	Emplid: Position: Address: Phone: Hire Dt:	KU0046 19360017 Senior Accounting Manager Guttenberg, NJ 925.555.1234 1986-08-29

Missing Photo Unmanaged



My Team

Betty Locherty		
	Emplid: Position: Address: Phone:	KU0007 19000230 Finance Director Buffalo, NY 555/123-4567

Arthuro Erickson		
	Emplid: Position: Address: Phone: Hire Dt:	KU0006 19000013 Manager-Finance Albany, NY 1987-05-12

Cynthia Adams		
	Emplid: Position: Address: Phone: Hire Dt:	KU0101 19000074 Corporate Controller Great Falls, MT 925/694-7901 1998-06-05

Martha Stankowski		
	Emplid: Position: Address: Phone: Hire Dt:	KU0116 19000100 Sr Accounts Payable Clerk Grass Valley, CA 925/694-7916 1999-07-01

Rosanna Channing		
	Emplid: Position: Address: Phone: Hire Dt:	KU0046 19360017 Senior Accounting Manager Guttenberg, NJ 925.555.1234 1986-08-29