AePlus

Reporting System

(My First PDF Report

Using PeopleCode)

SRI Technologies Pty Ltd
WebSite: www.sritech.biz

Email: sritech@sritech.biz

Table Of Contents

AePlus Reporting System

My First PDF Report using PeopleCode

In this tutorial, the purpose is to demonstrate what is involved when developer wants to develop a report using AePlus Reporting System. To begin with, before we get on to technical stuff, let us define the requirements that Report should address.

We have tried to set the requirements in a comprehensive way so that most features that the Reporting System offers get utilised. In a company we have employees who are paid thru a payroll system. Each employee belongs to a PAYGROUP and a DEPARTMENT in addition to the COMPANY they work for. Employee is paid certain amount against an Earning Code. They may have certain deduction against a Deduction Code. The **Input Data Stream** is a CSV file that contains multiple rows for an employee and has following fields:

- COMPANY, PAYGROUP, DEPARTMENT, EMPLID, EFFDATE, ERNDEDCODE, AMOUNT

  For the sake of simplicity, the input data stream is from the flat file and is sorted on COMPANY, PAYGROUP, DEPARTMENT and EMPLID. The input stream can be from the database table using While &SQL.Fetch(&Rec) loop.

Requirement:

The Requirement is to produce a report that:

1. Prints each employee as one line on the report consolidating following amounts as columns:
   a. Basic
   b. Allowances
   c. Deduction
   d. Tax
   e. Net (Calculated as Basic + Allowances + Deduction – Tax)
2. Prints Employee ID and Employee Name
3. On change of Company, start a new page.
4. Whenever there is a change in Company name, Pay Group or Department, a line should be printed:
   a. With their Code and Description before employee is printed (On-Break Before Event)
   b. Totals for that Code after employees have been printed. (On-Break After Event)
5. Finally grand totals should be printed for each column.
6. Further, if for an employee Total Tax is less than 700 or Total Net less than 1500 – do not include them in the report.

Let's get on with the steps that would be required to develop this report using AePlus Reporting System.

Pre-requisites:

The developer is expected to have followings:

1. Read/Write access to PeopleSoft App Designer and Data Mover.
2. Fair knowledge of PeopleSoft App Designer
3. Fair knowledge of Application Engine
4. Fair knowledge PeopleCode
5. Access to AePlus (App Engine Workbench and Reporting System Workbench)

Fields and Records Creation:

Report Name:

The very first thing is to give a name to this report that will be identified uniquely in the PeopleSoft database. Let us name it as AEP_MY_FRST_PDF. The length of this name must not be more than 15 characters as there will be technical objects created with this name in the database. You may choose to follow any naming conventions. We have started this with AEP – meaning AePlus Report.

Fields Creation:

We will be creating a **Derived Record** in PeopleSoft, but before we do so, let us create some fields that will go into this record. Most of these fields relate to fields in the **Input Data Stream** except few special fields that is used by AePlus Reporting System. Please create following fields and save them in a project called AEP_MY_FRST_PDF. We are creating all new fields so that there is no conflict with any existing fields in your database.

Field Name	Long Name	Short Name	Field Type	Size
AEP_COMPANY	Company	Company	Char	5
AEP_PAYGROUP	Pay Group	PayGroup	Char	3
AEP_DEPTID	Department	Department	Char	10
AEP_ER_DD_CODE	Earning/Ded Code	ErDed Code	Char	10
AEP_RPT_EVENTS	Aep Report Events	Rpt Events	Char	1
AEP_KEY_DETAIL	Key Detail	Key Detail	Char	30
AEP_AMOUNT	Amount	Amount	Number	10.2

Please note that you would not have to create field AEP_RPT_EVENTS as this would have been created as part of Reporting System installation.

Records Creation:

Now is the time to create AEP_MY_FRST_PDF record with Record Type as Derived/Work. Please add following field to the record. The fields must be in the order specified. In AePlus terminology, we will refer this record as ***rpt_aet_rec***. This record is used in controlling On-Break logic and holds all the Report Events PeopleCode Functions that developer maintains.

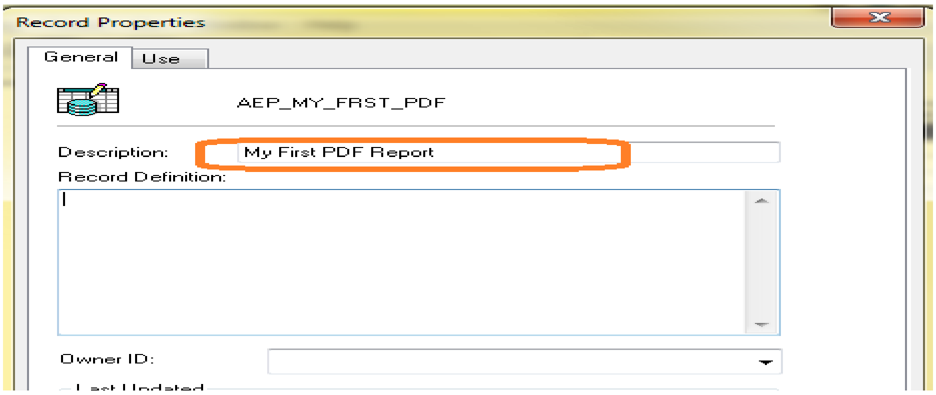
Field Name
AEP_COMPANY
AEP_PAYGROUP
AEP_DEPTID
AEP_RPT_EVENTS

When we submit a row of data that is received from the **Input Data Stream** into AePlus Reporting System, it is submitted as a Record. In AePlus terminology, we refer this as ***rpt_dat_rec***. In this example, since data is coming from CSV file, we will create ***rpt_dat_rec*** record that will hold the values of various fields read from CSV file. Please create AEP_MY_FRST_DAT record with record type as Derived/Work adding following fields.

Field Name
AEP_COMPANY
AEP_PAYGROUP
AEP_DEPTID
AEP_KEY_DETAIL
AEP_ER_DD_CODE
AEP_AMOUNT

Please note that ***rpt_dat_rec*** Record must have all the key fields that are in ***rpt_aet_rec****(i.e. AEP_COMPANY, AEP_PAYGROUP, AEP_DEPTID).* AEP_KEY_DETAIL is a special field that will be used by the consolidation process when we receive multiple rows for an employee from the **Input Data Stream**.

Finally, give your report name a description that you would like to print on the pdf report.



AEP_MY_FRST_PDF

Description: My First PDF Report

Record Definition:

Owner ID:

Last Updated

That's all about fields and records creations and we can talk about PeopleCode now.

PeopleCode Creation:

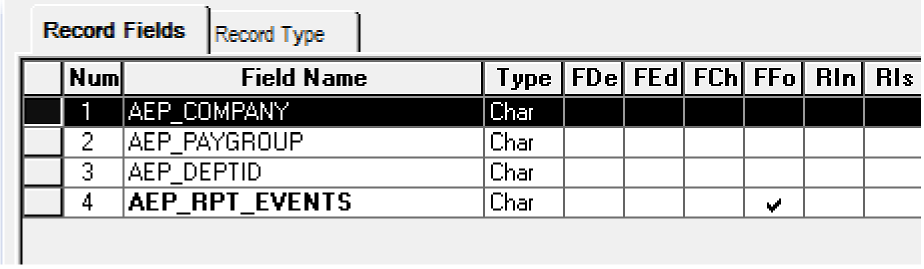
Before we proceed further, it is important to note here that our current report would have following Report Events Functions:

Report Events:

Report Events:	rpt_begin(…)
	rpt_end(…)
Page Events:	rpt_page_header(…)
	rpt_page_footer(…)
On-Break Events:	rpt_before_aep_company(…)
	rpt_after_aep_company(…)
	rpt_before_aep_paygroup(…)
	rpt_after_aep_paygroup(…)
	rpt_before_aep_deptid(…)
	rpt_after_aep_deptid(…)
Report Data Events:	rpt_detail_section(…)
	rpt_skip(…)
	rpt_init_variables(…)

On-Break Events refer to data condition when value of the key fields in current data row is different to value of the same field in previous data row.

All above report events (PeopleCode Functions) reside in FuncLib PeopleCode. For our current report, these will be in AEP_MY_FRST_PDF.AEP_RPT_EVENTS FieldFormula Object. The record definition would look like below when these PeopleCode functions have been loaded:



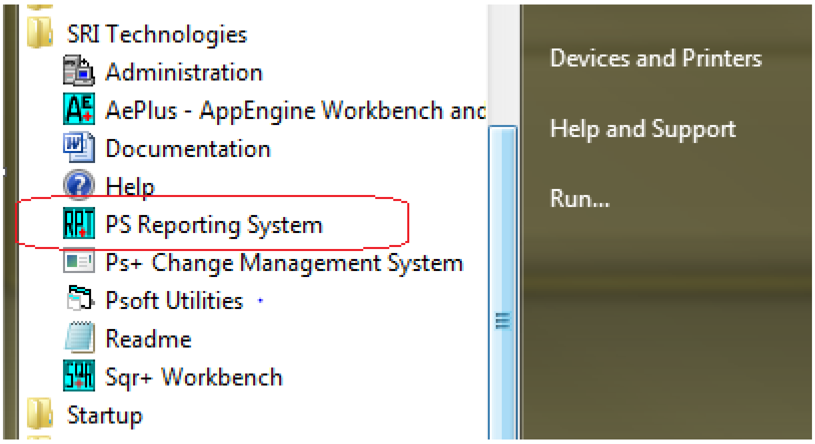
Num	Field Name	Type	FDe	FEd	FCh	FFo	RIn	RIs
1	AEP_COMPANY	Char						
2	AEP_PAYGROUP	Char						
3	AEP_DEPTID	Char						
4	**AEP_RPT_EVENTS**	Char				✓		

Please note that **On-Break Events** are required for all fields specified in AEP_MY_FRST_PDF record that appear before field AEP_RPT_EVENTS.

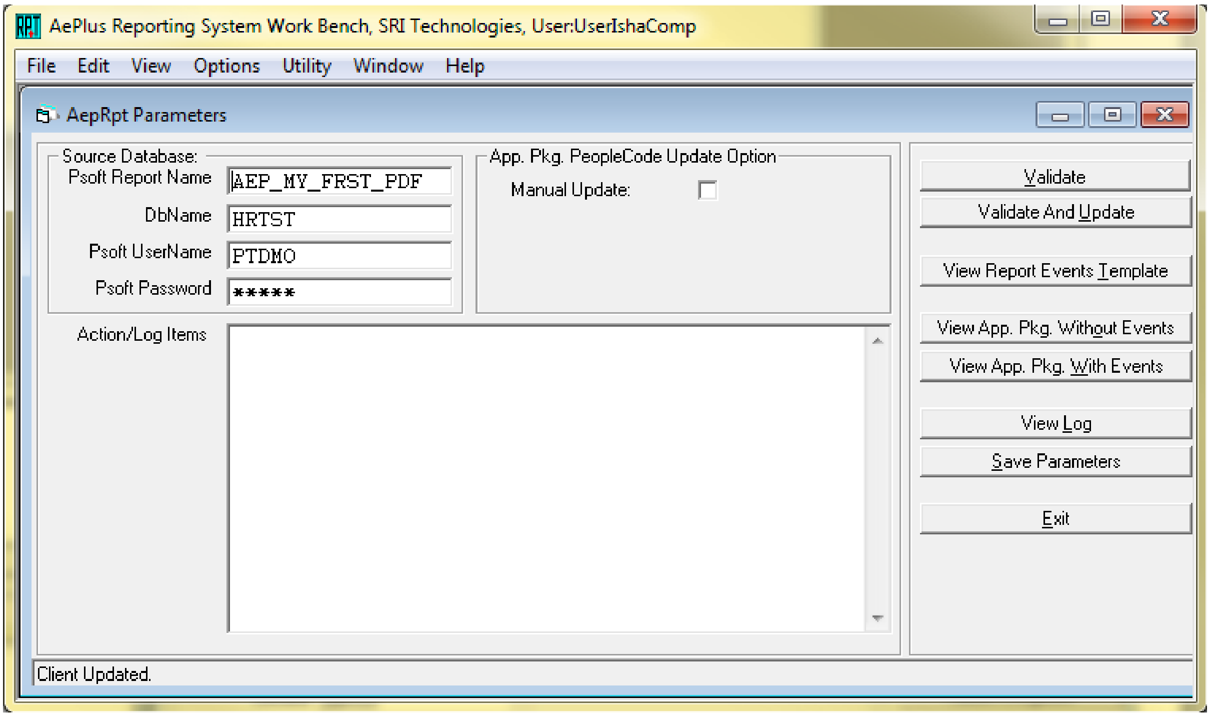
We will use AePlus Reporting System Workbench to create App. Pkg. AEP_MY_FRST_PDF.AEP_MY_FRST_PDF PeopleCode. This will also create PeopleCode Template for FuncLib functions that developer can fill-in as appropriate by calling various API's to construct pages of the report. Both Application Class and FuncLib PeopleCode are tightly coupled and must migrate in tandem from one environment to other. These Event Functions are triggered by AePlus Reporting System and developer should not remove or call them directly.

AePlus Reporting System Workbench:

Now is the time to get into PeopleCode stuff. We will generate PeopleCode Template using AePlus Reporting System Workbench. Reporting System Workbench can be invoked by running aeprpt.exe from Server folder. This can also be started from Windows Start Menu:



Once invoked successfully, you will get into following screen:


AePlus Reporting System Work Bench, SRI Technologies, User:UserIshaComp

File Edit View Options Utility Window Help
AepRpt Parameters

Source Database:

Psoft Report Name: AEP_MY_FRST_PDF

DbName: HRTST

Psoft UserName: PTDMO

Psoft Password: *****

App. Pkg. PeopleCode Update Option

Manual Update: □

Action/Log Items

Validate
Validate And Update
View Report Events Template
View App. Pkg. Without Events
View App. Pkg. With Events
View Log
Save Parameters
Exit

Client Updated.

Enter Prompts:

For our current report, enter prompts as:

Psoft Report Name: AEP_MY_FRST_PDF

This is our Report Name

DbName: HRTST

This is the database where we are developing Report

PsoftUserName: PTDMO

This is the User Is that we use to logon to AppDesigner

Psoft Password: <Password>

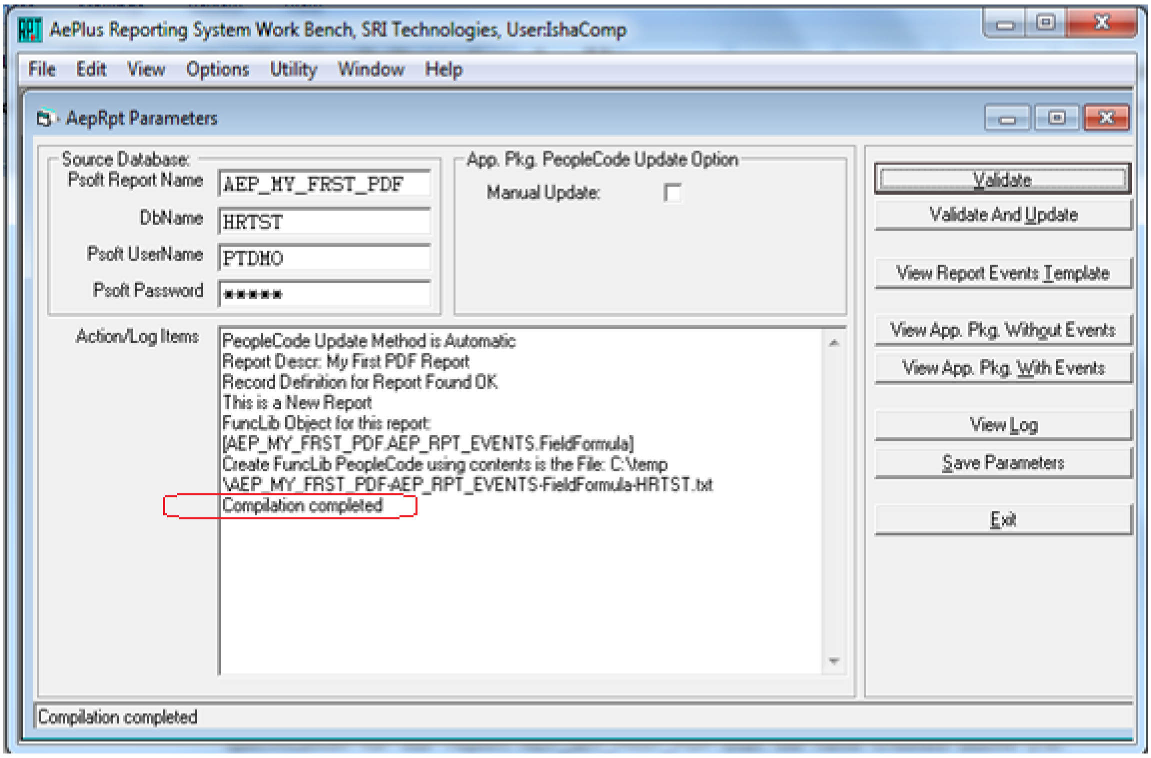
This is the password we use to logon to App Designer

Manual Update: Let this be unchecked.

We would like to create App. Pkg. PeopleCode automatically.

Validate Report:

After filling all the prompts, we will click 'Validate' button to validate the report specification for our report AEP_MY_FRST_PDF that we have created above (i.e. Record Definition AEP_MY_FRST_PDF). Since this is the new report and Manual Update is unchecked, the screen will look like:


AePlus Reporting System Work Bench, SRI Technologies, User:IshaComp
File
Edit
View
Options
Utility
Window
Help
AepRpt Parameters
Source Database:
App. Pkg. PeopleCode Update Option
Validate
Validate And Update
View Report Events Template
View App. Pkg. Without Events
View App. Pkg. With Events
View Log
Save Parameters
Exit

Psoft Report Name AEP_MY_FRST_PDF

DbName HRTST

Psoft UserName PTDMO

Psoft Password *****

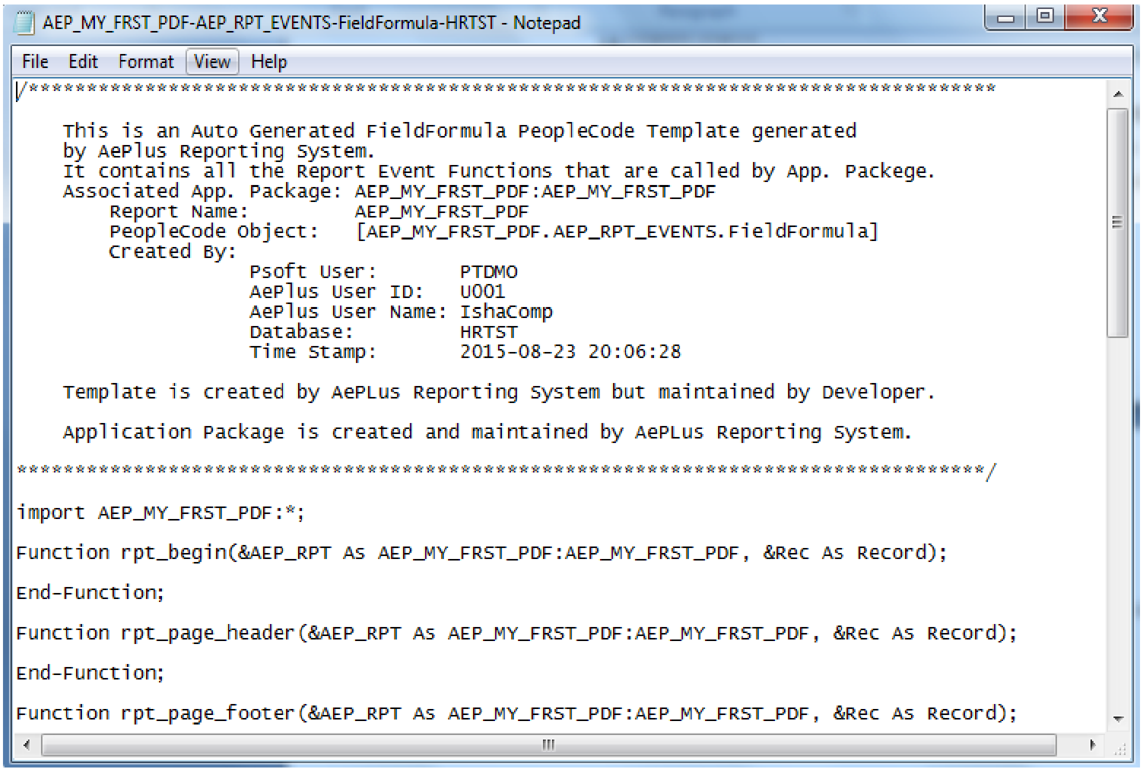
Manual Update: □

Action/Log Items

```
PeopleCode Update Method is Automatic
Report Descr: My First PDF Report
Record Definition for Report Found OK
This is a New Report
FuncLib Object for this report:
[AEP_MY_FRST_PDF.AEP_RPT_EVENTS.FieldFormula]
Create FuncLib PeopleCode using contents is the File: C:\temp
\AEP_MY_FRST_PDF-AEP_RPT_EVENTS-FieldFormula-HRTST.txt
Compilation completed
```

Compilation completed

As you can see the messages in Action/Log Items Textbox, report 'AEP_MY_FRST_PDF' is new meaning FuncLib and App. Pkg. PeopleCode associated with this report does not exists in the database. Also, Validation process has created FuncLib PeopleCode Template that can be viewed by clicking 'View Report Events Template'. The Template would look like:

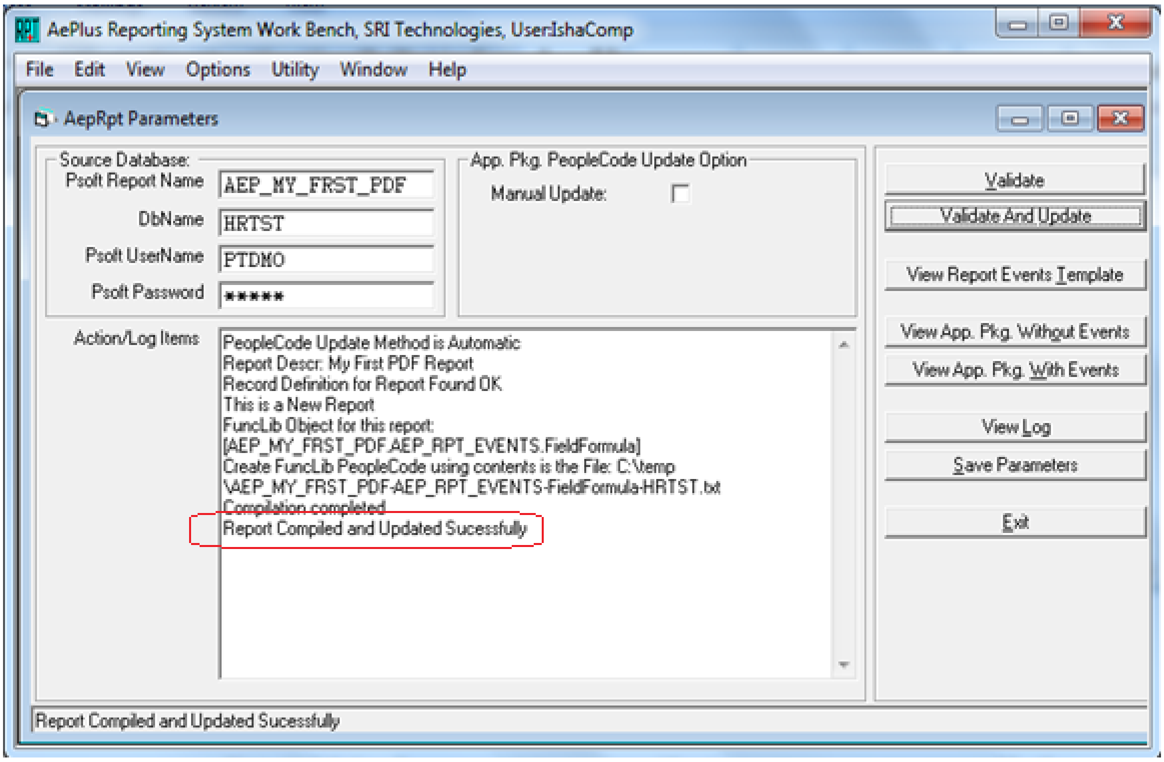


```
/************************************************************************************

    This is an Auto Generated FieldFormula PeopleCode Template generated
    by AePlus Reporting System.
    It contains all the Report Event Functions that are called by App. Packege.
    Associated App. Package: AEP_MY_FRST_PDF:AEP_MY_FRST_PDF
        Report Name:         AEP_MY_FRST_PDF
        PeopleCode Object:   [AEP_MY_FRST_PDF.AEP_RPT_EVENTS.FieldFormula]
        Created By:
                    Psoft User:        PTDMO
                    AePlus User ID:    U001
                    AePlus User Name:  IshaComp
                    Database:          HRTST
                    Time Stamp:        2015-08-23 20:06:28

    Template is created by AePLus Reporting System but maintained by Developer.

    Application Package is created and maintained by AePLus Reporting System.

************************************************************************************/

import AEP_MY_FRST_PDF:*;

Function rpt_begin(&AEP_RPT As AEP_MY_FRST_PDF:AEP_MY_FRST_PDF, &Rec As Record);

End-Function;

Function rpt_page_header(&AEP_RPT As AEP_MY_FRST_PDF:AEP_MY_FRST_PDF, &Rec As Record);

End-Function;

Function rpt_page_footer(&AEP_RPT As AEP_MY_FRST_PDF:AEP_MY_FRST_PDF, &Rec As Record);
```

Above PeopleCode that simply contains various Report Event Functions (without the function body) in file: C:\temp\AEP_MY_FRST_PDF-AEP_RPT_EVENTS-FieldFormula-HRTST.txt.

Validate And Update Report:

Since Validation Process went to success without any errors, we can proceed further by clicking 'Validate and Update' button that will recreate C:\temp\AEP_MY_FRST_PDF-AEP_RPT_EVENTS-FieldFormula-HRTST.txt file along with App. Pkg. AEP_MY_FRST_PDF in HRTST database. The screen will look like:


AePlus Reporting System Work Bench, SRI Technologies, User:IshaComp
File Edit View Options Utility Window Help
AepRpt Parameters
Validate
Validate And Update
View Report Events Template
View App. Pkg. Without Events
View App. Pkg. With Events
View Log
Save Parameters
Exit

Source Database:
Psoft Report Name: AEP_MY_FRST_PDF
DbName: HRTST
Psoft UserName: PTDMO
Psoft Password: *****

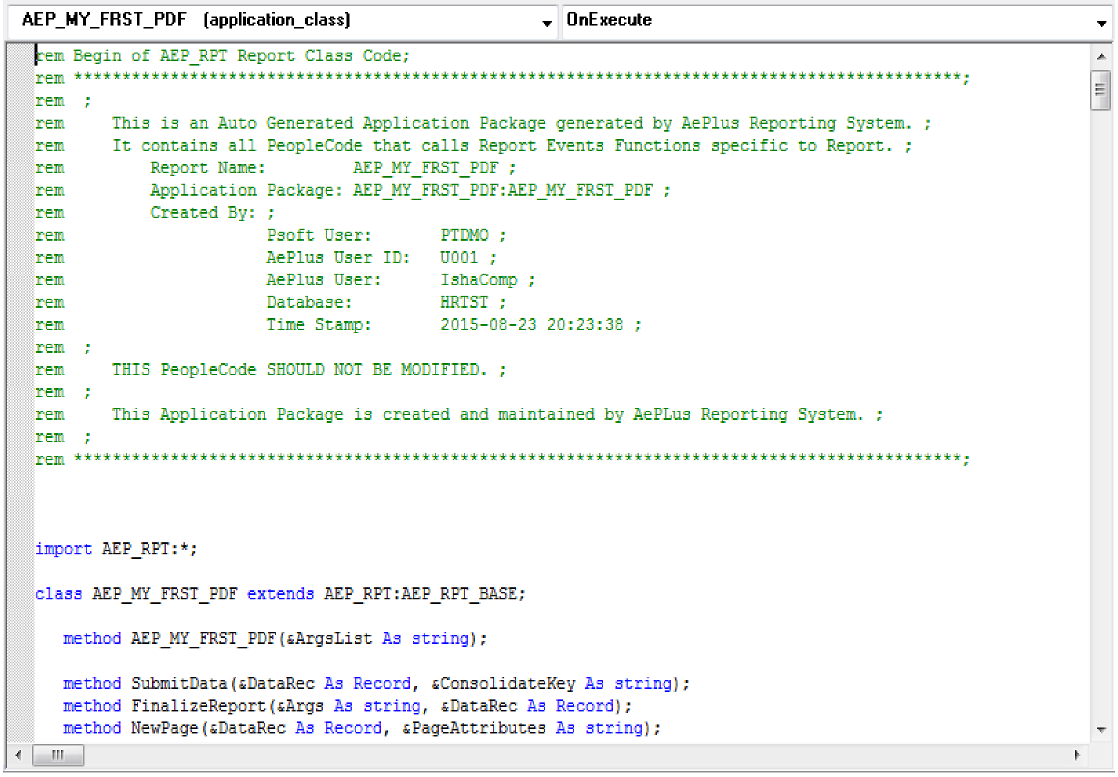
App. Pkg. PeopleCode Update Option
Manual Update: □

Action/Log Items

```
PeopleCode Update Method is Automatic
Report Descr: My First PDF Report
Record Definition for Report Found OK
This is a New Report
FuncLib Object for this report:
[AEP_MY_FRST_PDF.AEP_RPT_EVENTS.FieldFormula]
Create FuncLib PeopleCode using contents is the File: C:\temp
\AEP_MY_FRST_PDF-AEP_RPT_EVENTS-FieldFormula-HRTST.txt
Compilation completed
Report Compiled and Updated Sucessfully
```

Report Compiled and Updated Sucessfully

At this stage, you can view App. Pkg. AEP_MY_FRST_PDF in database using App. Designer that will look like:



```
rem Begin of AEP_RPT Report Class Code;
rem ********************************************************************************************;
rem  ;
rem     This is an Auto Generated Application Package generated by AePlus Reporting System. ;
rem     It contains all PeopleCode that calls Report Events Functions specific to Report. ;
rem         Report Name:         AEP_MY_FRST_PDF ;
rem         Application Package: AEP_MY_FRST_PDF:AEP_MY_FRST_PDF ;
rem         Created By: ;
rem                     Psoft User:       PTDMO ;
rem                     AePlus User ID:   U001 ;
rem                     AePlus User:      IshaComp ;
rem                     Database:         HRTST ;
rem                     Time Stamp:       2015-08-23 20:23:38 ;
rem  ;
rem     THIS PeopleCode SHOULD NOT BE MODIFIED. ;
rem  ;
rem     This Application Package is created and maintained by AePLus Reporting System. ;
rem  ;
rem ********************************************************************************************;



import AEP_RPT:*;

class AEP_MY_FRST_PDF extends AEP_RPT:AEP_RPT_BASE;

   method AEP_MY_FRST_PDF(&ArgsList As string);

   method SubmitData(&DataRec As Record, &ConsolidateKey As string);
   method FinalizeReport(&Args As string, &DataRec As Record);
   method NewPage(&DataRec As Record, &PageAttributes As string);
```

That's all about using Reporting System Workbench to generate PeopleCode objects. Now we need to enhance FuncLib PeopleCode to complete each Report Event Functions.

Developer can copy and paste contents of this file C:\temp\AEP_MY_FRST_PDF-AEP_RPT_EVENTS-FieldFormula-HRTST.txt into Record: AEP_MY_FRST_PDF, Field: AEP_RPT_EVENTS FieldFormula PeopleCode using App. Designer and enhance.

It is recommended that developer makes a dummy change in App. Pkg. PeopleCode also and save so that App. Pkg. PeopleCode Object gets inserted into App Designer Project to take care of migration to other databases.

The final step is to fill in PeopleCode against each Report Event function that is described further in this document.

PeopleCode Development/Maintenance (FuncLib):

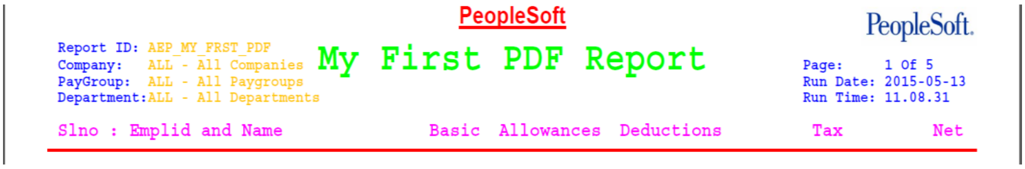
The next step will be fill-in these template functions. Let's do it one by one.

Function rpt_begin(…):

This function is called once by the AePlus Reporting System before we start printing on the report. We will leave this as blank as we do not intend to do anything here for this report. However, if you also intend to create a CSV file in addition to PDF, you could define the header line for CSV file here.

Function rpt_page_header(…):

This function is called by the AePlus Reporting System whenever NewLine() method finds that current page is full (no more lines can be printed on the page). It is also triggered by NewPage() method that forces new page even though the page in not full. Here, we will use some print() methods to decorate the header of the page. The requirement is that page header should look like:



```
                                   PeopleSoft
                                                                           PeopleSoft.
Report ID: AEP_MY_FRST_PDF
Company:   ALL - All Companies  My First PDF Report                  Page:      1 Of 5
PayGroup:  ALL - All Paygroups                                       Run Date: 2015-05-13
Department:ALL - All Departments                                     Run Time: 11.08.31

Slno : Emplid and Name                      Basic  Allowances  Deductions       Tax         Net
```

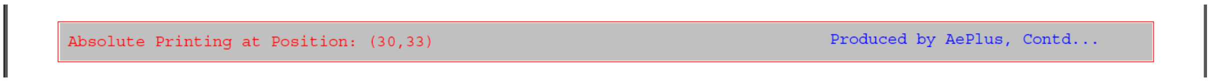
The corresponding PeoplCode is:

```
Function rpt_page_header(&AEP_RPT As AEP_MY_FRST_PDF:AEP_MY_FRST_PDF, &Rec As Record);

   Local number &xPoint, &yPoint, &xPoint2, &yPoint2;

&xPoint = &AEP_RPT.get_CurrLeftMargin();
&yPoint = &AEP_RPT.get_PageBottomRight_y() - 20;
&xPoint2 = &AEP_RPT.get_PageBottomRight_x();
&yPoint2 = &yPoint - 20;

&AEP_RPT.NewLine(1, &Rec);
&AEP_RPT.PrintStr(1, "PeopleSoft",
"FontName=Helvetica,FontSize=12,Bold,Underline,Centre=,Color=red");

rem print image at Top Right Corner of the page reducing size of the image to 20%;
&AEP_RPT.PrintImage(&AEP_RPT.get_PageBottomRight_x(), &AEP_RPT.get_PageTopLeft_y(),
0.2, "d:\sqrplus\docs\peoplesoft.jpg");

&AEP_RPT.NewLine(1, &Rec);
&AEP_RPT.PrintStr(1, "Report ID: ", "Bold,Color=blue");
&AEP_RPT.PrintStr(10, Upper(&AEP_RPT.ReportName), "Bold,Color=orange");

&AEP_RPT.NewLine(1, &Rec);
&AEP_RPT.PrintStr(1, "Company:", "Bold,Color=blue");
&AEP_RPT.PrintStr(10, "ALL - All Companies", "Bold,Color=orange");
&AEP_RPT.PrintStr(1, &AEP_RPT.ReportDescr, "FontSize=20,Bold,Centre=,Color=green");
&AEP_RPT.PrintStr(75, "Page:      " | &AEP_RPT.get_PageNum() | " Of ",
"Bold,Color=blue");
&AEP_RPT.PrintTotalPages("Bold,Color=blue");

&AEP_RPT.NewLine(1, &Rec);
&AEP_RPT.AdjustCurrPointY( - (&AEP_RPT.get_CharHeight() * 1.5));
&AEP_RPT.PrintStr(1, "PayGroup:", "Bold,Color=blue");
&AEP_RPT.PrintStr(10, "ALL - All Paygroups", "Bold,Color=orange");
&AEP_RPT.PrintStr(75, "Run Date: " | %Date, "Bold,Color=blue");

&AEP_RPT.NewLine(1, &Rec);
&AEP_RPT.PrintStr(1, "Department:", "Bold,Color=blue");
&AEP_RPT.PrintStr(10, "ALL - All Departments", "Bold,Color=orange");
&AEP_RPT.PrintStr(75, "Run Time: " | Left(String(%Time), 8), "Bold,Color=blue");
&AEP_RPT.NewLine(1, &Rec); /* blank line */

&AEP_RPT.NewLine(1, &Rec);
&AEP_RPT.PrintStr(1, "Slno : Emplid and Name", &PrintAtr_HeaderLine);
&AEP_RPT.PrintStr(&Pos_Basic, "Basic", &PrintAtr_HeaderLineRight);
&AEP_RPT.PrintStr(&pos_Allow, "Allowances", &PrintAtr_HeaderLineRight);
&AEP_RPT.PrintStr(&Pos_Ded, "Deductions", &PrintAtr_HeaderLineRight);
```

```
&AEP_RPT.PrintStr(&Pos_Tax, "Tax", &PrintAtr_HeaderLineRight);
&AEP_RPT.PrintStr(&Pos_Net, "Net", &PrintAtr_HeaderLineRight);
&AEP_RPT.NewLine(1, &Rec);

rem draw line;
&xPoint = &AEP_RPT.get_CurrLeftMargin();
&yPoint = &AEP_RPT.get_CurrPointY() + (&AEP_RPT.get_CharHeight() * 0.5);
&xPoint2 = &AEP_RPT.get_CurrPageWidth() - &AEP_RPT.get_CurrRightMargin();
&yPoint2 = &yPoint;
&AEP_RPT.DrawLine(&xPoint, &yPoint, &xPoint2, &yPoint2, "color=red");

   /* insert some vertical blank space */
&AEP_RPT.AdjustCurrPointY(&AEP_RPT.get_CharHeight() * 1.5);

End-Function;
```

Function rpt_page_footer(…):

This function is called by the AePlus Reporting System just before rpt_page_header() but for the previous page. Assuming that if footer of the page should look like:


Absolute Printing at Position: (30,33)
Produced by AePlus, Contd...

The corresponding PeopleCode will be:

```
Function rpt_page_footer(&AEP_RPT As AEP_MY_FRST_PDF:AEP_MY_FRST_PDF, &Rec As Record);

   Local number &xPoint, &yPoint, &xPoint2, &yPoint2;

rem draw rectangle at the footer of the page;
&xPoint = &AEP_RPT.get_CurrLeftMargin();
&yPoint = &AEP_RPT.get_PageBottomRight_y() + 20;
&xPoint2 = &AEP_RPT.get_PageBottomRight_x();
&yPoint2 = &yPoint - 20;
&AEP_RPT.DrawBox(&xPoint, &yPoint, &xPoint2, &yPoint2,
"lineThickness=0,color=red,fillcolor=lightGray");

rem fill is Opaque so printing should be done after filling the box;
&AEP_RPT.NewLine(1, &Rec);
   If &AEP_RPT.LastReportPage() = True Then
&AEP_RPT.AdjustCurrPointY( - (&AEP_RPT.get_CharHeight() * 2.0));
&AEP_RPT.PrintStr(65, "Produced by AePlus ", "color=blue");
   Else
&AEP_RPT.PrintStr(65, "Produced by AePlus, Contd... ", "color=blue");
   End-If;

rem some absolute printing;
&xPoint = &xPoint + 5;
&yPoint = &yPoint - 12;
&AEP_RPT.PrintStrAbs(&xPoint, &yPoint, "Absolute Printing at Position: (" |
Integer(&xPoint) | "," | Integer(&yPoint) | ")", "color=red");

End-Function;
```

Function rpt_before_aep_company(…):

This function is called by the AePlus Reporting System whenever value of company changes before employee data is printed on the report. Based on requirements we will print code and description. The PeopleCode may be:

```
Function rpt_before_aep_company(&AEP_RPT As AEP_MY_FRST_PDF:AEP_MY_FRST_PDF, &Rec As
Record);

   Local string &filename;
   Local string &descr;

&descr = "Before Company: " | &Rec.GetField(Field.AEP_COMPANY).Value | ", EffDt: " |
&Rec.GetField(Field.EFFDT).Value;
&AEP_RPT.NewPage(&Rec, "");
&AEP_RPT.PrintStr(1, &descr, &PrintAtr_BeforeTitle);

End-Function;
```

Function rpt_before_aep_paygroup(…):

This function is called by the AePlus Reporting System whenever value of Pay Group changes before employee data is printed on the report. Based on requirements we will print code and description. The PeopleCode may be:

```
Function rpt_before_aep_paygroup(&AEP_RPT As AEP_MY_FRST_PDF:AEP_MY_FRST_PDF, &Rec As
Record);

&AEP RPT.NewLine(2, &Rec);
&AEP RPT.PrintStr(1, "Before Paygroup: " | &Rec.GetField(Field.AEP_PAYGROUP).Value,
&PrintAtr_BeforeTitle);

End-Function;
```

Function rpt_before_aep_deptid(…):

This function is called by the AePlus Reporting System whenever value of Department before when employee data is printed on the report. Based on requirements we will print code and description. The PeopleCode may be:

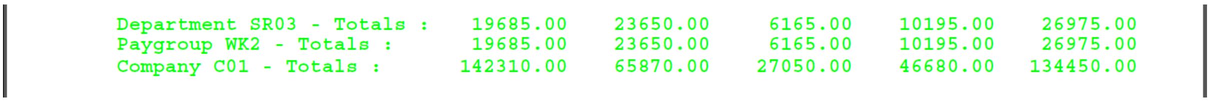
```
Function rpt_before_aep_deptid(&AEP_RPT As AEP_MY_FRST_PDF:AEP_MY_FRST_PDF, &Rec As
Record);

&AEP RPT.NewLine(3, &Rec);
&AEP RPT.PrintStr(1, "Before Dept: " | &Rec.GetField(Field.AEP_DEPTID).Value,
&PrintAtr_BeforeTitle | ",underline");

End-Function;
```

Function rpt_after_aep_company(…)

Function rpt_after_aep_paygroup(…):

Function rpt_after_aep_deptid(…):

This function is called by the AePlus Reporting System whenever value of Company/PayGroup/Department changes after employee data is printed on the report. Based on requirements we will print column totals. Assuming the format is like:



```
        Department SR03 - Totals :   19685.00    23650.00     6165.00    10195.00    26975.00
        Paygroup WK2 - Totals :      19685.00    23650.00     6165.00    10195.00    26975.00
        Company C01 - Totals :      142310.00    65870.00    27050.00    46680.00   134450.00
```

```
Function rpt_after_aep_company(&AEP_RPT As AEP_MY_FRST_PDF:AEP_MY_FRST_PDF, &Rec As
Record);

print totals and reset(&AEP RPT, 1, &Rec, "Company",
&Rec.GetField(Field.AEP COMPANY).Value, &gbl company total basic,
&gbl company total allow, &gbl_company_total_ded, &gbl_company_total_tax,
&gbl_company_total_net);

End-Function;

Function rpt_after_aep_paygroup(&AEP_RPT As AEP_MY_FRST_PDF:AEP_MY_FRST_PDF, &Rec As
Record);

print totals and reset(&AEP RPT, 2, &Rec, "Paygroup",
&Rec.GetField(Field.AEP PAYGROUP).Value, &gbl paygroup total basic,
&gbl paygroup total allow, &gbl_paygroup_total_ded, &gbl_paygroup_total_tax,
&gbl_paygroup_total_net);

End-Function;

Function rpt_after_aep_deptid(&AEP_RPT As AEP_MY_FRST_PDF:AEP_MY_FRST_PDF, &Rec As
Record);

print totals and reset(&AEP RPT, 3, &Rec, "Department",
&Rec.GetField(Field.AEP DEPTID).Value, &gbl dept total basic, &gbl_dept_total_allow,
&gbl_dept_total_ded, &gbl_dept_total_tax, &gbl_dept_total_net);

End-Function;
```

Here we have created a function that prints column totals and resets them after these have been printed. The PeopleCode is:

```
Function print totals and reset(&AEP RPT As AEP MY FRST PDF:AEP MY FRST PDF,
&NeedLines As number, &Rec As Record, &key desc As string, &key val As string, &Basic
As number, &Allow As number, &Ded As number, &Tax As number, &Net As number)

   Local string &line;

rem common function - called from 'AFTER' break events;
&AEP RPT.NewLine(&NeedLines, &Rec);
&line = &key desc | " " | &key val | " - Totals :";
&AEP_RPT.PrintStr(5, &line, &PrintAtr_TotalLine);

&AEP RPT.PrintStr(&Pos Basic, NumberToString("%6.2", &Basic),
&PrintAtr TotalLineRight);
&AEP RPT.PrintStr(&pos Allow, NumberToString("%6.2", &Allow),
&PrintAtr TotalLineRight);
&AEP RPT.PrintStr(&Pos Ded, NumberToString("%6.2", &Ded), &PrintAtr TotalLineRight);
&AEP RPT.PrintStr(&Pos Tax, NumberToString("%6.2", &Tax), &PrintAtr TotalLineRight);
&AEP_RPT.PrintStr(&Pos_Net, NumberToString("%6.2", &Net), &PrintAtr_TotalLineRight);

rem initialise totals after printing;
&Basic = 0;
&Allow = 0;
&Ded = 0;
&Tax = 0;
&Net = 0;

End-Function;
```

Function rpt_skip(…):

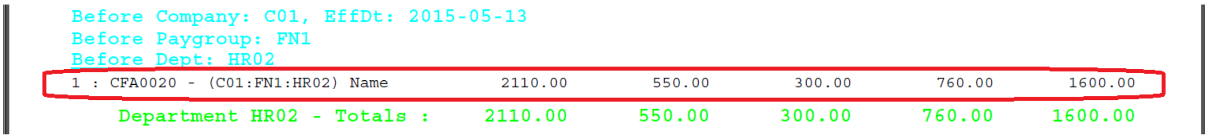
This function is called by the AePlus Reporting System to check if an employee should be included in the report after consolidating multiple rows read for an employee from **Input Data Stream**. Based on the requirements, the PeopleCode would be:

```
Function rpt_skip(&AEP_RPT As AEP_MY_FRST_PDF:AEP_MY_FRST_PDF, &Rec As Record) Returns
boolean;

rem if row needs to be skiped based on certain conditions, true should be returned;

   Local boolean&RtnSts;

&RtnSts = False;

   If &gbl tax< 700 Or
&gbl net< 1500 Then
&RtnSts = True;
   End-If;

   Return &RtnSts;

End-Function;
```

Consolidation/Aggregation and skip function is an important feature of AePlus Reporting System. You may have a situation where while consolidation/aggregation, elimination of employee is not possible (or too difficult) at SQL Select level when data row is received from the database tables. This feature gives developer a flexible way where complex requirements can be handled by just a few lines of PeopleCode. We think that developer would prefer this approach instead of writing complex SQL that would be difficult to develop and maintain.

Function rpt_detail_section(…):

This function is called by the AePlus Reporting System to print employee detail line after consolidating multiple rows read for an employee from **Input Data Stream**. This is called only if rpt_skip function returns false value. Here we will print one line per employee and after printing, will update column accumulators that are required in rpt_after events. Assuming that format for the detail line is like:



```
     Before Company: C01, EffDt: 2015-05-13
     Before Paygroup: FN1
     Before Dept: HR02
     1 : CFA0020 - (C01:FN1:HR02) Name           2110.00        550.00        300.00        760.00        1600.00
          Department HR02 - Totals :        2110.00      550.00      300.00      760.00      1600.00
```

```
Function rpt_detail_section(&AEP_RPT As AEP_MY_FRST_PDF:AEP_MY_FRST_PDF, &Rec As
Record);

   Local string &print_str;
   Local string &PrintAtr_detail_right = "right";

&AEP_RPT.NewLine(1, &Rec);
&print_str = &AEP_RPT.get_RptRowNum() | " : " |
&Rec.GetField(Field.AEP_KEY_DETAIL).Value | " - (" |
&Rec.GetField(Field.AEP_COMPANY).Value | ":" | &Rec.GetField(Field.AEP_PAYGROUP).Value
| ":" | &Rec.GetField(Field.AEP_DEPTID).Value | ") Name";
&AEP_RPT.PrintStr(1, &print_str, &PrintAtr_Default);

&AEP_RPT.PrintStr(43, NumberToString("%6.2", &gbl_basic), &PrintAtr_detail_right);
&AEP_RPT.PrintStr(55, NumberToString("%6.2", &gbl_allow), &PrintAtr_detail_right);
&AEP_RPT.PrintStr(67, NumberToString("%6.2", &gbl_ded), &PrintAtr_detail_right);
&AEP_RPT.PrintStr(79, NumberToString("%6.2", &gbl_tax), &PrintAtr_detail_right);
&AEP_RPT.PrintStr(91, NumberToString("%6.2", &gbl_net), &PrintAtr_detail_right);

rem accumulate break levels totals;
&gbl_company_total_basic = &gbl_company_total_basic + &gbl_basic;
&gbl_company_total_allow = &gbl_company_total_allow + &gbl_allow;
&gbl_company_total_ded = &gbl_company_total_ded + &gbl_ded;
&gbl_company_total_tax = &gbl_company_total_tax + &gbl_tax;
&gbl_company_total_net = &gbl_company_total_net + &gbl_net;

&gbl_paygroup_total_basic = &gbl_paygroup_total_basic + &gbl_basic;
&gbl_paygroup_total_allow = &gbl_paygroup_total_allow + &gbl_allow;
&gbl_paygroup_total_ded = &gbl_paygroup_total_ded + &gbl_ded;
&gbl_paygroup_total_tax = &gbl_paygroup_total_tax + &gbl_tax;
&gbl_paygroup_total_net = &gbl_paygroup_total_net + &gbl_net;

&gbl_dept_total_basic = &gbl_dept_total_basic + &gbl_basic;
&gbl_dept_total_allow = &gbl_dept_total_allow + &gbl_allow;
&gbl_dept_total_ded = &gbl_dept_total_ded + &gbl_ded;
&gbl_dept_total_tax = &gbl_dept_total_tax + &gbl_tax;
&gbl_dept_total_net = &gbl_dept_total_net + &gbl_net;

&gbl_grand_total_basic = &gbl_grand_total_basic + &gbl_basic;
&gbl_grand_total_allow = &gbl_grand_total_allow + &gbl_allow;
&gbl_grand_total_ded = &gbl_grand_total_ded + &gbl_ded;
&gbl_grand_total_tax = &gbl_grand_total_tax + &gbl_tax;
&gbl_grand_total_net = &gbl_grand_total_net + &gbl_net;

End-Function;
```

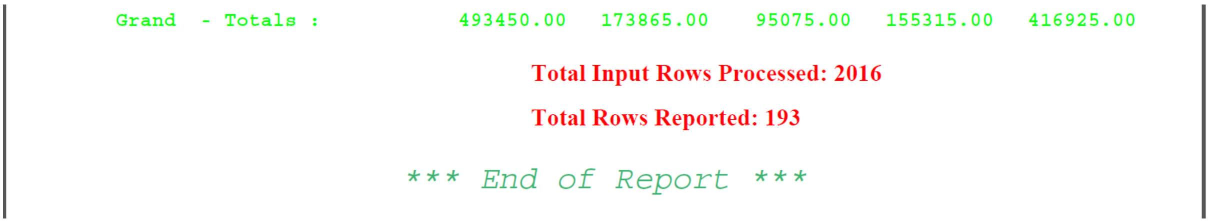
Function rpt_init_variables(…):

This function is called by the AePlus Reporting System whenever there is a need to reset variables that are derived from row read from **Input Data Stream** and used in consolidating multiple rows received for an employee. In this report following code will be required:

```
Function rpt_init_variables(&AEP_RPT As AEP_MY_FRST_PDF:AEP_MY_FRST_PDF, &Rec As
Record);

reminit variables;
&gbl_net = 0;
&gbl_basic = 0;
&gbl_allow = 0;
&gbl_ded = 0;
&gbl_tax = 0;

End-Function;
```

Function rpt_end(…):

This function is called once by the AePlus Reporting System when all the rows from **Input Data Stream** have been processed. Developer should use this function to print any grand totals, statistics or End of Report Message. Assuming the format is:



```
        Grand  - Totals :          493450.00   173865.00    95075.00   155315.00   416925.00

                              Total Input Rows Processed: 2016

                              Total Rows Reported: 193

                                  *** End of Report ***
```

PeopleCode would be:

```
Function rpt_end(&AEP_RPT As AEP_MY_FRST_PDF:AEP_MY_FRST_PDF, &Rec As Record);

   Local string &PrintAtr_str = "FontName=Times-
Roman,color=16711680,bold,FontSize=12";

   If &AEP_RPT.DataReported() = True Then
print_totals_and_reset(&AEP_RPT, 10, &Rec, "Grand", "", &gbl_grand_total_basic,
&gbl_grand_total_allow, &gbl_grand_total_ded, &gbl_grand_total_tax,
&gbl_grand_total_net);
   End-If;

&AEP_RPT.NewLine(1, &Rec);
&AEP_RPT.NewLine(1, &Rec);
&AEP_RPT.NewLine(1, &Rec);
&AEP_RPT.PrintStr(40, "Total Input Rows Processed: " | &AEP_RPT.get_InputRowNum(),
&PrintAtr_str);

   /* insert some vertical blank space */
&AEP_RPT.AdjustCurrPointY(&AEP_RPT.get_CharHeight());

&AEP_RPT.NewLine(1, &Rec);
&AEP_RPT.PrintStr(40, "Total Rows Reported: " | &AEP_RPT.get_RptRowNum(),
&PrintAtr_str);

&AEP_RPT.NewLine(1, &Rec);
&AEP_RPT.NewLine(1, &Rec);
&AEP_RPT.NewLine(1, &Rec);
&AEP_RPT.PrintStr(1, "*** End of Report ***", "fontsize=16,centre=,italic,color =
rgb(60 : 179 : 113)");
&AEP_RPT.NewLine(1, &Rec);
&AEP_RPT.NewLine(1, &Rec);

End-Function;
```

That’s the end of the report event functions.

Miscellaneous code in FuncLib(…):

In order to facilitate accumulators at company, paygroup and department level, we have created component level variables:

```
Component number &gbl company total basic, &gbl company total allow,
&gbl company total ded, &gbl company total tax, &gbl company total net;
Component number &gbl paygroup total basic, &gbl paygroup total allow,
&gbl paygroup total ded, &gbl paygroup total tax, &gbl paygroup total net;
Component number &gbl dept total basic, &gbl_dept_total_allow, &gbl_dept_total_ded,
&gbl dept total tax, &gbl dept total net;
Component number &gbl grand total basic, &gbl_grand_total_allow, &gbl_grand_total_ded,
&gbl_grand_total_tax, &gbl_grand_total_net;
```

Also, to facilitate consolidation as employee level, we have created some global variables (though these can be declared at component level).

```
Global number &gbl_basic, &gbl_allow, &gbl_ded, &gbl_tax, &gbl_net;
```

We have also created some constants to hold Print Attributes and Column Positions:

```
rem Formatting constants;
Constant &PrintAtr HeaderLine = "FontSize=10, Bold, Color=magenta";
Constant &PrintAtr HeaderLineRight = "FontSize = 10, Bold,Color = magenta, right";
Constant &PrintAtr BeforeTitle = "FontSize=10,Bold,Color=cyan";
Constant &PrintAtr TotalLine = "FontSize=10,Bold,Color=green";
Constant &PrintAtr TotalLineRight = "FontSize=10,Bold,Color=green,right";
Constant &PrintAtr_Default = "";

rem column position constants;
Constant &Pos Basic = 43;
Constant &pos Allow = 55;
Constant &Pos Ded = 67;
Constant &Pos Tax = 79;
Constant &Pos_Net = 91;
```

PeopleCode Development/Maintenance (AppEngine):

That’s all about putting code against various Report Events and now is the time to write some code to initiate the report. We are using Application Engine PeopleCode for this. Let’s create a new Application Engine MY_FRST_PDF. Create a PeopleCode step and add following code:

```
import AEP_MY_FRST_PDF:*;

rem global variables used in event functions;
Global number &gbl_net, &gbl_basic, &gbl_allow, &gbl_ded, &gbl_tax;

Function InputLineToRecord(&AEP_RPT As AEP_MY_FRST_PDF:AEP_MY_FRST_PDF, &InpLine As
string, &Rec As Record)

&Rec.GetField(Field.AEP_COMPANY).Value = &AEP_RPT.getFieldByNum(&InpLine, 1, ",");
&Rec.GetField(Field.AEP_PAYGROUP).Value = &AEP_RPT.getFieldByNum(&InpLine, 2, ",");
&Rec.GetField(Field.AEP_DEPTID).Value = &AEP_RPT.getFieldByNum(&InpLine, 3, ",");

&Rec.GetField(Field.AEP_KEY_DETAIL).Value = &AEP_RPT.getFieldByNum(&InpLine, 4, ",");
&Rec.GetField(Field.EFFDT).Value = %Date;
&Rec.GetField(Field.AEP_ER_DD_CODE).Value = &AEP_RPT.getFieldByNum(&InpLine, 6, ",");
&Rec.GetField(Field.AEP_AMOUNT).Value = Value(&AEP_RPT.getFieldByNum(&InpLine, 7,
","));

End-Function;

Function ConsolidateDetailsRows(&AEP_RPT As AEP_MY_FRST_PDF:AEP_MY_FRST_PDF, &InpRec
As Record, &RowNumber As number)

   Local string &ErnDedCd, &ErrMessage;
   Local number &Amount;

&ErnDedCd = &InpRec.GetField(Field.AEP_ER_DD_CODE).Value;
&Amount = &InpRec.GetField(Field.AEP_AMOUNT).Value;

   Evaluate &ErnDedCd
   When = "E01"
&gbl_basic = &gbl_basic + &Amount;
      Break;
   When = "E02"
&gbl_allow = &gbl_allow + &Amount;
      Break;
   When = "D01"
&gbl_ded = &gbl_ded + &Amount;
      Break;
   When = "D02"
&gbl_tax = &gbl_tax + &Amount;
      Break
   When-Other
&ErrMessage = "Invalid Earning/Deduction Code: " | &ErnDedCd | " - input ignored at
record number: " | &RowNumber;
&AEP_RPT.ToLog("E", &ErrMessage);
      Break;
   End-Evaluate;

&gbl_net = &gbl_basic + &gbl_allow - &gbl_ded - &gbl_tax;

End-Function;

Local AEP_MY_FRST_PDF:AEP_MY_FRST_PDF &my_rpt;
Local File &fMYFILE;
Local Record &Rec;
Local number &rownum;
Local string &Line, &RptAetRecName, &RptDatRecName, &ArgList;

&my_rpt = create AEP_MY_FRST_PDF:AEP_MY_FRST_PDF(“”);

rem build argument list;
&RptAetRecName = "AEP_MY_FRST_PDF";
&RptDatRecName = "AEP_MY_FRST_DAT";
&ArgList = "rpt_aet_rec=" | &RptAetRecName | ",";
&ArgList = &ArgList | "rpt_dat_rec=" | &RptDatRecName | ",";
&ArgList = &ArgList | "PageFooterSize=25";

If &my_rpt.InitReport(&ArgList) = True Then

&Rec = CreateRecord(Record.AEP_MY_FRST_DAT);
&rownum = 0;
&fMYFILE = GetFile("c:\sqrplus\sqr\demo05.csv", "R", %FilePath_Absolute);
   If &fMYFILE.IsOpen Then
```

```
      While &fMYFILE.ReadLine(&Line);
&rownum = &rownum + 1;
         If &rownum> 1 Then /* ignore header row in file */

InputLineToRecord(&my rpt, &Line, &Rec);
&my rpt.SubmitData(&Rec, &Rec.GetField(Field.AEP KEY DETAIL).Value);
ConsolidateDetailsRows(&my rpt, &Rec, &rownum); /* not required if every input row
should appear in the report */

         End-If;
      End-While;
&fMYFILE.Close();

&my_rpt.FinalizeReport("NoDataMsg=*** No Data to Report ***", &Rec);

   End-If;
End-If;

&my_rpt.ToLog("", "End");
```

Running the Report:

That's all – now you can run the App Engine to generate your pdf report. We have not created Process Definition for this but you can run the Application Engine on the client using AePlus Workbench.

After running this app engine and all goes well, following files will be created:

1. Pdf Report: aep_my_frst_pdf_rpt.pdf
2. Process Log File: aep_rpt*.log

Producing CSV File:

CSV file can be produced with very little additional programming efforts. In this example, we would like the CSV file to have following columns:

SlNo
Company
Pay Group
Department
EmplID
Name
Basic
Allowances
Deductions
Tax
Net

For the header line, we will put some PeopleCode in rpt_begin function (that was left blank earlier) and the code will look like:

```
Function rpt_begin(&AEP_RPT As AEP_MY_FRST_PDF:AEP_MY_FRST_PDF, &Rec As Record);

   Local string &CsvHeaderStr;
   Local string &Delim = ",";

rem setup header live for CSV file;
&CsvHeaderStr = "";
&CsvHeaderStr = &CsvHeaderStr | "SlNo" | &Delim;
&CsvHeaderStr = &CsvHeaderStr | "Company" | &Delim;
&CsvHeaderStr = &CsvHeaderStr | "Pay Group" | &Delim;
&CsvHeaderStr = &CsvHeaderStr | "Department" | &Delim;
&CsvHeaderStr = &CsvHeaderStr | "EmplID" | &Delim;
&CsvHeaderStr = &CsvHeaderStr | "Name" | &Delim;
&CsvHeaderStr = &CsvHeaderStr | "Basic" | &Delim;
&CsvHeaderStr = &CsvHeaderStr | "Allowances" | &Delim;
&CsvHeaderStr = &CsvHeaderStr | "Deductions" | &Delim;
&CsvHeaderStr = &CsvHeaderStr | "Tax" | &Delim;
&CsvHeaderStr = &CsvHeaderStr | "Net";

&AEP_RPT.CsvHeader(&Delim, &CsvHeaderStr);

End-Function;
```

Further, in rpt_detail_section function, we will write PeopleCode so that for each row that appears in the report will also appear in CSV file. However, we do not print everything in the detail line (like Company, Paygroup etc), we will use two types of Print Attribute to address this:

1. Use “csv=n” for Columns that appears in the detail line.
2. Use “CsvOnly=n” for columns that do not appear in the detail line.
   Where n is the column number.

So the PeopleCode in rpt_detail_section will look like:

```
Function rpt_detail_section(&AEP_RPT As AEP_MY_FRST_PDF:AEP_MY_FRST_PDF, &Rec As
Record);

   Local string &print str;
   Local string &PrintAtr_detail_right = "right";
```

```
&AEP RPT.NewLine(1, &Rec);
&print str = &AEP RPT.RptRowNum | " : " | &Rec.GetField(Field.AEP_KEY_DETAIL).Value |
" - (" | &Rec.GetField(Field.AEP COMPANY).Value | ":" |
&Rec.GetField(Field.AEP_PAYGROUP).Value | ":" | &Rec.GetField(Field.AEP_DEPTID).Value
| ") Name";
&AEP_RPT.PrintStr(1, &print_str, &PrintAtr_Default);

remCsvOnly Columns - these do not appear in the report but in CSV file;
&AEP RPT.PrintStr(1, NumberToString("%6.2", &AEP RPT.RptRowNum), "CsvOnly=1");
&AEP RPT.PrintStr(1, &Rec.GetField(Field.AEP COMPANY).Value, "CsvOnly=2");
&AEP RPT.PrintStr(1, &Rec.GetField(Field.AEP PAYGROUP).Value, "CsvOnly=3");
&AEP RPT.PrintStr(1, &Rec.GetField(Field.AEP DEPTID).Value, "CsvOnly=4");
&AEP RPT.PrintStr(1, &Rec.GetField(Field.AEP KEY DETAIL).Value, "CsvOnly=5");
&AEP RPT.PrintStr(1, "Last Name, " | &Rec.GetField(Field.AEP_KEY_DETAIL).Value,
"CsvOnly=6");

&AEP RPT.PrintStr(43, NumberToString("%6.2", &gbl_basic), &PrintAtr_detail_right |
",csv=7");
&AEP RPT.PrintStr(55, NumberToString("%6.2", &gbl_allow), &PrintAtr_detail_right |
",csv=8");
&AEP RPT.PrintStr(67, NumberToString("%6.2", &gbl_ded), &PrintAtr_detail_right |
",csv=9");
&AEP RPT.PrintStr(79, NumberToString("%6.2", &gbl_tax), &PrintAtr_detail_right |
",csv=10");
&AEP RPT.PrintStr(91, NumberToString("%6.2", &gbl_net), &PrintAtr_detail_right |
",csv=11");

...

...

End-Function;
```

By default, Reporting System only produces PDF file. Following line of code will be required immediately after InitReport() method to produce CSV file also:

```
If &my_rpt.InitReport(&ArgList) = True Then
        &my_rpt.Csv = True;
        …
        …
End-If;
```

Rest all is taken care by AePlus Reporting System that produces additional file aep_my_frst_pdf_rpt.csv. Whenever, a column value has delimiter (comma) in it, the value is enclosed in double quote. For example name: Last Name, CFA0065 will stay in one column when file is opened using excel.

Working with Page Size and Page Orientation:

AePlus Reporting System defaults to following Page Attributes:

Page Size:	A4
Page Width:	595.3 points
Page Height:	841.9 points
Page Orientation:	Portrait
Font Name:	Courier
Font Size:	8
Left Margin:	25 points
Top Margin:	25 points
Right Margin:	25 points
Bottom Margin:	25 points
Page Footer Size:	25 Points

The print area is determined by subtracting margins and footer size from the physical page size. This implies that there is no special allocation for Page Header and can vary depending on how developer uses it. These defaults can be overridden either initially in InitReport() method or later using NewPage() method.

A page in the report can be a mix of any page sizes. For example, you can start the report with page size A4 - Portrait and then change it to A4 - Landscape and then to A3 - Portrait and back to A4 - Portrait.

AePlus reporting system automatically adjusts attributes of the page whenever page is altered so that print contents on the page can be adjusted using current attributes of the page thru PeopleCode.

The default font for the page is always Courier (fixed character size). Based on font size, System calculates number of standard lines that can be printed on the page. It also calculates number of standard characters that a print line can accommodate. Following lines are shown in the process log file (aep_rpt.log) that system generates:

```
Page Font Size:          8
Char Width:              5.904 Points
Char Height:             8.440 Points
Standard Chars in Line:  92
Standard Lines in Page:  86
Page Footer Size:        25 Points
Print Area:              Top Left Corner of page: (25,816) Point,
                         Bottom Right Corner of the page: (570,25) Point,
                         Point to Inch/MM conversion formula: 1 Point = 1/72 inch,
                         mm = (Point * 25.4) / 72
```

Working with Fonts, Sizes and Colours:

AePlus Reporting System supports following standard PDF Fonts.

1. Courier
2. Helvetica
3. Times-Roman
4. Symbol
5. ZapfDingbats

Above fonts along with their variation (Bold and/or Italic) are automatically available for use.

In addition, TrueType font can also be used but these must be loaded first. For example, if you want to use Arial.ttf font, following PeopleCode will be required immediately after InitReport() method. The font is also embedded in the pdf document.

```
If &my_report.InitReport(&InitArgs) = true then
&Status = &my_report.LoadCustFont("C:\Windows\Fonts\Arial.ttf");
    ...
    ...
    While &SQL.Fetch(&Rec)
&my_report.SubmitData(&Rec,...)
        ...
        ...
End-while
&my_report.FinalizeReport("", &Rec,...)
End-if;
```

AePlus Reporting System supports virtually every colour. Following is the list Standard Colours that can be used by name:

1. Black
2. Blue
3. Cyan
4. DarkGrey
5. Grey
6. Green
7. LightGray
8. Magenta
9. Orange
10. Pink
11. Red
12. White
13. Yellow

In addition, colours can be specified by their integer value or by their RGB value. Following three lines of PeopleCode would produce identical results.

```
&AEP_RPT.PrintStr(1, "*** End of Report ***", "color=red");
&AEP_RPT.PrintStr(1, "*** End of Report ***", "color=16711680");
&AEP_RPT.PrintStr(1, "*** End of Report ***", "color=rgb(255,0,0)");
```

Summary:

Developing PDF reports using AePlus Reporting System is very simple and fast. Let's summarise the steps:

1. Create Record Definition with the name as Report Name given to it using App Designer.
2. Generate FuncLib and App. Pkg. PeopleCode using AePlus Reporting System Workbench.
3. Edit and complete FuncLib PeopleCode Report Event Functions using App Designer.
4. Create App Engine to initiate Report Application Class and submit row of data received from **Input Data Stream** to AePlus Reporting System.
5. Run App Engine to generate PDF report.