



AePlus
Reporting System
(Printing Invoice
Using PeopleCode)

SRI Technologies Pty Ltd
Web Site: www.sritech.biz
Email: sritech@sritech.biz

Table Of Contents

AEPLUS REPORTING SYSTEM.....	3
PRINTING INVOICE USING PEOPLECODE	3
<i>Pre-requisites:</i>	3
<i>Requirement:</i>	3
FIELDS AND RECORDS CREATION:.....	4
<i>Report Name:</i>	4
<i>Fields Creation:</i>	4
<i>Records Creation:</i>	4
PEOPLECODE TEMPLATE CREATION:	5
PEOPLECODE DEVELOPMENT/MAINTENANCE (FUNCLIB):.....	6
<i>Function rpt_begin (...):</i>	6
<i>Function rpt_page_header(...):</i>	6
<i>Function rpt_page_footer(...):</i>	6
<i>Function rpt_skip(...):</i>	6
<i>Function rpt_detail_section(...):</i>	6
<i>Function rpt_init_variables(...):</i>	9
<i>Function rpt_end(...):</i>	9
<i>Miscellaneous code in FuncLib(...):</i>	9
PEOPLECODE DEVELOPMENT/MAINTENANCE (APPENGINE):.....	10
RUNNING THE REPORT:	10

AePlus Reporting System

Printing Invoice using PeopleCode

In this tutorial, the purpose is to demonstrate what is involved when developer wants to print Invoice using AePlus Reporting System. Printing Payslip will be very similar to what we do here.

Pre-requisites:

The developer should go thru my_first_pdf_report_using_peoplecode tutorial document.

Requirement:

The Requirement is to produce a PDF version of the Invoice:

1. The Invoice and the related data are already available in various tables in the system.
2. Generate invoice for following Invoice Numbers:
 - a. INV-0001, INV-0002, INV-0003, INV-0004, INV-0005
3. Name of the PDF file generated should have invoice number in it.
4. The Invoice will be single page and has the layout.

INVOICE			
ABC Comporation - INV-0001			
12 John Place - INV-0001 Sydney Australia	Date: Invoice#: Customer ID:	12-Mar-2015 INV-0001 CUSTINV-0001	
BILL TO			
Contact: John Smith - INV-0001			
XYZ Corporation - INV-0001 23 New Building - INV-0001 City Circle Canberra - Australia		Due Date: Net Amount:	12-Apr-2015 5153930.00
Description	Unit Price	Qty	Amount
Item INV-0001:1	100.00	3.00	300.00
Item INV-0001:2	300.00	7.00	2100.00
Item INV-0001:3	1000.00	12.00	12000.00
Item INV-0001:4	4100.00	18.00	73800.00
Item INV-0001:5	20600.00	25.00	515000.00
Item INV-0001:6	123700.00	33.00	4082100.00
Other Comments INV-0001: We have not received any purchase order and items have been delivered.		Sub Total: Tax: Other: Total:	4685300.00 468530.00 100.00 5153930.00

5. The developer may use relative or direct printing on the page as necessary information required may not be available if page is filled sequentially line by line.

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

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_INVOICE_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:

Assuming that following fields are already in the system (created while my_first_pdf_report_using_peoplecode tutorial), we will not be creating any additional fields.

Field Name	Long Name	Short Name	Field Type	Size
AEP_RPT_EVENTS	Aep Report Events	Rpt Events	Char	1
AEP_KEY_DETAIL	Key Detail	Key Detail	Char	30

Records Creation:

Create AEP_INVOICE_PDF(*rpt_aet_rec*) record with Record Type as Derived/Work. Please add following fields to the record. We do not have any on-break events to process so first field should be AEP_RPT_EVENTS. We are adding AEP_KEY_DETAIL to this as we do not intend to create *rpt_dat_rec* record. i.e. In this case, *rpt_aet_rec* and *rpt_dat_rec* records will be same.

Field Name
AEP_RPT_EVENTS
AEP_KEY_DETAIL

For sake consistency, you may give report name a description. However this is not going to appear on the Invoice.

That's all about fields and records creations. Now is the time to get into PeopleCode stuff. We will generate PeopleCode Template using AePlus Workbench.

PeopleCode Template Creation:

AePlus workbench will create various *FieldFormula* PeopleCode functions in AEP_INVOICE_PDF.AEP_RPT_EVENTS. The record definition will look like:

Record Fields		Record Type								
Num	Field Name	Type	FDe	FEd	FCh	FFo	RIn	RIe	RDe	
1	AEP_RPT_EVENTS	Char				✓				
2	AEP_KEY_DETAIL	Char								

The AePlus workbench creates following dummy functions for given *rpt_aet_rec* Record (i.e. for AEP_INVOICE_PDF record).

Report Events: rpt_begin(...),rpt_end(...)
Page Events: rpt_page_header(...),rpt_page_footer(...)
Data Events: rpt_init_variables, rpt_skip, rpt_detail_section

In addition it also generates AEP_INVOICE_PDF.AEP_INVOICE_PDF Application Class that drives AePlus Reporting System. 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.

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.

Function rpt_page_header(...):

We are generating a single page invoice and will have nothing in header to print. We will leave this as blank.

Function rpt_page_footer(...):

We are generating a single page invoice and will have nothing in footer to print. We will leave this as blank.

Function rpt_skip(...):

We are not doing any consolidation/aggregation, so this function should always return false value as we do not want to skip any data that we receive from **Input Data Stream** that has been submitted to the AePlus Reporting system using &my_rpt.SubmitData() method.

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

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

    Local boolean&RtnSts = false;

    Return &RtnSts;

End-Function;
```

Function rpt_detail_section(...):

This function is called by the AePlus Reporting System to print each invoice. Here we will put the PeopleCode to print entire invoice for given Invoice Number based on the layout given.

We will use mix of line by line and absolute printing on the page using various Print Attributes for given text of information.

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

    Local number &xPoint, &yPoint, &xPoint2, &yPoint2, &TotalAmt_save_X,
&TotalAmt save Y;
    Local number &i, &UnitPrice, &Units, &Amt, &TotalAmt, &TotalBeginLine, &tax,
&other, &lines, &invnum;;
    Local string &InvoiceNumber;

    &InvoiceNumber = &Rec.GetField(Field.AEP_KEY_DETAIL).Value;
    &invnum = Value(Right(&InvoiceNumber, 4));

    rem force new page with new file;
    &AEP_RPT.NewPage(&Rec, "newfile=");
    rem override default filename;
    &AEP_RPT.SetPdfFileName(&InvoiceNumber);

    rem draw rectangle around the page;
    &AEP_RPT.DrawBox(&AEP_RPT.get PageTopLeft x(), &AEP_RPT.get PageTopLeft_y(),
&AEP_RPT.get PageBottomRight x(), &AEP_RPT.get PageBottomRight_y(),
"lineThickness=1,color=red");
```

```

&AEP RPT.NewLine(1, &Rec);
&AEP RPT.NewLine(1, &Rec);
&AEP RPT.NewLine(1, &Rec);
&AEP RPT.NewLine(1, &Rec);
&AEP RPT.NewLine(1, &Rec);
&AEP_RPT.PrintStr(60, "INVOICE", "FontName=Helvetica,FontSize=40,Bold,Color=blue");

&AEP RPT.NewLine(1, &Rec);
&AEP RPT.PrintStr(1, "ABC Comporation - " | &InvoiceNumber, &PrintAtr_Bold20 |
",Color=magenta");

&AEP RPT.NewLine(1, &Rec);
&AEP RPT.PrintStr(1, "12 John Place - " | &InvoiceNumber, &PrintAtr_Bold16_Blue);
&AEP RPT.PrintStr(45, "Date:", &PrintAtr_Bold16_Blue);
&AEP_RPT.PrintStr(&AmtPos, "12-Mar-2015", &PrintAtr_16 | ",right");

&AEP RPT.NewLine(1, &Rec);
&AEP RPT.PrintStr(1, "Sydney", &PrintAtr_Bold16_Blue);
&AEP RPT.PrintStr(45, "Invoice#:", &PrintAtr_Bold16_Blue);
&AEP_RPT.PrintStr(&AmtPos, &InvoiceNumber, &PrintAtr_16 | ",right");

&AEP RPT.NewLine(1, &Rec);
&AEP RPT.PrintStr(1, "Australia", &PrintAtr_Bold16_Blue);
&AEP RPT.PrintStr(45, "Customer ID:", &PrintAtr_Bold16_Blue);
&AEP_RPT.PrintStr(&AmtPos, "CUST" | &InvoiceNumber, &PrintAtr_16 | ",right");

&AEP RPT.NewLine(1, &Rec);
&AEP RPT.NewLine(1, &Rec);
&AEP RPT.NewLine(1, &Rec);
&xPoint = &AEP RPT.get CurrLeftMargin() + 10;
&yPoint = &AEP RPT.get CurrPointY();
&xPoint2 = &xPoint + 350;
&yPoint2 = &yPoint - 30;
&AEP RPT.DrawBox(&xPoint, &yPoint, &xPoint2, &yPoint2,
"lineThickness=0,fillcolor=lightGray");
&AEP RPT.AdjustCurrPointY((&AEP RPT.get CharHeight() * 2.5));
&AEP_RPT.PrintStr(2, "BILL TO", &PrintAtr_Bold20);

&AEP RPT.AdjustCurrPointY((&AEP_RPT.get_CharHeight() * 2.0));
&AEP RPT.NewLine(1, &Rec);
&AEP_RPT.PrintStr(2, "Contact: John Smith - " | &InvoiceNumber, &PrintAtr_Bold20);

&AEP RPT.NewLine(1, &Rec);
&AEP_RPT.PrintStr(2, "XYZ Corporation - " | &InvoiceNumber, &PrintAtr_16);

&AEP RPT.NewLine(1, &Rec);
&AEP_RPT.PrintStr(2, "23 New Building - " | &InvoiceNumber, &PrintAtr_16);

&AEP RPT.NewLine(1, &Rec);
&AEP RPT.PrintStr(2, "City Circle", &PrintAtr_16);
&AEP RPT.PrintStr(45, "Due Date:", &PrintAtr_Bold16_Blue);
&AEP_RPT.PrintStr(&AmtPos, "12-Apr-2015", &PrintAtr_16 | ",Color=red,bold,right");

&AEP RPT.NewLine(1, &Rec);
&AEP RPT.PrintStr(2, "Canberra - Australia", &PrintAtr_16);
&AEP RPT.PrintStr(45, "Net Amount:", &PrintAtr_16 | ",Bold,Color=blue");
rem save current x,y to print the total later;
&TotalAmt save X = &AEP RPT.ColPositionToPoint_X(&AmtPos);
&TotalAmt_save_Y = &AEP RPT.get_CurrPointY();

rem print header line surrounded in the box;
&AEP RPT.NewLine(1, &Rec);
&xPoint = &AEP RPT.get CurrLeftMargin() + 10;
&yPoint = &AEP RPT.get CurrPointY();
&xPoint2 = &AEP RPT.get PageBottomRight_x() - 10;
&yPoint2 = &yPoint - 30;
&AEP RPT.DrawBox(&xPoint, &yPoint, &xPoint2, &yPoint2,
"lineThickness=0,fillcolor=lightGray");
&AEP RPT.AdjustCurrPointY((&AEP RPT.get CharHeight() * 2.5));
&AEP RPT.PrintStr(2, "Description", &PrintAtr_Bold20);
&AEP RPT.PrintStr(39, "Unit Price", &PrintAtr_Bold20);
&AEP RPT.PrintStr(60, "Qty", &PrintAtr_Bold20);
&AEP_RPT.PrintStr(75, "Amount", &PrintAtr_Bold20);

rem print items;
&AEP RPT.AdjustCurrPointY((&AEP_RPT.get_CharHeight() * 1.0));
&TotalAmt = 0;
  For &i = 1 To&invnum + 5
&AEP_RPT.NewLine(1, &Rec);

&UnitPrice = &UnitPrice * &i + 100;
&Units = &Units + 2 + &i;
&Amt = &UnitPrice * &Units;
&TotalAmt = &TotalAmt + &Amt;
&AEP RPT.PrintStr(2, "Item " | &InvoiceNumber | ":" | &i, &PrintAtr_16);
&AEP RPT.PrintStr(&UnitPos, NumberToString("%6.2", &UnitPrice), &PrintAtr_16 |
",right");

```

```

&AEP RPT.PrintStr(&QtyPos, NumberToString("%6.2", &Units), &PrintAtr_16 | ",right");
&AEP RPT.PrintStr(&AmtPos, NumberToString("%6.2", &Amt), &PrintAtr_16 | ",right");
End-For;

rem position to print invoice totals;
&TotalBeginLine = 37;
&AEP_RPT.SetCurrPointY_AtLine(&TotalBeginLine);

&AEP RPT.NewLine(1, &Rec);
&AEP RPT.PrintStr(&UnitPos, "Sub Total: ", &PrintAtr_16 | ",Bold,Color=blue");
&AEP RPT.PrintStr(&AmtPos, NumberToString("%6.2", &TotalAmt), &PrintAtr_16 |
",right");

&AEP RPT.NewLine(1, &Rec);
&AEP RPT.PrintStr(&UnitPos, "Tax: ", &PrintAtr_16 | ",Bold,Color=blue");
&tax = &TotalAmt * 0.1;
&AEP_RPT.PrintStr(&AmtPos, NumberToString("%6.2", &tax), &PrintAtr_16 | ",right");

&AEP RPT.NewLine(1, &Rec);
&AEP RPT.PrintStr(&UnitPos, "Other: ", &PrintAtr_16 | ",Bold,Color=blue");
&other = 100;
&AEP_RPT.PrintStr(&AmtPos, NumberToString("%6.2", &other), &PrintAtr_16 | ",right");

&AEP RPT.NewLine(1, &Rec);
&AEP RPT.PrintStr(&UnitPos, "Total: ", &PrintAtr_16 | ",Bold,Color=blue");
&TotalAmt = &TotalAmt + &tax + &other;
&AEP RPT.PrintStr(&AmtPos, NumberToString("%6.2", &TotalAmt), &PrintAtr_16 |
",right,color=red,bold");
&AEP RPT.PrintStrAbs(&TotalAmt save_X, &TotalAmt save_Y, NumberToString("%6.2",
&TotalAmt), &PrintAtr_16 | ",Color=red,right,bold");

rem print comments;
&AEP RPT.SetCurrPointY_AtLine(&TotalBeginLine);
&xPoint = &AEP RPT.get CurrLeftMargin() + 10;
&yPoint = &AEP RPT.get CurrPointY();
&xPoint2 = &xPoint + 255;
&yPoint2 = &yPoint - 30;
&AEP RPT.DrawBox(&xPoint, &yPoint, &xPoint2, &yPoint2,
"lineThickness=0,fillcolor=lightGray");
&AEP RPT.AdjustCurrPointY((&AEP RPT.get CharHeight() * 2.5));
&AEP RPT.PrintStr(2, "Other Comments", &PrintAtr_Bold20);
&AEP RPT.NewLine(1, &Rec);
&lines = &AEP RPT.PrintStrWrap( False, &Rec, 2, 33, &InvoiceNumber | ": We have not
received any purchase order and items have been delivered.", &PrintAtr_16, "-");

End-Function;

```


Function rpt_init_variables(...):

Since we are not doing any consolidation, we can leave this function as blank.

Function rpt_end(...):

We are leaving this function as blank. But you may want to print some summary information here.

That's the end of the report event functions.

Miscellaneous code in FuncLib(...):

We have created some constants to hold Print Attributes and Column Positions so that these can be used to minimise code maintenance effort that could rise in the future:

```
rem Formatting constants;
Constant &PrintAtr Bold20 = "FontName=Helvetica,FontSize=20,Bold";
Constant &PrintAtr Bold16 Blue = "FontName=Helvetica,FontSize=16,Bold,color=blue";
Constant &PrintAtr_16 = "FontName=Helvetica,FontSize=16";

rem column position constants;
Constant &UnitPos = 55;
Constant &QtyPos = 65;
Constant &AmtPos = 88;
```

PeopleCode Development/Maintenance (AppEngine):

That's all about putting code against various 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 INVOICE_PDF. Create a PeopleCode step and add following code:

```
import AEP_INVOICE_PDF:*;

Local AEP_INVOICE_PDF:AEP_INVOICE_PDF &my_rpt;
Local Record &Rec;
Local number &i;
Local string &InstanceArgList, &RptAetRecName, &RptDatRecName, &InitArgList,
&AbsFolder;
Local string &sql_str;

&sql_str = "Select PRCSOUTPUTDIR from PS CDM LIST where PRCSINSTANCE = :1";
SQLExec(&sql_str, AEP_RC_AET.PROCESSINSTANCE, &AbsFolder);
If All(&AbsFolder) Then
&AbsFolder = &AbsFolder | Left(&AbsFolder, 1)
End-If;
&InstanceArgList = "abspath=" | &AbsFolder;
rem &InstanceArgList = &InstanceArgList | ",pi=" | AEP_RC_AET.PROCESSINSTANCE;
&my_rpt = create AEP_INVOICE_PDF:AEP_INVOICE_PDF(&InstanceArgList);

rem build argument list;
&RptAetRecName = "AEP INVOICE PDF";
&RptDatRecName = &RptAetRecName;
&InitArgList = "rpt aet rec=" | &RptAetRecName | ",";
&InitArgList = &InitArgList | "rpt_dat_rec=" | &RptDatRecName | ",";

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

&Rec = CreateRecord(Record.AEP_INVOICE_PDF);

For &i = 1 To 5
&Rec.GetField(Field.AEP_KEY_DETAIL).Value = "INV-000" | &i;
&my_rpt.SubmitData(&Rec, &Rec.GetField(Field.AEP_KEY_DETAIL).Value);
End-For;

&my_rpt.FinalizeReport("", &Rec);

End-If;

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

This AppEngine code can also be found in AEP_EXM project that can be loaded using AppDesigner from sritech\aeplus folder.

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, following files will be created:

1. Pdf Reports:
 - a. inv-0001.pdf
 - b. inv-0002.pdf
 - c. inv-0003.pdf
 - d. inv-0004.pdf
 - e. inv-0005.pdf
2. Process Log File: aep_rpt*.log

