

Reporting Template

By Denis Fafard
Business Analyst
WCB - Alberta

Problem

- Reports built to different standards
 - Hard to maintain
 - Hard to support unless you're the author
- Time Consuming to login and access the same databases all the time
- New BAs do not have a standard to follow
 - Difficult to learn all others' styles
 - Tough to remember all database names (30+)

Solution

- Set up a single standard for coding reports
- Put all the repeated database login code into a single SAS file.
- Abstract out the need to remember specific database names.



* Report Title: eSUG Sample Report
* Requested by: Matt Malczewski
* Purpose/Desc: Sample report that really does nothing but show the logon procedure to access data. SAS report below is strictly a sample and is probably filled with syntax errors.
*
* Criteria: Acct registered not prior to current year,
* Credit Balance greater than \$100.
* Account must be open.
*

* Schedule: Ad-Hoc
* Author: Denis Fafard Date: Nov 6, 2013
*

* Modifications
* CCYY-MM-DD: ZZ %1%

```
;  
%INCLUDE "\\ReportServer\BA\Denis\PathNames.sas";  
%INCLUDE "&InputPath\MTAnyOBS.SAS";  
%LET ENVIRON=PROD; %INCLUDE "&ODBCPath\ODBC_IFMS.SAS";  
%LET ENVIRON=PROD; %INCLUDE "&ODBCPath\ODBC_AIMS.SAS";
```

*Set up all pathnames;
*Utility for checking empty data;
*PS Financials Database;
*Acct Registration Database;

```
Data _Null_;  
* Set up variables to be used for this report;  
RegDT = INTNX('YEAR',Today(),0,'B'); *Get the beginning of the year;  
OutFile = "&TempPath\Credit_Acct.HTM"; *Report output file;  
*OutFile = "&ReportPath\Credit_Acct.HTM";  
Bal_Limit = -100; *Minimum Credit Amt;
```

```
Call SymputX('RegDT',RegDT);  
Call Symputx('OutFile',OutFile);  
Call Symputx('Bal_Limit',Bal_Limit);  
RUN;
```

* Go find all the accounts for the given year with credit balances;

```
Proc SQL;  
Create Table Credits as  
Select Cust_ID, Bal_amt  
From IFMS.PS_Cust_Data  
Where Bal_Amt LT &Bal_Limit;  
QUIT;
```

... CONTINUED FROM PREVIOUS SLIDE ...

* Go Find all the open accounts. This could be combined with above SQL;

Proc SQL;

Create Table Open_Acct as

Select Cust_ID, Start_DT

From AIMS.Employer1

Where Acct_Sts = 'OPEN'

AND Start_DT GE RegDT;

QUIT;

Proc Sort Data=Credits NOEQUALS; By Cust_ID; **RUN;**

Proc Sort Data=Open_Acct NOEQUALS; By Cust_ID; **RUN;**

* Merge the data into the final table for output;

Data Final;

Merge Open_Acct (IN=Opn Drop=Start_DT)

Credits (IN=CR);

By Cust_ID;

If Opn AND CR;

RUN;

*Put the results out into a simple HTML file;

ODS HTML BODY = "&OutFile";

Proc Print Data=Final NOOBS Label Split='/';

Label Cust_ID = 'Account/Number' Bal_Amt = 'Balance/Amount';

Format Bal_Amt Dollar15.2 ;

Title 'Credits on Open Accounts';

FOOTNOTE1 j=center color=blue height=10pt

"Data run as of &SYSDATE9 at &SYSTIME";

RUN;

ods _all_ close;

/*****

*** THIS IS THE END. ***

*****/

PATHNAMES.SAS

```
/*-----*/
/*      SAS Macro Definition for Paths to Folders      */
/*      */
/* This file should be included into the SAS program via */
/* the %include statement.  For example:                */
/*      */
/* %include "PathNames.sas";                            */
/*      */
/* Any libname, infile, file, or filename statements can */
/* reference this path using the syntax:                */
/*      */
/* libname xxxx "&temppath\subdirxxx";                  */
/* filename xxxx "&temppath\yyyyyyyy";                */
/*      */
/* Local version for Denis Fafard.                      */
/* DO NOT COPY THIS PROGRAM INTO THE PRODUCTION DIRECTORY. */
/*-----*/
%global TempPath;
%let TempPath = C:\Users\Denis\Downloads;

/* Path to MS Excel */
%global MSExcel;
%let MSExcel = C:\Program Files\Microsoft Office\OFFICE11\Excel.exe;

/* Path for all ODBC drivers & security type files */
%global ODBCPath;
%let ODBCPath = \\ReportServer\BA\Denis\ODBC;

/* Standard Output path for public reports */
%global ReportPath;
%let ReportPath = \\ReportServer\BA\Reports;

/* For all input programs. */
%global InputPath;
%let InputPath = \\ReportServer\BA\Denis;
```

ODBC_IFMS.SAS

```
/*----- **
**          GENERIC ODBC DRIVER FOR PS/IFMS          **
**          **                                       **
** This file should be included in your main SAS file in a line such as **
**   %INCLUDE "&ODBCPath\ODBC_IFMS.SAS"           **
**          **                                       **
** This will set up an "IFMS" libname to link to IFMS in either production, **
** pre-production (not all tables available) or development, based on the **
** global macro variable "ENVIRON".                **
**          **                                       **
** REQUIRED: The following variable must be set in the calling program: **
**   %LET ENVIRON=xxxxxx;                          **
**          **                                       **
** Where xxxxxx = PROD, PREPROD, TRAINING or any development IFMS schema **
**   (e.g. QTV1, QTU9, QTI2, ...)                  **
**          **                                       **
** To use this libname, call "IFMS" inside PROC SQL. E.g., **
**   SELECT *                                       **
**   FROM IFMS.PS_DUN_CUST                          **
**          **                                       **
**----- **
** If this ODBC driver uses RACF security set RACF=1 **
** If this ODBC driver uses Windows security set RACF=0 **
**----- */
%GLOBAL RACF ; %LET RACF = 1 ;
```

```
%INCLUDE "&ODBCPath\SECURITY.SAS" ;
```

```
DATA _NULL_ ;
  ENVIRON=UpCase (SYMGET ('ENVIRON')) ;
  IF ENVIRON='PROD' THEN DO ;
    CALL SYMPUTX ('SCHEMA', 'QTP1') ;
    CALL SYMPUTX ('DATASRC', 'IFMSP') ;
  END ;
  ELSE IF ENVIRON='PREPROD' THEN DO ;
    CALL SYMPUTX ('SCHEMA', 'QTR1') ;
    CALL SYMPUTX ('DATASRC', 'IFMSR') ;
  END ;
  ELSE IF ENVIRON='TRAINING' THEN DO ;
    CALL SYMPUTX ('SCHEMA', 'QTE1') ;
    CALL SYMPUTX ('DATASRC', 'IFMSE') ;
  END ;
  ELSE DO ;
    CALL SYMPUTX ('SCHEMA', ENVIRON) ;
    CALL SYMPUTX ('DATASRC', 'IFMSU') ;
  END ;
RUN ;
```

ODBC_IFMS.SAS

... CONTINUED FROM PREVIOUS SLIDE ...

```
LIBNAME IFMS ODBC DBPROMPT=&DBPROMPT DEFER=NO
```

```
USER=&USER PASSWORD=&PASSWORD DATASRC=&DATASRC SCHEMA=&SCHEMA;
```

```
Data _Null_;
```

```
/* Check to see if the LIBNAME function was successful. Password may have expired */
```

```
/* or even the ODBC call might have failed. Denis Fafard May 23, 2008 */
```

```
If &syslibrc NE 0 Then
```

```
    Do x = 1 to 10;
```

```
        Put 'ODBC Error - Check your password or ODBC driver.';
```

```
        Put "User = <&User> DataSrc = <&DATASRC> Schema = <&SCHEMA>";
```

```
        Put ' ';
```

```
    end;
```

```
/* Uncomment next line if you prefer SAS to shutdown if LIBNAME fails.
```

```
Abort Return */
```

```
RUN;
```


SECURITY.SAS

```
/*-----*/
/*          SAS security parameters          */
/*                                          */
/* This file should be included into the SAS program via */
/* the %include statement.  For example:      */
/*                                          */
/*  %include "SECURITY.sas";                */
/*                                          */
/* Parameters:                               */
/*  user          - Mainframe RACF user id    */
/*  dbprompt      - prompting, YES or NO     */
/*  password      - Mainframe RACF & Windows passwords */
/*                                          */
/* <<<<<<This should be a hidden & secure file>>>>>>>> */
/*  <<<<<<<<<< BUAHHHAAAHA AAAHAAAA >>>>>>>>>>>>>>>> */
/*-----*/
```

```
%GLOBAL USER DBPROMPT PASSWORD ;
```

```
%LET USER = Denis9814 ;
```

```
%LET DBPROMPT = NO ;
```

```
Data _NULL_ ;
```

```
Length PASSWORD $ 9 ;          *Sets the max. length;
```

```
If &RACF = 0 Then PASSWORD = "WindowPWD" ;
```

```
If &RACF = 1 Then PASSWORD = "OtherPWD" ;
```

```
Call Symput('PASSWORD',PASSWORD);
```

```
RUN;
```

Overall Flow

