

The Other Side of SAS:

Besides Research,
What's it Good For?

Dawn Campbell

February 11, 2005

Introduction

- ◆ UofM Administrative Systems
- ◆ "Systems Renewal"
- ◆ Previously Academic Computing
Consultant/Administrator/Programmer
- ◆ SAS user since 1982.

SAS in Manitoba

- ◆ User base reflects Manitoba's strong research community.
- ◆ SAS is generally perceived as being a academic/research tool.
- ◆ It is that and more.

SAS in Systems Renewal

- ◆ Program code translator
- ◆ Legacy data explorer
- ◆ Multi-source data integrator
- ◆ Interface Facilitator

Program Code Translator

- ◆ Problem: We need to produce many database loader scripts.
 - Moderately complex, very repetitive.
 - Prone to clerical error.
 - Based on existing database segment descriptors.

Original Sample (partial)

```
011526 01  EDUC-REC.
011530      10  FILLER                PIC X VALUE  ''.
011531      10  OUT-EMPLNO            PIC 9(6) .
011534      10  FILLER                PIC XXX VALUE ', '.
011539      10  EDLIN                 PIC 99.
011540      10  FILLER                PIC XXX VALUE ', '.
011541      10  EDUCTYPE              PIC X.
011542      10  FILLER                PIC XXX VALUE ', '.
011550      10  DEG-NAME              PIC X(15) .
011551      10  FILLER                PIC XXX VALUE ', '.
011580      10  DEGCNTRY             PIC XXX.
011581      10  FILLER                PIC XXX VALUE ', '.
011590      10  EDACTDTE             PIC 9(05) .
```

SAS code

Macro source too long for slide –

Available as handout.

```
libname macrolib v8 "S:\directory\Segments\sas\";
options mstored sasstore=macrolib;
* Syntax: %vextract(table,dir=c:\ccm,test=*) ;
%vextract(seducat) ;
```

Result: SQL create

```
create table seducat_t (  
    empl_no          number(6,0) ,  
    edlin            number(2,0) ,  
    eductype         varchar2(1) ,  
    deg_name         varchar2(15) ,  
    compl_yr         varchar2(2) ,  
    institut         varchar2(30) ,  
    degcntry         varchar2(3) ,  
    edactdte         number(5,0) ,  
    compl_yr2        varchar2(4) ,  
    compl_yr8        varchar2(8)  
);
```

Result: SQL loader control

```
REPLACE INTO TABLE seducat_t
FIELDS terminated by "," optionally enclosed by '"' (
  empl_no      "LPAD(:empl_no ,6,'0')",
  edlin        NULLIF edlin=BLANKS
               "TO_NUMBER(:edlin , '99')" ,
  eductype     NULLIF eductype=BLANKS "RTRIM(:eductype)" ,
  deg_name     NULLIF deg_name=BLANKS "RTRIM(:deg_name)" ,
  compl_yr     NULLIF compl_yr=BLANKS "RTRIM(:compl_yr)" ,
  institut     NULLIF institut=BLANKS "RTRIM(:institut)" ,
  degcntry     NULLIF degcntry=BLANKS "RTRIM(:degcntry)" ,
  edactdte     NULLIF edactdte=BLANKS
               "TO_NUMBER(:edactdte , '99999')" ,
  compl_yr2    NULLIF compl_yr2=BLANKS "RTRIM(:compl_yr2)" ,
  compl_yr8    NULLIF compl_yr8=BLANKS "RTRIM(:compl_yr8)" )
```

Result: SAS import (part 1)

```
%let segment=seducat;

libname saslib "c:\ccm\SASLIB";

filename &segment. "r:\directory\&segment..dat" lrecl=1000;

proc import out=saslib.&segment.

    datafile=&segment.      dbms=dlm      replace;

    delimiter=',';

    getnames=no;

    datarow=1;

quit;

data saslib.&segment.;

    length EmployeeNumber 4;

    set saslib.&segment.;

    EmployeeNumber=input(var1,6.0);

run;
```

Result: SAS import (part 2)

```
proc datasets library=saslib nolist;
  modify &segment.;
  rename      var1=empl_no
             var2=edlin
             var3=eductype
             var4=deg_name
             var5=compl_yr
             var6=institut
             var7=degcntry
             var8=edactdte
             var9=compl_yr2
             var10=compl_yr8      ;
  index create EmployeeNumber / updatecentiles=never;
quit;
```

Legacy Data Explorer

◆ Problems:

- Database content begins in 1970s.
- Edit checks were not all in place on day 1.
- Definitions have changed – data was not always changed to match.

Legacy Data Explorer

- ◆ Solution - “traditional” SAS procs:
 - PROC IMPORT to read in all data as character from delimited text files.
 - PROC FREQ to find range of content and scale of problems
 - Biggest problems go to data extract programmers to be filtered at source.

Legacy Data Explorer Sample

```
libname saslib "c:\ccm\SASLIB";

%macro studyfield(segment,field);
title "Values in segment: &segment. Field: &field.";
proc freq data=saslib.&segment.;
    tables &field. / missing nocum nopercnt;
run;
title;
%mend studyfield;

%studyfield(seducat,eductype);
%studyfield(seducat,deg_name);
```

Data Integrator



Problem:

- Data needed for system conversion come from various data sources:
 1. Oracle databases
 2. Sybase databases
 3. Excel Spreadsheets

Data Integrator

◆ Solutions:

- PROC IMPORT
- LIBNAMEs to access SAS Native Engines for Oracle and Sybase.
- Combine data using PROC SQL or DATA STEP merges.
- Use LIBNAME or PROC EXPORT to copy results to required platform.

Data Integrator Samples

```
libname test1 oracle
```

```
user='username' password='password'
```

```
path='databasename' ;
```

```
libname test2 sybase
```

```
user='username' password='password'
```

```
database='databasename' server='servername'
```

```
schema='dbo' ;
```

Interface Facilitator

- ◆ Our most technically complex use of SAS in the HR renewal project.
- ◆ Required special measures to meet production jobs standards.
- ◆ Uses Oracle database pass-through facility to call compiled Oracle procedures from inside SAS programs.

Cautions

- ◆ Understand how the other database handles dates and nulls.
- ◆ You may need to use `datepart()` in comparisons.
- ◆ A database null is not exactly a blank, and not exactly a SAS missing value.
- ◆ If a data type in a database is neither numeric or character (e.g. BLOB) SAS may not be able to handle it at all.

Interface Facilitator

- ◆ Could have been accomplished in PERL, if
 - we'd had the required tools
 - we'd had the expertise
 - we'd had the time
- ◆ We didn't.

Interface Facilitator

- ◆ Requirement – simplest case.
 - Unix CRON task calls shell script
 - Shell script calls PERL script
 - PERL script calls SAS
 - SAS determines target database environment, etc.
 - SAS reads Sybase tables

Interface Facilitator (cont.)

- SAS subsets, merges, and transforms Sybase data
- SAS copies data to Oracle low-security staging tables
- SAS calls Oracle procedure to copy data from low-security environment to high-security environment
- SAS logs all actions and errors

Conclusion

- ◆ Even after many years of SAS use, I still regularly find new ways to use the software.
- ◆ I wish I had the time to learn more of the new tools SAS keeps adding to their product.

Questions?



Contact: Dawn_Campbell@Umanitoba.ca