

Paper 097-30

Budget Living with Pseudo ETL Creating your own ETL tool through SAS®

David Steves, SunTrust Bank, Atlanta, Georgia
Denise Figliozzi, SunTrust Bank, Atlanta, Georgia

ABSTRACT

In today's world data processing needs to run faster, better, and more efficiently. The push for the latest data is evermore important and can be facilitated through the use of an ETL tool. The acronym ETL stands for extraction, transformation, and loading. Companies evaluate different ETL tools to purchase. But, is it truly necessary? Your company may already have some of the basic SAS components that you can use as an ETL tool.

This paper will go over three components of the ETL (Extraction, Transformation, and Loading) and illustrate which SAS components can be utilized.

INTRODUCTION

Everyday company executives wake up to make business decisions based on the most current available data. This data will need to be processed and synthesized to make sound business decisions. One of the many tools used to collect this type of information is called an ETL. Currently, many software companies offer ETL tools, but, is it an absolute necessity to purchase one? Companies have become more cost conscious over the years and tend to procrastinate on expensive purchases. Well, your company may already have the SAS components necessary to create an ETL.

In the olden days, we did not have any data processing called ETL. The creation of synthesized data was called data processing. These ETL tools may be faster and collectively better, but, how fast is fast? Savvy SAS programmers can generate quick results and can emulate an ETL tool. In this paper, we will illustrate the extraction process, the transformation process, and the loading process while detailing the associated SAS components utilized.

EXTRACTION

Extraction for our purposes means: the moving of data from location A to location B. Many years ago we referred to the extraction process as "pulling data." We can easily pull data from a UNIX server to another UNIX server with following example:

(this example is for an existing SAS dataset that is ready to be extracted)

```
# -----
# Extraction Portion One:  this scenario includes creating the dataset
# that needs to be extracted on UNIX server A
# -----

libname etl '/archive/etl';

**** Creating flat file from SAS dataset.  Needed for FTP  ****
PROC CPORT DATA=etl.xtract  FILE="/etl/outputs/xtract";  run;

*** Compresses the SAS dataset  ****
data _null_;
  call system("compress /etl/outputs/xtract");
run;

*** creates the ftp script*****;
filename ftpcmds "/transfer/ftp_xtract.txt";

data _null_;
  file ftpcmds pad lrecl=180;
  put "user uztamatta pasta2";
  put "binary";
  put "put /etl/outputs/xtract.Z  /compressed/xtract.Z";
run;
```

```

**** This portion of code runs the ftp script that ships a compressed
data set and ftps it over to another server location in a compressed
state.

filename doftp pipe "ftp -n 99.99.999.997 < /transfer/ftp_xtract.txt";

data _null_;
  infile doftp;
  input;
  put _infile_;
run;

# -----
# Extraction Portion Two :The following code takes a compressed dataset
and makes it into as SAS Dataset available for SAS using the CIMPORT
Procedure: (This runs from UNIX Server B)
# -----
libname staging "/etl/staging/data";

data _null_;
call system("uncompress /compressed/xtract.Z");
run;

proc cimport data=staging.xtract FILE="/compressed/xtract";run;

```

The extraction portion utilized both the SAS/BASE and SAS/CONNECT components.

**Note how easily we created this code. Some ETL tools may not be as straightforward.*

TRANSFORMATION

Transformation for our purposes means the data is manipulated into formats that we want in our final results. In the following example of code, we are accessing a SAS dataset and preparing for the Oracle® table load. Each variable is examined and transformed according to certain business rules. The data is then put into a temporary SAS table where in the third step of the process, the SAS table gets loaded into an Oracle table.

```

proc format;
  value $ctoone
    'C' = '1'
    OTHER = '0'
  ;
  value $ytoone
    'Y' = '1'
    OTHER = '0'
  ;
run;

libname staging "/etl/staging/data";

data xtractload (keep=
  CIN_NBR CRF_CIN_FLG FULL_ACCT_NBR HH_NBR INDV_NBR
  MDM_AUDIT_ADD_DT MDM_AUDIT_ADD_USER MDM_AUDIT_MOD_DT
  MDM_AUDIT_MOD_USER
  MDM_CURR_FLG MDM_MTH_PERD_KEY MDM_YRMTH_KEY
  PRIME_OWNER_FLG RCRD_ID TIN_NBR TIN_TY_CD)
  staging.err_xtract(keep=
  CIN_NBR CRF_CIN_FLG FULL_ACCT_NBR HH_NBR INDV_NBR
  MDM_AUDIT_ADD_DT MDM_AUDIT_ADD_USER MDM_AUDIT_MOD_DT
  MDM_AUDIT_MOD_USER
  MDM_CURR_FLG MDM_MTH_PERD_KEY MDM_YRMTH_KEY
  PRIME_OWNER_FLG RCRD_ID TIN_NBR TIN_TY_CD);
length
  CIN_NBR $16. CRF_CIN_FLG $1. FULL_ACCT_NBR $20.
  HH_NBR $13. INDV_NBR $16. MDM_CURR_FLG $1.

```

```

        MDM_MTH_PERD_KEY 6. MDM_YRMTH_KEY 6. PRIME_OWNER_FLG $1.
        RCRD_ID $16. TIN_NBR $9. TIN_TY_CD $1.;
format MDM_AUDIT_ADD_DT MDM_AUDIT_MOD_DT datetime20.
        MDM_MTH_PERD_KEY 6. MDM_YRMTH_KEY 6.;
set staging.xtract;

CIN_NBR = trim(left(cn_numinf));
CRF_CIN_FLG = put(cn_typecode,$ctoone.);
FULL_ACCT_NBR = trim(left(ac_numinf)); /* Has nulls */
INDV_NBR = trim(left(en_numinf));
MDM_AUDIT_ADD_DT = datetime();
if timepart(MDM_AUDIT_ADD_DT) >= '23:59:59.50't then
MDM_AUDIT_ADD_DT = MDM_AUDIT_ADD_DT + 1;
MDM_AUDIT_ADD_USER = "&userid.";
MDM_AUDIT_MOD_DT = datetime();
if timepart(MDM_AUDIT_MOD_DT) >= '23:59:59.50't then
MDM_AUDIT_MOD_DT = MDM_AUDIT_MOD_DT + 1;
MDM_AUDIT_MOD_USER = "&userid.";
MDM_CURR_FLG = 'Y';
MDM_MTH_PERD_KEY = &perd.;
MDM_YRMTH_KEY = &yrmth.;
PRIME_OWNER_FLG = put(cn_primeflg,$ytoone.);
if ac_recordid in ('',' ') then do;
        RCRD_ID = 'CRF00' || trim(left(cn_numinf));
        end;
else
        RCRD_ID = trim(left(ac_recordid));
TIN_NBR = trim(left(cn_tininf)); /* Has nulls */
TIN_TY_CD = trim(left(cn_tincde)); /* Has nulls */
HH_NBR = trim(left(hh_numinf));
if HH_NBR in ('',' ') then output staging.err_xtract;
                                else output xtractload;

run;

```

The transformation portion utilizes the SAS/BASE component.

LOADING

For our purpose, loading is referred to as placing the data in its final destination. In this case it is an Oracle table. Now that the data has been staged for loading, it is time to make the connection to the destination server and push the data over.

```

libname myora oracle user=&userid. password=&userpw. path=&path.;

proc sql;
        insert into myora.xtract (BULKLOAD=YES BL_OPTIONS='ERRORS=0'
        BL_LOG="/etl/log/BL_XTRACT.log"
        BL_CONTROL="/etl/log/BL_XTRACT.ctl")
        select
                MDM_MTH_PERD_KEY,
                RCRD_ID,
                CIN_NBR,
                MDM_YRMTH_KEY,
                MDM_AUDIT_MOD_DT,
                MDM_AUDIT_MOD_USER,
                HH_NBR,
                MDM_CURR_FLG,
                PRIME_OWNER_FLG,
                MDM_AUDIT_ADD_USER,
                MDM_AUDIT_ADD_DT,
                INDV_NBR,
                CRF_CIN_FLG,
                TIN_TY_CD,
                TIN_NBR,
                FULL_ACCT_NBR
        from
                xtractload;

quit;

```

The loading portion utilized the SAS/ACCESS to Oracle component.

CONCLUSION

We have illustrated the SAS components utilized in our own pseudo ETL tool that consist of SAS/BASE, SAS/CONNECT, and SAS/ACCESS to Oracle. SAS has many other components that could make your ETL tool even more efficient. You can save your company money by utilizing all of the SAS components currently available in your version of SAS. In the long run, opt for SAS as your ETL solution; but in the interim, pseudo ETL fits into the current budget. Pseudo ETL is budget living.

ACKNOWLEDGEMENTS

The authors would like to express their appreciation to the following people for their assistance in this paper:

Patrick Ryan – SunTrust Bank
Tony Rodgers – SunTrust Bank
Randall Anderson – SunTrust Bank

CONTACT INFORMATION

David Steves
SunTrust Bank
303 Peachtree Ctr. N.E.
Atlanta, GA 30303
(V) 404.827.6581
(F) 404.658.4083
david.steves@suntrust.com

Denise A. Figliozzi
SunTrust Bank
303 Peachtree Ctr. N.E.
Atlanta, GA 30303
(V) 404.230.5547
(F) 404.658.4083
denise.figliozzi@suntrust.com

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration. Other brand and product names are registered trademarks or trademarks of their respective companies.