Paper 294-2013

# Time Series Data: Anatomy of an ETL Project

Leonard Polak, Wells Fargo Technology and Operations, Irvine, CA, USA

## ABSTRACT

It's one thing to study SAS® tools and another to apply them to actual situations. In this paper, we follow along as web data is copied and transformed—and ultimately made available to users.

## INTRODUCTION

Our credit risk group wants to automate a repetitive manual process. The process involves downloading files from a web site and modifying these files using Microsoft Excel, before importing the results into SAS.

The goal of the project is to standardize and automate the process. SAS tools used in the new process include: Enterprise Guide, SAS functions, macro facility, and dictionary tables.

The complete SAS program can be found in the appendix. Code snippets are used within the paper to describe particular features.

## EXTRACT

To start, we extract source data. Our source data is on the Internet. It comes from the DataBuffet®, a subscription service of Moody's Analytics, Inc. We enter our vendor assigned user id and password to gain access to the site, see www.economy.com. After authentication, we setup a listing report, which Moody's calls a basket. We enter a report name, which Moody's calls a mnemonic. Our source data may be forecast data or historical data.

Next, we select from a variety of options. Per user request, frequency is set to monthly or quarterly. And decimal places for numeric variables are set to 2. We uncheck four boxes corresponding with four unneeded header rows: Description, Frequency, Source, and Full Description. Per developer preference, the file type is set to comma separated values, or csv, We select an output file name and append include a vendor supplied macro date variable. In the example, 'CPI_FUELS_UTILITIES_%DATE.CSV': 'CPI_FUELS_UTILITIES_' is the output file name; '%DATE' is the date/time stamp; and 'CSV' is the output file type.

| Monthly File | Quarterly File |
|---|---|
| `Mnemonic,XERT.AK,XERT.AL,XERT.AR,XERT.AZ`<br>`Jan-12,33.90,227.20,129.60,297.90`<br>`Feb-12,33.50,224.60,126.70,295.00`<br>`Mar-12,33.60,225.90,128.60,296.10`<br>`Apr-12,34.70,227.30,129.50,297.00` | `Mnemonic,FHOFHOPI.AK,FHOFHOPI.AL,FHOFHOPI.AR,FHOFHOPI.AZ`<br>`2012Q1,287.87,284.21,244.70,238.67`<br>`2012Q2,287.24,281.27,241.03,240.12` |

**Table 1. Sample Source Data**

At this point, we go to the web site's scheduling tool. Here, we name our schedule, schedule a delivery time, and choose a delivery format. In our case, the delivery format is email.

At the appropriate time, we receive email messages with attachments. The attachments contain time series source data files. Monthly data files are received on the first day of each month and quarterly data files are received on the first day of each quarter. The attachments are copied to a Windows folder and then sent to our Unix server using FTP.

As the number of variables in the source data files varies, PROC IMPORT is used to import the source data files into SAS data sets. The asterisk wildcard is used in the DATAFILE option, as we want to import the data based on year and month, ignoring the remainder of the date/time value. One of the files contains null observations, so we exclude these observations.

```
/*Note the asterisk in the datafile option*/
proc import
    out      = &inFile
    datafile = "&inFolder&inFile._&MF1DB_yymmn6.*.CSV"
    dbms     = csv replace;
run;

/*Note the where data set option*/
%if &inFile = RPROBM %then %do;
    data &inFile;
        set &inFile (drop = RPROBM_PR where = (Mnemonic ^= .));
    run;
%end;
```

## TRANSFORM

Now that the source data is in SAS, it is time for transformations.

Having a standard SAS date variable is the first order of business. For the monthly files, PROC IMPORT interprets the first field values as date values; but unfortunately, for the quarterly files, PROC IMPORT interprets the first field values as date/time values. To handle this discrepancy, we use the SAS dictionary tables to get the format into a macro variable. Then we use an if statement and create a new variable.

```
/*Dictionary table used to get format value into macro variable*/
proc sql noprint;
    select strip(format) into :fmt
    from dictionary.columns
    where libname = upper("work")
        and memname = "&inFile"
        and name    = "Mnemonic";
quit;

/*Use macro variable in if statement*/
%if &fmt = MONYY. %then %do;
%else %if &fmt = DATETIME. %then %do;

/*Create new SAS date variable*/
if missing(Mnemonic) then delete; else mneDate = Mnemonic;
if missing(Mnemonic) then delete; else mneDate = datepart(Mnemonic);
```

Next we need to clean up the variable names. Again, using the SAS dictionary tables plus the rename statement, we transform the variable names appropriately.

```
/*Create string for rename statement*/
proc sql noprint;
    select strip(name) || " = " || substr(name,length(name)-1,2),
           strip(name)
           into :ren     separated by ' ',
                :varlist separated by ' '
    from dictionary.columns
    where libname = upcase("work") and memname = "&inFile"
        and name not like 'Mnemonic';
quit;
```

```
/*Add string to code*/
rename &ren;
```

Then, we add calculated variables. We use a macro variable to populate a source report variable. We add several date calculations. We add an observation to each data set, to accommodate for a missing observation. This is done per customer specification. The vendor generally does not add the observation for the prior period before the source data is collected. For example, on October 1, the September data is missing. The end = option in the set statement is used to add the new observation.

```
/*Add calculated variables*/
yyyyqq         = cats(put(year(mneDate),4.),"0",put(qtr(mneDate),1.));
yyyyq          = put(mneDate,yyqn5.);
yyyymm         = put(mneDate,yymmn6.);
snpshot_yyyymm = input(put(intnx('qtr',mneDate,0,'end'),yymmn6.),6.);
code           = "&mnemonic";

/*Add observation to bottom of data set*/
if eof then do;
Mnemonic       = intnx("dtqtr",Mnemonic,1,"begin");
mneDate        = intnx("&interval",mneDate,1,"begin");
yyyyqq         = cats(put(year(mneDate),4.),"0",put(qtr(mneDate),1.));
yyyyq          = put(mneDate,yyqn5.);
yyyymm         = put(mneDate,yymmn6.);
snpshot_yyyymm = input(put(intnx('qtr',mneDate,0,'end'),yymmn6.),6.);
```

Lastly, we format numeric variables consistently. Note that we can use the SAS name list to format all numeric variables with 12.2, then reformat the date variables.

```
/*Format all numeric variables, then reformat the date variables*/
format _numeric_ 12.2 Mnemonic datetime. mneDate mmddyy10. snpshot_yyyymm 6.;
```

## LOAD

At this point, we load the completed SAS data sets into a library for customer use. We create a macro that properly processes a monthly or quarterly job.

```
/*Run monthly or quarterly job*/
%macro doIt;
    %if %sysfunc(mod(&MC0DE_month,3)) = 0 %then %do;
        %macBuffet(inFile = CPI_FUELS_UTILITIES)
        %macBuffet(inFile = HOME_PRICE_INDEX)
        %macBuffet(inFile = MANHEIM)
        %macBuffet(inFile = MORTGAGE_FIXED_RATE)
        %macBuffet(inFile = XEQAUDEL120M)
        %macBuffet(inFile = XEQAUDEL120M_PCT)
        %macBuffet(inFile = XEQAUDEL30M_PCT)
        %macBuffet(inFile = XEQRADEL30M)
    %end;
        %macBuffet(inFile = PAYROLL_RETAIL)
        %macBuffet(inFile = RPROBM)
%mend doIt;

%doIt;
```

To submit a Unix batch job, go to the directory containing the SAS program and use the noterminal option on the command line.

```
nohup sas databuffet.sas -noterminal &
```

Though development work was done using Enterprise Guide 5.1, the project is converted from an .egp file to a .sas file. This allows the code to be run in batch mode and the x command to become available. The x command is used to set file permissions.

```
/*Set permission*/
x "cd /prod/econ_vars/output/";
x "chmod 744 *.sas7bdat";
```

## CONCLUSION

In this paper, we walk through user requirements and present code. Hopefully one or more of the ideas presented are helpful. The new process is superior to the old process.

| Old Process | New Process |
|---|---|
| Manual data downloads. | Scheduled and automated data downloads. |
| Data transformed manually. | Automated data transformations. |
| Great difficulty dealing with dates. | Resolved difficulties with dates. |
| Ad-hoc emailing of result sets to additional users. | Added SAS library and set appropriate permissions. |

**Table 2. Key Improvements**

## REFERENCES


- Carpenter, Art. 2004. *Carpenter's Complete Guide to the SAS Macro Language, Second Edition*. Cary, NC: SAS Institute Inc.
- Cody, Ron. 2004. *SAS Functions by Example*. Cary, NC: SAS Institute Inc.
- Lafler, Kirk Paul. 2004. *PROC SQL: Beyond the Basics Using SAS*. Cary, NC: SAS Institute Inc.
- Murphy, William C. “Changing Data Set Variables into Macro Variables.” Available at http://www2.sas.com/proceedings/forum2007/050-2007.pdf.


## ACKNOWLEDGMENTS

Thanks to Joseph Ofei with Wells Fargo, who provided challenges, and Skylar Bowman with Moody’s Analytics, who provided support.

## CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Leonard Polak
Wells Fargo Technology and Operations
23 Pasteur
Irvine, California 92618
MAC E2718-020
Tel 949-753-3807
leonard.polak@wellsfargo.com

---

## APPENDIX: SOURCE CODE

```
****Start;

%let runDate      = %sysfunc(today());
%let runDate      = %sysfunc(intnx(month,&runDate,-1,end));
%let MC0DB_5      = %sysfunc(putn(%sysfunc(intnx(month,&runDate,0,begin)),5.));
%let MC0DE_month  = %sysfunc(putn(&runDate,month.));
%let MC0DE_yymmn6 = %sysfunc(putn(&runDate,yymmn6.));
%let MF1DB_yymmn6 = %sysfunc(putn(%sysfunc(intnx(month,&runDate,1,begin)),yymmn6.));
%let QC0DB_5      = %sysfunc(putn(%sysfunc(intnx(qtr,&runDate,0,begin)),5.));
%let QC0DE_yyq    = %sysfunc(putn(%sysfunc(intnx(qtr,&runDate,0,end)),yyq.));

%put %sysfunc(putn(&runDate,mmddyy10.));
%put %sysfunc(putn(&MC0DB_5,mmddyy10.));
%put &MC0DE_month;
%put &MC0DE_yymmn6;
%put &MF1DB_yymmn6;
%put %sysfunc(putn(&QC0DB_5,mmddyy10.));
%put &QC0DE_yyq;

%let inFolder     = /prod/econ_vars/input/;

libname outLib '/prod/econ_vars/output';



****ETL;

%macro macBuffet(inFile = );

    proc import out      = &inFile
                datafile = "&inFolder&inFile._&MF1DB_yymmn6.*.CSV"
                dbms     = csv replace;
    run;

    %if &inFile = RPROBM %then %do;
        data &inFile;
            set &inFile (drop = RPROBM_PR where = (Mnemonic ^= .));
        run;
    %end;

    proc sql noprint;
        select scan(name,1,"_") into :mnemonic
        from dictionary.columns
        where libname = upcase("work") and memname = "&inFile"
            and name not like 'Mnemonic';
    quit;

    proc sql noprint;
        select strip(name) || " = " || substr(name,length(name)-1,2),
               strip(name)
```

```
            into :ren     separated by ' ',
                 :varlist separated by ' '
    from dictionary.columns
    where libname = upcase("work") and memname = "&inFile"
        and name not like 'Mnemonic';
quit;

proc sql noprint;
    select strip(format) into :fmt
    from dictionary.columns
    where libname = upper("work")
      and memname = "&inFile"
      and name    = "Mnemonic";
quit;

data _null_;
    if "&fmt" = "MONYY." then call symputx("interval","month");
        else if "&fmt" = "DATETIME." then call symputx("interval","quarter");
run;

%if &fmt = MONYY. %then %do;
    data outLib.&inFile._&MC0DE_yymmn6 (drop = i);
        set &inFile end = eof;
        array numvar(*) &varlist;
            if missing(Mnemonic) then delete; else mneDate = Mnemonic;
            yyyyqq         = cats(put(year(mneDate),4.),"0",put(qtr(mneDate),1.));
            yyyyq          = put(mneDate,yyqn5.);
            yyyymm         = put(mneDate,yymmn6.);
            snpshot_yyyymm = input(put(intnx('qtr',mneDate,0,'end'),yymmn6.),6.);
            code           = "&mnemonic";
        output;
        if eof then do;
            Mnemonic       = intnx("&interval",Mnemonic,1,"begin");
            mneDate        = intnx("&interval",mneDate,1,"begin");
            yyyyqq         = cats(put(year(mneDate),4.),"0",put(qtr(mneDate),1.));
            yyyyq          = put(mneDate,yyqn5.);
            yyyymm         = put(mneDate,yymmn6.);
            snpshot_yyyymm = input(put(intnx('qtr',mneDate,0,'end'),yymmn6.),6.);
            do i = 1 to dim(numvar);
                if numvar(i) ^= . then numvar(i) = .;
            end;
            output;
        end;
        format _numeric_ 12.2 Mnemonic mneDate mmddyy10. snpshot_yyyymm 6.;
%end;
%else %if &fmt = DATETIME. %then %do;
    data outLib.&inFile._&QC0DE_yyq (drop = i);
        set &inFile end = eof;
        array numvar(*) &varlist;
            if missing(Mnemonic) then delete; else mneDate = datepart(Mnemonic);
            yyyyqq         = cats(put(year(mneDate),4.),"0",put(qtr(mneDate),1.));
            yyyyq          = put(mneDate,yyqn5.);
            yyyymm         = put(mneDate,yymmn6.);
            snpshot_yyyymm = input(put(intnx('qtr',mneDate,0,'end'),yymmn6.),6.);
            code           = "&mnemonic";
        output;
        if eof then do;
            Mnemonic = intnx("dtqtr",Mnemonic,1,"begin");
            mneDate  = intnx("&interval",mneDate,1,"begin");
            yyyyqq         = cats(put(year(mneDate),4.),"0",put(qtr(mneDate),1.));
            yyyyq          = put(mneDate,yyqn5.);
            yyyymm         = put(mneDate,yymmn6.);
            snpshot_yyyymm = input(put(intnx('qtr',mneDate,0,'end'),yymmn6.),6.);
```

```
                do i = 1 to dim(numvar);
                    if numvar(i) ^= . then numvar(i) = .;
                end;
                output;
            end;
            format _numeric_ 12.2 Mnemonic datetime. mneDate mmddyy10. snpshot_yyyymm
6.;
    %end;
        rename &ren;
    run;

%mend macBuffet;

%macro doIt;
    %if %sysfunc(mod(&MC0DE_month,3)) = 0 %then %do;
        %macBuffet(inFile = CPI_FUELS_UTILITIES)
        %macBuffet(inFile = HOME_PRICE_INDEX)
        %macBuffet(inFile = MANHEIM)
        %macBuffet(inFile = MORTGAGE_FIXED_RATE)
        %macBuffet(inFile = XEQAUDEL120M)
        %macBuffet(inFile = XEQAUDEL120M_PCT)
        %macBuffet(inFile = XEQAUDEL30M_PCT)
        %macBuffet(inFile = XEQRADEL30M)
    %end;
        %macBuffet(inFile = PAYROLL_RETAIL)
        %macBuffet(inFile = RPROBM)
%mend doIt;

%doIt;



****Permission;

x "cd /prod/econ_vars/output/";
x "chmod 744 *.sas7bdat";
```