Paper 185-30

SPM at 1:00 A.M.

Gary McQuown and Brett Peppe
Data and Analytic Solutions, Inc., Fairfax, Virginia

Abstract

In order for SAS®[1] Strategic Performance Management to provide users with the most timely information, new data should be incorporated into the scorecards on a frequent and timely basis. This process involves extraction, transformation and loading (ETL) of the data, calculating the appropriate metrics, transforming the results into XML, and publishing the XML to the SPM scorecards. While each step in the process can be done manually, automating the process saves time and money and increases quality control. Therefore a data-driven automated process is the ideal means of providing the SPM user with the most recent and accurate data, while decreasing errors and overall costs.

Topics covered include an overview of the automated SPM related processes currently utilized by the U.S. Coast Guard Deepwater project. Examples of the data-driven automation code, tips and tricks to keep the process running smoothly are included. Anyone currently using or planning to use SAS® Strategic Performance Management, similar SAS solutions or any other SAS automated processes should find the information useful.

Introduction

The United States Coast Guard (USCG) under the Dept. of Homeland Security is currently using SAS® Strategic Performance Management (SPM) to monitor and manage its Deepwater acquisition task. Deepwater is a multi-billion dollar effort to rebuild and modernize the aging Coast Guard fleet. The SPM application contains over 200 metrics in over a dozen scorecards. Each metric depicts the current performance, acceptable level of performance and a color coded icon that shows if the acceptable level of performance has been met. The system is automated to the extent that it will load and publish existing metrics automatically at 1:00 A.M. as data becomes available. This paper details the steps necessary to implement a simple automation process. The code contained within may be used to automate both SPM 1.4 and 2.0, or modified to work with similar applications.

SPM

SAS® says that "SAS Strategic Performance Management (SPM) is an enterprise intelligence tool. It enables your organization to establish and monitor strategic goals and measurements at the enterprise level and to align the organization by creating a hierarchy of scorecards."[2] SPM has also been described as a "Balanced Scorecard" or a "Report Card" that monitors the status of an organization.

The following image depicts the Key Metric View for the USCG Deepwater Program Executive Office (PEO) scorecard (Figure 1). This is the highest level view for the application which contains information on all of the available metrics. Metrics at this level are normally "parent" or "group" metrics, a compilation of related lesser metrics. The performance of the group metric is typically assigned the minimal performance value for any of its "child" metric. Therefore, group level metrics tend to be general in nature and most can be drilled into to reveal very detailed information on each underlying child metric. Drill downs are accomplished through the use of "fly over" links that open new windows containing the more detailed information. This process grants the end user the ability to quickly identify which metrics are of immediate interest (those that are red) and then to identify the reason behind the current score (reviewing the child metrics responsible).

---

[1] 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 trademarks of their respective companies.

[2] *Automating and Customizing SAS® Strategic Performance Management Course Notes*, SAS Institute page 1-6, 2001.

Figure 1: Key Metric View for Program Executive Office (PEO) Scorecard

Overview of the Process

The Auto Publish Process has three distinct stages: **Pre Publishing, Publishing and the Post Publishing**. As with most ETL tasks, the majority of the work performed is in the front end. The process described here is SPM (v1.4) centric, but there is no reason that should not work with SPM (v2.0). However, SPM 2.0 automatically performs some of the intermediate chores that our SPM 1.4 implementation requires.

The data used to develop the metrics to be displayed by SPM currently originates as .CSV files, SAS data sets, or Oracle tables. The Auto Publish process is designed to perform the ETL tasks necessary to create the metrics and to automatically publish the results to the scorecard. The Auto Publish process is to perform this task as new data becomes available. Although there are no defined methodologies for the ETL task, we determined that the creation of a single, all encompassing SAS data set was most appropriate. This "Loadtable" would contain the most recent information on every metric within the process as well as any information required for the metrics publishing (Appendix 1).

As each group of metrics is different, each group requires a process tailored to the data, algorithms used to assign the metric and any peripheral information to be provided. This is done through the use of group specific macros, general use utility macros and a parameter file. The parameter file contains information on every metric and denotes to which group it belongs. This file is able to pass along values such as the proper duration of the metric, the base of any URL, the proper rowkey and columnkey, and the upper and lower acceptable values. It is used to avoid "hard coding" values that may change over time and to simplify the addition of additional metrics.

As the name implies, group specific macros perform group specific tasks. This includes reading the data and assigning group specific algorithms that determine the appropriate value for each member within the group. From the culmination of member values, the appropriate value of the group metric is assigned. Most often, the minimal value for all members is cast as the group value. This strategy is sometimes called watching the "weakest link" or wanting to know the bad news early. Depending upon the group and to some

degree the metric involved, an exception report is generated for non conforming values. The final chore for the group specific macros is to place the results into the Loadtable.

SPM 1.4 requires that data be exported through the publication process in the form of XML. SPM 2.0 is table driven and uses a combination of MySQL and SAS datasets through the ETL studio. Regardless of how the scorecard information is stored, the publication process must recover data from the Loadtable into the appropriate format and file. Utility macros are used for this and a variety of other common tasks. Common tasks include assisting with the importation and exportation of the data, archiving, record keeping, and creating exception reports for out of range or missing data, and emailing exception reports to those responsible for the metrics.

SPM 2.0 will automatically update the information on the SPM website as soon as new data is pushed into the MySQL database. Our process with SPM 1.4 requires the use of a batch file to pass parameters to SPM along with the invocation. At invocation SPM 1.4 needs to know the location of the XML files to be processed, the environment to be published and certain date parameters.

Depending upon the complexity of the SPM application, there may be additional tasks that need to be performed after publishing. At the USCG these tasks include updating the print utility and modifying date values in certain files.

Pre Publishing

The purpose of the pre-processing process is to update the Loadtable SAS data set that contains the information to be published and to invoke the publishing process. The Loadtable data set contains the "varname" of the variable to be published, the unique "row_key" and "column_key" combination assigned by the SPM applications, the current performance value of the metric, the "group" that the metric belongs to, the "URL" where additional information is located and any other additional information to be passed to the SPM application for publishing. In addition to updating the Loadtable file, some record keeping is done to keep track of which metrics were processed and to document any exceptions that may have occurred. Email alerts are sent out upon the completion of a successful load, in the event that exceptions were discovered and if "pushed" data is not delivered on time to the proper location.

The following is a list of the tasks performed by the Auto_Load process (chronological order):

1. Creates date specific macro variables
2. Archives the last loadtable
3. Reads the Parm file to determine what input files are expected
4. Checks for new data delivered or data to be pulled
5. Creates a list of new data groups to be processed
6. Invokes input programs by group
    a. Reads the input file or extracts the data
    b. Archives the data in SAS format
    c. Archives the original file if .CSV
    d. Updates scheduling file
    e. Performs data transformation as required
    f. Calculates metric
    g. Calculates metrics acceptability
    h. Updates Loadtable
    i. Creates Automatic Links (drill down reports, etc. as needed)
    j. Emails exception reports (as needed)
7. Export Loadtable data in XML format
8. Send notification that new metrics have been processed

If no new data is available, only steps 1, 2, 3 and 4 are performed.

Due to the long descriptive names often associated with scorecards which are difficult to manipulate using the manual interface, we chose to delineate all metrics by "varname", a short, unique, yet descriptive name that can be used for sorting and merging. A "group" name was used to further categorize the metrics and to determine which import macro is used for metrics that shared significant traits.

The SAS programming required by most of the Pre-Process tasks is usually straight forward. However some groups of metrics may be more difficult than others due to the complexity of the metrics and their

accompanying algorithms. The two tasks that are more specific to SPM are the creation of Automatic Links and the exportation of the results in XML format. Macros for both can be found in the appendix.

The Automatic Links macro associates a URL with the proper SPM scorecard. The URL may be a link to additional information in the form of a drill down report created with ODS, or a link to another SAS process that dynamically generates a dynamic web page. Base information for the process is contained in a parameter file. In this instance, the required information is the path and format for each URL. Any portion of the path that is date specific is stored as a macro variable so that it resolves correctly within the process.

The AUTO_POST_XML.sas macro is used to convert SAS data by group into XML. As you only want to publish information that has recently changed, only the metrics or groups of metrics that have been processed should be converted. While processes up to this point have relied on “varname” to identify metrics, this process must output the information by “rowname” and “rowkey” for SPM to process. Once again, some of the information is obtained from the parameter file (period, target, etc.).

Publishing

When using SPM 1.4, it is necessary to invoke the process on a host machine. Since our SPM implementation uses Windows Server, this was done by creating a batch program and issuing a system call similar to the following;

```
data _null_;
  rc = system('d:\spm_batch.bat');
run;
```

The creation of the .bat file (appendix 4) is done with a combination of put statements and macro variables. The .bat file should contain the following parameters:

1)		Quoted path to SPM map.exe
2)	–datasourceload	Quoted directory of XML files to load
3)	–dir	Quoted directory where source data resides
4)	–publish	Quoted directory to publish to
5)	–id	The ID with period MMMYYYY
6)	–stamp	Period end date DDMMMYYYY

The end result is an executable command that should look something like this:

```
"C:\Program Files\SAS\Strategic Performance Management\1.4\map.exe"
-datasourceload "D:\Coast Guard Performance Mgmt\USCG-Deepwater\datasource"
-dir "D:\Coast Guard Performance Mgmt\USCG-Deepwater"
-publish "\\<Path to Server>\compass_publish$"
-id USCG.Dec2004
-stamp 31Dec2004
```

Post Publishing

Due to the complexity of the USCG SPM application, there are some activities that must take place on the remote web server after the scorecards have been published. The most significant is the running of a non SAS program to produce the data for the Print Utility. This utility allows users to print a MS Word formatted document containing all of the metrics and their scores. The conditional remote start is accomplished through a SAS program that is invoked on the remote server (NT Scheduler) at a specific time every morning (2 A.M.). If the program locates the ‘trigger’ file produced by the auto_publish.sas program, the Post Publishing batch files are invoked. If the ‘trigger’ file has not been sent, no post processing will take place.

Of course no task would be complete without some degree of quality control and oversight, and a SPM implementation is no exception. One of the most important tasks is to insure that all of the files remain ‘in sync” so that every metric is accounted for and processed correctly.

Conclusion

The end result of the efforts at the USCG is an automated SPM 1.4 application that processes and publishes metrics nightly at 1:00 A.M. if new data is available. In addition to publishing the new values, the application also produces a series of quality related exception reports and intermediate files for analysis. This process will be modified as SPM is migrated from 1.4 to 2.0 and many of the publishing related tasks are handled automatically through the new software. However, the ETL requirements and the vast majority of the macrocode used to meet them will continue to be of use for version 2.0 and later.

References:

SAS Institute, Inc., *Automating and Customizing SAS® Strategic Performance Management Course Notes*, SAS Institute page 1-6, Cary NC, 2001

Acknowledgements:
Special thanks to the United States Coast Guard, Greg Cohen of the USCG, Jimmy Barillaro of SAS Institute, and Dr. Dawn Li of Data and Analytic Solutions, Inc.

Contact Information

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

Author Names	Gary McQuown & Brett Peppe
Company	Data and Analytic Solutions, Inc.
Address	3127 Flintlock Road
City state ZIP	Fairfax, VA 22030
Work Phone:	703.628.5681
Email:	mcquown@DASconsultants.com
	brettpeppe@DASconsultants.com
Web:	www.DASconsultants.com

Publications by these authors and others may be found at http://www.dasconsultants.com/pubs.html

Appendix:

Please note that all code may be used without the author's consent and "as is" at the user's discretion.

Appendix 1

Suggested Fields for the Loadtable Data Set

ALERT_SET	Flag 1 = Unique Method, 0 = General Method
DATASOURCE	Name of input file or data set
GROUP	Group Name: Risk, CDRL, etc.
LOWER	Numeric boundary between Red and Yellow
OBJECTIVE	Objective Classification
PERIOD	Time Period: Year, Month, Quarter, etc.
PERSPECTIVE	Perspective Classification
THRESHOLD	Icon flag (1-red, 2-yellow, 3–green)
UPPER	Numeric boundary between Green and Yellow
URL	Link to auxiliary information (drill downs, etc.)
URL_TARGET	"_new_" to open a new window
CARDKEY	Card Key identification for SPM
CARDNAME	Card Name identification for SPM
LAST_UPDATE	Date of last update
PERFORMANCE	Numeric value for metric
ROWKEY	Row Key identification for SPM
ROWNAME	Row Name identification for SPM
TARGET	Display value between Green and Yellow
TREND	Recent trend indicator
VARNAME	Unique name of metric

Appendix 2

Pre Processing: Item 5. i. Creates Automatic Links

```
%macro automatic_links(list);

 %let list = &list STOP;
 %LET i = 1;
 %DO %WHILE (%scan(&list , &i) ^= STOP );

  %let group  = %scan(&list, &i);

proc sort data=sdata.parms out=_temp
(where=(upcase(group)=upcase("&group"))) ;
  by cardkey;
run;

data _null_;
  set _temp (obs=1);
  url = translate(url,' ', '0D'x);
  url = compress(url);
  check = length(url);
  call symput("ck", check);
  call symput("URL", left(trim(URL)));
  call symput("datasource",compress("&group.Report.xml"));
run;

  %put ck = &ck;
  %put url = &url;
```

```
  %put datasource= &datasource;

  %if &ck > 0 %then
  %do;
      data null;
      length colkey $3 ;
            file "D:\XXX\YYYY\Deepwater\datasource\&datasource.";
      set work._temp end=eof;
            colkey = 'c_3';
            if (_n_ eq 1) then
            do;
            put '<CELLS>';
            end;

            PUT " <DATA>";
            PUT " <TABLE>" cardkey +(-1) "</TABLE>";
            PUT " <ROW>" rowkey +(-1) "</ROW>";
            PUT " <COLUMN>" colkey +(-1) "</COLUMN>";
            PUT " <URL_TARGET>" url_target +(-1) "</URL_TARGET>";
            PUT " <URL>" "&url" "</URL>";
            PUT " </DATA>";
            PUT " <ACTION><TYPE>SAVE</TYPE></ACTION>";
      IF (eof) THEN PUT "</CELLS>";
      RUN;
  %end;

%LET i = %EVAL(&i+1);
%END;

%mend automatic_links;
```

Appendix 3
Pre Process Item 7: Export Loadtable data in XML format

```
data null;

%macro auto_post_xml;

%put varn = &varn ;

/** SELECT ALL METRICS FROM EACH GROUP THAT ARE BEING UPDATED **/
proc sql noprint;
    select rowkey into: rowkey_list separated by '*' from
hdata.loadtable
    where group = "&varn";
    select cardkey into: cardkey_list separated by '*' from
hdata.loadtable
    where group = "&varn";
      select varname into: varname_list separated by '*' from
hdata.loadtable
    where group = "&varn";
      select period into: period_list separated by '*' from
hdata.loadtable
    where group = "&varn";
      select count(group) into: numobs from hdata.loadtable
      where group = "&varn";
quit;
```

```
 /** SELECT A SPECIFIC METRIC TO BE PROCESSED  **/

%do jj = 1 %to &numobs ;
   %put jjb = &jj;
  %let bsc_scorecard_row  = %scan(&rowkey_list, &jj, '*' );
  %let bsc_scorecard_key  = %scan(&cardkey_list, &jj , '*' );
  %let bsc_period         = %scan(&period_list, &jj, '*' );

  %let xmlname = &bsc_scorecard_key._&bsc_scorecard_row..xml;

   %put bsc_scorecard_row = &bsc_scorecard_row;
   %put bsc_scorecard_key = &bsc_scorecard_key;


    data null;
          file_txt = "D:\XXX\YYYY\Deepwater\datasource\&xmlname";
           call symput('datasource', file_txt);
         run;

         %put datasource=&datasource;

         data odd(keep=key_metric_column data);
            set  metadata.key_metric_columns
                 (rename=cname=key_metric_column);
            data = .;
         run;

         proc transpose data=odd
           out=temp.odd(drop=oldvar) name=oldvar label=oldlabel;
            format key_metric_column;
            id key_metric_column;
         run;

         data temp.odd;
            set temp.odd;
            from_date = .;
            to_date =  .;
         run;

         /* Current metric name into a macro variable   */
         data _null_;
           set metadata._rows(where=(rowkey="&bsc_scorecard_row" and
                 cardkey="&bsc_scorecard_key"));
           call symput('bsc_scorecard_row_desc', rowname);
           call symput('bsc_scorecard_row_code',rowkey);
         run;

         /* Current Scorecard into a macro variable.        */
         data _null_;
           set metadata._scorecards
           (where=(cardkey="&bsc_scorecard_key"));
           call symput('bsc_scorecard_card_desc', cardname);
           call symput('bsc_scorecard_card_code',cardkey);
         run;

         /* Get information from summary dataset */
```

```
/* scorecard and metric. */
data odd;
   set datasrc.loadtable;
   where (cardkey eq "&bsc_scorecard_card_code") and
         (rowkey eq "&bsc_scorecard_row_code");
run;

   /* The dataset TEMP.ODD is used by the LSA Wizard to
        create the XML files */
data odd2 temp.odd;
    merge  temp.odd odd;
run;

 data _null_;
   set temp.odd end=eof;

   file "&datasource";

   b = put(intnx("&bsc_period","&cur_date",-1,"b"),date9.);
   e = put(intnx("&bsc_period","&cur_date",-1,"e"),date9.);

   if (_n_ eq 1) then put '<CELLS>';

       /* begin and end date */
   put " <DATA>";
   put " <TABLE>" cardkey +(-1) "</TABLE>";
   put " <ROW>" rowkey +(-1) "</ROW>";
   put " <START>" b +(-1) "</START>";
   put " <END>" e +(-1) "</END>";
   put " </DATA>";

    /* trend  */
   put " <DATA>";
   put " <TABLE>" cardkey +(-1) "</TABLE>";
   put " <ROW>" rowkey +(-1) "</ROW>";
   put " <COLUMN>" 'c_0' "</COLUMN>";
   put " <VALUE>" trend "</VALUE>";
   put " </DATA>";

   /* performance   */
   put " <DATA>";
   put " <TABLE>" cardkey +(-1) "</TABLE>";
   put " <ROW>" rowkey +(-1) "</ROW>";
   put " <COLUMN>" 'c_3' "</COLUMN>";
   put " <VALUE>" performance "</VALUE>";
   put " </DATA>";

   /* threshold   */
   put " <DATA>";
   put " <TABLE>" cardkey +(-1) "</TABLE>";
   put " <ROW>" rowkey +(-1) "</ROW>";
   put " <COLUMN>" 'c_17' "</COLUMN>";
   put " <VALUE>" threshold "</VALUE>";
   put " </DATA>";

   /* target   */
   put " <DATA>";
```

```
              put " <TABLE>" cardkey +(-1) "</TABLE>";
              put " <ROW>" rowkey +(-1) "</ROW>";
              put " <COLUMN>" 'c_6' "</COLUMN>";
              put " <VALUE>" target  "</VALUE>";
              put " </DATA>";

              /* put " <ACTION><TYPE>SAVE</TYPE></ACTION>"; */

              if (eof) then put "</CELLS>";
            run;

  /** %let jj = %eval(&jj+1); **/
  %end;
%mend auto_post_xml;
```

Appendix 4
Auto_Publish.sas

```
/*************************************************************
auto_publish.sas

This program writes a batch job that will instruct SAS SPM to read
the XML files from the data source directory and then to publish.

*************************************************************/

%macro auto_publish;

%let exec_loc      = "C:\Program Files\SAS\Strategic Performance
Management\1.4\map.exe";
%let store_loc     = "D:\Coast Guard Performance Mgmt\USCG-Deepwater";

%let publish_loc   = "\\123.456.7.89\compass_publish$";
%let datasource    = "D:\Coast Guard Performance Mgmt\USCG-
Deepwater\datasource";

%put &pre_month_b;
%put &pre_month_e;

data _null_;

monyy = lowcase(put(&pre_month_b, monyy7.));
edate = lowcase(put(&pre_month_e, date9.));

monyy=upcase(substr(monyy,1,1))||substr(monyy,2);
edate=upcase(substr(edate,1,3))||substr(edate,4);

call symput("monyy", monyy);
call symput("edate", edate);
run;

%put &monyy;
%put &edate;

%let a = %bquote (&exec_loc);
%let b = %bquote (-datasourceload &datasource);
%let c = %bquote (-dir &store_loc);
```

```
%let d = %bquote (-publish &publish_loc);
%let e = %bquote (-id USCG.&MONYY);
%let f = %bquote (-stamp &edate);

/* Print .bat file, execute, and set up Post Publish process */

data _null_;
   a = "&a" ;
   b = "&b" ;
   c = "&c" ;
   d = "&d" ;
   e = "&e" ;
   f = "&f" ;
   file 'd:\spm_batch.bat' lrecl=500;
   put a b c d e f;
run;

data _null_;
   rc = system('d:\spm_batch.bat');
run;

/**   the creation of the following program "run_post_pub_<date>.html
triggers a chron job run on the Web server to run Post Processes to
update the Print Utility and to format dates in the score card
**/

ods html nogtitle  nogfootnote

body="\\123.456.7.89\compass_publish$\support\run_post_pub_&cur_date..h
tml";
options ls=90;

title;
footnote;
data a;
  date = put(date(), date9.);
  time = put(time(), time9.);
  process = "DUMMY TO CALL PRINT UTILITY";
run;

proc print data=a;
run;
ods html close;

%mend auto_publish;
```