

PROCESSING TESDATA MEASUREMENT FILES  
with the  
STATISTICAL ANALYSIS SYSTEM (SAS)

J. William Mullen, Peat, Marwick, Mitchell & Co.

ABSTRACT

The availability of non-sensor based measurement capability, as provided by the MS-108 Channel Event Monitor and the MS-109 Communications Monitor, has added significantly to the volume of measurement data available to the analyst. Merging of this measurement data with measurement data from the host system, such as RMF data, has become a very desirable capability. This paper describes techniques for the transfer of MS Performance Data Files (PDFs) to the host system utilizing the Statistical Analysis System (SAS) for data transfer, analysis and reporting.

INTRODUCTION

The availability of non-sensor based measurement capability, the MS-108 and MS-109, has provided the performance analyst with a significant increase in the amount and types of data available for system tuning, management reporting and capacity planning activities. This increase in previously unavailable measurement capability and data is particularly true in the communications measurement area. In concert with this increase in data is the necessity for additional disk storage for data base purposes, computing power for large data reduction requirements and flexibility in analysis and reporting of the measured data. The current MS System is limited in its capacity to store and provide reduction/reporting capability for the significant amount of measurement data that is now available (i.e. MS-109).

The availability of disk and/or tape storage and processing power available on host mainframes and the utilization of the data reduction and reporting capabilities available with the Statistical Analysis System (SAS) provide an appropriate alternative for data storage, reduction, analysis and reporting.

The first step in processing MS System files at the host system is the transfer of the data from the MS System to the host into a form suitable for processing. This transfer is accomplished via SAS and the placement of the data in a SAS data set. This transfer requires an understanding of the format of the MS System files and the representation of data in these files. This section will discuss the format of Performance Data Files (PDFs) stored at the MS System, the representation of data stored in the PDFs and methodologies for transfer of PDF data to the host system. SAS coding examples will be used where appropriate

to illustrate the technique being discussed.

PERFORMANCE DATA FILE STRUCTURE

Performance Data Files on the MS System are random files (TYPE=1) consisting of multiple 1536 byte records. The Trident disk used to store the PDFs utilizes 512 byte sectors and thus each PDF record requires six sectors of Trident space. The first record in the PDF is a header record followed by a variable number of data records whose number is dependent upon the PDF time interval chosen when the PDF was allocated. An example would be a PDF whose data is being stored in 5 minute intervals. The PDF would contain 289 data records which would consist of a header record and 288 data records. Since allocation on the Trident disk is in tracks, the PDF would also contain additional 1536 byte records which are formatted when the PDF is built but are not used for data recording.

The PDF header contains descriptive data identifying the owner, title, begin and ending dates of the measured data and accumulator descriptions for each of the 16 allowable accumulators. This descriptive data identifies the accumulator as to type of data being stored, the base value for the accumulator and provides a 20-byte description for accumulator identification purposes during the reporting process. The descriptive data in the header record is stored in Ascii format. The base values for each of the 16 allowable accumulators are stored in single precision floating-point format.

Each PDF data record represents one unit of the time interval designated when the PDF was built. A 5 minute PDF will have 288 data records, a 10 minute PDF will have 144 data records, etc.

Within each data record, there are 32 groups of 17 halfwords for data storage. The first 31 groups provide data storage for the individual days of the month and the 32nd group is used for summary data (i.e. Sunday, Monday, etc.). Measurement data is stored in the first 17 halfwords of each group and the number of occurrences of update for this day is stored in the 17th halfword of each group. Measured data is stored as percentages of the base for the respective accumulator and is converted to true values by the MS Performance Management Software (PMS11) during report generation. All halfword accumulator values (percentages) are stored in integer binary and do not need translation at the host before processing.

PERFORMANCE DATA FILE TRANSFER

Currently, magnetic tape is the only medium available for transfer of MS system data to the host system. MS System PDFs are stored off-line by transferring them from the Trident disk to tape using the dump utility (//UTIL) on the MS System. The format of the //UTIL tapes and their processing at the host system will be discussed later in this paper. A more suitable method for processing PDFs on the host system is to place the PDF header and data records on tape in sequential format with unblocked, 1536 byte records. To accomplish this, the PDF format must be changed from random to sequential. The Attribute (/AT) command is used specifying TY=0 for the PDF to be changed. The PDF can then be written to tape in sequential format with the //COPY utility available at the MS System. Multiple PDFs can be placed on the same tape using this method (i.e. for 10 minute PDFs, 20 PDFs can be placed on one tape in sequential format). The following example procedure will place two PDFs on a tape in sequential format:

```

AS 285          Assign LU2 to tape
/AT PDF1 TY=0   Make PDF sequential
/AC PDF1 1      Activate on LUI
//COPY         Invoke COPY program
BS 1536        Specify disk block
C              Copy PDF records
EX             Exit the COPY program
/AT PDF1 TY=1   Make PDF random
/CL 1          Close LUI
/AT PDF2 TY=0   Make PDF2 sequential
/AC PDF2 1      Activate on LUI
//COPY         Invoke COPY program
BS 1536        Specify disk block
C              Copy PDF records
EX             Exit COPY program
/AT PDF2 TY=1   Make PDF random
/CL 1          Close LUI
WF 2           Write tape mark
RW 2           Rewind tape

```

The output tape now contains two PDFs with a fixed sequential format, 1536 byte records and can now be processed at

the host system. SAS program processing of the tape is divided into two sections: header record processing and data record values conversion and processing.

PDF HEADER RECORD PROCESSING

Performance Data File header record processing requires recognition of a record as a PDF header, input of the required header data fields and retention of the data fields from the header record used in processing the PDF data records. If utilizing the sequential format with only one PDF on the input tape, PDF header recognition is very simple:

```

INFILE PDFIN ;
INPUT @1 DATA1 PIB1. @ ;
IF _N_ = 1 THEN GO TO DATAP ;

```

header processing

```

DONE: INPUT; DELETE;
DATAP:      process data records

```

Multiple PDFs on the same tape in the sequential format require specific header record recognition. With the PMS11 header record format, the acronym 'PMS2' is present in the header record at location 459. A method for determining the presence of the acronym requires use of the TRANSLATE facility available in SAS 79.3. The following is an example of PDF header recognition:

```

INFILE PDFIN ;
INPUT @459 PM2 $4. @ ;
PM2=TRANSLATE(PM2,X'D7D4E2F2',
              X'504D5332') ;
IF PM2 = 'PMS2' THEN GO TO DATAP ;

```

header processing

```

DONE: INPUT; DELETE;
DATAP:      process data records

```

Each occurrence of a header record requires input and retention of the new header information for subsequent processing of the following PDF records.

Processing the PDF header record fields again requires the use of the SAS TRANSLATE facility. This is accomplished via a macro TRANS, set up to translate the header record fields needed from ASCII to EBCDIC. The first 600 character positions of the header record contain all the ASCII fields that require translation. Due to the SAS restriction limiting a character field to 200 characters, the first 600 characters are input as 3 separate data elements to accomplish the translation process. Once the 3 data elements are translated, the relevant fields (owner, title, start and stop dates, accumulator descriptions) may be extracted from their respective data element fields using the SUBSTRING function.

The following SAS code may be used to translate the data fields:

```

MACRO TRANS
EBCDC1=' numeric hex values ' ;
EBCDC2=' alpha hex values ' ;
EBCDC3=' special characters ' ;
ASCI11=' numeric hex values ' ;
ASCI12=' alpha hex values ' ;
ASCI13=' special characters ' ;
%

TRANS
INFILE PDFIN ;
INPUT @1 DATA1 $200. @191 DATA2 $200.
      @361 DATA3 $200. @453 INTVAL PIB2.
      @1437 (BASE1-BASE16) (RB4.)
      @1505 (TYPE1-TYPE16) (PIB1.) @ ;

DATA1=TRANSLATE(DATA1,EBCDC1,ASCI11,
                EBCDC2,ASCI12,EBCDC3,
                ASCI13 ) ;
DATA2=TRANSLATE(DATA2,EBCDC1,ASCI11,
                EBCDC2,ASCI12,EBCDC3,
                ASCI13 ) ;
DATA3=TRANSLATE(DATA3,EBCDC1,ASCI11,
                EBCDC2,ASCI12,EBCDC3,
                ASCI13 ) ;

```

All relevant character data fields are now translated and individual data elements may be substringed from the DATA1, DATA2 and DATA3 fields. Once the header data is extracted, it must be retained while reading and processing data records until the next header record is encountered.

Accumulator types and bases were input in concert with the DATA1, DATA2 and DATA3 fields. The type fields for each accumulator indicate the type of measurement data for that accumulator (time, count, average value, maximum, minimum) and determine the application of the accumulator's base value when reporting measurement from the accumulator. Use of the type and base fields will be discussed in the section on processing data records.

With completion of input, translation and formatting of the PDF header record, the data may be output to a SAS data set for report generation. Certain data elements should be retained (PDFID,type and base fields) for use in processing the data records. Though the processing of the header record does require considerable overhead, this overhead is relatively small in comparison to overall processing of the total PDF.

#### PDF DATA RECORD PROCESSING

A PDF data record is present for every time interval as specified when the PDF was built. Each data record contains 32 groups of 17 halfwords. Each of the first 31 groups represents measured data for the days of a given month with the 32nd group not used in PMS2.

Within each group, the first 16 half-words are the accumulator values for the measured time period and the 17th half-word is a count of update occurrences for the group. The 16 accumulator values are halfword integer binary (PIB2.) percentages and represent the measured data as a percentage of the base value for the accumulator. Time (T) type accumulators always have a base value of 1 and a value from 0%-100%. Count (C) type accumulators may have a value range of 0%-327.67% which represents the measured data as a percentage of the accumulator base value in counts-per-second. Average (V) value, maximum (X) value and minimum (N) value accumulator types will contain a value range of 0%-327.67% which is the measured data as a percentage of the base value for the accumulator. Accumulator values are converted to actual data values by the PMS2 software during report generation. When inputting the data with SAS, accumulator values are converted so that the actual measured data value for the accumulator is stored in the SAS data set.

Accumulator data values may be input using two approaches. The first of the se is to input each day(group) of 17 values(this includes the occurrence value) together and store the converted accumulator values and the occurrence field as a single observation in the SAS data set. The following SAS code may be used to do this:

```

DATAP:
PTR=1 ; *set to beginning of data ;
PTIME=PTIME + INTVAL ;
DO I = 1 TO 31 ;
  INPUT @PTR (ACCV1-ACCV16) (PIB2.)
        +32 OCCU PIB2. @ ;
  BASEM *accu. conversion macro ;
  PDFMT *name accu. values ;
  DAY=I ;
  OUTPUT PDFDATA ;
END ;

```

Two macros are used in the above sequence. The BASEM macro converts each accumulator (ACCV1-ACCV16) to an actual value based upon that accumulator's type and base. The PDFMT macro assigns a descriptive name to each of the 16 accumulator values (i.e. CPU BUSY=ACCV1; PROBLEM=ACCV2;). This method was used since multiple macros could be developed and the appropriate macro for the PDF being processed selected and concatenated ahead of a single PDF processing program.

The second approach is to input each of the accumulator values individually and store the converted value as a single observation in the SAS data set. The following is an example of the SAS code to accomplish this:

```

DATAP:
  PTR=1 ;
  PTIME=PTIME + INTVAL ;
  DO I1 = 1 TO 31 ;
    DAY = 11 ;
    DO INDEX3 = 1 to 17 ;
      INPUT @PTR ACCV PIB2. ;
      BASES
      ACCNO = INDEX3 ;
      OUTPUT PDFDATA ;
    END ;
    PTR = PTR + 34 ;
  END ;

```

The BASES macro performs the same conversion function as the BASEM macro, but uses the value of INDEX3 as an indicator of the accumulator being processed.

Observations output to the SAS data set will also contain data elements such as PDF month, year and PDF identification that were extracted from the PDF header record and retained for data record processing. The day of the month, time interval value and accumulator number are derived during data record processing. Once the PDF data has been placed in the SAS data set, the analyst has many options for analysis and reporting as well as merging the PDF data RMF and/or SMF data for analysis and reporting.

Placing the PDF data on tape in sequential format requires little more effort than the standard //UTIL method and provides the analyst flexibility of processing the data with SAS as well as other high-level language processors.

#### //UTILITY TAPE FORMAT PROCESSING

Placing Performance Data Files on tape at the MS System is usually done using the //UTILITY program. The //UTILITY program reads the PDF to be dumped and blocks the output tape, placing 4 PDF records in each block. Each block of PDF records on the //UTIL tape is preceded by a 28-byte header which is included in the data block. Standard blocksize on the //UTIL tape is 6172 bytes per block except for the last block dumped for each PDF. This short block is due to the space allocation algorithm for the Trident disk which results in additional, unused sectors being formatted during the PDF building process.

Processing of PDF header and data records from a //UTIL tape follows the same procedures as the processing from the sequential format tape with additional SAS code required to recognize tape blocks that contain PDF header records and to de-block the tape data records and obtain the individual PDF records.

Recognition of tape blocks containing PDF header records can be accomplished by

interrogating the 28-byte //UTIL header record at the beginning of each tape data block. The first block of PDF records, containing the PDF header record and 3 data records, will have an '\*T' as the first 2 bytes of the //UTIL header. Subsequent tape blocks containing only PDF data records will have binary zeroes (X'0000') as the first 2 bytes of the //UTIL header (the binary representation for the '\*T' will be X'2A54'). Upon recognition of the '\*T' in the first 2 bytes of the tape data block, header record processing for the PDF can begin by moving the SAS pointer past the 28 byte //UTIL header and entering the normal PDF header record processing code. For PDF data record blocks, the SAS pointer is moved past the 28 byte //UTIL header and normal data record processing code is entered. As the //UTIL program dumps the PDF to tape at the MS System, the name of the PDF being dumped is placed in each //UTIL header record at location 6. This provides the opportunity to select specific PDFs from a //UTIL tape for transfer to a SAS data set. The selection can be accomplished by single character recognition in the PDF name or by again using the TRANSLATE feature to translate the total 6 character PDF name and selecting the PDF(s) to be processed by a direct compare on the PDF name.

An example of the SAS code to process //UTIL format tapes:

```

INFILE PDFIN LENGTH=LEN ;
INPUT @1 UTLID $2. @ ;
IF LEN ^= 6172 THEN DELETE ;
IF UTLID ^= '2A54'X THEN GO TO DATAP ;

```

```

PTR = 29 ;

```

process header data

```

PTR = 1565 ; *past //UTIL and PDF header;
LVAL = 3 ; *deblocking loop value ;
GO TO DATAP1 ;

```

```

DATAP:

```

```

  PTR = 29 ; *to first data record ;
  LVAL = 4 ; *deblocking loop value;

```

```

DATAP1:

```

```

  PTR1 = PTR ; *PTR1 for deblocking;

```

Example 2:

```

INFILE PDFIN LENGTH=LEN :
INPUT @1 (UT1-UT2) (PIB1.) @ ;
IF LEN ^= 6172 THEN DELETE ;
IF UT1 ^= 42 AND UT2 ^= 90
  THEN GO TO DATAP ;

```

In both examples, the presence of the //UTIL header indicating a tape data block containing a PDF header is checked. In the first example, the PDF header is processed

and the SAS pointer is then advanced to the first data record in the header block. The variable LVAL is set to 3 to control deblocking of the tape data block for a PDF header record block and is set to 4 to control deblocking of a standard PDF data record block. The LENGTH parameter is used on the INFILE statement so that short tape data blocks occurring at the end of each PDF dump may be discarded.

Reporting of the PDF data from the SAS data set is left to the imagination of the analyst. The intent of this paper and accompanying code examples is to help the analyst overcome the difficulties of data transfer of PDFs from the MS System to the host processor. Additional examples of reporting and analysis techniques will be the subject of a future paper.

#### SUMMARY

The ability to transfer Performance Measurement Files from the TESDATA MS System to the host mainframe for data reduction, analysis and reporting demonstrates the flexibility and power of SAS in aiding the performance analyst to reduce and analyze large amounts of detail measurement data and present this data in a wide variety of formats. The ability to merge the MS System data with measurement data from host software monitors such as RMF gives the analyst the opportunity to do analysis on both internal and external measurement data from the same time intervals to better understand the factors governing a computer configuration's performance and to enhance the analyst's ability to report the data to management in the required formats.

The problems with storage and data reduction of large amounts of data at the MS System are alleviated by the data transfer capability provided by SAS. Use of the measurement data from the MS System at the host will also provide the analyst reporting flexibility in that many periods of data can be represented on one SAS chart or graph. This will enhance the analyst's ability to do long term tracking of configuration usage and performance and supply additional inputs to an installation's capacity planning efforts.

APPENDIX A.  
SAS MACRO EXAMPLES

The macros referenced in the preceding text are given here to demonstrate the SAS code necessary for data transfer. In some cases the total macro is not included since the lines included are adequate for completion of the macro.

TRANS Macro

The TRANS macro is used to convert the ASCII data in the PDF header to EBCDIC. The definitions are divided into three parts containing numeric data, alpha data and special characters.

```
MACRO TRANS
EBCDC1='C1C2C3C4C5C6C7C8C9D1D2D3D4D5
        D6D7D8D9E2E3E4E5E6E7E8E9'X;
EBCDC2='FOF1F2F3F4F5F6F7F8F9A4B4C4D
        4E4F505A5B5C5D5E5F60616A'X;
EBCDC2='6B6C6D6E6F7A7B7C7D7E7F'X ;
ASCII1='4142434445464748494A4B4C4D4E
        4F505152535455565758595A'X;
ASCII2='30313233343536373839002E3C28
        2B212621242A293B5E2D2F21'X;
ASCII3='2C25403E3F3A23402C3D2220'X;
%
```

TYPEC Macro

The TYPEC macro extracts the accumulator type from the type fields obtained from the PDF header record. The type fields in the PDF header record are input as PIB1. variables with each of the 16 accumulator types derived from the type fields.

```
MACRO TYPEC
TYPE1=FLOOR(TYPE1 / 10) ;
TYPE2=FLOOR(TYPE2 / 10) ;
TYPE3=FLOOR(TYPE3 / 10) ;
.
.
.
TYPE15=FLOOR(TYPE15 / 10);
TYPE16=FLOOR(TYPE16 / 10);
%
```

The type of the accumulator is derived from the tens digit position of the variable:

- 10 = Time Accumulator
- 20 = Count Accumulator
- 30 = Average Value Accumulator
- 40 = Maximum Value Accumulator
- 50 = Minimum Value Accumulator

PDFMT Macro

The PDFMT macro is used to assign

descriptive variable names to each of the 16 accumulator values after input. The name assigned will be used during the report processing for summarization of data and report item descriptions. This form of the macro assumes that the 16 accumulator values are input as a single observation for storing in the SAS data set.

```
MACRO PDFMT
CPU_BUSY=ACCV1 ;
PROBLEM =ACCV2 ;
CH1_BUSY=ACCV3 ;
CH2_BUSY=ACCV4 ;
.
.
.
LCH3_BSY=ACCV16 ;
%
```

This macro can be expanded to encompass all PDFs that might be processed by checking the PDF identification extracted from the PDF header and using this id to determine the set of assignment statements to be executed. As PDFs are added or changed, the analyst need only update the macro to incorporate the additional or changed PDFs.

BASEM Macro

The BASEM macro is used to convert the percentages contained in the 16 accumulators to actual data values based upon the type of accumulator as determined from the PDF header record type field. This conversion is the same as the conversion performed by the PMS2 software during report generation at the MS System.

```
MACRO BASEM
IF TYPE1 = 1 THEN
    ACCV1 = (ACCV1 / 1000) * 100 ;
IF TYPE1 = 2 THEN
    ACCV1 = (ACCV1*BASE1*INTVAL*
            60) / 10000 ;
IF TYPE1 > 2 THEN
    ACCV1 = (ACCV1*BASE1) / 10000;
.
.
.
IF TYPE16 = 1 THEN
    ACCV16 = (ACCV16/10000) * 100;
IF TYPE16 = 2 THEN
    ACCV16 = (ACCV16*BASE16*INTVAL*
            60) / 10000 ;
IF TYPE16 > 2 THEN
    ACCV16 = (ACCV16*BASE16) / 10000;
%
```

### BASES Macro

The BASES macro performs the same function as the BASEM macro on the accumulator values when each of the 16 accumulator values are input as a single observation. The BASEM macro is dependent upon the loop control index variable to determine the accumulator value it is working with at the time. In the text, the loop variable used in the example code was INDEX3 and is used here in the sample macro code.

```
MACRO BASES
IF INDEX3 = 1 THEN GO TO IX2 ;
IF TYPE1 = 1 THEN
  ACCV = (ACCV / 10000) * 100 ;
IF TYPE1 = 2 THEN
  ACCV = (ACCV*BASE1*INTVAL*60)
  / 10000 ;
IF TYPE1 > 2 THEN
  ACCV = (ACCV*BASE1) / 10000 ;
GO TO ENDB ;

IX2:
IF INDEX3 = 2 THEN GO TO IX3 ;
.
.
.
process accumulator 2
.
.
.
IX16:
.
.
.
process accumulator 16
.
.
.
ENDB:
%
```

### EXTRAC Macro

The EXTRAC macro is used to obtain the PDF header record fields from the DATA1, DATA2 and DATA3 fields after translation from ASCII to EBCDIC. The EXTRAC macro uses the SAS SUBSTRING facility to extract the fields required for data record processing.

```
MACRO EXTRAC
OWNER=SUBSTR(DATA1,1,20) ;
TITLE=SUBSTR(DATA1,21,20) ;
CRMN =SUBSTR(DATA1,41,2) ;
CRYR =SUBSTR(DATA1,44,2) ;
PSRM =SUBSTR(DATA1,47,2) ;
PSRY =SUBSTR(DATA1,49,2) ;
PSTM =SUBSTR(DATA1,55,2) ;
PSTY =SUBSTR(DATA1,58,2) ;
LSTU =SUBSTR(DATA1,61,8) ;
DSC01=SUBSTR(DATA1,97,20) ;
DSC02=SUBSTR(DATA1,121,20) ;
DSC03=SUBSTR(DATA1,145,20) ;
.
.
.
DSC16=SUBSTR(DATA3,73,20) ;
PDFID=SUBSTR(DATA3,97,2) ;
%
```