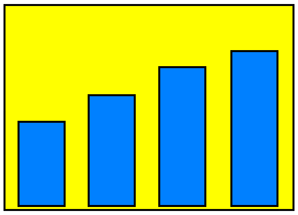
Philadelphia SAS Users Group (PhilaSUG) March 12, 2013
Philadelphia University

POSTERS

Trend Reporting Using the MXG® Trend Performance Database

Neal Musitano Jr.
U. S. Department of Veterans Affairs, Information Technology Center
Philadelphia, Pennsylvania

ABSTRACT



This paper is about my user experience on trend reporting for z/OS server performance using the MXG Trend Performance Database (PDB). In order to implement trend reporting with graphs and charts, the trend pdb must be built first. Thus my user experience of building the trend pdb is included. MXG users build a daily pdb and they use it to produce daily reports for z/OS performance monitoring. However; because of daily tasks or firefighting duties, or deadlines in getting out daily reports or other reasons they sometimes overlook building the trend performance database. Additional reasons for the delaying the trend pdb implementation, include uncertainties of what is involved in building the trend pdb, what to include in it, unfamiliarity of the trend pdb structure and what information to collect for the trends. This paper will address those issues to help users get started with z/OS trend reporting and building the MXG trend performance database.

This paper assumes that you have been using MXG software for a while and that you build daily or weekly pdb's on a regular basis, but that you haven't yet started building a trend pdb. If you have already started with your trend pdb, this paper might seem too elementary, in that it doesn't include enough details on all the trend input datasets. That is because with all the information that can be trended, this paper covers only a fraction of the data files that can be trended.

KEYWORDS – z/OS MXG TREND PDB PLOT GCHART

INTRODUCTION

On the z/OS platform the RMF and SMF record types are produced as part of the daily System Management Facility (SMF) processing. MXG software processes these records into a daily performance database consisting of a hundred or more MXG/SAS datasets. The trend pdb is often built from the weekly pdb and contains summarized performance data. The result is that longer term trending data can be saved in a relatively small file.

Our initial trend reports were produced by reading multiple daily, weekly and monthly pdb's in one job step. But, setting up a trend pdb reduces errors, and provides a means to produce the trend reports with less JCL and less SAS code. Also implementing the trend pdb may eliminate the need for JCL with multiple monthly tape inputs, thus speeding up the process to produce reoccurring trend reports.

MXG software examples and JCL can be adapted to each sites needs. The edited examples work fine for my site, but might not be exactly what is wanted at your site. Thus each user should check the MXG members for what is best suited for their site. But, a working example is usually helpful and should help with initial questions about what is involved with basic trend setup and reporting.

INITIAL THINGS TO DO – PRE- PREPARATION

The MXG source library contains documentation, source code, JCL and information on building daily, weekly, monthly, trend and all MXG pdb information. A scan of the 8,000+ dataset members for MXG Version 30.01 yielded 200 plus of the members with some reference to "trend" thus it might be a bit overwhelming at first to jump into trending.

Very briefly MXG source contains DOCxxxxx members containing documentation, JCLxxxxx members with JCL examples, ANALxxxx members for numerous analysis programs, EXxxxxxx members for user exits, GRAFxxxx members with graph examples, VMACxxxx members with macros, TRNDxxxx members of interest for this paper, and much more. The first member of interest for trend information would be "DOCTREND". This starts you out and you are directed to member to member "PDBTREND" for the initially build of the trend performance database and to member "JCLTREND" to update the trend pdb on a suggested weekly reoccurring schedule.

The MXG trend performance database is the same type of z/OS dataset as the daily performance database. On disk it is a physical sequential (PS) dataset with standard record (FS) format. When allocated using SAS/MXG it will be build with half track blocking but it is up to the user to determine how much space to allocate. The MXG example member "PDBTREND" provides information on building the initial trend pdb.

The JCL example in Figure #1 to build the trend pdb, uses modified member "PDBTREND" with a weekly PDB for input. This reduces the job run time and keeps the build as simple as possible. If your daily pdb's are on disk, the daily pdb space allocation should work ok for a start for the trend pdb space allocation, or doubling it if you are concerned about not allocating enough space. As each site's workload is different, what works for my site might not be best for your location. That may hold many years to a decade's worth of trending data depending on how much you trend and to what detail level.

This paper keeps to a minimum the amount of data that is input to the trend PDB for the initial trend build. The SAS code will select the WEEK.RMFINTRV data and the WEEK.ASUM70PR data. These input files are referenced in the MXG source members TRNDRMFI and TRND70PR. It is very easy to understand the SAS code setup here. TRNDXXXX is used for the SAS code to select the input that you want in the trend PDB. There are over 40+ TRND members covering a multitude of performance datasets, and only two are used in this example.

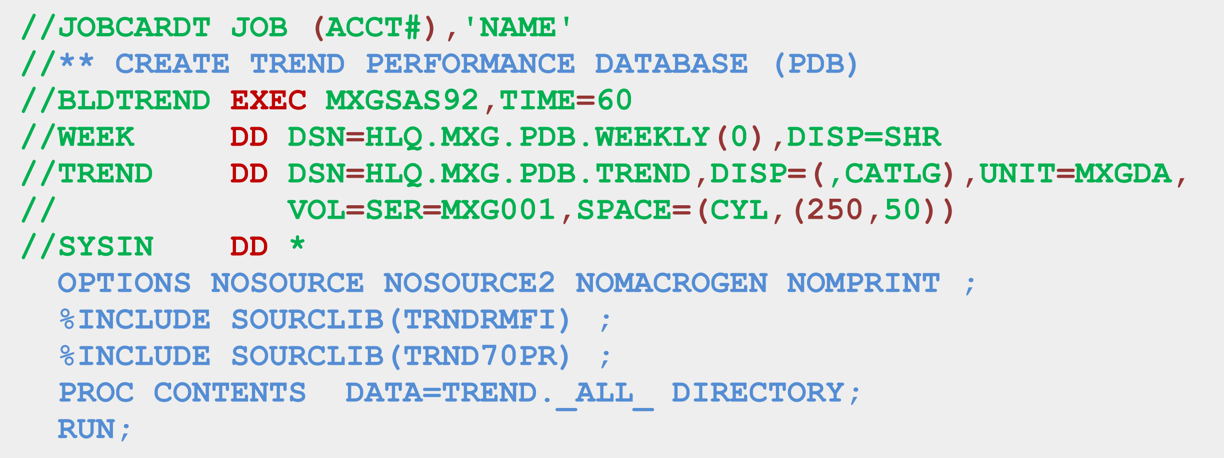
In the Figure #1 JCL example, the modified member "PDBTREND" allocates the MXG trend pdb which is the z/OS dataset HLQ.MXG.PDB.TREND. "HLQ" is an abbreviation for high-level qualifier, the first qualifier of the z/OS dataset. Note, that many SAS datasets (i.e. TRNDRMFI and TRND70PR) can exist within the trend pdb which is a z/OS dataset.

The example will create the trend pdb and populate it with summarized data by processing the selected input from the weekly pdb. The SAS PROC CONTENTS is then used to provide information on the trend pdb, what is in it as well as size and usage. That information can be used to adjust the size allocation of the trend pdb. If the trend pdb space allocation seems too small for future growth, it can at this point be deleted and the job can be run again with a different space allocation.

INITIAL TREND PERFORMANCE DATABASE BUILD – PDBTREND

The Figure #1 JCL example below of member “PDBTREND” was edited using the z/OS Time Sharing Option, Interactive System Productivity Facility TSO/ISPF or simply TSO, and the “HILITE ON” setting was used to make the JCL easier to read. From the edit session the PF1 key can be used for HELP or the PROFILE command can be used to check the edit session setting.

Figure #1 SAS BATCH JOB JCL TO CREATE THE TREND PDB



```
//JOBCARDT JOB (ACCT#),'NAME'
//** CREATE TREND PERFORMANCE DATABASE (PDB)
//BLDTREND EXEC MXGSAS92,TIME=60
//WEEK      DD DSN=HLQ.MXG.PDB.WEEKLY(0),DISP=SHR
//TREND     DD DSN=HLQ.MXG.PDB.TREND,DISP=(,CATLG),UNIT=MXGDA,
//             VOL=SER=MXG001,SPACE=(CYL,(250,50))
//SYSIN     DD *
  OPTIONS NOSOURCE NOSOURCE2 NOMACROGEN NOMPRINT ;
  %INCLUDE SOURCLIB(TRNDRMFI) ;
  %INCLUDE SOURCLIB(TRND70PR) ;
  PROC CONTENTS  DATA=TREND._ALL_ DIRECTORY;
  RUN;
```

If the job ran successfully, then additional weekly data can be input into the trend pdb to populate it with more historical information. That would be accomplished by running the job again for week(-1), week(-2), etc. The newly created trend pdb disposition must now be changed to “DISP=OLD”, since SAS will now be rewriting to the existing SAS trend datasets “TREND.TRNDRMFI” and “TREND.TRND70PR” within the trend pdb. Otherwise the job will fail with an error message indicating “ERROR: WRITE ACCESS TO MEMBER TREND.TRNDRMFI.DATA IS DENIED.”

The MXG TRNDRMFI and TRND70PR source files above make use of the VMXGSUM macro. This macro is used in MXG for all the trend processing. The macro has over 100+ input parameters, and the user can specify numerous code modifications as needed. It is a good idea to take a look at this macro code and documentation to get a better understanding of the MXG processing involved for trending.

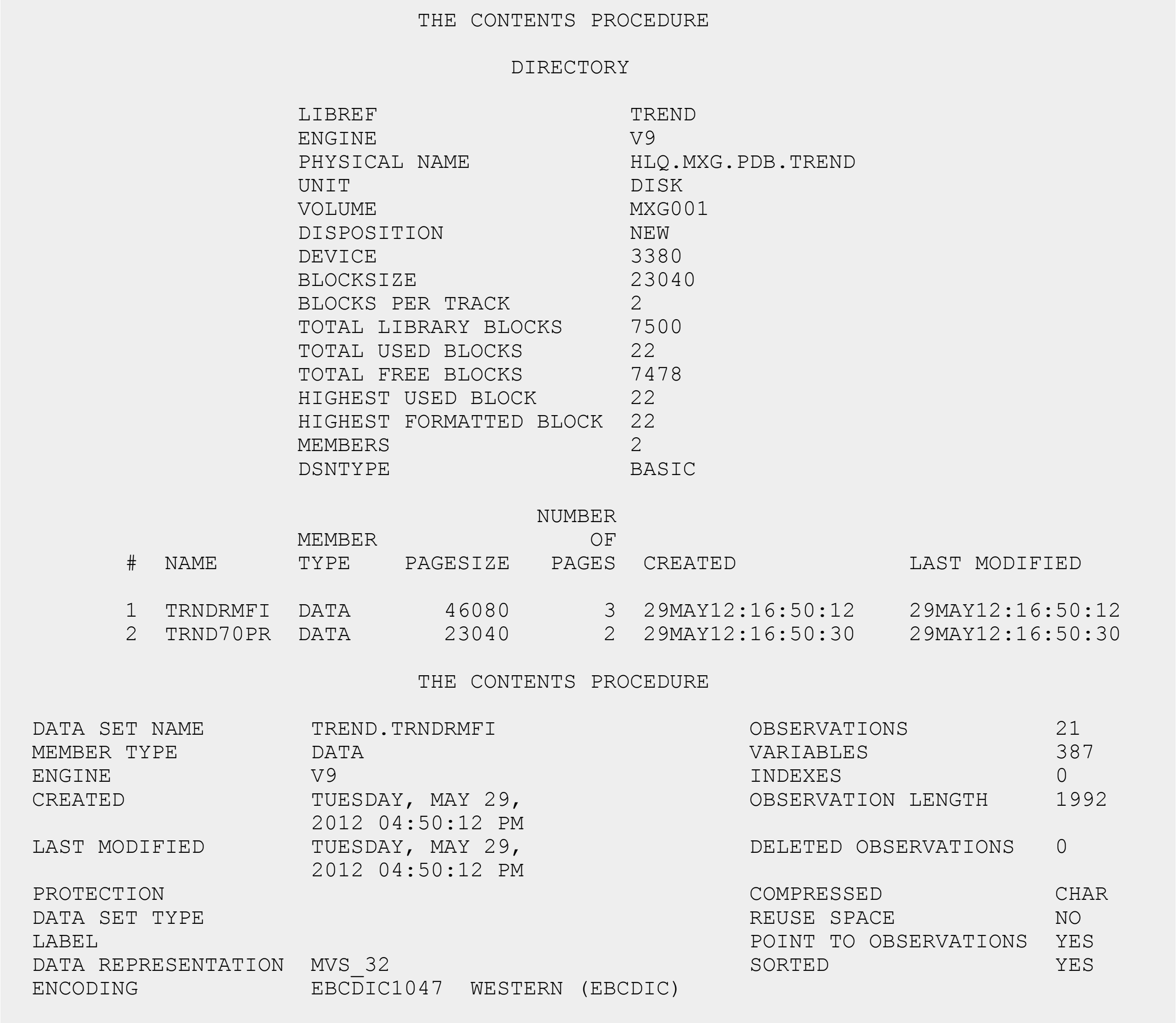
VMXGSUM summarizes trend information for the specified “INTERVAL” in TRNDRMFI and TRDN70PR members. My first attempts in using this were ok, but I couldn’t produce trend reports at daily levels, example for the past 25 days. That is because the default trend interval in these members is &INTTRND and is set by member VMXGINIT to summarize the data at a weekly level, thus the weekly input. That greatly reduces the disk space needed for trend data. For my site, the interval is “DATE” which uses more space, but provides trend information at the daily detail level for each day by shift. Over 100+ SAS datasets are in a daily pdb, but not as many files are needed for trending, reducing the trend disk space to a minimal amount.

Users may also want to check member IMACSHFT, for their shift definitions, since the default trend information is summarized by week and shift. Some sites may want shift indicators of “1”, “2”, “3” or similar for shift definitions. The default MXG shift definitions work great, but may be different from what is wanted at your site. MXG gives you the flexibility to decide, and it’s up to each user site to decide what shifts to define and to report on.

WHAT’S IN THE MXG TREND PERFORMANCE DATABASE

The PROC CONTENTS output from the Figure #1 example is used to review the details of what is in the trend performance database. Figure #2 displays the information from PROC CONTENTS. This is a list of the contents of all the datasets in the example trend pdb and the directory. It’s a good idea to get an understanding of what is in the trend pdb.

Figure #2 SNIP OF OUTPUT FROM THE CONTENTS PROCEDURE



```
                              THE CONTENTS PROCEDURE

                                     DIRECTORY

                     LIBREF                   TREND
                     ENGINE                   V9
                     PHYSICAL NAME            HLQ.MXG.PDB.TREND
                     UNIT                     DISK
                     VOLUME                   MXG001
                     DISPOSITION              NEW
                     DEVICE                   3380
                     BLOCKSIZE                23040
                     BLOCKS PER TRACK         2
                     TOTAL LIBRARY BLOCKS     7500
                     TOTAL USED BLOCKS        22
                     TOTAL FREE BLOCKS        7478
                     HIGHEST USED BLOCK       22
                     HIGHEST FORMATTED BLOCK  22
                     MEMBERS                  2
                     DSNTYPE                  BASIC

                                         NUMBER
                     MEMBER                  OF
           #  NAME       TYPE    PAGESIZE    PAGES  CREATED             LAST MODIFIED

           1  TRNDRMFI  DATA       46080        3  29MAY12:16:50:12    29MAY12:16:50:12
           2  TRND70PR  DATA       23040        2  29MAY12:16:50:30    29MAY12:16:50:30

                              THE CONTENTS PROCEDURE

  DATA SET NAME         TREND.TRNDRMFI                  OBSERVATIONS           21
  MEMBER TYPE           DATA                            VARIABLES              387
  ENGINE                V9                              INDEXES                0
  CREATED               TUESDAY, MAY 29,                OBSERVATION LENGTH     1992
                        2012 04:50:12 PM
  LAST MODIFIED         TUESDAY, MAY 29,                DELETED OBSERVATIONS   0
                        2012 04:50:12 PM
  PROTECTION                                            COMPRESSED             CHAR
  DATA SET TYPE                                         REUSE SPACE            NO
  LABEL                                                 POINT TO OBSERVATIONS  YES
  DATA REPRESENTATION   MVS_32                          SORTED                 YES
  ENCODING              EBCDIC1047  WESTERN (EBCDIC)
```

The example isn’t using all possible trend data members. The TRND70LP, TRNDCEC and TRNDCELP also contain very useful data for reporting on your system and CPU usage. But to keep this paper short, only two TRND members are used.

It would be a good idea to review the newly created trend pdb disk utilization after each job execution, during the initial updates and after the weekly update or daily update, depending on how you implement trend reporting at your site. This will give you an idea how quickly or slowly it is growing as it gets updated.

JCL AND PROGRAM PREPARATION - JCLTREND

The JCL in Figure #1 can also be modified to create an “OLDTREND” performance database. Or as an alternative the “OLDTREND” z/OS dataset can be allocated using the TSO Utilities option 3.2. The OLDTREND pdb is used during reoccurring trend processing.

The MXG member “JCLTREND” is used for updating the trend pdb on a regular schedule. The first activity is to create a backup of the trend pdb. Thus if a problem occurs during the trend update, a backup copy is available without resorting to using system saves and tape backups.

Figure #3 “JCLTREND” TO UPDATE THE TREND PDB – USUALLY WEEKLY

```
//JOBCARDT JOB (ACCT#),'NAME'
//** UPDATE TREND PERFORMANCE DATABASE (PDB)
//BLDTREND EXEC MXGSAS92,TIME=60
//WEEK     DD DSN=HLQ.MXG.PDB.WEEKLY(0),DISP=SHR
//TREND    DD DSN=HLQ.MXG.PDB.TREND,DISP=OLD
//OLDTREND DD DSN=HLQ.MXG.PDB.OLDTREND,DISP=OLD  /*BKUP CPY*/
//SYSIN    DD *
  OPTIONS NOSOURCE NOSOURCE2 NOMACROGEN NOMPRINT ;
  PROC COPY INDD=TREND OUTDD=OLDTREND;
  %INCLUDE SOURCLIB(TRNDRMFI) ;
  %INCLUDE SOURCLIB(TRND70PR) ;
  PROC CONTENTS  DATA=TREND._ALL_ DIRECTORY;
  RUN;
```

The JCL in Figure #3 is a modified version of source member “JCLTREND”, with numerous comments removed, and not as complex, since it isn’t trending CICS and additional performance information. However, it will provide useful trending information on the CPU usage for the system that it is running on. As can be seen, the first activity is to copy the existing trend pdb to a backup.

CLEANUP ACTIVITY

Before continuing you might want to clean up your trend pdb. My initial attempts with different trend members, and setting up my own trend members and interval setting, and using some older inherited shift definitions, and using daily, weekly and monthly inputs left several extra trend member datasets in my testing trend pdb. To clean up the new trend database “PROC DATASETS” can be used to print details on the trend PDB and to delete the excess trend datasets without deleting the entire trend performance database. The code follows and can be useful for the clean-up.

```
PROC DATASETS DDNAME=TREND DETAILS ;
     DELETE TRNDXXXX   ;     /* TREND FILE TO DELETE */
```

INITIAL TREND REPORT

Both members “PDBTREND” and “JCLTREND” include code to plot trending activity. Figure #4 displays the continued code with some modifications. We update the trend pdb every day using the daily pdb as input. That is because we have an interval of “INTERVAL=DATE” specified in our members TRNDRMFI and TRND70PR instead of the &INTTRND default “week” interval. That’s not better or worse than the default, since each individual each site has its own needs. Our user.sourclib trend members were copied from the MXG.SOURCLIB several years ago. Reviewing trend members for updates is a good idea, since a new MXG release is produced every year.

Figure #4 CODE CONTINUATION FROM FIGURE #3

```
OPTIONS PAGESIZE=27 LINESIZE=79 NODATE NONUMBER;
DATA CPUBUSY (KEEP=SYSTEM DATE PCTCPUBY);
SET TREND.TRNDRMFI;
 IF SHIFT = '2';
 DATE=DATEPART(STARTIME);FORMAT DATE DATE9.;
 IF DATE GE TODAY()-7;
PROC PLOT;
 BY SYSTEM;
 PLOT PCTCPUBY*DATE/VAXIS=0 TO 100 BY 10;
```

The output is displayed in Figure #5 using simulated data. The default code would display a plot of over a year’s trending for the prime shift, for each week. The above code was modified to display only seven days of information and uses options to reduce the line and pagesize to make it more readable for this paper. The plot provides a quick review to check that the trend pdb was updated successfully.

Figure #5 OUTPUT FROM SAS PROC PLOT – FOR QUICK REVIEW OF TRENDS.

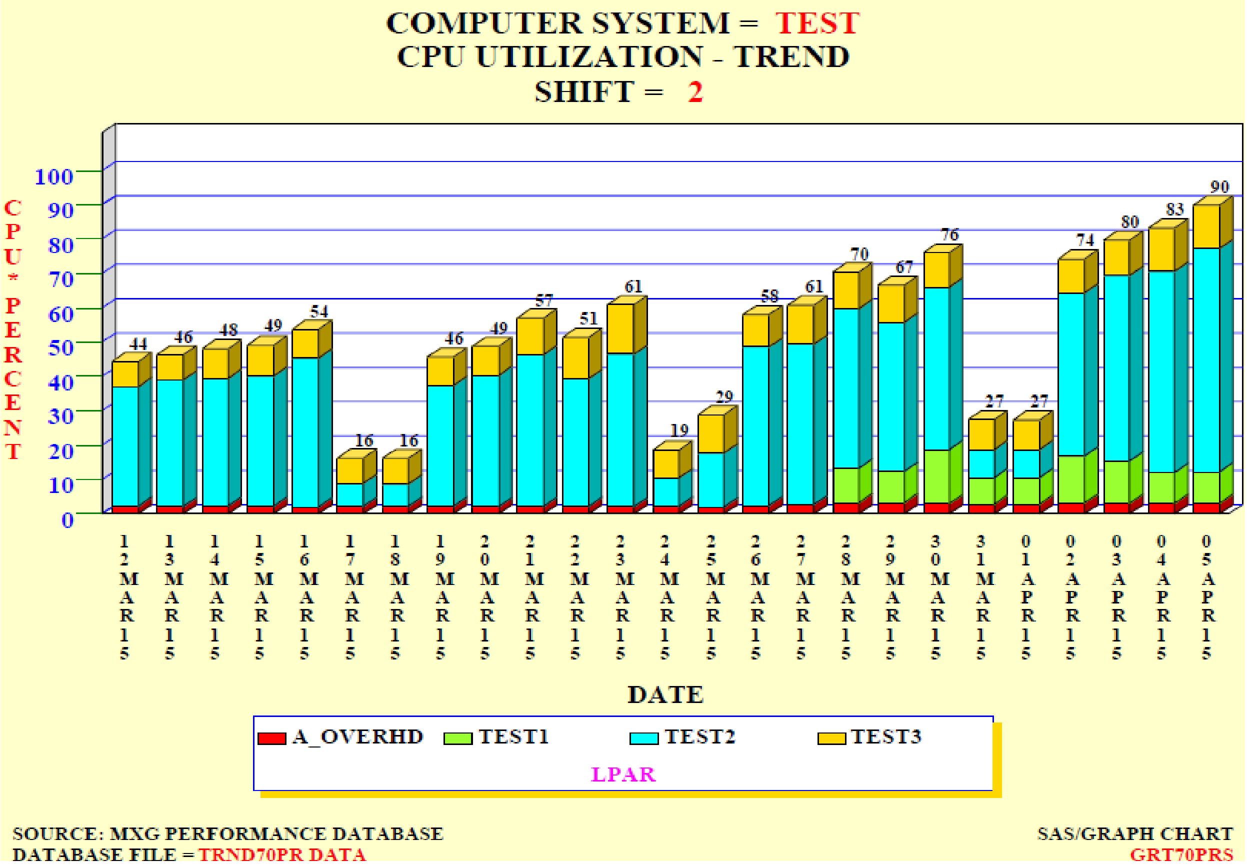
```
------------------------------- SYSTEM*ID=SAS1 --------------------------------

          PLOT OF PCTCPUBY*DATE.  LEGEND: A = 1 OBS, B = 2 OBS, ETC.

PCTCPUBY |
         |
   100.0 +                                                                   A
    90.0 +
    80.0 +                                                        A
    70.0 +
    60.0 +                                             A
    50.0 +                       A          A
    40.0 +            A
    30.0 + A
    20.0 +
    10.0 +
     0.0 +
         |
         --+----------+----------+----------+----------+----------+----------+-
       10SEP2014  11SEP2014  12SEP2014  13SEP2014 14SEP2014 15SEP2014 16SEP2014
                                         DATE
```

TREND REPORTING OUTPUT

Our trend reports are directed to our Unix file structure on the z/OS system. Thus the reports and graphics are available internally to those who need to see them, and they can be e-mailed if needed.

Figure #6 displays a chart of short term trends of CPU Busy usage by each LPAR for simulated test data for a computer system with a handful of LPARS. This chart was directed to a pdf file, which produced very sharp graphics. The code for this example is in the appendix. The chart was produced from the TREND.TRND70PR data that has detailed trend information on LPAR usage.

Figure #6 CPU SHORT TERM TREND CHART – USAGE BY LPAR



CPU Short Term Trend Chart – z/OS® Usage by LPAR

Using simulated data, the chart displays CPU percent busy for the past 25 days for three LPARS on a computer system and the overhead usage. Using #BYVALS, allows the computer system that collected the data and the shift to be displayed in the titles by taking the values from the TRND70PR data.

The TRND70PR data is used for the above chart, since that is the first member that I started for LPAR trend data collection. But users might want to consider using MXG graph examples GRAFLPAR, GRAFRMFI, GRAFTRND and other members TRND70LP, TRNDCEC and TRNDCELP for reporting.

CONCLUSION

The trend build, the update, the BASE/SAS trend plot and the SAS/GRAPH VBAR3D chart examples that are displayed used z/OS SAS and MXG JCL procedures that worked successfully without any problems. The trend performance database is updated by automated operations scheduling software and the subsequent trend reporting plots and charts are produced after the update.

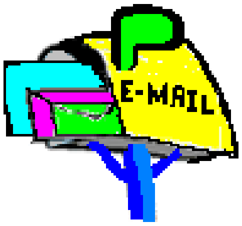
Producing the SAS/GRAPH output via the SAS Output Delivery System (ODS) in a pdf document is an alternative to the GIF or PNG formats and results in very sharp graphics. The display of graphs on the intranet web server improved ease of use in reviewing performance data and continues to increase the use and visibility of MXG and SAS/GRAPH software. Overall productivity increased by making the charts and reports more easily available for review by the technical staff and our managers needing access to the information

REFERENCES

IBM z/OS - MVS System Management Facilities (SMF) – SA22-7630-17
IBM z/OS – Resource Measurement Facility (RMF) User's Guide – SC33-7990-18
IBM Redbooks at http://www.redbooks.ibm.com/redbooks.nsf/redbooks/
Merrill's Expanded Guide to Computer Performance Evaluation Using the SAS System
Merrill's Expanded Guide Supplement and Technical Newsletters for Users of MXG®
MXG Archives at WWW.MXG.COM
Computer Measurement Group Information at http://www.cmg.org/
SHARE Information at http://www.share.org
Philadelphia SAS Users Group Presenter Information at http://www.philasug.org/call4pap.htm
NorthEast SAS Users Group information and proceedings at http://www.nesug.org
SAS Technical Support at http://support.sas.com
SAS-L archives at http://www.listserv.uga.edu
SAS Online Documentation at http://support.sas.com/documentation/onlinedoc
SAS Proceedings at http://support.sas.com/events/sasglobalforum/previous/online.html
SAS Global Forum Guidelines http://support.sas.com/events/sasglobalforum/2012/package.html

CONTACT INFORMATION

Your comments and questions are welcome.



Neal Musitano Jr.
Department of Veterans Affairs
Information Technology Center-284/31
P. O. BOX 7545
Philadelphia, PA. 19101-7545

Phone 215-842-2000 ext. 4102 EMAIL Neal.Musitano@va.gov

Note: A modified test MXG/PDB was used for the information displayed, not to include the actual values of detailed production computer performance data in the paper.

APPENDIX – SAMPLE CODE.

Text can be copied from the PDF document by using the **‘select text tool’** icon on the PDF toolbar, or by a mouse right click and selecting the select tool.

```
/** MEMBER=GRT70PRS                                             **/
/** PROJ:   Z/OS MONITORING VIA MXG                             **/
/** DESC:   25 DAY LPAR SHORT TERM TREND VBAR3D CHART           **/
/** DATE:   JAN 01, 2013                                        **/
/** INPUT:  TRND70PR DATA    OUTPUT:  VBAR3D CHART TO PDF FILE  **/
OPTIONS NOBYLINE NODATE NONUMBER ORIENTATION=LANDSCAPE ;

%LET MY_FILE = TRND70PR;                  /** FILE TO PROCESS   **/

DATA &MY_FILE  ;                          /** DATA STEP         **/
FORMAT DATE DATE7.                        /** FORMAT STATEMENT  **/
       LPAR $CHAR10.;

SET TREND.&MY_FILE ; ;                    /** SET TO READ INPUT **/
    DATE = DATEPART(STARTIME) ;           /** DATE USED IN CHART**/

    IF TODAY() - DATE LE 25 ;             /** 25 DAYS LIMIT BARS**/

     /* ONLY CHART KNOWN LPARS - MAY BE MODIFY AS NEEDED        **/
        LPAR=LP1NAME   ; PCTCPU=PCTL1BY; OUTPUT; /** LPAR1 BUSY **/
        LPAR=LP2NAME   ; PCTCPU=PCTL2BY; OUTPUT; /** LPAR2 BUSY **/
        LPAR=LP3NAME   ; PCTCPU=PCTL3BY; OUTPUT; /** LPAR3 BUSY **/
        LPAR=LP4NAME   ; PCTCPU=PCTL4BY; OUTPUT; /** LPAR4 BUSY **/

        LPAR='A_OVERHD'; PCTCPU=PCTOVHD; OUTPUT; /** OVERHD PCT **/
     /* REVIEW MEMBER TRND70PR FOR DETAILED CALCULATIONS        **/
     /* REVIEW MEMBER VMXG70PR FOR MORE DETAILED CALCULATIONS   **/

PROC PRINT DATA=&MY_FILE;                 /** PRINT FOR REVIEW  **/
RUN;                                      /** RUN THE PROCEDURE **/

                                          /** SORT CHART BYVALS **/
PROC SORT DATA=&MY_FILE; BY SYSTEM SHIFT DATE; RUN:

ODS LISTING CLOSE;                        /** ODS CLOSEOUT      **/

GOPTIONS RESET=GLOBAL                     /** RESET GRAPH OPTS  **/
        GSFMODE=REPLACE                   /** RESET GSFMODE     **/
        DEV=SASPRTC                       /** DEVICE PRINTER    **/
        GUNIT=PCT                         /** GRAPH UNITS PCT   **/
        ROTATE=LANDSCAPE                  /** LANDSCAPE SETTING **/
        HTEXT=2.0 FTEXT='Times/bold'      /** TEXT SETTINGS     **/
        XMAX= 9.0IN YMAX=5.0IN            /** GRAPH X Y SIZE    **/
        CBACK=CXFFFFCC ;                  /** BACKGRD COLOR     **/

ODS PDF  style=PRINTER                    /** PDF AND STYLE     **/
         notoc                            /** NO TOC            **/
         author='your name here'          /** AUTHOR            **/
         title='SAS/MXG GRAPHIC REPORTS'  /** TITLE             **/
         keywords='SAS MXG Z/OS GRAPH'    /** PDF KEYWORDS      **/
         subject='SHORT TERM CPU TREND'   /** PDF SUBJECT       **/
         FILE='/mxg/trnd/short_lpar_trnd.pdf'; /** PDF FILE     **/
```

```
 /** SAS GRAPH PROC GCHART FOR 3D VERTICAL BAR CHART            **/
 /** DO NOT INCLUDE OFFLINE LPARS WITH MISSING VALUES FOR NAME  **/

 PROC GCHART DATA=&MY_FILE (WHERE=(LPAR NE ' ')); BY SYSTEM SHIFT;
 FORMAT PCTCPU 4.0  ;

 /** SAS GRAPH TITLE STATEMENTS FOR CHART TITLES                **/

 TITLE1 H=3.6 C=BLACK 'COMPUTER SYSTEM = ' C=RED ' #BYVAL1' ;
 TITLE2 H=3.6 C=BLACK 'CPU UTILIZATION - TREND'             ;
 TITLE3 H=3.6 C=BLACK 'SHIFT =  '          C=RED ' #BYVAL2 ';

 /** VERTICAL AND HORIZONTAL AXIS  AND FOOTNOTE STATEMENTS      **/

AXIS1 WIDTH=2                        /** VERTICAL RESPONSE AXIS **/
      LABEL=(H=2.8 COLOR=RED ROTATE=90 ANGLE=-90
             'CPU*PERCENT')
      MAJOR=(H=2 COLOR=GREEN W=2)    /** AXIS MAJOR TICK MARKS  **/
      MINOR=NONE                     /** AXIS MINOR TICK MARKS  **/
      ORDER=(0 TO 100 BY 10)         /** CPU AXIS ORDER INFO    **/
      VALUE=(H=2.8 COLOR=BLUE);      /** CPU AXIS VALUE INFO    **/

AXIS2 WIDTH=2                        /** HORIZONTAL DATE AXIS   **/
      LABEL=(HEIGHT=2.9 C=BLACK)     /** DATE LABEL INFO        **/
      VALUE=(H=2.2 COLOR=BLACK);     /** DATE AXIS VALUES INFO  **/

FOOTNOTE1 HEIGHT=2.2                 /** FIRST GRAPH FOOTNOTE   **/
          J=L '  SOURCE: MXG PERFORMANCE DATABASE'
          J=R 'SAS/GRAPH CHART  ';

FOOTNOTE2 HEIGHT=2.2                 /** SECOND FOOTNOTE        **/
          J=L '  DATABASE FILE = '   /** JUSTIFY LEFT           **/
              C=RED  "&MY_FILE DATA" /** NOTE DOUBLE QUOTE      **/
          J=R 'GRT70PRS  ';          /** SOURCE MEMBER NAME     **/

 /** PATTERN AND LEGEND STATEMENTS FOLLOW                       **/

PATTERN1 COLOR=RED      V=SOLID;     /** RED - FIRST BAR COLOR  **/
PATTERN2 COLOR=CX99FF33 V=SOLID;     /** BRIGHT GREEN COLOR     **/
PATTERN3 COLOR=CYAN     V=SOLID;     /** CYAN - THIRD BAR COLOR **/
PATTERN4 COLOR=GOLD     V=SOLID;     /** GOLD - FOURTH COLOR    **/

LEGEND1 VALUE=(HEIGHT=2.5 C=BLACK) ACROSS=4
        CFRAME=WH CBORDER=BLUE CSHADOW=GOLD
        LABEL=(H=2.5 C=MAGENTA POSITION=(BOTTOM CENTER))
        ;
VBAR3D DATE / TYPE=SUM SUMVAR=PCTCPU
            DISCRETE                 /** EACH DATE CHARTED      **/
            SUM                      /** PRINT SUM ON BAR       **/
            RAXIS=AXIS1              /** CPU RESPONSE AXIS      **/
            MAXIS=AXIS2              /** DATE AXIS              **/
            REF=90 AUTOREF CREF=BLUE /** REFERENCE LINES        **/
            SHAPE=BLOCK              /** SHAPE INFO             **/
            CFRAME=WHITE             /** COLOR OF FRAME         **/
            SUBGROUP=LPAR            /** SUBGROUP               **/
            LEGEND=LEGEND1 ;         /** LEGEND TO USE          **/
RUN; QUIT;                           /** RUN AND CLOSE GCHART   **/
ods _all_ close ;                    /** CLOSE ODS              **/
```