

E-Mail z/OS CICS Exception Reports in Plain Text Format Using DATA _NULL_ and MXG® Software

Neal Musitano Jr.

**U. S. Department of Veterans Affairs, Information Technology Center
Philadelphia, Pennsylvania**

ABSTRACT

This paper describes the process of e-mailing z/OS CICS exception reports from a z/OS mainframe platform by using the SAS® e-mail engine, DATA _NULL_ and MXG software. The plain text report is generated and e-mailed to z/OS and CICS software support technicians from a z/OS batch job that uses the BASE SAS JCL procedure.



Z/OS software support personnel are able to identify potential software problems very quickly by receiving a prompt e-mail notice from the daily job that examines the MXG CICSTRAN dataset for any unusual CICS transaction exception conditions. Additionally, if no CICS exceptions are detected then the job ends without sending a blank e-mail. This is accomplished by checking the SAS working dataset of z/OS CICS transaction abends for an observation count. This also eliminates unnecessary spam type e-mail, since the e-mail is only sent if an exception is detected.

Although, this paper specifically details e-mail of z/OS CICS transaction exceptions, the plain text e-mail format can be used with data from any input file.

KEYWORDS – z/OS MXG CICS E-MAIL DATA _NULL_

INTRODUCTION

On the z/OS platform the CICS type 110 monitor records are written to DASD as part of daily System Management Facility (SMF) processing. MXG software then processes the CICS monitor record into a Performance Database (PDB) consisting of several MXG/SAS datasets. The largest of these is the CICSTRAN dataset. This dataset can become very large with millions of observations and hundreds of variables containing detailed information every CICS transaction. The dataset is output to tape at many z/OS sites. But with some MXG customization the number of variables in the dataset can be reduced from several hundred per observation, to just what is needed at each computer center.

There are several additional MXG SAS datasets with CICS information. The CICSTRAN dataset is used in this paper since it has information on individual online CICS transaction aborts that are reported on in the example e-mail. Each MXG PDB dataset contains performance variables in a SAS readable format that can be processed by SAS jobs. MXG also provides examples for analysis and plots using BASE SAS and SAS/GRAPH.

SAS CONFIG PREPARATION

To use a z/OS batch SAS job for e-mail the SAS system configuration must identify the e-mail system that z/OS will use. The option of EMAILSYS=SMTP to use the Simple Mail Transfer Protocol (SMTP) e-mail must be specified. Note: SMTP is also the name usually assigned to the z/OS started task that many z/OS sites use for e-mail.

Additionally the SMTP server must be identified. This is accomplished by using the EMAILHOST option in the SAS configuration file. This example identifies the EMAILHOST by using the z/OS host Internet Protocol (IP) address that is used by the z/OS Transmission Control Protocol (TCP) TCP/IP profile. You may have to check with your server or system people to verify your server address. There are additional options listed in the SAS online documentation, but for many sites specifying these two SAS configuration options is all that is needed to use z/OS SAS e-mail.

Figure #1 SAS z/OS CONFIG OPTIONS FOR E-MAIL

```
* CHANGE TO WHERE YOUR SMTP EMAIL SERVER IS RUNNING
EMAILSYS=SMTP
EMAILHOST=10.123.456.7
* EMAILHOST=152.122.242.2   <= comment out old host address
```

Z/OS JOB CONTROL LANGUAGE (JCL) SETUP & PREPARATION

The daily z/OS MXG processing builds the MXG Performance Database and the CICSTRAN dataset used in this example, i.e. the z/OS dataset HLQ.MXG.PDB.CICSDAY(0). HLQ is an abbreviation for high-level qualifier, the first qualifier of the z/OS dataset that contains the MXG CICSTRAN dataset. Note, that many SAS datasets can exist within one z/OS MXG Performance Database dataset.

The newly built daily MXG CICSTRAN dataset can now be processed using the MXGSASV9 JCL procedure or with the standard SAS procedure that is shipped with SAS version 9 and used in this example.

The Figure #2 JCL example below uses LIBRARY and SOURCLIB DD cards to reference the MXG FORMATS and SOURCLIB datasets, these files are used for most MXG processing. The CICSTRAN DD card is used to reference the MXG CICSTRAN.CICSTRAN dataset.

Figure #2 SAS BATCH JOB JCL TO E-MAIL z/OS CICS EXCEPTION REPORT

```
//JOBNAME JOB
//*****
//* JCL FOR SAS CICS EXCPTION E-MAIL
//*****
//STEP1 EXEC SASV913
//LIBRARY DD DSN=HLQ.MXG.FORMATS,DISP=SHR
//SOURCLIB DD DSN=HLQ.MXG.SOURCLIB,DISP=SHR
//CICSTRAN DD DSN=HLQ.MXG.PDB.CICSDAY(0),DISP=SHR
//SYSIN DD *
```

If the e-mail options are not in the SAS configuration file, another alternative is to specify the options in the batch job JCL. This is also a good way to test the option settings before updating the SAS configuration file with new settings. Ideally updating the e-mail setting in one place, i.e. the configuration file is preferred over updating a large number of JCL setups.

Figure #3 SPECIFYING Z/OS E-MAIL OPTIONS IN THE JOB JCL

```
//STEP1      EXEC SASV913,  
//  OPTIONS='EMAILSYS=SMTP EMAILHOST=10.123.456.7'  
//LIBRARY   DD DSN=HLQ.MXG.FORMATS,DISP=SHR  
//SOURCLIB  DD DSN=HLQ.MXG.SOURCLIB,DISP=SHR  
//CICSTRAN  DD DSN=HLQ.MXG.PDB.CICSDAY(0),DISP=SHR  
//SYSIN     DD *
```

PROCESS THE INPUT DATASET

The input data is the MXG CICSTRAN dataset, with transaction activity details on z/OS CICS transactions. This is usually a large dataset and may be on tape at many z/OS sites, but fortunately very few production transactions abort. The option of MSGLEVEL=I displays additional informational messages in the log about dataset processing, index usage, sorting and more.

For this e-mail exception report the selection criteria are non-zero transaction abort codes from two production CICS regions i.e. CICSPRD1 and CICSPRD2. Note: zero in this example is eight character bytes of hexadecimal zero's. Additional edits or filters of the CICS transactions to select can also be implemented as each z/OS computer site's needs vary.

Figure #4 READ THE CICSTRAN DATASET AND SELECTION CRITERIA

```
OPTIONS MSGLEVEL=I; /* additional information notes */  
  
DATA CICS_ABENDS;  
  
    SET CICSTRAN.CICSTRAN (WHERE=(ABCODE NE '0000000000000000'x));  
    IF APPLID IN ( 'CICSPRD1', 'CICSPRD2' ) ;  
  
RUN;
```

The above example creates a working CICS_ABENDS dataset i.e. WORK.CICS_ABENDS. The "WORK" part of the dataset name is a default and doesn't have to be specified in the references that follow. This dataset contains information on the CICS transactions that aborted, if any.

IF NOTHING TO E-MAIL -- THEN END THE JOB

Any observations? The next step is to check the working dataset of CICS transaction abends for an observation count. This step uses DATA _NULL_ and the SET statement NOBS option to acquire the dataset observation count from the descriptor portion of the CICS_ABENDS dataset before the DATA _NULL_ step executes. If the CICS_ABENDS dataset observation count is zero, the program ends with a return code of "04". This eliminates any unneeded spam or junk e-mail.

Figure #5 USING DATA _NULL_ , SET STATEMENT AND NOBS OPTION

```
/* BEFORE THE SET STATEMENT EXECUTES, THE OBSERVATION COUNT */
/* IS AVAILABLE TO THE DATA STEP, VIA THE NOBS OPTION          */

/* IF ZERO OBSERVATIONS THEN ABORT E-MAIL JOB EXECUTION      */

DATA _NULL_;

    /* DISPLAY THE OBSERVATION COUNT IN THE LOG */
    PUT "NUMBER OF TRANSACTION ABENDS IS = " ABENDCOUNT ;

    /* ABORT JOB IF ZERO OBSERVATIONS          */
    IF ABENDCOUNT =0 THEN ABORT RETURN 04;

    /* PLACE THE OBSERVATION COUNT IN ABENDCOUNT VARIABLE */
    SET CICS_ABENDS NOBS=ABENDCOUNT ;

STOP;                               /* STOP THE STEP      */
RUN;                                 /* EXECUTE THE STEP   */
```

There are several ways to code the DATA _NULL_ step to check for zero observations. Although, it takes some time to get used to and to understand that the “*abendcount*” variable used in the above example, gets assigned the total observation count of the CICS_ABENDS dataset before the step actually starts to execute.

The ABORT statement with the RETURN option will terminate the job, if there are no observations. For most of the program executions the “*abendcount*” variable’s value is zero, and the program displays the zero observation count in the SAS log before ending.

If there are observations in the CICS_ABENDS dataset, there is no need for the DATA _NULL_ step to continue processing every transaction in the dataset, thus the STOP statement is used to stop the step.

In the event of an unusual CICS activity or event, many transactions could abort unexpectedly due to various reasons. Thus, the options statement is now used to limit the dataset observations that are processed in the steps that follow it. This keeps the e-mail report to a reasonable size in the event of excessive transaction aborts. Optionally PROC OPTIONS is also used to display additional option settings.

Figure #6 OPTIONS STATEMENT TO LIMIT OBSERVATIONS

```
OPTIONS OBS=100 ;          /* LIMIT OBSERVATIONS TO 100          */
PROC OPTIONS;              /* OPTIONAL DISPLAY OPTIONS SETTINGS */
```

NEXT DEFINE THE E-MAIL FILE AND E-MAIL OPTIONS

The FILENAME statement is used to define MYFILE as an EMAIL file type. With the FILENAME statement several options for defining the E-MAIL file are available. The FROM option specifies an e-mail sending address. The e-mail address doesn't have to be a real e-mail address, but it should relate to the subject and be easily identifiable to the eventual recipient. With SAS version 9 the sender option can be used to respecify the address, and helps avoid some e-mail address problems.

Figure #7 DEFINE THE E-MAIL FILE AND E-MAIL OPTIONS

```
/* DEFINE THE EMAIL FILE, TO, FROM, SPECIFICS, ETC */

FILENAME MYFILE EMAIL
  FROM=( 'SAS_MXG_CICS_MONITOR@VA.GOV' )
  SENDER=( 'SAS_MXG_CICS_MONITOR@VA.GOV' ) /*V9 OPTION */
  TO=( 'Neal.Musitano@va.gov'
        'CICS.PERSON@VA.GOV'
        'ZOS.PERSON2@VA.GOV' )
  REPLYTO=( 'Neal.Musitano@va.gov' )
  SUBJECT='ALERT CICS EXCEPTION REPORT'
  TYPE='TEXT/PLAIN' ;
```

Using the REPLYTO option to define an e-mail address that the e-mail recipients can reply to is a good feature. Thus, if our software support personnel, who receive the e-mail report, have a question about the e-mail or want more details on future e-mails, they have a person to contact.

The TYPE of TEXT e-mail is PLAIN and not HTML. The e-mail report is sent as a plain text report and doesn't use the Output Delivery System (ODS) that would often be used with "PROC PRINT" or "PROC REPORT" reports.

USING DATA _NULL_ AND PUT STATEMENTS FOR THE REPORT

DATA _NULL_ is used to create the plain text e-mail report using PUT statements. The first PUT statements create the report titles and are followed by detailed transaction data.

Figure #8 DATA _NULL_ AND PUT STATEMENTS TO CREATE THE REPORT

```
DATA _NULL_;
  FILE MYFILE;                /* MYFILE IS THE EMAIL FILE */
  TODAY = TODAY();            /* DATE TIME INFORMATION */
  TIME = TIME();

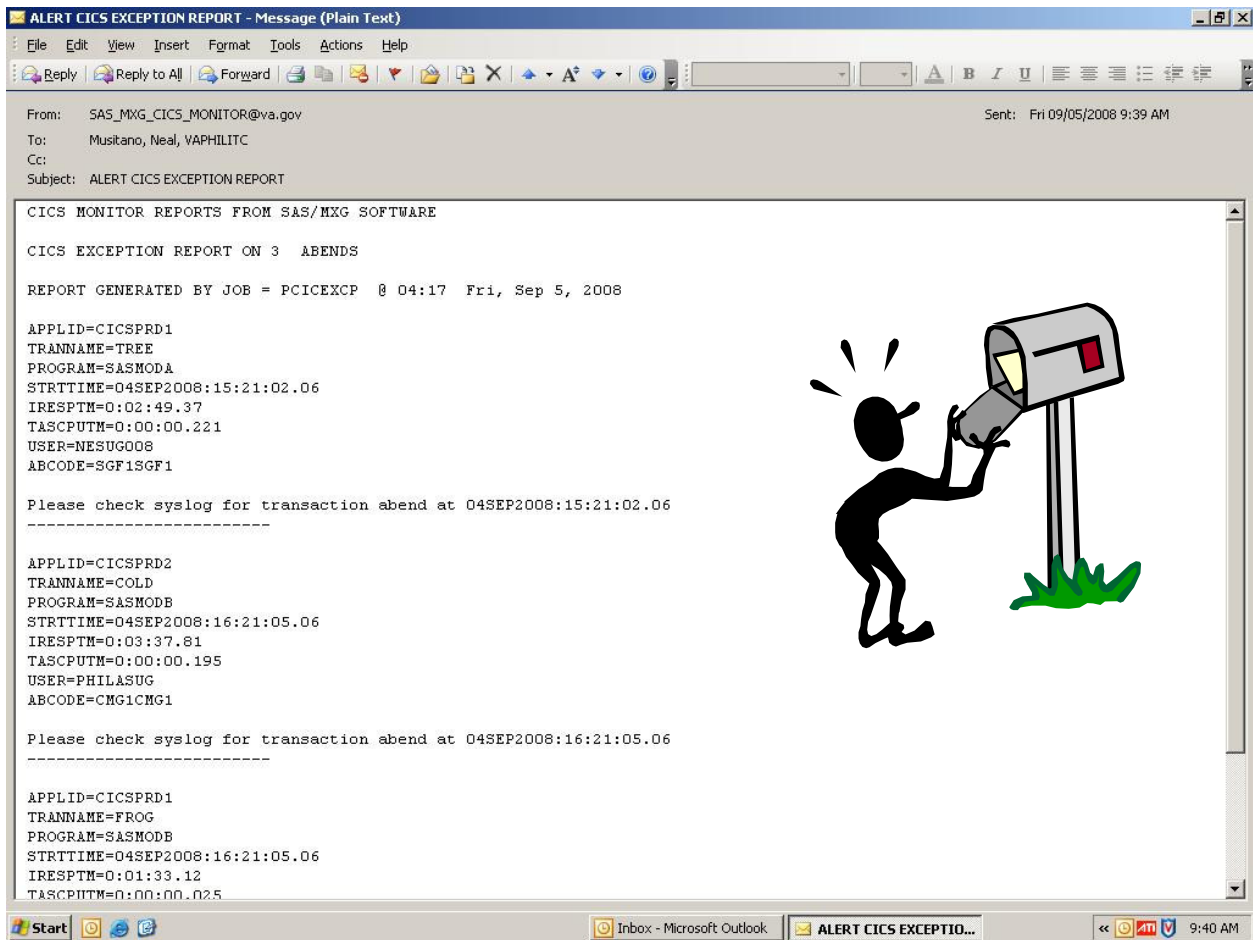
  PUT 'CICS MONITOR REPORTS FROM SAS/MXG SOFTWARE' / ;
  PUT 'CICS EXCEPTION REPORT ON ' ABENDCOUNT ' ABENDS' /;
  PUT "REPORT GENERATED BY JOB = &SYSJOBID"
    ' @ ' TIME TIME5.0 ' ' TODAY WEEKDATE18.;
  PUT ' ';

  DO WHILE ( LASTTRAN= 0);
    SET CICS_ABENDS NOBS=ABENDCOUNT END=LASTTRAN ;
    PUT APPLID=;              /* CICS REGION */
    PUT TRANNAME = ;         /* TRANSACTION NAME */
    PUT PROGRAM = ;         /* APPLICATION PROGRAM NAME */
    PUT STRTTIME = ;        /* TRANSACTION START TIME */
    PUT IRESPTM = ;        /* INTERNAL RESPONSE TIME */
    PUT TASCPUTM = ;        /* TASK CPU TIME */
    PUT USER= ;             /* IDENTIFY THE USER */
    PUT ABCODE = ;         /* FirstLast ABORT CODE */
    PUT ' ';
    PUT 'Please check syslog for abend at ' strttime;
    PUT '-----' ;
    PUT ' ';
  END;                        /* END OF DO WHILE LOOP FOR LASTTRAN */
  STOP;                       /* STOP THE STEP */
RUN;
```

The PUT statements creating the titles display the job creating the report and the time the job started. Within the DO WHILE loop the CICS_ABENDS data is processed and the PUT statements produce the plain text report with details on the CICS region, abending transaction name, program name and more.

Using the PUT statement with the VARIABLE= format simplifies the program code and produces an easy to read output format with the variable name and its value. The sample report is displayed on the next page in Figure #9.

Figure #9 E-Mail z/OS CICS Exception Report



SCREEN PRINT OF THE E-MAIL Z/OS CICS EXCEPTION REPORT

Figure #9 above displays a print screen of the z/OS e-mailed plain text z/OS CICS exception report. The report lists the reporting job and date time information. The abending transaction details are printed. The print includes the selected variable names and values of interest to CICS support software specialists.

The report although in plain text, provides software support personnel the information that they requested. Additionally, the report can be easily expanded to include displaying additional CICS transaction variables if needed or requested.

The Screen Beans clip art is used to highlight this page and isn't included in the actual e-mail report.

CONCLUSION

The z/OS Base SAS JCL procedure is used in this example, and daily CICS exception checking job is scheduled by automated software. Using SAS and MXG software for monitoring z/OS CICS performance for exception conditions is an efficient use of our computer resources, software and personnel.

Using SAS and DATA _NULL_ allowed the plain text e-mail reports to be quickly developed and implemented in a quick, efficient and very timely manner. The z/OS Base SAS JCL procedure is easily used to generate and e-mail the CICS exception reports to our software support technicians.

Producing and e-mailing the z/OS CICS exception report to our software support technicians provides them with the plain text e-mail reports on time, eliminating lost and misrouted printed reports.

REFERENCES

IBM z/OS - MVS System Management Facilities (SMF) – [SA22-7630-13](#)
IBM z/OS – Resource Measurement Facility (RMF) User’s Guide – [SC33-7990-11](#)
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 <http://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 Presentation Guidelines <http://support.sas.com/events/sasglobalforum/2008/package>

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 FAX 215-381-3456 EMAIL "Neal.Musitano@va.gov"

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.

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.