

MP CONNECT: Warp Engine for SAS (Multi-Processing in the Sun Solaris Environment).

Pablo J. Nogueras

CitiFinancial International, Risk Management Technology, Irving, Texas

Abstract

When you are assigned a project, the first question asked by the assignor is not “How will you program the project?”, is not “What kind of Quality Control will you use?”, or is not “How much data will you use?”. The question asked is “How FAST can you get me the results?”. There are various programming techniques in SAS that allow one to increase execution speed. One such technique is the use of Parallel Processing or Multi-Processing, that is the execution of self-contained tasks simultaneously. This paper will demonstrate the use of MP CONNECT (part of SAS/CONNECT) to decrease execution time SAS programs.

Introduction

MP Connect is a feature of SAS/CONNECT that allows a programmer to take advantage of their multi-processor box or processors connected via a network. MP Connect first appeared in SAS version 8. It has continued with various improvements through SAS versions 8.1, 8.2, 9.0, and 9.1.3.

My objective is twofold: Examine the capabilities of MP Connect and apply those capabilities to a real-world application.

MP Connect

MP Connect has the capability to reduce processing time by sub-dividing programming tasks across 2 or more processors. In theory, one should reduce the amount of processing time by the amount of processors. Thus, 2 processors should reduce time by 2, 3 processors by 3, etc.

However, processors are not the only part of our computing systems. There is I/O and system overhead that must be accounted for. When these are taken into account, the relationship begins linear and then begins to “flatten out” as more overhead processing is required as more processors are added.

Locations of MP CONNECT documentation are provided below:

- SASV8 Online DOC path: SAS/CONNECT and SAS/SHARE, SAS/CONNECT User's Guide, Changes and Enhancements, Version 8 Multi-Process (MP) CONNECT
- What's New in SAS Software for Release 8.1, SAS/CONNECT
- What's New in SAS Software for Release 8.2, SAS/CONNECT
- SAS HELP, SAS/CONNECT

All SAS Documentation is Copyright 1999 by SAS Institute Inc., Cary, NC, USA. All rights reserved.

Data Values Program

At Citigroup, we created a program to perform rudimentary analysis on all the variables in a dataset. This program is used when first developing a load program to verify the values. It is also, used on a monthly, quarterly, or yearly period to QA data values within our datasets. The program analyzes character variables with frequency counts and numeric variables with PROC Univariate.

Depending on the number of observation and the number of variables (rows and columns for you newer programmers), the time to execute the program varies. Since the program executes all of its SAS statements sequentially, we saw many execution times of 8 hours, 16, even 24 hours.

The Data Values program is included in the Appendix.

Data Values Program – Sample Print

Data represents values for JUN2004 data ----- run on: 110CT04

09:38 Monday, October 11, 2004 1

The UNIVARIATE Procedure
Variable: XXXXXXXXXX (XXXXXXXXXXXXXXXXXXXX AMOUNT)

Moments

N	215488	Sum Weights	215488
Mean	651.523315	Sum Observations	140395456
Std Deviation	3811.21751	Variance	14525378.9
Skewness	15.7260807	Kurtosis	349.46136
Uncorrected SS	3.2215E12	Corrected SS	3.13003E12
Coeff Variation	584.970241	Std Error Mean	8.21017075

Basic Statistical Measures

Location		Variability	
Mean	651.5233	Std Deviation	3811
Median	100.0000	Variance	14525379
Mode	0.0000	Range	203035
		Interquartile Range	183.38000

Tests for Location: Mu0=0

Test	-Statistic-	-----p Value-----
Student's t	t 79.35564	Pr > t <.0001
Sign	M 100394.5	Pr >= M <.0001
Signed Rank	S 1.018E10	Pr >= S <.0001

Quantiles (Definition 5)

Quantile	Estimate
100% Max	199394.35
99%	11277.28
95%	2276.36
90%	563.15
75% Q3	213.38
50% Median	100.00
25% Q1	30.00
10%	5.00
5%	0.00
1%	0.00
0% Min	-3641.04

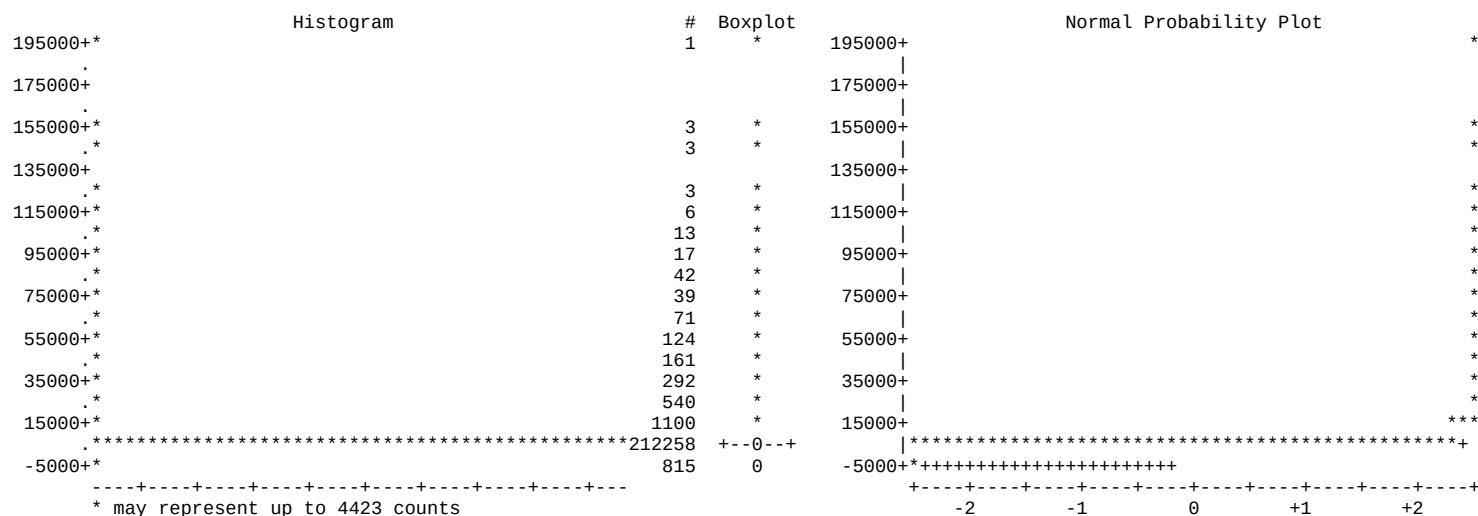
Extreme Observations

-----Lowest-----		-----Highest-----	
Value	Obs	Value	Obs
-3641.04	107248	147005	198820
-2859.00	134928	153140	22348
-2500.00	104325	154089	41114
-2261.17	135603	159244	203860
-1820.00	161374	199394	207147

Data Values Program – Sample Print (continued)

Data represents values for JUN2004 data ----- run on: 11OCT04

09:38 Monday, October 11, 2004



Report represents values for JUN2004 data,			10:39 Monday, October 11, 2004	1
Variable: XXXXXX Description: XXXXXXXXXXXX CODE				
Less than 50 Discrete Values --	25 Discrete Values --	215,488 Total Population		

XXXXXXXXX CODE	# of Records with Value	% of Total Population
06	70,614	32.7693
02	36,902	17.1249
10	32,641	15.1475
44	22,602	10.4888
07	13,129	6.0927
09	10,351	4.8035
17	10,058	4.6675
00	8,125	3.7705
08	6,356	2.9496
19	3,184	1.4776
39	472	0.2190
03	312	0.1448
42	198	0.0919
11	174	0.0807
04	105	0.0487
01	102	0.0473
05	87	0.0404
20	48	0.0223
13	10	0.0046
12	5	0.0023
34	5	0.0023
18	4	0.0019
41	2	0.0009
15	1	0.0005
28	1	0.0005

N = 25

Data Values Program – MP CONNECT

With the release of SASV8 and the addition of asynchronous processing in SAS/CONNECT, I researched MP CONNECT and how it could be applied to our SAS programs. With the help of David Cedillo, we came up with a revision to our Data Values program. We knew that each process could “analyze” each variable independently of the other variables.

Our test showed that increasing the number of processes, directly impacted the time through a divisor effect. $\text{New Time} = \text{Old Time} / \# \text{ of Processes}$. Now this is not an exact formula as there is overhead associated with each new process, but can be used as an educated guess.

The Data Values program with MP CONNECT is included in the Appendix.

Data Values Program – Benchmark

Below is the environment and timings (before MP CONNECT) associated with Data Values program.

- Sun E10K
- Solaris Version 8
- 40 Processors
- 52G Memory
- 324G of SAS Work Space Available (in 4 groups - 150, 70, 62, 62)
- 215,488 Observations
- 1357 Variables
- NOTE: SAS Institute Inc., SAS Campus Drive, Cary, NC USA 27513-2414
- NOTE: The SAS System used:
 - real time 8:55:40.31
 - cpu time 8:55:06.96

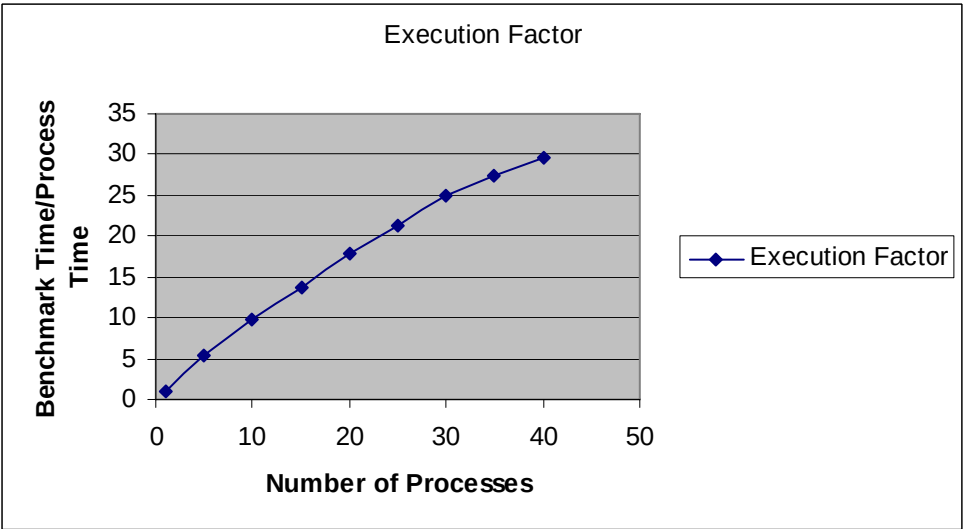
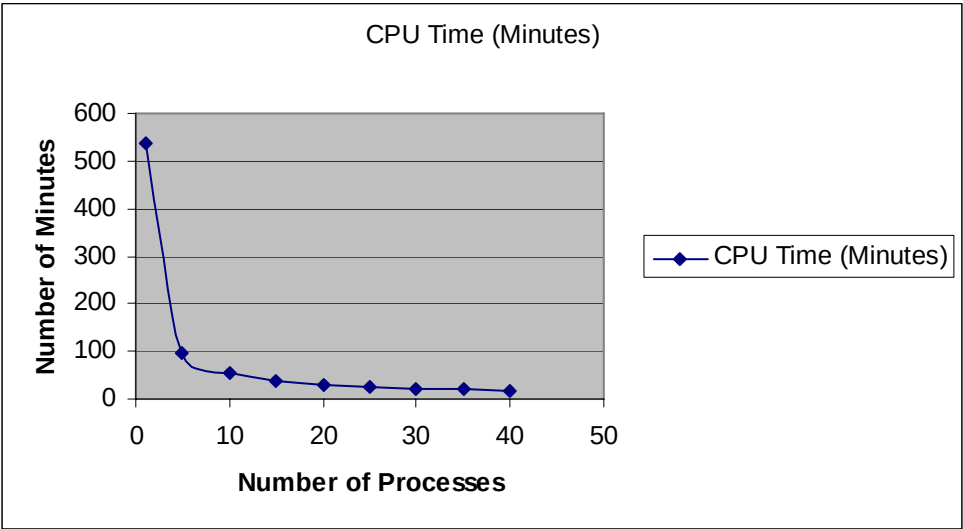
Data Values Program – MP CONNECT – Benchmark

The table below examines the relationship between the number of process, the CPU time, and the Execution Factor. The execution factor is the old time divided by the new time. We can clearly see the effect of I/O and system overhead when we increase usage beyond 10 processors.

Number of Processors	CPU Time (Hours)	CPU Time (Minutes)	Execution Factor
1	8.917	535.02	1
5	1.633	97.98	5.460502143
10	0.9	54	9.907777778
15	0.65	39	13.71846154
20	0.5	30	17.834
25	0.417	25.02	21.38369305
30	0.358	21.48	24.90782123
35	0.325	19.5	27.43692308
40	0.3	18	29.72333333

Data Values Program – MP CONNECT – Benchmark (Continued)

I have included graphs to illustrate the relationships between number of processors and execution time and between number of processors and Execution Factor (Actual vs. expected scaling factor).



Conclusion

MP Connect works as a tool to DECREASE execution time. The example I presented worked on a single dataset, manipulating many variables. One could also use MP CONNECT logic in programs that use independent datasets e.g. You need to look at 36 months of history for customers using your monthly datasets. You may program the task using 12 MP CONNECT session at one time and thus, reducing your execution time approximately 12 times.

Issues that one must take account when using MP CONNECT:

- It will work on a single processor box. NOT RECOMMENDED.
- The more processes you execute simultaneously, the more memory, I/O, and disk resources are used.
- It is not recommended you program MP CONNECT to execute more tasks than processors on your box.
- SAS does not have an option to limit the number of MP CONNECT processes that can be executed.
- One must work with the users to avoid scenarios such as: 25 users each using 25 MP CONNECT processes on a 30 processor box.
- Work Library – depending on the SASCMD= used and how you allow people to allocate WORK Libraries, the default of creating each MP CONNECT process in the same WORK Library may cause I/O or Space issues.

Contact Information

Pablo J. Nogueras
Lead Analyst, CitiFinancial International – Risk Management Technology
290 East John Carpenter Freeway
Irving, TX 75062
972-652-1046
pablo.j.nogueras@citigroup.com

Acknowledgements

Multiprocessing with Version 8 of the SAS System, Cheryl Doninger, SAS Institute Inc.
David Cedillo, CitiFinancial International, Decision Science

Further Reading / Research

SAS Community: Scalability and Performance <http://support.sas.com/rnd/scalability/index.html>

Notices

SAS and SAS/CONNECT are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries.
SUN and SOLARIS are registered trademarks or trademarks of SUN Corporation in the USA and other countries.
Other brands and names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries.

Appendix

Data Values Program

```
dm 'clear log' ;
dm 'clear output' ;
/*****
/**      data_value.sas
/* Project : XXXXXXXXXXXXXXXXXXXXXXXX Date: XXXXXXXXX
/* Requestor: XXXXXXXXX
/* Analyst : XXXXXXXXX
/*
/*
*****/

/*****/
/***** BEGINNING OF PARAMETERS *****/

libname indata '/cdsg/cis/mf/uk/aims/data/' ;

options yearcutoff=1950 obs=max symbolgen mlogic mprint source2 ;

%let dsname = mf200406 ;
%let month = %substr(&dsname,3,6) ;
%let dspre = ZZ ;
%let dset = indata.&dsname ;
%let outpath=/cdsg/users/noguerap/pgms/mp_connect_test/output/;

run ;

data _null_ ;
    length month $7.;
    month =put(input("&month",yymn6.),monyy7.);
    call symput('month',month);
run ;

/***** END OF PARAMETERS - DON'T MODIFY AFTER HERE!!! *****/
/*****/

filename pgm1 temp ;

run;
proc contents data= &dset out=conttemp noprint ;

/*****/
/* BUILD DYNAMIC ANALYSIS CODE
/*
*****/

filename temp1 temp;
run;

data _null_ ;
    file pgm1 ;
    set conttemp ;

    if label='' then label='Not Available';
    string='Variable: ' || compress(name) || ' Description: ' || (label);

    fname=compress(trim("&outpath")||trim("&dspre")||'_'||trim(name)||'.txt' );

    if fileexist(fname) then file_xst = 1 ;
    else file_xst = 0 ;

    put 'proc printto; ';
    put 'run; ';

    put 'filename out1 "' fname '"';
    put 'run;';

    put 'proc printto file=temp1 new; ';
    put 'run; ';

    /*** Dataset gt 0 Section and omit normal processing if null ***/
    if nobs > 0 then
```

```

do ;

/**** Proc Univariate Section ****/
if type=1 then do;
  put 'title "Data represents values for &month data ----- run on: &sysdate " ; ' ;
  put ' ' ;
  put 'proc univariate data= &dset plot; ' ;
  put ' var ' name ' ; ' ;
  put 'run; ' ;

  put 'proc printto; ' ;
  put 'run; ' ;

end;
/**** End Proc Univariate Sect ****/

/**** Data Frequency Section, output top 50 or less frequencies ****/
else if type=2 then do;
  put 'data ' name ' (index=( ' name ')); ' ;
  put ' set &dset (keep= ' name '); ' ;
  put 'run; ' ;

  put 'data ' name ' ; ' ;
  put ' set ' name ' nobs=nobs end=eof; ' ;
  put ' by ' name ' ; ' ;
  put ' rec = nobs ; ' ;
  put ' attrib count format=comma12. label="# of Records with Value" ' ;
  put ' per label="% of Total Population"; ' ;

  put ' retain count countd 0 ; ' ;

  put ' if first.' name ' then do; ' ;
  put ' count =1 ; ' ;
  put ' countd=sum(countd,1); ' ;
  put ' end; ' ;
  put ' else count = count + 1 ; ' ;

  put ' per = (count/rec)*100; ' ;

  put ' if last.' name ' then ' ;
  put ' do ; ' ;
  put ' if eof then ' ;
  put ' do ; ' ;
  put ' if countd > 50 then flag = "More than 50 Discrete Values"; ' ;
  put ' else if rec = 0 or (rec=1 and " name " = " ") then flag = "All Values Missing"; ' ;
  put ' else if countd < 50 then flag = "Less than 50 Discrete Values"; ' ;

  put ' rec=compress(trim(rec)); ' ;

  put ' call symput("flag",flag); ' ;
  put ' call symput("rec",put(rec,comma12.)); ' ;
  put ' call symput("dis",put(countd,comma12.)); ' ;
  put ' end; ' ;
  put ' output ' name ' ; ' ;
  put ' end; ' ;
  put 'run ; ' ;

  put 'title1 "Report represents values for &month data," ; ' ;
  put 'title2 ' string ' ; ' ;
  put 'title3 "&flag --&dis Discrete Values --&rec Total Population"; ' ;

  put 'proc sort data= ' name ' ; ' ;
  put ' by descending count; ' ;
  put 'run ; ' ;

  put 'proc print data= ' name ' (obs=50) n noobs label; ' ;
  put ' var ' name ' count per; ' ;
  put 'run ; ' ;

  put 'proc datasets library=work nolist ; ' ;
  put ' delete ' name ' ; ' ;
  put 'run ; ' ;

end ;
/**** End Data Freq Sect ****/

```

```

end ;
/**** End Datasets with 1 or more recods ****/

/**** Null Dataset Section ****/
else if nobs = 0 then
do ;
put 'proc printto ; ' ;
put 'run ; ' ;

put 'data _null_ ; ' ;
put ' file temp1 mod ; ' ;
put ' put "No Observations for Qtr Ending &month as of &sysdate"; ' ;
put 'run ; ' ;
end ;
/**** End Null Dataset Sect ****/

put 'proc printto; ';
put 'run; ';

/**** Output Section - Appends new data to top of file ****/
put 'data _null_ ; ' ;
put ' file temp1 mod ; ' ;
put ' put " " ; ' ;
put ' put "-----" ; ' ;
put 'run ; ' ;

/**** Check if file previously existed ****/
if file_xst = 1 then
do ;
put 'data _null_ ; ' ;
put ' file temp1 mod ; ' ;
put ' infile out1 ; ' ;
put ' input ; ' ;
put ' put _infile_ ; ' ;
put 'run ; ' ;
end ;

put 'data _null_ ; ' ;
put ' file out1 ; ' ;
put ' infile temp1 ; ' ;
put ' input ; ' ;
put ' put _infile_ ; ' ;
put 'run ; ' ;
/**** End Output Section ****/

put 'filename out1 clear ; ' ;
put 'run; ';

run;

%include pgm1 ;
run;

```

Appendix

Data Values Program – MP CONNECT

```
dm 'clear log' ;
dm 'clear output' ;

/* SAS PROGRAM DOCUMENTATION ----- */
/* */
/* PROGRAM NAME: data_values_mp.sas */
/* */
/* PROGRAMMER : Pablo J. Nogueras */
/* */
/* PURPOSE : Create Data Value Dictionaries for SAS Dataset */
/* */
/* REQUESTOR : XXXXXXXXXXXXXXXX */
/* */
/* INPUT : SAS datasets */
/* */
/* OUTPUT : Text files containing Proc Univariate (Numeric) or */
/*          Daststep Frequency (Character) data. The frequency data is */
/*          limited to top 50 discrete values. */
/* */
/* CALLED BY : n/a */
/* */
/* CALLS : n/a */
/* */
/* SCHEDULED : n/a */
/* */
/* VARIABLES : n/a */
/* */
/* ----- */
/* Revision History */
/* */
/* Programmer Revision Date */
/* ===== */
/* P NOGUERAS Modification of Original Program and David Cedillo 07/02/04 */
/* Program */
/* ----- */

/***** BEGINNING OF PARAMETERS *****/

/* The autosignon and sacmd options are necessary for MP Connect processing.
Autosignon=Yes allows you to create a new "remote" SAS session on the
current computer without having to specify login information.
Sascmd= specifies the location of the SAS executable. Depending on the
OS (in this case Solaris, you may have to specify the exact path. */

options obs=max pagesize=120 mlogic mprint symbolgen macrogen source2
autosignon=yes sascmd="/opt/sasv8/sas";

libname indata "/cdsg/cis/mf/uk/aims/data" ;

%let dsname = mf200406 ;
%let month = %substr(&dsname,3,6) ;
%let dspre = XA ;
%let dset = indata.&dsname ;
%let outputpath = /cdsg/users/noguera/pgms/mp_connect_test/output/;

/* Processess (usually equal to processors) dedicated to task */

%let maxsesn = 5 ;

run ;

data _null_ ;
length month $7;
if "&month" = " " then do;
month = intnx("month",today(),-1);
end;
else do;
```

```

        month =put(input("&month",yymmn6.),monyy7.);
end;

call symput("month",month);
run ;

/***** END OF PARAMETERS - DON'T MODIFY AFTER HERE!!! *****/
/*****/

/* Create temporay file to hold dynamic code */

filename pgm1 temp ;

/* Build dataset from Proc Contents to feed Dynamic Code creation */

proc contents data= &dset out=conttemp noprint;
run;

/*****/
/* BUILD DYNAMIC CODE */
/*****/

/* Each Variable will create an RSUBMIT block for each MP CONNECT Process */

data _null_;

    file pgm1 ; /* Output to TEMP file */
    set conttemp end=eof;

    if label='' then label='Not Available'; /* Trap Missing Labels */

    /* Create title string for output */
    string='Variable: ' || compress(name) || '      Description: ' || (label);

    /* Create filename to store output from Procedure */
    fname=compress(trim("&outpath")||trim("&dspre")||'_'||trim(name)||'.txt');

    /* If the file exists previously then we want to set a flag to add to the original file */
    if fileexist(fname) then file_xst = 1 ;
    else                      file_xst = 0 ;

    month = "&month" ;

    x + 1 ; /* MP Connect counter and job name */

    put 'rsubmit process = job' x ' wait=no ; ' ;

    put 'libname indata "/cdsg/cis/mf/uk/aims/data" ; ' ;

    put 'proc printto; ';
    put 'run; ';

    put 'filename temp1 temp ;';
    put 'run;';

    put 'filename out1 "' fname '" ;';
    put 'run;';

    put 'proc printto file=temp1 new; ';
    put 'run; ';

    /*** Dataset gt 0 Section and omit normal processing if null ***/
    if nobs > 0 then
        do ;

            /*** Proc Univariate Section ***/
            if type=1 then do;
                put 'title "Data represents values for ' month 'data ----- run on: &sysdate"; ';

                put 'proc univariate data= indata.' memname ' plot; ';
                put ' var ' name ' ;';
                put 'run; ';

                put 'proc printto; ';
                put 'run; ';

            end;

        end;

```

```

/**** End Proc Univariate Sect ****/

/**** Data Frequency Section, output top 50 or less frequencies ****/
else if type=2 then do;
  put 'data ' name ' (index=( ' name ')); ';
  put ' set indata.' memname '(keep= ' name '); ';
  put 'run; ';

  put 'data ' name ' ; ';
  put ' set ' name ' nobs=nobs end=eof; ' ;
  put ' by ' name ' ; ' ;
  put ' rec = nobs ; ' ;
  put ' attrib count format=comma12. label="# of Records with Value" ' ;
  put ' per label="% of Total Population"; ' ;

  put ' retain count countd 0 ; ' ;

  put ' if first.' name ' then do; ' ;
  put ' count =1 ;';
  put ' countd=sum(countd,1); ' ;
  put ' end; ' ;
  put ' else count = count + 1 ; ' ;

  put ' per = (count/rec)*100; ' ;

  put ' if last.' name ' then ' ;
  put ' do ; ' ;
  put ' if eof then ' ;
  put ' do ; ' ;
  put ' if countd > 50 then flag = "More than 50 Discrete Values"; ' ;
  put ' else if rec = 0 or (rec=1 and " name " = " ") then flag = "All Values Missing"; ' ;
  put ' else if countd < 50 then flag = "Less than 50 Discrete Values"; ' ;

  put ' rec=compress(trim(rec)); ' ;

  put ' call symput("flag",flag); ' ;
  put ' call symput("rec",put(rec,comma12.)); ' ;
  put ' call symput("dis",put(countd,comma12.)); ' ;
  put ' end; ' ;
  put ' output ' name ' ; ' ;
  put ' end; ' ;
  put 'run ; ' ;

  put 'title1 "Report represents values for ' month 'data,"; ' ;
  put 'title2 ' string ' ; ' ;
  put 'title3 "&flag --&dis Discrete Values --&rec Total Population"; ' ;

  put 'proc sort data= ' name ' ; ' ;
  put ' by descending count; ' ;
  put 'run ; ' ;

  put 'proc print data= ' name ' (obs=50) n nobs label; ' ;
  put ' var ' name ' count per; ' ;
  put 'run ; ' ;

  put 'proc datasets library=work nolist ; ' ;
  put ' delete ' name ' ; ' ;
  put 'run ; ' ;

end ;
/**** End Data Freq Sect ****/

end ;
/**** End Datasets with 1 or more recods ****/

/**** Null Dataset Section ****/
else if nobs = 0 then
do ;
  put 'proc printto ; ' ;
  put 'run ; ' ;

  put 'data _null_ ; ' ;
  put ' file temp1 mod ; ' ;
  put ' put "No Observations for Qtr Ending &month as of &sysdate"; ' ;
  put 'run ; ' ;

```

```

    end ;
    /*** End Null Dataset Sect   ***/

    put 'proc printto; ';
    put 'run; ';

    /*** Output Section - Appends new data to top of file   ***/
    put 'data _null_ ; ' ;
    put ' file temp1 mod ; ' ;
    put ' put " " ; ' ;
    put ' put "-----" ; ' ;
    -----" ; ' ;
    put 'run ; ' ;

    /*** Check if file previously existed   ***/
    if file_xst = 1 then do;
        put 'data _null_ ; ' ;
        put ' file temp1 mod ; ' ;
        put ' infile out1 ; ' ;
        put ' input ; ' ;
        put ' put _infile_ ; ' ;
        put 'run ; ' ;
    end ;

    put 'data _null_ ; ' ;
    put ' file out1 ; ' ;
    put ' infile temp1 ; ' ;
    put ' input ; ' ;
    put ' put _infile_ ; ' ;
    put 'run ; ' ;
    /*** End Output Section   ***/

    put 'filename out1 clear ; ' ;
    put 'run; ';

    put 'endrsubmit ; ' ;

    /*** MP Connect Control Section ***/

    /* If MP Connect counter is greater than Max Processes then begin regulating
       number of concurrent processes.
       Example:
       If Max Processes = 4 then Job 5 will wait for Job 1, Job 6 will wait for Job 2,
       and Job y will wait for Job y - 4 (Max Processes) */

    if x > &maxsesn then do;
        y = x - &maxsesn ;
        put 'waitfor _any_ job' y ' ; ' ;
        put 'signoff job' y ' ; ' ;
    end ;

    /* If end of file (last variable processed) then create signoff statements for remaining
       processes.
       Number of remaining processes = total variables - (total variables - max processes) */

    if eof then
        do ;
            remjob = y + 1 ;
            do i = remjob to x ;
                put 'signoff job' i ' ; ' ;
            end ;
        end ;
    put ' ' ;
    /*** End MP Connect Control Section ***/

run;

%include pgm1; /* Include dynamic code for execution */
run;

```

Appendix

End of Log Outputs

- i. Baseline – One Process, No MP CONNECT
- ii. NOTE: SAS Institute Inc., SAS Campus Drive, Cary, NC USA 27513-2414
- iii. NOTE: The SAS System used:
- iv. real time 8:55:40.31
- v. cpu time 8:55:06.96

- vi. 5 Processes, MP CONNECT
- vii. NOTE: SAS Institute Inc., SAS Campus Drive, Cary, NC USA 27513-2414
- viii. NOTE: The SAS System used:
- ix. real time 1:38:06.88
- x. cpu time 1:13.75

- xi. 10 Processes, MP CONNECT
- xii. NOTE: SAS Institute Inc., SAS Campus Drive, Cary, NC USA 27513-2414
- xiii. NOTE: The SAS System used:
- xiv. real time 54:32.13
- xv. cpu time 1:13.82

- xvi. 15 Processes, MP CONNECT
- xvii. NOTE: SAS Institute Inc., SAS Campus Drive, Cary, NC USA 27513-2414
- xviii. NOTE: The SAS System used:
- xix. real time 38:55.23
- xx. cpu time 1:17.90

- xxi. 20 Processes, MP CONNECT
- xxii. NOTE: SAS Institute Inc., SAS Campus Drive, Cary, NC USA 27513-2414
- xxiii. NOTE: The SAS System used:
- xxiv. real time 30:18.78
- xxv. cpu time 1:20.13

- xxvi. 25 Processes, MP CONNECT
- xxvii. NOTE: SAS Institute Inc., SAS Campus Drive, Cary, NC USA 27513-2414
- xxviii. NOTE: The SAS System used:
- xxix. real time 25:20.33
- xxx. cpu time 1:28.12

- xxxi. 30 Processes, MP CONNECT
- xxxii. NOTE: SAS Institute Inc., SAS Campus Drive, Cary, NC USA 27513-2414
- xxxiii. NOTE: The SAS System used:
- xxxiv. real time 21:34.87
- xxxv. cpu time 1:31.36

- xxxvi. 35 Processes, MP CONNECT
- xxxvii. NOTE: SAS Institute Inc., SAS Campus Drive, Cary, NC USA 27513-2414
- xxxviii. NOTE: The SAS System used:
- xxxix. real time 19:29.24
- xl. cpu time 1:34.45

- xli. 40 Processes, MP CONNECT
- xlii. NOTE: SAS Institute Inc., SAS Campus Drive, Cary, NC USA 27513-2414
- xliii. NOTE: The SAS System used:
- xliv. real time 18:09.77
- xlvi. cpu time 1:32.36