FORMING SAS TIME-SERIES DATA SETS FROM THE INTERNATIONAL FINANCIAL
STATISTICS DATA TAPE OF THE INTERNATIONAL MONETARY FUND

ALEX ANCKONI III, THE GEORGE WASHINGTON UNIVERSITY

I. INTRODUCTION

The International Monetary Fund (IMF) supplies a number of important financial, economic, and fiscal data series in tape format. The data cover monthly, quarterly and annual information where available for all member countries. Coverage includes as much as four decades of data (depending upon country, frequency and specific data type). As provided, the data are not convenient for time series analysis using SAS procedures. If desired, the data can be placed the SAS data set on a direct access storage device for subsequent analysis. While the use of the IMF's other data tapes are not covered in this paper, the tape formats are sufficiently similar that the described program should provide an important "starting point" to analysts requiring the formation of SAS data sets from these data sources.

This paper provides instruction on the use of an included SAS program which will read specified data from the International Financial Statistics (IFS) data tape (which contains approximately 21,000 time series) and forms a SAS time series data set which can then be used directly with the various SAS analysis procedures. If desired, the program can access storage device for subsequent analysis. While the use of the IMF's other data tapes are not covered in this paper, the tape formats are sufficiently similar that the described program should provide an important "starting point" to analysts requiring the formation of SAS data sets from these data sources.

The next section of this paper describes the three-step operation of the SAS program. The last section provides suggestions for further development of the program. The program is provided in complete form in the appendix to this paper.

II. PROGRAM DESCRIPTION

General Description

As presently configured, the program is able to select any given number of data series from any given number of countries by placing the desired country codes and data subject codes into the appropriate conditional ("IF") statements. The program restores the appropriate number of zeros so that all data are given in complete form (e.g. not in billions of dollars, etc). The first output is the table of descriptor information for each data series transferred from the tape to the desired SAS data set. The output of Step One is used to set the observation parameters for the sub-setting portion of Step Two of the program. This review of the Step One output in preparing to execute Step Two also allows the analyst to confirm the data being taken from the tape is indeed present on the tape for the frequency and data type of interest to the analyst.

Step Three describes and provides the programming necessary to place the series SAS data sets formed in Step Two on a direct access storage device.

The specific example to be used for the demonstration code is to satisfy a requirement for annual, quarterly and monthly time series for the foreign exchange rates in both US dollars per unit of local currency and local currency per US dollar (lines ag and ae of the IFS), for the US dollar level of foreign exchange reserves minus gold (line 11.d of the IFS), for the consumer price index (line 34 of the IFS), and for the local currency value of exports (line 70 of the IFS) of four countries. The descriptors for which the data are desired are the United States (IFS country code=111), Kuwait (IFS country code=443), Qatar (IFS country code=453), and Saudi Arabia (IFS country code=456). It is further desired that the time series be loaded onto a direct access storage device as a SAS formatted data set with the annual, quarterly and monthly data in separate data set partitions. The code necessary to read the data from the tapes and form temporary SAS data sets is described in detail in the appendix at the end of the paper. The additional JCL lines and SAS code necessary to form a permanent SAS data set based upon the temporary data sets formed in Step Two are shown in Step Three.

Step One

The first section of code accomplishes the actual reading of the IFS tape and forms a series of four data sets, one containing the description material for each of the data series extracted from the tape, the next containing the annual data, the third containing the quarterly data, and the fourth containing the monthly data. As will be developed below, the data structure at the of Step One is not satisfactory for time-series analysis using SAS's analytical procedures.

The first five lines of Step One code contains the Job Control Language (JCL) code necessary for IBM OS execution at the George Washington University. Note that the IFS tape used is a 6250 BPI density tape. While the IMF supplies the data tapes in both 1600 BPI and 6250 BPI, the 6250 BPI tapes contain data covering a longer period of time (see the IMF tape description material for more detail). You will be able to obtain the necessary JCL for your computer facility from your information center technical assistance personnel. Also please note that the code presented utilizes a few features which are not available in SAS versions prior to the Version 5 release (these...
are in the area of character string manipulation).

The next line is an optional OPTION line that (when the asterisk is removed) will keep the output readable on a standard 80 column terminal when using IBM's Xedit Display mode (you may want a different number for your terminal and/or editor system). This is followed by the opening of the four data sets to be formed (DESCRIPT, ANNUAL, QTRLY, and MONTHLY). The KEEP statement excludes unnecessary variables from appearing in the final data sets. The RETAIN statement is used to carry variable values from the data descriptor lines read from the tape to the actual data lines where the information is used to condition the output.

The LENGTH line provides character width information for the variables read to test the type of information line being read. The ARRAY line sets up the array TS which will be used to store the information as it is read from the tape. The INFILE line actually begins the tape reading process. The first INPUT line reads the first character and the second character of each line of information on the tape and assigns this information to the indicated variable names (note the trailing @ at the end of this INPUT line which is used to hold the observation reading process on the current observation on the tape). If the first character read is a P, the observation from the tape is deleted since this indicates a line of Cobol code which is at the front of each tape. If the observation is not deleted, the SOURCE variable is formed as indicated (this is almost always an I for the IFS data tapes, it will be different for the other IMF tapes).

If the data line on the tape is not a Cobol code line, then it is either a descriptor record or a data record. The next line uses the conditional IF structure to examine whether it is a descriptor record (e.g. the FREQ variable is not A or Q or M -- the FREQ variable having a value of A, Q, or M would indicate the line is a data line). If the data is a descriptor record, the continuation INPUT statement reads the IMF country code (IMFCOUNTRY) and the data series subject code (SUBJCODE) and holds the input pointer in the same observation line while it uses the compound conditional IF statement to check whether or not the data series is one of the desired data series (obviously, if the analyst needs other data from other countries, different and/or more entries can be used in the conditional IF statement). The first "sticky point" encountered in the use of the IFS tapes is to be noted here. Note that the data subject codes on the tape are not identically equal to the data series subject codes in the IMF's monthly published version of the IFS. The structure of the subject code on the tapes is explained on pages 8 and 9 of the IMF's documentation for the computer tape subscription previously referenced. After a bit of practice, the conversion from the more familiar identification code in the written publication to that of the tape subject code is easily accomplished (This could be automated as a further development of the program presented in this paper, the inverse of the necessary code is in the program discussed below and provided in the appendix).

Next, the desired variables from the descriptor record are read. Then, any non-blank PARTNER variable is translated to XXX for use as an indicator suffix string in the subsequent formation of the CODE variable. The more familiar CODE variable (as shown in the IMF's monthly IFS publication) is then formed from the tape's CODEX variable using SAS's string manipulation functions. The data read from the record with the attendant formed variables are then output to the DESCRIPT data set.

After the descriptor record is output to its data set, the annual, quarterly and monthly data sets are formed from the data lines on the tape using the next group of code. Starting with the INPUT statement that reads the third character on the tape record, the DO WHILE statement tests for the line being a data line (e.g. the third character is an A indicating an annual data record, a Q indicating a quarterly record, or an M indicating a monthly record) and a record from the country and data series desired. If the conditions of the DO WHILE statement are met, a record is read using the next INPUT statement. The second "sticky point" of the tape reading program is the reading of the data on the tape with the SAS format ZD6, which corresponds to the method of storing the sign of the data in the most byte along with the last digit which is used by the COBOL program which forms the IMF data tapes. The ?? term used in the data INPUT specification suppresses the printing of invalid character error messages and prevents the incrementing of the SAS_ERROR counter while setting invalid data to the missing value indicator.

The data are then read into the TS data array and the values are adjusted to the full natural number format by placing the appropriate number of zeros to the right of read numbers using the information read from the tape for the DIMUNIT variable. As the final element of Step One of the program, the data are segregated into three data sets depending upon whether the data is of an ANNUAL, QTRLY, or MONTHLY time series. As a part of this segregation, the appropriate time parameters and/or variable names are formed and added to each segregated data set. At the end of Step One, the ANNUAL, QTRLY and MONTHLY data sets are printed. These printed output are used to confirm the availability and time span of the desired data series and to set the FIRSTOBS and OBS parameters in Step Two of the program which rearranges the data sets to the more familiar time series form.

Step Two

Step Two of the provided program has three natural segments. The first segment forms the annual time series, the second segment forms the quarterly time series, and the third segment forms the monthly time series.

In the formation of the annual time series, a number of annual data sub-sets are formed—with each containing the annual data for one data series and the YEAR value. The observation limits for the FIRSTOBS and OBS parameters for each individual data series are taken from the ANNUAL data set.
printout obtained in Step One of the provided program. When each data sub-set is formed, the variable VALUE is RENAMED to the variable name desired for the data series in the final time series data set. After all of the desired annual data sub-sets are formed, they are MERGED BY YEAR to form the desired set of annual time series. The PDMC PRINT of DATA ANN merely shows the content and structure of the final annual time series data set.

In the formation of the quarterly time series, the QTRLY data set from Step One must first be TRANSPOSED for the variables QTR1, QTR2, QTR3, and QTR4, and BY the variables INCOUNTRY, CODE, and YEAR in this given order to form an internal data set GQP temp which allows easy formation of a series of quarterly data sub-sets which can then be merged to form the desired quarterly data set. Before specifying the quarterly sub-sets, the code beginning with the DATA QT2 statement is used to form a date corresponding to the first day of the quarter in the standard SAS YYQ4 format (obviously the date could be easily modified to represent the midpoint or end of the time period with a suitable code adjustment). In the data sub-set formation, the data series renaming and the keeping of the date and value variables in the data sub-sets is similar to the procedure used in the annual data sub-set formation. The numbers of the FIRSTOBS and OBS parameters are set from the output of Step One and is facilitated by remembering that there are four quarterly data values on each observation line of the Step One output. Hence, if the particular country variable data is given on observations 50 through 65 of the Step One output, the FIRSTOBS parameter for that variable's sub-set will be 200 and the OBS parameter will be 263. The quarterly data sub-sets are then MERGED BY DATE and the resultant dataset QUARTER is printed for information.

An almost identical process is followed to place the monthly data in the desired time series format. Due to the similarity, the code will not be further discussed here except to note that the twelve data values per observation line in the monthly data output of Step One must be considered when setting the FIRSTOBS and OBS parameters to form the monthly data sub-sets.

At this point the three data sets ANNUAL, QUARTER, and MONTHLY exist as normal SAS time-series structured data sets and are available for any analysis desired to be added to the program. The next section describes the additional programming that would be used to place these data sets on a direct access storage device (DASD) as they are formed on the IBM OS system at the George Washington University.

Step Three (DASD Storage of the Data Sets)

To form permanent SAS data sets by placing the ANNUAL, QUARTER, and MONTHLY data sets formed above on a DASD, the first new line of code would be a JCL line placed after the last JCL line shown in the provided program. This line might look like the following (very dependent upon local equipment, number of variables in your data set, your account number, and local system rules. Again, it would be prudent to discuss this with the technical assistance personnel at your information center):

```
//TEST DD DSN=BA212345.TEST,DISP=(,CATLG),
// VOL=REF=SPNG87,SPACE=(TRK,(10,5))
```

Then, the following code is placed at the end of the Step Two code to place the data sets on the DASD (in the manner and location indicated by the added JCL line) in partitioned permanent data set form:

```
DATA TEST.ANNUAL; SET ANNUAL; OUTPUT TEST.ANNUAL;
DATA TEST.QUARTER; SET QUARTER; OUTPUT TEST.QUARTER;
DATA TEST.MONTHLY; SET MONTHLY; OUTPUT TEST.MONTHLY;
```

III. RECOMMENDATIONS FOR FURTHER PROGRAM DEVELOPMENT

With some additional thought and testing, the program could be made to use the more familiar data series codes used in the IMF's written literature rather than the somewhat unfamiliar subject code used on the tape (since the translation is one-to-one mapable and of relatively straightforward form). Such a program modification would be a substantial improvement in the "user friendliness" of the tape reading program.

With a bit more thought and testing, the program could be made to operate reliably in one step. That is, tests for data availability and data span could be incorporated into the data restructuring process so that analyst intervention was not required between steps one and two. While this might result in the need to redo some analyses when a data series was used for the first time (e.g. the series may not be present, or may start at a later date than desired), in the cases of repeat application of "well known" data series, this form of problem would be non-existent.

Since the International Monetary Fund statistical series are of major importance to financial economists around the world, the SAS Institute and the International Monetary Fund should consider the development of a formal SAS PROC that would facilitate the formation of SAS data sets from any of the four IMF data tapes. Perhaps the program described in this paper will encourage such an important development.

It is requested that the author be informed of any further development of the provided program.

ENDNOTES

1. A brief description of the data tapes and subscription details are to be found in the "International Monetary Fund Publications Catalog" which is available from the International Monetary Fund, Publications Unit, 700 19th Street, N.W., Washington, D.C., 20431, USA.

2. SAS is a registered trademark of SAS Institute Inc., Cary, NC, USA.

3. The tape data structure is described in detail in the publication "Documentation - Computer
4. Portions of Step One of this program were inspired by and have their roots in the much more elegant and menu driven system (which stores the data from all four of the IMF data tapes on direct access memory units) which has been developed at the US Government's Agency for International Development by Mr. Ross Merlin of PinRotton Computer Consultants, Alexandria, Virginia, USA.

5. The assistance of Dr. Adnan Soufi of King Abdulaziz University in Jeddah, Saudi Arabia during the initial testing and operational evaluation of this program is gratefully acknowledged.

6. This and the following discussion of the code is meant to be only an indicative guide to the code, not an exhaustive treatment which is prohibited by the space constraints of this paper. It is assumed the reader is a knowledgeable SAS user and that the reader has access to the SAS Users Guide: Basics and the previously cited IMF tape data structure documentation while studying the code being discussed. IBM is a registered trademark of the International Business Machines Corporation.

7. If more than 50 data sub-sets are to be merged, the analyst must first merge groups of 50 and then merge the aggregated datasets to stay within SAS's limit of 50 when merging data sets.

APPENDIX

IFS TAPEREADING PROGRAM

J=TIME2
JRC="PLEASE MOUNT TAPE VOL=SER=095471 READ ONLY"
LIM=2,10,10
// EXEC SAS, V=508
// IMFTAPES DD DSN='IFSDATA',UNIT=TAPE250,
// VOL=SER=095471,DISP=SHR
*********STEP ONE BEGINS HERE**********;
*OPTIONS LS=73;
DATA DESCRIPT(KEEP-SOURCE CTRLSRCE CODE VERSION DATATYPE DIMUNIT DEC DESCRIPT COUNTRY MAGNITUDE BASEYEAR DATASRCE)
ANNUAL(KEEP-SOURCE IMFENTRY CODE YEAR VALUE) QTRLY(KEEP-SOURCE IMFENTRY CODE YEAR QTR= QTR4)
MONTHLY(KEEP-SOURCE IMFENTRY CODE YEAR MONTH-MONTH12);
RETAIN IMFENTRY CTRLSRCE CODE VERSION DATATYPE DIMUNIT SUBJCODE DEC DESCRIPT COUNTRY MAGNITUDE BASEYEAR DATASRCE PARTNER;

AUTHOR'S ADDRESS

Alex Anckonie III
Business Administration Department
The George Washington University
Washington, D.C. 20052 USA
IF TS'=. THEN DO;
IF INDEXC(DIMUNIT,'6ACEFV*/1')=0 THEN EXP=O-DEC;
ELSE IF INDEXC(DIMUNIT,'BD')=0 THEN EXP=2-DEC;
ELSE IF INDEXC(DIMUNIT,'l29MS')=0 THEN EXP=3-DEC;
ELSE IF INDEXC(DIMUNIT,'34T')=0 THEN EXP=6-DEC;
ELSE IF INDEXC(DIMUNIT,'QTR')=0 THEN EXP=9-DEC;
IF EXP=. THEN EXP=O-DEC;
TS=TS*lO**EXP;
END;
END;
IF FREQ='A' THEN DO OVER TS;
*ANNUAL TIME SERIES;
YEAR=19DO + START YR + l;
VALUE=TS;...
IF VALUE..... THEN OUTPUT ANNDAL;
END;
ELSE IF FREQ='Q' THEN DO;
*QUARTERLY TIME SERIES;
IF SUM (OF MONTH1-MONTH4) NE • THEN DO;
YEAR=19DO + START YR;
QTR1=MONTH1; QTR2=MONTH2; QTR3=MONTH3;
QTR4-MONTH4;
OUTPUT QTRLY;
END;
IF SUM (OF MONTHS-MONTHS) NE • THEN DO;
YEAR=START YR +1900 +1;
QTR1-MONTHS; QTR2=MONTH6; QTR3=MONTH7;
QTR4=MONTH8;
OUTPUT QTRH;
END;
IF SUM (OF MONTH9-MONTH12) NE • THEN DO;
YEAR=1900+START YR+2;
QTR1-MONTH'; QTR2-MONTH10; QTR3-MONTH11;
QTR4-MONTH12;
OUTPUT QTRllY;
END;
ELSE IF FREQ='M' THEN DO;
*MONTHLY TIME SERIES;
YEAR=1900+START YR;
IF SUM(OF MONTHI-MONTH12)..... THEN OUTPUT MONTHLY;
END;
INPUT @3 FREQ $1. @;
END;
PROC PRINT DATA=DEseRIPT;
PROC PRINT DATA=ANNVAL;
PROC PRINT DATA=QTRLY;
PROC PRINT DATA=MONTHLY;
***STEP TWO BEGINS HERE*******;
*BEGIN WITH ANNUAL PROCESSING**;
DATA ANNUAL1;
SET ANNDAL(FIRSTOBS=1 OBS=38);
RENAME VALUE=USAE; KEEP YEAR VALUE;
DATA ANNUAL2;
SET ANNDAL(FIRSTOBS=39 OBS=76);
RENAME VALUE=USAG; KEEP YEAR VALUE;
DATA ANNUAL3;
SET ANNDAL(FIRSTOBS=77 OBS=114);
RENAME VALUE=USRMG; KEEP YEAR VALUE;
sixteen data set operations omitted due to their similarity to the shown steps
DATA ANNUAL20;
SET ANNDAL(FIRSTOBS=587 OBS=616);
RENAME VALUE=SAEX; KEEP YEAR VALUE;
DATA ANN;
MERGE ANNUAL1 ANNUAL2 ANNUAL3 ANNUAL4 etc through all of the sub-sets to
DATA MONTH1;
  SET MON1(FIRSTOBS=721 OBS=1080);
  RENAME COL1=USRMG; KEEP DATE COL1;

sixteen data set operations omitted due to their similarity to the shown steps

DATA MONTH2;
  SET MON2(FIRSTOBS=5713 OBS=6048);
  RENAME COL1=SAEX; KEEP DATE COL1;

DATA MONTH;
  MERGE MONTH1 MONTH2 MONTH3 MONTH4 etc through all the sub-sets to MONTH18 MONTH19 MONTH20;
  BY DATE;

PROC PRINT;
  ID DATE;

******END OF PROGRAM *****;

NOTE:
The first-time user of this program, the author will send a copy of the source file and output (LISTING) file from an actual run to interested parties sending a request to the author at the above given address.