

Using Troll Version 12 on the CCS Computer with Access to
the CSO databank

by

John Costigan

John D. FitzGerald

January 1986

IR£2

**Confidential: Not to be quoted
until the permission of the Author
and the Institute is obtained.**

1. INTRODUCTION

This manual describes a new version of TROLL* and a new TSO command list which have been developed in the ESRI to allow users access the data contained in the CSO databank. At present this interface only handles the time series files in the CSO databank. It is hoped to extend it to cover the cross section files in 1986. As it operates at present the user must specify all the series he or she wants from the CSO databank at the time of calling the command list or starting a TROLL session. Generalisation to allow users to specify CSO files at any stage in a TROLL session is not planned for the immediate future. This manual also provides a technical description of the new SAS/TROLL interface and some details of other minor changes to TROLL.

Section two of this manual describes how the new interface actually works. Section three describes how to use the new version of TROLL to access CSO data as well as to carry out all the normal TROLL tasks. Section four describes some simple SAS procedures for obtaining details of the names and contents of the files and series in the CSO databank. Section Five gives details of the SAS macro used to transfer the data together with details of how it can be revised to take account of new CSO files or series. (This section is only of interest to those maintaining the system). Section Six gives details of the TROLL macro. Section 7 describes a simple routine which allows users to transfer data from TROLL to external disk or tape files.

2. HOW THE CSO DATABANK INTERFACE WORKS

This section describes how the new interface works. From the point of view of the normal user knowledge of this information is not necessary before using the system. However, for advanced users, or those involved in maintaining or developing the system in the future, this section should provide an outline of how the different elements of the interface fit together.

*The new subroutine incorporated into TROLL to transfer the data is copyright the ESRI.

At the point of calling the TSO command list to initiate a TROLL session the user, in addition to all the normal options, has the option of specifying the names of a file or the names of individual time series which are to be transferred to TROLL. If the user does not use this option TROLL will be called in the usual way using any other normal TROLL options specified when the command list was called. If, however, the option of specifying the names of CSO time series is used, the TSO command list first calls the SAS programme. The names of the required series are passed automatically to that programme and, on execution, it transfers the required data to a temporary disk file. The command list then automatically calls the revised version of TROLL. Up to this point the user has only had to specify the names of the required series when calling the command list and the execution of SAS has not been apparent to the user so long as the description of the CSO time series to be transferred was correctly entered. The user next finds himself or herself in TROLL in the normal way. After entering the normal system option (SYSIN80 or BATCH) at the start of the execution of the TROLL programme the user should immediately call a TROLL macro which reads the CSO data from the temporary file into TROLL, storing each series in TROLL under the same name used in the CSO databank. The macro prompts the user for the (optional) name of an archive in which the CSO data are to be stored. It also asks the user whether, in the event of series already existing with the same names as those to be loaded, the preexisting series are to be overwritten. When the macro completes execution the user is then in a normal TROLL environment into which the CSO data have been loaded and execution continues in the normal manner. If more series are to be loaded the user must log out of TROLL and commence this procedure again. It is strongly recommended that the user load the CSO series using the relevant TROLL macro before carrying out any other TROLL task as that macro deletes all existing SEARCH arguments which may have been specified earlier in a TROLL session.

3. HOW TO USE TROLL VERSION 12 WITH THE CSO DATABANK INTERFACE

There are two ways of using TROLL: interactively under TSO, or in BATCH mode. The interface which allows access to the CSO databank is at present only available under interactive operation. The procedure for using TROLL interactively is described in Section A. The procedure for using it in batch mode is described in Section B. It is assumed that each user has his or her own library. In the examples shown the user is assumed to have the name TA64ZTR.

Department of Finance users have names beginning	R060
Department of Transport users have names beginning	PC6A
Department of the Environment users have names beginning	CA05
ESRI users have names beginning	TA64

A. New Troll command list (for interactive use)

To enter TROLL first log into TSO. e.g.

```
LOGON TA64ZTR/SPL0D
```

When the machine returns with three asterisks press enter.

When the machine returns with the message READY then type TRL12 followed on the same line, by any (or some) of the options specified below. Each successive option should be separated from the previous option by at least one blank space.

The options available are divided into two groups - options used in normal TROLL operation, a, and options used to access the CSO databank, b. If it is desired, options can be spread over a number of lines by ending each line with a "+".

- (a) (i) The ability to access files other than the user's own file and the system files to which the user has automatic access. (These latter are TROLLSYS, SYSLIB and USERLIB).

USR2(TA64.A) These options allow the user to specify the
 USR3(TA64.M) additional files that they require access to.
 . : The name in brackets should be the first two
 . : segments of the name by which the computer
 . : knows the file. These two segments correspond
 USR7(RO60.R) respectively to the first four and fifth characters
 of the LOGON ID.
 Up to a maximum of six additional users can be
 accessed in this way.

- (ii) If the TROLL user wants to work in a TROLL library other than that associated with the user's logon ID permission must be given under ACF (the CCS security package). Provided permission has been granted this option is invoked by coding.

USRNAM(TA64.A) where the user wants to work in the library of
 user TA64ATR

- (iii) If the TROLL user wants to vary the destination of offline output this can be done using the OUT option. The default is that offline output (using TROLL commands beginning with "O") generated in a TROLL session is routed to the W queue when the user terminates the TROLL session in a normal manner. (This option also applies to output destined for device 7 such as offline graphics output).

OUT(A) In this case the output is routed to the A queue.

- (iv) The user has the option of running TROLL with all the input to TROLL being taken from a member of a partitioned data set (PDS). (This amounts to the same thing as running TROLL in batch mode except that the job begins execution immediately and the terminal is locked till the job completes). The name of the partitioned data set and its member are supplied as shown below:

DSNAME(TA64.PDS.A(SPLOD)) where the PDS is TA64.PDS.A
 and its member, SPLOD, contains
 the input for a batch TROLL job

- (v) In the case where the user elects to exercise option (iv) and to take the input for TROLL from a stored member of a PDS it may be useful to route all TROLL output to a destination other than the screen.

This allows the user to store all output to look at it at leisure and, if so desired, route the output to the printer. This option is invoked by coding:

OUTNAME(W) where the "W" specifies the W queue as the destination for all TROLL online output.

- (vi) A further option which may be useful when the PLOT package is used is the ability to route all output destined for device seven to a permanent disk file. For example, if certain graphics devices are to be used, the output for them from the PLOT package will be directed by TROLL to device 7. A further programme will read this data as input and route the output to the chosen plotter. As a result, it will be important in such cases to be able to specify the temporary storage file directly. This is done by the following command;

DISK(TA64.PDS.A(GRAPHICS)) where the output is to be stored in the member called GRAPHICS of the partitioned data set TA64.PDS.A. The file referred to in this option must already exist and must have previously been catalogued. Care should be taken if using the working PDS if the output from TROLL is likely to be large.

(b) CSO databank interface options:

(i) FILE(LRAM)

where LRAM is the name of the time series file in the CSO databank. This option must be specified if any series are to be loaded from the CSO databank. The names used by the CSO are four characters long. The last letter defines the periodicity of the series:

M = monthly
Q = quarterly
H = half yearly
L = greater than annual

See the attached list for all the CSO databank time series files. (p. 9)

(ii) CODES(LRAM001-LRAM003 LRAM007 LRAM129-LRAM157)

This option allows the user to specify which series from the CSO file specified in FILE are to be transferred to TROLL. It can

consist of a single code, a list of codes separated by blank(s) a range of codes separated by a dash or any combination of these. If the value of the parameter contains a space it must be enclosed in apostrophes (').

Valid examples: CODES(LRAM001)
 CODES('LRAM001-LRAM005 LRAM127')
 CODES(LRAM001-LRAM125)

If this option is not specified all the contents of the CSO FILE (e.g., LRAM) will be transferred.

If a series is specified on the codeslist that is not included in the file then that series will be loaded containing all zeros.

(iii) START('1981 4')

This option allows the user to specify the start date for all the series to be transferred. If a start year is specified then a start period must also be specified, otherwise it defaults to the series start date. The year may be in 4 or 2 digit form, e.g., (1984) (84). If this option is taken then the values must be enclosed in apostrophes, e.g., ('1983 2'). If the start date is not specified the data are transferred for the full period for which they are available in the CSO Databank.

(iv) END('1985 12')

This option allows the user to specify the end date for all the series to be transferred. Coding instructions are the same as for START. If END is not coded values for the series requested are transferred up to the latest date available in the CSO databank.

Examples:

TRL12 FILE(LRAM) START('1985 1')

This transfers all the series in LRAM starting in January 1985.

```
TRL12 FILE(BCAQ) CODES('BCAQ001-BCAQ005 BCAQ010') +
END('1980 3')
```

Six series ending in the third quarter (inclusive) of 1980 are

```
transferred:   BCAQ001
                BCAQ002
                BCAQ003
                BCAQ004
                BCAQ005
                BCAQ010
```

```
TRL12 FILE(NAAA) CODES(NAAA0602-NAAA0605) +
START('1980 1') END('1980 1')
```

The four series NAAA0602 NAAA0604 and NAAA0605 for the year 1980 are loaded.

Note on using START and END dates:

It is up to the user to make sure that the START date specified is on or after the series start date. If it is not it defaults to the series start date. Likewise with the end date it must come on or before the series end date. For a list of the correct start dates see the attached list of time series files in the CSO data bank.

The start and end periods of monthly time series range from 1 to 12, for annual and longer than annual the period is 1, for quarterly it is 1 to 4 and for half yearly the periods are 1 and 2. Any other periods are invalid and if specified will cause the date to default to the series date. A warning message will appear to this effect.

The TROLL macro to load the series into TROLL is called CSODB. To execute it when TROLL has been entered and the system option specified (e.g., SYSIN80) TROLL will prompt you with

TROLL command:

Enter &CSODB.

Troll will then ask you whether you want to archive the CSO data - enter YES or NO. If you enter YES TROLL will ask you for the archive name. Enter the

archive name, e.g., CSO. Finally you will be asked whether you want to overwrite any files which may exist in your library with the same names as the CSO files to be loaded, enter YES or NO. TROLL will then load the CSO files and return control to you. The session will then continue in the normal fashion.

B Running TROLL in Batch Mode

The following example of the JCL required is given for the user with LOGON ID TA64ZTR. It assumes that the user only wants to access his or her own TROLL library, together with those system libraries normally accessed (TROLLSYS, SYSLIB and USERLIB).

```
//TA64ZTRA JOB (TA64,0000),'TEST',CLASS=S
// EXEC TROLL12
//USER DD DSN=TA64.Z.TROLL,DISP=OLD
//SYSIN DD *
```

The name of each TROLL user's own library must be coded on the USER DD card. This name is formed from the logon ID by taking the first four characters of the ID, followed by a full stop and the fifth character, followed by a full stop and then the name TROLL. Additional user's libraries can be accessed by adding the following DD card before the SYSIN DD card.

```
//FT14F001 DD DSN=R060.R.TROLL,DISP=SHR
//FT15F001 DD etc.
up to //FT19F001.
```

The default output for this procedure is queue W. This can be changed to the A queue by replacing the EXEC card with the following

```
// EXEC TROLL12,CL=A
```

Time Series Files:

- AIAA: 600 Annual series from the Census of Industrial Production. All series start in 1973. There are series for each of 6 variables and for 100 NACE categories.
- APBM: 83 Monthly agricultural price series. These are all absolute price series, not index series. Most series start in January 1976.
- BCAQ: 25 Quarterly building and construction series. The series relate to planning permissions, and to earnings and hours worked in the industry. Some series start in Q1, 1969, but others have later start dates.
- CNAL: 320 Census of Population series, covering all censuses from 1841. Population and other variables are given by sex, county, etc.
- CNBL: 500 Census of Population series, starting in 1926. Population and other variables are given by sex, marital status and age group.
- CPAQ: 97 Quarterly consumer price series. 72 are index series to the different published bases, and the remaining 25 are national average price series. Most series start in Q4, 1968, but others have later start dates.
- LRAM: 94 Monthly Live Register Series, including all series from the monthly Live Register statement. The main series start in January 1967.
- LRBM: 423 Monthly Live Register series. 372 of these are the industrial analysis series, given by sex for each of 124 industries, starting in January 1973; the remaining 51 are the short-time series, starting in January 1980.
- LRCQ: 312 Quarterly series from the age analysis of the Live Register. All series start in Q1, 1980.
- LRDH: 384 half-yearly series from the age-by-duration analysis of the Live Register. All series start in H1, 1980.
- MIAM: 106 Monthly series from the Monthly Industrial Inquiry. Series are given, both unadjusted and seasonally adjusted, for each NACE category. Most series start in July 1975.
- MIAQ: 106 derived quarterly series from the Monthly Industrial Inquiry. Most series start in 1976.
- MIAA: 56 derived annual series from the Monthly Industrial Inquiry. Most series start in 1976.
- NAAA: 378 Annual National Income and Expenditure series, starting in 1970.

- QIAQ: 45 Quarterly employment series from the Quarterly Industrial Inquiry. The series are classified by NACE. The earliest series start in Q1, 1973. (Note: this file is not ready yet. It will be added shortly).
- RSAM: 70 monthly retail sales index series. Series are given for each description of business, both unadjusted and seasonally adjusted. All series start in January 1968.
- TRAM: 33 Monthly motor registration series. The earliest series start in January 1961.
- TRBM: 65 Miscellaneous monthly tourism and transport series, relating to passenger movement, etc. All series start in January 1961.
- TSAM: 225 monthly trade series: imports, exports and import excess for each of the 75 sections and divisions. Most series start in January 1972.
- VSAQ: 56 Quarterly vital statistics series, including the main series from the quarterly CSO publication. The main series start in Q1, 1960.

Member ISFILES last updated on 16.4.85.

Section 4: How to get detailed information on the names and contents of the files and series in the CSO databank

There are two ways of accessing the above information (a) in batch and (b) interactively.

(a) Batch mode is recommended as each user group should have a hard copy containing the codes and labels of each time series file. It may be necessary to update this periodically as new time series are added to the databank.

The following is the JCL required.

```
//JOBNAME JOB (ACCOUNT),'Your name',CLASS=*,MSGCLASS=A
//EXEC QA62SAS,CL=A,COPIES=1
//SYSIN DD *
  PRINDEX ALL TYPE=SHORT;
//
```

On line 1 replace 'JOBNAME' by the appropriate job name and 'account' by the relevant accounting information. Replace the '*' by the appropriate class letter assigned to you or your organisation. MSGCLASS = A specifies that the output goes to the printer - if you want to view the output on the screen (using option ISPF 3.8) first before printing use MSGCLASS = W.

On line 2: Replace CL = A by CL = W if you want to view the output first before printing otherwise the output goes straight to the printer.

(b) Interactively

This is really only useful for checking the labels associated with certain codes. It may be used to display the contents of a time series file, although only one observation is displayed on each screen at any one time. It can become quite tedious searching down through a file if there are many series in it.

Steps required: At TSO command level type QA62SAS DB and press enter. After a few moments a split screen will appear. If the message 'WELCOME TO SAS AND THE CSO DATABANK' appears press PF15 (the end key). In split screen mode the top half is used as the log and the bottom half for entering commands. Type FSINDEX TS; on line 0001 and press PF15. The arrow keys are useful for moving around the screen.

On the next few pages is a detailed account of how FSINDEX works. There is a good help facility (PF13).

To leave FSINDEX press PF15. This returns you to the original split screen. To exit from SAS type BYE on the programme editor command line and press enter. This returns you to TSO command level.

CSO DATA BANK MACRO: FSIINDEX

INTERACTIVE DISPLAY AND SEARCHING OF INDEX

Introduction:

The FSIINDEX macro, which can only be used in interactive mode, uses the SAS FSBROWSE procedure to display or search through the contents of index, the file of data bank structured documentation. Each observation, which consists of 14 variables, is formatted to fit on a single screen.

With this macro, you cannot see more than one observation on the screen at the one time. The macro FSLABEL allows this; for details enter 'HELP FSLABEL;' For further information on the file index (whose fully-qualified name is CSODB.INDEX), Enter 'Help Index;'

The Call:

This macro is not called. It takes effect as soon as it is loaded with the statement 'FSIINDEX' on line 00001 in the program editor screen. It can be loaded any number of times in the one session.

The Parameters:

TS for time series files and CS for cross sectional files.

NOTES:

The SAS procedure FSBROWSE is used for displaying or searching through the observations in a SAS data set. In this case the File CSODB.INDEX, which is sorted by code. It allows a specially formatted screen to be used for display; in this case, the format details are in the file CSODB.INDEXSCR. When you enter the load statement, the screen displays the first observation of index. The program function keys applicable here are as follows:

PF1 or PF13: Displays information about a variable
(Name, type, format, etc.)

PF2 or PF14: Returns to SAS

PF3 or PF15: Returns to the primary menu

PF5 or PF17: Repeats the last FIND (F), LOCATE (L) or SEARCH (S)
command. Each time the key is pressed, the procedure finds
the next observation meeting the search criteria

PF7 or PF19: Displays the previous observation

PF8 or PF20: Displays the next observation

PF12 or PF24: Moves the cursor to the command line

The commands that can be entered on the command line are:

NNNN: Type the number of the observation you want to display
and press enter.

F List: Finds observations meeting one or more criteria. Each
criterion is specified in the form:
'variable-name operator value'
where the operator can be any one of:
= < > > = < < =.

F @ List: Finds observations meeting any one of the listed criteria.

Name Variable: This is used in conjunction with an 'L' or 'L:' command.
It gives the name of the variable to be used in the locate
command.

L Value: Finds the next index observation whose variable in the name
command has the specified value.

L: Value: Locates values of a character variable beginning with the
specified characters.

STR Value List: This is used in conjunction with the 'S' or 'S@' command.

It identifies the variables to be searched for embedded strings.

S Character-Strings: Gives the strings to be searched for in the variables identified in the STR command.

S@ Character-Strings: Searches for observations that contain any of the specified strings.

Any F, L or S Command can be repeated by using the PF5 or PF17 key. When searching character values, you can use any combination of upper and lower case characters, except for the variables TITLE 1, TITLE 2 and LABEL, where an exact match of the characters is searched for.

If you wish, you can modify the screen temporarily or permanently. For details, see the SAS/FSP User's Guide.

Index contains 14 variables:

Character: CODE, FILE, TITLE 1, TITLE 2, START, END, LABEL

Character, Formatted: UNITS, PCODE, SACODE, CCODE

Numeric: NDEC

Numeric, Formatted: DTLU, TMLU.

For further details of Index, enter 'HELP INDEX;'.
'

For general rules on using macros, enter 'HELP MACRULES;'
'

Examples:

1. To find the next observation after 'LRAM' relating to a seasonally adjusted quarterly series:

```
F CODE>LRAM SACODE = Y PCODE = 0
```

2. To find the next observation last updated between certain dates:

```
F DTLU>1FEB85 DTLU<13MAR1985
```

3. Using name and L commands to locate certain codes:

```
Name code
```

```
L LRAM001
```

```
L: TRAM1
```


4. Using the STR and S Commands:

STR TITLE 1 TITLE 2

S MONTH VEHICLES

(If you use this, you will find monthly series relating to vehicles in the files LRBM, MIAM, TRAM, TSAM, etc.)

4. TROLL Macro for loading CSO DATA

Set out below is a copy of the TROLL macro which loads the series transferred from the CSO databank. It uses the TROLL DATABASE task to carry out the task.

```

CSODB -
&GET
&1"DO YOU WANT TO ARCHIVE THE CSO DATA? 'YES' OR 'NO' : " &END
&IF &1 CNE YES &GOTO L1 &IFEND
&GET &2"WHAT ARCHIVE NAME DO YOU WANT : " &END
SEARCH DATA &2 W;
&L1:
&GET &3"DO YOU WANT TO OVERWRITE EXISTING SERIES? 'YES' OR 'NO' : " &END
DATABASE CSO;
OVERWRITE &3;
LOAD NOARCHIVE ALL;
QUIT;
DELSEARCH ALL;

```

5. SAS MACRO

```

2 *****
*
4 *   DATE COMPLETED   :   8 JANUARY 1986
*
6 *   AUTHOR             :   JOHN COSTIGAN
*
8 *   THE ECONOMIC AND SOCIAL
*   RESEARCH INSTITUTE.
10 *
*   PROGRAM FUNCTION :   TO EXTRACT TIME SERIES FROM THE
12 *   CSO DATA BANK AND WRITE THEM TO
*   A TEMPORARY DATASET READABLE BY
14 *   TROLL.
*
16 *   MAINTENANCE      :   THIS CONCERNS INSERTING NEW TIME
*   SERIES CODES IN THE CODE LISTS,
18 *   USED BY THIS MACRO. DETAILED
*   INSTRUCTIONS ARE GIVEN BELOW.
20 *
22 *****;
%MACRO FND(FILE, CODES=, START=, END=) / STMT;
24 %TSG ALLOC F(A) DA('QA62.SAS.CSOTS') SHR;
RUN;
26 OPTIONS DQUOTE;
%LET FILE=%UPCASE(&FILE);
28 %LET SRTYR=%UPCASE(%SCAN(%SCAN(&START,1,%STR('%')),1,' '));
%IF %LENGTH(&SRTYR) = 2 %THEN %LET SRTYR = 19&SRTYR;
30 %LET STPRD=%UPCASE(%SCAN(%SCAN(&START,1,%STR('%')),2,' '));
%LET ENDYR=%UPCASE(%SCAN(%SCAN(&END,1,%STR('%')),1,' '));
32 %IF %LENGTH(&ENDYR) = 2 %THEN %LET ENDYR = 19&ENDYR;
%LET EDPRD=%UPCASE(%SCAN(%SCAN(&END,1,%STR('%')),2,' '));
34 %LET CODES=%UPCASE(%SCAN(&CODES,1,%STR('%')));
%IF %SUBSTR(&FILE,4,1) = Q %THEN %DO;
36 %IF &STPRD > 4 %THEN %LET STPRD = ;
%IF &STPRD = 2 %THEN %LET STPRD = 4;
38 %IF &STPRD = 3 %THEN %LET STPRD = 7;
%IF &STPRD = 4 %THEN %LET STPRD = 10;
40 %IF &EDPRD > 4 %THEN %LET EDPRD = ;
%IF &EDPRD = 2 %THEN %LET EDPRD = 4;
42 %IF &EDPRD = 3 %THEN %LET EDPRD = 7;
%IF &EDPRD = 4 %THEN %LET EDPRD = 10;
44 %END;
%IF %SUBSTR(&FILE,4,1) = H %THEN %DO;
46 %IF &STPRD > 2 %THEN %LET STPRD = ;
%IF &EDPRD > 2 %THEN %LET EDPRD = ;
48 %IF &STPRD = 2 %THEN %LET STPRD = 7;
%IF &EDPRD = 2 %THEN %LET EDPRD = 7;
50 %END;
%IF %SUBSTR(&FILE,4,1) = A %THEN %DO;
52 %IF &STPRD > 1 %THEN %LET STPRD = ;
%IF &EDPRD > 1 %THEN %LET EDPRD = ;
54 %END;
%IF %SUBSTR(&FILE,4,1) = L %THEN %DO;
56 %IF &STPRD > 1 %THEN %LET STPRD = ;
%IF &EDPRD > 1 %THEN %LET EDPRD = ;
58 %END;
%IF %SUBSTR(&FILE,4,1) = M %THEN %DO;
60 %IF &STPRD > 12 %THEN %LET STPRD = ;
%IF &EDPRD > 12 %THEN %LET EDPRD = ;
62 %END;
DATA SPEC;
64 FILE FT04F001 FLOWOVER;

```

INSTRUCTIONS FOR ENTERING NEW SERIES, USING AN EXAMPLE TO ADD CNBL038-CNBL040 TO THE FILE CNBL.

- (1) (A) REMOVE THE FILENAME EG. CNBL030-CNBL040 BECOMES '038 -040 '
NOTE THE SPACE INSERTED AFTER EACH NUMBER. THIS IS NECESSARY AS 4 DIGITS ARE NEEDED FOR EACH SERIES WHEN IT IS TRANSFERRED TO TROLL.
(B) INSERT THE SERIES CODES IN SEQUENCE IN THE CODE LIST, AS FOLLOWS.

BEFORE INSERTION:

CNBLVAR1=
001 -026 031 -036 041 -056 061 -086 101 -126 131 -156 161 -186 201 -226
231 -256 261 -286 303 -306 311 -326 333 -336 341 -356 363 -366 371 -386';
CNBLVAR2=
403 -406 411 -426 433 -436 441 -456 463 -466 471 -486 503 -506 511 -526
533 -536 541 -556 553 -566 571 -586 603 -606 611 -626 633 -636 641 -656
663 -666 671 -686 701 -726';
ARRAY CNBLVAR *& CNBLVAR1-CNBLVAR2;

AFTER INSERTION:

CNBLVAR1=
001 -026 031 -036 038 -040 041 -056 061 -086 101 -126 131 -156 161 -186
201 -226 231 -256 261 -286 303 -306 311 -326 333 -336 341 -356 363 -366';
CNBLVAR2=
371 -386 403 -406 411 -426 433 -436 441 -456 463 -466 471 -486 503 -506
511 -526 533 -536 541 -556 553 -566 571 -586 603 -606 611 -626 633 -636
641 -656 663 -666 671 -686 701 -726';
ARRAY CNBLVAR *& CNBLVAR1-CNBLVAR2;

IT IS IMPORTANT THAT EVERY LINE, WHICH IS NOT THE LAST IN THE LIST, (A LIST MAY COVER SEVERAL VARIABLES EG. CNBLVAR1 & CNBLVAR2) SHOULD BE EXACTLY 80 COLUMNS WIDE. THE LAST LINE MAY BE LESS THAN 80 COLUMNS.

(C) NO VARIABLE EG. CNBLVAR1, SHOULD BE MORE THAN 160 CHARACTERS LONG (OR TWO LINES) UNLESS IT IS THE LAST VARIABLE EG. CNBLVAR2. IN THIS CASE, IT MAY NOT BE MORE THAN 200 CHARACTERS IN LENGTH. IF IT IS, A THIRD VARIABLE EG. CNBLVAR3, SHOULD BE CREATED MAKING CNBLVAR2 160 CHARACTERS LONG, WITH THE REMAINDER IN CNBLVAR3. THIS NEW VARIABLE SHOULD BE INCLUDED ON THE ARRAY LINE EG.

ARRAY CNBLVAR *& CNBLVAR1-CNBLVAR3;
AIAAVAR1='001 -100 151 -250 301 -400 451 -550 601 -700 751 -850';
ARRAY AIAAVAR *& AIAAVAR1;
APBMVAR1='001 011 -018 021 -028 031 -035 041 -045 051 -053 061 -063 071 -074 081 -084 101 -125 151 -164 201 -203';
ARRAY APBMVAR *& APBMVAR1;
BCAQVAR1='001 -004 011 -014 021 -025 031 -034 041 -044 051 -054';
ARRAY BCAQVAR *& BCAQVAR1;
CNALVAR1='101 -132 201 -232 301 -332 401 -432 501 -532 601 -632 701 -732 801 -832 901 -932';
ARRAY CNALVAR *& CNALVAR1;
CNBLVAR1=
001 -026 031 -036 041 -056 061 -086 101 -126 131 -156 161 -186 201 -226
231 -256 261 -286 303 -306 311 -326 333 -336 341 -356 363 -366 371 -386';
CNBLVAR2=
403 -406 411 -426 433 -436 441 -456 463 -466 471 -486 503 -506 511 -526
533 -536 541 -556 553 -566 571 -586 603 -606 611 -626 633 -636 641 -656

```

663 -666 671 -686 701 -726';
ARRAY CNBLVAR  *% CNBLVARI-CNBLVAR2;
CRAQVARI='001 -003 011 -021 031 -043 051 -063 201 -209 221 -243 301 -325';
ARRAY CPAQVAR  *% CPAQVARI;
LRAMVARI='001 -027 051 -077 101 -127 201 -203 301 -304 501 -503 551 -553';
ARRAY LRAMVAR  *% LRAMVARI;
LRBMVARI='001 -124 151 -274 301 -424 441 -457 461 -477 481 -497';
ARRAY LRBMVAR  *% LRBMVARI;
LRCQVARI=
001 -008 011 -018 021 -028 031 -038 051 -058 061 -068 071 -078 081 -088
101 -108 111 -118 121 -128 131 -138 211 -218 221 -228 231 -238 241 -248';
LRCQVAR2=
251 -258 261 -268 271 -278 281 -288 291 -298 311 -318 321 -328 331 -338
341 -348 351 -358 361 -368 371 -378 381 -388 391 -398 411 -418 421 -428';
LRCQVAR3='431 -438 441 -448 451 -458 461 -468 471 -478 481 -488 491 -498';
ARRAY LRCQVAR  *% LRCQVARI-LRCQVAR3;
LRDHVARI=
001 -008 011 -018 021 -028 031 -038 041 -048 051 -058 061 -068 071 -078
101 -108 111 -118 121 -128 131 -138 141 -148 151 -158 161 -168 171 -178';
LRDHVAR2=
201 -208 211 -218 221 -228 231 -238 241 -248 251 -258 261 -268 271 -278
301 -308 311 -318 321 -328 331 -338 341 -348 351 -358 361 -368 371 -378';
LRDHVAR3=
401 -408 411 -418 421 -428 431 -438 441 -448 451 -458 461 -468 471 -478
501 -508 511 -518 521 -528 531 -538 541 -548 551 -558 561 -568 571 -578';
ARRAY LRDHVAR  *% LRDHVARI-LRDHVAR3;
MIAMVARI='NWD5 NWD7 001 -041 051 -053 061 -063 101 -103 201 -206 501 -541
551 -553 561 -563 601 -603';
ARRAY MIAMVAR  *% MIAMVARI;
MIAQVARI='001 -041 051 -053 061 -063 101 -103 201 -206 501 -541 551 -553
561 -563 601 -603';
ARRAY MIAQVAR  *% MIAQVARI;
MIAQVARI='001 -041 051 -053 061 -063 101 -103 201 -206';
ARRAY MIAAVAR  *% MIAAVARI;
NAAAVARI=
0101-0115 0151-0165 0201-0225 0251-0275 0301-0312 0351-0362 0401-0412 0451-0462
0501-0509 0551-0559 0601-0609 0651-0659 0701-0711 0751-0761 0801-0811 0851-0861';
;NAAAVAR2=
0901-0910 0951-0960 1001-1017 1051-1067 1101-1119 1151-1169 1201-1217 1251-1267
1301-1311 1351-1361 1401-1411 1451-1461';
ARRAY NAAAVAR  *% NAAAVARI-NAAAVAR2;
QIAGVARI='001 -041 051 -053 551 -553';
ARRAY QIAGVAR  *% QIAGVARI;
RSAMVARI='001 -016 051 -053 101 -116 501 -516 551 -516 601 -616';
ARRAY RSAMVAR  *% RSAMVARI;
TRAMVARI='001 -017 021 -023 031 -040 051 -053';
ARRAY TRAMVAR  *% TRAMVARI;
TRBMVARI='001 -004 101 -106 201 -212 221 -232 241 -252 261 -272 301 -307';
ARRAY TRBMVAR  *% TRBMVARI;
TSAMVARI='001 -075 101 -175 201 -275';
ARRAY TSAMVAR  *% TSAMVARI;
VSAQVARI='001 -003 011 -016 021 -022 031 -035 101 -115 201 -225';
ARRAY VSAQVAR  *% VSAQVARI;
F = SYMGET('FILE');
C = SYMGET('CODES');
DO N = 1, NUMOBS;
SET A.&FILE POINT=N NOBS=NUMOBS;
IF N = 1 THEN DO;
PUT F;
%IF %QUOTE(&CODES) = ALL %THEN %DO;
PUT &FILE.VAR *% '&';
%END;
%ELSE %DO;
C = TRANSLATE(C,' ',F);
C = COMBBL(C);

```

```

C = TRANWRD(C, '- ', ' ');
%IF &FILE = NAAA %THEN %DO;
C = TRANWRD(C, ' ', ' ');
C = TRANWRD(C, '- ', ' ');
%END;
PUT C '&';
%END;
M = MONTH(DATE);
Y = YEAR(DATE);
%IF &SRTYR NE 1 %THEN %DO;
%IF &STPRD EQ %THEN %DO;
%PUT INVALID START PERIOD - DEFAULTS TO SERIES START DATE;
%END;
%ELSE %DO;
IF MDY(&STPRD,1,&SRTYR) >= DATE THEN DO;
M = SYMGET('STPRD');
Y = SYMGET('SRTYR');
END;
%END;
%END;
IF LENGTH(M) EQ 1 THEN PUT Y 4. @11 M 1.;
ELSE PUT Y 4. @10 M 2.;
CALL SYMPUT('SRTYR',Y);
CALL SYMPUT('STPRD',M);
END;
IF N = NUMOBS THEN DO;
M1 = MONTH(DATE);
Y1 = YEAR(DATE);
%IF &ENDYR NE 2 %THEN %DO;
%IF &EDPRD EQ %THEN %DO;
%PUT INVALID END PERIOD - DEFAULTS TO SERIES END DATE;
%END;
%ELSE %DO;
IF MDY(&EDPRD,1,&ENDYR) <= DATE THEN DO;
M1 = SYMGET('EDPRD');
Y1 = SYMGET('ENDYR');
END;
%END;
%END;
CALL SYMPUT('ENDYR',Y1);
CALL SYMPUT('EDPRD',M1);
IF LENGTH(M1) EQ 1 THEN PUT Y1 4. @11 M1 1.;
ELSE PUT Y1 4. @10 M1 2.;
END;
END;
STOP;
DATA OUTPUT;
FILE FT04F001 MOD ;
SET A.&FILE;
%*****
%#
%# IT IS NOT A COMPLICATED MATTER TO INSERT NEW TIME SERIES CODES :
%# IN THE FOLLOWING CODE LISTS. THERE IS NO RESTRICTION ON THE LENGTH:
%# OF EACH LIST, ONLY THE SEQUENCE IS IMPORTANT. CONTINUING THE ABOVE;
%# EXAMPLE TO ADD CNBL038-CNBL040 :
%# :
%# :
%# BEFORE INSERTION :
%# :
%# %IF &FILE = CNBL %THEN %DO;
%# ARRAY CNBLST I& CNBL001-CNBL026 CNBL031-CNBL036 CNBL041-CNBL056 ;
%# CNBL061-CNBL086 ;
%# CNBL101-CNBL126 CNBL131-CNBL156 CNBL161-CNBL186 ;
%# CODES DELETED FOR CONVENIENCE
%# CNBL201-CNBL226 CNBL231-CNBL256 CNBL261-CNBL286 ;
%# CNBL641-CNBL666 CNBL671-CNBL686 CNBL701-CNBL726;
%# %END;
%#

```

```

%* AFTER INSERTION :
%*
%* %IF &FILE = CNBL %THEN %DO;
%* ARRAY CNBLLST I& CNBL001-CNBL026 CNBL031-CNBL036 CNBL038-CNBL040 ;
%* CNBL041-CNBL056 CNBL051-CNBL086 ;
%* CNBL101-CNBL126 CNBL131-CNBL156 CNBL161-CNBL186 ;
%* CODES DELETED FOR CONVENIENCE ;
%* CNBL201-CNBL226 CNBL231-CNBL256 CNBL261-CNBL286 ;
%* CNBL641-CNBL666 CNBL671-CNBL686 CNBL701-CNBL726;
%* %END;
%*****
%IF &FILE = AIAA %THEN %DO;
ARRAY AIAALST I& AIAA001-AIAA100 AIAA151-AIAA250 AIAA301-AIAA400 ;
AIAA451-AIAA550 AIAA601-AIAA700 AIAA751-AIAA850;
%END;
%IF &FILE = APBM %THEN %DO;
ARRAY APBMLST I& APBM001 APBM011-APBM018 APBM021-APBM028 ;
APBM031-APBM035 APBM041-APBM045 APBM051-APBM053 ;
APBM061-APBM063 APBM071-APBM074 APBM081-APBM084 ;
APBM101-APBM125 APBM151-APBM164 APBM201-APBM203;
%END;
%IF &FILE = BCAQ %THEN %DO;
ARRAY BCAQLST I& BCAQ001-BCAQ004 BCAQ011-BCAQ014 ;
BCAQ021-BCAQ025 BCAQ031-BCAQ034 ;
BCAQ041-BCAQ044 BCAQ051-BCAQ054;
%END;
%IF &FILE = CNAL %THEN %DO;
ARRAY CNALLST I& CNAL001-CNAL032 CNAL101-CNAL132 CNAL201-CNAL232 ;
CNAL301-CNAL332 CNAL401-CNAL432 CNAL501-CNAL532 ;
CNAL601-CNAL632 CNAL701-CNAL732 CNAL801-CNAL832 ;
CNAL901-CNAL932;
%END;
%IF &FILE = CNBL %THEN %DO;
ARRAY CNBLLST I& CNBL001-CNBL026 CNBL031-CNBL036 CNBL041-CNBL056 ;
CNBL061-CNBL086 ;
CNBL101-CNBL126 CNBL131-CNBL156 CNBL161-CNBL186 ;
CNBL201-CNBL226 CNBL231-CNBL256 CNBL261-CNBL286 ;
CNBL303-CNBL306 CNBL311-CNBL326 CNBL333-CNBL356 ;
CNBL363-CNBL366 CNBL371-CNBL386 CNBL403-CNBL406 ;
CNBL411-CNBL426 CNBL433-CNBL456 CNBL441-CNBL456 ;
CNBL463-CNBL466 CNBL471-CNBL486 CNBL503-CNBL506 ;
CNBL511-CNBL526 CNBL533-CNBL536 CNBL541-CNBL556 ;
CNBL571-CNBL586 CNBL603-CNBL606 CNBL611-CNBL636 ;
CNBL641-CNBL666 CNBL671-CNBL686 CNBL701-CNBL726;
%END;
%IF &FILE = CPAQ %THEN %DO;
ARRAY CPAQLST I& CPAQ001-CPAQ003 CPAQ011-CPAQ021 CPAQ031-CPAQ043 ;
CPAQ051-CPAQ063 CPAQ201-CPAQ209 CPAQ221-CPAQ243 ;
CPAQ301-CPAQ325;
%END;
%IF &FILE = LRAM %THEN %DO;
ARRAY LRAMLST I& LRAM001-LRAM027 LRAM051-LRAM077 LRAM101-LRAM127 ;
LRAM201-LRAM203 LRAM301-LRAM304 LRAM501-LRAM503 ;
LRAM551-LRAM553;
%END;
%IF &FILE = LRBM %THEN %DO;
ARRAY LRBMMLST I& LRBM001-LRBM124 LRBM151-LRBM274 LRBM301-LRBM424 ;
LRBM441-LRBM457 LRBM461-LRBM477 LRBM481-LRBM497;
%END;
%IF &FILE = LRCQ %THEN %DO;
ARRAY LRCQLST I& LRCQ001-LRCQ008 LRCQ011-LRCQ018 LRCQ021-LRCQ028 ;
LRCQ031-LRCQ038 LRCQ051-LRCQ058 LRCQ061-LRCQ068 ;
LRCQ071-LRCQ078 LRCQ081-LRCQ088 LRCQ101-LRCQ108 ;
LRCQ111-LRCQ118 LRCQ121-LRCQ128 LRCQ131-LRCQ138 ;
LRCQ211-LRCQ218 LRCQ221-LRCQ228 LRCQ231-LRCQ238 ;
LRCQ241-LRCQ248 LRCQ251-LRCQ258 LRCQ261-LRCQ268 ;

```

LRCQ271-LRCQ278 LRCQ281-LRCQ288 LRCQ291-LRCQ298
 LRCQ311-LRCQ318 LRCQ321-LRCQ328 LRCQ331-LRCQ338
 LRCQ341-LRCQ348 LRCQ351-LRCQ358 LRCQ361-LRCQ368
 LRCQ371-LRCQ378 LRCQ381-LRCQ388 LRCQ391-LRCQ398
 LRCQ411-LRCQ418 LRCQ421-LRCQ428 LRCQ431-LRCQ438
 LRCQ441-LRCQ448 LRCQ451-LRCQ458 LRCQ461-LRCQ468
 LRCQ471-LRCQ478 LRCQ481-LRCQ488 LRCQ491-LRCQ498;

%END;

%IF &FILE = LRDH %THEN %DO;

ARRAY LRDHLST I& LRDH001-LRDH008 LRDH011-LRDH018 LRDH021-LRDH028
 LRDH031-LRDH038 LRDH041-LRDH048 LRDH051-LRDH058
 LRDH061-LRDH068 LRDH071-LRDH078
 LRDH101-LRDH108 LRDH111-LRDH118 LRDH121-LRDH128
 LRDH131-LRDH138 LRDH141-LRDH148 LRDH151-LRDH158
 LRDH161-LRDH168 LRDH171-LRDH178 LRDH201-LRDH208
 LRDH211-LRDH218 LRDH221-LRDH228
 LRDH231-LRDH238 LRDH241-LRDH248
 LRDH251-LRDH258 LRDH261-LRDH268 LRDH271-LRDH278
 LRDH301-LRDH308 LRDH311-LRDH318 LRDH321-LRDH328
 LRDH331-LRDH338 LRDH341-LRDH348 LRDH351-LRDH358
 LRDH361-LRDH368 LRDH371-LRDH378 LRDH401-LRDH408
 LRDH411-LRDH418 LRDH421-LRDH428 LRDH431-LRDH438
 LRDH441-LRDH448 LRDH451-LRDH458 LRDH461-LRDH468
 LRDH471-LRDH478 LRDH501-LRDH508 LRDH511-LRDH518
 LRDH521-LRDH528 LRDH531-LRDH538 LRDH541-LRDH548
 LRDH551-LRDH558 LRDH561-LRDH568 LRDH571-LRDH578;

%END;

%IF &FILE = MIAM %THEN %DO;

ARRAY MIAMLST I& NWD5 NMD7 MIAM001-MIAM041 MIAM051-MIAM053
 MIAM061-MIAM063 MIAM101-MIAM103 MIAM201-MIAM206
 MIAM501-MIAM541 MIAM551-MIAM553 MIAM561-MIAM563
 MIAM601-MIAM603;

%END;

%IF &FILE = MIAQ %THEN %DO;

ARRAY MIAQLST I& MIAQ001-MIAQ041 MIAQ051-MIAQ053 MIAQ061-MIAQ063
 MIAQ101-MIAQ103 MIAQ201-MIAQ206 MIAQ501-MIAQ541
 MIAQ551-MIAQ553 MIAQ561-MIAQ563 MIAQ601-MIAQ603;

%END;

%IF &FILE = MIAA %THEN %DO;

ARRAY MIAALST I& MIAA001-MIAA041 MIAA051-MIAA053 MIAA061-MIAA063
 MIAA101-MIAA103 MIAA201-MIAA206;

%END;

%IF &FILE = NAAA %THEN %DO;

ARRAY NAAAALST I& NAAA0101-NAAA0115 NAAA0151-NAAA0165 NAAA0201-NAAA0225
 NAAA0251-NAAA0275 NAAA0301-NAAA0312 NAAA0351-NAAA0362
 NAAA0401-NAAA0412 NAAA0451-NAAA0462 NAAA0501-NAAA0509
 NAAA0551-NAAA0559 NAAA0601-NAAA0609 NAAA0651-NAAA0659
 NAAA0701-NAAA0711 NAAA0751-NAAA0761 NAAA0801-NAAA0811
 NAAA0851-NAAA0861 NAAA0901-NAAA0911 NAAA0951-NAAA0960
 NAAA1001-NAAA1017 NAAA1051-NAAA1067 NAAA1101-NAAA1119
 NAAA1151-NAAA1169 NAAA1201-NAAA1217 NAAA1251-NAAA1267
 NAAA1301-NAAA1311 NAAA1351-NAAA1361 NAAA1401-NAAA1411
 NAAA1451-NAAA1461;

%END;

%IF &FILE = QIAQ %THEN %DO;

ARRAY QIAQLST I& QIAQ001-QIAQ041 QIAQ051-QIAQ053 QIAQ551-QIAQ553;

%END;

%IF &FILE = RSAM %THEN %DO;

ARRAY RSAMLST I& RSAM001-RSAM016 RSAM051-RSAM053 RSAM101-RSAM116
 RSAM551-RSAM553 RSAM601-RSAM616;

%END;

%IF &FILE = TRAM %THEN %DO;

ARRAY TRAMLST I& TRAM001-TRAM017 TRAM021-TRAM023 TRAM031-TRAM040
 TRAM051-TRAM053;

%END;

%IF &FILE = TRBM %THEN %DO;


```

ARRAY TRBMLST I& TRBM001-TRBM004 TRBM101-TRBM106 TRBM201-TRBM212
TRBM221-TRBM232 TRBM241-TRBM252 TRBM261-TRBM272
TRBM301-TRBM307;
%END;
%IF &FILE = TSAM %THEN %DO;
ARRAY TSAMLST I& TSAM001-TSAM075 TSAM101-TSAM175 TSAM201-TSAM275;
%END;
%IF &FILE = VSAQ %THEN %DO;
ARRAY VSAQLST I& VSAQ001-VSAQ003 VSAQ011-VSAQ016 VSAQ021-VSAQ022
VSAQ031-VSAQ035 VSAQ101-VSAQ115 VSAQ201-VSAQ225;
%END;
%IF &SR TYR NE 1 %THEN %DO;
BEG = MDY(&STPRD,1,&SR TYR);
IF NOT (DATE >= BEG) THEN DELETE;
DROP BEG;
%END;
%IF &ENDYR NE 2 %THEN %DO;
FIN = MDY(&EDPRD,1,&ENDYR);
IF NOT (DATE <= FIN) THEN STOP;
DROP FIN;
%END;
OUTYEAR = YEAR(DATE);
PUT OUTYEAR 4.;
%IF %QUOTE(&CODES) NE ALL %THEN %DO;
PUT (&CODES) (20.9);
%END;
%ELSE %DO;
DO OVER &FILE.LST;
PUT &FILE.LST 20.9 @;
END;
PUT;
%END;
%MEND FND;
RUN;
%LET FILE1 = %SCAN(&SYSPAR4,1,#);
%LET CODE2 = %SCAN(&SYSPAR4,2,#);
%LET STRT3 = %SCAN(&SYSPAR4,3,#);
%LET ENDY4 = %SCAN(&SYSPAR4,4,#);
FND &FILE1 CODES=%STR('&CODE2') START=%STR('&STRT3') END=%STR('&ENDY4');

```

6. Copying TROLL DATA FILES to non-TROLL DISK FILES

One or more TROLL data files can be copied to a sequential or partitioned dataset in a form in which they can be accessed, either by other computer programmes, or can be readily copied to tape. This job is accomplished using the TROLL DATABASE task using the command STORE to copy each series to the external file. Set out below is an example of a job to copy three series X, Y and Z:

```
DATABASE CSO;
STORE X Y Z;
QUIT;
```

The "DATABASE CSO;" command is essential no matter what data are to be transferred. The store command can take a list of variables or names defined using the "*" or ">" symbols.

The disk file to which the series are transferred is specified using the DISK option when first logging into TROLL, e.g., TRL12 DISK('TA64.TEMP'). The format in which the series are stored is as follows:

Record columns format variable

1	1-4	I4	Number of observations in the series
	5-8	I4	Number of TROLL name segments in the series name
	9-12	I4	Periodicity of the data
	13-16	I4	Start year
	17-20	I4	Start period
	21-24	I4	End year
	25-28	I4	End period
	29-36	A8	First name segment - TROLL user ID
	37-44	A8	Second name segment - DATA
	45-52	A8	Third name segment
	53-60	A8	Fourth name segment (if any)
	61-68	A8	Fifth name segment (if any)

	69-76	A8	Sixth name segment (if any)
2	1-20	F20.9	First observation
	21-40	F20.9	Second observation
	41-60	F20.9	Third observation
	61-80	F20.9	Fourth observation

Additional records are used to contain the rest of the observations in the series. The number of records is determined by the number of observations in the series, specified in Record 1.

Records 1 and 2 are then repeated for each successive series down to the end of the list specified in the STORE command.

Set out below is a FORTRAN subroutine which will read the data from the disk file as well as an example of the output.

```

C-----SAMPLE SUBROUTINE TO READ DATA FROM A DISK FILE.THE DATA-----
C-----ON THE FILE WERE STORED THERE USING THE TROLL DATABASE TASK.-----
C-----THE DISK FILE IS REFERRED TO AS DEVICE 4.-----
C-----THIS SUBROUTINE SHOULD BE CALLED REPEATEDLY TILL THE CONTROL-----
C-----VARIABLE I2 IS RETURNED WITH THE VALUE 1 INDICATING AN END OF----
C-----FILE.-----
      SUBROUTINE TRLDAT(NAME,NOBS,STYEAR,STPER,ENYEAR,ENPER,PER,DATA,I2)
C-----
C   THE CAST IN ORDER OF APPEARANCE:
C   NOBS   = NUMBER OF OBSERVATIONS IN THE SERIES
C   LNAME  = THE NUMBER OF NAME SEGMENTS IN THE TROLL NAME
C   PER    = THE PERIODICITY OF THE SERIES
C   STYEAR = THE START YEAR OF THE SERIES
C   STPER  = THE START PERIOD OF THE SERIES
C   ENYEAR = THE END YEAR OF THE SERIES
C   ENPER  = THE END PERIOD OF THE SERIES
C   NAMES  = THE FULL TROLL NAME OF THE SERIES
C   DATA  = THE DATA
C   NAME   = THE FINAL NAME SEGMENT OF THE TROLL NAME
C   I2     = A CONTROL VARIABLE:ON RETURN 1 IF END OF FILE REACHED.
C           ON RETURN 0 IF MORE DATA TO BE READ.
C-----
      REAL*8 NAME,NAMES(6)
      REAL*4 DATA(2000)
      INTEGER PER,STYEAR,STPER,ENYEAR,ENPER,NOBS,LNAME
      I2=0
      READ(4,1,END=100)NOBS,LNAME,PER,STYEAR,STPER,ENYEAR,ENPER,NAMES
      READ(4,2)(DATA(I),I=1,NOBS)
      NAME=NAMES(LNAME)
      RETURN
1   FORMAT(7I4,6A8)
2   FORMAT(4F20.0)
100  I2=1
      RETURN
      END

```

53	5	121971	11975	5TA64MTR DATA	PRICE	CSO	LRAM001
				0.623999953		0.632999957	0.635999978
				0.636999965		0.640999973	0.640999973
				0.642999947		0.647999942	0.650999963
				0.657999992		0.664999962	0.666999996
				0.667999983		0.674999952	0.675999999
				0.681999981		0.687999964	0.691999972
				0.698999941		0.708999991	0.713000000
				0.717999995		0.723999977	0.723999977
				0.724999964		0.738999963	0.745999992
				0.750999987		0.759999990	0.763999999
				0.768999994		0.773999989	0.774999976
				0.777999997		0.786999941	0.789999962
				0.796999991		0.804999948	0.810999990
				0.815999985			

53	5	121971	11975	5TA64MTR DATA	PRICE	CSO	LRAM002
				0.482999980		0.484999955	0.486999989
				0.488999963		0.493999958	0.494999945
				0.494999945		0.496999979	0.498999953
				0.498999953		0.501999974	0.503999949
				0.504999995		0.508999944	0.508999944
				0.510999978		0.513999999	0.515999973
				0.516999960		0.525999963	0.529999971
				0.532999992		0.537999988	0.546999991
				0.548999965		0.552999973	0.560999990
				0.565999985		0.572999954	0.582999945
				0.589999974		0.594999969	0.606999993
				0.614999950		0.625000000	0.629999995
				0.631999969		0.638999999	0.642999947
				0.644999981			

53	5	121971	11975	5TA64MTR DATA	PRICE	CSO	LRAM102
				6240.000000000		6330.000000000	6360.000000000
				6370.000000000		6410.000000000	6410.000000000
				6430.000000000		6480.000000000	6510.000000000
				6580.000000000		6650.000000000	6670.000000000
				6680.000000000		6750.000000000	6760.000000000
				6820.000000000		6880.000000000	6920.000000000
				6990.000000000		7090.000000000	7130.000000000
				7180.000000000		7240.000000000	7240.000000000
				7250.000000000		7390.000000000	7460.000000000
				7510.000000000		7600.000000000	7640.000000000
				7690.000000000		7740.000000000	7750.000000000
				7780.000000000		7870.000000000	7900.000000000
				7970.000000000		8050.000000000	8110.000000000
				8160.000000000			