

WRITE

```
WRITE (FILE='filename string',  
      FORMAT=BINARY or DATABANK or EXCEL or  
      FREE or LABELS or LOTUS or RB4 or RB8 or '(format text string)',  
      FULL,UNIT=I/O unit number)  
      list of variables ;
```

Function:

WRITE is used to write variables to the screen, output file, or an external file. The output may be labelled, free format, or a format of your specification. WRITE can create Lotus, Excel, and binary files. PRINT is synonymous with WRITE.

Usage:

WRITE is the inverse of the READ statement: series or other variables which are read with a particular READ statement may be written by a WRITE statement of the same form.

When the list of variables contains only series, WRITE writes one record for each observation in the current sample (unless the format statement specifies more than one record); this record contains the value of all the series listed on the statement for that observation. If only one series is listed, WRITE writes at least one observation per record.

If the list of variables on the WRITE statement includes some which are not series, the variables are written to the file one at a time, with one record per variable. Symmetric, triangular, and diagonal matrices are written in compressed storage mode unless the FULL option is specified. You are responsible for making sure your format allows for enough data points if you use the FORMAT='format string' option. Subscripted matrices are treated like scalars.

In labelled or FREE format: series, scalars, and matrices, missing values are shown as the character . (period) and a warning is printed.

A single WRITE command will write either a single matrix or several series to a spreadsheet file. If the file specified already exists, it is overwritten by the new file. If you write a matrix, the file will consist only of numbers. If you write series, their names will be put in the first row. The first column will contain dates (or observation numbers), and each series will be put in a column below its name. Series are written under the control of the current sample. If there are gaps in the sample, observation numbers will be used instead of dates in the first column. Dates are written as the last day of each period, and formatted as Month-Year.

Options:

FILE= *filename string* specifies the actual name of the file where the data is to be written.

FORMAT=BINARY or **DATABANK** or **EXCEL** or **FREE** or **LABELS** or **LOTUS** or **RB4** or **RB8** or *'(format text string)'* specifies the format in which the data is to be written or printed. The default is LABELS unless the FILE option is also specified, in which case the default is FREE. Each format option is described below.

WRITE

FORMAT=BINARY specifies that the data is to be written in single precision (REAL*4) format on the external file. This format for data is by far the most efficient if you do not plan to move the data to another computer and should be used, if possible, if you have a large amount of data. To read such a data file, use the READ command with the **FORMAT=BINARY** option. This format cannot be used for equations.

FORMAT=DATABANK specifies that the data are to be written to a TSP databank. This option requires the FILE= option also.

FORMAT=EXCEL writes an Excel spreadsheet file (similar to **FORMAT = LOTUS**). If the filename ends .XLS, this is the default.

FORMAT=FREE specifies that the data is to be written to an external file in a format determined by TSP. This option causes the data to be represented by six numbers per record with a field width of 15 characters and at least 7 significant digits. The exact format is chosen by the program to represent the numbers most conveniently.

FORMAT=LABELS specifies that the data is to be formatted as for printed output, with observation labels if they are series, and row and column numbers and titles if they are vectors or matrices. This option is the default if the output is being written to the output file or screen, or if the item being written is an equation.

FORMAT=LOTUS writes a Lotus 123 .WKx worksheet file. The variable names are written in the first row atop the series columns. If a **FREQ** other than N is in effect, dates are written in the leftmost column. **FORMAT=LOTUS** is the default if the filename string includes .WK . (See the READ command.)

FORMAT=RB4 is the same as **FORMAT=BINARY** (single precision binary).

FORMAT=RB8 is used for double precision binary.

FORMAT='(format text string)' specifies a fixed format with which the data is to be written. The quotes are required and should surround a Fortran **FORMAT** statement, including the parentheses but excluding the word **FORMAT**. If you are unfamiliar with the construction of a Fortran **FORMAT** statement, see **FORMAT**.

FULL/NOFULL specifies whether symmetric, triangular, and diagonal matrices are to be expanded before being written out. This option is ignored with **FORMAT=DATABANK** or **FORMAT=LABELS**.

UNIT= an integer number (usually between 1 and 4, or 8 and 99) which is the Fortran input/output number of an external file from which the variables listed will be written. This is rarely used.

Examples:

This example is the inverse of the example for the READ command.

```
WRITE(FILE='FOO.DAT') X Y Z ;
```

To check some transformed series:

```
PRINT X,LX,DX ;
```

The following example creates a spreadsheet file equivalent to the READ command examples:

```
FREQ Q;
```

WRITE

```
SMPL 48:1,49:1;  
READ CJMTL; 183.4 185.2 192.1 193.3 206.9; )  
READ PMTL; . .436 .562 .507 .603;  
WRITE(FILE='SML.WKS') CJMTL,PMTL;
```

The next example creates a spreadsheet file with the same data columns, but no dates or series names (since a matrix is used).

```
MMAKE M CJMTL,PMTL;  
WRITE(FILE='SMM.WKS') M;
```

SMM.WKS file (as shown in lotus):

	A	B
1	183.4	NA
2	185.2	.436
3	192.1	.562
4	193.3	.507
5	206.9	.603

Output:

WRITE produces printed output in the output file or screen, unless the FILE= option is used, in which case data are written to an external data file.

A single WRITE; or PRINT; command (without any options or arguments) can be used to restore echoing of commands to the output file. The NOPRINT; command can be used to stop such echoing.