READ (BYOBS, BYVAR, FILE=*'filename string'* or filename,
FORMAT=BINARY or DATABANK or EXCEL or FREE or LOTUS or RB8 or '(*format string*)',
FULL,NCOL=*number of columns*,NROW=*number of rows*, PRINT,SETSMPL,
TYPE=CONSTANT or DIAG or GENERAL or SYMMETRI or TRIANG, UNIT=*I/O unit number*)
list of series or matrices or constants ;
or
READ ;

Function:

READ is used to read series, matrices, or scalars into data storage. Normally, data will be read from an external file, but small quantities of data can be included in a "data section" at the bottom of your program file (when running TSP in batch mode). If you plan to repeatedly use a very large dataset, the fastest way to access it is as a TSP databank - see the OUT and IN commands for further details.

The most common usage of READ is to read several variables from an external file in free format, such as:

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

READ with no arguments is used in the TSP program section to transfer to the "data section" (see the STOP command for an example) and begin reading data until an END statement is encountered. Use this feature if you like to keep your data separate from your program, but in the same file.

Usage:

If TSP encounters a simple READ ; statement, it transfers to the data section and begins reading data until it reaches an END; statement (or end of file), at which point it returns to the line in the TSP program following the READ; statement. A NOPRINT; command in the data section will stop the data values from echoing.

A READ statement followed by a list of series names is the easiest way to read small quantities of data. If no options are specified, the data is assumed to follow the READ statement directly in free format, each number separated from the others by one or more blanks. Each group of data may be terminated by a semicolon (;), although this is not required. If there is more than one series to be read in, the order of the data is the first observation of each of the series, followed by the second observation of each of the series, and so forth.

There are two special features for free-format numbers. The first is the use of the dot (.) to specify a missing value (and is similar to the SAS convention). However, note that in other places in TSP (such as in formulas), dot (.) is treated as a DOT variable. Use @MISS or @NA for this purpose in a formula (see FRML). The other special feature is a repeated number, specified by an integer repeat count, a star (*), and the value to be repeated. For example, 3*0 is equivalent to 0 0 0. This is most often used for repeated zeroes in special matrices (like band matrices) and resembles the repeat count feature in the FORTRAN free-format (*) READ.

When the data is read in free format, the number of items read must be equal to the number of series times the number of observations in the current SMPL. TSP checks for this and prints a message when the check fails. TSP will determine the length of the SMPL itself if the SETSMPL option is on.

To read matrices use a READ statement with options to define the matrix, and the matrix's name for storage. Follow the statement with the numbers which compose the matrix in free format, a row at a time. That is, a 3 (# rows) by

2 (#of columns) matrix is read in the following order: (1,1), (1,2), (2,1), (2,2), (3,1), (3,2).

A matrix of any type may be read by specifying all its elements, but there are special forms for reading symmetric, triangular, and diagonal matrices. If a matrix is symmetric, only the lower triangle needs to be read, i.e., elements (1,1), (2,1), (2,2), (3,1), (3,2), (3,3), and so forth. The FULL option specifies whether the full matrix is being specified or only the lower triangle. If the matrix is triangular, you need only specify the transpose of the upper triangular portion: (1,1), (1,2), (2,2), (1,3), (2,3), (3,3), and so forth. If the matrix is diagonal, only the diagonal needs to be given: (1,1), (2,2), (3,3), ... It will be filled out with zeroes when used.

Spreadsheet files:

TSP can read series and matrices directly to or from spreadsheet files. The following files are supported by TSP:

Spreadsheet*	**Version**	**Filename Extensions**	**TSP Support**
Lotus 123, Symphony	1,2	.WKS .WK1 .WRK .WR1	Read, Write
Lotus 123	3	.WK3	Read
Lotus 123/J (Japanese)	1,2	.WJ1 .WJ2 .WK2 .WT2	Read, Write
Microsoft Excel	2	.XLS	Read, Write
Microsoft Excel	3,4	.XLS	Read
Microsoft Excel	5	.XLS, .XLW	not supported
Quattro Pro		.WQ1	Read (PC only)

* Note that TSP generally writes the oldest file formats, which are always readable by more recent spreadsheet releases.

Files should be in the format of the following example, SML.XLS, which consists of quarterly data, 48:1 to 49:1):

	A	B	C
1	Date	CJMTL	PMTL
2	Mar-48	183.4	#N/A
3	Jun-48	185.2	.436
4	Sep-48	192.1	.562
5	Dec-48	193.3	.507
6	Mar-49	206.9	.603

The above file could be read with the following command:

```
READ (FILE='SML.XLS) ;
```

Series are read from individual columns in the file. Series names are optionally supplied in the first row of the file (aligned) above the data columns. Dates may be given in the first column. Many different file configurations are possible, and it is possible to read in some series while ignoring others. Following are some simple guidelines for creating a spreadsheet file for TSP:

1. Put the column names in the first row. They should be valid series names in TSP (lowercase is fine - it will be converted to uppercase, but imbedded blanks and special characters are not allowed). If the file has no names, you can supply them when you read the file in TSP, but this can be inconvenient. If the file contains invalid names, the data can be read by TSP as a matrix, ignoring the current names, and you can supply your own names inside of TSP. TSP will not recognize names in lower rows or in "sheets below the first" (Lotus

version 3); they will be treated as missing values and numbers below them will be read as data for the original column names.

2. The second row must contain data.

3. If you are reading time series, the first column should contain dates. This will ensure the series are read with the proper frequency and starting date, regardless of the current FREQ and SMPL in TSP. Dates can be strings such as 48:1 or numbers formatted as dates (Mar-48, 3/31/48, etc.). You only have to supply enough dates so that TSP can detect the frequency (5 is enough to distinguish between quarterly and monthly). TSP ignores any dates after these, assuming that the data is contiguous (no missing periods/years, or SMPL gaps in TSP terminology). If you have missing periods/years for all series, leave the corresponding rows blank. Below is a table of examples showing recommended ways of defining the starting date and frequency with a dates column:

first date	**second date**	**resulting frequency**
(string dates - first character is ’ ^ or " in Lotus)		
1	2	A if current FREQ is A, otherwise N
48:	any	A
48:5	any	M
48:4	49:1	Q
1948:1	1948:2	M or Q, depending on further dates
48:1	49:1	A
a2:3	any	invalid
48:2	48:1	invalid
(numeric dates)		
12/31/48	12/31/49	A (any dates 365-366 days apart)
12/31/48	1/31/49	M (any dates 28-31 days apart)
12/31/48	3/31/49	Q (any dates 90-92 days apart)
12/31/48	1/ 1/49	N (any other date range)

If you have dates in other columns, they will be read as numbers. If you are reading a matrix, the date column will be ignored (i.e. it will not be read into the matrix).

4. Missing values can be represented by blank cells or by formulas which evaluate to NA, @NA, #N/A, etc.

To read in a spreadsheet file, use the FILE=’filename’ option. FORMAT=LOTUS or EXCEL is optional. If the filename contains one of the extensions listed earlier (.WKS, .WK3, .XLS, etc.), TSP checks the first few bytes of the file to confirm that it is one of the spreadsheet versions listed above. Conversely, if FORMAT=LOTUS or EXCEL is specified, but the filename does not contain a extension , then .WKS or .XLS is appended to the filename. To read a matrix (bypassing column names and dates), use the TYPE=GEN option. TYPE=CONSTANT is not supported for Lotus files; series will be defined instead. The NCOL=, NROW=, IFULL, UNIT=, etc. options are ignored, but SETSMPL is supported.

If no series names are supplied on the READ command, TSP looks for column names in the file and creates series with those names. If you supply series names, TSP attempts to match them to column names in the file. If the file does not have column names, you must supply a READ argument for each data column. If you are unsure of the file’s contents, check it with your spreadsheet or read it as a matrix. If you think the file has column names, but you don’t know what they are, try supplying a dummy name which won’t be matched. TSP will print an error message listing the column names in the file. If you are reading a matrix (using TYPE=GEN as mentioned above), TSP will create a matrix named @LOTMAT unless you supply an argument (matrix name) to READ.

If for some reason your series are in rows instead of columns, you can read the file as a matrix, transpose it, and UNMAKE the matrix into series.

TSP checks the first row in the file for string names (Lotus cells beginning with the characters ' ^ or "). The names are truncated to 8 characters (if necessary), and are translated to uppercase. They must be aligned above their corresponding data columns. If you have dates in the first column, no name is required for the date column. Any names with imbedded blanks will be ignored.

Options:

BYOBS/NOBYOBS specifies that the data is organized by observation **--** the first observation for all series, then the second observation for all series, etc. This is the default.

BYVAR/**NOBYVAR** specifies that the data is organized by series: all observations for the first series, then all observations for the second series, etc.

FILE=*'filename string'* or *filename* specifies the actual name of the file where your data is stored. If the filename string is 8 characters or less and does not include non-alphabetic or lowercase characters, it does not need to be enclosed in quotes.

FORMAT=BINARY or DATABANK or or EXCEL or **FREE** or LOTUS or RB4 or RB8 or '(*format text string*)' specifies the format in which your data is to be read. The default is free format, which means the fields (numbers) are separated by blanks or tabs and may be of varying length. Each of the format options is described in more detail below, and under FORMAT in this manual.

FORMAT=BINARY specifies that the data is in binary single precision (REAL*4) format. To read data in this format, it must be on an external file, since binary data cannot be intermixed in a TSP program input file. This method of reading data is quite fast **-** about the same as a TSP databank, but not quite as easy to use. This is the same as FORMAT=RB4. FORMAT=RB8 is for double precision binary.

FORMAT=DATABANK specifies that the data are to be read from a TSP databank.

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

FORMAT=LOTUS reads Lotus 123, Excel, or Quattro Pro worksheet files (most files with the extension .WKx, where *x* is any character).

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

FORMAT=RB8 is used for double precision binary.

FORMAT= a format string enclosed in quotes '(*format string*)' specifies the format with which the data are to be READ. 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 the FORMAT section of this *Reference Manual*.

FULL/**NOFULL** applies only to reading diagonal, symmetric, or triangular matrices. It specifies whether the complete matrix is to be read, or only the upper triangle (in the case of triangular), the lower triangle (symmetric), or the diagonal (diagonal).

NCOL= the number of columns in the matrix. This is required for a general matrix.

NROW= the number of rows in the matrix. This is required for a general matrix.

Either NROW or NCOL must be specified for symmetric, triangular, or diagonal matrices. These options only apply to matrices.

PRINT/NOPRINT specifies whether or not the data is to be printed as it is READed. This option applies only to free format READing. It may be set globally for the READ section by use of the NOPRINT statement (see NOPRINT).

SETSMPL/NOSETSMP specifies whether the SMPL is to be determined from the number of data items read. The default is on (SETSMPL) if no SMPL has been specified yet in the program and off otherwise. This option does not apply to matrix reading.

TYPE=GENERAL or SYMETRIC or TRIANG or DIAG or CONSTANT specifies the type of the matrix which is to be READed. GENERAL, the default, may be used for any rectangular or square matrix. SYMETRIC implies that the matrix is equal to its transpose; only the lower triangle will be stored internally to save space. TRIANG implies that the matrix is triangular (has zeroes above the diagonal). Although a lower triangular matrix is read, its transpose is stored since the TSP matrix procedures expect upper triangular matrices. DIAG means a matrix all of whose off-diagonal elements are zero. Only the diagonal is stored, and it is expanded before use. CONSTANT means a scalar or scalars are to be READed and none of the matrix options will apply. If no type is specified, a warning is printed and the matrix is assumed to be general.

UNIT= an integer number (usually between 1 and 4, or 8 and 99) which is the Fortran input/output unit number of an external file from which the variables listed will be READ. Usually, just FILE= is used, but UNIT= could be used to avoid typing in a long filename for several READ commands from the same file.

Examples:

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

A simple READ of one series in free format:

```
SMPL 1 9 ;
READ IMPT ; 100 106 107 120 110 116 123 133 137 ;
```

This example reads formatted data from the TSP input file:

```
SMPL 1 50 ;
READ (FILE='STATES.DAT',FORMAT='(F2.0,F4.0,3F4.1,F6.0/F7.0,5F4.1,F4.0)' )
   STATE X1-X12
```

where STATE.DAT contains:

```
01284952.326.4 1.9  4120
   19055553.320.813.7 8.9 5.46677
024592 0.021.211.0   403
   303168.176.819.815.5 8.29047
 ....and so forth....
```

After this data set has been read, the series STATE and X1 through X12 have the following values:

```
STATE: 1,2,.....
X1: 2849,4592,.....
X2: 52.3,0.0,.....
.....
```

```
X11: 8.9,15.5,.....
X12: 5.4,8.2,......
X12: 6677,9047,.....
```

Examples of reading matrices:

```
READ (NROW=4,NCOL=3) COEFMAT ;
0.32 0.5 1.3
0.30 0.4 1.35
0.25 0.61 1.1
0.28 0.55 1.23
;
READ (NROW=2,TYPE=SYM) COVAR ;
4.64 2.3 5.1 ;
READ (NROW=3,TYPE=TRIANG) TMAT ;
1
2 3
4 5 6 ;

READ (NCOL=5,TYPE=DIAG) BAND ;
110. 140. 0. 35. 50. ;
```

The matrices stored by these four examples are the following:

```
        COEFMAT              COVAR          TMAT              BAND

0.32    0.50    1.3      4.64    2.3      1    2    4     110    0  0   0   0
0.30    0.40    1.35     2.30    5.1      0    3    5       0  130  0   0   0
0.25    0.61    1.1                       0    0    6       0    0  0   0   0
0.28    0.55    1.23                                        0    0  0  35   0
                                                            0    0  0   0  50
```

Examples for spreadsheet files:

We will use this SML.WKS file (it is the Lotus version of the SML.XLS file shown earlier) in the following examples:

	^CJMTL	^PMTL
’48:1	183.4	NA
’48:2	185.2	.436
’48:3	192.1	.562
’48:4	193.3	.507
’49:1	206.9	.603

```
READ(FILE=’SML.WKS’);          ?  the series CJMTL and PMTL are defined.  FREQ Q and
                               ?  SMPL 48:1, 49:1 are set if there is no current FREQ or SMPL.

READ(FILE=’SML.WKS’,TYPE=GEN);    ?  creates the 5x2 matrix @LOTMAT, with the values
                                  ?  of CJMTL and PMTL in its columns.

READ(FILE=’SML.WKS’) PMTL;        ?  only reads in PMTL
```

Here is the nm3.wk1 file (as shown in Lotus, using numeric dates) for the following examples:

04/30/57	23.2	34.5	10.9
05/31/57	23.6	35.1	11.0
	23.9	35.8	11.2
	24.0		11.5

```
READ(FILE='NM3.WK1') SF LA SD;    ?  defines the monthly series SF, LA, and SD from 57:4 to 57:7.
                                  ?  If there is no current sample, this is the new sample with FREQ M.
                                  ?  The series LA will be given a missing value in its last observation.

READ(FILE='NM3.WK1') SF;    ? An error message is printed because three series names are required.
```

Output:

READ prints the data as it is read when the free format options is used and the PRINT options is on, otherwise READ produces no output, and stores all the data read in data storage under the series names specified.

Reference:

Fortran manuals for your computer installation will usually give particulars about Fortran formats.