READ PARAMETER

PURPOSE

Reads a parameter into DATAPLOT:

1. from a mass storage file; or
2. from within a CALLed DATAPLOT sub-program; or
3. from the terminal.

DESCRIPTION

The rules regarding READ PARAMETER are as follows:

1. Only one line is read. Unlike a READ of variables, no END OF DATA is searched for.
2. If more than one parameter is read, enter them on the same line separated by one or more spaces.
3. In scanning for the parameter, the full line image is scanned (for reading from a mass storage file, the full line image is 132 columns; for reading from within a sub-program and for reading from the terminal, the full line image is 80 columns). For variations on this, see the COLUMN LIMITS command.
4. Parameters can be free format. They need not be aligned in specific columns.
5. By default, all reads start from the beginning of the file (to override this this, see the SKIP and ROW LIMITS commands).

SYNTAX 1

READ PARAMETER <p1> <p2> ... <pk>
where <p1>, <p2>, ..., <pk> are the desired parameters.

This syntax is used to read parameters from the terminal or from a DATAPLOT sub-program. For example, READ PARAMETER A B.

SYNTAX 2

READ PARAMETER <file> <p1> <p2> ... <pk>
where <file> is the name of the mass storage file where the parameters reside;
and <p1>, <p2>, ..., <pk> are the desired parameters.

This syntax is used to read parameters from a file. For example, READ PARAMETER CALIB.DAT A B C.

EXAMPLES

```
READ PARAMETER CALIB. P1 P2
```

```
READ PARAMETER P
 5.2
```

NOTE 1

In order to determine whether the first argument is a file name or a parameter name, it looks for a period in the name. If it finds one, it assumes a file name. If it does not, it assumes a parameter name. If your file name does not contain a period, attach a trailing period (no spaces) to the file name on the READ command.

NOTE 2

DATAPLOT has no restrictions on the file name other than it be a valid file name on the local operating system and that it contain a period "." in the file name itself or as a trailing character. DATAPLOT strips off trailing periods on those systems where it is appropriate to do so. On systems where trailing periods can be a valid file name (e.g., Unix), DATAPLOT tries to open the file with the trailing period. If this fails, it then tries to open the file with the trailing period stripped off.

Some users prefer to give all data files a ".DAT" or ".dat" extension. Although this is a useful method for keeping track of data files, it is strictly a user convention and is not enforced by DATAPLOT in any way.

NOTE 3

File names are case sensitive on Unix file systems. For Unix, DATAPLOT attempts to open the file as given. If this fails, it attempts to open the file as all upper case characters. If this fails, it attempts to open the file as all lower case characters. All other currently supported systems are not case sensitive regarding file names.

As a further caution for Unix hosts, certain expansion characters (specifically ~ to refer to your home directory) are interpreted by the shell and are not recognized by the Fortran compiler. These expansion characters are interpreted as literal characters and do not yield the intended file name.

DEFAULT

1. If no file name in the READ PARAMETER command is specified, and if a CALL is being executed, then the parameter should be listed directly in the DATAPLOT sub-program immediately after the READ PARAMETER command.
2. If no file name in the READ PARAMETER command is specified, and if commands are being manually entered/executed 1-at-a-time from the terminal, then the parameter should be entered directly from the terminal immediately after the READ PARAMETER command.

RELATED COMMANDS

SERIAL READ	=	Perform a serial read.
READ	=	Read variables.
READ FUNCTION	=	Read a function.
READ MATRIX	=	Read a matrix.
READ STRING	=	Read a string.
LET	=	Define a parameter.

APPLICATIONS

Data input

IMPLEMENTATION DATE

Pre-1987

PROGRAM

```
. PURPOSE--GENERATE AND EXECUTE MENU FOR T TESTS ON 2 SUBSETS
. FROM A COMMON RESPONSE VARIABLE.
FEEDBACK OFF
ERASE
SET IO TERMINAL
.
WRITE “I-----------------------------------------I “
WRITE “I MENU FOR T-TEST ON SUBSETS OF THE DATA I “
WRITE “I-----------------------------------------I “
WRITE “ “; WRITE “ “
WRITE “THIS MENU ASSUMES THAT YOU HAVE ALREADY”
WRITE “DATA IN THE 2 VARIABLES-- “
WRITE “ Y = THE RESPONSE VARIABLE “
WRITE “ TAG = AN IDENTIFIER VARIABLE “
WRITE “ WHICH SPECIFIES DIFFERENT “
WRITE “ SUBSETS OF Y. “; WRITE “ “
WRITE “THIS MENU ASKS WHICH 2 SUBSETS YOU WANT COMPARED, “
WRITE “AND THEN CARRIES OUT THE T TEST. “
.
WRITE “ “
WRITE “TAG FOR SUBSET 1 = ?”
.
READ PARAMETER S1
WRITE “TAG FOR SUBSET 2 = ?”
READ PARAMETER S2
.
DELETE Y1 Y2; LET Y1 = Y; LET Y2 = Y
RETAIN Y1 SUBSET TAG S1; RETAIN Y2 SUBSET TAG S2
PRINT “THE TAGS WERE ^S1 AND ^S2 “
PRINT Y1
PRINT Y2
T-TEST Y1 Y2
SET IO
```