SERIAL READ

PURPOSE

Reads data into variables:

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

DESCRIPTION

The rules regarding SERIAL READ are as follows:

1. If there are k variables listed on the SERIAL READ statement, then all the data within each line image is read and sequentially fed into the next available elements of the k variables. There is no restriction on the number of data values on any given line image. The major difference between the READ and SERIAL READ commands is that READ reads "down" while SERIAL READ reads "across." Another major difference between the READ command and the SERIAL READ command is that the READ command only reads in the first k data values on a line whereas the SERIAL READ command reads in all data on a line. In practice, the READ command is more heavily used than the SERIAL READ command. An example of the use of the SERIAL READ command is

   ```
   SERIAL READ X Y
    1 1 2 4 3 9 4 16 5
    25 6 36 7 49
    8 64
    9 81
    10 100
   END OF DATA
   ```

   which results in X containing the 10 elements 1, 2, ... 10 and Y containing the 10 elements 1, 4, 9, ... 100. Note that READ X Y of the above data would result in X containing the 5 elements 1, 25, 8, 9, 10 and Y containing the 5 elements 1, 6, 64, 81, 100.
2. The number of variables being read at one time must be between 1 and 10, inclusive.
3. 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. Data values on a line image must be separated by at least one blank.
5. Data values can be free-format. They need not be aligned in specific columns.
6. The format of individual data values is general. It can be integer, floating point, or exponential. It is stored internally as a single precision real number.
7. By default, all reads start from the beginning of the file (to override this, see the SKIP and ROW LIMITS commands).
8. The analyst need not be concerned about the number of observations for each variable. DATAPLOT automatically determines and reports this value at the end of the read.
9. The read terminates when a line image is encountered which consists of

   END OF DATA (or END DATA)

   or when the end of the file is reached.

SYNTAX 1

SERIAL READ <x1> <x2> ... <xk>
where <x1>, <x2>, ..., <xk> are the desired variable names.

This syntax is used to read the data from the terminal or from a DATAPLOT sub-program. For example, SERIAL READ X Y.

SYNTAX 2

SERIAL READ <file> <x1> <x2> ... <xk>
where <file> is the name of the mass storage file where the data resides;
and <x1>, <x2>, ..., <xk> are the desired variable names.

This syntax is used to read the data from a file. For example, SERIAL READ CALIB.DAT X Y.

EXAMPLES

```
SERIAL READ CALIB. PRES TEMP TIME
SERIAL READ ASTM. Y1 Y2 Y3 X DAY LAB
SERIAL READ Y1 Y2 Y3 X DAY LAB
SERIAL READ Y
```

NOTE 1

The most common use of SERIAL READ is to read a large number of values, with many values per line, into a single variable. For example,

```
NLIST RANDN.DAT
SKIP 25
SERIAL READ RANDN.DAT Y
PLOT Y
```

NOTE 2

By default, DATAPLOT does free format reads. However, it has the capability for supporting FORTRAN style formats. Formatted reads can be about 10 times faster on many systems which can be helpful for large data files. Enter HELP READ FORMAT for more details.

NOTE 3

Blank lines in data files are ignored.

NOTE 4

DATAPLOT supports the ability to embed comment lines within the data file. Enter HELP COMMENT CHECK for details.

NOTE 5

In order to determine whether the first argument is a file name or a variable 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 variable 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 6

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 7

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.

NOTE 8

DATAPLOT searches for the specified file in the current working directory. If it does not find the file, it tries to find the file in the default DATAPLOT directory. For this reason, all of the DATAPLOT sample data files will be found by simply entering the name with no additional directory information needed. For example,

```
SKIP 25
READ MAVRO.DAT Y
PRINT Y
```

The list of sample data files is given in the DATAPLOT Reference Files chapter. If your implementation of DATAPLOT does not automatically find the sample data files, contact your local site installer.

DEFAULT

If no file name in the SERIAL READ command is specified, and if a CALL is being executed, then the data values should be listed directly in the DATAPLOT sub-program immediately after the SERIAL READ command (do not forget the END OF DATA statement).

If no file name in the SERIAL READ command is specified, and if commands are being manually entered/executed one at a time from the terminal, then the data should be entered directly from the terminal immediately after the SERIAL READ command (also terminated by an END OF DATA statement).

SYNONYMS

None

RELATED COMMANDS

READ FUNCTION	=	Read a function.
READ MATRIX	=	Read a matrix.
READ PARAMETER	=	Read a parameter.
READ STRING	=	Read a string.
READ	=	Perform a read.
SET READ FORMAT	=	Define a FORTRAN style format for reads.
DATA (LET)	=	Enter data values into a variable.

APPLICATIONS

Data input

IMPLEMENTATION DATE

Pre-1987

PROGRAM

```
NLIST RANDN.DAT
SKIP 25
READ RANDN.DAT Y
PLOT Y
```