SPREADSHEET GUIDE - PART FOUR

Command Reference Guide

Expression entry

An expression is rather like a mathematical formula that can be entered into a given cell. The formula normally remains invisible but the Notebook spreadsheet will work out the resulting solution and display that in the cell when it is on screen.

An expression is typed in on the edit line (after Next:) and on completion is inserted into the memory together with an order of calculation number. On completion, the memory remaining is also calculated and displayed and if automatic calculation is switched on, the worksheet is recalculated.

As you type the value of the expression will be worked out and displayed in the target cell, but the expression itself will be displayed inside brackets on the cell contents line at the top.

Evaluation of expressions

Expressions may contain functions, co-ordinate references, numbers, arithmetic operators, logical operators and special purpose operators. Expressions are evaluated according the rules of normal algebra.

As in algebra all operators are given an importance which determines the order in which the various parts of the expression are calculated. This order of importance is over-ridden by parentheses, each set of which is evaluated as if it were a self-contained expression. The innermost set of parentheses is calculated first and then the next innermost and so on working outwards.

In order to multiply two sets of parentheses, an asterisk symbol must be put between them, this is the computing convention for a multiply sign. You cannot have (2+5)(3+7) to multiply the two expressions; instead you must put (2+5)*(3+7).

The operators within the expression or set of parentheses are calculated in the following order:

()	parenthesis
+n, -n	unary plus and minus
n1%n2	percentage
n1^n2	raise n1 to the power of n2
n1*n2, n1/n2	multiplication and division
n1+n2, n1-n2	addition and subtraction
n1>n2, n1<n2, n1=n2	greater than' and 'less than' and 'equal'
n1]n2, n1[n2	greater than or equal' and 'less than or equal'
not(n1)	not.

If an exclamation mark is encountered on the entry line then the value of the expression to that point is determined and replaces the whole of the expression on the edit line.

If, for example, you have typed an expression such as 2*B1+20 the resulting value displayed in the cell will change if the entry that is held in B1 ever changes. By inserting an exclamation mark thus 2*B1!+20 the expression is calculated using the value that is held in B1 at that time. The resulting entry is therefore constant and no longer dependent on B1.

Alternatively, you could use the hash symbol # after the B1 and just the value of B1 would be replaced in the expression. By using 2*B1# you would get 2*1.17 on the edit line if cell B1 contained 1.17.

The ! should be used with caution as indeterminate results may occur if, for example, it is used from within a function.

Building expressions

The valid parts of an expression are as follows: numbers, co-ordinates, operators, conditionals, functions and user defined functions.

Numbers can be either be entered as normal digits or can be entered in exponential, scientific format. You could, for example, either enter 274529000 or 2.74529E8 (or 0.274529E9 or 27.4529E7 etc.). You can enter up to 15 significant digits. Internal calculations are performed to 16 or 17 significant figures. The range of numbers you may use is from 1E-38 to 1E+38. That is from

.000,000,000,000,000,000,000,000,000,000,000,000,01 up to 10,000,0-00,000,000,000,000,000,000,000,000,000,000.

Co-ordinates are references to other cells in the sheet and consist of a letter A..Z or a..z which identifies the column, followed by a 1 to 3 digit line number. It is very important to get the case of the column letter correct. The first 26 columns are capital A..capital Z. The next 26 columns are lower case a..lower case z.

Operators, listed below, are the normal mathematical symbols such as "+", "-", etc. Most are used in the form "expression" *operator* "expression".

A conditional is like an operator but is used to build an expression that evaluates to an answer of TRUE or FALSE (these are actually represented by the numeric values -1 and 0). Conditionals are normally used with the IF() or WHILE() functions. You might, for example, have IF(A1>7)... which means "if the value in cell A1 is greater than 7 do the THEN() part of the expression.". In this case the ">" is the conditional operator.

Functions comprise a named operation with its required parameters in brackets and where there are more than one, these parameters must be separated by commas. There are three types of function, those whose arguments have multiple parameters, those which have single parameters and those which have no parameters at all.

When an apparent function name is encountered it is checked against the list of built in functions. Functions may call other functions in their parameters. Indeed a function may even call itself amongst the parameters.

You can have a function as a parameter to a function as a parameter to a function... etc. However, there is a limit to the amount of "nesting" you can use. All you need to know is that if the nesting has become too great (i.e. if there are too many levels of brackets in your expression) an error message will come up and it will only be necessary to rearrange the expression so that it is placed into two entries rather than one.

A user-defined function acts just like one of the built in functions but makes use of a formula that you have entered onto the sheet and is called by giving the reference to the cell containing the formula to use with any parameters listed in brackets after it.

Expressions are evaluated as you enter them, not on completion. This means that you have dynamic error checking character by character. If you enter a wrong character you will get an immediate error message. To correct an error, backspace with the ⌫ key and then enter a new character.

Note that it is impossible for the Notebook spreadsheet to check for errors that are 'legal' in the context of the program but are not the correct expression that you intended to enter. For example, if you enter A1+3 when you really meant to type A3+1 no error can be detected because the spreadsheet thinks A1+3 is a valid expression.

If you are entering a complex expression such as a DO-WHILE formula then the calculation character by character can get bogged down. This is because it will be trying to do the whole DO loop as you enter the instructions. In this case first switch off the automatic calculation. This is done with the Automatic Calculation command.

The valid operators in an expression

Arithmetic Operators

+ Addition
- Subtraction
* Multiplication
/ Division
^ Exponentiation (raising to a power)
% Percentage

Logical Operators

Logical Operators act on arithmetic values or expressions (A and B below) and produce a result of TRUE (-1) if the condition is met or FALSE (0) if it is not.

=	Equal	(A = B)
>	Greater than	(A > B)
<	Less than	(A < B)
]		Greater than or equal (A] B)
[		Less than or equal (A [B)
			Not equal (A | B)
		(the | symbol is typed with ⌐─┐)

Built in functions

The following pages contain a list of all the functions recognised by the Notebook spreadsheet. They are given in alphabetic order. The following are one or two notes about some of the specific types of function.

Trigonometric functions

COS, COSR, SIN, SINR, TAN, TANR

There is no restriction on the size passed to these functions as 2*PI or 360 degrees will be repeatedly subtracted until X is within range. X may be negative.

ACOS, ACOSR, ASIN, ASINR

The value passed to these functions must be between -1 and 1

Logical Functions

You can enter the functions FALSE or TRUE where a logical expression is expected. FALSE will result in a conditional test failing while TRUE will make it succeed. The actual display in a cell holding the FALSE or TRUE function will be 0 or -1 respectively.

Logical functions analyse a list of logical values or expressions (given as X, Y, Z below) and return a value of TRUE or FALSE as defined above. In practice each expression or value is tested to see if it is -1 or 0. If it is -1 it is taken as TRUE, if it is 0 it is taken as FALSE.

Date and Time functions

Where a "date" is called for or returned by a function it is an expression of a particular date as a single number made up as follows - ddmm.yyyy. The dd part is 1 or 2 digits that specify the date (1..31), the mm part must always be given or read as two digits giving the number of the month (1..12). For months 1..9 it is 01..09. The yyyy part is the year and is always four digits.

Alphabetic list of built in functions

ABS(X)
The absolute value of X is returned. It is defined as the numerical value of X with a positive sign.

```
ABS(-2.345) = 2.345
ABS(2.345) = 2.345
```

ACOS(Y)
ASIN(Y)
ATAN(Y)
These functions determine the angles in degrees whose cosine, sine or tangent respectively is given by Y. The result is given in the range -90 degs to 90 degs.

ACOSR(Y)
ASINR(Y)
ATANR(Y)
These functions determine the angles in radians whose cosine, sine or tangent respectively is given by Y. The result is given in the range -PI/2 to PI/2.

AND(X, Y, Z.)
X, Y, Z. are in turn checked for truth. If they are all TRUE then the function returns a value of TRUE (-1).

```
AND (TRUE, TRUE, TRUE) = TRUE
AND (TRUE, FALSE, TRUE) = FALSE
```

AVERAGE(list)
The average of the non blank values in the list is calculated.

```
AVERAGE(3,4,5) = 4
```

BLEEP
Sets off a bleep. Use it as an audible signal. Can be useful to indicate that a recalculation is complete. Arrange for it to have the highest order of calculation number by including a reference to the last result calculated in the cell that contains BLEEP.

For example:

```
B9, BLEEP
```

CHOOSE(N, list)
The nearest integer value to N is found and the Nth item in the list is returned.

```
CHOOSE(3, 5,6,7,8,9) = 7
```

COMB(N, R)
The number of ways of combining R items from a total of N is calculated.

```
COMB(8, 3) = 56
```

COS(X)
COSR(X)
Returns the cosine of X where X is in degrees for COS and in radians for COSR.

```
COS(45) = 0.707106781186548
```

COUNT(list)
The list is checked for the number of values that are not blank.

```
COUNT(2,3,4,B5) = 3          cell B5 is blank.
```

CRD(column, line)
Can be used in some functions where a co-ordinate is expected. The column must, however, be expressed in a numerical form (A=1, Z=26 etc.). This function is useful in filling up tables from DO iterations.

```
SET(CRD(3,7), 10)          sets cell C7 to the value 10
```

DATE
Returns the date in the form of a single number, for example

```
312.1987 - being the 3rd December 1987
```

DATEAFTER(date, days)
Gives the date that will be the number of days specified after the input date. Be careful to put months 1 to 9 as 01 to 09.

```
DATEAFTER(1308.1992, 50) = 210.1992
50 days after 13th August 1992 is 2nd October 1992
```

DAY
Gets the current day of the month from the Notebook's internal clock. The result will be a number between 1 and 31.

DAYOFWK(date)
Returns the day of week as a number. Saturday has a value 0, Sunday 1, Monday 2 etc.

```
DAYOFWK(1308.1992) = 5 = Thursday

DAYOFWK(DATE) = 3 if date is 904.1963
              = Tuesday 9th April 1963
```

DAYOFYR(date)
Returns the number of days between January 1st and the given date. In this case, the date does not need to include a year but remember that leap years include an extra day.

```
DAYOFYR(904) = 99   9th of April is 99th day of the year
```

DAYSAPART(date1, date2)
Gives the number of days between any two specified dates.

```
DAYSAPART(904.1993, 2512.1993) = 260
```

DEC(X)
Returns the value of X minus 1. This is also intended for DO-WHILE iterations, primarily to act as a counter. This is a command function that will change the value held in the cell specified.

```
INIT(B1,10), DO(A1...A4), DEC(B1), WHILE(B1>0)
```

DELAY(*n*)

Does nothing until *n* seconds have elapsed. It may be a cell reference or a value. For practical reasons make it a cell reference with a value 0 until you actually want to run you application.

DO(*range*)

Allows subroutines, looping and iteration on the specified *range*. If iteration or looping is required the line will need to be terminated with a WHILE function.

DPART(*X*)

Takes the value of the decimal part of *X*, that is the part after the decimal point. Be warned that this function loses one significant figure for each figure that was previously before the decimal point. If you get unexpected results consider this.

```
DPART(5.78)   = 0.78
DPART(-3.45)  = -0.45
```

e

Gives the exponential constant *e* - the base of natural logarithms.

```
e=2.7182818284905
```

ERROR

A special function that if encountered in a calculation will bring up a message that an error has been called from that cell. Usually used in a check that values entered are in a permissible range.

```
IF (A1=0), THEN(ERROR), ELSE(3/A1)
prevents an attempt to divide by 0
```

EXP(*X*)

Raises the constant e to the power in the bracket following. A check is made to ensure that the exponent is not too great. This gives the natural anti-logarithm. Natural logarithms are calculated using LN().

```
EXP(3.14159265358979) = 23.1406926327793
```

FACT(*X*)

Works out the factorial of *X*. If *X* is not an integer it is rounded up or down to the next nearest integer before the factorial is calculated.

```
FACT(5.2) = 5*4*3*2*1 = 120
```

FALSE

Takes a value of FALSE (0). This may be used in logical expressions.

GROW(*value, percentage*)

Will increase a value by a percentage. The percentage can be negative.

```
GROW (300, 5) = 315
```

HOUR

Uses the Notebook's internal clock to give the current hour as a number between 0 and 23.

IF(*logic expr*), THEN(*expr 1*), ELSE(*expr 2*)

This group is used to build decision making into the worksheet. The *logical expression* is analysed and if TRUE, THEN *expression 1* is used to get the value for the cell. If the *logical expression* is FALSE then the ELSE *expression 2* is used instead. The expressions themselves can be further IF... sequences building up extremely complex decision making sequences.

IN(*port*)

Reads an 8 bit port given by the number or cell reference 'port'. Only included for compatibility with other versions of the Cracker spreadsheet. Its use on the Notebook should be avoided at all cost.

INC(*X*)

Returns the value of *X* plus 1. It is general purpose but is mainly included for convenience in creating loops. It is a command function that will update the value held in the cell specified.

```
INC(2.345) = 3.345
INC(A3) will add one to whatever is held in A3 and then
write it back.
```

INIT(*crd, val*)

Will initialise a cell within a specified value. The cell containing this function will have a low recalculation number. The function is used for setting up loops and iterations. It is very similar to the SET function except that it is guaranteed a low recalculation number.

```
INIT(B7, 10) will set cell B7 to contain the value 10.
```

INT(*X*)

Takes the value of the integer part of *X*, that is the part before the decimal point, with the sign retained.

NOTE: this definition may not be the same as that to which you are used if you are familiar with some computer languages such as BASIC. The function NINT() behaves more like you might expect.

```
INT(5.78) = 5
INT(-3.45) = -3
```

INTERP(*N, range*)

Compares *N* with each value in the *range* to find the two values between which *N* lies. The two values from the adjacent line or column (to the right or below) are then interpolated to give the final value. The interpolation is done by taking the proportionate distance that *N* lies between the first two values and applying it to the adjacent values.

LN(*X*)

Determines the natural logarithm to the base *e* of the value in brackets. The routine will check for negative or zero arguments which are not allowable.

```
LN(10) = 2.30258509299405 and EXP(2.302585...) = 10
```

LOG10(*X*)

Determines the logarithm to the base 10 of the value in brackets.

```
LOG10(1234) = 3.09131515969722 (and 10^3.09131... = 1234)
```

LOOKUP(*N, range*)

N is compared with each value in the *range* to find the first one it is greater than. The value from the adjacent line or column (below or to the right) is then returned.

MAINTITLE(*crd*)

A graphics function explained in the description of the Trace command and Tutorial XV.

MAX(*list*)

The *list* is scanned and the maximum non blank value returned.

```
MAX(2, 7, 3) = 7
```

MIN(*list*)

The minimum non blank value in the *list* is returned.

```
MIN(2, 7, 3) = 2
```

MINUTE

Reads the Notebook's clock and returns the current minute as a number between 0 and 59.

MONTH

Reads the Notebook's clock and returns the current month as a number between 1 and 12.

NINT(*X*)

The nearest integer value to *X* is found.

```
NINT(2.23) = 2
NINT(5.67) = 6
NINT(-3.45) = -3
```

NOT(*X*)

The truth of *X* is checked and the opposite is returned as the function value. If the value of *X* is TRUE then FALSE (0) is returned. If the value is FALSE then TRUE(-1) is returned.

```
NOT(TRUE) = FALSE          NOT(3>5) = TRUE
```

NPV(%, *list*)

Calculates the present value of the cash flow list using the discount % as specified. See tutorial for example.

OR(*X, Y, Z*)

X, Y, Z are in turn checked for truth. If any of them are TRUE then the function returns a value of TRUE (-1)

```
OR(TRUE, FALSE, FALSE)  = TRUE one item is TRUE
OR(FALSE, 35, 7) = FALSE as none is TRUE
```

OUT(*port, value*)

Outputs a *value* given by a number or cell reference to the *port* given by a number or cell reference. This should not be confused with the OUT command which sends characters specifically to the printer.

NOTE: This function is only included for compatibility with other versions of the Cracker spreadsheet. Its use on the Notebook is to be avoided at all costs as it will almost certainly crash the computer and the only recourse will be to switch on holding down the [Function][Stop] and [⌂] keys. All data in the Notebook will then be lost.

PERM(*N, R*)

Finds the number of ways of permuting *R* items from a total of *N*.

```
PERM(5, 3) = 60
```

PI

Returns the value of the constant PI - the ratio of a circle's circumference to its diameter.

```
PI = 3.14159265358979
```

POS(*X*)

Gives the value of the argument if it is positive. If it is negative then it returns a value of 0.

```
POS(2.345) = 2.345
POS(-2.345) = 0
```

RND

An integer random number between 0 and 127 is found. This number is taken from the Z80 refresh counter. A new value is returned each time the function is recalculated.

```
RND = 23
RND = 120
RND = 3
```

ROUND(*X*)

Takes the value of *X* rounded to two decimal places. It is of use in financial calculations to avoid cumulative errors caused by including fractions of a penny (cent etc.). The rounded value is not just displayed; it is also the value stored in the internal memory.

```
ROUND(1357.5679) = 1357.57
```

SECOND

The Notebook's clock is read and the current second is returned as a number between 0 and 59.

SET(*crd, value*)

Will set a cell to a particular *value*. The cell must initially be either blank or a constant. A formula will not be over-written. This command function is usually used with the DO function to fill up a table with values.

SIMPRULE(*step, range*)

Works out the numerical integration of the range by Simpson's rule. The first parameter is the *step* length. *Range* holds the values for integration. The range must be an odd number of values consecutively. The latter parts of the range can be blank and these will be ignored. This makes flexibility in setting up worksheets possible. An example is included in the tutorial.

SIN(*X*)

Determine the sine of angle *X*, where *X* is in degrees.

```
SIN(45) = 0.707106781186548
```

SINR(*X*)

Calculate the sine of angle *X*, where *X* is in radians.

```
SINR(PI/4) = 0.707106781186548
```

SQRT(*X*)

Determines the square root of *X*.

```
SQRT(2) = 1.4142135623731
```

STDEV(*list*)

Looks through the *list* and works out the standard deviation.

SUBTITLE(*crd*)

A graphics function explained in the description of the Trace command and in Tutorial XV.

SUM(*list*)

Will add up the non blank values in a list and will return the total.

```
SUM(2, 3, 4) = 9
SUM(-3, 4, 5, A4) = 4 if cell A4 contain -2
```

SUMIF(*crd1, crd2..crd3*)

This is a conditional adding up function. Only those items in the range *crd2...crd3* will be included that correspond to a cell entry in the column specified by *crd1* and are on the same line.

```
           A        B
        =======  =======
1                123.45
2       MARKER   23.45
3                245.56
4       MARKER   45.67

SUMIF(A1, B1...B4) gives 69.12
SUM(B1...B4) gives 438.13
```

NOTE: *crd1* can point to any cell in the column but must not point to a cell with a text entry in it or you will get an error message.

TAN(*X*)

Determine the tangent of angle *X*, where *X* is in degrees. *X*=90 or -90 will give an error as the result is infinite.

TANR(*X*)

Calculate the tangent of angle *X*, where *X* is in radians. *X*=PI/2 or *X*=-PI/2 will give an error as the result is infinite.

TIME

Reads the Notebook's clock and returns the time in the form of a single number, for example 1503.23 being 3 minutes and 23 seconds past 3 in the afternoon.

TIMELABEL(*crd...crd*)

A graphics function explained in the description of the Trace command and Tutorial XV.

TIMELAPSE

Returns the elapsed time since the ZEROTIME function was last operative. This is in seconds.

TRUE

Returns a value of TRUE, that is -1. May be used in logical expressions.

TYPEPLOT(*crd*)

A graphics function explained in the description of the Trace command and Tutorial XV.

VAL(*column, line*)

Gives the numerical value of the cell specified by the two parameters. The co-ordinates are defined in the same way as for the CRD function.

```
VAL(8, 3) = 7 if cell H3 contains 7
```

VAR(*list*)

Looks through the list and works out the variance. The variance is the standard deviation squared.

WHILE(*expr*)

Must be used together with a DO function. The expression must be logical and if TRUE the line will be recalculated from the start of the DO on that line. If FALSE the WHILE will do nothing.

XLABEL(*crd...crd*), or XLABEL(*crd, crd, crd,* **etc.)**
XMAXIMUM(*crd*)
XMINIMUM(*crd*)
XTITLE(*crd*)
XVALUE(*n, crd...crd*)

These are graphics functions explained in the description of the Trace command and Tutorial XV.

YEAR

Reads the Notebook's clock and gives the current year as a four digit number.

YMAXIMUM(*crd*)
YMINIMUM(*crd*)
YTITLE(*crd*)
YVALUE(*n, crdcrd*)

These are graphics functions explained in the description of the Trace command and Tutorial XV.

ZEROTIME

Resets the elapsed time counter. Probably best included in an IF, THEN, ELSE entry and used with TIMELAPSE.

Text entry

If you use the . command and the column was created with a default format set at Text, or use the ⌐-Format command and choose a Text format, or use the Global Format form of entry with the " command, then you will be set up for entering text. A marker will be put on the edit line to show the current column width, but you are free to over-write this with a longer entry. The marker just shows how much of the entry will display in the cell with its current width setting.

The prompt `Enter characters` will come up. Any number or character may be inserted and if a mistake is made the ⌐ key may be used to go back and correct errors. If the entry has been completed and you want to move to an adjacent cell and insert information there, then you can use the arrow keys instead of the ↵ key. This does, of course, mean that you cannot move back and forth along the line you are entering using the arrow keys. If you notice an error towards the start of the line finish entry and then use the Edit command to make the necessary amendments.

Complete command reference

⌐ - Automatic calculation command
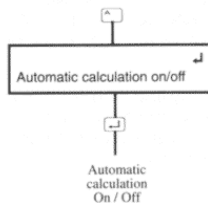
Automatic calculation, or perhaps it should be called automatic recalculation, is the feature by which as you enter any new data or formulae, or replace existing data by something new, the Notebook spreadsheet automatically updates the displayed worksheet in order to fully reflect these changes.

When you first start automatic calculation is switched on. You can turn off the automatic calculation by typing an ⌐ followed by ↵. Similarly, a second entry of ⌐ followed by ↵ will turn it on again.

As you get more experienced with the program you will probably find it useful to switch off the calculation process while entering long or complex formulae or perhaps large tables of data. The few seconds taken up each time to calculate and display the results can slow down the overall entry process.

When you switch on the automatic calculation again a full recalculation will immediately be carried out. If you wish to leave the automatic calculation off permanently then you can use the ! - Force recalculation command to force a recalculation at any time you would like it.

Automatic calculation proceeds as you type when you are entering new data or formulae. This is an important part of the Notebook spreadsheet's error catching features. As soon as a calculation or command has been entered that the spreadsheet cannot deal with the calculation will stop and you will know exactly where the mistake lies.

The only exception to the above rule is when you are using the Edit command to make small changes to existing entries. In this case calculation is suppressed until the editing session is completed or you would find that as soon as you delete a certain part of a formula it is probable that an error would be found. The Notebook spreadsheet therefore automatically turns off the calculation until the editing is finished, at which time a full update of the sheet is made.

New users of the sheet may find it preferable to avoid using the EDIT function and simply delete and re-enter any formulae that you wish to change. You are then assured of the benefits of error checking and the command line prompts.

B - Blank command

Whole sheet is blanked | No action | Current cell entry is blanked | Block of cells between coordinate1 (top left) and coordinate2 (bottom right) are blanked | One or more lines are blanked | One or more columns are blanked
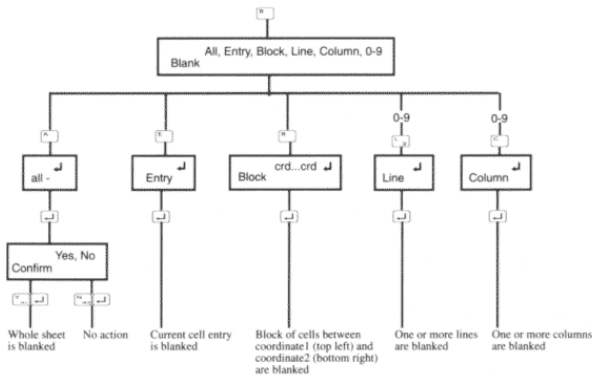
Blank removes the specified entries from memory but does not affect the structure of the worksheet. Blanked entries cannot be retrieved so if in doubt you should first copy the unaltered worksheet to a .MEM file before starting on complex rearrangements.

Before the blanking is carried out the program will check whether any of the items to be blanked are referred to elsewhere in the worksheet. If cross-references are found the command will stop and warn you. You may then choose to blank the cell even though there is a reference to it. The formula that refers to the cell that is blanked will then assume that the cell now contains zero and line of question marks will appear in the blanked cell to show that another cell relies on its value.

C - Copy command
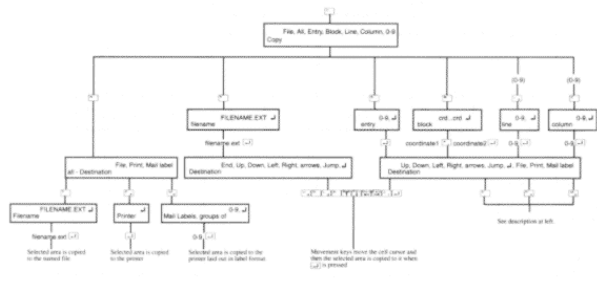
The COPY command is used to transfer copies of sections of the worksheet or entire files to other locations, other files or the printer. There are restrictions with just what can be transferred to where but even so this is a most useful command.

Copying within the worksheet

You have three options. Firstly you can make a single copy of an entry, line, or column to somewhere else in the sheet. Secondly you can copy several lines or columns in a single command to a new area. Thirdly you can copy one entry, line or column several times.

When you type C to start the copy command you can then type a number followed by C or L to say how many lines or columns you would like to copy.

If you are just copying a single line, column or entry, you may then type a number to say how many copies of it you would like.

Generally when making a copy you will be asked if you want to adjust the references. This means that if you are copying a line, all references to other locations on that line will be changed to the destination line. This preserves the sense of the calculations along

a line. The same applies to adjusting references down a column and, by extension, in a block.

If the formulae do not appear to be adjusted as you would wish you may be trying to copy something so that the copied items are all at an offset to some other item. In this instance follow the offset cell reference in the formula with a quote, '. This is often known as a "lagged" variable or a "relative reference". For example, a formula may be:

A1'+1

Copying to a printer

Copy also performs the role of a Print command and can be used to produce hard copy of specified data from the sheet, ranging from a single entry, lines, columns, blocks and the entire sheet. The printout of course reproduces the data as shown in the current display format and using current column widths etc.

Copying formulae to the printer

It is often useful to be able to print out a copy of the rules or formulae that underlie the worksheet display. To do this you must make the formula 'visible' by using the X -eXchange command. Next adjust the column widths, using N W, so that they can be seen fully and finally copy the area in which you are interested to the printer.

Areas larger than the displayed screen can, of course, be printed.

When printing out large worksheets you may often wish to fit as much information as possible on each sheet of paper. It is possible to switch to condensed print, if your printer is capable of it, by using the OUT command to send the appropriate codes. This is discussed in Tutorial X.

Preparing mail labels

This command will prepare printed mail labels. The addresses will usually be on a single line in the worksheet so you will have to insert markers where you require each new line to begin. This is done by inserting an extra column at the end of each address line which is given a format of Carriage Return. Remember to put in a carriage return column at the end of the address. You will get confused

results if you leave it out. You will need to adjust the column widths to correctly align the addresses onto the tops of the labels. A more detailed explanation is given in the tutorial guide.

Copying from files into the sheet

If you are already working on a sheet and then select the Copy File command to load in an existing sheet it may be put into any specified blank area of the worksheet or be placed as an addition at the end of the sheet. This command allows very flexible file merging.

The file to be read may either be in the format of a .MEM file or be a .DAT, .DIF or .TXT file. The program will check that the area to which the file is to be copied is vacant and is large enough to hold the data and if not you will just see the message Not enough space. Each file will be first read to determine its structure and then read for a second time to extract the data itself.

Each filename extension signifies to the Notebook spreadsheet that it should expect a certain type of information that is held in a certain form. The rules are as follows:

DIF format files are often used by graph plotting programs and many other spreadsheet programs can read and write data in the DIF format. However, only numerical values are saved/loaded in this way. The formulae used to create the results are not saved in this format of file. If you are trying to read DIF format files from another program you may need to rename the file to have the .DIF extension because the Notebook spreadsheet will not otherwise recognise it even though it has the correct format. Numbers will come in formatted to be General and text to be Text Left Justified.

To bring in data from a word processor or screen editor a file must have a **TXT** extension. The Notebook spreadsheet can read both normal ASCII text files and most Document files created by word processors. These are converted to normal text files as they are read in. (Document files differ from standard text in that they contain certain, normally invisible, control codes that are used by the word processor to include bold/italic etc. and to ensure that the right hand edge lines up etc.)

The data text should be laid out in tabular form. It is possible to bring in the numbers as values that the spreadsheet can use in

calculations. The requirements are that you set up the column widths so that each column in the worksheet coincides with one column on the incoming file. The way to visualise the operation is to picture the text coming in as being laid directly on the top of the worksheet as it currently is. Wherever it lands the spreadsheet will try to interpret it sensibly. Remember that the column divider will count as a space.

You must also set the default formats for the columns to be suitable for the incoming file. Be careful that stray items of text in a number column cannot be sensibly calculated and so will stop the operation (without harm). To get round this problem try starting with all the default formats set to Text Left and check that everything looks suitable. If necessary delete items that would cause a problem. Then copy the loaded file back out to another .TXT file. (of course if there are no problem cells then there is no need to create a second .TXT file, you can just reload the first one). Blank All and set the default formats to the final form using the [⬚][⬚] command and finally read in the second .TXT file. It's easier than it will seem on reading this.

Files with the extension of **DAT** are expected to contain just numbers. The files can be created with a word processor or other editor (including document files) or as the result of formatted output from a program written in BASIC or FORTRAN.

Files with the extension of **MEM** are intended purely for use from within the Notebook spreadsheet itself. They contain all the information for the program to be able to exactly reproduce the layout and display of the saved file. This is the normal type of file you use to save and load sheets on a day to day basis. You don't have to put the ".MEM" on the end of a name. It is added for you automatically if you don't use one of the other three extensions.

Copying to files

You can use the Copy command to copy the contents of the spreadsheet display to a file in memory. You can choose to save it as a .MEM file if it is to be reloaded into the spreadsheet. Alternatively, you can save to TXT, DAT or DIF files for use in other programs on different computers.

There are two types of Text file that can be read by other programs. The first with the extension .DAT will only copy numbers as they

appear on the screen to the file. The second with the extension .TXT will copy both numbers and also any text, titles and headings, as laid out on the screen. Both will copy any part of the worksheet. If you arrange your .DAT file correctly before saving you should be able to read it from a BASIC program if you want to do further processing.

Status lines and column dividers are not saved in such files.

To copy to a file for reading by other spreadsheets or graphics packages just give your file the extension .DIF for Data Interchange Format. Do your copying in the normal way and the program will make whatever adjustments necessary.

Cross reference checking

A section of the sheet cannot be deleted or blanked if there are other parts of the sheet which depend upon values that are about to be deleted. Similarly, if a section of memory is to be written to a .MEM file, it must not make references outside its own area. That section must be able to stand on its own so that filed sections to be merged in do not interfere with the running of the worksheet. In this way the dynamic error checking is maintained at all times. If you have trouble in erasing by line or column, go back to the Blank Entry command sequence and remove the entries one by one in the reverse order of their calculation numbers.

[E] - Edit command
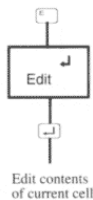
Entering the edit mode

You can edit the contents of any occupied cell. While editing, the automatic calculation feature will be switched off, so you may make errors in what you are typing without being warned. However, on completion of the editing a full recalculation will be carried out. You will then be warned of any errors, which is in direct contrast to what happens when you insert an expression for the first time. You may find it easier to overwrite expression entries rather than edit them, although no real harm can be done either way.

If an error is found when the recalculation is attempted the part that is in error will be displayed. At this point you may press [⬚][⬚] to re-edit the line. You can correct the line at this stage by using the [⬚] key to remove characters. Then make insertions that are necessary and finish off with a [⬚].

The Edit command is entered by typing [E] followed by [⬚] after which the contents of the current cursor cell will be displayed on the edit line and the cue will be located at the start of the line.

You can use the [⬚] and [⬚] keys to move along the line. The two delete keys can be used to delete characters to the left of, or under the cue. Pressing [⬚] will exit from the Edit mode and the amended line will be placed into the relevant worksheet cell. If [⬚] is pressed while editing, any changes are lost and the original cell contents are left unchanged.

[X] - Exchange command

The display is switched between showing the results of the calculations and the formulae that lead to those results
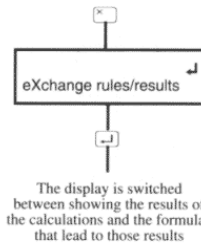
The eXchange command is entered by typing an [X] followed by a [⬚]. The effect of this command is to change the display in the cells from the numerical results to the rules or formulae from which they are calculated. It will usually be necessary to change the column width with the New Width command sequence. Go back to display of the answers by typing [X] once more.

Together with the formulae each expression cell is displayed with its order of calculation number in angle brackets followed by its formula. If [⬚] is pressed after the display has changed, the free memory figure will also change to show the amount of index table space that remains. When [X] is used to switch back to the normal display, the free memory figure will, once again, show how much space remains for cell data.

⌐F⌐ - Format command
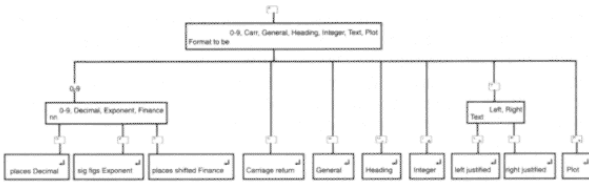
You use the ⌐F⌐ - Format command when you want to start typing data into a cell but first want to specify a special format for the cell rather than accept any of the default formats. Before you are asked to enter the data you will be asked what format you would like to give the data in this cell.

When entering data there are three types of format that could be applied. The first is the default column format (the one you gave when the column was first inserted). If you started the sheet using the ⌐Mo⌐-Make sheet option then all columns have a default column format of type General. If you start entering data by pressing ⌐.⌐ then the data you enter will be given this default column format.

The second format that may be used is the Global format. Any data you enter will be given this format if the entry is started by using either the (or the " command (' may be used in place of ").

Actually there are two global formats, one for text and one for expressions and numeric values. On start-up the text one is set to Text Left Justified and the value/expression one is set to General. You can change these formats to a variety of options using the New Global command.

The third format that a piece of data may have is that given by using the ⌐F⌐ - Format command.

The various formats that may be applied to a piece of data have no effect on the way an entry is stored in memory, only on the way it is displayed.

Once specified the format will stay with an entry even if moved or copied to a new position. This is because it is a piece of data that has a format and not a cell location.

Internally, the numbers are stored in a binary floating point format. It is not important to understand the details of the internal floating point number merely to know that it will ensure accurate calculations with up to 15 significant figures. Generally you would not want to display all the available accuracy so the program allows you to define just how many decimal places you want to show.

If you do not really know how many decimal places you can ask the spreadsheet for the vague format definition called General Format. It will display the number in a similar fashion to that of scientific pocket calculators. Thus if the number fits into the space available in its entirety then it will be shown in full. If it is too large or too small then it is converted into the Exponent form (see below). This format will make sure that whatever the result of a calculation the result will be displayable.

The maximum number of decimal places is 38. 15 significant figures is the most that can be entered or displayed but internal calculations are done to 16 or 17 significant figures.

Format types

The format options you have available when entering a value or expression into a cell can each be set to any one of a selection of Format Types detailed below. When specifying a cell format you can choose any of these options, the Global or Default Column formats can be assigned types using the New Global format command sequence.

Carr

CARRIAGE RETURN, this can only be used as the Default column format when you choose to insert a new column or change its format with the ND command. It should only be used with a blank column, usually of width 1 character. Its purpose is to put in intermediate end of line markers in mailing list and database applications. It has no other use. It is not possible to enter data into cells that have default Carriage return format.

#Dec

DECIMAL, the # defines the number of decimal places that will be displayed.

```
                                   =====A=====
format 5D, memory 123.456789, display   123.45679
format 2D, memory 123.456789, display      123.46
```

#Exp

EXPONENT, the # defines the number of significant figures to be displayed in the decimal part. The exponent form is sometimes called scientific notation. It consists of a decimal number followed by the power of 10 by which it must be multiplied to give the actual number. This power is called the Exponent. A letter "E" separates the number from the exponent.

```
                                 ======A======
format 5E, memory 123.4567, display   0.12346E+03
format 3E, memory 0.001234, display   0.123E-02
```

#Fin

FINANCE, this outputs all values in balance sheet format layout. The # represents the number of places shifted you want the results. This is the same as saying the number of times you want the number dividing by 10 before displaying it. With this facility you can display your answer in thousands (3F) or millions (6F). Commas are added every third place in the usual way of presenting financial output. Negative values are enclosed in brackets.

```
                                    =====A=====
format 0F, memory  12345.678, display  12,345.68
format 0F, memory -12345.678, display (12,345.68)
format 3F, memory  12345.678, display      12.3
```

Gen

GENERAL, this is a general purpose format that will display the value as simply as possible and as nearly in the way that it might be displayed on pencil and paper calculations. Unnecessary zeroes will be removed and if the number is too large or small it will convert to exponent format. The format bears some resemblance to that used on scientific calculators.

```
                                         ======A======
format G, memory 123.0,        display           123
format G, memory 0.123,        display         0.123
format G, 0.000000123456, display    0.123456E-06
```

Int

INTEGER, the value is displayed to the nearest whole number.

```
                                ====A====
format I, memory 567.89, display      568
```

TxtL

TEXT LEFT JUSTIFIED, is for alphanumeric entries. The memory contents will be displayed left justified in the cell.

```
                                   =====A=====
format TL, memory 'TEST', display  TEST
```

TxtR

TEXT RIGHT JUSTIFIED, is for alphanumeric entries. The memory contents will be displayed right justified in the cell. If the cell is smaller than the entry the display will be filled as if it was going to be left justified.

```
                                           =====A=====
format TR, memory 'TEST', display                 TEST
format TR, memory 'THIS IS A TEST', display THIS IS A T
```

Head

HEADING, is for alphanumeric (text) entries. The memory contents will be displayed in their entirety even if the column is not as wide as the entry. This format can therefore be used for titles when the column width is, at different times, liable to be varied. Adjacent entries will not be displayed if the heading over-rides them.

```
                             ===A=== ====B====
format H, memory 'THIS IS A TITLE',
                     display   THIS IS A TITLE
```

Plot

PLOT, will fill up a cell with asterisks to the nearest integer value of the stored number. This is used to provide a very simple histogram representation of data.

```
                             ====A====
format P, memory 5.556, display   ******
```

Overflow:

If there is insufficient space to display a numerical value then the cell will be filled with asterisks to avoid misleading you with incomplete figures. The same does not occur with text entries (c.f. also headings).

```
                             ===A===
format 4D, memory 1234.56789, display   *******
```

G - Get command
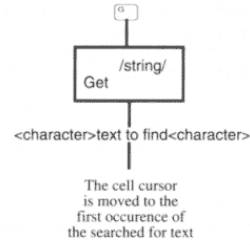
The Get command is used to search from the current cell location to the end of the worksheet for a particular string. The string is delineated on the edit line by a single printable character which must not be a letter or numeral. When this delineator or marker is next encountered on the line the string you are typing is assumed to be complete. This system is adopted to give you flexibility over which characters actually are included in the string.

The string length may be up to 30 characters not including the delineators. Only the text or expression as it appears on the contents line is searched. You cannot search for displayed values that appear as the result of manipulation by the current formats. The search proceeds across each column and then down a line and so on.

If repeated searching is to be used then after initially setting up the string it is merely necessary to follow the G command with the same two delineators as used before. The string that was previously used will be re-displayed and form the basis of the following search.

I₅ - Insert command

For column insertion you are now asked for the default column format - see the F command for more details

These options allow the destination to be selected - when ⏎ is pressed the new line(s) are inserted at the current cursor position or the bottom of the sheet if End was selected.
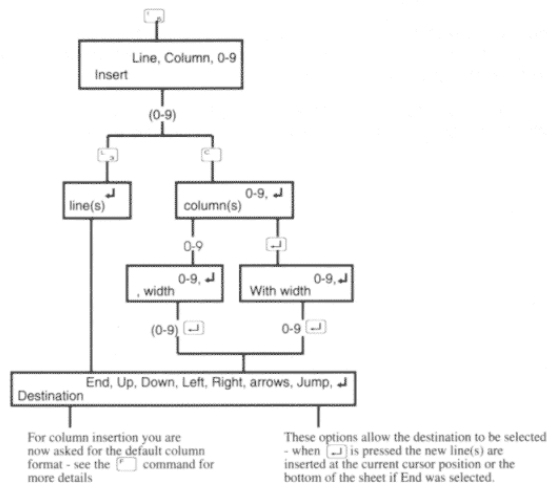
The Insert command is used to increase the size of the worksheet. The sheet does not initially exist but you can either press M₀ to make a blank sheet of fixed size and type. Or, if you choose to use the Insert command to create your sheet you must start by defining at least one column. Once a column is inserted you can insert lines. You can insert several columns or lines at a time.

You are asked for a destination so the spreadsheet knows where in the sheet you wish the new lines or columns to go. The first lines can only be inserted in front of the cursor (press ⏎) or after the cursor (press E for End). As the defined sheet builds in size the available options increase.

The options "Up, Down, Left, Right" and "arrows" in the destination prompt just mean that you can press U₄, D, L₃ or R or use the cursor keys to move the cursor. When you press ⏎ the line/column will be inserted above or to the left of the cursor. You can also use a jump destination to move the cursor to the position where you

would like the insertion to take place. See the Jump command description for more details.

If you are inserting one or more columns you will then be asked to specify a default format for the columns. See the description of the Format command for more details.

Increasing the size of the sheet does not use up any of your available memory, this only happens when data is entered.
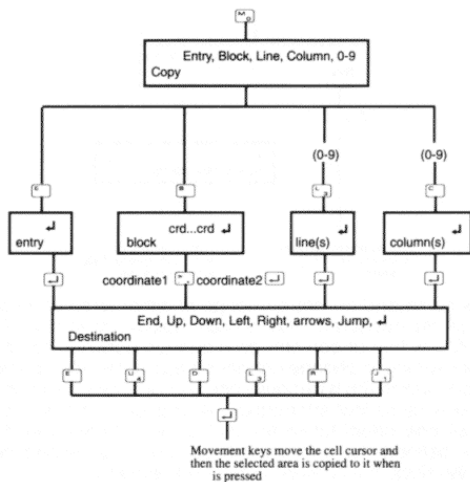
⌨ - Jump command
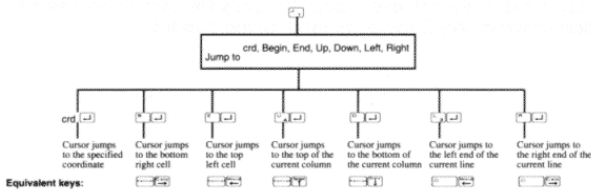
The Jump command is used to move the cursor directly to the cell you specify or edge of the worksheet. The crd prompt refers to the co-ordinate cell reference to which you want to go. The other prompts refer to the extremities of the worksheet. So ⌨ - Jump Right means take the cursor across all the columns to the right hand side of the worksheet. This command can be used both as a primary command and in answer to the request for destination.

⌨ will go to the top left of a sheet.

⌨ will go to the bottom right of a sheet.

⌨ will go to the top of the current column.

⌨ will go to the bottom of the current column.

⌨ will go to the start of the current line.

⌨ will go to the end of the current line.

Loading a file

The Notebook spreadsheet does not have a separate Load command. This is achieved using the Copy File command. See the description of the Copy command for more details.

⌨ - Move command

Movement keys move the cell cursor and then the selected area is copied to it when ⌨ is pressed

The Move command transfers a section of memory to a new destination. Unlike Copy, the original cell locations are blanked and all references to the transferred area, and within the block itself, will be automatically adjusted to the new location.

The same options are available as with the Copy command.

If several lines or columns are being moved an internal check is made that the destination specified will provide sufficient room.

⌨ - New command

Formats are now set using menus similar to the ⌨ command.

Column containing cell cursor is changed to display with the new width
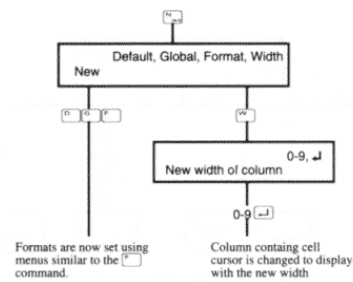
The New command defines a change of either width of a column, format of an occupied cell, default column format or Global format. A new format can only be a change between compatible types for example a change cannot be made between a Text and a Finance format. This is because one is character information and the other is numerical information. A change in default column format will only affect future entries that do not have their format explicitly defined. No change will be made to existing entries. Please also study the FORMAT command reference.

New Default format will change the default format for blank cells in the current column. (It over-rides the format that was set when the column was first inserted).

New Global changes the global format accessed when you start entry with " or (. If you set a text format this changes the global text format and if you set a numeric format this changes the global numeric format.

New Format is used to change the format of a piece of existing data. This cannot be used on a blank cell.

New Width will change the displayed width of a column.

⬚₆ - Out command

After selecting ⬚₆ the
characters you type until
you press ↵ will be
sent directly to the
printer
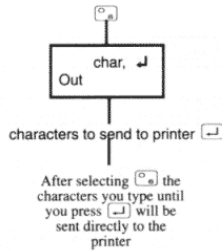
The OUT command is the way you output control commands to your printer. You can set up your printer for special features such as condensed print or different character font or a different colour if you have it available. The affects you can achieve on your printer are completely dependent on the features that it supports. The spreadsheet just send the codes that you type to the printer. It does not try to understand the effect they might have.

All characters including control codes are valid and so you cannot use the ⬚ key to backspace. If you make a mistake finish off that entry with a ↵ and start it again. You can abandon an Out command by pressing ⬚. To produce the "Esc" character that many printers use to access their specific features press ⬚+⬚. Finish off each entry with a ↵. Your characters will be echoed on the screen, control codes will be preceded by a ^. You will find the printer commands in your printer manual.

The printer command you will often want to send to your printer is the one to switch it to condensed print (17 characters per inch). If your printer is Epson compatible the code to do this is character code 15, (⬚+O). You would send this by typing ⬚ ⬚+⬚ ↵

If you use a laser printer and pages aren't fed out immediately after printing then this may be because it requires a "form feed" character to be sent to it. Type ⬚ followed by ⬚ (together) and finally ↵. On dot matrix printers you can type ⬚ then type ⬚ (together) to repeatedly feed one line through the printer for alignment. Press ↵ or ⬚ to finish.

417

⬚ₓ - Partition command

The screen may be partitioned either vertically or horizontally to give 1,2, or 4 available windows and the movements of the cursor locations within these windows may be defined as synchronised or unsynchronised for both the horizontal and vertical partition. Synchronising means that movements in one window will be matched by equivalent movements in the other. Without it the inactive window will be unchanging.

The horizontal partition location can be specified by taking a number from the grid that will be displayed on the screen. The vertical partition is similarly defined. Partitions may be removed by the PE partition end command sequence.

To jump the cursor from one window to another the slash commands are used. ⬚ will jump left to right and vice versa and the ⬚ will jump up and down between windows.

418

Printing a worksheet

The Notebook spreadsheet does not have a separate Print command. Printing is achieved using the Copy command to copy part or all of the sheet to a printer. See the description of the Copy command for more details.

A special destination "Mail label" can also be used to print address labels. See the tutorial for a worked example of this.

419

⬚ - Quit command

Pressing ⬚ at the spreadsheet's main menu is the same as selecting the ⬚ - Quit command.

The Quit command clears the screen and makes a return to the previous screen of the Notebook, where you can choose to start new, list stored or print a worksheet. A copy of your current worksheet is always saved in a file RESTART.MEM. However, you should get into the habit of using the Copy command to save your work to a file you name yourself because RESTART.MEM is overwritten next time you leave the spreadsheet.

When you next use the spreadsheet you can select the Restart command to load the existing RESTART.MEM file back in and continue on from where you were last working.

420

Saving a file

The Notebook spreadsheet does not have a separate Save command. This is achieved using the Copy All to File command. See the description of the Copy command for more details.
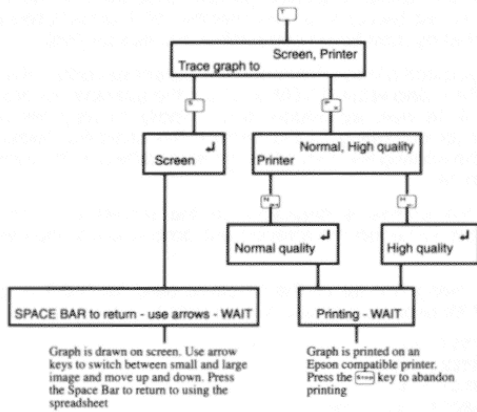
⌗ - Sort command
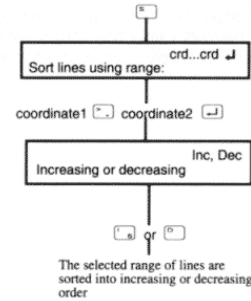
The SORT command allows sorting of lines using any specified part of any particular column as the list of items to be sorted. The sort may be by increasing or decreasing values and may be carried out on both numerical values and characters. You should avoid sorting where you have mixed formulae and constants in any column over the range or you may get very confusing results.

Note that mixing of text and numerical values in the sort should be avoided as this will give indeterminate results. Capital letters and lower case letters are taken to be of equal value. With text sorts numbers come after letters followed by blanks, punctuation marks and finally control codes. All punctuation is taken to be of equal value. Remember that any numbers that are entered as text characters can sort in unpredictable ways, for example "8" will sort as a higher value than "24".

If text or blank entries are within the specified numeric range, or vice versa, they will tend to sort to one or other end of the range.

⌗ - Trace graph command

Graph is drawn on screen. Use arrow keys to switch between small and large image and move up and down. Press the Space Bar to return to using the spreadsheet

Graph is printed on an Epson compatible printer. Press the [stop] key to abandon printing

The Graphics features of the Notebook spreadsheet work by using a set of special functions to point to cells containing the data to be plotted and the text for the labels.

Note that where crd is specified in these functions you must enter a reference to a cell co-ordinate. You can't use numerical or expression equivalents except where stated.

The Trace command requires a lot of memory to prepare the image to be output and may not be available if there is not much memory left in your Notebook. You could transfer some documents / spreadsheet files to disk or another computer and then delete them to make more space. Alternatively, you could use a PCMCIA memory card to make more space available.

When you use the Trace graph command you can have the output drawn on the screen or sent to an Epson compatible printer. When the output is drawn on screen it is initially drawn so that the whole chart fits on the screen. You can switch between this and a magnified view using the ⌨ and ⌨ keys. In the magnified view you can use the ⌨ and ⌨ keys to scroll up and down.

When you output to a printer you get a choice of Normal or High Density. The former is quicker but not as detailed as the second option. (These two options use either the ESC K or ESC L sequences supported by most Epson compatible graphics printers).

When you start a Trace command the current spreadsheet is saved in the file called RESTART.MEM and then the spreadsheet draws the image in its own workspace before finally showing this on the display (or outputting to the printer). This operation takes some time. The rotating line shows that the spreadsheet is still working on the picture.

Once the picture is displayed on the screen you can press Spacebar to reload the spreadsheet data and continue working on it.

The following is a list of the functions used for building graphs. Tutorial XV explains their use in more detail.

TYPEPLOT(crd)
MAINTITLE(crd)
SUBTITLE(crd)
TIMELABEL(crd...crd)
XLABEL(crd...crd), or **XLABEL**(crd, crd, crd, etc.)
XMAXIMUM(crd)
XMINIMUM(crd)
XTITLE(crd)
XVALUE(n, crd...crd)
YMAXIMUM(crd)
YMINIMUM(crd)
YTITLE(crd)
YVALUE(n, crd...crd)

Remember that the above are functions with arguments and as such must be entered into cells that have a numerical format even if they refer to some textual data such as a label. The cell that contains the label itself must of course be set to a text format.

If any errors are encountered while using the graphics an error message will be given and a return will be made to the spreadsheet.

Limits of the graphics commands

max length	MAINTITLE	24 chars
	SUBTITLE	35
	YTITLE	25
	XTITLE	25
	TIMELABEL	16
	XLABEL	24
number of	TIMELABELs	32
	XLABELs	9 (no. of lines) possible on one chart
	YVALUEs	9
	XVALUEs	9

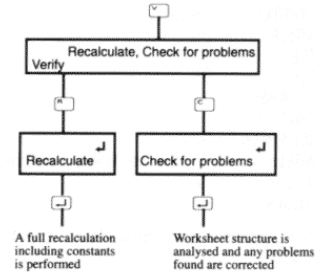If you use lengths longer than specified above they will be truncated or ignored.

NOTE

If you do any EDITING of the plotting instructions or make any changes that do not force a recalculation, you may end up with an error message or values that does not seem to be true. This is most likely if you are using direct values rather than references to cell co-ordinates.

The surest way to handle plotting functions that do not contain cell references is to overwrite them rather than edit them.

The instructions for plotting are worked out whenever a recalculation is done so if no recalculation has occurred the latest instructions may not have been properly determined. If you get such an error message just use the ! force recalculation command and try the plot again.

ⱽ - Verify command

A full recalculation including constants is performed

Worksheet structure is analysed and any problems found are corrected

The VERIFY command is used to cause a full recalculation including the constants within your worksheet. That is in contrast to the ! command which only recalculates expressions. Normally force a recalculation with the ! command. The Verify command is only normally used when you have brought some data into the spreadsheet from a .TXT, .DIF or .DAT file.

If you ask for "Verify Check for problems" then the structure of the worksheet will be analysed to look for inconsistencies and if any are found an attempt will be made to fix them. This is a safety precaution to ensure as much as possible that you are never left with an unworkable spreadsheet.

ʷ - Which files command

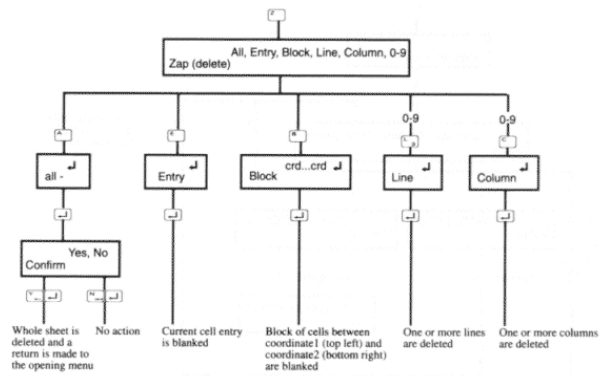Demonstration files are added to the list of available files
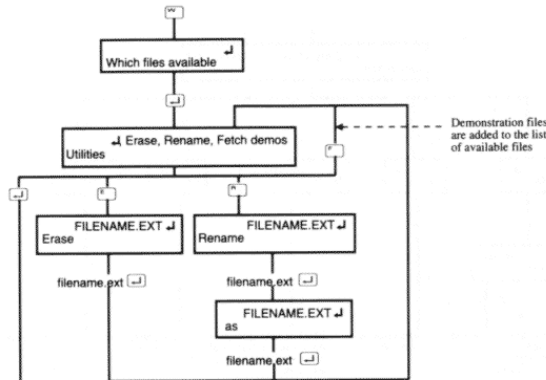
The Which files command is used to look at files associated with the Notebook spreadsheet and also to give the basic file handling operations you may want. They include erasing files, renaming files and fetching the built-in demonstration files. This allows the demos to be accessed without returning to the opening screen of the spreadsheet.

The files in the Notebooks memory will be read and all the compatible files, i.e. those that end in .MEM, .DAT, .DIF and .TXT will be displayed on the screen. After this you will be able to do any of the file management operations you want to.

If memory becomes full you may be able to use the Erase function to remove some unwanted files from memory to make extra space.

ᶻ - Zap (delete) command

Whole sheet is deleted and a return is made to the opening menu

No action

Current cell entry is blanked

Block of cells between coordinate1 (top left) and coordinate2 (bottom right) are blanked

One or more lines are deleted

One or more columns are deleted

Zap deletes the specified entries from memory. If the command refers to an entry or a block then the command acts just like the Blank command. For references to Line, Column, or All then the size of the worksheet is actually reduced by the command. Deleted entries CANNOT be retrieved so if in doubt first copy the worksheet to a file before embarking on complex rearrangements.

Before the deleting is carried out the program will check whether any of the items to be deleted are referred to elsewhere in the worksheet. If cross-references are found the command will stop and ask if you really want to blank the referenced items. If the cells in question remain on the sheet they will be filled with a row of '?' to signal that data is missing.

If you use the Zap All command and confirm that you want to delete everything by typing Y you will be returned to the opening menu of the spreadsheet as if you were starting from scratch.

! - Force recalculation command

The ! force recalculation command has two main purposes. The first is to cause a calculation when the Automatic calculation is off. It is a minimum recalculation and therefore takes less time than the Verify and recalculate command.

The second usage is for solving iterative problems such as simultaneous equations or other circular reference problems. In these types of problems there may be two cells which refer to each other. Initially they cannot both be valid but on repeated recalculation they will approach true answers.

* - Macro command

The * Macro command is initiated by typing an asterisk, *, followed by a reference to the co-ordinate of the cell in which the Macro is stored.

The purpose of this command is to call in a predefined set of commands. Just type the appropriate key presses for your desired commands into any text format cell. Where you would want to put a ⏎ use the @ symbol. Where you want to move the cursor use U, D, L or R to move it Up, Down, Left or Right.

It is valid to finish a sequence with another Macro command or even a reference to the same macro. With a self-repeating reference you can create macros that go on repeating until an error is found.

This command is ideal for repetitive changes of format down a whole column for example. The macro NFG@D*1@ placed in cell A1 will change an occupied cell's format to general and then move the cursor down. The *1@ at the end means that it is called again and so the cell underneath it is changed and so on. This will stop when an attempt is made to change an unoccupied or text cell or if the cursor is moved outside the worksheet.

The Macro can be used to create a loop of direct commands, which differs from a loop of functions such as you may use in an expression.

More detail on spreadsheet error messages

A-Z, or 0-9
Only a letter or numeral may be used at that point.

Ambiguity (see manual)
The order of calculation number can not be resolved without ambiguity. The expression must be preceded by a reference to the cell that created the SET( ) to which this cell refers. The solution to this error is shown in Tutorial XVI - Table filling.

Argument too large
If calculated the EXP( ) of this value would be greater than 1E+38. EXP will only accept a value up to about 88.

Argument > 1.0
ASIN, ACOS, ASINR, ACOSR can only have arguments between 0 and 1.

Cannot multiple move there
There is not sufficient space at your specified destination to move this group of lines or columns.

Column first
Please insert a first column before you try to enter lines.

Comma only
You seem to have confused the layout of this function's argument.

Def function badly set up
The layout of the defined function is wrong.

Division by zero
You are trying to divide by zero which would give infinity. The Notebook spreadsheet for all its power cannot handle that. This message may appear if you have blanked some cell entries.

DO( ) needed with WHILE( )
You cannot use the WHILE( ) function on its own. Part of the job it does is to seek out the DO( ) on the same line.

E+38 maximum
The maximum exponent that is available in the floating point numbers used in the Notebook spreadsheet is +38.

Entry too long
The maximum entry width is the free area in the middle of the entry line. This has a fixed limit of 67 characters. The error message appears if you try to type beyond this limit. The absolute maximum a cell can hold is 127 characters when loaded from a file.

ERROR called from <crd>
This is an error you yourself have called by using the ERROR( ) function.

FILENAME.EXT not found
The file named cannot be found in memory. Use the ⓦ for Which files command to find which files are available. They are also listed when you use the Copy File command.

Function (((((5))))) max
You may only nest functions and brackets to 5 levels.

Logical expression only
The IF( ) function must resolve to a value of TRUE, FALSE, -1 or 0.

Missing "("
You have too many right hand brackets.

Missing ")"
You have too many left hand brackets.

Move cursor
The command you are trying to do does not make sense unless you actually move the cursor.

Must be a constant
You cannot overwrite a formula with a SET( ) or INIT( ) function.

Must be a letter
A column letter is expected.

Must be "logical" (-1 or 0)
The required argument at this point must resolve to a logical value.

Must be one of +-*/ <>=%[]|), or ⏎
One of these operators are required.

Must not refer to this cell
The cell that contains the DO( ) function itself is in the range of its own argument. This is a circular call that cannot be resolved.

Negative argument
Cannot have a negative argument to a logarithm or square root.

No function "xxxxx"
This function you have entered is not one of the built in functions.

No such column
You have made a reference to a column outside the worksheet area currently defined.

No such line
You have referred to a line outside the worksheet area.

Not enough memory
There is not enough memory left to read in this file.

Not enough space
Part of your destination area would be outside the worksheet as currently defined.

Not found in this range
The LOOKUP( ) or INTERP( ) first argument was not present or enclosed by any two values in the list that you have specified.

Not in marked zone
You are trying to partition outside the area marked by a grid on the screen.

Number or value missing
With LOOKUP( ), CHOOSE( ) and INTERP( ) none of the items in the list can be blank.

Odd number of values required
Simpson's rule works on an odd number of values only.

Operator at end
You cannot end an expression with an operator.

Out of memory
The memory available for entering data is all gone. Your work to date will not be damaged. Try to split your work and write an independent portion to a file so that you can later consolidate the results.

outside worksheet
You have moved the cell cursor past the edge of the worklsheet.

Reference outside worksheet
This cell reference is outside the area of the worksheet as you have currently defined it.

Result more than 1E+38
You have done a mathematical operation that gives too large a result. This is probably because of an error in one of your constants.

Second "."
You can only have one decimal point in a number.

Separate these operators
You have two operators together. In addition, if you have two brackets together, i.e.)(then you must separate them with a *, thus)*(.

Should be one of +-*/ <>=%[]|), or ↵
These are the valid operators at this point.

Stack overflow, shorten entry
There are too many pending operations. Cure this by splitting the expression into two or more cells.

Text reference in an expression
In this expression you have made a cell reference to a text entry.

Text/Value change
You are trying to change a formula or constant into a text entry or vice versa. If you really want to do this, use [B][F] to blank the cell first.

The range must increase
Only ranges that extend over at least two columns or lines are really ranges. You see this error if the start and end coordinate are the same.

Too long
The String used by the Get command can only be 30 characters long at maximum.

Too many chars
Your filename has more than 8 characters.

Use partition first
You have used one of the slash commands before the partition has been made on which these commands work.

Worksheet too small.
You will have to expand the worksheet area before you can copy any data in from a file.

Wrong conditional layout
You have confused the layout of this conditional entry.

Wrong range layout
The way you have specified a range is not correct.

Zero argument
You cannot have a logarithm of 0.

0...9 or "."
A numeral is expected at this point (the start of an exponent)

15 sig figs max
The Notebook spreadsheet calculates to 16 or 17 significant figures but for security always rounds down to 15. You can only give 15 significant figures when entering numbers.

255 lines max
The maximum number of lines in a worksheet is 255.

38 places max
The floating point number system of the Notebook spreadsheet only goes down to E-38.

52 columns max
The maximum number of columns is 52, they are marked A-Z then a-z.

.TXT, .DIF, .DAT or .MEM
Only these file extensions are valid if you are trying to load or save a file.

"end" wrong here
It does not make sense to have "end" as the destination here.

"(" or "," first
A range must always be preceded by a bracket or comma. It cannot be used as part of an expression.

"#" cannot follow a range
You cannot get the current value of a range.

"x" is wrong after a co-ordinate
This character is not a correct following operator.

"x" is wrong here
This character is not one of the options you were offered in the prompt list.

"x" should be a numeral
Only a numeral would be valid here.

<B22> is used in <C3>
You have attempted to remove or overwrite an expression that is referred to in another expression. If you really don't mind losing the contents of the cell type Y. Blank entries will be assumed to contain zero for calculation purposes.

%% 1 Entry destroyed in calculation
%% 2 Column details corrupted
%% 3 Internal coded format error
%% 4 Entry not found to format
%% 5 Entry not found for calc number
%% 6 Index table corrupted
%% 7 Main storage corrupted
%% 8 Cannot delete index table item
%% 9 Negative number of index entries
%% 10 Zero length entry
%% 11 Does not exist
%% 12 Command corrupted
%% 13 Memory corrupted, saved, check copy carefully
%% 14 Opening space for index table error
%% 15 Invalid binary number format
%% 16 Number formatting error
%% 17 Number formatting error
%% 18 Number stack corrupted
%% 19 Operator decoding error
%% 20 System error
%% 21 Look-ahead buffer corrupted
%% 22 Disk write record error
%% 23 System error

These are system error flags, that signal that a serious error has occurred in the working of the program rather than in the entry that you have typed. All being well you should never see one, however, they are the long stops in the event of system bugs. If you get one

please note the exact circumstances of how it occurred and inform your supplier. As a general rule your work will not be damaged even if you find such a bug. The Notebook spreadsheet will automatically check that your work has not been damaged. If it has then it will be filed automatically. Just press the [⎋] key in the usual way and you will be able to carry on.