A Preliminary Reference Guide

Version 2.1J April 1996

AICS

A Preliminary Reference Guide

Version 2.1J April 1996

TO ENTER TEXT, AN EQUATION OR A QUESTION ON THE COMMAND LINE:

Simply begin typing,

Entry takes effect when the RETURN key is struck.

To delete an entry to left of cursor:

Backspace to correct input, retype

To cancel an entry:

Backspace to beginning of entry, retype

QUITTING A PROCESS OR CALCULATION

All subprocesses can be quit by typing // in place of a normal entry.

To interrupt an on-going process or calculation,

press CNTL-Y.

CNTL-Y will bring to an orderly halt:

serial database searches
spreadsheet recalculations
printing operations
database form displays

However, the time necessary for the CNTL-Y to take effect may range from immediacy to some minutes. The current process is generally completed. If the process is a database operation during recalculation, the current cell's operation is completed (serial searches are interrupted immediately).

The BREAK key is disabled while you are in QueryCalc. BREAK is re-enabled when you exit QueryCalc or use the /SYSTEM and /RUN commands to momentarily exit to use MPE.

To exit QueryCalc entirely, type:

/EXIT

ENTRIES

Values:

Range from 1E-77 to 1E+77

(All cell values are double precision real, 16.5 decimal places)

Labels and formulas:

Can contain up to 187 characters

Text and formulas:

Anything that begins with @, $, ', ^ or "

(Formulas are calculated from the most nested entry outward, with the exception of the IF statement phrases which are calculated left to right)

Cell references:

C12 - cell C12 on the current page, cell address will be incremented when replicated

DC34 - cell C34 on page D, cell address will be incremented when replicated

{C12} - absolute cell address C12, cell address will not be changed when replicated

STARTING AN ENTRY

To start a command line:

/

To start text entry:

" for right justified text

^ for centered text

' for left justified text

To start a text equation:

$

To start a database query question:

@

To start a numeric equation:

0-9, A-Z, +, -

CURSOR MOVEMENT

Basic cursor movement:

[f1]: Moves cursor to top of screen, then scrolls screens upward

[f2]: Moves cursor to bottom of screen, then scrolls screens downward

[f3]: Moves to previous page and homes cursor

[f4]: Moves to next page and homes cursor

[f5]: Moves cursor up 1 row

[f6]: Moves cursor down 1 row

[f7]: Moves cursor left 1 column

[f8]: Moves cursor right 1 column

Jump to a specified location:

/J A1: Moves cursor to location A1 (same page)**

/J MD12: Moves cursor to location D12 (on page M)**

/J G: Moves cursor to cell A1 on page G**

**Should the cursor land in a zero width column, the cursor moves to first adjacent non-zero column to the right on the line specified.

USING THE CELL EDITOR

To edit the contents of the current cell:

/E

To edit the last entry (accepted or not):

/REDO

To edit the last command:

/LAST

Once in the EDITOR:

Use the SPACE key to move underneath the beginning of text to be modified

Use the RETURN key to move down to the line section requiring modification

Once properly positioned, these initial letters have these effects:

D - begins deletion for the length of d's typed in
R - begins replace for the length of line typed in
I - begins insertion for the length of line typed in
A - begins appending text to end of original text
U - commands editor to leave edit mode and use text
DDI - deletes text for length of d's and then begins insertion
// - commands editor to leave edit mode with no effect
? - shows on-line help function

If the input does not start with one of these initial letters, then the replace function is invoked by default.

MATH AND LOGICAL OPERATORS

Math Operators

+ addition * multiplication
- subtraction / division
^ exponentiation

Math Functions

LOG( )
LN( )
SIN( )
COS( )
TAN( )
SINH( )
COSH( )
TANH( )
ABS( )
EXP( )
ASIN( )
ACOS( )
ATAN( )
ASINH( )
ACOSH( )
ATANH( )
INT( )
FRC( )
N!( )
PI( )
SQR( )
RND( )
SGN( )

List functions

NUM(A12:D34)	number of equation cells in rectangle A12 to D34
SUM(AC13:CD30)	sum of equation cells in cube AC13 to CD30
AVG(B12:B15,F1:G10)	average of all equation cells in specified areas
MAX(A1:Z90)	maximum of all equation cells in rect. A1 to Z90
MIN(G10:G30)	minimum of all equation cells in column G10 to G30
VAR(F3:G20)	variance of all equation cells in rect F3 to G20
DEV(H10:H15,Z15)	standard deviation of column H10 to H15 and Z15

Alternate Spreadsheet Extract Functions

EXTR("REPORT1",bb10) retrieve numeric value of cell BB10 in currently non-active spreadsheet REPORT1. If cell BB10 is a text cell, zero is set in. If cell BB10 does not exist, an error is declared.

EXTR$("RZTG",cd12) retrieve text value from cell CD12 in currently non-active spreadsheet RZTG. If cell CD12 is a numeric cell, the text value of the result is set in. If cell CD12 does not exist, an error is declared.

EXTR(a10+".PUB",gd33) Text equations may specify the non-active spreadsheet name. Cell A10 is presumed to be a cell containing text with the name of the spreadsheet to be accessed.

Logical IF functions (**IF** and **IF$**)

IF(Q2>3,10,15) If conditional phrase true, then 10, else 15

IF$(Q2>3,"not bad","not good") If conditional phrase true, then enter text "not bad", else enter text "not good". Because this IF statement returns text, a '$' is required at the beginning of the line.

IF(AC7=7,0,IF(AC7=6,1,IF(AC7=5,3,10)))
If cell AC7 is 7, enter 0
If cell AC7 is 6, enter 1
If cell AC7 is 5, enter 3
otherwise, enter 10

IF($B13>$"0" and SUM(G13:G30)<100,100,SUM(G13:G30))
If text cell B13 greater than "0"
and the numeric sum of G13 to G30
less than 100, then enter 100,
otherwise enter sum of G13 to G30

Rules for IF Statement Usage:

A generalized IF statement is of the form:

IF(equation.relop.equation, true phrase, false phrase)

where "relop" is any standard relational operator (<, =, etc.)

Rule 1: the equation segments may be either text or numeric, but they cannot be both. Text cannot be compared to a numeric equation and vice versa.

Rule 2: the true and false phrases may be either text or numeric, but they cannot be both. The true and false phrase types (text or numeric) must match the IF/IF$ statement type.

Rule 3: text equations are perfectly permissible in a numeric IF conditional phrase. Likewise, numeric conditional phrases are permissible in a text IF statement. The two forms of equations may be mixed in an AND or OR form so long as they compare properly within each equation segment (see last example above).

Rule 4: IF statements are calculated left to right rather than from the most nested condition outwards. Thus, chaining of IF statements is possible (see second to last example).

PICK(A12,34,56,123,34,234,56)

Depending upon the value of A12, the A12th argument is chosen. The value of the key is to fall in the range 1 to [number of choices]. The key and each of the arguments may be any numeric equation. The integer of the key value is used.

VLOOK(A12,B10:D15,2)

A value is chosen (horizontally) from the 3rd column of the table B10 to D15 depending upon the value of A12. The leftmost table values (B10 to B15) are searched for the largest value equal to or smaller than the value A12.

A12 is the key value and may be any numeric equation.

The table B10:D15 must be a complete numerically-valued table (no holes).

The value 2 is the row offset and must be an integer only.

HLOOK(A12,B10:D15,2)

A value is chosen (vertically) from the 3rd row of the table B10 to D15 depending upon the value of A12. The topmost table values (B10 to D10) are searched for the largest value equal to or smaller than the value A12.

A12 is the key value and may be any numeric equation.
The table B10:D15 must be a complete numerically-valued table (no holes).

The value 2 is the row offset and must be an integer only.

SPREADSHEET EDITING COMMANDS

Clear current cursor location's entry:

```
/CLR
```

Clear certain range on specified pages:

```
/CLR  B17:G20     clears cells B17 to G20 on current page
/CLR  A:C,M,Z     clears pages A to C, M and Z
/CLR  AB1:EC10    clears the cube B1 to C10 on pages A to E
```

Clear entire report:

```
/CLR  ALL
```

Delete column at cursor location and *n-1* columns to the right:

```
/DCOL  n
```

Delete row at cursor location and *n-1* rows downward:

```
/DROW  n
```

Insert *n* columns to the left of cursor:

```
/ICOL  n
```

Insert *n* rows above cursor:

```
/IROW  n
```

Insert *n* pages previous to the current page:

```
/IPAG  n
```

Delete the current page and *n-1* following pages:

```
/DPAG  n
```

Search the whole or partial spreadsheet for specified text and its replacement:

/S&R QueryCalc will ask for the "search for" text, the "replace" text, and the range of cells to be searched. Only equations (both numeric and query) can be modified, not text.

or

```
/S&R  "search text"  "replace text"  ae9:bg15
```

SPREADSHEET FORMATTING

Change column widths on a page:

/CWID 3	change current cursor's column width to 3 wide
/CWID B 0	change column b to 0 characters wide
	(cell widths are variable from 0 to 36 chars)
/CWID A:D,M,N 6	change columns A to D, M and N to 6 wide

To see the column widths defined for a page:

```
/COLS
```

To change the format for range of cells:

```
/FORM
```

Either text or numeric cells may be formatted. Text cells may only be right, left or centered justified. Numeric cells may be selected to have dollar signs, commas, the number of displayable figures after the decimal place, the form of negative signs and the choice of how zeroes will be displayed. Numbers may be displayed as normal decimals, in percentages, or in scientific or engineering exponential notation. Integer numbers are decimals with 0 numbers after the decimal point.

Important Note: When a cell range is formatted using the /FORM command, every cell in that range, if it does not already exist, is "created". QueryCalc does not generally create a cell until something is entered into it. Thus, formatting a large range of otherwise empty cells can have a substantial negative impact on performance. Empty formatted cells will take up non-useful room in the computer and may significantly slow processing times. The most common cell format used throughout the spreadsheet should be set using the /GFORM command below. Reserve the use of the /FORM command only for those special cell ranges you wish to have formatted differently from the global format.

To change the global format:

```
/GFORM
```

Global format commands are identical to individual cell formats discussed in /FORM above.

To change the cursor, background and border colors:

```
/COLOR
```

The cursor, the borders, and background may be selected to be displayed in gray, white or black.

RECALCULATING THE SPREADSHEET

Recalculate entire report without database inquiry:

```
!
```

Recalculate entire report with database inquiry:

```
!!
```

Partial recalculate:

!e9,f10:f30 recalculate math only, local page cells e9, f10 through f30

!A:C,K,N:P recalculate math only, pages A through C, K, N through P

!!M:Q,Z recalculate query questions and math, pages M thru Q,Z

!!e9:d15,B recalculates queries and math, local page cells e9 to d15 and all of page B

Calculate a single cell at the time of entry or edit:

3*5280/2 (a numeric equation with no cell references automatically recalculates on entry)

4*(5/A12)! (a math equation with cell references requires an explanation point to cause immediate recalculation. Without an explanation point, the entry is syntax checked and four hyphens are placed in the cell indicating acceptance)

@val of cost when productno is 45!
(a query question requires an explanation point to cause immediate recalculation. Without the explanation point, the entry is syntax checked and an "<ok>" is placed in the cell indicating acceptance)

To change the order of the recalculation (by rows or columns) for specific page(s):

/RECALC the basic command. Questions follow.

/RECALC ALL C sets all pages to column-oriented recalculation

/RECALC B:G,M R sets pages B through G and M to row-oriented recalculation mode

CELL REPLICATION

Replicate a single cell to current cursor location:

/REP A14 replicates cell A14 to current cursor location

Replicate a rectangle of cells to current cursor location:

/REP A14:B20 replicates the rectangle of cells A14 to B20 to current cursor location (current cursor location is the upper left hand corner of new rectangle)

Replicate rectangle of cells to place other than cursor location:

/REP A14:B20>C40 replicates rectangle to C40 (C40 is the upper left hand corner of the new rectangle)

Replicate rectangular pattern over again until list is exhausted:

/REP A14:B20>C40:M40 replicates rectangle repeatedly along the 40th row from C40 to M40 until the 'to' list is exhausted

Replicate one page to another:

/REP A>M replicates page A to page M

All cell references within the replicated equations are adjusted during the replication by the amount of the displacement (other than those cell references which are absolute). Cell references that refer only to the current page (B12 rather than GB12) are not adjusted during a page-to-page replication.

/REP AA1:AZ90>MA1:MZ90 replicates range of cells page A to M

The example above also replicates the entire page, but it is slower and does not copy over column widths.

SPREADSHEET MERGE

/COPY The copy command will allow you to import one or more pages from another QueryCalc spreadsheet and merge it (them) with the current spreadsheet.

The copy command will ask for the range of pages from the external QueryCalc spreadsheet to be imported.

Column widths from the external spreadsheet will also be copied in for the appropriate pages.

The specified pages will be overwritten; that is, if page F is specified to be copied, page F will be be copied from the external file to the current page F.The pages where the new pages "land" are expected to be blank or disposable. Generally, page insertion operations would precede the copy command.

LOADING AND SAVING FILES

File names:

Up to 8 characters each for filename, groupname, accountname

filename[.groupname[.accountname]]

QueryCalc files can only be stored in the user's current group
but can inputted from any other account to which he has access.

To load a filename:

/LOAD filename[.groupname[.accountname]]

To save a filename:

/SAVE filename

To see a listing of the directory of QueryCalc files:

/CAT *or* **/DIR**

To delete a QueryCalc file:

/KILL filename

The /KILL command, in contrast to an MPE file purge, immediately updates the QueryCalc directory. A filename purged by MPE will be retained in the QueryCalc directory although the file itself is no longer present. The directory will however clear itself of all "ghost" names during the next /SAVE command.

To display current file information:

/INFO

SETTING MARGINS

To set page margins, headers, and footers:

/MARG A screen will appear indicating the current magin settings (top of page margin, bottom of page margin, left margin and right margin), number of lines per page, automatic or manual paper feed and any headers or footers that have been defined.

To select a function to be modifed:

[f1]:	Modify page margin settings
[f2]:	Modify headers and footers
[f8]:	Exit routine

Setting Margins

To modify a margin setting, simply type over the value currently indicated. To tab from one value to another, press the RETURN key. You may exit at any time from margin setting modification by pressing the [f8] key.

Defining Headers and Footers

To add or modify a header or footer, simply type over the presently defined header or footer. To move the cursor through the header/footer lists, press the RETURN key. To clear a currently defined header or footer, press the [f1] key. You may exit at any time from the header/footer definition routine by pressing the [f8] key.

A header or footer is handled as a Text Equation and thus it is defined exactly as a text equation would be defined for a spreadsheet cell. All text functions may be used within a header or footer, as well as cell references off of the spreadsheet itself. A fully qualified cell reference (e.g., MD33) will retrieve the text value from the specified cell. A partially qualified cell reference (e.g., G12) will retrieve the text value from the specified cell on the page the header or footer currently applies to.

Text equations in the headers and footers must be preceeded by either an R,C,L to indicate whether the header or footer is to be right, center or left justified.

Examples:

C$"Brentwood Construction, Ltd" (centered justified simple text)

R$dat$3(sysdate) (right justified date function)

L$g33 (left justfied header/footer, text comes from cell G33 on current page)

PRINTING A REPORT

Printing a report:

Printing occurs from upper left visible corner to right margin marker and bottom margin marker as indicated on screen. Only the visible columns will be printed to either the system printer, the local terminal (and its printer, if any), or an MPE edit file.

The print routine asks which device you would like to have the output go to as a normal part of its procedure.

Print the specified pages:

/PRINT	the basic command
/PRINT ALL S	print all of the pages which exist to the system printer
/PRINT A:B,M,R:Z L	print pages A to B, M, and R to Z to local printer (your terminal)

Print the equations for the report:

/PRINTEQ	the basic command
/PRINTEQ ALL D MYFILE	print all of the existing equations to an MPE disc file named "MYFILE"
/PRINTEQ A:D,L,P:Q S	print the equations for pages A to D, L, and P to Q to the system printer

To change the system printer:

/PRINTER	the basic command. A list of available printers will be displayed. Press the RETURN key to move the selection bar down. Press the BACKSPACE key to move it up. Press the "=" key to make your selection.
/PRINTER ACCOUNTING	selects the printer named "ACCOUNTING." You need type in only a sufficient number of characters to uniquely define the printer.

PRINTER SETUP AND ENHANCEMENTS

Elaborate text enhancements (such as underlining, bold, superscripting) are possible when printing a report. There are no standard codes among the commonly available printers to perform these functions. Every printer is slightly different. To accomodate virtually every commercially-available printer, QueryCalc uses a 2-character enhancement sequence which is translated into the control sequence appropriate to the printer you are using.

The example on the following page was printed on an HP LaserJet using Helvetica, Courier, bold print, bold print alone, and underlining.

The QueryCalc codes to turn these features on and off are detailed in the pages following this example. The codes which you have selected to enhance a particular cell are transmitted at the beginning of the cell in which they appear. The standard QueryCalc sequences will be converted into the appropriate escape (control) sequences based on the printer you have selected. Cell enhancement is similar to cell formatting, which is accomplished with the /FORM command.

To enhance a cell or cell range:

/ENHANCE the basic command

/ENHANCE B1,X0 A1:A10

add the enhancements bold on (B1) and expanded print off (X0) to the cells in cell range A1 to A10.

Note: most printers retain the last enhancement request until it is changed or the printer is reset. Thus an enhancement should only be transmitted once (placed in the first necessary cell) rather than be repeated in each following cell. Use the cell ranges sparingly. You may unnecessarily overrun the maximum line length of 255 characters by constantly resending an enhancement request for each following cell in the row.

Note: blank cells (cells with no equations in them) cannot be enhanced. Should you wish to set an enhancement on or off where no cell currently exists, create a "dummy" cell. *Suggestion:* $" ", a text equation printing a single space.

/ENHANCE NONE AB2:CG5

clears all enhancement requests in the cell cube AB2 to CG5.

SCHEDULE OF INDIRECT COSTS
For the Year Ending Today
August 9, 1989

Operating Expenses:		
Salaries - officers	111,600.00	
Salaries - supervisor	968.50	
Payroll taxes	10,808.56	
Supplies and tools	4,916.87	
Gas and oil	11,945.84	
Truck maintenance	11,434.47	
Truck rental	600.00	
Truck insurance	5,298.15	
Equipment maintenance	2,846.40	
Equipment insurance	1,985.37	
Telephone	9,538.14	
Taxes and licenses	1,821.69	
Truck depreciation	21,546.65	
Equipment depreciation	15,417.14	
		210,727.78
Occupancy Expenses:		
Rent	16,550.00	
Utilities	2,541.46	
Cleaning service	1,542.96	
Property insurance	577.75	
Building maintenance	678.28	
Improvements depreciation	389.73	
		22,280.18
Operating Expenses:		
Salaries - officers	223,200.00	
Salaries - office	58,343.29	
Payroll taxes	14,755.16	
Employ benefits -- health/accident	24,077.74	
Employee benefits -- other	1,270.97	
Computer expenses	14,473.87	
Office supplies	10,058.83	
Bidding expenses	3,971.22	
Travel and entertainment	6,788.21	
Advertising and promotion	15,096.54	
Insurance	2,224.47	
Dues and subscriptions	6,284.30	
Legal and accounting	7,717.39	
Contributions	4,559.18	
Taxes and licenses	308.86	
Miscellaneous administrative	967.91	
Life insurance	1,727.50	
Office equipment depreciation	4,719.46	
Total Administrative Expenses		400,544.90

Total Indirect Expenses

633,552.85

PRINTER SETUP AND ENHANCEMENTS

To first define a specific printer, or to choose one that has already been defined, type:

/SETUP

This command will draw a screen showing you the printers which are presently defined on your system. When QueryCalc is shipped, these printers are predefined. They are:

0. The Null Printer (a psuedo printer in which all enhancements are suppressed)
1. HP PaintJet
2. HP LaserJet Plus (including the LaserJet II or 2000)
3. HP DeskJet
4. Reserved for future use
5. HP 2601 (Diablo 630) (user may purge and reenter new device)
6. HP QuietJet (user may purge and reenter new device)
7. HP ThinkJet (user may purge and reenter new device)
8. Undefined (user may define)
9. Undefined (user may define)

Ten printer types may be defined in total. The names for printers 0 through 4 are assigned by AICS and are the printers on which graphics may be printed. Printers 5 through 9 are user-assignable. When graphics are attempted to be printed on these latter devices, a blank area of 28 lines will result. You may redefine the escape sequences for any printer type (1 through 9). Only the Null Printer may not be redefined.

To choose a printer from the list to use, simply type:

The printer's number. That number will then begin to blink. Then type either ' // ' or press the [f8] key to exit.

To define or modify a printer's definition, type:

The printer's number to choose it. Then type an [M]. The chosen printer will be highlighted. You may either redefine the name, or press RETURN to continue onwards. Typing ' // ' or pressing the [f8] key terminates the definition at every point in the sequence.

[cont'd next page]

PRINTER SETUP (cont'd)

Once the printer name has been defined or left as is, a second screen will be drawn. The cursor (a highlighted area) will appear in the upper, leftmost box. Any predefined control sequence that is resident in the box will be invisible until you move the cursor to the next box. The cursor moves each time the RETURN key is pressed. A null entry (nothing entered) leaves the present entry in the box alone. Any entry of one character or more (other than BACKSPACES) replaces the contents of the box.

> You may press the appropriate function key to advance the cursor to the first box in each of the four modes shown in the second screen. You may also get to any one box by simply continuing to press the RETURN key.

To define a control character sequence suitable for your printer, three characters are given special definitions. They are:

' means *escape*

^ means *control*

\ means *define the following three-digit decimal number as a singly-transmitted byte*

Examples are:

'P is equal to the sequence [ESC]P.
'&dD is equal to the sequence [ESC]&dD.

^R is equal to CNTL-R (ASCII code value 18_{10})
^B is equal to CNTL-B (ASCII code value 2_{10})

\001 is equal to the value 1_{10} (= ASCII CNTL-A)
\253 is equal to the value 253_{10}

Using these prefixes, any ASCII sequence can be entered. Although no standards exist, all commercially available printers use either control characters or escape sequences to change such items as font, pitch, or print enhancements.

[cont'd next page]

PRINTER SETUP (cont'd)

To use the defined sequences in a QueryCalc report, one or more of the standard QueryCalc enhancement codes is used in the /ENHANCE command. These codes are 2-character sequences of the following form:

Sequence		
On	Off	QueryCalc meaning
p1	p0	Pica print pitch
e1	e0	Elite print pitch
c1	c0	Compressed printing
x1	x0	Expanded printing
l1	l0	Near-letter quality printing (NLQ)
b1	b0	Bold printing on
u1	u0	Underlining on
d1	d0	Double-strike printing
i1	i0	Italicized printing
v1	v0	Subscripting
^1	^0	Superscripting
m1		User mode sequence No. 1
m2		User mode sequence No. 2
m3		User mode sequence No. 3
m4		User mode sequence No. 4
m5		User mode sequence No. 5
s1		Character font set No. 1
s2		Character font set No. 2
s3		Character font set No. 3
s4		Character font set No. 4

The 2-character sequence that appears in the cell's formatting information text will be converted to the appropriate code sequence for the currently selected printer. Thus, the same report can be printed to several different printers from different manufacturers and it will work and look (approximately) the same.

Notes about printers:

> Not every printer responds in the same manner. In some, the enhancement does not take effect until the first printable (non-space) character is encountered. In others, the enhancement begins immediately on definition. In some, the enhancement request is terminated at the end of a line. In others, it remains in effect until you terminate it. You will simply have to experiment with your particular printer to determine its mode of operation. For HP printers, an enhancement generally remains in effect until it is changed or the printer is reset.

SORTING THE SPREADSHEET

Sort a rectangle of cells on a page by columns:

```
/SORTC
```

QueryCalc asks for more information using this short form.

or

```
/SORTC a13:n18 b d e d f a
```

Sort rectangle a13 to n18, first by column b, descending, then by column e, descending, then finally by column f, ascending. Three-level sorts are possible. Only the first key column is necessary.

Sort a rectangle of cells on a page by rows:

```
/SORTR
```

QueryCalc asks for more information using this short form.

or

```
/SORTR a13:n18 14 d 15 d 18 a
```

Sort rectangle a13 to n18, first by row 14, descending, then by row 15, descending, then finally by row 18, ascending. Three-level sorts are possible. Only the first key row is necessary.

Notes on usage: For both of the command forms above, it is important to remember to also include in the sort the label rows or columns for the sort rectangle. If they are not included, the labels will become meaningless to the matrix of cells that were just sorted. This is particularly disadvantageous if the label cells are referenced by query questions inside the matrix of cells just sorted. Either text or numeric cells may be sorted, but mixing the two in one key column or row will produce meaningless results.

Sort a series of contiguous pages by specific cells:

```
/SORTP
```

QueryCalc asks for more information using this short form.

or

```
/SORTP b:f a13 a b15 d f15 d
```

Sort pages B through F, first by cell a13, ascending, then by cell b15, descending, and then by cell f15, descending. Three-level sorts are possible. Only the first key cell is necessary. Either text or numbers may be sorted.

QUERY QUESTIONS

To open a database:

@OPENDB dbname [/password]

where "dbname" is the name of the database to be opened. A password will be asked to allow you entrance into the database. The database will be opened in a READ ONLY mode (IMAGE mode 5 or 6, dependent upon compatibility considerations; KSAM or MPE file open mode is read only, shared access, dynamic locking allowed).

A database remains open until you close it, load a new report, or perform a /CLR ALL.

The open databases and passwords necessary to reopen them are stored when the report is /SAVEd. Thus, a database only needs to be opened once during the lifetime of a report.

To close a database:

@CLOSEDB dbname

To examine the currently open databases:

@SHOWDB

All of the open databases, their local QueryCalc names, their database types, and their addresses will be shown. The default database will be marked with an arrow ("<--<<").

The *default database* is the one which will be presumed to be the database of interest, should you not specify one in your query question or command.

To default to another database:

@USING dbname

[cont'd next page]

QUERY QUESTIONS

General form of a query question:

```
@sum of contractamount*workcomplete/jobcost  when
   /                     /
(operation)        (dataitem equation)

amount-balance > 10000 or jobnum(5,8) ib  8000,8500
              /    /    /          /     /       /
       (d.i.e) relop   /     (d.i.e) relop       /
                      /                          /
  ("pattern" to be matched) ("patterns" to be matched)
```

Rules for a DATAITEM EQUATION:

1. Dataitem equations can use: +, -, *, /, ^

2. All dataitems used in the equation must come from the same dataset.

3. Parentheses are not allowed in a dataitem question.

IMPORTANT NOTE: Backslashes can be used to distinguish a dataitem name when mathematical operators are part of its normal name (such as +,-,etc) from a dataitem equation.

Example: \ age-operated-on \ - \ current-age \

Allowable operations for the above form:

SUM of ...
AVG of ...
MAX of ...
MIN of ...
DEV of ...
VAR of ...
VAL of ...

Each operation determines the statistic requested and records the record numbers of the records satisfying the qualifying statements.

[cont'd next page]

The "pattern" to be matched can be written either as one of the three following constructions:

1. *state is CA* — The pattern to be matched is CA. No quotes are required if there are no spaces in the pattern (see next example).

2. *employee is "Smith, Joe"* — Quotes are required to separate out the pattern from the sentence if non-alphanumeric symbols (anything other 0-9,a-z) must be used. If the IMAGE dataitem to be matched is a text dataitem of the "U" type, the pattern will be upshifted to match the database entries.

3. *productno is [ab12]* — Brackets indicate that the pattern to be matched will come off of the spreadsheet (in this case, from cell B12 on page A).

   or

   name is [$g34] — A spreadsheet text cell must be referenced with a leading "$" when the dataitem type is textual.

4. *acctcode(5) is P* — A dataitem which contains subitems is referenced as shown. If no subitem is specified, the first subitem is assumed.

5. *productnum(5,8) is [$CG7]* — A substring of a text dataitem (either (X or U type) can be matched as well as found, as shown. The indices indicate that the string to be matched begins with the fifth character and ends with the eight.

6. *@val of acctcode(9,3,5) when productnum(5,8) is 3415* — A substring of a subitem may be found or matched as shown. The substring of characters residing in positions 3,4,5 of the ninth subitem of *acctcode* will be the value returned when the indicated substring of *productnum* equals 3415.

7. *name is Hol@*
name is @ght
name is @erin@ — A "wild card" (@) may be used in text string matches. In the first example, any text value beginning with "Hol" will qualify the record. In the second, any text ending with "ght" will qualify. And in the third example, the string of characters "erin" anywhere in the text will qualify the record.

[cont'd next page]

QUERY QUESTIONS (cont'd)

Another query question form:

```
@num when category is 501
@find when category is 501
@pct when date ib 850101,851231 and jobnum>8000
```

These three query forms return only a single value: the number of records qualifying, but all three forms record the qualifying record numbers so that the qualifying records may be REREAD. None of these queries establish SUM, VAR, AVG statistics as there is nothing to take the sum or average of in these query questions.

The NUM and FIND query question forms are identical on a standard spreadsheet page, but are slightly differently used on a detail list /REPORT page. The FIND query question is used to define a new search group on a /REPORT page, while the NUM returns only the number of records qualifying. The PCT query question form returns the percentage of records qualifying of those records actually searched.

Once a group of records has been found, they may be repetitively REREAD:

```
@rereading, max of age-operated-on/100
```

The current list of records found by the last normal query question can be reread to find another dataitem equation's values. The current list of qualifying records is left undisturbed. The current statistics are modified to reflect the results obtained in this query.

```
@rereading, max of age-operated-on when zip ib 99001,99120
```

The current record list found by the last normal query can also be subqualified (given that all of the dataitems in the query question exist in the set used in the last normal query).

```
@level 2 rereading, find when state is AZ,CA,NM,NV,ID
```

Nine levels of subqualification exist in the REREADING query questions. Each level rereads the records found in the query level specified and creates a new sublist of of qualifying record numbers one level down. LEVEL 1 REREADING is equivalent to REREADING.

[cont'd next page]

QUERY QUESTIONS (cont'd)

If a query question calculates a sum, it actually calculates all of the statistics. These additional values may be obtained by USING CURRENT STATISTICS in the following manner:

```
@ucs max
     min
     avg
     sum
     dev
     var
     pct
     num
```

Interpretation: using the current statistics from the last query question, provide the additionally requested statistic for the question just asked.

These query questions calculate immediately. All of the information necessary for their calculation was stored during the initial query, thus their use has virtually no computational cost.

The intended use of this form is to provide a complete range of statistics about a set of records found during a query without having to re-search the entire set of records for each statistical value. [Example: obtain the average value using the normal query form and then find the max, min and standard deviation values with the UCS form.]

The current statistics are recalculated (lost) with each new, normal query question.

[cont'd next page]

QUERY QUESTIONS (cont'd)

The form to create a list:

```
@store in m socsecnum when state is VT and acctcode(5,8) is GY53
```

Creates the list M of dataitems SOCSECNUM for those records which meet the qualifying conditions.

The form to use a list:

```
@sum of amount-balance when socsecnum=!m and/or ...
```

The search dataitem SOCSECNUM is replaced by the values stored in list M. SOCSECNUM must be a search dataitem in the dataset which is being searched. Additional qualifying conditions may be added using and/or connectors. If the OR connector is used, the list phrase 'socsecnum=!m' must be repeated for each OR level.

The form to further qualify a list:

```
@store in m socsecnum when socsecnum=!m and dob ib 1940,1950
```

The list M is redefined by using itself as a search item. The new list M can be no larger than it was originally. It may be significantly smaller with the new qualifying strictures. This form of query list generation is necessary when all of the qualifying conditions cannot be obtained from a single dataset or database.

The form to create a hierarchy of sublists:

```
@store in s socsecnum when socsecnum=!m and state is AZ and/or ...
```

The list S is a proper subset of the list M of qualifying search items (that is, every item in the list S will be found in the list M).

[cont'd on next page]

QUERY QUESTIONS (cont'd)

The standard query question, as outlined on the second page of this section, creates a list of qualifying records as well as computing all of the statistics (max, min, var, avg, etc.), regardless of which statistic is requested.

Not all of the forms of all of the questions do this. The following is a table of what's modified and what's not when a query question is asked.

Form of Query Question Used	New List Created?	Statistics Modified?
@Sum	yes	yes
@Avg	yes	yes
@Var	yes	yes
@Dev	yes	yes
@Val	*only 1st qualifying record retained*	reset
@Max	*only max record retained*	yes
@Min	*only min record retained*	yes
@Num	yes	reset
@Pct	yes	reset
@Ucs, *xxx**	no	no
@Rereading,	no	yes
@Store in ...	yes	reset

*where *xxx* represents any verb (sum, avg, etc.)

[cont'd on next page]

QUERY QUESTIONS (cont'd)

To determine a database's structure:

```
@form [database name]
```

Once in the FORM mode, the function keys take on these values:

[f1]: display the master and detail set names

[f2]: display the entire database structure

[f3]: display the dataitems used in the database

[f4]: display the chained path structures of the database

[f5]: display help

[f6]: print all further displays to the system printer

[f7]: print all further displays to the terminal

[f8]: return to spreadsheet

Should a dataitem or dataset name be unknown, simply type in a best guess. FORM will display all similar names in the current database.

To see a specific dataset or a specific dataitem's attributes, type in the full name of the dataset or dataitem.

[cont'd on next page]

QUERY QUESTIONS (cont'd)

To show the contents of the current records found:

```
@show
```

A simple show command will show all of the dataitems for all of the qualifying records found during the last database query.

```
@show  productno,city,state
```

A subset of the dataitems for all of the qualifying records found during the last database inquiry can also be specified to be listed.

To sample the contents of any single dataset:

```
@show  invoices
```

All of the dataitems, record by record, will be shown for the dataset invoices. The database used is the currently open database.

```
@show  invoices:  amount,date
```

Only the dataitems amount and date in the dataset invoices will be shown, record by record, for the currently open database.

To show the contents of a user-created search list:

```
@show !a
```

The values held in search list A will be displayed.

QUERY FUNCTIONS

Elaborate query functions may be user-defined to gather information from multiple databases and/or datasets. Sixty such user-defined functions may exist in one QueryCalc report. Each function may be passed 0 to 6 numeric or text equations as parameters.

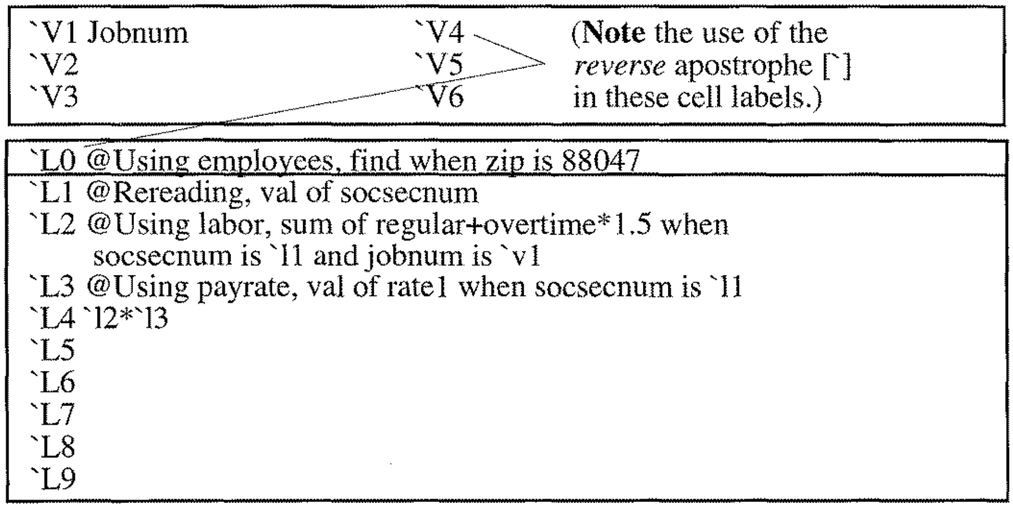
To define a query function:

/FUNCT	Enters the user-defined query function (UDQF) editor.
/FUNCT functname	Enters the UDQF editor and loads the desired function for further editing.

Purpose of user-defined query functions: Standard query questions in individual cells can gather information from only one dataset in one database at a time. This prohibits calculating a sum where data from one dataset must be multiplied by a value in found in another, item by item. The UDQFs allow this and much more complex mathematical manipulations of data which may be distributed over many datasets and databases, local and on remote HP3000s.

Example: You would like to obtain the sum of all of the wages paid to people who live in a particular area and who have worked on a particular job.

Solution (as it would appear on the UDQF editor screen):



```
`V1 Jobnum                  `V4
`V2                         `V5
`V3                         `V6

`L0 @Using employees, find when zip is 88047
`L1 @Rereading, val of socsecnum
`L2 @Using labor, sum of regular+overtime*1.5 when
       socsecnum is `l1 and jobnum is `v1
`L3 @Using payrate, val of rate1 when socsecnum is `l1
`L4 `l2*`l3
`L5
`L6
`L7
`L8
`L9
```

(**Note** the use of the *reverse* apostrophe [`] in these cell labels.)

Each line `L1-`L9 returns a result, marked as either text and numeric. The numeric value of line `L1 can be used in a succeeding query question or numeric equation by specifying `L1 (as in lines `L2 and `L3). The text value can be used in a text equation by specifying "`L1" (not shown in this example).

QUERY FUNCTIONS (cont'd)

Using the UDQF Editor:

[f1]: Moves cursor to top equation list
[f2]: Moves cursor to top of parameter list
[f3]:
[f4]:
[f5]: Moves cursor up one line (either equation or parameters)
[f6]: Moves cursor down one line (either equation or parameters)
[f7]:
[f8]: Exits editor (returns to spreadsheet)

Commands which work within the UDQF Editor very much as they do in the spreadsheet proper:

/E	Edits contents of cursor-indicated line
/REDO	Edits last line typed (whether accepted or not)
/CAT	Displays catalog of UDQFunctions defined for this report
/CLR	Clears the present line
/CLR ALL	Clears the entire function
/IROW n	Inserts *n* rows before the current line
/DROW n	Deletes current row and *n-1* following rows
/LOAD	Loads previously defined function into editor
/SAVE	Saves currently defined function
/KILL	Purges a previously saved function
/STEP	Steps through function, item-by-item, to insure accuracy of defined function.
@OPENDB	Opens a new database
@CLOSEDB	Closes a previously opened database
@FORM	Displays the form of database

QUERY FUNCTIONS(cont'd)

How the UDQF works:

Recall the example:

Problem: You would like to obtain the sum of all of the wages paid to people who live in a particular area and who have worked on a particular job.

```
`V1 Jobnum               `V4
`V2                      `V5
`V3                      `V6
```

```
`L0 @Using employees, find when zip is 88047
`L1 @Rereading, val of socsecnum
`L2 @Using labor, sum of regular+overtime*1.5 when
      socsecnum is `l1 and jobnum is `v1
`L3 @Using payrate, val of rate1 when socsecnum is `l1
`L4 `l2*`l3
`L5
`L6
`L7
`L8
`L9
```

Line `L0 is a FIND query question. Line `L0 is used to establish the list of all of the primary reference items. The "FIND when.." form is the exact equivalent of the "NUM when ..." query form. FINDs and NUMs can be used interchangeably both here and on the spreadsheet.

Other possible find statements for Line `L0:

`L0: find when socsecnum=!a
`L0: find when jobnum is `v1 and date ib `v2,`v3

Lines `L1 to `L9 may be query questions, numeric equations or text equations. The last line specified determines the answer returned. The returned value may be numeric or text, depending on the text or numeric nature of the final line.

Each line assumes the value returned by its equation, text or numeric or query. The line's value is stored as ASCII text. A line's value may be used in succeeding line equations. Before each line is executed, it is scanned for the [`] character, indicating that UDQF variables exist within the line. The line variable is then deleted and its value (as text) is reinserted into the line in its place.

Example: Before: `L5: JD("`l4")
After: `L5: JD("8/8/92") if `L4 = 8/8/92

QUERY FUNCTIONS(cont'd)

How to save your UDQF function:

/SAVE functname

The function name may be up to 16 characters in length and may be composed of any printing character.

The UDQF becomes a function that is unique to this QueryCalc report.

How to test your UDQF function:

/STEP

executes your /SAVEd function, line-by-line, visibly displaying the returned values so that you may be sure what you are calculating is what you want. If the final line is numeric, the current statistics are updated and displayed.

The step function halts at the end of one pass through the function. Execution may be resumed by pressing RETURN. Execution may be cancelled by pressing the [f8] key or typing "//" at the halt point.

CNTL-Y can also be used to interrupt execution.

How to use your UDQF function:

The general form is similar to other query questions:

@Using function, sum of functname(parm1,parm2,...,parm6)

where "sum of..." may be replaced by:

avg of...
max of...
min of...
var of...
dev of...
num of...
pct of... (always 100% for UDQFs)

Examples:

@Using function, sum of labor-burden(a15)

The function labor-burden is passed one numeric parameter from cell a15.

@Using function, avg of days-stayed($"PSH",1987)

The function days-stayed is passed two parameters, the first textual, the second numeric. Neither refers back to a spreadsheet cell.

TEXT EQUATIONS

A text equation is of the general form:

```
$" Last name: "+NAM$(A4)+" ........................."
```

The text equation functions are:

RJS$(B13,22) right justifies text found in B13 within a 22 character field width. Especially useful in query questions where text dataitems have been right justified.

CAP$(A12) capitalizes first letter only of the string found in A12

DNS$(B13+G14) downshifts all of text found in the concatenated cells

UPS$(D22) upshifts all of the text found in D22

NAM$(A32) capitalizes first letter of each new word in A32. Also capitalizes first letter following common name prefixes such as 'Mc','Mac','O",etc.)

SUB$(g15,6,9) extracts substring of text found in cell G15, beginning at character position 6, ending at position 9.

POS(FD34,"GRO") locates starting position of text string (GRO) in text found in cell FD34

DAT$1(text,offset) the date function will display the date in any one of seven different formats. The text portion of the command is expected to be an interpretable date, of some common (American) form like the following:

19450820	August 20, 1945
8/20/45	AUG-20/45
450820	20 Aug 1945

or

SYSDATE for the system's date.

The *offset* is a numeric equation or a simple numeric entry.

The output modes are:

DAT$1(.) =	880115
DAT$2(.) =	19880115
DAT$3(.) =	January 15, 1988
DAT$4(.) =	JAN 15 88
DAT$5(.) =	15-JAN-88
DAT$6(.) =	1/15/88
DAT$7(.) =	15.1.88

TEXT EQUATIONS (cont'd)

DAY$1(text,offset) returns day of the week when given a date and an offset. The text and numeric equations are identical to the date functions as used in **DAT$**.

Output modes are:

DAY$1(.) = Monday
DAY$2(.) = MON

HRS$(format) provides current time on either a 12 or 24 clock. Format must be equal to 12 or 24. Format may be a numeric equation.

Output modes are:

HRS$(12) = 3:13 PM
HRS$(24) = 15:13

MON$1(monthnum) returns month name in English when given a monthnumber. Monthnumber may be a numeric equation.

Output modes are:

MON$1(8) = August
MON$2(8) = AUG

A text equation example:

```
$"Current time: "+hrs$(12)+", "+dat$3(d6,a14)
```

The result would look like this:

Current time: 11:05 PM, February 14, 1989

where the date comes from cell D6 and the offset is found in cell A14.

NUMERIC DATE FUNCTIONS

JD(date)

date — Month/day/year written in any normal fashion. SYSDATE is also an acceptable entry. Julian date calculated is the day number beginning December 31, 1899. *Text equation.*

Examples:

JD("1/1/87")	Julian date of Jan 1, 1987
JD(a14)	Julian date of text in cell A14
JD(sysdate)	Julian date for today

MONTH(date)

date — Month/day/year written in any normal fashion. Simply the month in English is also acceptable. SYSDATE is also an acceptable entry. Month number is returned. *Text equation.*

Examples:

MONTH("Oct 13, 1989)	10 is returned
MONTH("July")	7 is returned
MONTH(sysdate)	month number for today

YEAR(date)

date — Month/day/year written in any normal fashion. SYSDATE is also an acceptable entry. Year number is returned in YYYY fashion. *Text equation.*

Examples:

YEAR("Jul-4-1776")	1776 is returned
YEAR(b34+c34)	year number of date in text format from cells B34 & C34 is returned.
YEAR("sysdate")	year number for today

ON-SCREEN & PRINTED GRAPHICS

Any one of six standard business graphs may be plotted on the terminal's screen, an HP PaintJet or HP LaserJet (Plus, II or 2000):

Standard Bar	*Clustered Bar*	*Point Plot*
Stacked Bar	*High/Low/Close*	*100% Chart (Pie)*

To define a graph, type:

/GRAPH

One QueryCalc page is required for each graph defined. The minimal information required to be filled in is the x-axis max and min and one data point. Labels are not required.

The graph template brought up by /GRAPH is a standard spreadsheet; that means that any cell requiring an input can be either a normal text label cell, a numeric cell, a text equation cell, or a query question.

Three questions on the graph template pertain only to graphs printed on an HP PaintJet printer: color, shading and transparency. If they are unanswered, default values will be chosen for the color graphs. The questions are ignored for on-screen and HP LaserJet (Plus, II or 2000) or DeskJet graphs.

Color choices:	white, red, yellow, orange, blue, turquoise, purple,green (only the first two letters need be entered in the cell)
Background shading:	yes or no (y or n)
Transparency mode:	yes or no (y or n)

A note on using the PainJtet with transparency film: The HP PaintJet will produce quite nice color graphs on transparency film. Transparency film does not dry immediately like PaintJet paper. This leads to a problem. QueryCalc would normally print the graphics portion first and then back up over the graph to print the appropriate text. But backing up will smear transparency film. To prevent this from happening, text must be printed at the same time as graphics on transparency film. This is QueryCalc's transparency mode.

Please note that QueryCalc's transparency mode does not put the PaintJet printer into the PaintJet's transparency mode. For the PaintJet, the transparency mode determines whether one or two inking passes per line will be used in drawing the graphs. To print on film, both transparency modes should be set. The easiest way to set the PaintJet into the HP transparency mode is to turn the printer off, press the SET TOF button, continue to hold it down while turning the printer on again and release the TOF button a second later. To reset the printer, simply turn it off and on again.

ON-SCREEN GRAPHICS (cont'd)

To see the defined graphs on your terminal, type:

/VIEW the basic command. A question asking for the page range will follow .

/VIEW A:D,M displays the graphs on pages A to D and M. Pages which do not contain graph definitions are ignored. " // " will exit the sequential graph displays.

To insert a defined graph into a spreadsheet, go to appropriate spreadsheet page and type:

/ATTACH the basic command. Questions asking which graph is to be attached and to which row will follow.

/ATTACH G Attaches the graph found on page G to the current cursor row on the current page.

/ATTACH G 5 Attaches graph found on page G to row 5 of the current page.

An attached graph will appear as 28 rows of text marked by ~'s. These rows are now blocked. No other use may be made of the cells forming these 28 rows. The graph from the specified (attached) page will be automatically inserted into this space when the page is printed to one of the printers which supports QueryCalc graphics.

To detach a graph from a page, type:

/DETACH the basic command. A question asking which graph is to be detached from the spreadsheet text will be asked.

/DETACH G Detaches graph G from the current page.

MISCELLANEOUS FUNCTIONS

/CALEN

A calendar displaying the current month, the previous month and the next month is obtained by this command. A carriage return returns you to your previous position in QueryCalc.

/CALEN 8/66

Any month within the range January, 1583 to December, 9999 may be displayed by either typing the month/year once you are in the calendar, or by specifying the alternate command as shown.

Years less than 100 are assumed to be in this century. The calendar implemented is the world-wide standard Gregorian calendar. The first full year of its implementation was 1583, hence the lower limit.

Function keys operate to increment or decrement the calendar by months or quarters.

/80

Sets terminal display width to 80 characters columns. *This command works only with the newer HP terminals (HP 700/9x and later).* The default mode for QueryCalc is 80 column display.

/132

Sets terminal display width to 132 character columns. *This command works only with the newer HP terminals t(HP 700/9x and later).* If this command is executed on an older terminal, the screen display will be fouled. To restore your terminal's display, simply type **/80**.

CASHFLOW(debt,term,apr,payment,buyout,*period,prepayments*)

Whichever of first the five variables is left blank is the value returned.

debt — may be variously considered to be debt, initial investment, present value, or equipment lease value. *Numeric equation.*

term — term in months. *Numeric equation.*

APR — annual percentage rate. *Numeric equation.*

payment — periodic payment or cashflow. *Numeric equation.*

buyout — may be considered to be equipment lease buyout, future value or principal remaining. *Numeric equation.*

period *(optional)* — period between payments in months. Assumed to be one per month if left blank. *Numeric equation.*

prepayments *(optional)* — number of prepayments. Primarily used in calculating equipment lease calculations. Assumed to be zero if left blank. *Numeric equation.*

Example 1: Loan amortization

What would be the interest rate be on a $50,000 loan over 5 years if the payment were $950 per month?

```
CASHFLOW(50000,5*12,,950,0)
```

The 3rd value, APR, is undefined, so it is calculated and returned. The buyout is zero, which means the loan is fully amortized in 5 years (5*12 months). The returned value: 5.28% APR.

CASHFLOW(debt,term,apr,payment,buyout,*period,prepayments*)

Example 2: Equipment leasing

What would the quarterly payments be on an equipment lease which required two prepayments, the original price was $25,000 with a down payment of 10%, a buy-out of $1500 and an annual percentage rate of 7.5% over 10 years?

CASHFLOW(25000*.9,120,7.5,,1500,3,2)

The 4th value, payments, is undefined, so it is calculated and returned. The actual amount leased is the unpaid balance, 90% of $25,000. The term is 120 months (10 years), the APR is 7.5%, the buyout is $1500, the period is one payment every 3 months (quarterly), and the number of prepayments is 2. The returned value: $751.14 per quarter.

Example 3: Regular interval cashflow

You own a building currently valued at $100,000. The property is depreciating at 8% per year. How much will it be worth is 3 years?

CASHFLOW(100000,36,-8,0,,12)

The 5th value, future value or principal remaining, is undefined so it is calculated and returned. No cash flows during the 3 years, so the cashflow (payments) value is 0. For purposes of correctly compounding interest on an 8% decrease in value, the period (in months) is 12 (1 year). The returned value: $77,868.80

CASHFLOW(debt,term,apr,payment,buyout,*period,prepayments*)

Example 4: Regular interval cashflow

You are a banker holding papers to a amortized loan of $36,000 over 15 years at 11%. You are 42 months into the loan and are now willing to sell the loan to another financial institution at 85% the value of remaining principal. How much money did you make (if any) and what was the actual APR you received, adjusted for the loss in principal you will take?

The solution must be in steps. Let's presume you start in cell A1 and go down the column. First we must deduce the monthly loan payment:

a1: CASHFLOW(36000,15*12,11,,0)

The value returned to cell A1 is the monthly payment for the original loan. The monthly payment is: $409.17.

Now, in cell A2, to calculate the remaining principal 42 months into the loan, we enter

a2: CASHFLOW(36000,42,11,A1,,)

We've used a cell reference here (perfectly legitimate in a numeric equation) to get the monthly payment. The principal remaining is the unknown. The value returned for principal remaining is: $31,966.06.

To calculate the total amount of money paid in to date, we must simply multiply the monthly rate times the number of months.

a3: A1*42

The total paid in is: $17,185.35.

Your income will be the interest generated (total payments less decrease in principal) less the loss you will take on discounting the principal remaining. That value is:

a4: A3-(36000-A2)-A2(1-.85)

Thus your income over the 42 months is: $8,356.50. And the APR you actually realized is:

a5: CASHFLOW(36000,42,,A1,A2*.85)

Remember, A1 contains the original monthly loan payment, A2 the principal remaining. The value A2*.85 is buyout you actually see. The APR calculated reflects this additional loss. The acutal APR is: 7.49%.

FINANCIAL FUNCTIONS(cont'd)

NPV(initial-investment,irr-desired,cashflows)

IRR(initial-investment,cashflows)

initial investment — The initial amount put into an investment. A positive number. *Numeric equation.*

irr desired — The internal rate of return desired per calculation period (may be thought of as weeks, months, years). To get APR, multiply the IRR rate by the number of periods in a year. *Numeric equation.*

cashflows — A table of values specifying cashflows. Several table ranges may be specified for this parameter if they are separated by commas. The specified ranges are read in order left to right, step and repeat until all rows in all ranges are read. Blank entries inside the rectangle after the last entry do not modify the result. Further money flowing from you to the investment is specified as negative numbers. Money returned to you from the investment is specified as a positive number. *Cell range(s).*

The NPV function calculates the net present value of the investment on the basis of the initial investment, the desired rate of return and the cashflows that have occurred. The IRR function performs somewhat the inverse function calculates the the rate of return that has been realized for the cashflows.

The cashflow tables may be separate ranges. The order of execution is the order you present. Order is very important to both functions.

As a mathematical note, every reversal in direction of cashflow implies an inflection point and raises the power of the necessary polynomial to be fitted by one. Some cashflows may not have a solution that is readily apparent to the algorithm used or that is realistic. The algorithm employed in QueryCalc is particularly stable and will deduce most reasonable internal rates of return.

NPV(initial-investment,irr-desired,cashflows)

IRR(initial-investment,cashflows)

Example 1: Return on investment with irregular cashflows

You are asked to invest $10,000 in a software development corporation. The expected quarterly cashflows are expected to look like this:

1st year:	-500	-1500	0	0
2nd year:	0	500	1100	3200
3rd year:	4100	5200	3600	2700
4th year:	3050	2200	1060	900

The negative entries indicate that cash is flowing away from you (the development company needed more money than was originally anticipated). Positive entries indicate that money is flowing towards you. Zero entries can be left blank. "Holes" in the table are assumed to be zeroes. Be careful about your tables. Money received further downstream is not worth as much as money received immediately.

What would be your annualized return per year (APR) for the four year period? Assume that this table is in the rectangle B8:E11.

The answer will be obtained by typing:

IRR(10000,B8:E11)*4

The returned IRR is for the implied period of the cashflows (every three months in this example) so it must be multiplied by 4 to get the correct APR. The answer is: 33.71%/annum.

NPV(initial-investment,irr-desired,cashflows)

IRR(initial-investment,cashflows)

Example 2: Calculating Net Present Value with irregular cashflows

Using the same example, what would the NPV be for the same investment, the same cashflows and a desired minimum return per annum of 15%? (Because we calculated the return to be 33.71% in the previous example, we know we'll exceed the minimum desired return.)

To calculate the NPV, type:

NPV(10000,15/4,B8:E11)

Once again, the 15% APR must be divided by the implied frequency of the cashflows, which is every quarter in this case. The calculated NPV is: $6,752.02.

Because the NPV is positive, we know we made money over and above the required 15%/annum return necessary. A negative NPV indicates that the desired return was not realized.

Example 3: Calculating the maximum investment for a certain return

Using this same example again, what would be the maximum initial investment be that would have returned 15%/annum with the specified cashflows? The solution is quite simple, but not necessarily obvious. Invest nothing initially (at least on paper) and the NPV calculated will be the "bankable" value of the investment.

NPV(0,15/4,B8:E11)

The returned value is: $16,752.02 (which as you may notice is the sum of the previous example's initial investment of $10,000 and the calculated NPV of $6,752.02). Any dollar amount initially invested that is less than $16,752.02 will return more than 15%/annum. Any higher investment won't return the desired minimum return.

FINANCIAL FUNCTIONS(cont'd)

DEPREC(calculation,type,initial-value,lifetime,corp-year-month, service-date,depreciation-date,*salvage-value,dbf*)

calculation — which type of calculation to be performed. Only NV (net remaining value) or YD (yearly depreciation) are acceptable entries. *Text equation.*

type — which type of depreciation calculation to be used. Acceptable entries are:

SL	straight line
SOY	sum of years
DB	declining balance
ACRS	accelerated cost recovery system
LIACRS	low income ACRS

Text equation.

initial value — the initial value of the item to be depreciated. *Numeric equation.*

lifetime — depreciable lifetime of the item in years. *Numeric equation.*

corp year month — first month of corporate year. *Text equation (although numeric entries are acceptable where 8 implies August).*

service date — year, month, day (written as a date in any normal fashion) when item first put into service. *Text equation.*

depreciation date — year, month, day (in any normal fashion) to specify date to which depreciation is calculated. SYSDATE is acceptable entry. Depreciation calculated only to the month and not to the day. *Text equation.*

salvage value *(optional)* — salvage value of the item. *Numeric equation.* If no value is specified, zero is assumed.

dbf (optional) — declining balance factor. 100 to 200 (percent) are only acceptable values. If no value is specified, 100% is assumed. *Numeric equation.*

DEPREC(calculation,type,initial-value,lifetime,corp-year-month, service-date,depreciation-date,*salvage-value,dbf*)

Example 1: Straight-line depreciation of equipment

What is the year-to-date depreciation and remaining net value on a tractor costing $175,000, purchased January 15, 1982? Salvage (resale) value is estimated to be $32,000. The serviceable lifetime is 7 years and the company's corporate year begins in October.

To calculate the year-to-date depreciation and remaining net value, type in a cell:

```
DEPREC("NV","SL",175000,7,"OCT","Jan 15, 82",sysdate,32000)
```

and

```
DEPREC("YD","SL",175000,7,10,"1/15/82","sysdate",32000)
```

The system date of the computer was July 15, 1987. The calculated values are:

Net remaining value: $60,940.48
Year-to-date depreciation: $17,023.81

Some variation in the way in which parameters may be entered was purposefully changed in the two examples above. The corporate-year first month was entered as "OCT" in the first function and simply as 10 in the second. The in-service date was similarly changed.

Some of the depreciation information may occur on the spreadsheet. Cell references may be readily substituted for each of the parameters, as in:

```
DEPREC("YD","SL",b15,c15,"oct",d15,sysdate,e15)
```

where the initial cost, lifetime, in-service date and salvage value are respectively taken off of the spreadsheet.

MACRO COMMANDS

Standard QueryCalc commands may be strung together to cause automatic execution of reports, either in session or batch mode.

/PROG — activates an editor that has been designed to look and act much like EDIT/3000. A macro program may be created internally within QueryCalc or externally with any other ASCII editor.

or

/PROG macroname

Commands available within the macro editor:

T[ext] macroname	load specified macro
A[dd]	adds line to current of file
A[dd] 17.1	adds line following line 17.1
M[odify] 4	modifies line 4
M[odify] 7/10	modifies lines 7 thru 10
D[elete] 16	deletes line 16
D[elete] 5/10	deletes lines 5 thru 10
D[elete] ALL	deletes entire program
R[eplace] 5	replace line 5
R[eplace] 16/20	replace lines 16 thru 20
S&R "old" "new" ALL	replace every occurrence of "old" with "new"
S&R "in" "out" 5/20	replace "in" with "out" in lines 5 thru 20
COPY 40/100 to 201	copies lines 40 thru 100 to 201
COPY 10 to 38	copies line 10 to 38
RENUM 1000,10	renumber program, starting at 1000 in steps of 10
RENUM 10	renumber program, starting at 10 in steps of 1
L[ist] ALL	list entire program ("all" is optional)
L[ist] 5	list line 5 or next succeeding line
L[ist] 10/40	list lines 10 thru 40
P[rint] ALL	prints entire program ("all" is optional)
P[rint] 10/40	prints lines 10 thru 40
K[eep] macroname	saves specified macro
S[ave] macroname	saves specified macro
X[execute]	executes macro file currently loaded in editor
E[xit]	exits the editor

Example: A simple program to load, execute, save the calculated results and print selected pages:

```
1  /load fgreport
2  !!
3  /save fgreport
4  /print b:g s
```

Almost all standard QueryCalc commands, numeric equations, text equations, text labels and query questions can be used in the macros. There do exist certain exceptions however.

The following commands cannot be executed through the macros (for obvious reasons in each case):

/COLOR
/E
/LAST
/REDO
/CAT
/DIR
/INFO
/SETUP
/MARG *(margins can be set in macros)*
/COLS
/CALEN
/PROG
/EXEC
/FUNCT

Additionally, graphics commands cannot be executed in a batch-drived macro mode because the graphs must be drawn onto a terminal screen (which does not exist in batch mode).

/VIEW *(not possible in batch mode)*

MACRO COMMANDS(cont'd)

Inserting information or an equation into a cell during macro execution:

```
AA1: "This is a text label
CB9: @Using qcdemo.employees, sum of amount
DE15: $"Value: "+aa12
```

The selected cell address is presented as shown. The insertion of the normal QueryCalc inputs proceeds exactly as it would if the cursor were located at that cell. *Page, row & column must be completely specified.*

Setting in default values for row, column and page during macro execution:

/J AE5	sets in page A, column E, row 5
/J B	sets in page B, column A, row 1
/J G15	sets in column G, row 15 on current page
GE15:(...equation...)	sets in page G, column E, row 15 as an implicit jump command

Relative cell addressing:

***:(...equation...)**	sets in equation at current row, column and page address. The character "*" indicates current cell address.
***+2R:(...equation...)**	sets in equation at current row+2, same column, same page. Current row is incremented before equation is injected into cell.
***+2R-1C+2P:(..eq..)**	row, column and page can all be incremented by specified amount from current position before injecting new equation into cell.

Alternative cell addressing:

%ROW=3 **%COL=%COL+3** **%PAG=%macrovariable**	The present row, column and page information is accessible and modifiable through the use of three predefined macro variables: %ROW, %COL, and %PAG.

Changing margin settings or selected printer during macro execution:

LEFT=132	sets left margin to 132 for subsequent printing
RIGHT=5	sets right margin to 5 for subsequent printing
TOP=6	sets top margin to 6 lines before printing commences
BOTTOM=2	sets bottom margin to blank lines of bottom of page
LINES=60	sets printing page length to 60

MACRO COMMANDS(cont'd)

Programming constructs available in the macro execution interpreter:

JUMPS

10 GOTO labelname
jumps to labeled line.

12 labelname:
labeled line. Only labelname allowed on the line. Labelnames may be 16 characters. 20 line labels are allowed. Line labels are terminated with a ":".

14 CALL labelname
executes user programmed subroutine commencing at line label. The subroutine must end with a **RETURN**.

LOCAL VARIABLES

14 %variable=equation
local variables are defined as the result of either numeric equations, text equations, query questions or other macro variables. Local variables are not typed as text or numeric. To use numeric variable in succeeding equation or query question, simply use name. To use text variable, use quotes around name (e.g. "%length").

20 macrovariables are allowed, the name of which may be no more than 9 characters in length.

IF STATEMENTS

```
30 IF $AB6=$"JAPAN" THEN DO
   .
   .
40 DOEND
```

if expression evaluates as true, lines between DO to DOEND are executed.

50 IF %START>2 THEN linelabel

if expression is true, a jump to indicated labeled line is executed.

60 IF $"%NAME" <> $"DALTON" THEN statement

if expression evaluates as true, any macro command or equation may be executed (e.g. /PRINT, !!, %value=4, @sum of ..., etc.)

REPETITIVE LIST EXECUTION

51 LISTFILE #1=filename[.groupname[.accountname]]
52 LISTFILE #3=!A

the listfile may be either an MPE flat ASCII file (as in line 51) or a QueryCalc @STORE file (as in line 52). Six independent listfiles (numbers 1 to 6) may be maintained simultaneously.

A listfile may be "restarted" (the record pointer reset to the beginning of the list) by simply respecifying the listfile again.

53 READ LISTFILE #4 reads next item from search item list 4 into local macro variable %LISTITEM4 for your use.

54 ON END OF LIST #3 labelname

establishes jump point when end of list 3 is reached.

60 END causes the end of execution of the macro. May be placed anywhere in the macroprogram.

MACRO COMMANDS(cont'd)

Executing the macro in session mode:

/EXEC
/EXEC macroname — executes macro in session mode. If macroname is unspecified, name is asked. CNTL-Y will break execution cycle.

Executing the macro in batch mode:

If the file QCMACROF is defined before QueryCalc is run, QueryCalc will automatically enter the macro execution mode. This process can be streamed or scheduled for execution at a particular time.

Example:

```
:STREAM
>!JOB USER/PASSWORD.ACCOUNT/PASSWORD,GROUP/PASSWORD
>!FILE QCMACROF=filename[.groupname[.accountname]]
>!RUN QC.QCPROGS.AICS (or use UDC if defined)
>!EOJ
>:EOD
```

Printing in the macro execution mode:

Printing proceeds in the macro execution mode exactly as it does through normal hand-typed commands. There is one difference however. In normal command mode, the print destination file (local terminal, system printer or MPE flat file) is opened at the beginning the print command and closed immediately on completion. In the macro execution mode, the print file remains open until either a new file is specified or the macro ends.

Prints to the disc file, local terminal or system printer are independent files. Printing may be directed to all three simultaneously without confusion. Changing a file destination for the system printer will have no effect on the current disc file.

Example:

```
/load myreport
Za1:@Using employees, store in !a socsecnum when zip is 88047!
listfile #1=!a
on end of list #1 exit
loop:
     read listfile #1
     ab7: '%listitem1
     !!a,b
     /print a,b d reportx
     goto loop
exit:
```

The file REPORTX will be opened the first time it is encountered. If REPORTX does not exist, it will be created based on current margin settings. REPORTX will remain open until the macro ends.

MACRO COMMANDS(cont'd)

Further examples of macro programs:

Example: You may have a distributed database on several remote machines if you were a retail chain operating a series of retail outlets. You may wish to get information on overnight sales from the store in Des Moines, place it on page A, then get information from Moline and place it on page B, and so on. With each retrieval the DS phone line must be dialed, hung up and redialed.

```
/system dsline 12;open;phnum=1515xxxxxxx
/system remote hello user.account/password,group
/system file db.dbgroup.dbaccount;dev=12#
/load report
!!a
/save report
/system remote bye
/system dsline 12;close
/system dsline 12;open;phnum=1815xxxxxx
/system file db.dbgroup.dbaccount;dev=12#
/load report
!!b
/save report
system remote bye
/system dsline 12;close
/system dsline 12;open;phnum=1417xxxxxx
/system file db.dbgroup.dbaccount;dev=12#
/load report
!!c
/save report
/system remote bye
/system dsline 12;close
/load report
!!e:n
/system file qclistf;dev=17;cctl
/print e:g s
/save report
```

Once all of the data has been gathered from remote machines, the summary pages (E thru N) are recalculated. The system printer is redefined and the critical pages (E thru G) are printed out and will be ready to be read in the morning.

Example: A report named DOCTORS creates lists of serviced patients based on a range of dates. The report programmed in DOCTORS specifies treatments performed and their costs for each doctor in the professional group which shares the computer.

```
>********************************
/LOAD DOCTORS
!!AB7:AB13
%L#=1
>********************************
NEWLIST:
IF %L#=1 THEN DO
    BA1:'Doctor:  Phillips
    %LIST=$"!A"
    %DISCFILE=$"DOCLST1"
DOEND
IF %L#=2 THEN DO
    BA1:'Doctor:  Hoyt
    %LIST=$"!B"
    %DISCFILE=$"DOCLST2"
DOEND
IF %L#=3 THEN DO
    BA1:'Doctor:  Wilson
    %LIST=$"!C"
    %DISCFILE=$"DOCLST3"
DOEND
IF %L#>3 THEN EXIT
START:
    LINES=5
    /PRINT B D %DISCFILE
    LINES=4
    LISTFILE #1=%LIST
    ON END OF LIST ENDLIST
READ:
    READ LISTFILE #1
    CA1:$"%LISTITEM1"
    !!C:D
    /PRINT D D %DISCFILE
    GOTO READ
ENDLIST:
    /PRINT E D %DISCFILE
    %L#=%L#+1
    GOTO NEWLIST
EXIT:
```

In this report, the printing destination disc file and the source list of patient names is changed out using the local macro variables %LIST and %DISCFILE.

LEAVING QUERYCALC

To momentarily return to MPE, type:

/SYSTEM*

MPE commands which do not require invoking a new process (such as :LISTF, :PURGE, :BUILD, :HELP, etc.) can be executed.

/SYSTEM FILE QCLISTF;DEV=103;CCTL

A one-line version of the /SYSTEM command can be formulated as above. This form is macro-executable. Command returns immediately to QueryCalc on both successful and unsuccessful execution.

To suspend QueryCalc and run another program, type:

/RUN EDITOR.PUB.SYS*

The name of the program must be qualified, if not in the user's signon group or account.

UDC's (user defined commands) and MPE's run command qualifiers are not supported.

Upon completion of the invoked program, QueryCalc resumes at the point of suspension.

To fully exit QueryCalc, type:

/EXIT

* *Note:* The /SYSTEM and /RUN commands may be disallowed by your HP3000 System Manager.

INDEX