3

Using the spreadsheet as a report generation platform

The 3-D Spreadsheet

Introduction to the Spreadsheet

Although the electronic spreadsheet is now extraordinarily popular and widely used, it's sometimes difficult to remember that it is not very old. The electronic spreadsheet was invented by Dan Bricklin in 1977 during his time as a graduate student at Harvard. As a business major, Bricklin was required to generate *pro forma* (paper spreadsheet) analyses of various commercial situations as part of his course work. Part and parcel of any spreadsheet is the characteristic that if you change one number, the effects ripple down the columns and across the rows. Bricklin invented the electronic spreadsheet simply because he grew tired of erasing holes in his papers due to constantly changing numbers.

The electronic spreadsheet makes an ideal report generating platform because of its inherent plasticity and its WYSIWYG nature. If you are not presently familar with spreadsheets, two attributes account for much of the spreadsheet's value. They are the ease by which columns and rows may be inserted, deleted and reformatted (while automatically adjusting all relevant cell references) and the capacity to replicate and edit blocks of cells. These two features alone eliminate a great deal of the effort necessary to put a report together. The design of QueryCalc especially emphasizes the report formatting features of the spreadsheet.

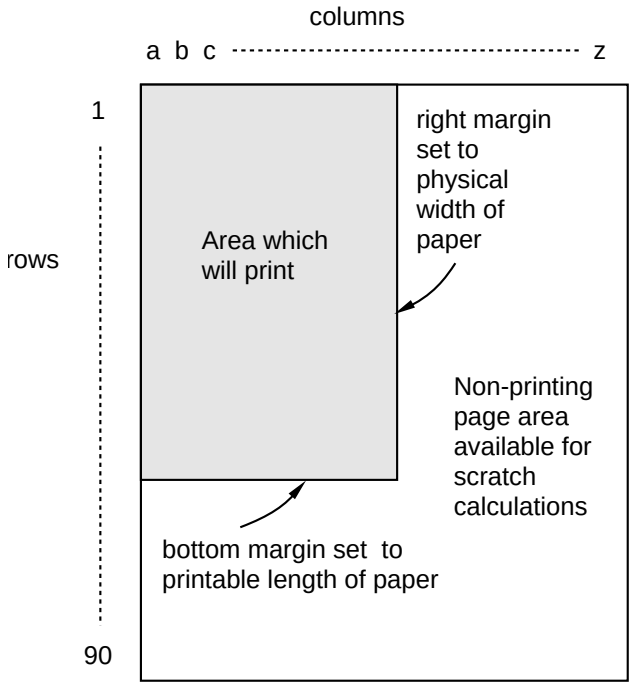
QueryCalc is a spreadsheet, and you can simply use it as a spreadsheet. But QueryCalc's more than that. QueryCalc is a combination of the most productive features of interactive computer languages, page-oriented word processors, spreadsheets and query languages. QueryCalc mimics a page-oriented word processing program as much as it does a spreadsheet. Where a standard spreadsheet is built as one very large, flat XY plane, QueryCalc is composed of 26, independently configurable pages.

You'll use QueryCalc differently than a standard spreadsheet. A PC-based spreadsheet is often used as a small database. Thousands of cells may be

used to store information such as outstanding invoices. If only eight cells in a row were used to list items such as customer name, account number, invoice number, invoice date, etc., a thousand cells will be consumed in listing of just 120 entries. One hundred entries obviously represents a very small database; nonetheless, 1000 cells take up a substantial area on the spreadsheet. You simply will not tend to use QueryCalc in such a fashion. IMAGE databases easily contain tens of thousands of records. A single cell in QueryCalc can request a summarization of all of those records.

Printing the QueryCalc Page

QueryCalc also differs from standard spreadsheets in the manner it which printing occurs. QueryCalc will always print from the upper left-hand visible corner of the spreadsheet page out to the right margin and down to the bottom margin. A PC-based spreadsheet, by comparison, prints by specifying ranges of cells taken anywhere from within the body of the spreadsheet. *In QueryCalc, the upper left-hand corner of each spreadsheet page is the image of the physical page that will be printed. Although the cell widths on each page are independently configurable, the printing area remains the same on every page of the spreadsheet.* That's because the size of the paper you're printing to won't change, from page to page.



Only fifteen rows will be displayed on your terminal's screen at any one time, although 90 rows are available on the spreadsheet page. No readily available piece of paper is greater than 90 rows in length. The printable length of the spreadsheet page will be dynamically determined based on: (1) the paper size you specify, (2) the number of lines per page, and (3) the top and bottom margin settings. As you increase the size of the top and bottom margins, or add headers or footers, you simultaneously shrink the number of rows available to you in the print area. The physical size of the paper can't change, therefore the printable area must shrink (for more information, see Chap. 7, "Printing the Report").

On a QueryCalc page, the printable area is defined by the right and bottom margin markers.

Learning to Become a Marginal Programmer

Traditionally, when numbers are entered into the rows and columns on an accountant's paper greensheet, they are placed there neatly and with precision. But the necessary scratch calculations are scribbled in the margins. We want you to use QueryCalc's margins in the same manner. All of the area to the right and below the margins on a page can be used for scratch calculations, temporary variables, or anything else you need to have available during report development, but don't wish to appear in the final printed form.

With a QueryCalc spreadsheet, *the margins don't even necessarily have to be at the sides*. Individual columns can be collapsed to zero width [using the /CWID command (see Ch. 4)]. Zero-width columns will no longer print, but the information in these hidden cells will recalculate normally and the results can be referenced as any other cell would be.

Cell definitions on any one of QueryCalc's 26 pages can reference cells anywhere in the spreadsheet. Sums (averages, variances or whatever) may range over columns, rows, rectangles, cubes or any combination of these. This feature accounts for the true three-dimensional nature of QueryCalc. But you'll find that the three-dimensional nature is more valuable than simply that. Although each page may be substantially different in what it is calculating and where it is obtaining its information, the easy reference to previously obtained values on other pages makes the preparation of complex reports simple.

The Three Standard Report Forms

People tend to only write three kinds of reports on the HP3000. They are:

- *The indefinite detail list report*
- *The management summary report*
- *Graphics*

Each of these forms is easily done in QueryCalc. The first, the *indefinite detail list* report, is of this form: "Find me all of the people who owe us money and print me a list". The formating structure for the indefinite list report tends to be generally simple, but the length of the report is unknown before it's run, thus its indefinite nature. The second report form, a *management summary* report, is usually much more complicated to put together, and is generally quite complex in its formatting. Information is not simply repeated over and over as it is a detail list report. A management summary report is composed of many summarizations taken from many

different sources and then placed on specific areas of the page. The spreadsheet structure is obviously ideal for constructing the management summary report. The first and most obvious reason is the flexibility allowed in reformatting the page layout. But the most important reason is the interactive nature of the spreadsheet.

The Four Cell Types of QueryCalc

Standard spreadsheets such as VisiCalc and 1-2-3 possess only two types of cells: *numeric equation* cells (which include simple numeric entries) and *text labels*. In contrast, QueryCalc has four cell types. They are the two VisiCalc cell types, *text equations* and *query questions*.

A **numeric cell equation** may be any one of the following forms:

```
34
sqr(g34)*1.0675*(g37+g38)
bd15*4
sum(aa1:cd13)
```

A **text label** is simply text written on the surface of the spreadsheet, not unlike this sentence. Text labels are not active cell types (that is, they do not change on recalculation). Examples are:

```
^Jan 1992
'Expense Report for FY94
```

A **text equation**, on the other hand, is an active cell type. The text equation extracts text information from other cells anywhere on the spreadsheet and recombines the text as you desire. Examples are:

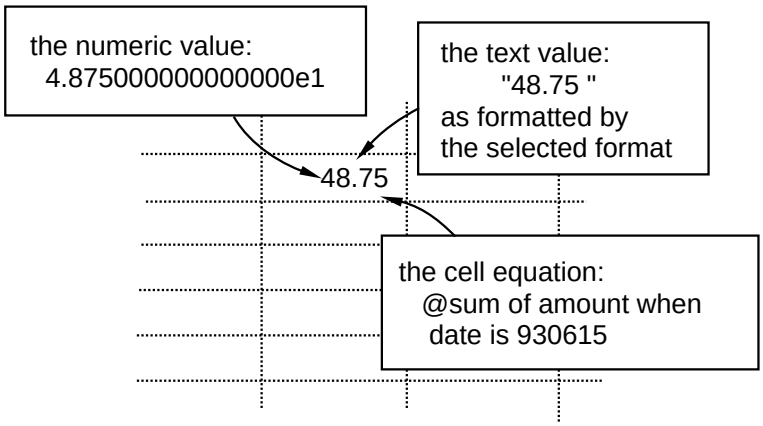
```
$"The value is: "+ups$(g34)
$a34+", "+a35
$dat$3(sysdate)+", "+hrs$(24)
```

A **query question** extracts and summarizes information from any of the databases (IMAGE, KSAM or MPE) which you presently have open. Examples are:

```
@Using invoices, sum of amount when
   category is 501 and date>890530
@Using qcdemo.labor, avg of overtime
   when jobnum is 8404
```

The Triple Nature of Every Cell in QueryCalc

Every cell in QueryCalc, regardless of cell type, is composed of three parts: the *cell equation*, the cell's *text value* and the cell's *numeric value*.†



QueryCalc is extremely weakly typed. That means you can add numbers and text together and get an answer. Any text cell that contains only numeric characters takes on the numeric value of that text (that is, "3956" = 3956). Text cells which contain both alpha and numeric characters (or punctuation symbols) cannot be converted to a number, thus they are given a numeric value of zero ("-56A-" = 0). Conversely, all cells have a text value. Text from any or all of the four cell types can be combined in a single text equation.

A numeric equation, such as

```
b37*1.4
```

extracts and uses the numeric value from the cell B37. A text equation such as

```
$"Result: "+b37
```

extracts the text (formatted) value from cell B37. *Although the cell formatting routine will often round off the text in a cell to the specified number of digits, the actual numeric value is never rounded off. The effect is only superficial.* The numeric equation always works with QueryCalc's maximum resolution of 16.5 decimal points and never throws away information.

---

†A detailed explanation of how a QueryCalc cell is constructed appears in the Technical Appendix, Section A: "How a QueryCalc Cell is Constructed".

The dual text/numeric nature of cells in QueryCalc is especially important to query questions. Text and numbers are the only two forms of items that can be stored in a database. To properly find (qualify) items in a database, text must be matched with text and numbers must be matched with numbers.

The following examples illustrate the point:

```
1. @Using invoices, sum of amount when
       category is [b7]

2. @Using invoices, sum of amount when
       category is [$b7]
```

For Example 1, presume that the dataitem CATEGORY is a numeric (I,J,K,P,Z or R) dataitem type. The equation in brackets which retrieves information off of the spreadsheet should therefore be a numeric equation (B7). A numeric value will properly be matched against a numeric value.

For Example 2, presume that CATEGORY is a text (X or U-type) dataitem. The equation in brackets must therefore be a text equation ($B7). The spreadsheet-extracted text will be matched against a text dataitem.

QueryCalc will do its best to properly match item types, but it may occasionally require your help. Should you ever have problems qualifying items in a database with values taken off of the spreadsheet, the problem almost always lies in mistaking a text field for a numeric field. *Text fields are used surprisingly often to store only numbers.* When numbers appear in a text field, they may look like numbers, but they're not. They're text characters and they must be treated that way.

Text and numeric equations which appear in query questions do not need to be simple spreadsheet data extractions. You may dynamically "manufacture" the patterns to be matched, as illustrated in the following examples:

```
@Using labor.tickets, avg of regular-hours
    when date is [190000+ab7*100+ab8]
```

(where date is a numeric dataitem)

```
@Using parts.partfields, find when partnum
    is [$"MJR"+hg34+"-00-1A"]
```

(where partnum is a text dataitem)

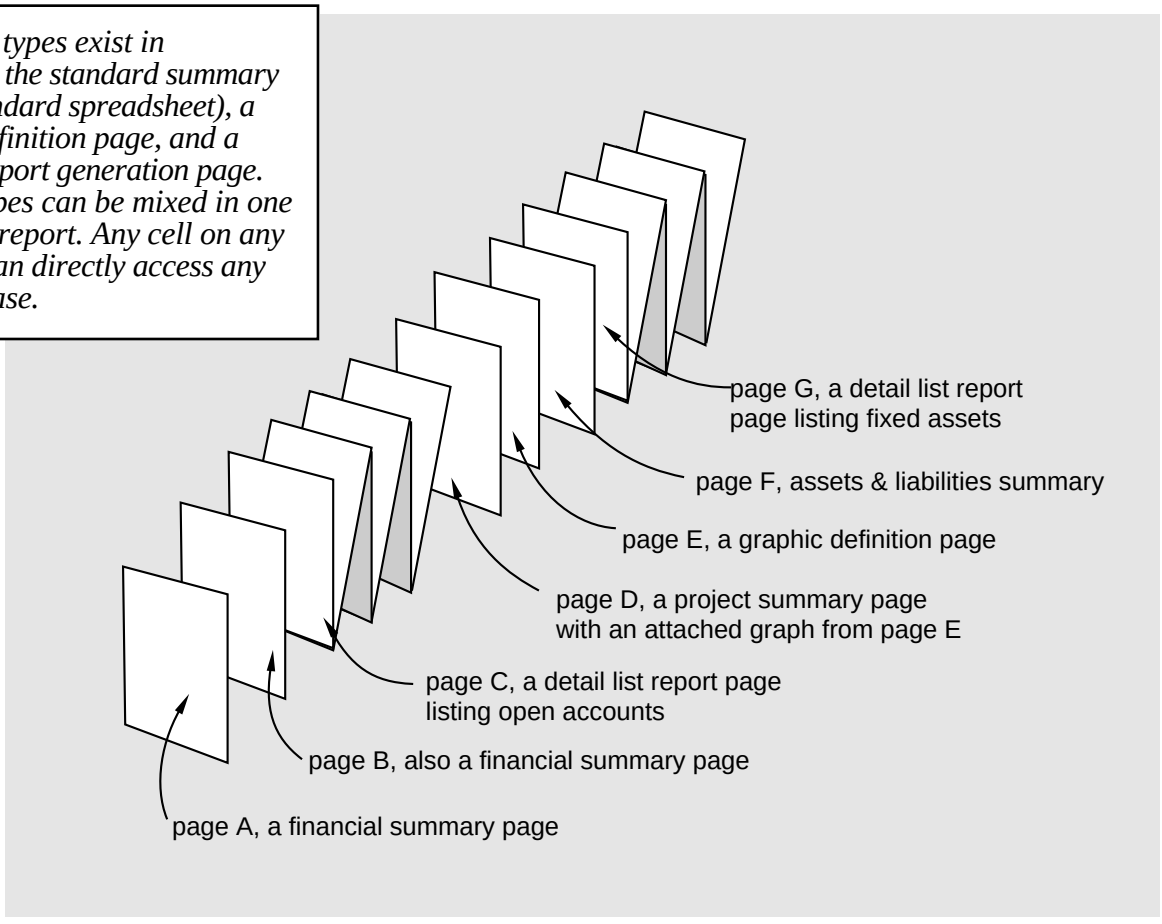
The Three Page Types of QueryCalc

A QueryCalc report is composed of three distinct page types:

- *summary pages*
  (standard spreadsheets with or without query questions in the cells)
- *graphics definition pages*
  (used to define a graph)
- *detail list report generation pages*
  (used to create a detailed list of items).

The cells found in each of the pages, however, are identical. Any cell on any page can access any of the open databases, local or remote, to which you have access. The combination of the three page types allows the creation of virtually every possible form of report. Although the page types are distinct in their usages, the standard spreadsheet summary page is the basis of all three page types.

Three page types exist in QueryCalc: the standard summary page (a standard spreadsheet), a graphics definition page, and a detail list report generation page. All three types can be mixed in one QueryCalc report. Any cell on any page type can directly access any open database.



A *graphics definition page* is a standard spreadsheet summary page with a graphic template superimposed. By filling in the requisite information in the template, any one of six basic graph types (*standard bar, stacked bar, clustered bar, pie, point plot, and high/low/close chart*) is defined. Information may come from anywhere else in the spreadsheet or be calculated immediately in the definition cell itself (see Chap. 9, "Graphics").

The *indefinite detail list report page* is the most distinct page type of the three page types found in QueryCalc. Only four columns are used on the report page. You cannot increase or decrease the number of columns, nor can you change the column widths. The columns have pre-defined usages. The first column, column A, is used only for cell equations. The second column, B, is used to the name the extracted or calculated dataitems. Columns C and D are optionally used to define sort order and allocated print width, respectively. Rows are used to define groupings within the report (see Chap. 13, "Indefinite Lists").

When the completed report is to be printed, only the standard spreadsheet summary pages will print as they appear. The graphic definition pages cannot of course be printed directly. Rather the information specified on the graphic definition pages is transformed into the graphs they define. Nor will the indefinite list report pages print as they appear. Although a detail list report is composed on only one spreadsheet page, it may result in several to several hundred actual (paper) pages of output.

A QueryCalc report may be any mixture of the three page types you desire. An example is shown on the previous page. The first two pages are financial summaries while the third page (indefinitely) lists all open accounts. The fourth page is an additional financial summary with an attached graph. The sixth page is a standard assets and liability summary while the seventh page is another detail list listing all current fixed assets. A report as complicated as that shown is never more than a simple combination of simple ideas used over and over again. Each cell on each page type is identical. Most importantly, the procedure to put each report page together is the same. You enter the cell equation, check the results, and interactively repeat the process until you are satisfied.

Concepts Introduced in Chapter 3

SUMMARY REPORT	a standard spreadsheet page. Cells may directly summarize database information.
GRAPHICS PAGE	a modified spreadsheet page with a predefined template. Information filling the cells defines the graph.
DETAIL LIST REPORT	an indefinite list of items extracted from a database(s), totalled and sorted within groups.
NUMERIC CELLS	a standard mathematical equation which may be as simple as a single number.
TEXT LABELS	Text written on the "surface" of the spreadsheet. Text labels are non-calculating cells.
TEXT EQUATIONS	equations capable of active text reorganization during recalculation
QUERY QUESTIONS	database summarizations.

InterChapter Reminder

The only way QueryCalc will become useful to you is for you to know what's in your databases. That means understanding how a database on the HP3000 is put together and what each dataitem in your own databases means. This advice doesn't apply solely to QueryCalc. It's only common sense. However, this is not difficult. If necessary, review Chapter 2. And then ask someone about the databases on your machine. If you are confused and you can't get help locally or from your applications program supplier(s), call us. We'll do our best to help. Our telephone numbers are:

(800) AICS-INC *(United States)*
(505) 524-9800 *(elsewhere)*
(505) 526-4700 *(FAX)*

Our hours are 8AM to 5PM, Mountain Time, weekdays. To relate our time zone to yours, if you are in London, subtract 7 hours from your local time. If you are in Sydney, add 7 hours to your local time.