This free back issue of
THE QUICK ANSWER is provided courtesy of…

The Monthly Newsletter for Sesame Database Manager
http://www.insidesesame.com
Read a Free Issue of Inside Sesame!

AND

L A N T I C A
S O F T W A R E   L L C.

Makers of Sesame Database Manager
Compatible with Symantec Q&A™
http://www.lantica.com

Importing Huge ASCII Data Files

Michael Berns

RECENTLY, I attempted to import into Q&A 5.0 for DOS a Fixed ASCII file containing all the registered voters in one of the five counties that make up New York City. The file, provided on a CD, consisted of 890,000 names/records in 56 fields/columns with a total of 455 characters per record. This single ASCII file weighed in at a whopping 486MB!

Others I talked to who had dealt with this particular file (available from a public source) complained that it was just too massive to import into any desktop program—database manager or otherwise. I initially thought it might take half an hour or so to import it into a Q&A database. But as it turned out, I was looking at a process that would take *days* on my older but not ancient PC. I couldn't afford to tie up my computer for this length of time, so I had to get inventive. What follows are the specifics of the problem, the "bright idea" I had, and the ultimate resolution to the problem of importing even *gigabyte*-sized ASCII files into Q&A.

Q&A looks like it can't cut it

I originally set up my Q&A database, VOTERS.DTF, with the 56 fields as defined by the structure of the Fixed ASCII source file and delineated in the "ASCII Fixed Guide" that came with the CD. I then went to *File / Utilities / Import / Fixed ASCII*, filled out my Merge Spec appropriately, and hit F10.

The first 1,000 records imported in a matter of seconds, the next 1,000 took little longer. The first 5,000 took 10 minutes; the next 5,000 took 30 minutes. I left the machine on and came back 60 minutes later. Q&A had slogged through the first 30,000 names. It took Q&A six hours to get just the first 60,000 done.

Q&A was progressively slowing down as it imported more and more records. At this rate, I calculated that it would take *days*—non-stop—to import all 890,000 names, assuming absolutely nothing went wrong. If I stopped the process, my database would have the 70-80,000 names imported so far. But if I then restarted it, Q&A would begin importing at record #1 all over again. Adding to my concern was the fact that I was hoping to get an updated copy of this file every year—and perhaps add another county or two next year, for another 2,000,000 or so records.

So Q&A seemed clearly not the way to go on this.

Any other options out there?

Microsoft Excel can only handle a maximum of 65,000 records. Quattro Pro can't handle a file of this size, either. Microsoft Word and WordPerfect were out of the question. My hope was that by using one of these other programs, I could somehow chop the massive file up into more manageable pieces. Nothing worked. I was told that Microsoft Access could do it, but that would involve purchasing the program, learning it from scratch, and then keeping my fingers crossed that it would somehow be able to import the data orders of magnitude faster than Q&A.

Then it struck me that the inordinate amount of time it was taking Q&A to import the file might not necessarily be due to the number of records in the source file, but the process of parsing and distributing each imported record into the corresponding 56 database fields. Since the process was

surely taxing system resources anyway, the more fields into which Q&A had to distribute the data, the more time this would add to the overall import process.

I realized that to test my theory I would have to try importing each record in the source file into a single large database field, and measure how much faster this went when compared with parsing and distributing each record into the 56 fields of varying lengths.

Testing the theory

I started by designing a database named MASTER.DTF with only the one field I named **RAW-DATA**. When I imported the 486MB file from the CD into this

Sesame Beta Released Broadly

Lantica Software, LLC has released the first version of it's *Sesame* Beta software that provides the testers with a real look at the integrated version of the program.

Prior to this release, testers had versions aimed at evaluation of specific features and operational parameters. This latest version has been released to more than 200 new Beta testers. It includes the Q&A import, new database design capability, and complete programming language capability.

Until this week only Alpha testers and a select group of Beta testers had seen these features. The overall responses have been extremely pleasing to both me and the development team.

"I'm delighted by the overwhelmingly positive response we've received from both the alpha test group and the initial beta test group. Everyone seems to be really having fun building and testing databases!" said Erika Yoxall, the *Sesame* Development Team Leader.

Bobbi Baker, a database professional who is a member of the early beta test group exclaims, "Sure, stability, speed, etc. are the most important features of a database. BUT when I take my laptop to meet with a potential client for the first time they want something to look at, both as an example of the software and my work. This is guaranteed to make a terrific first impression!"

According to Bob Scott, a Q&A power user, "*Sesame* is packed with amazing features and is surprisingly easy to use….The Lantica team has created an incredible product that is more than I could have ever asked for or dreamed of."

We are moving ahead with the testing process by reviewing and evaluating every comment or response from the Beta testers. This release marks a major milestone in the development effort.

Bill Halpern
CEO/President Lantica Software, LLC

database, it took just 25 minutes. That was progress.

Now it was time to split up the data into more manageable subgroups. In the case of New York City voter registrations, among a number of other codes used are what are called *assembly districts* or AD's. There are 12 AD's in the county—62nd AD to 73rd AD—with an approximately equal number of voters in each district. So I started by designing a new database called 62AD.DTF consisting of 57 fields. The first field was **RAW-DATA-IN** to match the **RAW-DATA** field in MASTER.DTF, and the next 56 fields were those listed in the "Guide" that came with the original CD source file. Later, I would copy this design to create 63AD.DTF, 64AD.DTF, and so forth, but first I had to perform a few more steps.

I returned to MASTER.DTF and added a new field I named **AD**—the field I would be using to break the database down to manageable sizes. (This field needed to be added to the database *after* importing the records from the source file. Adding it to the original design increased the 25 minutes mentioned earlier to a much longer time.) Using the "ASCII Guide" supplied with the file, I knew that the "AD" number I wanted started at column 145 in the file and was two characters in length.

I Mass Updated MASTER.DTF by setting **RAW-DATA** to field #1 at the Update Spec, and setting the second, **AD**, field to field #2. I typed the following program in the **AD** field:

```
#2 = @Mid(#1, 145, 2)
```

The Mass Update completed in 15 minutes. At this point, the **RAW-DATA** field contained 455 characters—an entire record from the source file—and **AD** contained the two-character assembly district code.

Still in MASTER.DTF, I went to *File / Copy / Copy selected records* from MASTER.DTF to 62-AD.DTF. At the Retrieve spec for MASTER, I typed "62" in the **AD** field. Then, at the Merge spec, I typed "1" into **RAW-DATA**— the field containing all 455 characters or one entire record. I hit F10 and headed for the refrigerator for something to drink. In 10 minutes, the 455 characters were copied from the **RAW-DATA** field in MASTER to the **RAW-DATA-IN** field in 62-AD.DTF.

I then ran a Mass Update on 62-AD.DTF. I left the Retrieve Spec blank to include all the records. At the Update Spec, I pressed Alt-F8 for the Spec I had created and saved earlier based on the "ASCII FIXED Guide" that came with the CD. (This saved Spec contained the programming to parse and distribute the long **RAW-DATA-IN** value to the 56 fields, consisting primarily of @Mid functions as shown in the earlier Mass Update program.) In 10 minutes, Q&A was done. Now, in 62-AD.DTF, I had all the information from the original CD for 82,000 of the 890,000 names, all nicely distributed into the corresponding fields.

The next step was to go back to *File / Copy* for 62-AD.DTF and copy its *design only* to another new database—this time 63-AD.DTF. I then repeated this to create 64-AD.DTF through 73-AD.DTF.

Then I returned to MASTER.DTF and, using the same steps as I did for copying the 82,000 records in the 62nd district to the 62-AD.DTF database, I copied the appropriate fields into 63-AD.DTF, 64-AD.DTF, and so forth. When all of these 11 databases were done, I redesigned each one to delete its **RAW-DATA-IN** field. I now had my 890,000 names segmented into 11 smaller, more manageable databases.

Most of the data I receive requests for is by assembly district, but occasionally I am asked for other criteria, such as a recent request I received for all Republicans under 40 years of age in all 11 assembly districts. The databases allow me to provide breakdowns by sex, date last voted, school district, absentee ballots, and many other criteria. In this case, I copied the design of the 62-AD.DTF file to a new database named UNDER40R.DTF. The Retrieve Spec I used included the **Date of Birth** field (">19630101") and Party ("R"). The Merge Spec I was able to leave blank. After I had copied all the fields on the approximately 2,000 young Republicans to the new database, I did the same with 63-AD.DTF, 64-AD.DTF, and so on. (I designed a macro speed this up.) In the end, I had a new database containing some 26,000 young Republicans. I exported the data from this database to a format that could be imported into an Excel workbook, which is what the requestor wanted. I imported the file into Excel, saved the workbook, and emailed it off.

Conclusion

I am sure that with a newer, faster computer, the original attempted import would have gone much faster. But I needed to work with the equipment I had. Fortunately, I had Q&A and knew my way around it enough to come up with a successful alternate technique. In my line of work I'm involved with a number of professional database engineers. They were all amazed that something as "old-fashioned" as Q&A for DOS had the wherewithal to handle a problem of this magnitude on a lowly desktop machine.

Now, after New York City, there's New York State!

Snag/Flag Gaps & Dups In Databases with Strict Numbering Sequences

TOM MARCELLUS

SOME types of database applications require a strict numbering sequence for things. In such cases it can be important to have a fast and reliable means of preventing duplicate numbers as well as catching gaps in the numbering sequence. Users make data entry errors and sometimes, for one reason or another, fail to enter critical data at all . Automatic @Numbering schemes in share databases can prove unreliable. Records can be inadvertently deleted. These are just a few of the gotchas that can cause a strict numbering sequence to be defeated.

Even with a numerical sort in a well-crafted report, it's easy to scan down lengthy columns of numbers and fail to spot one or more duplicate or missing numbers. 13846 followed by 13849—or 13451 followed by another 13451—in the middle of a couple of hundred other numbers might not jump right out at you, particularly if you're a bit tired or rushed.

Why strain your eyes (and *still* not be sure that you checked carefully enough) when you can have Q&A do the scanning faster and more accurately for you?

A sample scenario

A client called awhile back with a problem like this. His busy real estate title company writes thousands of checks every month for amounts ranging from as little as a few dollars each up to several hundred thousand dollars each. The actual check number assignments and check printing are taken care of by his Q&A application.

For the settlement process of each case, up to 30 checks might be issued. These checks need to start with a specific check number and be in strict check number order. His application could do this reliably, so that wasn't the problem. The *real* problem came during reconciliation.

When you get your bank statement each month, it shows the checks you wrote that cleared during the statement period. The list of check numbers might look something like this:

```
2345
2346
2347
*2349
2350
2351
*2353
2354
```

Check numbers preceded by an "*" indicate that there's one or more missing check numbers in the sequence—they haven't cleared the bank yet. When you reconcile your statement, you have to take these into account.

In the case of my client, cleared checks were being marked by his people in his "cleared" database as having cleared the bank. These he didn't have to worry about *per se*. What he *did* have to worry about were checks that *should have* cleared but didn't, or might have cleared but hadn't been marked in the database as such.

To assist him when reconciling each month, he wanted a simple "print-to-screen" report he could run for his "cleared" database that would flag missing check numbers in a way that would make them easy to spot. Having flagged them, he could then investigate just those ones.

I worked out the report Derived Column logic for him and sent him a sample database he could use to incorporate the logic into his own existing reconciliation reports. At that time, he only wanted to flag missing check numbers. Since then, I added another Derived Column to the report to additionally flag *duplicate* check numbers. It's easy to design a report that will do one or the other—or both—as you'll see.

Designing the report

To illustrate, I'll use a simple one-field database named CHECKNO.DTF (included in this month's download file). The single field, named **CheckNo**, is formatted "N" for numbers. You want to make it a number field so that Q&A will sort it in strict numerical sequence in a report or when viewing records on-screen with a Sort Spec. There are ways to sort numbers in a *text* field as well—even "numbers" containing dashes, prefixes and so forth—using the @Num and @Tonumber functions, but that's not our topic here.

For a report that flags *gaps* in a check number sequence, the Column/Sort Spec is:

```
CheckNo: 1, AS, I
```

This tells Q&A to make this column 1 of the report, sort the check numbers in ascending order, and make the

column invisible.

The Derived Columns for this report are:

```
Heading: Hidden Calc
Formula No. 1  @Text(#50 <> "" AND #1 - #50 > 1,
          "*")
Column/Sort Spec: 2,I

Heading: CheckNo
Formula No. 2  #2 + @Str(#1)
Column/Sort Spec: 4

Heading: Hidden CheckNo
Formula No. 3  #1
Column/Sort Spec: 50,I
```

This report will precede a check number with a "*" when there is a missing check number just prior to it. (See Figure 1.)

Here's how it works.

The check number for the record being processed is in column 1. Column 50, when Q&A gets to it, will be filled with a copy of that check number courtesy of column 2's formula.

The formula in the hidden column 2 is this:

```
 @Text(#50 <> "" AND #1 - #50 > 1, "*")
```

This tells Q&A:

If column 50 isn't empty and *the difference when subtracting column 1's value from column 50's value is greater than 1, then put a "*" in column 2.*

You need the *#50 <> ""* part to keep Q&A from treating the *first* check number processed as having a missing check number preceding it.

When processing the *second* record, column 50 will contain the check number from the *previously* processed record. At this point, you have the current check number as well as the previous check number, and you can do the math on them to see if their difference is greater than 1.

For each subsequent record processed, you'll always have the current record's check number as well as the previous record's check number. So the algorithm will hold true through to the end of the report.

Column 4 is the only column that prints. This column's formula—*#2 + @Str(#1)*—will prefix the check number with an "*" only if the preceding column (#2) contains an "*". In other words, if there's no gap (the difference between the two check numbers is 1), there will be nothing in column 2, so the next check number will print by itself with no "*".

For a report that flags *duplicate* check numbers, the Column/Sort Spec is the same as above. The Derived Columns are:

```
Heading: Hidden Calc
Formula No. 1  @Text(#50 <> "" AND #1 - #50 = 0,
          "**")
Column/Sort Spec: 2,I
```

```
Heading: CheckNo
Formula No. 2  #2 + @Str(#1)
Column/Sort Spec: 4

Heading: Hidden CheckNo
Formula No. 3  #1
Column/Sort Spec: 50,I
```

This report will precede a check number with "**" when there's a duplicate check number just prior to it. Of course, you can use any flag character or text, such as a "[DUP]" before the check number.

A combo report

To add both features to a report (a single "*" for missing check numbers and a double "**" for duplicate check numbers, you can use the following design that incorporates the gap-check as well as the dup-check formulas in the Derived Columns screen:

```
Heading: Hidden Calc for Gap
Formula No. 1  @Text(#50 <> "" AND #1 - #50 > 1,
          "*")
Column/Sort Spec: 2,I

Heading: Hidden Calc for Dup
Formula No. 2  @Text(#50 <> "" AND #1 - #50 = 0,
          "**")
Column/Sort Spec: 3,I

Heading: CheckNo
Formula No. 3  #2 + #3 + @Str(#1)
Column/Sort Spec: 4

Heading: Hidden CheckNo
Formula No. 4  #1
Column/Sort Spec: 50,I
```

Calculation-only columns like the ones in these reports have long been a mainstay of advanced Q&A users and developers. They give you a way to store temporary calculation results—or, in this case, an asterisk or two—to make your reports tell you more than just what's stored in the database records.

Tom Marcellus is editor of *The Quick Answer*

Figure 1. A report that flags missing check numbers.

@Help

EDITED BY WILLIAM HALPERN

Numbering Report Lines

The following report is from Excel. I have this same report layout in Q&A 4.0 but without the third "Item" column that numbers each line for each salesman then starts over again at the next break. Can you help me include this column in my Q&A 4.0 report. Thanks.

Vir

Vir's Excel report:

```
AS OF 03/19/2003

   SM      SALESMAN     Item STYLE#   QTY      PRICE

COR10   PETER CORVI      1  2092      1       285.00
                         2  9066      1       305.00
                         3  3076      1       235.00
                         4  9075      1       275.00
                         5  2086      1       280.00

  sub-Total:                                 1380.00

MIL45   BOB MILLER       1  2055      1       260.00
                         2  4067      1       175.00
                         3  2033      1       240.00
                         4  2009      1       315.00
                         5  2092      1       285.00
                         6  2366      1       360.00
                         7  2339      1       290.00

  sub-Total:                                 1925.00
  Total:                                     3305.00
```

There are two parts to the solution. One is numbering the lines, and the second is restarting that numbering each time the sorted value (salesman) changes. For Q&A for DOS, you do this with Derived Columns. Q&A creates the columns in a report sequentially, from left-to-right, then goes on to the next line of the report.

The Column/Sort Spec for the report should look like this:

```
SM: 10, AS
Salesman: 20, AS, CS
Style: 30
Qty: 40
Price: 50, T, ST
```

In preparing to print this report, after sorting all the records by their ID, Q&A will create column 10 for the first salesman ID, then move on to columns 20, 30, 40, and 50.

From the Column/Sort Spec, press F8 for the Derived Columns screen and create the following three Derived Columns:

```
Heading: Delayed SM
Formula: #10
Column Spec: 200, I

Heading: Item
Formula: @Text(#10 <> #200, 1) + @Text(#10 = #200,
       #190 + 1)
Column Spec: 25, F(JR)

Heading: Delayed count
Formula: #25
Column Spec: 190, I
```

The first Derived Column is a delayed copy of the SM column (#10), and is not "created" by Q&A until way out at column 200. It's made invisible so it doesn't appear in the report.

The third derived column is a delayed copy of the "Item" column, where Q&A will count the line numbers. This column is also invisible.

The middle Derived Column is the needed line numbering column. We assign it Column #25 so it appears in the report between #20 (Salesman) and #30 (Style). The @Text(n,m) function, according to Q&A's Help, "*Produces a text value consisting of n characters of x*." Here, however, we use @Text to perform a conditional calculation. The first part, *#10 <> #200*, is a question or "If" statement: "*Is #10 not equal to #200?*" If True, it returns the value "1"; if False, the value "0." When the SM changes value, and Q&A creates column 25, #10 for this line will be a new value. Since we haven't gotten to column 200 yet for this line, it still retains the value it had from the previous line of the report: column #10 does not equal column #200, so column 25 = 1—the first Item of this group.

What happens on the second and succeeding items in a group? Here, #10 of the new line would be equal to #200

of the previous record, so the first part of the formula returns a zero (0). What about the second part of the formula? It works in a similar fashion, but the question is, "*Does column #10 equal column #200?*" If True, you get the delayed copy of the line number from the previous record (column #190) plus 1; if False (when the SM changes), you get zero.

In Q&A for Windows, if you create this report in Q&A for DOS, it will run (as a Freeform Report) in QAWIN and turn out okay. However, if you try to create it as a Q&A for Windows Columnar report, it will not work—the Item column will just say "1" on each line). This is because Q&A for Windows does not calculate columns sequentially left-to-right like Q&A for DOS does.

Read-Only Records—Only for Some

I use Q&A 5.0 and want to be able to set up a database that certain people can review but not modify. I would love to be able to do this without resorting to assigning passwords to every user. Is there any way?

<div align="right">Betty</div>

As you allude to, the easiest way to do this is by assigning different levels of access rights to different people. The ones you want to restrict would have their access rights set to "No" in every category. That way they could see all the records but could not add to or modify them.

What you are asking for is a little tricky but can be done. It will require two different macro files and a few passwords. (But don't worry, these will be invisible to the users). One macro is for users with full access and one is for users with read-only access. The assumption I am making here is that users don't change computers. The secret is that you will use *Custom Menus* to replace some of the standard Q&A menus.

First, you will set just two User ID/Password combinations on the database in question. *Full/Full* and *None/None* (it's obvious which is which). Make sure to set the XLookup password in any other database that uses this one to *Full*.

Next, define a new Custom Menu by pressing *Shift-F2* and then *Create Menu*. Select *New Menu* and press Alt-F7 at the menu name. Choose *Q&A Main Menu*. The regular Q&A Main Menu has the choices:

```
F - File
R - Report
W - Write
A - Assistant
U - Utilities
X - Exit Q&A
```

Your new Custom Main Menu will be:

```
Item 1: Access the Main Menu       Macro Name: <altq>
Item 2: ........................    Macro Name:
Item 3: ....... Press Enter ....... Macro Name: F
Item 4: ...... To Access The ...... Macro Name: R
Item 5: ....... System Menu ....... Macro Name: A
Item 6: ........................    Macro Name: U
```

```
Item 7: ........................    Macro Name: X
Item 8:                             Macro Name:
Item 9:                             Macro Name:
```

Then create another custom menu called *Q&A Main Menu 2* which looks like:

```
        Menu Title...: Q&A Main Menu

Item 1: F - File              Macro Name: F
Item 2: R - Report            Macro Name: R
Item 3: W - Write             Macro Name: W
Item 4: A - Assistant         Macro Name: A
Item 5: U - Utilities         Macro Name: U
Item 6: X - Exit Q&A          Macro Name: X
Item 7:                       Macro Name:
Item 8:                       Macro Name:
Item 9:                       Macro Name:
```

The only additional macro you will need is Alt-Q. It will first clear and then set the correct passwords and then put the user in a new Main Menu. The Alt-Q macro will be different for the different users. For those with full access the macro is:

```
<begdef><altq><name>"<caps,>altq<caps.>"<vidon>
<capsf6><f6>full<enter>full<enter><menu>"*Q&A<sp>
Main<sp>Menu<sp>2"<enddef>
```

For those with restricted access, the macro is:

```
<begdef><altq><name>"<caps,>altq<caps.>"<vidon>
<capsf6><f6>none<enter>none<enter><menu>"*Q&A<sp>
Main<sp>Menu<sp>2"<enddef>
```

Due to the way custom menus work, you have to keep the fake Main menu with all the choices so that the macros will work. You also have to make the users put up with that extra keystroke all the time in order to keep the macros running from the right place. Just create a qamacro.asc file for each type of user, put it in their Q&A program directory, and go back to work.

Expanding Fields in Q&A 5.0 Report Design

I seem to have a problem that just drives me crazy with Q&A 5.0's Report module. I don't think I'm doing anything wrong, but I just can't seem to get the expanded field to work properly. Sometimes it gives me a full expanded window and sometimes I just get the cursor at the bottom of the screen and I have to type along a single ever-expanding line. I don't know where the end is and I have problems seeing what I have already typed. Am I missing a setting?

<div align="right">Nate</div>

This is not your mistake or a bad setting. It is a design deficiency in Q&A. In the report writer, in Design/Redesign mode, the F6 expand field key will only give you the ability to type on an expandable line along the bottom of the screen. However, in the Print / Make temporary changes mode, F6 in the same field gives you a full expanded window in which to work. This seems

Programming in *Sesame*

Alec Mulvey
with Bill Halpern

SESAME's developers set out to make programming in *Sesame* more powerful than Q&A, more consistent than Q&A, but as easy as Q&A. At the same time, *Sesame's* programming also needs to be *compatible* with Q&A, so that a Q&A database can be translated into a *Sesame* database and its programming will work. This has all been achieved. The compatibility is pretty much there, insofar as the Q&A programming is read in by *Sesame*, converted where required by the translation process, and will then work as *Sesame* programming, remaining viewable and readable by a Q&A programmer. There are some exceptions, some alterations that will need to be made, but by and large the process is automatic.

Overview of programming in Q&A

Before looking at how you go about programming in *Sesame*, let's briefly review programming-related facilities and tasks in Q&A.

In Q&A, you have one Program Spec. You can put programming in one or more fields—so you have one programming "slot" per field. Each of these "slots" (program locations) can contain:

- One or more on-field entry programming statements *or*
- One or more on-field-exit programming statements *or*
- One or more general programming statements

One of these fields can serve as the sole on-record-entry programming field, and one of them can be used as the sole on-record-exit programming field.

By "general programming statements," I mean ones that don't start with an "<" (on-field-entry execution) or a ">" (on-field-exit execution) as in the following examples:

```
#810:  #640 = #600 + #610 + #620 + #630
#810:  Inv Total = @Sum(ItemVal1..ItemVal4)
```

Q&A, as you know, will accept Field *Names* or Logical Field *Numbers* (LFN's).

Lastly, the operation of general programming statements can be altered by a couple of settings in Q&A which are somewhat obscure though vital. These are the programming options for *Automatic* or *Manual* Calculation mode and *Main program before field exit*, accessed not via the Program Spec (which is where they rightfully belong), but by pressing Shift-F8 when viewing any record in Search/Update or Add Data mode. (See Figure 1.)

If Calculation Mode is set to "Manual" (Q&A's default), then the only time general programming statements are executed is when you press the F8 (Calc)

key while viewing a record. If Calculation Mode is set to "Automatic," then general programming statements are triggered automatically every time the data in *any* field is changed.

I have found that many, or most, Q&A programmers use on-field-entry and on-field-exit programming exclusively. I believe that the reason is because they are unfamiliar with (or forget about) this programming options dialog where they can change the calculation mode from Q&A's default—which was perhaps sensible in the days of slow computers when calculations took a long time—to Automatic. I have never seen a need for Manual Calculation mode.

Main Program before field exit allows you to control just that: whether field exit programming gets calculated before any general programming statements in the "Main Program."

To illustrate how this works, suppose you have a field for **Style**, which might be a number such as 1234 or 6543, and a field for **Size**, which can be S, M, L or XL. The Product Code is a combination of these. Let's say you have the following data:

```
Style: 1234
Size:   M
Product Code:  1234M
```

And the following programming:

```
Style:
Size:  > Product Code = Style + Size
Product Code: #30:
IF Size = "S" THEN Size = "Small";
IF Size = "M" THEN Size = "Medium";
IF Size = "L" THEN Size = "Large";
IF Size = "XL" THEN Size = "Extra Large"
```

Now, with Calc mode set to Automatic, this programming will work fine if Main Program Before Field Exit is set to "NO." In this case the on-field-exit program


```
Calculation mode..............:  _Automatic    Manual

Main program before field exit?   Yes   No
```
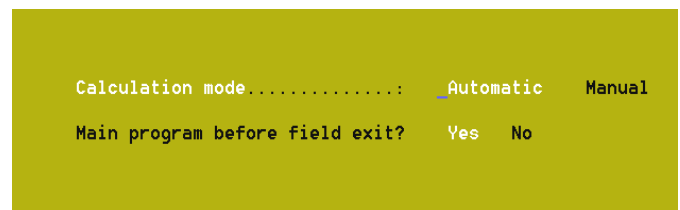
Figure 1. The programming options dialog. Pressing Shift-F8 from any record will display this screen.

calculates the Product code correctly, using just the single letter entered for the size— 1234M. But if this is set to "YES," then the general programming statement in field #30 is calculated first, and the Product Code becomes "1234Medium." Incidentally, if Calc mode is set to "Manual," then you will never see the full size text ("Small," and so forth), without pressing F8.

General Programming Statements must start with a Logical Field Number (for example, "#810:"). The LFN controls the order of execution of the statements.

You can also put programming in the Navigation Spec for the database. Virtually all programming is allowed, as long as it does not set a field to a value by using an "=" sign. The Navigation Spec and the Program Spec are quite independent. You can use another completely different set of LFN's in the Navigation Spec.

If you "back out of" a field with on-field-exit programming in it—using Shift-Tab, for example—then that programming is not activated.

When you perform XLookups, you need only specify the database name if it is in the default data location. If it is not, then you need to specify the path to the external database.

What does Q&A for Windows add?

Q&A 4.0 for Windows adds multiple forms capabilities. One database can have many forms. Each form can show any field from a database once. The "Master Form" for the database contains *all* the fields. All forms share the same Program Spec. However, each form has its own Navigation Spec.

And *Sesame*?

In *Sesame*, programming is *form-based*. You can have multiple forms, and each form has its own programming.

You don't use LFN's. *Sesame* converts these to field names for you on translation and you use field names exclusively thereafter.

When you program, you program a *copy* of the *Sesame* application, not the original. You then apply the changes to the original by a process called *Reconciling*.

In *Sesame* you have more programming "slots."

You have what's called *Global Programming* as well.

You have many more *events* you can use to trigger programming. One field can have programming for more than one event.

Sesame offers consistent capturing of events.

Sesame supports multiple databases within a single application.

Form-based programming

Imagine having multiple saved Program Specs that you can switch between, rather than just one! Well that is exactly what *Sesame* provides. You create multiple forms, each of which can be task-related. For example, you can have one form for adding a new employee containing most of the details; one for adding supplementary details

for an existing employee at a later time; another to cover the vacation history for an employee; and yet another for job appraisals or the like. Each can have its own design and programming relevant to the tasks it performs. There's no need to skip over unwanted fields or deal with complicated navigation.

The Designer file

When you program a *Sesame* database, you work with a Designer file (.DSR). This is a copy of your complete application, including the data at the time the DSR file was created. This way, you can make changes and test them while others are working on the live data. You can then reconcile the changes when you've finished your work.

Additional events

In Q&A, there are five events to which programming can be attached:

- On-record-entry
- On-field-entry
- General Programming ("on-record-change")
- On-field-exit
- On-record-exit

These are all supported in *Sesame*, but in addition you have *on-field-change* and *on-field-immediate-change*.

On field change programming is triggered only if the field has been modified. In Q&A you can detect, directly, only if *any* field has changed. Even Q&A 5.0's @Modified function refers to the record as a whole, not a particular field. The only way to detect a field change in Q&A is to use a "mirror" field or perform an XLookup to the same database record to compare the stored value with the current value in the field.

Sesame's on field immediate change is more sensitive still. It detects every change to the field *character by character*. So you can actually capture each keystroke and perform validations as the user is typing.

More programming "slots"

In Q&A there is one programming storage location per field. This can be attached to any of the five events mentioned earlier. In *Sesame,* you have one storage "slot" for each field on the form for *each type of event*. You'll see this more clearly in the examples that follow.

Multiple events in one field

Sesame allows you to have both on-field-entry *and* on-field-exit programming in one field. For that matter, you can have general programming in that field as well!

Consistent capturing of events.

When you move out of a field in *Sesame* that has on-field-exit programming, then it will *always* be triggered, no matter how you leave that field.

Let's now have a look at just how you go about programming in *Sesame.*

The Programming Editor
In *Sesame*, all programming is done in the Form Designer, a separate module from *Sesame* but which can be accessed from within the main *Sesame* screen. You open an application (*.DB) or an existing designer file (*.DSR) in the Form Designer, choose the form design function, and select Program Layout from the Commands menu, as shown in Figure 2.

The Program Editor opens in a separate window. (See Figure 3.) The editor window can be moved so that you can see the form design below it. On-field-entry and on-field-exit programming is attached to the relevant field and event, but general programming can be in any field or fields. In fact, all the general programming can be in one field. This is like Q&A, but in practice users seem to like putting the programming for a field *into* that field.

I say "fields," but actually it is not fields that you are programming. Rather, it's "Layout Elements," which are "boxes" or "containers" on the form to display data. Remember earlier where I said that in *Sesame* you program *forms*, not the database? Well a consequence of this is that programming is not attached to fields. If it were, then a field on two forms would have the same programming. In *Sesame*, the data containers on *each* form are optionally programmed. And again, these are called "layout elements." This distinction is necessary in a graphical (GUI) environment because you can have layout elements that don't coincide with *fields* in the traditional Q&A sense. You might have a dropdown selection list on the form, a series of radio buttons, or a display box for calculation results that is not tied (bound) to the database. (In other words, it is not a field). And you might want to include programming for any of these elements, which you *can* in *Sesame.*

The Program Editor is comprised of four distinct areas:

- Menu Bar
- Toolbar
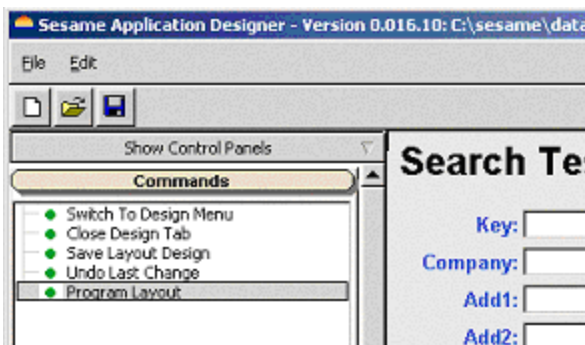- Programming Window
- Error Message Window

The Menu Bar gives you the ability to:

- Save your work and close the editor (Editor).
- Perform standard windows operations like cut, copy delete, undo (Edit).
- Find, search, and replace (Search).
- Test programming (Test).

On the toolbar there are three drop-down selectors:

- Elements
- Events
- Program Navigator

plus a line indicator.

The Element Selector drops down to let you select the layout element you wish to program. You then use the Event Selector to choose when the programming will be activated:

- On Field Entry
- On Field Exit
- On Form Change
- On Field Change
- On Form Entry
- On Form Exit
- On Field Immediate Change

Programming in any field can have any single or combination of triggering events. Each event is programmed in its own window. Imagine an array of layout elements/events with each combination being an available programming location. You only get to see programming in one of these locations at a time. Figure 4 shows an example of a programmed field from a translated Q&A database:

The Program Navigator lists, and allows selection of, all currently saved programs, the layout element they are associated with, and their activation event. (See Figure 5.) When a selected element has more than one activation
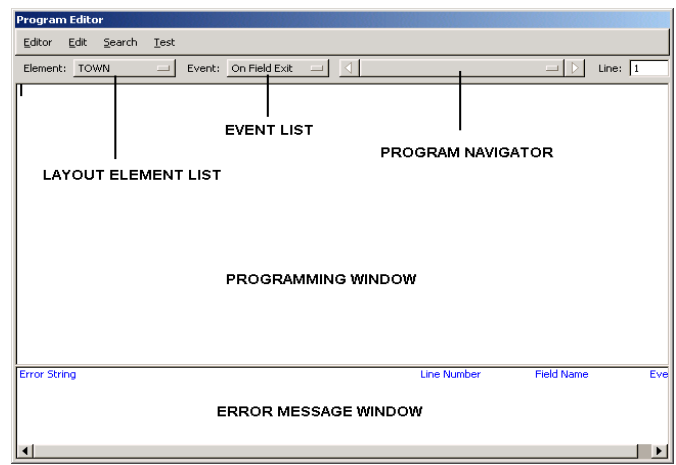

Figure 2. Selecting Program Layout in *Sesame* Designer

Figure 3. Sesame's Program Editor window

event, the right and left arrows on the Navigator let you scroll through the various event programming options. This way you get a list of just the programming locations in use.

In addition to these layout element programming locations, you also have "Form" and "Global Code." Form programming is just what it says—it's programming that applies to the whole form. In Q&A this still has to be put in a field. See the example shown in Figure 5. The programming is stored in "Form," and the event is "On Form Exit." This has in fact been translated directly from a Q&A database where it was in a field—an arbitrary field—which was set as the "On Record Exit Field" (F8 at the Program Spec).

Global Code is for programming which needs to extend beyond the scope of a single record. This might be, for example, running totals accumulated across records, or a flag that records the fact that a user has clicked on a particular button.

Using the Programming window

The programming window is a free-form text editor where you type your programming commands. Since *Sesame* programming, besides supporting the semicolon-delimited syntax of Q&A, also follows the rules of line position where you can structure programming statements by placing them on separate lines, the editor does not automatically wrap lines. You can keep your programming easy to read by using a structured tab indenting scheme on lines, like this:

```
If [condition] Then
    [statement]
    [statement]

Else
    [statement]
    [statement]

[New block of code]
```

You don't need a semicolon at the end of each statement—or anywhere.

When you've finished with any or all of your programming, you can click on the *Test* option on the Menu Bar to test it. In the error message window at the bottom of the editor, *Sesame* displays error messages for any statement with one or more syntax errors. (See Figure 4). Unlike Q&A, a programming syntax error in *Sesame* does *not* preclude saving the program. You can save your work—bugs and all—and close the editor, then return to it later for further work.

Any programming in *Sesame* is automatically saved when you exit or close the editor, whether via the menu choice or by simply pressing the Esc key.

To check the successful operation of programming after saving it, you can select "Preview Application" from the Designer menu. This opens the designer file in a fresh session of *Sesame*. You now have your data to use as test data—you can add records, delete records, and perform almost all tasks in this preview mode.

When you're ready to actually apply the changes, back in Designer you select "Reconcile Application" from the Designer menu. You select your application and confirm. *Sesame* intelligently finds all changes made between the Designer (.DSR) file and the *Sesame* (.DB) file, and applies them. The data in the *Sesame* application is not touched. The data in the DSR file stays there as sample data. This reconciliation process is used to apply all design changes, not just programming. These can even be structural changes, such as adding or deleting fields. Reconcile handles them all automatically.

Alec Mulvey is the owner of Keyword Software in the UK and a principal of Lantica Software, LLC. alec@keywordsoftware.com, www.keywordsoftware.com. Bill Halpern is co-owner of Professional Computer Technology Associates (PCTA) in Newtown, Pennsylvania and is President and CEO of Lantica Software, LLC. . bill@pcta-usa.com, www.pcta-usa.com.
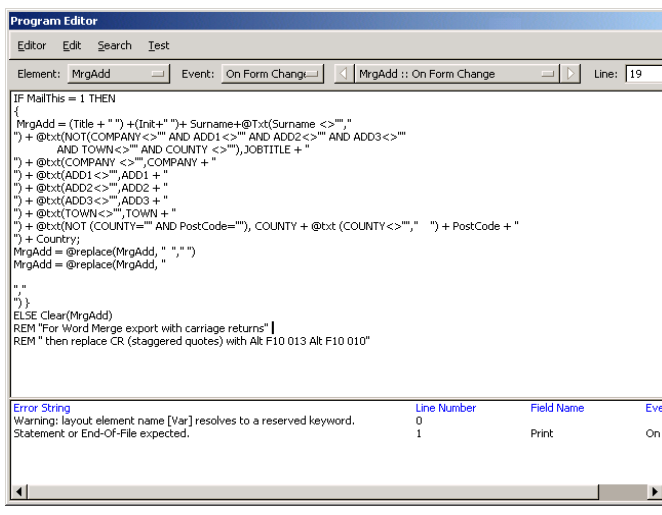
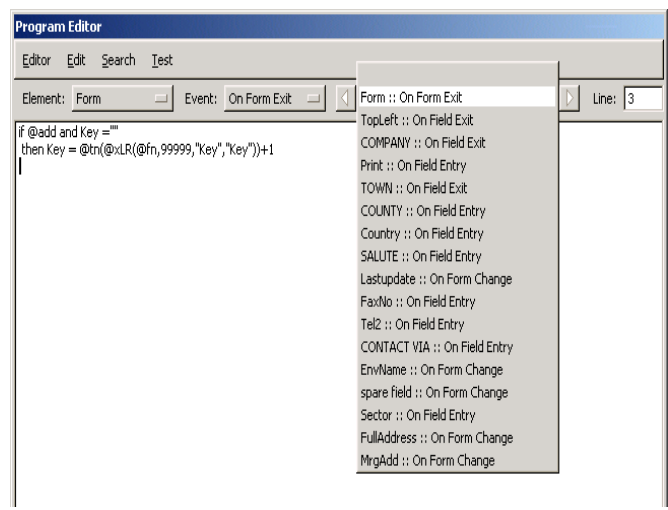**Figure 4**. Programming for "On-Form-Change" in the **MrgAdd** field.

Figure 5. Program Navigator listing all programming for the form.

@Help...

backwards or just plain wrong, but we can't change it.

Here's a pretty good workaround for Q&A 5.0. Use either the Write word processor or the temporary changes print mode of the report module to type your lengthy text. Then use F11 to highlight it and copy it to the clipboard. Following that, go into Design/Redesign, press F6 so you are on the long line at the bottom, and press F12 to paste the copied text into the field. No more scrolling left and right or worrying about what you missed.

A Preliminary Note for Novell Users

We have been running into some very disturbing developments with Q&A 4.0 and 5.0 running on Novell networks where the data is being kept on a Novell server volume. It seems that the problems occur with Novell version 5 or 6. They appear as lost records or frequent and continuous (though intermittent) database corruption. Preliminary indications are that they also involve the way Novell partitions the volumes and the way it caches disk writes. Almost all types of corruption are related to network read/write errors. We have been able to temporarily solve these issues by moving the data from a Novell volume and placing it on an external drive which is not Novell formatted or controlled. More information and analysis is going on and we will keep you informed as we find out more.

Q&A Win and Large Hard Drives

In last month's *Quick Answer*, John Dow stated: "*By the way, I am convinced that Q&A for Windows doesn't have any problem with large drives and that repartitioning a large hard drive is not necessary.*"

As much as I respect John's knowledge and abilities, I have to put my two cents in here. I have discussed this very issue in great detail with the people who did technical support for Q&A for Windows at Symantec. They are adamant that Q&A for Windows is extremely sensitive to large hard drives and will corrupt databases used in partitions larger than 2GB. In addition, I have had some very personal experiences with totally trashing brand new Q&A 4.0 for Windows databases by simply adding a single record on a 40 GB drive. I also repair databases for users whose only wrongdoing seems to be that they were running on large hard drives. We fix one or two of these per week. Once we get them repaired and back to the client on a partitioned drive, the problems seem to go away.

So, I must counter with this: *I am convinced that Q&A for Windows* does *have a severe problem with large drives and that partitioning a large hard drive is absolutely necessary*.

You cannot use Lesspace with Q&A for Windows. You have to partition the drive.

Bill Halpern is co-owner of Professional Computer Technology Associates (PCTA) in Newtown, Pennsylvania and is President and CEO of Lantica Software, LLC. 215-598-8440. bill@pcta-usa.com, www.pcta-usa.com.

Huge Data Files

When I read through the first draft of Michael's article, I thought, "why didn't he try using CHOPTEXT (available from the Free Downloads page at www.quickanswer.com) to split that huge Fixed ASCII file into more manageable pieces"? (In a subsequent conversation, Michael indicated that he wasn't familiar with the utility.) As it turned out, though, CHOPTEXT probably wouldn't have done much, if anything, for the time *problem. Chopping up the ASCII file and*

importing the resulting multiple files into his MASTER database would have been tanamount to importing the entire ASCII file in one days-long gulp. Importing the chopped files into multiple *databases would have speeded up the process, but defeated the purpose of having a single MASTER database, with the ability to readily extract, from a single source, groups of records based on whatever selection criteria were required, as in the example given in the article. —Ed.*