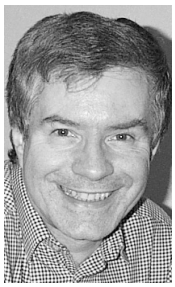
How to Avoid Common Programming Traps



ALEC MULVEY

The Quick Answer
The Do-It-Yourself Guide to Q&A
November 2001
Volume 12 Number 11

MARBLE PUBLICATIONS

WHEN I work with databases created by others, I often see opportunities to make the database do more of the work, program it to be less cumbersome or operate faster, or simply make the programming neater and more manageable. This is natural since I've been doing it longer. There is an array of programming "flaws" that I keep encountering, particularly with databases that have been programmed by the manager of the business. I've a good idea why this is. It's not their fault—they've just been taking the lead from the Q&A documentation and help screens. Just like the defaults when you install Q&A from the original disks, the manuals and help screens didn't spell out the best practice at the outset and were never updated to correct these issues.

This is not a tutorial on "Good Programming." I'd just like to cover the main issues as I see them and show you ways that I consider better. In any case, now would be a good time to "clean up" your programming and get it in good shape for converting to "*Sesame*" when the time comes.

Alec's basic tests for faulty programming

I have two rudimentary tests to check for obvious programming flaws. I reckon I should be able to go into any Q&A database and:

1. Tab through all the fields in any order. Nothing should change.

2. Press F8 (Manual Recalc) repeatedly without anything changing or untoward happening. No data should change.*

** An exception to this rule is data that is date- or time-sensitive, such as a **Days Overdue** field that calculates the age of an invoice based on the system date.*

Problem #1 is usually seen because the following rule has been broken:

All data directly dependent upon an entry in one field should be calculated immediately upon leaving that field.

Let's take the common task of looking up some customer data. Your customers have unique customer ID's. You type the customer ID (for example, "JONESB" or "12884"), or select it from a picklist of some sort. What happens then? Well, so often I see the following:

```
Customer Ref:
Customer: <#2: XLu("F:\Data\Customer.dtf", #1, "X#1", "X#2", #2)
Address1: <#3: XLu("F:\Data\Customer.dtf", #1, "X#1", "X#3", #3)
```

And so on.

What's wrong with this?

Well several things, starting with the more minor flaws:

- The XLookup database pathname should not be typed in full. It should be in the default data directory (Global Options). Then, instead of "*F:\QA5DATA\CUSTOMER.dtf*" all you need is "*CUSTOMER*". (You don't need the .dtf filename extension.)

- Logical Field ID numbers are being used instead of field names. Many people get turned off from field names because they think it makes their programming run slower or because at some stage Q&A rejected a field

Continues on page 3

Sesame Seeds

Sesame ODBC and Export Options

Continual progress has been the keyword with *Sesame* development at Lantica Software. You probably received our latest email on the coming beta testing cycle. We are now in the process of assembling a team of qualified beta testers (from among several hundred registered beta tester applicants) that will take *Sesame* through its paces and give us the feedback we need to complete the product development cycle.

I'd like to bring you up to date on the latest developments.

ODBC Compliance

Our *Sesame* ODBC (Open Database Connectivity) has been developed to the point that Microsoft Word can now read a *Sesame* database! This means that you will be able to merge your database information using *Sesame's* own merge facilities or from inside Microsoft Office products.

An important component of a good database manager is the product's ability to share its data across platforms and programs. ODBC is a major milestone in accomplishing that.

Enhanced Export Options

Another important component is the ability to export data in a format that is usable by other products and systems. The standard is to be able to easily export database information in a variety of widely-accepted ASCII formats. Q&A is competent in this regard. *Sesame* will be much, much better!

We have just implemented the export engine and I want you to be the first to see it. Q&A does its ASCII export through a series of menu choices and field selections.

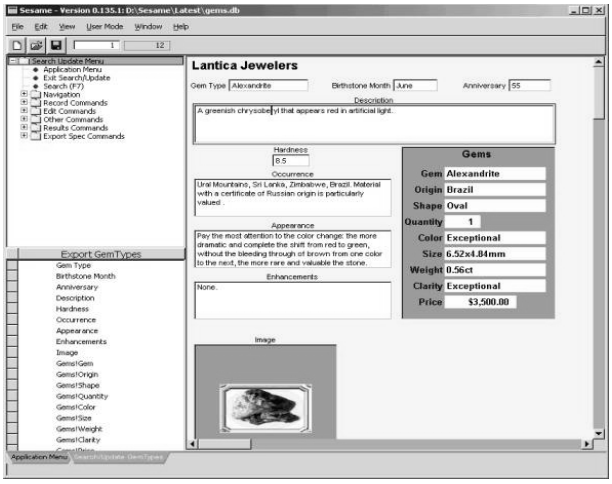


Figure 1. Exportable fields shown in the lower left hand corner of screen.

Sesame will also do that, but we have built the feature right into the main database view! As Figure 1 shows, you can see all the exportable fields in the lower left hand corner of the screen. This includes not only the fields in the main form, but also the sub-form fields (if any).

By simply clicking on the fields to export, you can set the order of the fields to include in the export file as well as specify, for each selected field, a "padded" fixed length (which is very useful for number fields that need to be front-filled with zeros), or a truncated length, where you only want to export a specific number of characters from a lengthy text field. See Figure 2 on page 11.

Note that you can select not only fields in the primary record, but also those in the sub-records. We have decided to treat each combination of primary

Continues on page 11

The Quick Answer™

Editor / Publisher
Tom Marcellus

The Quick Answer (ISSN 1052-3820) is published monthly (12 times per year) by Marble Publications, Inc., 1927A Harbor Blvd., Costa Mesa, CA 92627 USA

Cost of domestic subscriptions: 12 issues, $109; Outside the U.S., $129; Online edition, $79. Single copy price: $11; outside the U.S., $12.50. All funds must be in U.S. currency. Back issues are available upon request for the same price as a single copy.

POSTMASTER: Send address changes to The Quick Answer, 1927A Harbor Blvd., Costa Mesa, CA 92627.

Address editorial correspondence, @HELP questions, or requests for special permission to: Marble Publications, Inc., The Quick Answer, 1927A Harbor Blvd., Costa Mesa, CA 92627 Phone 800-780-5474 or 949-722-9127. Fax 949-722-9127, mailbox@quickanswer.com

On the Web at www.quickanswer.com

Reach Us

Phone 800-780-5474 / 949-722-9127
Fax 949-722-9127
Email mailbox@quickanswer.com
Web http://www.quickanswer.com
Mail Marble Publications
The Quick Answer
1927A Harbor Blvd.
Costa Mesa, CA 92627 USA

Letters

Just a short note to say how very helpful was your @Help column in the June 2001 issue dealing with Q&A slowdowns caused by Norton Antivirus software. The suggestion mainly helped me with Norton's effect on other programs, namely Quicken. Dramatic improvement. Many thanks.

—Peter Scott, Nine Two Five Chartering, Inc.
Brooklyn, NY, ntf@ninetwo5.com

Programming Traps... cont'd from page 1

name at the Program Spec. (This would be because the field name used is invalid for programming. See my article "All about Field Names" in the February 1999 issue of *The Quick Answer*.) Just stick to plain ol' letters and numbers for field names—but don't begin a field name with a number—and you'll be OK.

- The field numbers are incrementing by ones. If you must use field numbers, you're much better off incrementing them by tens, such as #80, #90, and so forth. This way you can insert fields between #80 and #90 without having to renumber them all.

But there are more significant flaws.

Strategic Flaw #1

The XLookups are in separate fields. This means that the **Address1** field is only filled on entering it. The user then has to move to the **Address2** field to look that up and fill it, and so on. Better would be to put a *Goto* command at the end of the field, so that Q&A is instructed, on entering the **Address1** field, to look up the value for that field, then move to the Address2 field, where that value is looked up, and so on. But far better still is to do all the XLookups in one go. After all, once you've specified a customer via a key value reference (whether you've typed it or selected it from a list), that's everything Q&A needs to know to perform all the XLookups for that customer. You want *all* the data relevant to that customer to be retrieved straightaway. You do this by entering all the XLookup (or Lookup table) programming into the Customer field as an *on-field-exit* statement or series of statements. So the programming for the Customer field becomes:

```
Customer Ref: >#10: XLookup("CUSTOMER", Customer
Ref, "Customer Ref", "Customer", Customer)
Customer:
Address1:
```

This statement can then be extended to auto-fill the other customer-related fields. The programming, though, remains in the **Customer Ref** field. There are two ways to do this. One way is to simply repeat the XLookup for each field, separating the programming statements with semicolons, like this:

```
Customer Ref:
>#10: XLookup("CUSTOMER", Customer Ref, "Customer
Ref", "Customer", Customer);
XLookup("CUSTOMER", Customer Ref, "Customer Ref",
"Address1", Address1);
XLookup("CUSTOMER", Customer Ref, "Customer Ref",
"Address2", Address2);
```

And so on.

Alternatively, you can chain together multiple lookups in a single XLookup command. After the initial XLookup, instead of closing the parentheses and writing the next XLookup statement, you can type another comma and then repeat the last two parameters—the external lookup field and the internal target field, like this (see the sidebar):

```
>#10: XLookup("CUSTOMER", Customer Ref, "Customer
Ref", "Customer", Customer,
      "Address1", Address1,
      "Address2", Address2,
       "City", City)
```

Strategic Flaw #2

The data is susceptible to becoming invalid on subsequent edit.

In the examples above I've used the XLookup *statement*. You can alternatively use the @XLookup *Function*.

They both seem to do the same thing—that is, look up some information in an external database based on matching fields. So what's the difference?

Well the syntax is a little different. To use the earlier example, for the first two fields, you would have the following:

```
XLookup Statement
>#10: XLookup("CUSTOMER", Customer Ref, "Customer
Ref", "Customer", Customer,
      "Address1", Address1)

@XLookup Function
> Customer = @XLookup("CUSTOMER", Customer Ref,
"Customer Ref", "Customer");
 Address1 = @XLookup("CUSTOMER", Customer Ref,
"Customer Ref",
"Address1")
```

The first example (where XLookup is used) can be thought of as saying to Q&A:

Multiple @XLookups Might be Faster

Under some conditions, over networks particularly, using a single XLookup with multiple lookup and target field parameters might be slower than using multiple @XLookup commands (note the "@"). In his tip, "Get Faster XLookups Using a Q&A Quirk" in the November 1996 issue, Gordon Meigs reported that the multiple @XLookup technique was 30-percent faster over a network than using a single XLookup command with multiple external/internal target field parameters. For applications with lots of external file lookups, you should conduct your own tests to determine any performance variations between the two techniques. The single XLookup command is certainly more economical from a programming standpoint and much easier to write.

"Go look this up: Match this field in this database with that field in the external database. Retrieve the data from the specified field in that external record, and put it in this target field."

The @XLookup command says pretty much the same thing, except that you are telling Q&A, *in advance of the lookup*, where to put the retrieved data.

Just a subtle difference at first glance. But there are actually three substantial differences.

First of all, you can't "chain" the @XLookup function commands—you have to say that the one field should get the value from the lookup in a single statement.

The second difference is that the data returned by the @XLookup command is a temporary variable that you can do something with *before* placing it in the target field. Consider this statement:

```
Discounted Amount = @XLookup("Prices", Item,
"Item", "Price") * .9
```

Here, the retrieved *Price* value is a variable in memory, so you can multiply it by .9 (to give a 10-percent discount) and put the resulting value (90, if the Price is 100) in the **Discounted Amount** field.

To do this with an XLookup command, you would have to put the Price value in the target field *before* performing any calculations on it. Here's another example:

```
If  @XLookup("Prices", Item, "Item", "Price") > 100
Then […what to do with the result]
```

You see how the @XLookup enables you to do conditionals without having to first copy the value into a field.

The third major difference is in what happens if an XLookup fails to find a match.

To quote the Q&A *Application Programming Tools Manual*:

"If the XLookup doesn't find a match, there is no change to the destination field."

vs. this:

"If @XLookup doesn't find a match, the @XLookup function returns a blank or null value."

This difference is huge. I think the @XLookup function is much better in this case, for if you enter a *wrong* Customer reference, you see the results immediately—the target fields are blanked. In contrast, if you use the XLookup Statement, you can get invalid data. How? Well, suppose a user enters a customer reference that correctly looks up and retrieves the customer information. Some time later another, or the same, user goes to this record and, for some reason, changes the customer reference to one that does not exist. If the XLookup statement is used, then the lookup will fail and the original address data will stay there—but now with the *wrong* Customer reference. This is likely to be most undesirable! With the @XLookup function, all the lookup fields are cleared if an incorrect reference is entered. Far more satisfactory.

When you have all the programming for the customer address in the **Customer Ref** field, then you should make all the lookup fields (Customer, Address1, Address2 etc) Read-only. On leaving the **Customer Ref** field, then, the data is all looked up and the cursor moves to the first field beyond the customer details.

By programming in this way, you can safely tab through the only field that accepts the cursor —the **Customer Ref** field, in this case—and all that will happen is that the XLookups will be performed again. If the customer details are unchanged, the data in the dependent fields will not change.

I will make the general point here that some data you might *want* to change, such as the telephone number of a customer if it subsequently changes. But often you won't—especially things like prices in invoices. But this topic of "freezing" the data—when to do it, when not to do it, and how to do it—is a topic for another day.

Rule #2: Press F8 (recalculate) and nothing untoward should happen.

This problem can be caused by one or both things:

- Program Recalc Mode is set to *Manual*.
- Programming statements are invoked in the incorrect order.

Problems caused by Manual Calc Mode

If I'm going to use Calc Mode at all, I always set it to *Automatic*. Always. To me, Manual Mode is like setting:

Do you want your data to be correct only some of the time? YES No

When you set Calc Mode to Manual, then general programming (programming that is *not* specifically set to execute on-field-entry or on-field-exit) is calculated *only* when you press the F8 (Calc) key. Until then, you're likely to be viewing incorrect data.

Here's a general programming statement:

```
#320: Total = @Sum(Item1..Item8)
```

If the database is set to Manual Calc Mode, the **Total** field will be incorrect no matter what you put in the Item fields *until* you press F8. And if you forget to press F8 before saving or resaving the record, the **Total** field will *remain* incorrect.

With program Calc Mode set to Automatic, on the other hand, all *general* programming statements (those not specifically set to execute only on-field-entry or on-field-exit) are calculated *every time you change any field and leave that field*. (See the sidebar.)

I do come across databases programmed expecting the F8 key to be used to calculate. These need to be changed. What if a user forgets to press F8? Or presses it at the wrong time? Or presses it more than once?

You set the Calc Mode by pressing Shift-F8 from any record in Add Data or Search / Update mode. Set the mode and press F10. (See Figure 1.)

Figure 1. The Calc Mode dialog box.

Manual Calc Mode is a throwback to the days when we had PCs, hard disks and networks that were extremely slow by today's standards. (Run your heavily-programmed database on a 286 PC and you'll see what I mean.) On today's lightning fast PC's, Manual Mode is an anachronism—*and* a liability—perhaps with one exception.

If the database contains no Auto Calc programming, you could use a Manual Calc program, for example, to pop-up a pick-list for a certain field from anywhere in the record.

Three Programming Execution Modes

In Q&A, your programming statement(s) can be:

General programming:
```
#30 : Tax = Tax Rate * Invoice Total
```

or on-field-entry (Userselect lists, for example):
```
<#60 = @UserSelect(" Small, Large, Medium")
```

or on-field-exit (Look up customer details from a customer reference code as in the examples above)

You also have *on-record-entry* and *on-record-exit* programming execution options, but we're not concerned with those here.

With "*Sesame*," you will be able to use all three types of programming options *in the same field* if you like!

Suppose you have a **Status** field that frequently needs updating. When a customer or vendor, say, earns a new status, you don't want to have to tab through a host of fields to get to the **Status** field. Instead, you could use a Manual Calc program in the **Status** field that pops-up a pick-list (Q&A 5.0 only) of Status Codes. This way, you could press F8 from *anywhere* in the record and your Status Code list would display for you to update the Status field.

Problems caused by programming calculated in incorrect order

Look at the example shown in Figure 2.

The programming is calculated in the order of the Logical Field Numbers (LFNs). Notice in this example that Field #90 relies on a total in Field #100. But because #90 is calculated *before* #100, the Tax calculation will usually be wrong. After entering or changing one item the Tax will change, yes, but it will be one calculation behind. Tellingly, if you press F8 to force a new recalculation, the tax will change again. This is a warning sign that all is not well in the programming of this database. One result when pressing F8 once and a different one when pressing F8 a second time is a sure sign of this incorrect-calculation-order problem.

This typically occurs on more complex forms, where fields on one page refer to fields on earlier pages and where the programmer has naturally tried to number the fields from "top" to "bottom." It helps if you can restrict how many fields get a LFN. By concentrating programming in fewer fields you can reduce how complicated your numbering scheme has to be. The only fields that need to be numbered are those that have a general programming statement in them. Fields with on-field-entry and on-field-exit programming do not need to be numbered. You can refer to other fields by name. This avoids these "programming order" problems.

Lastly, don't make all your programming of one type. In a well-crafted database there will likely be the need for both on-field-entry and on-field-exit programming and also general programming. Take this example:

```
Date 1:
Date 2:    No of Days:

Rate/Day:     Rental charge:
```

Concludes on page 12

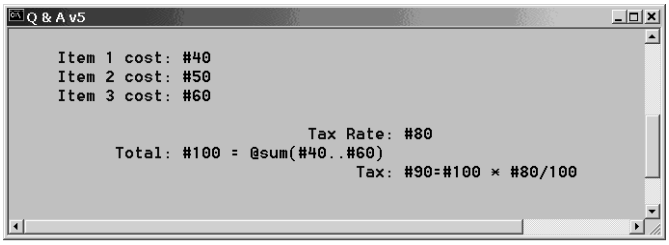


Figure 2. Incorrect program order.

@Help

EDITED BY WILLIAM HALPERN

Security for Reports

I have a client who wants to set up reports in Q&A 4.0 for DOS in such a way that people who can input and edit records cannot print/produce/create/generate reports. (This is to prevent people from printing lists of info to steal the data). Do you know any way of doing this, other than using a slave file having no reports and a Master file with reports whose access is not given to these people?

Henri

Seems like we have been getting more and more requests lately regarding security and data protection. This one is really pretty easy to implement. As you probably know, Q&A's normal security selections do not allow you to restrict the printing of reports. You can restrict the Design / Redesign of reports but that will not stop anyone who can edit records from making temporary changes to and printing existing reports. However, there is a security setting that most users don't take advantage of—*Field Level Security*. After assigning regular security limits to a database by assigning the users their ID's and passwords with their associated rights, you can then assign security levels to each and every *field* in the database. That's right, you can make a field:

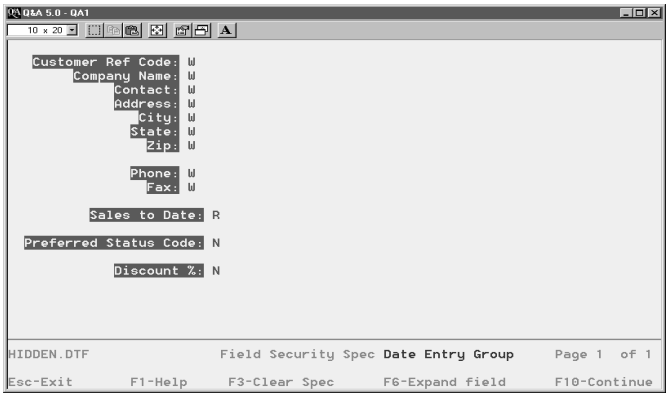


Figure 1. Setting field-level security. Here at the Field Security Spec, one field is made (R) Read-Only and the last two are set to (N) No Access. Users in the "Data Entry Group" won't be able to edit the **Sales to Date** field and won't even see the **Preferred Status Code** and **Discount %** fields. Nor will they be able to run reports containing these two fields.

W = Read and Write, where users can see and change data in the usual unrestricted way.

R = Read Only, where users can see but not change data.

N = No Access, where users can't even see the field.

You get to this spec by choosing *File / Design / Secure a file / Field Level Security* from the Q&A Main menu. (See Figure 1.) The third option—*No Access*—is the key.

First make sure that normal users, under their regular password rights (*File / Design / Secure a file / Assign access rights*) have no rights to Design / Redesign reports. I usually only give them rights to Enter / Edit data.

Next, redesign your database layouts so that they have a *one-character field* in them (that will not be used for any data) and set that field to No Access (N) in the Field level Security Spec. Assign this Spec to all regular users of the database (you cannot assign it to anyone who has full administrative or password rights) by pressing F-10 and then Alt-F7 for a list of the users.

The only users who will even see this field when viewing or adding records are those with administrative rights, and you can make it non-obtrusive by marking it *Read-Only*.

Finally, redesign all of your reports to include that new hidden field as an *invisible* column (for example, 99,I). This will not affect or change the way any of your reports look or print. The column is invisible but Q&A still considers it an integral part of the report. For any regular user *without* administrative rights, however, it will remove that report from the list of available reports they see. Since they have no rights to design a new report and they cannot see the old reports—ergo, they cannot print *any* reports. (See also *Privatize Your Sensitive Reports* in the May 2001 issue.)

Windows 2000 & Q&A for DOS Printer Drivers

So my Q&A was working fine down in DOS mode when I was running NT4. Now I have upgraded to Windows 2000 Professional and Q&A works fine down there at the C:

Stumped?

Send your Q&A questions to **@Help, The Quick Answer, Marble Publications, Inc., 1927A Harbor Blvd, Costa Mesa, CA 92627** or email to **mailbox@quickanswer.com**. Include your name, address, phone, and your Q&A version number (and whether DOS or Windows) and a detailed description of the problem. We'll publish those questions we feel are of general reader interest; individual responses aren't possible.

prompt, but won't print anything. I do not seem to have a compatible font or a printer driver in Windows 2000 that Q&A can find. Or, I don't know how to tell Q&A where to look in the font/driver file in Windows 2000. I think the big problem is that the printer I like to use with Q&A for my labels is a dot-matrix Panasonic KX-P 1624, and the disks I have for it are 5 1/4 inch floppies and my current machine can't read them. I tried installing the IMB Proprinter driver from Windows 2000 (supposedly compatible with the Panasonic) into Q&A, but that did not work either. Do you have any suggestions for a fix?—or is it new printer time?

Jim

I don't really understand what it is you're trying to do. But since so many people seem to have some sort of confusion over printers and printer drivers, let me cover some old ground regarding this issue. Before I do that, however, I hope that you are not trying to run Q&A from a DOS prompt! If you are, go to www.pcta-usa.com / win2000.htm to see how to properly set up Q&A to run under Windows 2000.

Where Q&A for DOS is involved, there are *two (2)* places a printer is installed:

1. In *Windows*, so that the operating system knows how to handle jobs sent to it, *and*…

2. In *Q&A*, so that Q&A knows what kinds of printer control codes to send to Windows.

Q&A doesn't use Windows' printer drivers and Windows doesn't use Q&A's printer drivers. One of these printer installations sends the codes (#2 above) and the other forwards them to the printer (#1).

As far as Windows 2000 is concerned, you do *not* need your old 5¼" floppy disks. If you *Add* a Printer in Windows 2000 (*Start Menu / Settings / Printers)* and select from the list of available printer drivers that come with the operating system, you will find the KX-P1624 in the standard printer list. That takes care of #1 above.

In Q&A, under *Utilities / Install Printer*, you'll find *KX-P1124/1524/1624/2124/2624* under Panasonic. This covers #2. Select this printer and go off and print your labels.

In general, to ensure proper Q&A printer compatibility:

- Stay away from *cheap* ink-jet printers that are designed to print *only* from Windows programs.

- Avoid the use of those 3-in-1 or 4-in-1 combination devices that fax, scan, copy, and print. They do each of these functions acceptably (usually) but none expertly. They also depend exclusively on Windows drivers.

- Always use a standard printer port (LPT or COM) rather than a USB type connected printer. (Q&A cannot work with USB connected printers.)

- If you need to have a color printer, check with the manufacturer to make sure that the printer can emulate or support printer control language (PCL) 2, 3, 4, or 5. If it can, then you can tell Q&A it is printing to an HP LaserJet II / III laser printer.

- By far the best bet solution: Use a real HP LaserJet or compatible printer for Q&A. Why? Because HP LaserJets are designed to be backward compatible—that is, you can successfully drive a *modern* LaserJet using a much *older* Q&A LaserJet printer driver.

Unexplained Q&A Errors

Q&A 5.0 is giving us some error references (such as Ref. 0535) that we cannot find listed in any documentation. In addition, we get strange performance only in Q&A such as certain keystrokes not doing what they are supposed to do and, sometimes, even Windows system errors when we try to print. We have made no changes to our system and the errors are not consistent. Any ideas?

Doug

Let's start with the 0535 error. This is an indication of an error accessing or utilizing the index (.IDX) file. What we have seen is that this can be caused either by a damaged database or by an intermittent network read error. Try the fix on the database first. Remove all the Speedy Fields at the Speed-Up Spec, press F10 to save, and then replace them and press F10 again. This will re-index the database. Then, as an extra safety precaution, copy the database design to a new name then copy all the records to that new database. Then throw away the old database and rename the new one to the old name. If the error still appears randomly, you are most likely experiencing network issues. Even I have had these on my own systems and have always been able to fix them by replacing either my network cable or the NIC (Network Interface Card) in the offending computer.

As for the second part, I have seen this type of situation happen with a few isolated clients. Usually, there is no apparent reason. The databases are healthy and have been recovered successfully. The errors are random and inconsistent. One of Symantec's old approaches for this type of error was to reinstall the program on the basis of a possible virus infection in an executable. While this sometimes, very infrequently, will work, we do not recommend this type of shotgun approach. Before going to this trouble, erase or rename, your QA.CFG file and restart Q&A. you will then have to reset Q&A to your preferences, including your printers. I think that you will find it will solve your problems. For some reason Q&A 5.0's configuration file seems much more susceptible to damage and corruption than those of the earlier versions.

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.

The Quick Answer

ALL CAPS, no caps, and Initial Caps

A.D. VELPER

CALL me a "purist," but they *forced* English grammar on me. There was no escaping it. I misplaced all my pens and pencils, made paper airplanes from my composition notebook, waged wars with spitballs and sawed-off drinking straws, and daydreamed of going to Mars. Still, they hammered into me how to write things down in a way that others (and even I) could read. You know, capital and lowercase letters, an occasional period and comma, verbs that connect to nouns, and all that—the basic stuff you're supposed to have down by the time you leave high school.

Well, I didn't give any of this much thought until the age of email arrived and I became dependent on it.

Why is it that some folks write even their personal emails with no thought of their being read by a human? It isn't just a matter of hasty language when you can't figure out what the person wants or is trying to say!

Maybe they don't think it's really writing, being just a gaggle of electrons and all. Maybe they hate to type, and find in email a convenient way to prove it. Maybe they feel they won't be heard across this ultra cheap medium unless they employ an alien tongue. Or maybe avoiding the Shift key like it had a razor blade sticking out of it is OK for email because it has such a short life span and nobody really cares about it anyway.

This ALL-CAPS or no-caps thing is the oddest quirk of all. Why *do* some people write things in all uppercase or all lowercase letters? Do they send out their business letters and faxes that way? It's a bit off-putting, not to mention hard on the eyes.

When someone pounds out an all-uppercase email message ("WHERE CAN I DOWNLOAD THE Q&A 4.0 PRINTER DRIVER FOR OUR NEW DAEWOO LEGANZA?") what does shouting it at the recipient accomplish? I picture this burly in-your-face boot camp drill sergeant reading some hapless grunt the Riot Act before demanding another 50 pushups.

On the flip side, an all lowercase email ("hi, i need help with my symantec q & a, linda"), though touching in a mousy sort of way, is equally strange. I wonder if this stems from being instructed (correctly) to type their email *address* in all lowercase, then deciding (incorrectly) that more must be better.

Some people even go to the added trouble of *reversing* the case. Just the other day I received an email with a Subject line that read: "q&a CONVERSION."

But why am I going on about all this? ("GET TO THE POINT, a.d.!") Well, because it's a convenient lead-in to the subject of my article this month.

Put a cap on it

You don't hear Q&Aers moaning much about issues with all uppercase, all lowercase and initial caps (where the first letter of each word is capitalized with the remaining letters in lowercase) because Q&A lets you format a text field for any of these styles.

You have your Format Spec codes **TU** (text uppercase), **TL** (text lowercase), **TI** (text initial caps) and just plain **T** for text, however you type it into the field. One of these is likely to accommodate your preference or the kind of data you're entering—for example, **TU** for a state abbreviation field, **TL** for an email address field, perhaps **TI** for a first and last name or company name field, and **T** for most other text fields. It's a snap. You just choose your format code and Q&A displays the text in that style.

So how come I just recently heard from *two* Q&Aers with questions about how to "convert" from one style to the other?

I didn't prod them as to why they needed to do this. About all I could do was point them to the Format Spec and mention formatting codes in Reports (about which they already knew), and confirm that Q&A's programming language has no built-in functions to convert one style to another.

Other languages do have such functions. In JavaScript, for example, you have your *string*.toUpperCase() and *string*.toLowerCase() functions. In VBScript you've got your Ucase(*string*) and Lcase(*string*). In PERL you've got uc(*string*) and lc(*string*) functions. PERL also gives you a ucfirst(*string*) function which puts the first character of the string in uppercase. None of these languages, though, give you an initial caps function that works like Q&A's "TI" formatting code or Q&A Write's *Title* feature (*F8 Options* / *Block Operations* / *T - Title*).

But back to the point: In Q&A there is no way to *programmatically* (with programming) convert one style of text to another. Well, actually, there is, but it's no cakewalk.

Oops!—Now the cat's out of the bag and you're dying to find out how. So I suppose I'll have to show you.

Sample database—a capital idea!

We'll need a little database with a few fields to demonstrate the programming and results. Figure 1 shows a record in CAPSORNO.DTF (or CAPsOrNo.dtf, if you prefer. I really wanted to name it CAPSOrNot.dtf, but that was one too many characters. At any rate, it's a Q&A for DOS 4.0 database that's included in this month's download file for Online Edition subscribers. Everything here will work in a Q&A for Windows database as well.)

Capsorno's fields are all plain text fields formatted "T" for text. I want you to know this up front so you won't think I'm cheating with the formatting codes and such.

(Incidentally, one of the leading causes of temporary blindness are Q&A databases with *all* the field names and *all* the data in all uppercase. Avoid them!)

In the first field (named **String**), you type your text. When you press Enter, the all-uppercase equivalent of **String** appears in the **ALL CAPS** field, the all-lowercase version appears in the **no caps** field, and the initial caps version appears in the **Initial Caps** field.

The three fields below the dividing line, **ACAPS**, **ncaps** and **Icaps**, contain programs that convert whatever's typed into **String** to one of the other corresponding styles. You don't necessarily need all these fields. If you just want to convert **String** to all caps, then all you need are the **String**, **ALL CAPS** and **ACAPS** fields. These temporary field(s) you can place on a back screen page and/or "hide" them however you like. They need be only a single character wide.

Here's the programming for all the programmed fields. I'll follow this with my usual astute commentary:

String field

```
> Clear(ALL CAPS, no caps, Initial Caps);
  ACAPS = @Replace(String, " ", "^");
  ncaps = @Replace(String, " ", "^");
  Icaps = @Replace(String, " ", "^");
  Goto ACAPS
```

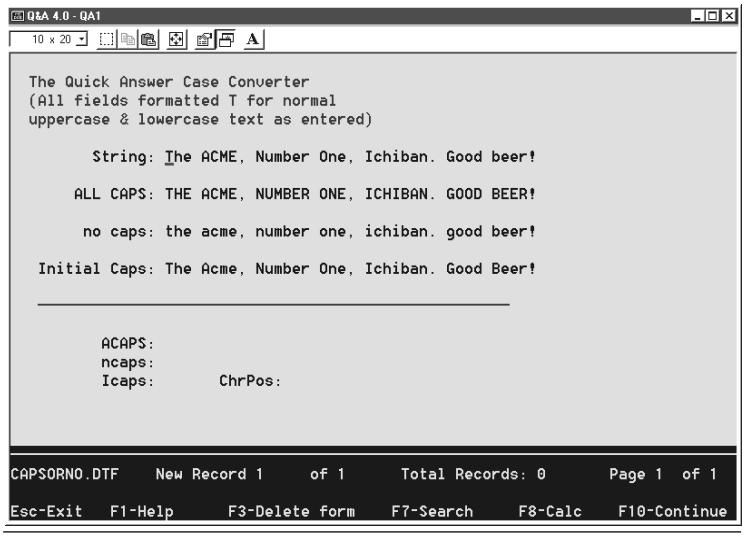


Figure 1. This database converts a text string to three different styles.

ACAPS field

```
< If ACAPS = "" Then {
  ALL CAPS = @Replace(ALL CAPS, "^", " ");
  Goto ncaps };

 If  @Asc(@Left(ACAPS, 1)) >= 97
 And @Asc(@Left(ACAPS, 1)) <= 122
 Then
 ALL CAPS = ALL CAPS +
 @Chr(@Asc(@Left(ACAPS,1)) -32)
 Else
 ALL CAPS = ALL CAPS + @Left(ACAPS, 1);
 ACAPS = @Mid(ACAPS, 2, 500);
 Goto ACAPS
```

ncaps field

```
< If ncaps = "" Then {
  no caps = @Replace(no caps, "^", " ");
  Goto Icaps };

 If @Asc(@Left(ncaps, 1)) >= 65
 And @Asc(@Left(ncaps, 1)) <= 90
 Then
 no caps = no caps +
 @Chr(@Asc(@Left(ncaps,1)) + 32)
 Else
 no caps = no caps + @Left(ncaps, 1);
 ncaps = @Mid(ncaps, 2, 500);
 Goto ncaps
```

Icaps field

```
< If Icaps = "" And @Instr(Initial Caps, "^") > 0
  Then {

  ChrPos = @Instr(Initial Caps, "^");
  ChrPos = @Str(ChrPos + 1) +
  @Mid(Initial Caps, ChrPos + 1, 1);
  If @Asc(@Right(ChrPos, 1)) >= 97 And
     @Asc(@Right(ChrPos, 1)) <= 122 Then {
  Initial Caps = @Left(Initial Caps,
  @Instr(Initial Caps, "^") - 1)
  + " " + @Chr(@Asc(@Right(ChrPos, 1)) - 32 ) +
  @Mid(Initial Caps, @Num(ChrPos) + 1, 500);
  Goto Icaps } };

  If Icaps = "" And @Instr(Initial Caps, "^") = 0
  Then
  If @Asc(@Left(Initial Caps, 1)) >= 97 And
  @Asc(@Left(Initial Caps, 1)) <= 122 Then {
      Initial Caps =
       @Chr(@Asc(@Left(Initial Caps, 1)) -32 ) +
       @Mid(Initial Caps, 2, 500);
       Clear(ChrPos);
       Goto String }
       Else {
       Clear(ChrPos);
       Goto String };

  If @Asc(@Left(Icaps, 1)) >= 65
  And @Asc(@Left(Icaps, 1)) <= 90
  Then
  Initial Caps = Initial Caps +
  @Chr(@Asc(@Left(Icaps,1)) + 32)
  Else
  Initial Caps = Initial Caps + @Left(Icaps,
 1);
  Icaps = @Mid(Icaps, 2, 500);
  Goto Icaps
```

About the programming

As I mentioned earlier, the **String** field is where you type the value you want converted to one of the three styles (all uppercase, all lowercase or initial

caps). It's on-field-exit program clears the three converted text fields and initializes the three utility fields with the **String** value. I use a special character ("^") here to serve as a temporary placeholder for spaces because it makes my programming easier (for me) to debug, even though I never find bugs in my programming.

The cursor goes to the **ACAPS** field first. This is the field that converts the text in **String** to all uppercase for the **ALL CAPS** field .You may have to read this program from bottom to top to make sense of it. The last line of the program says "Goto ACAPS" (the same field) because it's an iterative or looping program—it processes the text value one character at a time.

The two key Q&A functions here are *@Asc* and *@Chr*. @Asc returns the character's ASCII decimal code. (Your Q&A *User Guide* should have a table of ASCII characters in an appendix.) If that code is 65 through 90, you know it's an uppercase letter (A-Z). If it's 97-122, then you know it's a lowercase letter (a-z). This enables you to have the programming convert only letters of the alphabet, leaving all other characters (numbers, punctuation, whatever) unaffected.

@Chr is the counterpart of @Asc. You hand @Chr an ASCII decimal value and it hands you back the corresponding character.

As this program runs, it builds the **ALL CAPS** value one character at a time, while deleting the temporary string in **ACAPS** character by character as it progresses. In other words, for each loop of the **ACAPS** program, **ALL CAPS** grows while **ACAPS** gets eaten up, to mix a metaphor.

When **ACAPS** contains no more characters to process, then the *first* line of its program executes. The "^" placeholder characters in **ALL CAPS** are replaced by the spaces they stood in for, and control is passed to the **ncaps** field to process the same original **String**, but this time into all lowercase letters for the **no caps** field.

I direct your attention to the statement in **ACAPS** that ends with "-32)". If you subtract 32 from the ASCII decimal code of any lowercase letter, you'll get its uppercase equivalent. (For example, "a" minus 32 = "A".) Conversely, if you add 32 to an uppercase ASCII decimal code, you'll get the ASCII decimal code of its corresponding lowercase letter. Didn't know that? Neither did I until 15 years ago. And since then I've used this vital know-how at least once.

Anyway, so the **ACAPS** program is subtracting the ASCII decimal codes to get the uppercase equivalents, while the **ncaps** program is adding them to get the lowercase equivalents. (And both are skipping any characters that aren't in the alphabet.) Otherwise, the two programs are essentially the same.

The **Icaps** field contains much of the same code, but includes an additional routine that finds the character (along with its position in the string) that starts a word (or follows a space, if you prefer). It needs the **CharPos** (character and position) field to help it with this and is the only one of the three utility fields that requires it.

The programming supports a **String** length of 500 characters. If you need more, increase that value wherever it occurs in the **ACAPS**, **ncaps** and **Icaps** programs.

Now please don't write me asking what to do about initial caps for the McHenrys, LaTicias and Rio del Sols of this world. Not even Q&A knows what to do about these guys. (Or does it? See the sidebar.)

There you have it.

So what good is any of this? Hey, I don't get paid to be the Director of Usefulness around here. Plus, I've got incoming emails to grade.

A.D Velper has temporarily abandoned work on his Q&A Voice Recognition module in favor of more beach time. He can be reached at mailbox@quickanswer.com.

Honorable Mention

- Q&A does a darned good job when it comes to applying your database's field formatting settings (TU, TL and TI) to other areas of the program. For example, if you format a field for Text Initial Caps, you'll get Text Initial Caps by default in your merge docs, reports and even export files.

- In Reports, you can change the text format but, curiously, only to a limited extent. You can use the F(TU) command (which Q&A will change to F,T,U) in the Column/Sort Spec to output the field in all uppercase, but Q&A *won't* accept F(TL) or F(TI). Go figure.

- There's an interesting difference in the way Q&A applies initial caps to a TI-formatted field and how it capitalizes words when using Write's *Title* feature.

  If you type "john's a mccarthy" into a TI-formatted field, you'll get "John's A Mccarthy" when you press Enter. But if you type the same thing into a plain T-formatted field, press F6 then F8, choose *Block Operations / T - Title* and apply the Title case to it, you'll get "**John's a McCarthy**" (lowercase "a" and capital "C"!) more like a true title case with "McCarthy" spelled correctly .

  You might find some use for this. With Q&A 5.0, which supports the @Macro command, you might be able to write a clever on-field-exit macro that takes care of the text conversion for you.

Sesame Seeds... cont'd from page 2

record/sub-record as a single record for export. In other words, if you are exporting an invoice with line-item sub-records, *Sesame* will generate a separate record or line in the export file for each line-item in the record. The result will be a separate data line containing both the information from the main form plus the applicable information from each line-item.

Once you have determined what you want to export, *Sesame* then gives you options that many Q&A users have always wanted. (See Figure 3). You can select any field *separator* you want (a comma—the usual—or tab, custom character(s), or none). As for the field *delimiter,* such as quotes around the values, you can tell *Sesame* to delimit only the text fields or to place the delimiters around all exported values, including numbers and dates.

This option will come in very handy for export to financial and mailing software packages. You can select double quotes ("), single quotes (') or even a custom delimiter to place around the fields. (Something Q&A can't do.) And most importantly—and something *else* Q&A can't do—you can add a set of field names to the top of the record (usually called a *header record*) so that the receiver or receiving program knows just what data is in the file, what order it's in, and what field names are linked to it.

The final export file will include all the records you last retrieved and in the same sorted order as you specified in that retrieve. Figure 4 is an actual export file and shows the way *Sesame* handles sub-records. Note in the header that the fields from the sub-records (Gems) are listed with a "!" character to differentiate them from the main form records.

As always, we appreciate and thank you for your continued support and interest.

—Bill Halpern, President/CEO, Lantica Software

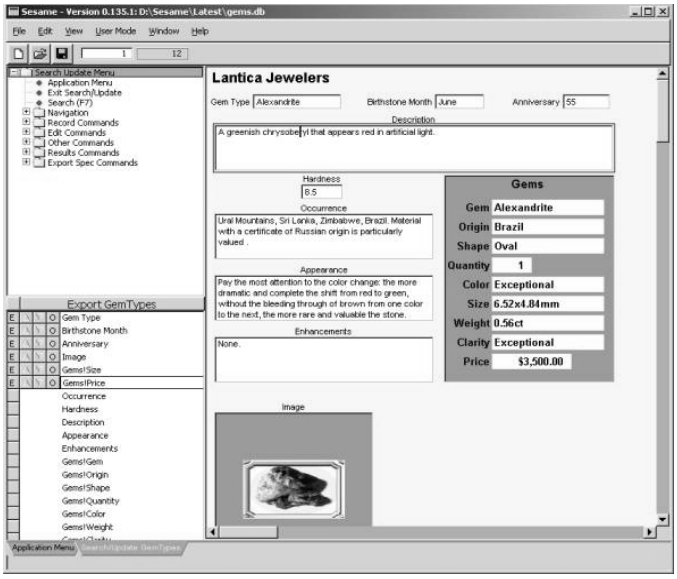


Figure 2. Selecting fields to export.

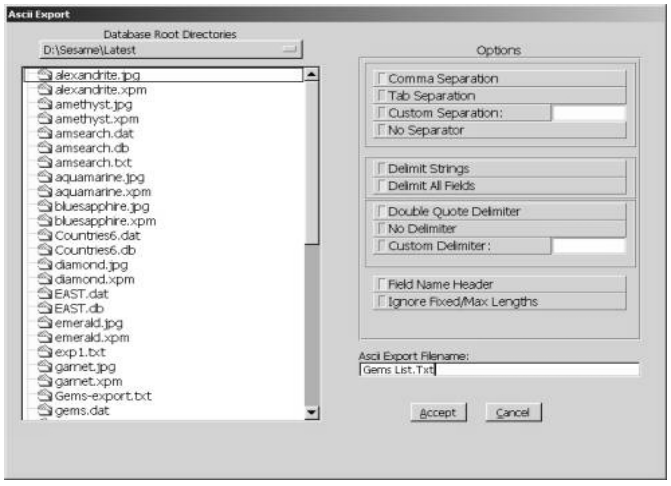


Figure 3. *Sesame*'s Export Options screen.

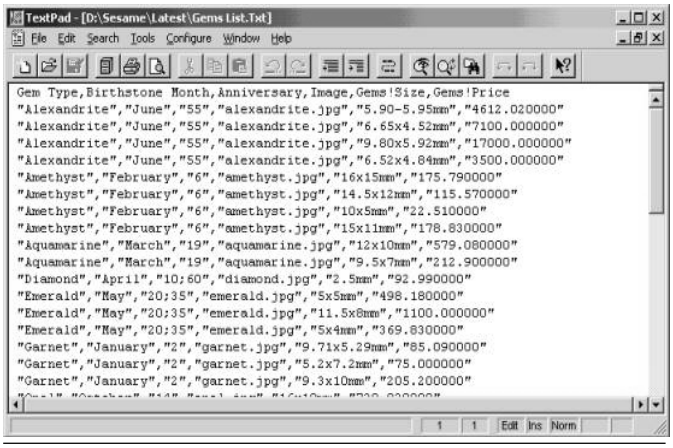


Figure 4. A *Sesame* export file.

Wanna Beta?

A few weeks ago, Lantica Software conducted a mass emailing to everyone we could find with an email address who was connected to Q&A in some way and/or who had filled out one of our "interest" forms at www.lantica.com.

The purpose of this campaign was to announce the upcoming *Sesame* beta testing program and give anyone who had not already done so the opportunity to tell us if they'd like to be considered as a beta testing candidate.

If you did not receive this email, then we do not have your current email address in our Q&A/*Sesame* email database and you should go to www.lantica.com soon and fill out the "Keep Me Updated" form there. If you'd like to be added to our list of *Sesame* beta testing candidates, be sure to check the checkbox that asks that question.

We do not yet have a definite date when formal beta testing will begin. But be assured that it will begin as soon as the product is ready for it.

Programming Traps... cont'd from page 5

Here, there are three input fields (**Date 1**, **Date 2** and **Rate/Day**). You can simply put a general programming statement in the **Rental Charge** field and set the Calc Mode to Automatic. That way, when *any* of the input fields are changed, you will get the correct result. To use on-field entry programming in the **Rental Charge** field would work but only when you first filled out the form. But if any of the input fields were subsequently changed (for example one of the dates, or the rate per day), then the final figure would be correct only if the operator actually took the trouble to tab into the **Rental Charge** field— and would they know to do so?

Why are these flaws so common?

Well, it's excusable. When you create a new database in Q&A, the Calc Mode is set to Manual by default. To change the mode you have to know to press Shift-F8 *and* press it during Add Data or Search / Update, not at the Program Spec! So novice programmers, stuck with Manual Calc, find that the only way they can keep their data correct is to use on-field-entry and on-field-exit programming, and they stick with that. The *Application Programming Tools Manual* gives many examples of using Logical Field Numbers rather than field names, including the clumsy "X#1" XLookup example. Likewise, the fields are incremented by ones in many of the examples. So don't feel bad if your databases are full of the "before" examples in this article.

You might find that deficient programming has left erroneous data in your databases. This can often be found by constructing retrieves especially to find data that is mismatched. Now's the time to get your programming in good shape for future migration to "Sesame."

Alec Mulvey is a director of Lantica LLC, and also owns Keyword Training & Consultancy in Ascot, near London, England, and has been building Q&A applications and training clients for 11 years. Keyword Training is the UK distributor for the International English edition of Q&A. Fax +44-1344-884-111, alec@keywordtraining.com, www.keywordtraining.com.

Two small Q&A 5.0 databases, Customer.dtf and Orders.dtf, are included in this month's download file for Online Edition subscribers. They're designed to demonstrate some of the points in this article

MARBLE
PUBLICATIONS
1927A Harbor Blvd #389
Costa Mesa, CA 92627 USA

November 2001

First Class