Keeping a Status Field Up-To-Date—*Always*

ALEC MULVEY

The Quick Answer
The Do-It-Yourself Guide to Q&A
June 2003
Volume 14 Number 6

MARBLE PUBLICATIONS

A couple of jobs I recently worked on took me into the area of On-Record-Entry and On-Record-Exit programming, and forced me to look in depth at how this is handled by Q&A—and, of course, any differences relating to this between Q&A and Sesame. It's actually something of a minefield, and is not covered comprehensively by the Q&A *Users Guide*. What's more, it forced me to look more closely at the nature of data stored in a database as well as the nature of reports.

What's the status?

One of the clients I did some work for uses Q&A to maintain a portfolio of properties with rents falling due. He wanted to have very clear and unambiguous messages at various stages of rents becoming due, then overdue by a short time, then more overdue. He was adamant that these messages had to appear automatically, requiring no user intervention.

The obvious way to do this is by using On-Record-Entry programming.

With Q&A, you can divide programming into these two distinct classes:

- On-Entry/On-Exit programming.
- General programming, or what the Q&A *Users Guide* calls "Calc Statements."

Here's what the Q&A *Application Programming Tools Manual* (*APTM*) says about the two:

> *There are two kinds of programming statements: calc statements and on-entry or on-exit statements. Calc statements begin with a Field ID and are all calculated at the same time. On-entry or on-exit statements are calculated only when entering a field, or when entering or exiting a record....*
>
> *On-entry and on-exit statements are only executed when entering or exiting a field, or when entering or exiting a record. On-entry and on-exit statements are* not *calculated by pressing F8, or by having the calculation mode set to automatic.*

Now, in my client's case, the status is dependent on *several* fields, such as the date when rent is next due, payment date, amount paid, arrears, and so forth. The details aren't important. Suffice it to say that the Status can be *OK*, or *overdue up to 21 days*, or *overdue more than 21 days*. All this, of course, is very much dependent on the current system date. Hence the status needs to be recalculated every day.

On-Record-Entry programming will mean that the status is recalculated every time the record is visited. Opening the record automatically gives the correct status—it will never be wrong. Or will it? If you then edit the record—post a payment, for example—the status again needs to be recalculated. In other words, you want the programming to continue to update the status while the record is open and being edited. My quest was to engineer a programming strategy that would work for all situations, always giving the correct status.

One idea was to use a general programming statement, setting the calculation mode to Automatic. This is done by pressing Shift-F8—not at the Program Spec as you might expect, but when viewing a record in either Search/ Update or Add mode. (See Figure 1 on page 3)

With the necessary calculations set to automatically execute as a general programming statement like the following, the program will run with every change to *any field* in the record:

Continues on page 3

Letters

THANK you for your excellent article last month on using ROBOCOPY to quickly backup the data on a computer to a removable hard drive. The article could save a lot of people much trouble, but I find that people are reluctant to adopt this kind of backup regime. Maybe they are frightened of hardware solutions.

I feel guilty for not having submitted a similar article earlier on XCOPY and disk cloning, because that is how I have been backing up for the last three years. I don't just copy all my files to a second hard drive, but create a bootable clone of my main hard disk on a backup hard drive in a removable caddy.

ROBOCOPY is designed to work only on NT, 2000 and XP machines. Since I'm a Windows 98SE user, I am very grateful for your discovering and mentioning the very similar program called XXCOPY, which is more universal than ROBOCOPY. Now that I have discovered it, I will replace my XCOPY routines with XXCOPY and save a lot of time using its convenient /CLONE switch.

I have two removable hard disk caddies in my computer. I store one bootable cloned hard disk in a spare PC and a one in a completely different location. The advantage of creating exact cloned hard disks is that, if necessary, you can just replace your working one with the clone. A little-known fact is that a Windows 9x computer will usually self-adapt to new hardware with a few reboots, so a cloned hard drive can simply be put into any new machine. You may need to install specific drivers for a different graphics card, modem, mouse and the like, but the basic machinery will be sorted out for you. (This may only be true of Windows 9x PC's.)

In a nutshell, all one does is prepare a second bootable hard disk (internal, external or removable) that the computer will recognize as, say, Drive D:\. Then, having installed XXCOPY, enter the following command in a DOS window:

```
XXCOPY C:\*.* D: /CLONE
```

and sit back and watch the magic happen.

For more details on the XXCOPY disk closing process, see *http://www.xxcopy.com/xxcopy10.htm.*

For more on the command line switches supported by XXCOPY, visit *http://www.xxcopy.com/xxcopy30.htm.*

You can download a freeware copy of XXCOPY from *http://www.xxcopy.com/download/xxcopy.zip.*

I hope this will extend the value of your article to users of Windows 9x and those (more "anxious" than you) who wish to create true clones of their hard disks.

Roger Skidmore, rogerskid@supanet.com

Indeed, XXCOPY is very similar to ROBOCOPY, and where ROBOCOPY will run only on NT-class machines (Windows NT, 2000 and XP), XXCOPY can be used on any Windows PC. (XXCOPY cannot, however, create a bootable disk on an NT-class machine.)

Like ROBOCOPY, XXCOPY copies only changed files, so daily backups are completed in minutes.

I've been using ROBOCOPY now for more than a month. As I wrote last month, it backs up my entire XP computer to my external USB 2.0 hard drive in about 90 seconds. (I'm a real speed freak in this regard.)

To compare ROBOCOPY with XXCOPY, I partitioned my external hard disk into G:\, H:\ and I:\ drives, then set up this XXCOPY batch file:

```
XXCOPY C:\ G:\ /
CLONE /YY
XXCOPY E:\ H:\ /
CLONE /YY
XXCOPY F:\ I:\ /
CLONE /YY
```

The C:\, E:\ and F:\ drives are the three partitions on my two internal hard drives. The G:\ through I:\ drives are the mirror partitions on the external disk. I use the /YY switch so that XXCOPY doesn't continually prompt for confirmations of various things..

Using the batch file, I ran a complete initial partition-to-partition XXCOPY clone of my computer's entire hard drive structure. Then, at the end of the following day, I ran a full XXCOPY backup. It took 3 1/2 minutes — just over twice as long as ROBOCOPY. I got the same result

Concludes on page 11

The Quick Answer™

Editor / Publisher
Tom Marcellus

The Quick Answer (ISSN 1052-3820) is published monthly in a downloadable "electronic" (Acrobat PDF) edition by Marble Publications, Inc., 1927A Harbor Blvd., Costa Mesa, CA 92627 USA. A Subscriber ID is required to download each issue.

Cost of subscriptions: 12 issues, $89 anywhere in the world. Single copy price: $9. All funds must be in U.S. currency. Back issues available upon request for the same price as a single copy. Back issues from 1990 through 2001 available in printed format. Back issues from 1995 onward available in Acrobat PDF files.

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

This publication is intended as a general guide. It covers a highly technical and complex subject and should not be used for making decisions concerning specific products or applications. This publication is sold as is, without warranty of any kind, either express or implied, including but not limited to implied warranties for the publication, quality, performance, merchantability, or fitness for any particular purpose. Marble Publications, Inc., shall not be liable to the purchaser or any other person or entity with respect to any liability, loss, or damage caused or alleged to be caused directly or indirectly by this publication. Articles published in The Quick Answer do not necessarily reflect the viewpoint of Marble Publications, Inc.

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**

Status Field cont'd from page 1

```
 #80: If Date Due < @Date - 21 THEN...
```

By making this programmed field the On-Record-Entry field as well, this would seem to meet the requirements.

You can have just one On-Record-Entry field and one On-Record-Exit field. They can be the same field, in which case the programming would be calculated on entering the record and executed again on leaving the record.

Setting the On-Record-Entry and/or On-Record-Exit field is done where you would expect: at the Program Spec. At the Program Spec, press F8, and you see what is sometimes called the "Entry/Exit Profile Box." (See Figure 2.)

You specify the Field ID number for the On-Record-Entry field, do the same for the On-Record-Exit field, then press F10. You have to use Field ID numbers (but without the "#" sign), you cannot use field names.

So, with the field containing the status calculation also set as the On-Record-Entry field, all should be OK—right? No! The On-Record-Entry programming works just fine, but any edits to the record do *not* then force a recalc. Remember the quote from the APT manual:

> *On-entry and on-exit statements are only executed when entering or exiting a field, or when entering or exiting a record. On-entry and on-exit statements are* not *calculated by pressing F8, or by having the calculation mode set to automatic.*

You can see this and try it out by using the sample ON-ENTRY.DTF database included in this month's download file. (It's a Q&A 4.0 for DOS database. You can open it in Q&A 5.0 or Q&A for Windows if you use one of these versions.)

In the first example, we have the fields..

```
Field1:
Field2:
Message: #30: Message = Field1 + " " + Field2 + "
" + @Str(@Time)
```

Suppose that **Field1** contains "Hat" and **Field2** contains "Coat." With no On-Record-Entry programming field defined, the **Message** field always shows *Hat Coat [time]*. Change "Hat" to "Gloves" and the Message will change to *Gloves Coat [time]*.

But make Field #30 the On-Record-Entry field, and you will get the correct text *only* on entering the record. Make a change to **Field1** or **Field2** and the change will *not* be reflected in the **Message** field. To see the correct message, you need to leave the record and return to it. To prove that this is On-Record-Entry programming at work, make a change, press F10 to move to the next record, then press F9 to return. You'll see the new, correct message. Now press Shift-F7 (restore record), and the old, former message re-appears.

So, the On-Record-Entry field *cannot* be a general programming statement because it won't then pick up any edits that are made.

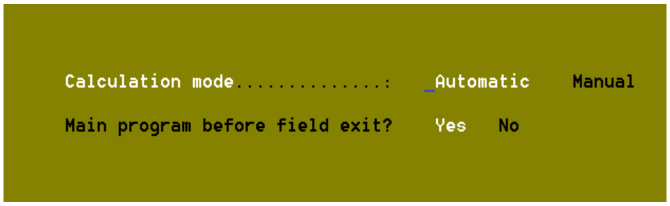
How about making it an On-*Field*-Exit programming statement?

```
Message: >#30: Message= Field1 + " " + Field2 + "
" + @Str(@Time)
```

This works. The programming is activated as On-Record-Entry/-Exit programming—both on *exiting* the field (because of the ">" at the beginning of the statement), and on entering the record (because it is designated as the On-Record-Entry field at the Entry/Exit Profile Box). Try this out in the sample database by just adding the ">" to the beginning of the statement.

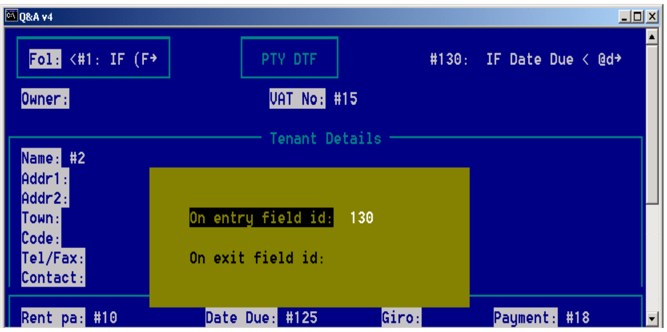
However, the On-Field-Exit program is still not sufficient. It allows for only *one* field to be changed before forcing the recalculation. The status, in this case, is dependant on *several* fields. When any one of them is changed, I want the status to be accurately updated, as with a general programming statement combined with Automatic Calculation. The answer to my problem is: I can't do it. Not with just one programming statement. I am forced to do something I try strenuously to avoid: put

Continues on page 8



```
Calculation mode..............:   Automatic    Manual

Main program before field exit?   Yes    No
```

Figure 1. Programming Calculations Options dialog—Shift-F8 from any database record.

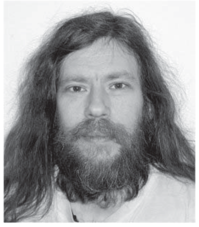


```
Fol: <#1: IF (F→          PTY DTF          #130:  IF Date Due < @d→

Owner:                          VAT No: #15

                  Tenant Details
Name: #2
Addr1:
Addr2:
Town:           On entry field id:  130
Code:
Tel/Fax:        On exit field id:
Contact:

Rent pa: #10        Date Due: #125      Giro:       Payment: #18
```

Figure 2: The Entry/Exit Profile Box, where On-Record-Entry/Exit fields are set.

Sesame Seeds

Power and Simplicity—A Sample *Sesame* Application



Mark Lasersohn

DURING the course of software development it is often the developers themselves that are tasked with testing a particular concept or idea. In that Q&A is often used to develop applications which are not databases in the strictest sense of the word, I thought it would be a good thing to put together a *Sesame* application that is not much like a database at all. Could *Sesame* and SBasic be used to create such a creature and, if so, what sort of creature would that be?

As a programmer, one of the applications that often comes up in my daily life is an odd beast called a Version Control System. A VCS, can (among other things) allow a number of programmers to work on a set of files without stepping on each others toes. In other words, I can check out a file from the group of files that comprise the "system" and lock that file so no one else can alter it until I check it back in. You, another programmer, can do the same with any other file. At any time, either of us have "read-only" access to all of the files. This prevents two or more people from editing the same files at the same time, and yet gives everyone access to the files they need to build the system.

It occurred to me that *Sesame* could be used to build a simple system like this. But how? A VCS does a lot of things not commonly associated with databases. Or so I thought, until I started to work on it. As a matter of fact, I was nearly done before I began. The entire project only took about three hours, and most of that time was actually spent trying to decide between competing, equally viable solutions. As you will see, the VCS application contains certain features and tools that you, as a developer, might find useful for creating what can serve as a kind of *Sesame* runtime application.

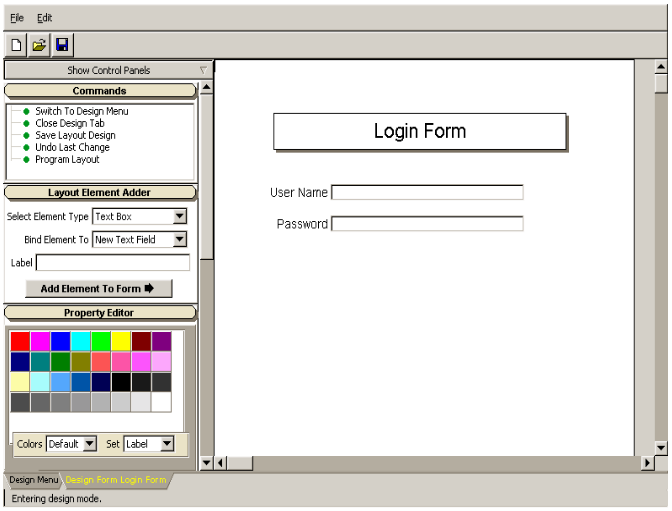
The starting point

The first thing I needed was a login form so my application would know who had checked out which file, and so programmers could not check files out willy-nilly. I've stripped the following forms down to the bare essentials for clarity's sake, so no aesthetic complaints from the peanut gallery.

Take a look at Figure 1. Simple enough—two fields and a static text element ("Login Form") to let you know where you are. But under the hood it gets to be fun. As you probably know by reading recent "Sesame Seeds" articles in these pages, *Sesame* allows programming to be attached to the form and associated with form-level events. Since I needed to go to a different form once someone is done logging in, I attached the following SBasic program to the form's on-form-exit event:

```
var success as int

if(((User Name = “Bill”) and (Password =
“Halpern”))
or ((User Name = “Alec”) and (Password =
“Mulvey”))
or ((User Name = “Erika”) and (Password =
“Yoxall”))
or ((User Name = “Ellen”) and (Password =
“Eckhouse”)))
{
    NotifyForm(0)
    success =
@SelectTreeItem(“NewSesameApplication!Forms!Search/
Update!Files Form!Files Form”)
    GlobalValue(“User Name”, User Name)
}
else
{
    NotifyForm(1)
}
```

As you can see, I've hardcoded the user names and passwords, which isn't great style, but will do if you know who your programmers are. After checking that I really have a legit programmer, the program calls "NotifyForm." By passing NotifyForm a "0" (zero), I am telling *Sesame* that the form can advance. If the person logging-in is an evil hacker trying to steal my code, NotifyForm is called



Select Element Type: Text Box
Bind Element To: New Text Field
Label:
Colors: Default
Set: Label
Entering design mode.

Figure 1. A simple login form for the Version Control System.

with a “1” and they get to try logging in again. A neat addition to this program would be a count on the number of login attempts, locking the fields and logging an error after one too many unsuccesful tries.

On a correct login the program calls “@SelectTreeItem.” SelectTreeItem is one of the functions provided by *Sesame* to help automate an application. It allows the programmer to select and run commands from the *Sesame* command tree. In this case, I am telling the system to leave this form and open the Search/Update spec for my other form, “Files Form.” (See Figure 2.)

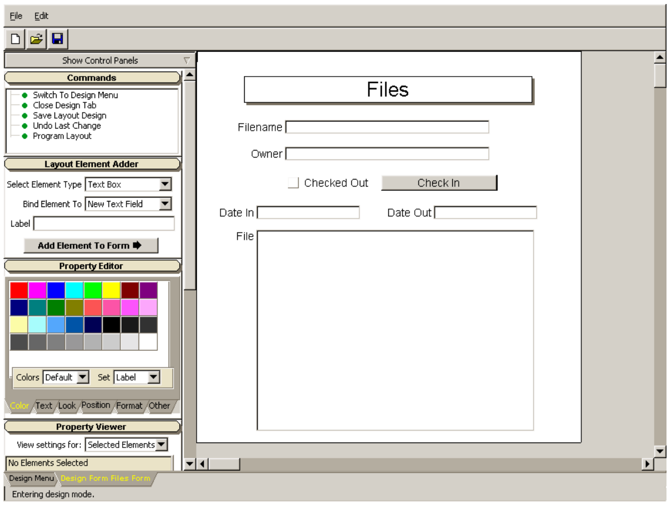
The last line for a successful login uses the GlobalValue function to save the name of the user login in a place where both forms can get to it. It should be noted that a value saved with GlobalValue is retained between runs of *Sesame* and is available to any database within an application file. By using the advance form (F10) to cause the actual login, *Sesame* can save a record of the login, which might be useful later if our evil hacker does manage to guess that I’m using last names as passwords.

The main application form

The Files Form is the main form in the Version Control System. It allows the user to enter a new file into the system and check files in or out. It tracks the dates and has a cut-’n-paste field to get or put the file. And since I force the user from the login form to the Search/Update spec of this form, the user can search for the file they want, or check which users have which files checked out.

Once they begin browsing, my on-form-entry program for the form runs:

```
if(FileExists(Filename))
{
    File0 = @insert(Filename)
}
if(Checked Out = 0)
{
    visibility(Check Out, 1)
    visibility(Check In, 0)
}
else
{
    visibility(Check Out, 0)
    if(@GlobalValue(“User Name”) = Owner)
    {
        visibility(Check In, 1)
    }
    else
    {
        visibility(Check In, 0)
    }
}
visibility(File0, 0)
```



Select Element Type: Text Box
Bind Element To: New Text Field
Label:
Colors: Default
Set: Label
View settings for: Selected Elements
No Elements Selected
Entering design mode.

Figure 2. The main form in the Version Control System.

This program reads the file in question into the cut-’n-paste field, checks if this file is checked in or out, and sets the visibility of the different elements accordingly. Looking at Figure 2, you may notice that something is a little fishy in Denmark: where is the “Check Out” button that the programming refers to? Well, fact is, it is sitting directly under the “Check In” button. Since only one of the two buttons can be visible at any time, there is no need to give them each their own space on the form. Notice also that the “Check In” button appears only for the user who checked the file out. This reduces the “willy-nilly” factor.

The “Check Out” button appears if this file is not currently checked out. It is a command button that runs the following program:

```
Checked Out = true
Owner = @GlobalValue(“User Name”)
visibility(Check Out, 0)
visibility(Check In, 1)
Date Out = @Date
visibility(File0, 1)
```

First, it sets the read-only boolean field “Checked Out” to true. It sets the Owner field to the login name of the user using the function @GlobalValue to retrieve that value from the server. It turns off the “Check Out” button and turns on the “Check In” button. It sets the “Date Out” field to the current date. And lastly, it turns on the “file” field to allow the user to copy its contents from within *Sesame* to their development environment or editor.

The “Check In” button does nearly the same thing but in reverse:

```
Checked Out = false
Owner = “”
visibility(Check Out, 1)
visibility(Check In, 0)
Date In = @Date
visibility(File0, 1)
File0 = “”
```

The big difference is that it clears the “file” field, thereby giving the user a nice fresh slate on which to paste their code back into the system. My Version Control system doesn’t actually save the code until the user advances the record using F10 and triggers the form’s on-exit event program:

Concludes on page 12

@Help

EDITED BY WILLIAM HALPERN

Stopping a Macro-Driven Procedure On Error

I have a situation that I just don't know how to deal with. I have followed this column for years and have used the advice to develop a neat little process that automatically sends out letters and E-Mails to my customers every day. The selection process is based on certain fields being filled in the master database. The routine is run by macro and does many things like selecting the records, exporting them, importing them into a second interim database, mass-updating them for additional information, and finally creating merge files for MS Word and Outlook. The process works fine if there are in fact records that need to be processed. However, if there are no records, the macro goes wild and does all kinds of nasty things to my databases. I know I could do a manual test for records and then run the macro or not. But I want the system to run automatically each time Q&A is started on a certain workstation. I use Q&A 5.0. Is there a way to place "conditionals" in a macro.

Stephanie

Thanks for the kudos. The best praise we can get is to see someone put our advice to use. There are always ways to do things in Q&A that its designers never thought of or never really intended it to do. In Q&A 5.0, checking a macro-driven procedure midway through and then branching based on a condition is possible. It involves three separate macros, an interim database, a batch file, a shell command, and optionally a Q&A write document. It might seem complex at first, but it really isn't. A runaway macro is about the worst thing a Q&A user wants to experience, and it is usually due to the macro expecting something to be there when it isn't. Here's what you are trying to accomplish.

You want Q&A to search for applicable records and export them to a text file. Then, you want to test for the existence of that file (did Q&A find any records with which to generate the file) and continue if it is found, or stop and display a message if it is not found. The key here is a way to test for the success of the file creation.

You start with a simple two-field database called **sendmchk.dtf** that is not used by anyone but the system. Make sure you add a single record to this database (anything in the first field) and save it. The two text fields are programmed this way:

```
<#1: If @Update Then {
#10=@Shell("C:\Qadata\Sm1.bat");
#10="";
#10=@Insert("C:\Qadata\sm.txt");
Goto #10
}

<#10: If @Instr(#10,"notify") > 0 Then {
@Macro("Continue notify")
}
Else @Macro("quit notify")
```

This example assumes that C:\Qadata is the default folder for Q&A databases.

Then you write this simple batch file in Q&A Write and save it with Ctrl-F8 to C:\Qadata\Sm1.bat:

```
dir C:\qadata\notify.txt /b > C:\qadata\sm.txt
```

Next, create the three macros required:

```
<begdef><alt1><name>"mail<sp>notification"<vidon>
<esc><esc><esc>wg<capsf4><enter>notify.txt<f3>y<esc><esc>
<esc>fues<capsf4>master.dtf<enter><capsf4>notify.txt
<enter><altf8> your retrieve spec here
<enter><f10><altf8> your_merge_spec_here
<enter><f10><f10>n<esc><esc><esc>fs<capsf4>sendmchk.dtf
<enter><f10><enddef>
*
<begdef><nokey><name>"continue<sp>notify"<vidon><capsf10>
<esc><esc><esc>fuisnotify.txt<enter> here you
would place your macro that does your actual
processing <esc><esc><esc><enddef>
*
<begdef><nokey><name>"quit<sp>notify"<vidon><capsf10>
<esc><esc><esc>wt<enter> either type in a message
to appear on the screen or call up a pre-designed
write document with your failure message in it at
this point <enter><esc><esc><esc><enddef>
```

That's it. Now, let's see what it does. The first macro is named Alt1 and can be run on startup. It does three things:

1. Erases the file left over from the last run so that the testing can work properly.

2. Uses your saved Retrieve and Export specs to find and export the data (from master.dtf) to a text file called *notify.txt*, then presses "N" in case no records were found.

3. Opens the interim two-field sendmchk.dtf database in Search/Update mode and presses F10 to search the record.

Here's where Sendmchk.dtf takes over. It uses Q&A 5.0's @Shell command to run the batch file shown earlier. If the export was successful, there will be a text file named *notify.txt*, and the batch file will write its title to another text file named *sm.txt*, which will be inserted (via @Insert) into the second field. The programming will then determine whether to continue or quit. If the file exists and has been inserted, the second macro (Continue Notify) will be launched. If not, the third macro (Quit Notify) will be run.

Each of these macros quits the check database and then does what you want it to do based on success or failure to this point.

This might seem like a lot to do at first glance, but I bet you can create this routine in not much more time than it took you to read about it.

Needs a "Tickler File"

I work for an engineering firm. One of the other assistant's jobs is to keep track of contractors' insurance policies when a job starts, make sure they haven't expired, and get them renewed if they have. Is there some kind of tickler file we could make so that we could print out a report every couple of weeks saying something like XYZ's disability insurance expires on May 1, 2003? The data in the file would be stuff like the name of the company, date and type of contract, location of contract, and the like.

Ann

If by "tickler" you mean something that runs automatically or on a timed schedule, then, no, you *can't* do that in Q&A. (Well, you *could*, but it would require a second copy of Q&A and Windows Task Scheduler, and….It's a topic for a whole article.) On the other hand, if you just want Q&A to run a report that tells you your contractors' insurance expiration status within a desired time-frame, it's a perfect application for a Q&A 5.0 macro driven report printed to Screen or a Printer. You can easily define the report (or reports) with simple Retrieve specs. For example:

```
>={ @Date } .. <= { @Date + 30 }
```

in the policy expiration field will find all policies expiring on a date within the next 30 days.

Or:

```
>={ @Date } .. <{ @Date + 8 }
```

in any date field, retrieves what will expire in the coming week (seven days)

You can include any fields you want in the report, and you can have the report run via a macro which Q&A launches on startup (autostart macro). Using a method similar to the one explained in the answer to the previous reader's question, you can even build a little Q&A database that loads the date, tests it, and runs the report if it is Monday. Otherwise, it quits and leaves you at the Main Menu. How's that for a tickler at the beginning of each week? You can also vary the interim test database to use @Userid and @Date and get specific startup ticklers for the person who logged on to that copy of Q&A.

The programming in this test database can be something like this:

```
<#1: If @Dow$(@Date) = "Monday" then
@Macro("Tickler") Else @Macro("Quit")
```

Or:

```
<#1: If @Dow$(@Date) = "Monday" Then {
If @Userid="John" then @Macro("JTickle");
If @Userid="Mary" then @Macro("MTickle");
If @Userid="Sam" then @Macro("STickle")
}
       Else @Macro("Quit")
```

Remote Control and Windows 2000/XP

We are running Windows 2000 Server for Small Business on a new and very powerful Dell server. We have two remote workstations running XP Home and XP Pro, respectively. We have PCAnywhere 10.5 installed on both workstations and the server. The workstations *in* our office are comparable machines, all running XP Pro. We use Q&A for DOS 4.0. When we link up the host and remote workstations, anything that's a Windows program zooms. Then, we get to Q&A on a remote workstation. It is so slow that we start looking for a hammer to "persuade" it to accept a normal typing speed. What would help here? Our computer tech is working on many angles. He has gone to the speed settings in PCAnywhere and adjusted them upward. No success. I would really like to fix this before *Sesame* comes out. Could it be that Norton Antivirus is slowing Q&A down? Might Norton Internet Security do that as well? There is a program from Netopia called "Timbuktu Pro" that their tech people claim will help, but I hate to reinvent the wheel. Netopia says that they do have screen refresh issues with DOS programs running in full screen, but not when they're running in a window. We can't work in a little Q&A for DOS window, straining our eyes to read that tiny type. So, I don't know if their solution would be workable either.

Bernie

Continues on page 9

Status Field cont'd from page 3

the same programming in two places. An On-Record-Entry statement to show the status on first viewing the record (without editing the record) and also the same programming as a general programming statement, so that any change to *any* field will show the correct status.

How do I configure the On-Record-Entry field?

```
   Message: #30: Message = ...

or Message: >#30: Message = ...

or Message: <#30: Message = ...
```

It makes sense, in this case, for it to be the first example—a general programming statement. As such, it will *only* be calculated on entering the record. This avoids un-necessarily recalculating it twice.

A brief digression

This point reminds me of the other related job I did recently. I was examining a client database that I hadn't seen before, and I was looking at it separately from its other dependent databases. This, incidentally, is a good way of quickly seeing what other databases are being looked up—you receive an error message, "*Cannot open lookup file...*" showing the full pathname of the database Q&A is attempting to search.

I saw that the first field in the database had extensive XLookups—dozens of them. This was not really necessary. They were doing things like looking up all the customer details by matching the customer ID. They really should have been made conditional in some way, such as executing only during Add Data, or only if some key fields were blank, with an "override" field if it should need to be looked up freshly. But, even worse, they were all being looked up *twice*! The first field—the designated On-Record-Entry field—had its programming statement starting with a "<" On-Field-Entry symbol. So all the programming was executed on entering the record, and again on entering the field. A total waste of resources, placing a huge overhead on record browsing and increasing network traffic to the detriment of all users.

You can see this effect in the sample database. Put this statement in the second field:

```
<#10: Message1= Message1 + " " + #1
```

Then make it the On-Record-Entry field. Make the first field on the form Read-Only, which will then make the second field the first available data entry field. Do this and you will see that the text in Field #1 is added to Message1 twice every time you enter the record. If you just recall the first record, pressing F9 has the effect of "entering" that record freshly, so you can see the results of an F10/F9 immediately.

Remember, you can press Shift-F7 to see the record as it was before any of the programming was activated.

One downside of On-Record-Entry programming is that all records are automatically "changed." If you retrieve a record, then press Escape before editing it, you will still get the "*Your changes to this record have not been saved...*" message, which can be annoying.

Now he wants a status report!

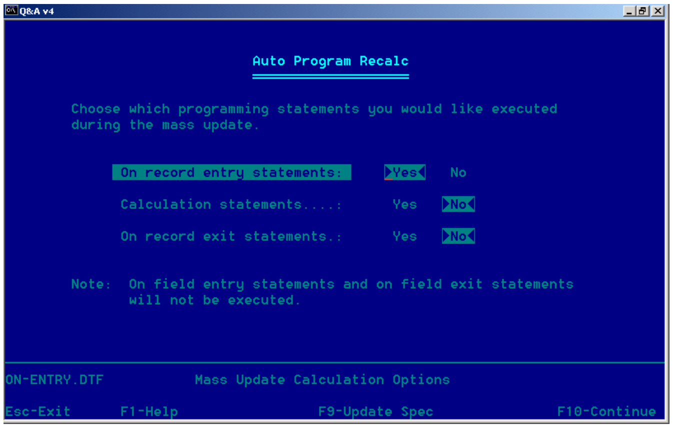
Getting back to my initial brief: after making the modifications, the status was always correct. The client was happy. Then he says that he wants to be able to print out a report, at any time, of overdue records showing the accurate status. CLUNK! Another problem. This might be obvious to you, but the programming in the database will not affect records that aren't opened. Data doesn't change by itself.

Using this example, suppose the database is untouched for a month, and then a report is printed from it showing the status and days overdue. The information will be sorely out of date, reflecting when it was last updated, which is when the records were last viewed.

What's the solution? There are two ways to go about it. Firstly, you can open each record. Either manually, by advancing through each record in Search/Update mode, or automatically, by performing a Mass Update. When doing a Mass Update, it is important set the right options. A Mass Update will, by default, only execute the programming in the Mass Update's Update Spec. If you want the database's *own* programming to be executed, you need to specify this by pressing F8 (at the Update Spec) for the Auto Program Recalc screen. (See Figure 3.)

The default is for all database programming *not* to be executed. You can see that you have options to execute On-Record-Entry statements, General programming

Continues on page 10



```
                    Auto Program Recalc

   Choose which programming statements you would like executed
   during the mass update.

        On record entry statements:    [✓ Yes]  No

        Calculation statements....:     Yes    [✓ No]

        On record exit statements.:     Yes    [✓ No]

   Note:  On field entry statements and on field exit statements
          will not be executed.

ON-ENTRY.DTF          Mass Update Calculation Options
Esc-Exit     F1-Help              F9-Update Spec                F10-Continue
```

Figure 3. The Mass Update Auto Program Recalc screen.

@Help...cont'd from page 7

You have run into the problem with Windows XP/2000 and remote control software. It is not Q&A, but *any* DOS program running in a window on an XP or 2000 Host machine. The problem exists with PCAnywhere, Carbon Copy, and a few others I have tried. I do not know about Timbuktu. Let's not even discuss what happens in full screen mode where PCAnywhere will stop responding and you will have to get someone at the host side to put Q&A back into a window. There is no way around this that we have been able to find. For the most part, we have required that our remote clients keep one Windows 98 computer available just for use with PCAnywhere and Q&A. I know that this is not an elegant approach, but it does work. A way around having a separate machine is to set up a dual-boot computer that runs both 98 and XP. For normal operations, you stay in XP and then when you need PCAnywhere, you reboot to 98.

If Timbuktu Pro works with Q&A for DOS running in window mode, that would be wonderful. If you are saying that Q&A for DOS will only run in a too-small a window in XP or 2000, read our previous columns in these pages where we show you how to make your Q&A window much larger. In fact, I run Q&A in nothing but a window. What you give up in full screen mode just isn't worth the small difference in size.

There's one other solution, though a costly one: The Windows XP version of Citrix will also work with Q&A for DOS and Windows 2000/XP very successfully.

Q&A Basic Requirements

My hard drive crashed today. It looks like I should just buy a new computer. I've heard that some of today's hardware won't support programs like Q&A. Is there a hardware specification reference list and operating system instructions available anywhere that you know of? I'm visiting www.qaug.com right now and can find no mention of Windows XP. Any help is greatly appreciated.

Murph

Here's a very brief summary of Q&A's requirements:

Q&A 4.0/5.0 for DOS

- Windows 3x, 9x, ME, NT, 2000, XP.
- NO shared RAM (on-board video cards and some laptops).
- Proper memory settings in shortcut.
- Lesspace.
- If networked, mapped network drives rather than UNC (\\server) drives.
- Windows Networking rather than Novell (4 or higher).
- Parallel (not USB) printer that supports PCL 3,4, or 5 or emulates one in Q&A's list of printers. (Avoid "multi-function" devices that print, fax, copy, etc., and "Windows only" printers that won't print from DOS programs.)

QA 4.0 for Windows

- On Windows NT, 2000 or XP needs John Dow's NTFIX patch (www.johntdow.com).
- Databases must be stored on a partitioned drive that's less than 2GB.
- If networked, Windows Networking rather than Novell (4 or higher).
- Is touchy with some printers (no known list).

Entries Disappear Too Fast

This may be hard to explain and there may be no fix. I have a database where people enter sales for individual customers. It is not an invoice system. I use it strictly as a marketing and tracking tool. Every time a customer orders something, I want my people to go to that customer record and enter that sale amount. The programming then adds that to the previous total, updates the date of the last order, and clears the new entry so that it cannot be added to the total again. Pretty simple and all I need. But here's the problem: Q&A is too fast. As soon as the person types the sale amount and presses Enter, the new amount instantly disappears and they don't know if they typed it correctly. They don't want to do it again for fear of an error. So, I have a built-in error creation system. Help!

Tom

There's more than one way to skin a cat and almost always more than one way to solve a problem in Q&A.

If all you're looking for is a visual confirmation that a new amount entered has in fact been added to the Total, you could use conditional @Msg statements like the following (in conjunction with your existing programming) to display a message at the bottom of the screen when the cursor leaves the **New Sale** field:

```
> If New Sale > 0 Then

If @Instr(New Sale, ".") = 0 Then
@Msg("Sale Amount $" + @Str(New Sale) + ".00" + "
added to the Total.")

Else If @Len(New Sale) - @Instr(New Sale, ".") = 1
Then
@Msg("Sale Amount $" + @Str(New Sale) + "0" + "
added to the Total.")

Else
@Msg("Sale Amount $" + @Str(New Sale) + " added to
the Total.”);

  [Additional Statements]
```

Three versions of the @Msg statements are needed to “tweak” the money value from the **New Sale** field so it appears properly formatted as text in the message. The result will be a message like this:

```
Sale Amount $35.50 added to the Total.
```

You could instruct users that if the amount they entered was incorrect, entering it again preceded by a minus sign (such as “–35.50”) will reverse it out.

For something that really gets “in your face,” with Q&A 5.0 you could work the above @Msg statements into an @Askuser dialog or an @Msgbox instead. An @Askuser dialog could inform them in advance of the sale amount about to be added to the total, and they could press “Yes” to add it, or “No” to cancel it. You could optionally add a supplementary visual cue by using @Color to temporarily change the color of the **New Sale** field, say, to red after entry of the amount, as a further reminder to the user while working in the record that this one has already been updated.

If you need more than this, you can always make your programming statement field the *On-Record-Exit* field. This way the calculations won’t take place until the record is saved. While they’re in the record no calculations will take place. Their current entry won’t disappear—they can review and change it if need be. Even pressing F8 (the manual Calc key) won’t do anything. But when they save the record by pressing F9, F10 or Shift-F10, the programming will run and the new totals (and new date) will be there. All you have to do is to tell them to enter the new sales figure, review their entry, and press F10.

There’s yet another way to accomplish the task. You could use Q&A 5.0’s XPost command to post sales and dates from *Database A* to the matching customer record in *Database B* in real time. When XPosting to a number or money field, you can use the *Add* option to tell Q&A to add the posted amount to the amount already in the target **Total Sales** field. You should use If/Then programming to protect against double entry here, too. And you will also need a unique field, such as a Customer ID field, that XPost can use to match the records in both databases. With this method, your XPost statement in *Database A* would tell Q&A to find the matching record in *Database B* and XPost/Add the new amount to the **Total Sales** field. A second XPost statement could post the sale date to the corresponding date field in the same customer record.

One more option before I go. You can make use of mirror fields that show the user the current entry, its date and even its time, then have Q&A clear these fields On-Record- Exit. This way, users can review and correct their entries, and be confident in those entries *and* the totals before leaving the record.

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.

Status Field cont’d from page 8

statements, and On-Record-Exit statements. These options are not mutually exclusive—you can choose one, two, or all of them. For our example, either On-Record-Entry or Calculation Statements would work since they have the same programming. Calculation statements would of course activate *all* such general programming in the database. The Update Spec itself would be blank in this case. You’d save the Spec to ensure that the right options are used. You’d then run the report after running the Mass Update, and all the status information would be current.

The second option is to build the necessary programming into the report itself by means of derived columns. Because of the nature of derived column programming, it can be trickier to achieve the desired result. You can’t, for example, chain together successive statements separated by semicolons and thus work in stages to a result. The column’s formula must be a single programming expression which produces a value to print in the column.

Here’s an example of a derived column that puts the word “Overdue” in an extra column if there is a Date Due and it is more than 21 days before the current date:

```
Heading: Overdue?
Formula: @Txt(Date Due > 0 AND Date Due < @Date-
21,”OVERDUE”)
Column Spec: 40,C
```

The point about this technique is that the data is not itself being updated. The database still contains outdated information. For my client, I determined that in fact both were needed—a report with derived columns *and* a Mass Update, run by a macro from a custom menu. So I wound up having put status-related programming in the application *three* times!

Bad programming?

Nah. Good business!

Concludes on page 12

Quick Tip My Favorite Windows Utilities

Zap Unnecessary Windows Startup Programs

Many Windows problems are caused by the loading of inappropriate or unneeded programs at startup. **Startman** is a free start-manager utility that checks all the programs your computer loads on startup. It reaches far deeper than Windows' own MSCONFIG utility, makes recommendations (which are usually worth taking), and allows you to delete startup program entries you really don't ever want to run again. It examines Registry entries as well as WIN.INI entries. It has slightly reduced functionality with XP but is essential for Windows 9x users. Since using it, my computer loads more quickly and seems to have fewer hiccups. You can get more information on Startman and download it at the following location:

http://www.spywareinfo.com/downloads/startman

For more info on startup programs in general, see:

http://www.pacs-portal.co.uk/startup_index.htm and *http://www.answersthatwork.com*

Click on "Libraries of Answers that Work," then click on "The TASK LIST" in the right hand panel. You can search alphabetically for the most common Windows startup programs, find out who their makers are, what the programs do, and get advice on whether you can safely remove them.

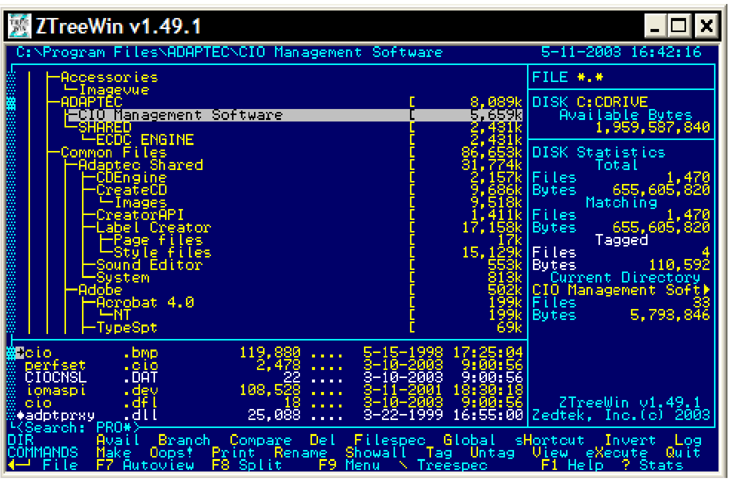
Windows File Manager Replacements

My favorite Explorer-enhanced file manager, **PowerDesk**, is now at version 5. It's free, and it does almost everything a solid file manager should do, including providing a transparent Zip-file manager when used with WinZip. Far more useful and user-friendly on a daily basis than Windows Explorer. See *http://www.v-com.com/product/pd_ind.html*

And for those DOS veterans who yearn for the days of XTREE GOLD, there is the look-alike 32-bit file manager **ZTREE** (see graphic below) which, once installed, can conveniently be called from Q&A 5.0's Alt-F9 External Programs screen. See *http:/www.ztree.com*

All these utilities are freely downloadable.

Roger Skidmore, rogerskid@supanet.com



```
ZTreeWin v1.49.1
 C:\Program Files\ADAPTEC\CIO Management Software           5-11-2003 16:42:16
  │ ├Accessories                                     │FILE *.*
  │ │└Imagevue                                       │
  │ ├ADAPTEC                        [         8,089k │DISK C:CDRIVE
  │ │├CIO Management Software       [         5,659k │   Available Bytes
  │ │└SHARED                        [         2,431k │       1,959,587,840
  │ │ └ECDC ENGINE                  [         2,431k │
  │ ├Common Files                   [        86,653k │DISK Statistics
  │ │├Adaptec Shared                [        31,774k │       Total
  │ ││├CDEngine                     [         2,157k │Files           1,470
  │ ││├CreateCD                     [         9,686k │Bytes     655,605,820
  │ │││└Images                      [         9,518k │      Matching
  │ ││├CreatorAPI                   [         1,411k │Files           1,470
  │ ││├Label Creator                [        17,158k │Bytes     655,605,820
  │ │││├Page files                  [            17k │       Tagged
  │ │││└Style files                 [        15,129k │Files               4
  │ ││├Sound Editor                 [           553k │Bytes         110,592
  │ ││└System                       [           813k │  Current Directory
  │ │├Adobe                         [           502k │CIO Management Soft▶
  │ ││├Acrobat 4.0                  [           199k │Files              33
  │ │││└NT                          [           199k │Bytes       5,793,846
  │ ││├TypeSpt                      [            69k │
 ▸cio         .bmp      119,880 ....  5-15-1998 17:25:04 │
  perfset     .cio        2,478 ....  3-10-2003  9:00:56 │
  CIOCNSL     .DAT           22 ....  3-10-2003  9:00:56 │
  iomaspi     .dev      108,528 ....  3-11-2001 18:30:18 │
  cio         .dfl           18 ....  3-10-2003  9:00:56 │  ZTreeWin v1.49.1
 ♦adptprxy    .dll       25,088 ....  3-22-1999 16:55:00 │Zedtek, Inc.(c) 2003
 └<Search: PRO*>
DIR        Avail  Branch  Compare  Del  Filespec  Global  sHortcut  Invert  Log
COMMANDS   Make  Oops!  Print  Rename  Showall  Tag  Untag  View  eXecute  Quit
◄┘ File    F7 Autoview  F8 Split   F9 Menu  \ Treespec       F1 Help  ? Stats
```

XXCOPY... cont'd from page 2

the following day. I'm a stickler for speed, but at under four minutes, XXCOPY is certainly no laggard.

XXCOPY takes a long time deciding that it can't copy a file that's in use on the computer. (Neither program will copy files that are in use.) ROBOCOPY supports a /R:0 (retries:0) switch that skips such files without a noticeable delay.

To prepare for the test, I used the new PartitionMagic 8.0 (easier, faster and more goof-proof than ever) to partition my external drive into three NTFS volumes (to mirror my internal drives) and also convert the FAT 32 partition on my older internal DISK2 to NTFS. The previous 15GB of data on my computer shrunk to a little over 9GB. (It isn't compression—NTFS is simply a more efficient file system.) This might actually enable ROBOCOPY to run even faster, but I would need to make another full ROBOCOPY initial *backup before I could test that.*

If I were a Windows 95/98 or (Heaven forbid!) ME user, I would certainly want XXCOPY for it's handy CLONE feature to create the bootable disk on the external hard drive in case of internal hard drive or total system board failure. Because I run XP (where XXCOPY can't create a bootable drive), the difference in speed between the two utilities would tend to keep me with ROBOCOPY. But since I've now partitioned my external disk to match the partitions on my internal disks, I figure I may as well stay with XXCOPY. If the extra time it takes manages to get under my skin, I'll switch back to the speedier ROBOCOPY. —Ed.

@Status Field...cont'd from page 10

How does *Sesame* compare?

Sesame offers all of Q&A's programming-related features and more. Q&A supports the following "events" (actions that are "trappable" and can have programming attached to them):

- On field entry
- On field exit
- On record entry
- On record exit
- General programming statements

This last one, general programming, has been renamed in *Sesame* as to "On Form Change" because that's what it is. Change any field on the form and the whole lot gets recalculated.

Sesame adds two more events. The most significant of these is On Field Change. Programming can be attached to a field which is only executed when that specific field's value has been changed. This is a valuable addition and obviates all these "mirror fields" many of us have been using in Q&A to detect a change in an important field.

The other substantial change is that you can have programming attached to *multiple* events in one field. So you can have on-field-entry programming, general programming and on-exit programming *in the same field* if you want. Far more flexible than Q&A.

Alec Mulvey is a director of Lantica Software, LLC and also owns Keyword Software & Consultancy in Ascot, near London, England. Alec has been building Q&A applications and training clients for 12 years. Keyword Software is the UK distributor for the International English edition of Q&A. alec@keywordsoftware.com, http://www.keywordsoftware.com.

Version Control...cont'd from page 5

```
var file_handle as int

file_handle = FileOpen(filename)
if(file_handle > -1)
{
    FileWrite(file_handle, File0)
    FileClose(file_handle)
}
```

Here I am using *Sesame's* file I/O capabilities to save the "file" field back into the file from whence it came when we first arrived at this form.

Well, that's it. A couple of very simple forms and less than sixty lines of code produce a workable Version Control System. There is one more nicety I added, just to make it a little easier to use. I recorded a simple macro that opens the Add Data form for the login screen. then I created a shortcut that opens *Sesame* with the following command line arguments:

```
sesame –closed –macro open_login.mac
VersionControl.db
```

By using the "-closed" switch, I prevent the user from having access to the command tree, thereby preventing them from adding new logins and passwords. The "-macro" switch is used to load and run a macro when *Sesame* starts. In this case it takes the user directly to the login form, giving them no choice but to login or leave. These finishing touches make *Sesame* act much more like a runtime application than a general database, and simplify it for the intended purpose.

Mark Lasersohn is a principal of Lantica Software, LLC and Technical Lead for the *Sesame* product development team.