

Everybody's Own Daily Follow-Up (To-Do) List



TOM MARCELLUS

WHAT two things are folks most likely to do when they first arrive at their desks each morning? Check their email and review their day's to-do list. Since you don't need Q&A to check your email for you, I thought I'd show you an uncomplicated little technique adaptable to most any shared database that gives the people using that database their own daily list of who they need to follow-up with today. I briefly mentioned this in my July 2001 article, "Add a Pop-up Callback Scheduler to Your Database," and a couple of readers asked me to expand on it. This database feature was originally designed for a small company with a 12-user network, and has proven useful in keeping its people on top of their daily follow-up tasks.

A sample database

We need to start with a database. It could be most any database that tracks things that require periodic follow-up. I've made my sample database really basic to show only the essential elements. (It's included in the download file for Online Edition subscribers.) If you decide you could use something like this, it should be easy to adapt what I'll show you to your own application.

Figure 1 shows a sample record from Cases.dtf. The database is password protected, and each user has a unique User ID and password. The fields to keep an eye on are **Client Name**, **Phone**, **Status**, **Last Activity Date**, **Follow-up Date**, and **Assigned To**, as these are the ones involved in follow-up.

This particular shared database tracks cases that are assigned to different users by an administrator. The assigned user then becomes the contact point for the client and takes on the responsibility for the case's resolution. (Your application requirements will likely be different). Thus the need for periodic follow-ups

The report

The unique thing about this little follow-up system is that whoever runs the *single* follow-up report gets only his or her follow-ups. I mean it's simple

STARTQA - Q&A

Client Name: Bill Jones
Address: 1234 Main St.
City: Costa Mesa
State: CA
Zip: 96755
Phone: 949-766-8900
Assigned To: Tom
Status: Not Good
Last Activity Date: Aug 20, 2001
Follow-up Date: Aug 23, 2001

This database must be password-protected so its report can be run and give pertinent results for each user. The sample records in it are mainly for User "Tom" with password "TJM". See the September Quick Answer for details.

CASES.DTF Retrieved form 1 of -- Total Forms: 7 Page 1 of 1
Esc-Exit F1-Help Alt+F6-Table F7-Search F8-Calc F10-Continue

Figure 1. A sample database record (very basic) designed for periodic follow-up.

The Quick Answer

The Do-It-Yourself Guide to Q&A

September 2001

Volume 12 Number 9

- 1 Everybody's Own To-Do List
Tom Marcellus
- 2 Sesame Seeds—'Sesame' Date and Time in a Client/Server Environment
- 2 Tip—Date Range Searches
- 4 Speedy Fields Revisited
John Dow
- 5 Tip—Windows 2000 Quirks
- 6 @Help—Edited by *Bill Halpern*
 - Fixing 'Bad' Pick-List Values
 - Creating a Formatted String Message
 - Build Me a 'Time-Bomb' Please
 - Windows Me Blues
- 8 Use Macros to Help Program Your Database, Part 2
Alec Mulvey
- 12 Tip—Add a PDF File Viewer to Your Database

MARBLE
PUBLICATIONS

almost to a fault. Each person using the database has a unique User ID, and each record contains a User ID (the person the case was assigned to).

First, the Retrieve Spec:

```
Next Follow-Up Date: <={ @Date }  
Assigned to: {@UserID}
```

The retrieval expression in the **Follow-up Date** field tells Q&A to include just the records that are dated for follow-up today or earlier. Each user, after they've contacted their client on the follow-up date, sets a new follow-up date in that client's record. (See "Add a

Continues on page 3

Date and Time in a Client/ Server Environment

When you and your data are in the same place, determining current date and current time are easy. Whether you are running stand-alone or across a network, as long as the client machine and the server machine live in the same time zone, all is well.

However, if you are a widget company with travelling salespeople who need to enter orders while on the road, it gets a bit trickier. Say you design an Orders database. The intent is for your salespeople to be able to access that Orders database through networking or the Internet and enter orders. You want each record to show the date and time the salesperson entered it.

You put the Orders database on your server, which is located in California. You send your salespeople out to Texas, New York and Colorado. They sell lots of widgets. Returning to their respective hotel rooms later that evening, they all boot up their laptops, log in around the same time, bring up the Orders database, and start entering the orders. They get to the **DateEntered** field and press Ctrl-F5 to insert the current date. They get to the **TimeEntered** field and press Alt-F5 to insert the current time.

Here's where it gets tricky. There are two places 'Sesame' can look for the information. Sesame can ask each Client (salesperson's laptop) or Sesame can ask the Server. If Sesame asks each Client, even though all the orders are being entered at the same time, they will be stamped several hours apart. If the salespeople have been sent to other countries, even the dates could be different. However, if Sesame asks the Server, then all the records will have consistent information, but the salespeople may be confused by what look like incorrect dates and times.

After much consideration, we went for data consistency. When asked for the current date or time, Sesame will use the Server to determine the information. This will have no visible effect on those of you who are running stand-alone. It will have no visible effect for those

of you who run a local network, unless, for some reason, you set your Server to a different locale than the rest of the computers in your office. For those of you who set up Sesame to be accessed over the Internet or a wide area network, when asked to provide the current date or current time, Sesame will ask the Server. This will keep the data itself date- and time-consistent.

For dates and times used by programming, like @DATE and @TIME, we intend to provide functions that the programmer can use to specify which they want— @CLIENTTIME and @SERVERTIME, etc. @DATE and @TIME will still work, but we will provide you with some more specific choices, if you want to use them.

Erika Yoxall,
Technical
Coordinator, Lantica
Software, LLC

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.

Copyright © 2001 by Marble Publications, Inc. All rights reserved. No part of this periodical may be used or reproduced in any fashion (except in the case of brief quotations embodied in articles and reviews) without the prior written consent of Marble Publications, Inc.

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

Q&A is a trademark of Symantec Corp. Other brand and product names are trademarks or registered trademarks of their respective holders.

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

Quick Tip

Date Range Search

You might search for an inclusive range of dates by using a date field Retrieve Spec like this:

>=5/1/2001..<=8/31/2001 [for May thru August 2001]

But as Alec Mulvey recently reminded us, the following simplified spec retrieves the same records:

5/1/2001..8/31/2001

Oregon Connection

Q&A enthusiast, *Quick Answer* subscriber, and regular Q&A User Group Bash attendee **Peter Venuto** will be moving to Medford, Oregon and is interested in getting in touch with other Q&Aers in the area with the aim of starting an informal Q&A user group. If interested, write to Peter at qzqbqq@mindspring.com.

Follow-Ups. . . cont'd from page 1

Pop-up Callback Scheduler to Your Database” in the July 2001 issue for a little doodad that calculates and enters callback dates for you.) Of course, if the user neglects setting a new follow-up date, the same “follow-up” will show up again on tomorrow’s to-do list—a kind of built-in incentive to keep one’s own records current.

The retrieval expression in the **Assigned to** field is the heart of the thing. It further isolates the records to be included in the report, selecting only the clients that are assigned to the person who is running the report.

So, when Jim Smith runs his to-do list, the retrieval parameters tell Q&A: “Hey, show today’s follow-ups for Jim Smith.”

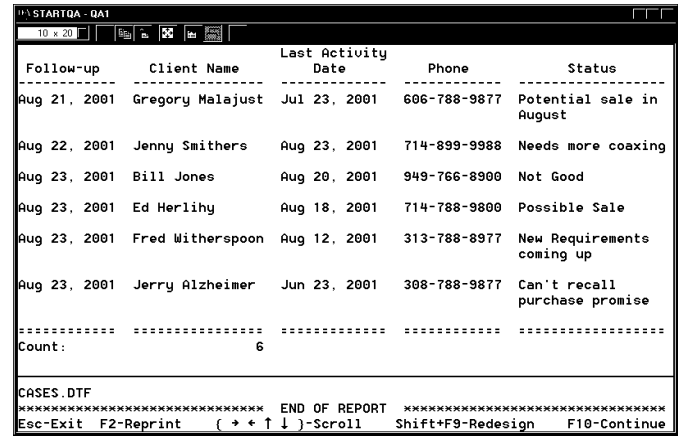
Of course the database administrator can log in as anybody and see that individual’s follow-ups for the day.

The Column/Sort Spec for the report can be as simple as this:

```
Next Follow-Up Date: 1,AS
Client Name: 5,AS,C
Last Activity Date: 6
Phone: 15
Status: 25
```

These specs produce a report (printed to screen is usually best) that’s sorted by ascending follow-up date (so missed follow-ups appear at the top of the list), then by client name. The user can, of course, press F2 from the report screen to print a hard copy.

That’s really all there is to it. Figure 2 shows a sample follow-up report printed to screen on August 23, 2001.



Follow-up	Client Name	Last Activity Date	Phone	Status
Aug 21, 2001	Gregory Malajust	Jul 23, 2001	606-788-9877	Potential sale in August
Aug 22, 2001	Jenny Smithers	Aug 23, 2001	714-899-9988	Needs more coaxing
Aug 23, 2001	Bill Jones	Aug 20, 2001	949-766-8900	Not Good
Aug 23, 2001	Ed Herlihy	Aug 18, 2001	714-788-9800	Possible Sale
Aug 23, 2001	Fred Witherspoon	Aug 12, 2001	313-788-8977	New Requirements coming up
Aug 23, 2001	Jerry Alzheimer	Jun 23, 2001	308-788-9877	Can't recall purchase promise
Count:	6			

CASES.DTF
***** END OF REPORT *****
Esc-Exit F2-Reprint (+ + ↑ ↓)-Scroll Shift+F9-Redesign F10-Continue

Figure 2. A personalized callback list.

Tom Marcellus is editor of *The Quick Answer*.

Instant gratification

It would be neat if each user, first thing in the morning, could have their day’s follow-ups displayed instantly, rather than their having to go through the cumbersome process of keying (1) Report, (2) Print a Report, (3) finding the report in the List of Reports, (4) selecting it and pressing enter, then (5) answering “No” to *Do you want to make temporary changes?* But a shortcut that bypasses these steps isn’t easy.

Access to the database (and its reports) requires a password, and you can’t design a macro that obtains it from the user, verifies it, then runs the correct report. Try creating a macro that verifies a password then goes on to run a procedure, and you’ll see why. If the password fails the verification test, things get messy.

Fortunately, there’s a workaround—and it’s ideal for a situation like this. (Why make users trudge through the steps of running a report when all they want to see are their callbacks for the day?) With Q&A 5.0, you can make the display of each user’s callback report a no-brain one-key operation.

Start by creating a one-field database named Login.dtf. Password protect Login.dtf with the same User IDs and passwords as in the database containing the password-protected report. In Login’s single field at the Program Spec, add an @Macro command like this:

```
< If @Add Then @Macro("Run To-Do List")
```

Record the “Run To-Do List” macro. It should look something like this:

```
<begdef><nokey><name>"Run<sp>To-Do<sp>List"  
<vidoff><esc><esc>rp<capsf4><f10>CASES<enter>  
1.<sp>Daily<sp>Follow-ups<sp>for<sp>User<sp>ID  
<enter>n<enddef>
```

Here, Cases.dtf is the name of the main database, and the report to run in Cases is 1. *Daily Follow-ups for User ID*.

Next, add an Alternate Program (Utilities / Set Alternate Programs) to your Q&A Main menu that looks like this:

```
Alternate program 1: altt  
Menu selection.....: To-do Today
```

Finally, create the Alt-T macro:

```
<begdef><altt><name>"<caps>,altt<caps>"<vidoff>fa  
<capsf4><f10>LOGIN<enter><enddef>
```

With these elements in place, when a user presses “T” at the Main menu, they’ll be prompted for their User ID and password (if not already supplied) for Login.dtf. But they’ll never see Login.dtf, because as soon as their password is verified, the “Run To-Do List” macro is invoked, running their to-do List report in CASES (without another password prompt). With video in the macros turned off, the shortcut will be virtually transparent and seamless.

Speedy Fields Revisited

JOHN T. DOW



BEING the kind of person who attempts to practice what he preaches, I wanted to tell you about a Q&A database I recently worked on that opened my eyes. As I wrote a couple of months ago ("The Mechanics of Speedy Fields," July 2001), I haven't been able to detect any threshold number of speedy (indexed) fields beyond which performance suddenly deteriorates. All I've been able to see is incremental performance degradation as more and more fields are made speedy.

My article concluded that having some duplicate key (indexed) values doesn't cause performance problems, but that massive duplication of key values is the real factor that hurts performance.

Well, I got a real shock when we started testing the application prior to going live with it. The database had 10 speedy fields and performance was atrocious. What I'm referring to is this scenario: If you did a search that brought up a set of records, perhaps 30, and then you pressed F10 to move through the records even without making any changes, the time to move from record to record could be measured in seconds, sometimes more than 30 seconds per record.

Clearly this was not acceptable performance.

I quickly tried to determine what the problem was. *If I removed all the speedy fields, the performance problem vanished.* That was interesting information, but not very helpful as we couldn't eliminate the speedy fields.

I ruled out other factors. It didn't matter whether I used Q&A for DOS or Q&A for Windows. It didn't matter whether DTFWIN was running. It didn't matter whether the database was open two or more times (using multiple copies of Q&A, managed by DTFWIN).

What did matter was whether the database was being shared.

There are three scenarios:

- A) The database is local (running on a standalone PC).
- B) The database is remote (stored on a server).
- C) The database is remote and opened by another copy of Q&A.

Lesson #1

Q&A's performance was best in scenario A.
Performance deteriorated a little in scenario B.

Performance deteriorated tremendously (virtually died) in scenario C.

Note that in scenario C the second Q&A merely has to have the database open; it doesn't have to be doing anything with it at all.

I went through a systematic process in which I removed all the indexes and added them back one by one, testing the performance after each was added. This testing involved accessing the database across the network, with a second Q&A simultaneously having the database open (scenario C).

Everything went fine as I added indexes—until I indexed the telephone area code field. (For this application, we needed to locate people by their area code.) Some of these values were what I call *massively* duplicated. For example, out of the 55,000 records in the database, 3,557 records had the same area code. There were four area codes with 2,000 or more duplicates and nine with 1,000 or more. The remaining area code values had fewer than 1,000 duplicates.

Lesson #2:

In scenario C, fields with massively duplicated values killed performance.

Another field with a similar problem was the state of residence. Clearly this would have even more duplicates than area codes, as there are far fewer states than area codes.

Fortunately, searches on area codes and states were not critical and we could do those searches without making the fields speedy.

But there was another field with massive duplication that was very important—indeed, crucial to the application, and it absolutely had to be indexed. So I needed to find a solution to the problem. And there was a solution. Read on.

Lesson #3:

Minor adjustments to field values can make duplicate fields no longer massively duplicated, making the scenario C performance essentially equivalent to scenario B performance.

Using an area code field to illustrate, I'll explain how to change a massively duplicated field so that it doesn't hurt performance. (In our case it was a different field, but

an area code field is a good example because everyone is familiar with it.)

In the database in question, 3,557 records had the area code value “908.” I appended to the area code a plus sign (“+”) followed by the last three digits of the record number. (“+” was my choice. You might prefer a different character.)

Now, instead of the area code field containing “908,” it might contain “908+034” or “908+629” or “908+001.” When searching, we just need to ignore everything after the “+” sign. For example, Instead of searching for “908,” we now had to search for “908+..”.

Here’s why this works: Out of the 55,000 records, the following ended in “001”: 00001, 01001, 02001, 03001, 04001, 05001, ..., 50001, 51001, 52001, 53001, 54001, and 55001. In other words, there were 56 out of 55,000 records where the field ended with “001.” Essentially, we divided by 1,000 by appending one of 1,000 “random” numbers to each key value. Dividing 3,557 by 1,000 means that this technique resulted in an average of only 3.5 duplicate values per key. When the number of duplicates is this low, it does not adversely impact Q&A’s performance in scenario C.

You might object to messing up a field this way. If so, you can easily keep the original field pristine and add another field that’s out of sight, has the extra characters, and is indexed. So you might have AREACODE (visible) and AREACODEINDEXED (not visible and read only) in your database. You’ll of course have to add the appropriate programming to copy the AREACODE value to AREACODEINDEXED and add the extra characters. But don’t have your programming set AREACODEINDEXED willy nilly; have it update the field

only if the area code field has actually changed.

Otherwise, Q&A will think it has to update the index for AREACODEINDEXED more often than it really needs to.

Finally, you might wonder why I included the “+” sign in the field. It isn’t really needed in the area code field, since all area codes are three characters long. One could easily search on “908..” without any ambiguity. I included the “+” sign to show a generic technique that would be useful if your key values vary in length.

Suppose the field had been last name. In my Pittsburgh, Pennsylvania phone book, I find the following names: Kowal, Kowalcheck, Kowalchick, Kowalczuk, Kowalczyk, Kowalec, Kowalecki, Kowalesky, Kowalewski, Kowalik, Kowalkowski, Kowall, Kowallis, Kowalo, Kowalok, Kowals, Kowalski, and Kowalsky. Searching on “Kowal..” would yield the wrong results. With the “+” character at the end of the actual name, however, you can unambiguously search on “Kowal” simply by specifying “Kowal+..”.

Remember that the indexed value cannot be longer than 16 characters. “Kowalkowski+123” is 15 characters and so would be okay.

Summary

For shared networked databases, avoid massively duplicated indexed fields by appending “+” and an appropriate “random” number to fields in the database, and then append “+..” to search values in Retrieve Specs.

John Dow is the creator of a host of valuable Q&A utilities, including DTFDATA, DTFDOCTR and DTFWIN, and he developed the patch that enables Q&A for Windows to run in Windows 2000. He is also a principal of Lantica Software, LLC. www.johntndow.com, 412-973-9473.

Quick Tip

Windows 2000 Quirks

DOS

I’ve mentioned often enough in my @Help column how much I like Windows 2000. That opinion grows stronger each day I use it. However, while it is a solid and stable operating system, it still has a few little quirks that affect Q&A for DOS users, so let me clear some of them up.

Setting up the shortcut

Windows 2000 thinks that it’s smarter than you and knows what you want to do. So if you try to set up a desktop shortcut by dragging the batch file that runs Q&A for DOS with Lesspace to the desktop, Windows 2000 will not give you all of the options on setting the memory and display that you need to have. The way around this is to drag and paste a shortcut to the desktop for QA.COM rather than the batch file, make all of the setting changes, and *then* change the command line to replace qa.com with the name of your batch file.

Making full screen mode the default

While I don’t particularly like running Q&A in full screen mode anymore, a lot of folks still prefer the large full screen that makes Q&A look like you’re running it in DOS.

Of course, you can still switch Q&A for DOS between full screen and window mode by pressing *Alt-Enter* but, unlike earlier versions of Windows, Windows 2000 won’t remember the way you last ran it. If you want Q&A to run in full screen mode permanently (*Alt-Enter* will still allow temporary mode-switching), right-click on your Q&A desktop shortcut, select properties, and then click on the *Screen* tab. In Screen properties, click on *Full Screen* and then click *OK*. Q&A will always start in full screen mode from that point on.

Bill Halpern



Fixing 'Bad' List Values



I have a problem that I created. I'm looking for an easy solution only because of the amount of time and effort it will take for me to correct it the proper way. Here's my dilemma. I originally started with a simple system that selected a charge rate for my customers from a Userselect pick-list containing "chg1,chg2,chg3,chg4,chg5". The selected rate is used in my orders and inventory databases and lookups. I named and programmed key fields the same as the list selections. As my business grew, I had to update the list to ultimately include 25 different rates. To keep them in the correct ascending order on the pick-list, I named them "chg01,chg02" through "chg25". Now, on any companies with charges through "9," Q&A tells me it cannot find field "01" or "02" and so forth. Do I have to go through all the programming and field naming again?

Bart

Don't feel bad—you're not the first to do something like this, and you won't be the last. It's the benefits of Q&A 5.0's power on the one hand that makes for a potential bane to the developer on the other. It's so easy to make changes to Q&A programming and database designs that we forget our responsibility to document the design and changes so that we know what will happen when we institute a change. So much for the "soap box." Q&A's power to the rescue again. In the programmed field (**Charge**, in your case) where you display the Userselect list, add this to the end of it:

```
; If @Mid(Charge,4,1) = "0" Then #11 =  
@Replace(charge, "0", "")
```

This tells Q&A to look at the selected item, examine the 4th digit to see if it's a zero and, if it is, replace it with nothing. Thus "chg07" will be converted to "chg7" automatically. Because the zero can only be in the 4th position in "chg01" through "chg09," the program will only act on those selections. So there, we've saved you the hours. Now go put them to good use, and be sure to document your design so that you won't have to deal with this kind of dilemma again!

Creating a Formatted String Message

Here's my problem: I need to create a message in my database that assembles some information for my clients. I print their records and send them to them for informational purposes. I want to tell them the date on which I faxed or mailed their record. The message goes in the **Note** field and I want to get the client's name from the **Customer** field. I've tried to do this with the following statement:

```
>Note: If Note = "fax" Then Note = "Faxed to " +  
Customer + " on " + @Date;  
If Note = "mail" Then Note = "Mailed to " +  
Customer + " on " + @Date
```

I had hoped that this statement would create something like this in the **Note** field:

Faxed to Acme Corporation on September 1, 2001

But I get no results. What am I doing wrong?

Kenny

The problem is that you're trying to combine different types of field values without first converting them to a common string format. Q&A can't add dates to text, dates to numbers, numbers to text, or the like. It's like the old apples and oranges problem. How much is "Kenny" plus the date 9/1/2001? The answer is *nothing*—and that's what Q&A is giving you. But how much is "Kenny" plus the text of the characters that make up the date 9/1/2001? The answer to that is: "Kenny 2001/09/01." To convert numbers or dates to text, you have to use Q&A's @Str ("at string") function. The @Str(@Date) value would be 2001/09/01 on September 1, 2001 because Q&A stores dates in a yyyy/mm/dd format. But by separating the parts of the date, you can make the output read *September 01, 2001*. Here's what your programming statement needs to say:

```
> Note: If Note = "FAX" Then Note = "Faxed to " +  
Customer + " on " +  
@Month$(@Date) + " " + @Str(@Rt(@Date,2)) + ", " +  
@Str(@Lt(@Date,4));
```



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.

```
If Note = "mail" Then Note = "Mailed to " +
Customer + " on " +
@Month$(@Date) + " " + @Str(@Rt(@Date,2)) + ", " +
@Str(@Lt(@Date,4))
```

Or:

```
> Note: If Note = "FAX" Then Note = "Faxed to " +
Customer + " on " +
@Month$(@Date) + " " + @Str(@Day(@Date)) + ", " +
@Str(@Year(@Date));
If Note = "mail" then Note = "Mailed to " +
Customer + " on " +
@Month$(@Date) + " " + @Str(@Day(@Date)) + ", " +
@Str(@Year(@Date))
```

This will give you the fax message:

```
Faxed to Acme Corp on September 01, 2001 (or
September 1, 2001)
```

Or a Mailed note of:

```
Mailed to Acme Corp on September 01, 2001 (or
September 1, 2001)
```

When you need to combine text values with other types of values to generate a text string, liberal use of the @Str function with the non-text values usually does the trick.

Build Me a Time-Bomb, Please

I have a very sticky problem. I'm in a business that depends on outside salespeople. They need to have copies of my Q&A databases with them, so I allow them to copy them to their portable computers when they leave the office. But because I never know when they might take another position—possibly with one of my competitors—I'd like to take measures to keep my data safe. Is there some way I can make the database self-destruct after a certain time? This is extremely important to me. Any help is appreciated.

Ron

I have mixed feelings about telling you how to do this. It can be very, very dangerous. First of all, let me point out that there is no *sure* way to automatically destroy your data. The sophisticated user will always find a way around the protection. However, for the casual user, what I'm going to suggest should work. Remember, you will always need to be aware of the dates and keep a good set of backups. If it were my system, I'd make sure that this procedure was run only on a *copy* of the database to be given to people who need to have it outside the office.

First, you want to make sure that the database(s) are password protected and that the salespeople and others *cannot* program them. Assign them the password *User*.

Next, you will want to number *all* your fields in the Program Spec. (For this example, let's assume there are 100 fields). In the first field of the database add this programming statement:

```
<#1: If @Date > @Td("Put Destruct Date Here") and
@Userid = "User" Then { Clear(#1..#100); @Save }
```

After the "destruct" date, as long as the user is logged in with the salesperson's password, this program will clear all the fields, save the record, then get the next record and do the same thing. It will keep clearing and resaving the records until the Esc key is pressed. The result will be a database containing all blank records. This will only work in Q&A 5.0. In earlier versions that don't have the @Save function, it must be left out and all that will happen is that the data in the current record won't display on the screen. (You'll still be able to export or copy the data.) Plus, it will only work when *searching* the database. You'll have to be very diligent in keeping the statement current so that valid users aren't affected by it. *Careful, careful, careful!*

Windows Me Blues

I've followed all the instructions at quickanswer.com and related Web sites about running Q&A on my Windows Me PC. Q&A seemed to install and run fine for a while. But now I keep getting random, non-repeatable lock-ups and slowdowns. Is there any more information you can give me to resolve this problem?

Fred

Windows Me has become a real sore point with me. I am extremely unhappy with it. At first, we thought that many of the memory and lock-up issues came from hardware configuration in the newer machines. While this is true in certain PCs, we've found that most of the issues have to do with the operating system itself. On PCs that could not get Expanded memory from Windows Me, simply changing the O/S to Windows 98 SE brought the Expanded memory back. It had nothing to do with the hardware! It was Windows Me! Even Microsoft, right on the box, states that Me is a *Home* operating system. They recommend Windows 98 or Windows 2000 for business use. And Q&A is not the only issue! We have had accounting systems, CD burners, Office programs, networks, and the like simply lock up in Me. The only things that seem to run correctly and consistently are games, graphics and music programs. More than one manufacturer has told me that they do *not* recommend installing their software in Me.

My friend—you are in O/S hell! You cannot "upgrade" to Windows 98 because it is older than Me and will not install on an Me PC. However, Windows 2000 will. For some of my clients, I've had to Fdisk the drive, reformat it, and start from scratch with a fresh Windows 98 installation. I have no quick answer (no pun intended) to your problem. The approach I've taken in almost every case is to remove the entire Windows Me operating system and replace it with 98 or 2000. Everything is then stable and runs as expected. Good luck!

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.

Use Macros to Help Program Your Database—Part 2

ALEC MULVEY

DOS



IN Part 1 of this article in the June 2001 issue, I showed you a few of the ways I employ macros to help automate the process of repetitive programming in a database. You learned how this can save time, reduce drudgery, and avoid those all-too-easy typos and other mistakes. This month, I'd like to walk you through a case study—an actual job I had to do recently that utilized this “temporary macro” technique not only for repetitive programming but also for redesigning a database form.

Review of essential techniques

Efficiency with temporary time-saving macros comes with a familiarity with the range of Q&A's word-processor keyboard shortcuts. There's quite a few of them, and they can come in very handy. A table of these shortcuts is available by pressing F1 in Write. I also suggest the following general guidelines when programming:

- Do not use field ID numbers for on-field exit programming. Use field names instead.
- Name fields logically, using the line or line-item number as the suffix.

There will be examples of these in the case study I'll show you.

The scenario

The client, “XYZ Co.,” is an engineering and quality assurance firm. On a set schedule, they go out to factories to perform tests, including taking samples for laboratory testing. The database we'll look at is used to store details of visits for the coming year and the tests to be performed. Beyond that the details are of no concern to us here.

Figure 1 shows page two of a sample record from the original database form.

The basic task I had with this database was to double the number of testing jobs to be done (“1 Fee” through “4 Fee” in the Figure 1 example) to allow up to eight test samples.

However, along the way I saw several other improvements worth making:

- The “No” field—a quantity field—should be right-aligned.
- The “Type” field and the un-labelled field (“162..”) were not wide enough.
- The last field (XYZco) would benefit from commas separating the thousands.
- In the lower half of the screen, the Income column precedes the Expenditure column—the opposite of the section above it. (The Lab column is net cost per item to the XYZ company and the unlabelled column is the price charged.) Consistency was clearly desirable.

There might be other ways to improve the layout or design, but in a developer's role, any desire to perfect a design must be tempered by the constraints of not making radical changes that users of the application could find uncomfortable.

Of course, you can correct alignment and commas in a matter of seconds (File / Design / Customise / Format Values). Adding the commas code (N, C or M,C) makes the field contents one character wider, so this field also needs to be wider.

Adjusting “column” widths

The first task was to make the columns the correct width. The way I did this was to find a suitable sample record, one with lengthy data values and a full complement of filled-in fields (sometimes it is necessary to create such a record with dummy data) and then get the first row correct by trial and error. Once it was the way I wanted it, and saved, then the others could be adjusted accordingly—manually or by using a macro.

In this example, you will see that the previous developer used the “>” field terminator character to limit

Q & A v1									
Tests / Visits									
CLIENT: Marble Rubber Inc									
SCHEME: A									
1 FEE: 00AS	No: 33	TYPE: SCHEME →	LAB: 126.00	162. →	XYZco: 5346.00				
2 FEE: 00AM	No: 32	TYPE: SCHEME →	LAB: 141.00	189. →	XYZco: 6048.00				
3 FEE: 00AST2	No: 5	TYPE: SCHEME →	LAB: 162.00	202. →	XYZco: 1010.00				
4 FEE: 00AMT2	No: 5	TYPE: SCHEME →	LAB: 219.00	231. →	XYZco: 1155.00				
TOTAL	No: 75		LAB: 10575.00		XYZco: 13559.00				
Job number 1: R1RTA001	Income 1: 3660.93	Expenditure 1: 2855.25							
Job number 2: R2RTA001	Income 2: 3660.93	Expenditure 2: 2855.25							
Job number 3: R3RTA001	Income 3: 3660.93	Expenditure 3: 2855.25							
Job number 4: R4RTA001	Income 4: 2576.21	Expenditure 4: 2009.25							

Figure 1. Page 2 from the original database design. Notice the four “Fee” rows.

the width of most fields. This is fine if you want additional spaces between columns for readability. Here, though, we needed the extra width, so they would be removed or adjusted.

Figure 2 shows the “before” design, and Figure 3 shows it “after.”

The next tasks were to (1) Adjust the other three rows so that they were like the first row, and (2) Add another four rows to wind up with a total of eight. There was more than one way to do this. One approach might have been to just copy four more rows like the existing ones, then use a macro to adjust the other seven to be like the first. The snag with this was that the four new rows couldn't have Q&A's internal field codes (“AK” for example). This meant that the macro would have to do different things to some rows than to others. The way I would normally do this is to first use a simple temporary macro to adjust rows 3 and 4, then copy one of these rows, delete the field codes from the new fifth row, then copy that three more times. Let's look at this in detail.

Recording a temporary macro like this was covered in Part 1 of this article. In this case, it is fairly simple, a matter of navigating left and right along the row and adding and deleting spaces. Here are a few guidelines on this:

- Be consistent with your starting and ending points. I would start the macro at the left margin of the row to be adjusted, and finish it by pressing the Down Arrow key once. This way, if the next line is to get the same treatment, you simply run the macro again. If it isn't (for example if there is a page break or other fields intervening), you just press down arrow a few times to get to the next starting point.
- Make the macro's first keystroke the Home key. This way you don't have to be particular about being at the very left margin—it establishes your starting position.
- Take control of overtype/insert mode. In Write as well as at the database Design/Redesign screen, Q&A will stay in the mode you set. In the database though (during data entry), Q&A's native mode is overtype, and it will revert to this every time you tab to another field. The Insert key merely toggles you between the two modes—which is fine for switching to overtype

mode when your macro has just entered a new field, because you know what mode you are in. Unknown to many Q&A users, there is a shortcut key that always puts you in Insert mode—*Alt-I*. Although there is no corresponding key combination that sets overtype mode, you can always press Alt-I followed by the Insert key to do so. It is the usefulness of this key combination that has always led me to discourage users from using Alt-I as a macro key identifier. Without it, there is no way of *absolutely* setting insert or overtype mode.

- Use the Q&A navigation keyboard shortcuts. For example, Ctrl-Left and Ctrl-Right take you one word to the left or right—very useful for this macro.

Here's the macro, assigned as before to the key to the left of the “1” key:

```
<begdef>'<name>"temp"<vidon><home><alti><ins><ctrlrgt><ctrlrgt><del><ctrlrgt><ctrlrgt><ctrlrgt><ctrlrgt><sp><ctrlrgt><ctrlrgt><rgt><del><ctrlrgt><sp><ctrlrgt><ctrlrgt><sp><alti><sp><sp><home><dn><enddef>
```

Once recorded, row two is adjusted like row one, and it takes literally two seconds to adjust rows three and four. Of course, the saving in time and effort is greatly increased if you have six, 10 or 15 rows to adjust.

The next task was to add the new rows (line items) five through eight. You don't need a macro for this part. Put the cursor on row five, then:

1. Press Home to move to the left margin.
2. Press F5 (copy).
3. Press End to select to the end of the row.
4. Press F10 to copy the data.
5. Press Home then Down Arrow to move the cursor to the beginning of the row below.
6. Press F10 to insert the data. You now have row four repeated.
7. Change the row heading from “4” to “5.”

Once you have a clean row five with all the codes removed, this row then needs to be copied three more times. Do *not* press F10 to save the form here—doing so will add field codes to row five which you will then have

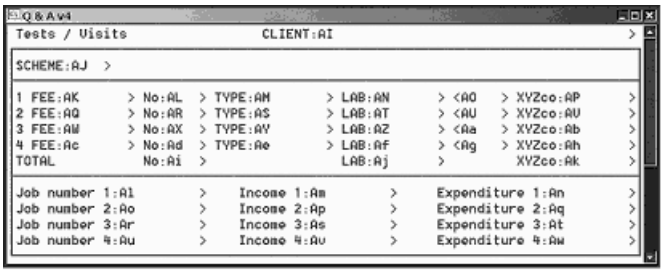


Figure 2. Design view of the original layout

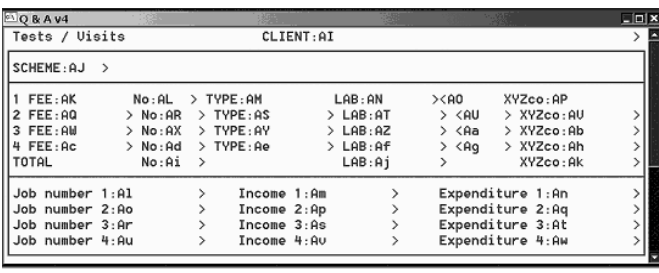


Figure 3. After adjusting just the first row.

to remove again. If, like me, you prefer to constantly save as you are working, you'll just have to sit on your hands at this stage!

With the cursor at the left margin of row five, press F5 to copy the row, press End, then press F10. Next, press Enter to move to the row below and press F10 to "paste" the row. This will be row six.

To create rows seven and eight, there's no need to "copy" again. The row you've just copied is still in Q&A's unseen internal clipboard. (See the sidebar.) To "paste" the row again, simply press Shift-F7. Another Shift-F7 and you've got row eight as well. Number the rows correctly and press F10 to save the design.

At the Format Spec

Filling in the Format Spec is a good example where a simple macro can save time. Rather than just entering the first new row's format codes, record a temporary macro as you do so. The remaining rows will then take a second each. This is particularly useful in Q&A 5.0, where you often have to press F6 to expand a field, then press F6 again to close it.

Here's the macro for the Format Spec:

```
<begdef> `<name>"TEMP"<vidon>TU<tab>N<sp>JR<tab><tab>M<sp>JR<tab>M<sp>JR<tab>MC<sp>JR<enter><enddef>
```

The last design change is to reverse the income and expenditure columns in the lower half of the screen. In fact, they can be positioned directly below the columns from which they are derived, which is much more user-friendly. This is another good example of where a macro can assist. This macro swaps the fields around and aligns them with the columns above:

```
<begdef> `<name>"TEMP"<vidon>TU<tab>N<sp>JR<tab><tab>M<sp>JR<tab>M<sp>JR<tab>MC<sp>JR<enter><enddef>
```

With a couple of bits of fixed text to serve as improved column headings, that completes the design changes. Now we need to program the new fields. Unfortunately, with Q&A you cannot copy a field and all its attributes.

A fresh look at the programming

The original developer's programming in this database was far from optimum. Every field had a Logical Field

Tests / Visite	CLIENT:AI
SCHEME:AJ >	
1 FEE: AK	No: AL > TYPE: AM LAB: AH > <AO XYZco: AP
2 FEE: AR	No: AR > TYPE: AS LAB: AT > <AU XYZco: AU
3 FEE: AR	No: AR > TYPE: AS LAB: AZ > <AA XYZco: AB
4 FEE: AR	No: AR > TYPE: AS LAB: AF > <AG XYZco: AH
4 FEE: AR	No: AR > TYPE: AS LAB: AF > <AG XYZco: AH
TOTAL	No: AI LAB: AJ XYZco: AK

Figure 4. One new row—removing Q&A's field codes.

More on the QA4 / QA5 Internal Clipboard

The clipboard in use here is not the expanded F11/F12 clipboard found only in Q&A 5.0. Neither is it the Windows clipboard. It functions similar to both, though. It acquires its contents whenever a "block" operation is used in Q&A Write. These block operations include: F4 (delete word), Shift-F4 (delete line), Ctrl-F4 (delete to end of line), F3 (delete block), F5 (copy), Shift-F5 (move), and so on. In fact, after any operation in Write using the F3, F4 or F5 key (press F1 in Write for a full list). The "clipboard" contents are retained until replaced with new text or until you exit Write. This means that Shift-F7 can be used to move or copy text between documents.

Number (LFN); every field with derived values had its own programming, no field names were used, and there were other minor deviations from best practice. For example, here's the programming in one of the totals fields:

```
#1160:  
#1160=(#1120*#1090)+(#1140*#1100)+(#1202*#1194)+(#1050*#1224);  
if #1160=0 then #1160=""
```

Not easy to make sense of, eh? Notice the superfluous "#1160:"—it isn't needed when the field starts with "#1160 = " anyway. The final part after the ";" clears the field to suppress a zero total, which is quite legitimate. As well as being hard to construct and hard to edit, the use of LFN's exclusively is that you run into field numbering problems. (This field calculates this field based upon this and that—only that field has not itself been updated because it has a lower LFN.) You can get into convoluted rounds of re-numbering fields to overcome this. The smarter way is to use field names wherever possible. As I showed you in Part 1, this also gives you the benefit of being able to perform, with programming, simple arithmetic on groups of fields if you use the same field names with incremented numeric suffixes. (Field1, Field2, and so forth.) This is a good example. Figure 5 shows how I applied the more logical field names to the fields.

1 FEE: FEES11	No: No11	TYPE: TYPE11	LAB: LAB11	XYZco: XYZco	XYZco: S11
2 FEE: FEES12	No: No12	TYPE: TYPE12	LAB: LAB12	XYZco: XYZco	XYZco: S12
3 FEE: FEES13	No: No13	TYPE: TYPE13	LAB: LAB13	XYZco: XYZco	XYZco: S13
4 FEE: FEES14	No: No14	TYPE: TYPE14	LAB: LAB14	XYZco: XYZco	XYZco: S14
TOTAL	No: T0		LAB: LAB1	XYZco: GROSS1	

Figure 5. New Field Names.

(This naming scheme is not perfect: the field named XYZ11 is not the one labelled XYZco, for example, but there were constraints concerning existing names. There are other pages where similar fields are numbered 21, 22 and so forth.) See the sidebar.

Reprogramming

Another advantage of using Field names in programming is that, to a certain extent, you can “copy and paste” programming from one field to many other fields, changing only the numeric suffix. I covered this in Part 1 of this article.

You cannot readily adapt: #455 = #360 * #682

But you can easily change $S21 = XYZ21 * No21$ if the corresponding fields on the next line are S22, XYZ22 and No22.

There is no point going into the programming details. Rather than have each field contain programming that derives its value, I observed the rule that when several fields have values that are derived from an entry in one other field, then all the calculations should be in the field that has the entry typed into it, as an on-field-exit program. Then the derived fields can be made read-only so they can't individually be changed from the keyboard.

So, the programming in the first field (Fees11) became:

```
>TYPE11 = @XLU("SERVICES", FEES11, "key", "3");
LAB11 = @XLU("SERVICES", FEES11, "key", "1");
XYZ11 = @XLU("SERVICES", FEES11, "key", "2")
```

And the programming in the last field (S11) became:

```
#1030:
S11 = XYZ11 * No11
```

Note that a LFN (#1030) is required here, as it must be a general programming statement that will recalculate instantly if either the Item type or the Quantity is changed.

These are then copied into the corresponding fields below. All the other programming in these fields, and all the now-redundant Logical Field Numbers in the fields No11, Type 11, Lab11 and so forth, are removed. This leaves just the two chunks of programming above per row.

The totals can also be re-programmed along these simplified lines:

```
#1110 = @SUM(No11..No18)
#1130 = @Sum(S11..S18)
```

Unfortunately, I know of no easier way, in Q&A, of simplifying the “array” programming in the LAB1 field:

```
#1120 = @SUM(No11 * LAB11, No12 * LAB12, No13 * LAB13, No14 * LAB14, No15 * LAB15, No16 * LAB16, No17 * LAB17, No18 * LAB18)
```

As for the Expenditure and Income fields, when I first

Renaming Fields

Take care when renaming fields. Programming in the same database that refers to the renamed field *by name* will be automatically updated, as will derived column formulas in reports. However, XLookups in other databases and, in particular, field codes in mail merge documents, will not be automatically updated and will need to be updated manually. Take care that all the implications of renaming fields are evaluated beforehand. Having said that, the benefits of applying sensible field names and using them universally in programming are, I consider, overwhelming.

got hold of the database, there was a programming statement in each of the 12 fields. In fact, all that is happening is that the costs are being spread among the four visits per annum—or they were. The client now wants four visits per year for home customers and three per year for overseas customers. So I added a field to the CLIENTS database that fills in a new field UKO (UK/Overseas) according to whether there is a value in the “Country” field in CLIENTS. This then references an internal Lookup Table in the database in question which contains the figures that determine how the costs and fees are to be split. It is then easy for the client to modify these as needed without getting involved with programming. The programming for all these fields was then put into just the one field, Job Number 1.

```
#1300: IF FEES11 <>"" THEN {
Job11 = @LEFT(BASE JOB NO,1) + "1RT" + SCHEME1 +
@RIGHT(BASE JOB NO,3) ;
Income11 = GROSS1 * @TN(@LU(UKO,2)) / 100 ;
Expenditure11 = LAB1 * @TN(@LU(UKO,2)) / 100 ;
```

...to...

```
IF @LU(UKO,1) = "4"
THEN {
Job14 = @LEFT(BASE JOB NO,1) + "4RT" + SCHEME1 +
@RIGHT(BASE JOB NO,3) ;
Expenditure14 = LAB1 -
@Sum(Expenditure11..Expenditure13) ;
Income14 = GROSS1 - @SUM(Income11..Income13) }
ELSE Clear(Job14; Expenditure14; Income14)
}
ELSE Clear(Job11..Job14;
Income11..Income14;
Expenditure11..Expenditure14)
```

Conclusion

The necessary design changes were made much simpler by recording, along the way, several temporary macros. Each was given the macro keycode “” —a key that proved to be as convenient as it is unused—and each was named “temp,” with no intent to save it. Because this helped me avoid agonising over what to name it and where to save it, it made everything simpler and more practical to use

these temporary macros. The benefit is not only in time saved but in the very considerable gain of avoiding the typos that are almost inevitable in doing it manually. Make temporary, record-as-you-go macros part of your day-to-day database design and programming.

The screenshot shows a window titled "Q & A v4" with a menu bar and a toolbar. The main area displays a table with columns for "SCHEME: #1", "(each)", "(each)", and "(Total)". The table contains 8 rows of data, each starting with "FEE: >TYPE1 No:" followed by "TYPE:", "LAB:", and "XVZco:". The bottom row is "TOTALS..". Below the table, there are four sections: "Job number 1: #1300: IF+", "Expenditure 1:", "Income 1:", "Job number 2:", "Expenditure 2:", "Income 2:", "Job number 3:", "Expenditure 3:", "Income 3:", and "Job number 4:", "Expenditure 4:", "Income 4:".

Figure 6. The finished programming. Much neater!

Alec Mulvey is a director of Lantica Software, LLC, and also owns Keyword Training & Consultancy in Ascot, near London, England. He 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 and John Dow's Utilities. Fax +44-1344-884-111, alec@keywordtraining.com, www.keywordtraining.com.

Quick Tip

Add a PDF Viewer to Your Database

As this issue was about to go to press, we received an interesting email from a Q&A 4.0 user with a "wish list" for the coming "Sesame" database manager. The fellow said that his business relies heavily on Acrobat PDF files and expressed hopes that Sesame would let him link his PDFs to his database records so he could, with a mouse click, view any linked PDF.

Not only will Sesame let you do this, heck, you can do it today in Q&A 5.0. If the record contains the full path to the linked .pdf file in a **PDF** field, simply program **PDF** this way:

```
< If PDF <> "" Then @Msg(
@Shell("d:\acrobat4\reader\acrord32.exe " + PDF))
```

When you click on the **PDF** field, Q&A will launch Acrobat Reader and display the linked PDF file without having to leave the record you're on. Your path to Acrobat Reader will likely differ. And remember to use DOS abbreviations for folders and filenames that don't meet the 8.3 DOS convention. For example, *Program Files* would be *Progra~1*.

The
**Quick
Answer**
online

Visit Our Web Site at
www.quickanswer.com

- Complete Back Issue Index 1990 to date
- Comprehensive Topic Index 1990 to date
- Info on Online Subscription Benefits

- Info on the Q&A successor product
- Q&A Tips and Ideas
- Searchable Solutions Database
- Dozens of Free Files to Download
- Full Database Applications
- Q&A Consultants Directory
- Links to Useful Web Sites & More!

Find out about savings and benefits when you switch to the Online Edition

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

September 2001

First Class