The Quick Answer

The Do-It-Yourself Guide to Q&A

January 2000

Volume 11, Number 1

MARBLE PUBLICATIONS

Print Your Merge Letters On-the-Fly

TOM MARCELLUS

DOS 5.0 Application in ONLINE Edition

If your business relies on correspondence to make things happen, automating your merge-letter printing can improve productivity dramatically.

PICTURE this: You're looking through your client database records. You come to somebody who needs "follow-up letter 3A." So you pop-up your list of merge letters and click on *Letter 3A.* Q&A prints it on-the-spot then automatically adds a note to the record that it has been sent. You press F10 a few more times until you come to a client who needs "New Gizmo Bulletin 2C." So you print that letter for that client the same way. In a few minutes, your correspondence chores are done for the day and you can move on to other things.

If this sounds like an easy way to print routine merge letters—it is. While viewing anyone's record, you can select any letter to print and Q&A will immediately print that letter merging the data from that record. No Write/Get. No figuring out which filename it is. No Retrieve Spec.

Genesis of the technique

Last summer I upgraded a company from Q&A 4.0 to 5.0. They run a 12-user network where everyone works primarily in a big database I'll call Case.dtf.

Each Case record contains many details about a person and a hospital where they received treatment. In addition to the usual contact information on the person, the Case records contain a host of other data, much of which gets updated as the case progresses through various stages. The database includes a field that identifies the staff member to whom the case is assigned, a status field to denote the last action taken on the case, and a follow-up date.

New cases (persons) are often contacted first by letter. As the case progresses, the staff member working it might need to send several *other* letters to the person. Any of a dozen or more *different* letters might be needed depending on the type of case it is and what developments occur during the course of it. Each letter is a canned Q&A merge letter—that is, the text of the letter is fixed, with the variable information (the person's name and address and several other fields) merged from the person's database record at printing time.

To improve productivity, the supervisor wanted staff members be able to select and print any merge letter for the person whose record they're working on without having to leave the record. A staff member might need to update the person's record with some new development, print an appropriate next letter for the person, edit something else in the record, save it, and then move on to another case. They wanted the flexibility of being able to print any merge letter on demand—without having to switch from File to Write, open a document, or fill out a Retrieve Spec.

We did it—and I'll show you how. You too can build an instant letter-printing capability into any Q&A 5.0 database. Once you have everything set up, you'll marvel at how easy it is to print your routine merge letters.

A letters database

I'll assume you already have an assortment of merge letters that you might want to print by selecting them from a pop-up list when you're viewing records. You could have just a few such letters, or a whole slew of them—this approach can accommodate virtually any number of documents as long as they're all in the same directory.

To make it easy to add and remove letters from your letters pick-list, you'll use a database (Letters.dtf) to store their descriptions and filenames. (See Figure 1.)

Continues on page 3

Quick Tip

Good Morning Y2K

Now that "Year 2000" issues are no longer in the future, here's a recap of the essentials concerning Q&A and dates in the 21st century.

Most importantly, ***for each database with a date field, you must set a global date format that includes the full four-digit year.*** (In Q&A DOS, select File / Design / Customize / Format Values, then press F10 at the Format Spec for the Global Format Options screen). This is *not* optional—errors will result from using a two-digit year format. Europeans should set a format that has the day before the month.

This way, beginning January 1, 2000, as long as your computer knows the year is 2000 or later, then the following functions will correctly return 21st century dates:

- Using @Date as an initial value.
- Using @Date in programming.
- Pressing Ctrl-F5 to auto-fill a field with the current date.
- Entering a date and specifying the year in full; for example, 11/2/2000.
- Entering a date but omitting the year; for example, 11/2 or 11 Feb.

The important points to remember are these:

- If you do not include the year when entering a date, then Q&A will assume the current year according to your PC's internal clock.
- If you use a two-digit year, Q&A will assume the 1900s.

You will probably need to specify full dates in Retrieve Specs, including Report Retrieve Specs. A Retrieve Spec to include all sales for the last two quarters could be typed like this—Sale Date: **1 Oct 99..31 Mar**—if the current date were in April 2000. And to include, in April 2000, all service calls booked for the following year, your Retrieve Spec could be typed like this: Service Date: **1 Apr..31 Mar 2001**

Also keep in mind that dates displayed when a file is highlighted at Q&A for DOS' list of databases screen will show incorrect years unless the Y2KFIX patch is applied; and QABACKUP, the backup utility that comes with Q&A for DOS, will not work beyond 1999.

Correcting Bad 21st Century Dates—Another Option

I like the following program for correcting abbreviated date entries such as 12/3/01:

```
> If @Year(Date) < 1950
Then Date = @Replace(Date, @Yr(Date), @Yr(Date) + 100)
```

To Q&A, 12/3/01 is a 1901 date, but the program corrects it to a 2001 date. However, some clients have asked me to modify the program because it won't let them enter an invalid date—Q&A simply clears the entry. To me, this is desirable, but some people just *insist* on being able to enter non-dates like these:

15 Jan 00 (or the week after)
A.S.A.P.
Oct 96

These, of course, are all invalid entries for a date field, and would normally display Q&A's red error message: "*This date can't be reformatted -- Is it a valid date?*" I consider this tighter control to be a good thing, and view such entries in a Date field as evidence that an additional text field should be added for them. If, however, you want a Date field to behave in its normal manner, make the program conditional on the date being after 1900 as well, like this:

```
> If @Yr(Date) < 1950 and
@Yr(Date) >= 1900
Then Date = @Replace(Date,
@Yr(Date), @Yr(Date) + 100)
```

Alec Mulvey

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, $89; Outside the U.S., $109; Online edition, $59. Single copy price: $11; outside the U.S., $12.50. All funds must be in U.S. currency. Back issues are available upon request, for the same price as a single copy.

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

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

On the Web at www.quickanswer.com

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**

Merge Letters...cont'd from page 1

Letters.dtf requires just two ordinary text fields named *Description* and *Filename*. The descriptions will appear on the pick-list, while the filenames will be those of the actual documents. Q&A 5.0's pick-lists (Userselect, XUserselect) can display only up to 19 characters of the selections on them, so make the Description field 19 characters. As for the Filename field, be sure each filename you enter conforms to the *eight-dot-three* DOS filenaming convention and that an actual Q&A Write merge document with that same filename exists.

At the Speed-Up Spec, make both fields Speedy/ Unique by typing *SU* in them.

Add a record to the Letters database for each merge letter that you want to appear on your letters pick-list in the main database.

The main database

The main database (referred to earlier as the "Case" database) is the one that contains all the data on your clients, customers, vendors, patients—or whatever. (Figure 2 shows a bare-bones sample database that's included in the download file for Online Edition subscribers.) This is also the database where you'll select and print the merge documents you've added to your Letters database. In addition to whatever other fields are in the database, you'll need the following fields, at least for the sample application I'm using to illustrate:

```
Record No
Name
Address
CSZ
Notes

SendLetter
WriteMacro
ShellLetter
Temp1
Temp2
```

The *Record No*, *Name*, *Address*, *CSZ* (City, State, Zip) and *Notes* fields are assumed to already be in your database. The *Record No* field can be named anything, but it must contain a text value or incremented number (such as @Number) that *uniquely* identifies each record in the database. (The approach we're employing to print merge letters uses this unique record identifier to select the record to merge at printing time.)

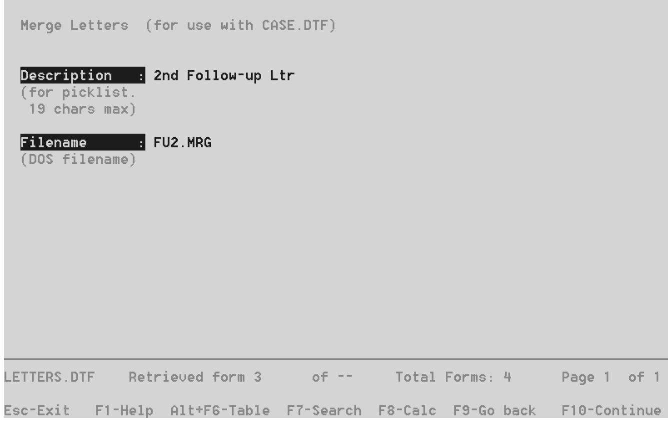
As for the other fields, you might have fields for *First Name* and *Last Name* instead of just a *Name* field. You might have two address fields instead of one. Or, you might have separate city, state, and zip code fields instead of a single *CSZ* field. In the programming I'll show later, you'll have to take any differences into account and revise your programming accordingly.

The *SendLetter* field (its field label can be *Send Letter?*, but its field name should be *SendLetter*) is coded at the Format Spec with *YU* for *Yes/No-Uppercase*. Use the Initial Values Spec (*File / Design / Customize / Set Initial Values*) to set its default value to *N* for *No*. Adjacent to the *SendLetter* field are the *WriteMacro* then the *ShellLetter* fields. Both can be just one character wide with no field label.

On the last page of the form, hidden from view (though not hidden in my sample database) are the *Temp1* and *Temp2* fields. Labels for these text fields are optional, and they can be one character wide and colored to match the form's background so they're invisible. But they must be named *Temp1* and *Temp2* in the Field Names Spec for the programming I'll show.

Because the merge printing process will need to access this database when it's already open (you'll see why shortly), you'll need to set its sharing mode to *Allow* even if it's not being shared on a network. To do this, select *File / Design a file / Secure a File / Declare sharing mode*, and set the Sharing Mode to *Allow*. Exit and restart Q&A for the new setting to take effect.

Following is the required programming. Revise the code as necessary to accommodate your own field names and requirements.



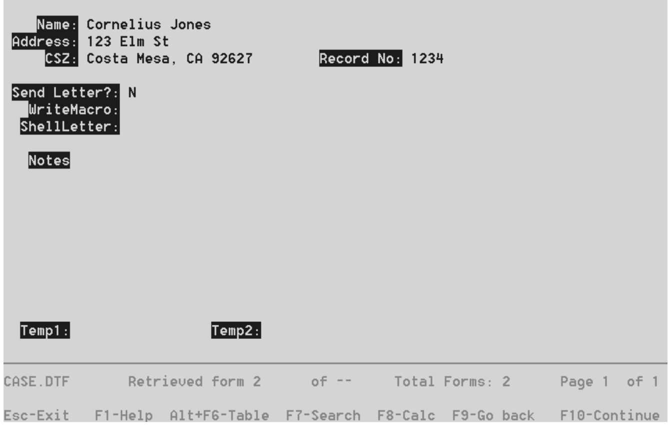
```
Merge Letters  (for use with CASE.DTF)

Description   : 2nd Follow-up Ltr
(for picklist.
 19 chars max)

Filename      : FU2.MRG
(DOS filename)

LETTERS.DTF   Retrieved form 3     of --    Total Forms: 4     Page 1  of 1
Esc-Exit  F1-Help  Alt+F6-Table  F7-Search  F8-Calc  F9-Go back  F10-Continue
```

Figure 1. Use a simple letters database like this one to store the descriptions of your merge letters (as you want them to appear on your pick-list) along with their DOS filenames.



```
   Name: Cornelius Jones
Address: 123 Elm St
    CSZ: Costa Mesa, CA 92627       Record No: 1234

Send Letter?: N
  WriteMacro:
 ShellLetter:

  Notes




 Temp1:                 Temp2:

CASE.DTF      Retrieved form 2     of --    Total Forms: 2     Page 1  of 1
Esc-Exit  F1-Help  Alt+F6-Table  F7-Search  F8-Calc  F9-Go back  F10-Continue
```

Figure 2. This sample database shows just the essential fields. Your own database will have many more.

SendLetter

```
> If SendLetter = “N” Then Cnext;

If @XLookup(“Case”, Record No, “Record No”,
“Record No”) = “” Then {
@Msgbox(“ “,”Record must be saved before
printing”,” “);
SendLetter = “N”; goto SendLetter };

If SendLetter = “Y” Then if Name = “” or Address =
“” or CSZ = “”
Then { @Msgbox(“Complete Mailing Address
required.”, +
“Name, Address and CSZ fields”,”must be filled.”);
SendLetter = “N”; Cnext };

If SendLetter = “Y” then {
Temp1 = @XUserselectR(“Letters”, “Description”,
“Filename”, “”, “”);
Temp2 = @XLookup(“Letters”, Temp1, “Filename”,
“Description”) };

If @Askuser(“Print “ + Temp2 + “ (merge letter)
for”,Name + “?”,” “)
Then {
WriteMacro = “<begdef><alt1><name>” + @Chr(34) +
“<caps,>alt1<caps.>” +
@Chr(34) + “<vidoff>wg<capsf4><f10>” + Temp1 +
“<enter><f2><f10>tom<enter>tom<enter>” + “
“ + “{File No =” + @Chr(34) + Record No + @Chr(34)
+ “}” + “ “ + “<f10><enter><esc><esc>x<enddef>” +
“
“ + “*” }
Else Send Letter = “N”
```

To display the pick-list of merge letters to print, you’ll enter a “Y” in the SendLetter (*Send Letter?*) field. (The field’s default or initial value is “N.”)

Because you’re merge printing information from the displayed record, that record must exist in the database. This means that a *new* record must be saved before printing. Accordingly, the first thing the program does is check to make sure the record is on disk. If it isn’t, you’re prompted to save it before printing.

(It’s a good idea to always re-save a record before printing a merge letter for it. This way, any changes just made to it are on disk, ensuring that the letter includes the most recent data.)

The program next makes sure the name and address fields are filled so that the merge letter will be properly addressed. Have your own program check here to make sure that any other merged fields are filled. You don’t want your letter to wind up missing essential information.

If everything checks out okay, Q&A will pop-up a pick-list of available merge letters from the external Letters.dtf database. The XUserselectR command displays the letter *descriptions* and returns the *filename* of the selected letter to the Temp1 field. The @XLookup command then returns the letter *description* to the Temp2 field, using the filename in Temp1 as the key. This way, you’ve got the letter’s description *and* filename in separate fields where the programming to come can use them.

The program now prompts you to confirm that you really want to print the selected letter for the person whose record you’re viewing. If you answer *Yes*, then the program creates what for lack of a better term I call a *formatted macro string*—that is, a field value that’s destined to become a runable macro.

For example, if the selected merge document’s filename is *Intro.mrg* (short, perhaps, for introductory merge letter) and the current record’s *Record No* field contains *15514*, the program would create the following formatted macro string in the *WriteMacro* field:

```
<begdef><alt1><name>”<caps,>alt1<caps.>”<vidoff>wg
<capsf4><f10>Intro.mrg<enter><f2><f10>tom<enter>tom
<enter>{Record No =”15514"}<f10><enter><esc><esc>x
<enddef>
*
```

Note that in this macro, as well as in the program that creates it, the *tom<enter>tom<enter>* password box sequence is only for password-protected databases. If your main database isn’t password protected, then you don’t want your macro attempting to enter a User ID and password here. If the database *is* password protected, replace this string with a valid User ID and password that grants minimal Enter/Edit data rights to the database.

The retrieval expression—in this case, *{Record No ="15514"}*—can fail if it doesn’t wind up all on one line when saved to a disk file (which happens next). When writing the program, turn Insert mode on, type an opening double quote mark, press Enter to move down a line, then type a closing double quote mark before adding the retrieval expression. Keep the expression at or near the beginning of a line (as in the example above) and you’ll eliminate any possibility that it might wrap.

WriteMacro

```
< If WriteMacro = “” Then Cnext;
  @Macro(“Write Macro”)
```

Program control passes from the *SendLetter* field to the *WriteMacro* field. If the WriteMacro field doesn’t contain a macro string, nothing happens. If it does, the *Write Macro* macro shown below is invoked, saving the macro string to disk.

```
<begdef><nokey><name>”Write<sp>Macro”<vidoff>
<f6><f2><dn><dn><dn><dn><end><f10><enter>
<capsf4><f10><dn><enter>letter.asc<enter>
<lft><enter><f6><tab><enddef>
```

If you press this macro’s keystrokes (starting with the F6) from any database field, you’ll see what the macro does better than I can describe it. Essentially, it saves the formatted macro string to the Q&A program files directory to a file named Letter.asc. The *Write Macro* macro, of course, must be in your active macro file, which is usually Qamacro.asc.

If you’re feeling a bit lost at this point, think of it this way: The programming in the *SendLetter* field constructs an Alt-1 macro containing two variables (the unique identifier from the record you’re viewing and the filename of the

Continues on page 11

How to Program and Use Linked Lists, Part 3

ERIKA YOXALL

This powerful programmer's tool can help simplify development of complex applications

5.0 Application in ONLINE Edition

FOR the past two months, I've talked about linked list theory and described what linked lists do and why you might care. This month, I'll show you linked lists being used in a practical application.

Most businesses have some type of inventory that needs tracking. The problem with inventory is that you have to keep track of lots of items that have very little in common. Say you have a bowling supply store. Bowlers need, among other things, gloves, shoes and those peculiar little towels. Towels are a standard size, but come in different colors. Gloves are in different sizes depending on whether they're for men or women. Shoes come in a three-dimensional matrix of size, width and color. If a customer wants to know what color the size sevens come in, you need to be able to find out. You have several choices to solve your problem:

- You can design a database with all the fields you think you might ever need for any item and enter records for Shoes, Gloves and Towels. Each record has fields to store up to ten each of colors, sizes, widths, styles, etc. This gets all your data in one record, but most of the fields in each one will usually be empty, wasting huge amounts of screen and disk space. You also have to add more fields if you start carrying a product with more than ten colors or a different category of characteristics.

- You can design a database where each combination of characteristics becomes a different inventory item. Enter records for Yellow Towel, Red Towel, Blue Towel, Size 7D Brown Shoes, Size 7D White Shoes, etc. This makes redesigns less frequent, but forces you to page through piles of records to figure out what you have available.

- You can use linked lists.

Since this article is about linked lists, let's explore that choice!

As with the two previous articles, you need to build two databases, one to store the node records (which I'll call LINK3.DTF) and one to work with the nodes (LINKINV.DTF). You can download these databases from my Web page at *http://www.hammerdata.com/Downloads/link3.zip*.

The LINK3 database

This simple database has seven fields, the five minimum required fields for linked list structure (**NodeID**, **Parent**, **Child**, **Previous**, **Next**), **Description**, and **NodeDesc**. All fields are formatted as Text and all except Description are Speedy. NodeID is Speedy Unique. (In Part 2, I specified some rules that no node could have both a *Previous* and a *Parent*, etc. These rules applied to that example only. Linked list nodes can be specified however needed to make the tree build properly.)

At the Initial Values Spec, set NodeID to @Number.

Only one programming statement is needed:

```
NodeDesc = @Str(NodeID) + “ “ + Description
```

This gives Q&A something human readable to display in Userselect lists.

LINK3 is now ready to be populated with records. Every record, regardless of whether it represents an inventory item, a color, a size or anything else is stored in the exact same structure. You determine the way the record will be handled by how you set the links. My LINK3 has forty-three records representing three inventory items and their tree of options. The records creating one branch of that tree look like this:

NODE	Description	Parent	Child	Previous	Next
2	Shoes	None	5	1	3
5	Size	2	11	None	None
11	Seven	5	16	None	12
12	Eight	None	18	11	13
13	Nine	None	20	12	None
16	Width	11	24	None	17
17	Color	None	27	16	None
18	Width	12	30	None	19
19	Color	None	32	18	None
20	Width	13	33	None	21
21	Color	None	36	20	None
24	C	16	None	None	25
25	D	None	None	24	26
26	E	None	None	25	None
27	Brown	17	None	None	28
28	Black	None	None	27	29
29	White	None	None	28	None
30	D	18	None	None	31
31	E	None	None	30	None
32	Brown	19	None	None	None
33	C	20	None	None	34
34	D	None	None	33	35
35	E	None	None	34	None
36	Black	21	None	None	37
37	White	None	None	36	None

Continues on page 8

@Help

EDITED BY GORDON MEIGS AND WILLIAM HALPERN

'Not Enough Memory' Messages

When trying to print mailing labels from a small database (about 800 records), we're getting a "*Not Enough Memory to Complete Your Operation*" message. We've been printing labels from this database for years without problems. We recently upgraded from Q&A 4.0 to 5.0, installed Speedfix, and modified some macros. (I copied an existing label format to a new name, and modified the macro to use it.) We have Q&A 5.0 running on workstations in 1.9G hard drive partitions, and the macro file and databases reside on a shared NT network drive. This "Not Enough Memory" message is driving me crazy. What am I missing?

Charles Mares via email

There are a few clues in your message. You've upgraded from Q&A 4.0 for DOS to Q&A 5.0 for DOS and have done almost everything correctly. You've made sure the hard drive partitions in which you're running Q&A are under 2G, and you've applied the 5.0-specific Speedfix patch so Q&A 5.0 can run smoothly on your high-powered Pentium PCs.

One thing you don't mention, however, is any *memory* setting changes required for both the computer and Q&A 5.0. You're using a label design that we assume includes font formatting. This means that Q&A will require some Expanded memory—*real* Expanded memory, not the virtual kind that Windows likes to automatically make available to programs. We've addressed this memory issue before. It's very important. The only way to make sure that Q&A has *real* expanded memory is to do the following:

1. Force the computer to make it available by making sure that the config.sys file includes the following three lines:

```
device=c:\windows\himem.sys
device=c:\windows\emm386.exe RAM
dos=high,umb
```

2. Force Q&A to use the Expanded memory by making sure that the properties of the Q&A desktop shortcut include memory settings of 1024 in both the Expanded (EMS) and Extended (XMS) memory settings. (In some cases Windows won't let you choose these settings until you've changed the config.sys file accordingly and rebooted.)

With these two changes, pressing Ctrl-F3 at Q&A's Main menu will bring up the System Information screen with the three memory lines looking something like this:

```
                                DOS/TSR   Q&A   FREE   TOTAL
14.  Conventional memory          67K    573K     0K    640K
15.  Expanded memory (EMS)         0K    128K   896K   1024K
16.  Extended memory (XMS)         0K      3K  1020K   1020K
```

This should take care of those cryptic "Not Enough Memory..." messages.

Database Damage Not Y2K Related

I just read the December 1999 *Quick Answer* article about QABACKUP not being Y2K compliant. We use a backup system on our server to back-up all the files on our network, so I assume we don't need to be concerned about the QABACKUP utility not working after December 31, 1999. However, I've had to recover our databases from time to time, and I've used Q&A's database Recovery and Copy features, which I assume are Y2K compliant. (Can you confirm this?) When recovering a database the last time, I did everything according to the instructions. But when I then tried to copy the design of the database, Q&A began blinking "*Please wait*." I let it go for three hours and the message stayed there. Did Q&A lock up for some reason? Is there anything I can do if the copy operation appears not to be running? Or was it running and the database design was just too big to copy? Because I could not copy the design, I did not do the re-indexing, but the database recovery was successful. Thanks for any help you can provide.

Jaci, via email

We've said it before. We'll say it again: Year-2000 compliancy has to do *only* with date values, date displays, and date functions. All of Q&A's built-in features and functions will work as they were designed to work in 2000

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.

and beyond. Even the file date display in Q&A for DOS' List File screens (which reveals the last access date improperly unless you've applied the Y2KFIX patch) won't affect the operation of Q&A or the integrity of your databases. The issue you're reporting here isn't Year 2000-related at all; it's a problem of *database corruption*—and it looks as though you have more than one type of it. Database corruption can usually be handled with one of several methods. Some are simple and quick, some are more complex and time-consuming. These methods include the following, which are not listed in any specific order of importance or effectiveness:

- Running a null Mass Update on the database. (Mass updating with blank Retrieve and Update Specs.)

- Reindexing the database by first removing all Speedy fields, then re-specifying the same Speedy fields.

- Using Q&A's built-in database Recovery feature.

- Copying the design of the database, then copying all the records to the new design.

- Exporting the records from the database, then importing them into an empty backup copy of the database.

- Extracting the design, programming, data, and reports from the database using John Dow's DTFDOCTR and/or Symantec's QEXTRACT, and rebuilding a new database from scratch.

- Sending the database to a professional Q&A data recovery service. (We perform this service.)

Each of these database repair methods are independent of the others. However, there can be times when more than one method will be required to fix a badly damaged database. For example, you are using the following time-tested three-step method:

1. Using Q&A's Recovery feature to correct the most common database problems and "pack" the database records.

2. Copying the database design to a new name, then...

3. Copying all the records to this new database to ensure that index- and data-related problems are corrected.

In a majority of cases, this three-step method will solve database corruption problems. But when you have a severely damaged database that might have internal form design corruption, corrupted records, generational report problems, corrupted initial values or template issues, or the like, Q&A's Recovery feature might report that it was successful and you *still* won't be able to use the database. You most assuredly will *not* be able to copy its design. You might not even be able to sort the records or extract the data using QEXTRACT. In such severe corruption cases, these operations can cause the database to lock up. The size of the database has nothing to do with it. It's the *extent of the corruption* that matters. In general, if you're performing an operation during which Q&A displays a progress bar that doesn't move for more than 30 seconds, or you get a "please wait" message that lasts more than a few minutes, you can assume that the current operation has failed, that the database is suffering from some kind of internal damage, and that your computer is probably locked up.

Database Recovery on Novell Networks

If you're running Q&A on a Novell network (which we've long recommended as the best network for Q&A applications) and you have to recover a database, *DO NOT* under any circumstances attempt to recover the database on the server by running Recover from a Windows workstation. There are those of you who will say, "I do that all the time." Trust us on this one. Just like those who've gotten away with running Q&A on a large (greater than 2G) hard drive—*eventually you will corrupt your databases*. Before recovering a Q&A database that's used on a network, you *must* copy it to a local workstation drive, perform the recovery operation there, then copy it back to the network drive.

Q&A for Windows Prints Blank Reports

Over the past few weeks we've received several calls and messages concerning a problem with printing columnar reports in Q&A for Windows. It seems that when you preview the report on screen, everything looks fine. However, when you actually send the report to the printer, only the headers and footers print. The body of the report—the data area—is blank even though the appropriate amount of *space* for the data is there.

After some checking, we heard from Kim Leet's people in data recovery at Symantec. This report printing problem is a known bug in Q&A for Windows 4.0 that was fixed in version 4.01. Yes, there *is* a version 4.01. The only way you can tell is to click on *Help / About* in your copy of Q&A for Windows. If you get a small Q&A Logo screen with *Info* and *OK* buttons, then you have version 4.01 or later. If the screen includes only an *OK* button, then you have version 4.0. The problem has to do with the HP LaserJet 4 printer driver in the 4.0 version only. Fortunately, there's a work-around. Go into your printer's properties, and change the default printing resolution from 600 dpi to 300 dpi. That's all it takes.

Bill Halpern and Gordon Meigs own Professional Computer Technology Associates in Newtown, Pennsylvania, 215-598-8440, www.pcta-usa.com, email pcta@comcat.com.

Linked Lists...cont'd from page 5

The tree built by following all the links in the above records looks like this:

```
Shoes
    Size
        Seven
            Width
                C
                D
                E
            Color
                Brown
                Black
                White
        Eight
            Width
                D
                E
            Color
                Brown
        Nine
            Width
                C
                D
                E
            Color
                Black
                White
```

Once LINK3 is populated with records, the information can be put to work in the second database.

The LINKINV database

LINKINV has all the programming to recurse through the nodes in LINK3 and do useful things with the information. My LINKINV is a fairly straightforward database which allows you to choose an inventory item, then offers a progression of all the options available for that particular item. This could be a lot fancier and have lots more functionality, but I don't want to obscure the technique. Here's the list of fields in LINKINV. They are all Text fields.

Home: One-character field in the upper left corner for the cursor to rest in

Select an Inventory Item: Offers a list of inventory items

Node: Stores the NodeID of the selected inventory item

Select Item Options: Offers progressive drill-down lists of options for the selected inventory item

Item Description: Multiline field to display your choices

MakeList: Builds UserSelect lists for the two "Select" fields

CurrentNodeID: Keeps track of where in the node tree the code is looking

Loop: Tells MakeList what kind of list to build

Three fields get programming: Select an Inventory Item, Select Item Options and MakeList. Remove comments marked with *** if pasting code into Q&A.

```
< Select an Inventory Item:
***Check to see if we already got the list of items. If
not, reset fields and get the list of items. Note: I
arbitrarily decided that my inventory items would be the
first set of nodes, thereby starting with 1.
If Loop <> “Stop” Then
{
    CurrentNodeID = 1;
    MakeList = “”;
    Loop = “Item”;
    Item Description = “”;
    GoTo MakeList
}
***If we already got our list of items, offer them as a
UserSelect list. When an item is selected, store its
NodeID and move on to Item Options.
Else
{
    Select an Inventory Item = @UserSelect(MakeList);
    Node = @Left(Select an Inventory Item, @Instr(Select
an Inventory Item, “
“));
    Select an Inventory Item = @Del(Select an Inventory
Item, 1, @Instr(Select an Inventory Item, “ “));
    Item Description = Select an Inventory Item;
    GoTo Select Item Options
}

< Select Item Options:
***If we have selected and item and this is the first
time we are looking for its options, get the selected
inventory item’s Child.
If Select an Inventory Item <> “” And Loop <> “Stop2”
Then
{
    MakeList = “”;
    CurrentNodeID =
@XLookup(“LINK3”,Node,”NodeID”,”Child”);
    Loop = “Option”;
    If CurrentNodeID <> “None” Then
    {        GoTo MakeList
    }
    Else
    {         @Msg(“No additional options!”);
              GoTo Home
    }
};

***If this is not the first level of options, see if the
selected option has a Child. If so, get the next level of
options.
If Loop = “Stop2” Then
{
    Select Item Options = @UserSelect(MakeList);
    Item Description = Item Description + “
“ + “   “ + @Del(Select Item Options, 1, @Instr(Select
Item Options, “ “));
    CurrentNodeID = @XLookup(“LINK3”,@Left(Select Item
Options, @Instr(Select Item Options, “ “)-1), “NodeID”,
“Child”);
    If CurrentNodeID <> “None” Then
    {         MakeList = “”;
              Loop = “Option”;
              GoTo MakeList
    }
    Else
              GoTo Home
}

< MakeList:
***Get the list of inventory items.
If Loop = “Item” Then
{    MakeList = MakeList +
@XLookup(“LINK3”,CurrentNodeID,”NodeID”,”NodeDesc”);
    CurrentNodeID =
@XLookup(“LINK3”,CurrentNodeID,”NodeID”,”Next”);
    If CurrentNodeID = “None” Then
    {         Loop = “Stop”;
              GoTo Select an Inventory Item
    }
    Else
    {         MakeList = MakeList + “;”;
              GoTo MakeList
    }
};
```

Concludes on page 12

The Quick Answer

The Nuts & Bolts Of Converting to Q&A for Windows, Part 2

IN last month's Part 1, I ended off with a look at a "menu" form that contains nothing but clickable buttons for retrieving other forms. Another way to link forms is to place link buttons on each form.

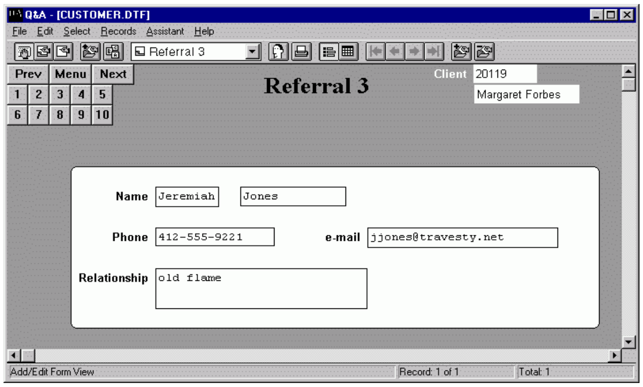
For my customer database, I created a form called "Main" that contains basic information about the customer. In this case, it has the customer's name and identification number and a few other items that provide a profile of the person. The database is configured so that this form is seen first when the database is opened.

If there's a typical flow to the information as it's being entered or reviewed, it's a simple matter to set up buttons on the forms to link them together in a sequence that corresponds to that flow. For example, each form can have a button that says "Next" and one that says "Prev." By clicking the "Next" buttons, you can move through the forms in a logical sequence.

Because there's an outline structure here, it's possible to have "Next" and "Prev" move only along the second level of the outline and not go to the third level. In the above example, successive clicks of the "Next" button therefore navigate from the "Main" form to "Address 1," "Family 1," "Referral 1," and "Position 1."

In this simplified scheme, how would you get to "Referral 3" if clicking on "Next" only gets you to "Referral 1"? Well, you could put a set of buttons labelled "1" through "10" on each of the referral forms. That way, if you're on 1 and want to go to 3, you just click on the button marked "3."

Figure 1 shows what a simple referral form might look like.



Referral 3

Client 20119

Margaret Forbes

Name Jeremiah Jones

Phone 412-555-9221

e-mail jjones@travesty.net

Relationship old flame

Add/Edit Form View | Record: 1 of 1 | Total: 1

Figure 1. A simple referral form with navigation buttons.

Defining all those forms

Once you've done a little experimentation and decided how you want your forms to look, it's a good idea to make a form to serve as a template. This form should have a title in an appropriate font, standard fields (such as the customer ID and name), and standard buttons.

As you define successive forms by taking fields from the master form, you can use the Windows clipboard to paste the entire template form into the new form. While the pasted items are all still highlighted, grab any one of them and move them as a set to the proper place on the page. Use your mouse to position the items, then fine tune the position using the cursor keys.

When you put a button on a form, you associate it with a script. Scripts typically automate actions like displaying a Retrieve Spec, running a report, printing a record, adding a new record, deleting a record, moving to a field, or displaying a different form. Q&A for Windows scripts aren't like Q&A for DOS macros. You create a script by pointing and clicking through the operations you want to it perform, then saving the script with a name.

By pasting the buttons from the template form, you will have "Next," "Menu," and "Prev" buttons on the new form. But the scripts associated with the "Next" and "Prev" buttons will be wrong—they'll be whatever you had in your template form. You'll have to double-click each one to change its attributes, then find the proper script in the popup list and double-click on it. Fortunately, there's a shortcut.

Remember that the "Next" and "Prev" buttons don't go to the third level of the outline. Hence for both address forms these buttons are associated with the same scripts. So, you can define the form "Address 1" and then use "Save As" to create "Address 2." The second form will look just like the first one and all the buttons will have the right scripts. Of course, you'll need to change the title and the names of the fields on the second form. Similarly, after you create the first referral, family, and position forms, you can quickly make copies for the other ones.

By copying forms like this, you can quickly define many forms. When they're all done, you have to make sure the buttons are all correct.

DTFSPECS has the ability to list the items on the forms. Furthermore, you can select any or all items by type. So you can see or print a report showing all buttons and their associated scripts. Since the navigation through the forms isn't necessarily in alphabetical order, you can tell DTFSPECS to display forms in an order you define. Simply use DTFSPECS to generate a list of forms and print the list to a file. Next, edit the file to remove the header and trailer lines. Then, using the editor, put the forms in the desired order. From then on, when running DTFSPECS, specify the name of this file and it will display forms and their items in your desired order.

If you generate a button report with the forms in the correct order, it's a simple matter to use it to confirm that all the buttons are properly defined.

Another problem caused by copying forms is that most of the forms will have the wrong field names. To correct this, you go to each field with the mouse, double click on it and select the correct name.

If you're using DTFSPECS, you can reduce this effort substantially. If all fields on the "Family 1" form have field names starting with "Fam1" and all fields on the "Family 2" form have field names starting with "Fam2," then you need only change "Fam1" to "Fam2" and you're done. You simply name the form, enter "Fam1" and "Fam2," and it changes all the fields at once.

Comments on forms design and grids

Forms look better if there's an order that's visually apparent and eye-pleasing. Q&A for Windows has a useful option that's common to Windows graphical design tools. It's called *Snap to Grid*. I usually define the grid at four per inch. That means that fields and other items are placed on the form at a resolution of ¼-inch.

Sometimes, later in the form design process, I change this to a finer resolution when placing particular items that work better if they're not on this rather course grid. It's also possible to turn the Snap to Grid option off. Since this snapping option pertains only to the mouse, you can always fine tune positions of items using the cursor keys.

If you create a new item, it will automatically snap to the grid. However, If you paste an item into the form, it won't snap to the grid. And if you move the item with the mouse, it will only allow you to place it in positions with increments matching your grid size, but it will always remain off the grid by the same amount—a nuisance.

There are three solutions to this problem: (1) You can use the cursor keys to adjust the position in tiny increments until the item is aligned with the grid. (2) You can use the mouse to move just the upper left corner, then the lower right corner; this will snap all four sides to the grid. (3) You can use DTFSPECS, which can snap all items of a given type on one or more forms.

Q&A for Windows has the option of snapping to font height. I find that this doesn't work well consistently and I recommend against using it. It's easier when laying out the fields and field labels to see them butting up against each other at that ¼-inch resolution.

Ultimately, of course, you want your fields to be reduced in vertical size so that their height is appropriate for their font size. My recommended technique of not snapping to font height will only save you time if you can adjust the size of all fields quickly and reliably when your form design has been finalized. As you may have guessed, I've also added that capability to DTFSPECS. You simply adjust one field to the proper height, then use DTFSPECS to set all fields on all forms to the same height. It will do it in one operation that takes just a few seconds.

Fine adjustments

It's very difficult to use a mouse to position an item on a form with precision. In fact, if you have Snap to Grid turned on, you won't be able to do it at all.

I suggest you use the mouse to select one or more items, then use the arrow keys to move them. Hold down a key and watch the items move in small increments. If you select more than one item, they will move together as a unit, maintaining their positions relative to each other.

For example, if you have a template form and paste it on to a new form, the items will be overlaid someplace fairly near center. It probably won't be anyplace close to where you want them. After pasting, they will still be selected. Move the mouse over one of the items just pasted and grab it by pressing and holding the left button. Move the entire set to near where you want them to be, then fine tune with your arrow keys.

I've recommended setting the grid to a coarse four-per inch because this yields an arrangement that's not too crowded. However, I've found the need to place some items closer together than this grid allows. For example, I had a set of about six date fields arranged vertically. Spacing them at 1/4 inch, the six dates required 1.5 inches. Well, I didn't have that much vertical space on the form where they needed to go. In such a case, you could change the grid spacing and move them with the mouse to new positions, one at a time. To do that you'll need to set the grid size, move them, evaluate the result, then perhaps reset the grid size and move them again. Not fun.

Here's a different technique for changing the spacing between them to a smaller (or larger) spacing while ensuring that the spacing remains uniform. Say you want to move them closer together. Begin by moving the second item up, even if it overlaps the first item somewhat. Use the mouse or the up arrow key, whichever works better. Do the same for the third through fifth items. Now, put the sixth item where you want it to be. At this point you have the top and bottom items properly positioned, with the others jumbled in between them (but still in the correct order). Now select them all. Then on the "Arrange" menu select "Alignment." Click the "Distribute" radio button in the "Top to Bottom" section of the dialog, then click OK. Q&A for Windows will distribute the items evenly. If the spacing isn't right, press Ctrl-Z (Edit / Undo). Then click someplace outside those items to deselect them, then move the bottom item up or down as you wish, and repeat.

John Dow is a Q&A developer and creator of a host of utilities and add-on programs for Q&A. Demo versions of many of his utilities can be downloaded from his Web site at www.johntdow.com.

Merge Letters. . .cont'd from page 4

merge letter you selected) and stores it in the *WriteMacro* field. The *WriteMacro* field's program then calls another macro (*Write Macro*) that saves this newly constructed macro string to disk, creating a perfectly usable macro file out of it. It is this Alt-1 macro—created *on-the-fly* by the programming— that actually prints the right merge letter and merges the right database record, as you'll see.

ShellLetter

```
< If WriteMacro = “” Then Cnext;

ShellLetter =
@Shell(“Q:\QA1.EXE -pQ:\QA5\QAUSER
-alQ:\QA5\LETTER.ASC -m1”);
Send Letter = “N”;
Clear(WriteMacro, ShellLetter);

If @Askuser(“ “,”Enter ‘Letter Sent’ in Notes?”,”
“) Then {
Notes = Notes + “
“ + @Str(@Month(@Date)) + “-” + @Str(@Dom(@Date))
+ “-” + @Right(@Year(@Date),2) + “ Sent “ + Temp2
+ “ letter.”;
Clear(Temp1, Temp2)
}
Else CNext
```

The *Write Macro* macro's last command is Tab, which moves the cursor to the ShellLetter field. Here, the @Shell command launches a second copy of Q&A to do the actual letter printing. Notice the *-p* personal path switch, followed by the name of the network directory (in this case, “Q:\QA5\QAUSER”) where a generic Q&A configuration file is stored. In this case, it's the same path for all users, pointing to a QA.CFG file that provides consistent printing results for all users who are assumed to be printing to the same network printer. (All the merge documents are set to print to the same network printer as well.) PC's not sharing the Q&A program don't require this personal path switch. For a standalone PC, your @Shell command might look more like this:

```
ShellLetter =
@Shell(“Q:\QA5\QA1.EXE -alC:\QA5\LETTER.ASC
-m1”);
```

The *-al* switch tells Q&A to autoload the Letter.asc macro file (saved earlier to the Q&A program files directory by the *Write Macro* macro) and run its Alt-1 macro, the only macro in the file.

Continuing with the example, the Alt-1 macro, now running in Q&A Copy 2, gets the *Intro.mrg* merge letter in Write, presses F2 and F10 to print it, then supplies the User ID and Password (*tom* and *tom* in this case) in order to access the password-protected database. It then types the retrieval expression—in this case, *{Record No ="15514"}*—into the first field at the Retrieve Spec (which must be wide enough to accept it) and presses F10 to retrieve the record whose data is then merged with the letter.

The macro then escapes back to the Main menu where it presses “X” to exit the second copy of Q&A and return you to the record you were on when you started.

The rest of ShellLetter's program can now execute. The SendLetter field is changed back to its default “N” and the WriteMacro, ShellLetter and Temp fields are cleared. You're asked if you want to add a note that the letter has been sent. If you say Yes, then a new line is added to the Notes field that includes the date and letter description in the following format:

```
1-12-00 Sent Introductory letter.
```

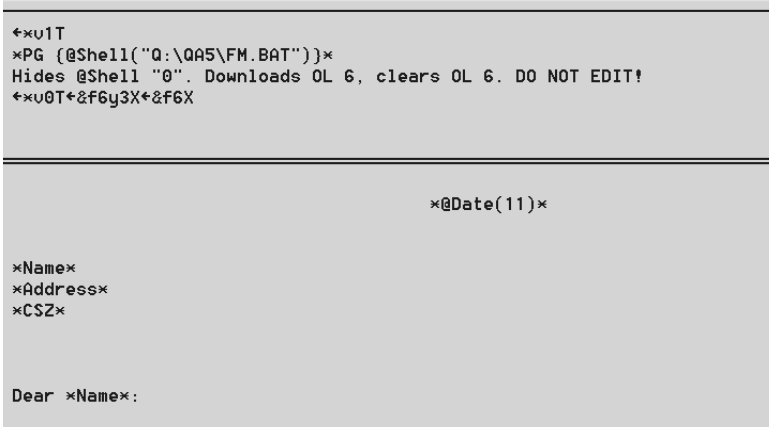
All this might seem rather complex in terms of set-up, but from a user's standpoint it couldn't be easier. You simply select the letter to print and Q&A prints it, merging the data from the record you're on. Since you're still in the same record, you can update any of the other fields, such as the next follow-up date, before moving on.

Letterhead printed on-the-fly, too

The company I worked with on this application used a LaserJet 6L as their network printer. They needed to print their letters on company letterhead, but the 6L doesn't have a second bin. So we had to find a way to make the application actually *print* the letterhead at the top of each merge document as it passed through the printer.

Fortunately, the letterhead consisted of a simple black-and-white graphic logo along with the company name, address, phone number, and so forth. Using FORMagic as reviewed in the December 1998 and January 1999 issues (“Graphics Printing Comes to Q&A DOS,” and “More FORMagic Tricks”), I created a LaserJet printer overlay from a scanned copy of the company's logo, then added commands in the header of each merge document to download the logo to the printer at printing time and print it at the top of the page. (See Figure 3.)

This, of course, happens in the background. Users



```
←*v1T
*PG {@Shell("Q:\QA5\FM.BAT")}*
Hides @Shell "0". Downloads OL 6, clears OL 6. DO NOT EDIT!
←*v0T←&f6y3X←&f6X
```

```
                                        *@Date(11)*


*Name*
*Address*
*CSZ*


Dear *Name*:
```

Figure 3. The four lines of code in the header of this Q&A 5.0 merge letter print the company's letterhead logo (a black and white graphic) at the top of page, while the normal printing process prints the rest of the letter. You can make documents like

don't have to deal with any of it. They just print their merge letters as described earlier. They see only the resulting merge document that comes out of the printer with the letterhead and letter printed on plain paper.

The first header line in the document is the LaserJet printer control code for "solid white" printing. (In other words, *no* printing.) This suppresses the "0" (return code) produced by the @Shell command on the subsequent line.

On the second header line, the @Shell is inside the Q&A Write **PG { ... }** merge program, which tells Q&A to execute the command at printing time. (Calling @Shell this works only in merge documents.) In this case, @Shell calls FM.BAT, a batch file containing the following commands:

```
echo Downloading Letterhead to Network Printer
Q:
cd\FMAGIC
FMDOWNLD LOGO LPT2
```

The batch file changes to the FMAGIC subdirectory under the Q:\ directory in this case, and runs FORMagic's FMDOWNLD utility that downloads the LOGO overlay file to the LPT2 LaserJet 6L printer. (When the LOGO overlay was created in FORMagic, it was designated as overlay 6.)

The third header line (comments) doesn't print.

On the fourth line, the first printer control code turns regular (black) printing back on. This is followed by the final codes that invoke overlay 6 (printing the graphic logo on the page), then clear the overlay from the printer.

At this point in the process, the letterhead at the top of the page has been printed and the rest of the merge document is now printed.

And that's how we print the company's letterhead on-the-fly, right along with each merge document.

Tom Marcellus is editor of *The Quick Answer* and Webmaster at www.quickanswer.com

Linked Lists. . .cont'd from page 8

```
***Get the next level of options for the selected
inventory item.
If Loop = "Option" Then
{
    MakeList = MakeList +
@XLookup("LINK3",CurrentNodeID,"NodeID","NodeDesc");
    CurrentNodeID =
@XLookup("LINK3",CurrentNodeID,"NodeID","Next");
    If CurrentNodeID = "None" Then
    {     Loop = "Stop2";
          GoTo Select Item Options
    }
    Else
    {         MakeList = MakeList + ";";
              GoTo MakeList
    }
}
```

Try this out. Add a record to LINKINV, select different items, and navigate through the different option trees.

The beauty of this structure is that you can have an infinite number of items with an infinite number of options and levels of options without ever having to change your database structure. You can use programming to put the information together in whatever format you choose. If you want to, try adding some information to the structure of LINK3 and using it. For example, you might add a field called ValueChange that adds or subtracts from the total price of the item.

There are many ways to put this elegant and flexible structure to work in your systems to create efficient and effective Q&A applications.

Erika Yoxall owns Hammer Data Systems in Garrettsville, Ohio and hosts the Q&A User Group's Web site at www.qaug.com. 330-527-4018, hammer@apk.net.

MARBLE
PUBLICATIONS
1927A Harbor Blvd. PMB 389
Costa Mesa, CA 92627

January 2000 Issue