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

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

AND

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

Build an Automatic Report Printer

TOM MARCELLUS

Q&A reports can be a problem. You carefully created a custom report for the people in your office. You tell them to use *Print a Report* rather than *Design/Redesign a Report* when they need to run it. But they use *Design/Redesign* anyway, and there goes your careful creation.

The troubles are compounded if the report requires the user to supply one or more retrieval parameters at run time. Report Retrieve Specs don't support pop-up pick-lists of field values. Nor can you guide them along with custom help screens or @Msg messages. So they've now not only redesigned your report, but made it unusable with faulty retrieval information saved into the design. Pretty soon, you hear from an employee that "your" report doesn't work.

Sure, you can protect the database and require passwords to run its reports, use macros to simplify and bullet-proof the report-printing process, create a dozen or more "versions" of the same report in an attempt to accommodate all the various options—or implement any of a host of other "solutions" that work around the basic problem of running the right report the right way to get the right information.

Or you can get inventive and approach the problem another way altogether.

I'll show you one of those other ways.

The approach I'm talking about is *not* to run your critical Q&A reports in the usual way at all—via the Reports menu and List of Reports screen—but to run them from a database record. (See Figure 1.) Why? The biggest reasons are *control* and a simplified user interface. A programmable database form gives you almost unlimited user interface design capabilities and control options where Q&A's Report writer gives you none. You'll see this as we develop an approach in this article. By running certain "qualified"
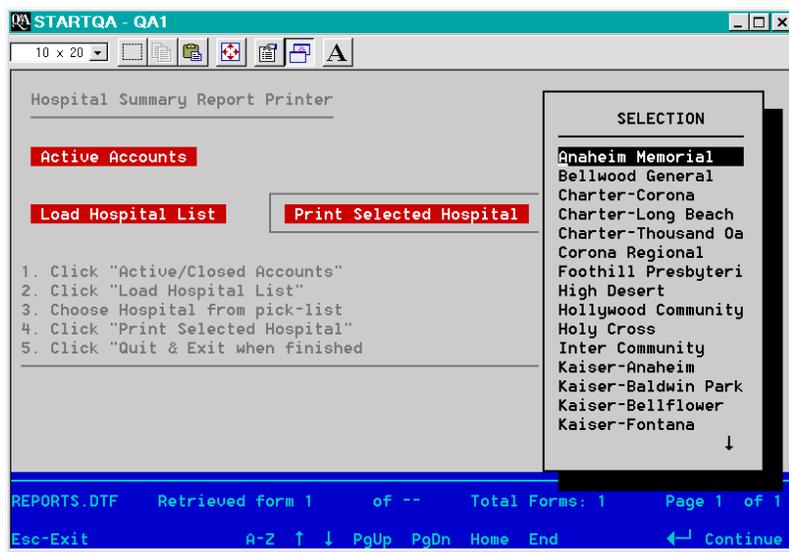
Figure 1. Using a pick-list to select the hospital to print a report for. Sure beats having to type hospital names into a Retrieve Spec.

reports from a database, you avoid all the complexities—and bugaboos—of doing it the "old-fashioned" way:

• Users will have to work extra hard to make a report-printing mistake because it's too simple. (Some will argue that users will always find ways to make *new* mistakes no matter *how* simple you make it.)

• You can employ user-friendly pick-lists and the like to allow point-'n-click selection of report run-time variables such as the range of records to include. No more manual typing of these. (See Figure 1.)

How Batch Posting Can Trash Databases

As this issue neared publication date , Erika Yoxall (erika@hammerdata.com) circulated an email message regarding a serious Q&A Batch Posting bug she had just uncovered. John Dow (jtd@pgh.net) joined the discussion, relating his experience with the very problem Erika describes and the conditions under which he has seen it. This important email exchange, minimally edited for publication, is presented here for our readers. —Ed.

Like many of us, I built a database (INVNTORY) for a client with a scrollable history field. They enter transactions into another database (PRDCTION). Each PRDCTION record then posts a line into the History field in the matching INVNTORY record. The History field is supposed to be cleared at the beginning of each year using a Mass Update, but the client forgot. After several years, a few of the records accumulated close to 32K of history. That's when the fun began.

Their posting procedure began to show two different problems, depending on the operating system.

On a Windows 98 PC, when the attempted post caused the History field to become larger than 32K, Q&A threw a "*System Error: Not enough memory to complete this operation*" message, wrote garbage into the target record, and corrupted the target database. In one case, it corrupted INVNTORY so badly that Q&A did not even recognize it as a database anymore.

The Windows 2000 PC was even more fun. It showed all the same behavior as the Windows 98 machine and, as an added bonus, deleted PRDCTION.DTF (the Post *source*) from the computer! PRDCTION.IDX was still there, but the DTF was gone. Neither Q&A's file manager, nor Windows Explorer, nor the DOS dir command could see it.

John Dow replies

I know someone who's been struggling with the posting problem you mention (deleting the DTF file).

He's seen it occasionally for years.

He sells software to retailers and they often change or upgrade their systems resulting in strange and unexpected problems.

I can report that the problem isn't Win 2000. And it isn't that it exceeded 32K in the field.

It only matters how much conventional memory Q&A needs and how much is available.

We saw the problem on older operating systems when lots of memory was taken up by drivers.

If the conventional memory available is below some threshold, you get a message that there's not enough memory and nothing bad happens. (Example from my testing: at or below 557 it won't post and says so.) If the memory is above another threshold, posting occurs without problem. (At or above 560, posts without problem.) It is when the amount of memory lies between the two thresholds that this nasty behavior is seen. You get a message about there not being enough memory and then as

Editor / Publisher
Tom Marcellus

- And perhaps best of all, users can't accidentally (or with malice aforethought) modify your sacred report design when running the report from a database.

Here are some of the criteria involved in running a report in the sample application I'll be showing you:

1. You select "Active" or "Inactive" accounts as the first variable report retrieval parameter. (The operative term is *select*—not *type in*.)

2. As the second variable report retrieval parameter, you select the object or "thing" for which you want to run the report. In the example I'll show—based on an actual application—the object is a particular hospital out of dozens of hospitals with which the company does business. It could just as well be vendors, customers, departments or whatever.

3. You click a button that says "Print Report for Selected Hospital."

It's as easy as that. No having to go through the Report menu or List of Reports screen. No filling out a Retrieve Spec (where most report-running mistakes are made), and no having to "F10" through the Column/Sort Spec and Print Options screens (where users sometimes get "bright ideas" about "improving" a carefully constructed report design).

The report-running *techniques* are what matter here, not the reports themselves. As you'll see, those techniques can accommodate a wide variety of report-printing needs.

Background
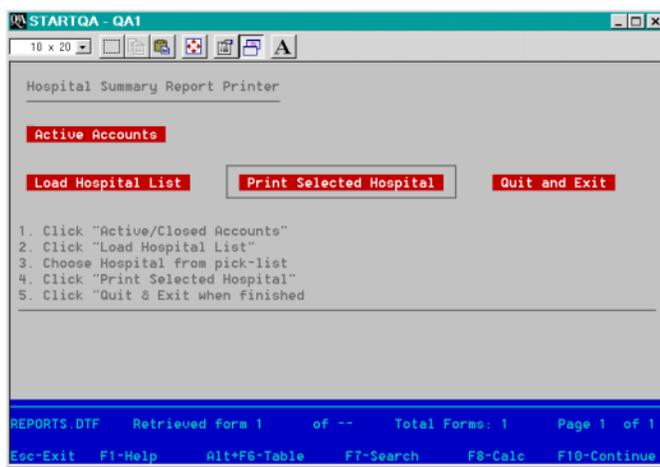First, a little background. What prompted this unusual

Figure 2. The user interface for the report generator. It's just a database record.

report-printing approach was a Q&A application that dealt with patients, hospitals, and third-party insurance requirements.

For injuries received under certain conditions—such as a victim of a crime—a hospital treating a patient for such injuries might be entitled to reimbursement from California's Medi-Cal or other state program.

My client's clients—hospitals—supply "case" information (patient and injury/treatment-related details). My client starts a case file on the treated person, contacts them and runs a kind of investigation to determine how and to what extent the treatment qualifies for coverage.

So my client has this database of cases—some active or open, some inactive or closed (which are eventually moved out to an archive database)—as well as a separate database of the hospitals it works with.

Report printing a pain
Management at the company needed to run month-end and on-demand reports for their client hospitals that showed the statuses of active cases. They also needed an on-demand report showing just the closed or inactive cases for a given hospital.

They started by designing two reports—Active and Inactive—for each hospital. Not only was running the month-end reports this way laborious in the extreme, as their number of client hospitals grew, they were soon facing Q&A's maximum reports per database limit.

When they tried a more generic report approach that required user Retrieve Spec input, they often got bad results for different *data entry-related* reasons, such as if the hospital name wasn't typed in exactly right at the report's Retrieve Spec. They needed an easier, less error-prone way to print these reports.

Ideally, management wanted to be able to run the reports off a picklist of hospitals. They wanted a minimum of training and minimum user intervention, so anyone could be easily and quickly shown how to do it..

From the volume of month-end reports, as each active report was "point-'n-click" printed for a hospital, they wanted the pick-list to redisplay with that hospital no longer on it. This way, the person doing the printing could simply pick the next hospital now at the top of the list and run the report for that one. Bang, bang, bang right down the list until each report had been printed and the list was empty. The chance of running the same report twice for the same hospital, or skipping a hospital, was thereby minimized.

But they still needed to be able to go in and print, say, an active or inactive report—or both—for any hospital at any time.

A "report-running" database and a few simple macros wound up doing the trick. This month's download file contains a little working application, complete with reports, that demonstrates the whole "system" I'll be showing you. See "About the Sample Application" section near the end of this article.

Designing the application

Figure 2 on the previous page shows the one-page, one-record database that serves as the report generator interface. The dark, button-like elements labeled **Active Accounts**, **Load Hospital List**, **Print Selected Hospital** and **Quit and Exit** are fields programmed to work like click-on buttons.

Figure 1 on page 1 shows the hospital selection list that displays automatically as required. There are a few "invisible" fields on the form as shown on the database design screen in Figure 3 below.

The best way to show how the report generator works is to take up the programming and macros that come into play at each step in the report printing process. There are a variety of "unusual" or advanced features involved. One or two might float your boat even if the application as a whole isn't the kind of thing you need right now.

Initialize

```
< If @Add Then {
@Msgbox("You cannot add records to this
database.","Please
use Search/Update mode only."," "); @Macro("Esc")
};

Print Report for = "Active Accounts";
Load Hospital List = "Load Hospital List";
Print Selected Hospital = "Print Selected
Hospital";
Quit = "Quit and Exit";

If Initialize = "1" Then Goto Insert List;
If Hospitals <> "" Then
Userselect(Hospitals,Selected Hospital);
```

The **Initialize** field is the first field on the form and its program above executes on field entry. If you're trying to use Add Data mode, you'll get nowhere. A message box will pop-up telling you that you can use Search/Update mode only. This is followed by a macro named "Esc" that unceremoniously knocks you back out to the File menu.

```
<begdef><nokey><name>"Esc"<vidoff><esc>y<enddef>
```
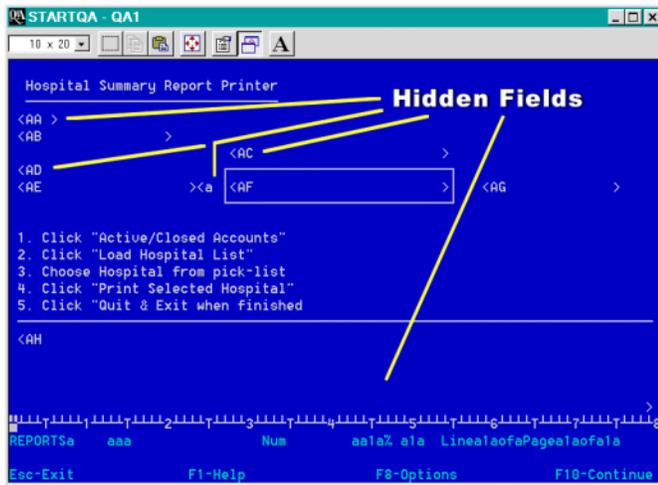

Figure 3. The database design screen. The AA, AD, a and AH fields are invisible fields.

You have to initially add one record to the database before plugging in this program. From then onward, the database will contain just the one record that's accessible strictly in Search/Update mode.

Once you've opened the database in Search/Update mode (via a menu macro, presumably), the second block of programming executes, setting the labels on the four "button" fields. The **Print Report For** field is set to "Active Accounts" (the default), the **Load Hospital List** field is set to "Load Hospital List," the **Print Selected Hospital** field is set to "Print Selected Hospital," and the **Quit** field is set to "Quit and Exit." These four fields are designed to be just wide enough to accommodate their descriptive labels and are colored at the Change Palette screen to make them stand out from the rest of the form.

The third block of programming then executes. Its first statement executes if the **Initialize** field contains "1." It includes a Goto command, so if the condition is met then the second statement doesn't execute because the cursor has already left the field. You'll see what roles these two final statements play when we look at the programming in the respective fields.

Print Report For

```
< If Print Report for = "Active Accounts" Then
{ Print Report for = "Closed Accounts"; Chome };

  If Print Report for = "Closed Accounts" Then
{ Print Report for = "Active Accounts"; Chome }
```

Print Report For is the second field on the form. A statement in the previous (**Initialize**) field initialized this field to "Active Accounts." But by clicking on it, you can toggle it between "Active Accounts" and "Closed Accounts." Whatever this field is set to when the report is run determines which of the two *types* of report is run for the selected hospital.

Selected Hospital

This field ("AC" in Figure 3) is invisible on the form until a hospital is selected from the pick-list. Its program is:

```
< If Hospitals <> "" Then
Userselect(Hospitals,Selected Hospital); Chome
```

When the cursor enters this field, if the **Hospitals** field contains anything, the pick-list of hospitals displays, populated with the values in the **Hospitals** field. The selected hospital is placed in the **Selected Hospital** (this) field and the cursor is sent to the **Initialize** field. (The **Initialize** field is a kind of resting place for the cursor.)

How does the **Hospitals** field get populated with the list of hospitals? I'll get to that in a moment.

Load Hospital List

After you select "Active Accounts" or "Closed Accounts," you can click on this button field whose label is "Load Hospital List." Here's where things get interesting. First, the program for this field:

```
<
If Hospitals = "" Then {
Initialize = "1";
@Macro("Generate Hospital List")
};

If Hospitals <> "" Then
If @Askuser("Are you sure you want to","start from
the beginning","with a NEW hospital list?") Then
{
Initialize = "1";
Clear(Hospitals);
@Macro("Generate Hospital List")
}

Else Chome
```

If the **Hospitals** field is empty, there are no values for the pick-list. In such a case it's this program's job to go out and get them. It sets the **Initialize** field to "1" so that when the "Generate Hospital List" macro returns control to this database, it will know what has just occurred and therefore how to proceed.

Here's the "Generate Hospitals List" macro:

```
<begdef><nokey><name>"Generate<sp>Hospital<sp>List"
<vidoff><capsf10><esc>rp<capsf4><f10>hospitls<enter>
_FOR<sp>PROGRAMMING<sp>-<sp>HANDS<sp>OFF!<enter>ni
<capsf4>c:\windows\temp\hospitls.txt<enter>y<esc>fs
<capsf4><f10>reports<enter><f10><enddef>
```

The macro "presses" Shift-F10 to save the database record, escapes back to the Main menu and runs the "*_for programming – hands off!*" report in the HOSPITLS database. This report generates a list of all the hospital names, one per line, with each one followed by a semicolon. It looks like this when viewed in Notepad or Q&A Write:

```
        Hospitals
—————---------------
Anaheim Memorial ;
Bellwood General ;
Charter-Corona ;
Charter-Long Beach ;
Charter-Thousand Oaks ;
Corona Regional ;
Foothill Presbyterian ;
Etc....
```

The macro prints the report output to DISK to a file named C:\WINDOWS\TEMP\HOSPITLS.TXT, overwriting any existing file of the same name.

Finally, the macro returns to the REPORTS database (and the lone record) where it was launched.

Recall that the **Initialize** field was set to "1" just before this macro was launched, and that it contains this statement:

```
If Initialize = "1" Then Goto Insert List
```

So, on returning to the record, control passes to the **Insert List** field, an invisible field just to the right of the **Load Hospital List** field. It contains the following program:

Insert List

```
< If Initialize = "1" Then
{
  Clear(Initialize);
  Hospitals =
@Insert("C:\WINDOWS\TEMP\HOSPITLS.TXT");
  Hospitals = @Del(Hospitals, 1, 33);
  Hospitals = @Repllas(Hospitals, ";", "");
  Hospitals = @Replace(Hospitals, "
", "");
  Selected Hospital = @Userselect(Hospitals);
  Chome
}
```

If the **Initialize** field contains "1" (which it does after the "Generate Hospitals List" macro has run), then the block of statements executes. First, the **Initialize** field is cleared. Then the HOSPITLS.TXT file (just generated by the report) is inserted into the **Hospitals** field—the large invisible multi-line field at the bottom of the form.

The @Del command deletes the report column heading line from the inserted file, removes the trailing semicolon from the string, then replaces all the carriage returns with nothing, so the **Hospitals** field winds up containing a long string of semicolon separated hospital names that looks like this:

```
Anaheim Memorial ;Bellwood General ;Charter-Corona
; Charter-Long Beach ;Charter-Thousand Oaks
;Corona Regional ;Foothill Presbyterian ;Hollywood
Community ;Holy Cross ;Inter Community ; Etc....
```

If all you needed to do was display a pick-list of hospitals, you could simply use an XUserselect list on HOSPITLS.DTF instead of going out, running a report and inserting the resulting file in the record. But remember, there's a requirement that as each hospital's report is run, it's then to be taken *off* the pick-list so that the next hospital in line appears at the top of the "refreshed" pick-list. By inserting the list as we have, we can handily delete the name of the hospital whose report was just run, as you'll see in a moment.

Print Selected Hospital

We're getting close to the end of the programming. This **Print Selected Hospital** program is the real workhorse of the application. Here it is, followed by an explanation:

```
<  If Selected Hospital = "" Then
{
@Msgbox("No Hospital has been selected."," ","Load
Hospital List?");
Goto Initialize
};

If @Len(Hospitals) < 3 Then
{
@Msgbox(" ","No Hospitals to print"," ");
Clear(Initialize, Print Report for, Selected
Hospital, Insert List, Hospitals, Print Selected
Hospital, Load Hospital List);
@Exit
};
```

@Help

Programming Calculations Still Don't Work

I read your article on automatic calculations but I still don't understand what I am doing wrong. I have a **Payment Due** field (#25) and a **Discount** field (#35) and I want to create a simple programming statement in the **Total** field the says:

```
Total = #25 - #35
```

Every time I try and enter this statement, Q&A says: "*Not a valid Program Spec. Press F1 for help, or see APT Manual, Pg. 14.*" The cursor is positioned on the "T" of the field name **Total**. The field name is valid and it is a money field as are the other two. Where am I going wrong?

Calvin

As I stated in my July 2002 column, Q&A does not execute calculation programming based on a field's screen position; it does so based on logical field number order. Therefore, to have an automatic or manual (F8), calculation statement work for your **Total** field, you *must* assign it a logical field number in the Program Spec and *use* that field number.

This is *not* the case for programming set to execute

Figure 1. Get any record on-screen then press Shift-F8 to set the Calculation Mode for the database.

on-field-entry (<) or on-field-exit (>), because Q&A knows when these calculations are to take place. In these kinds of fields, you can use field names without a field number. For example, if you wanted to fill the **Total** field when you move the cursor to it, you could use < *Total = #25 - #35* — a valid statement. But if you want the **Total** to be correct whenever a field is changed, you would have to use something like *#40: Total = #25 - #35* or just *#40 = #25 - #35*, and set the programming to automatic calculation. You can set a database's programming to automatic (or manual) execution by getting any record on-screen and pressing Shift-F8. (See Figure 1.)

Reports Can Use Info from Other Databases

I have a database where I keep all the information on my orders. I have a second database where I track receipt of outstanding COD's from UPS. (They are marked as "Open" in the **Status** field until I receive the payment.) Both of these databases reference the order number, but for reasons too numerous to go into here, I have to keep the tracking of COD payments completely separate. Now I am running into a situation where I need to prepare a report for UPS showing the outstanding COD's. There is not enough information in the COD database to prepare the report, and the Orders database does not track which payments are outstanding. Is there any way to accomplish what I need to do other than doing a printout from both databases and then preparing a manual listing, or redesigning my orders database to track status? Please help.

Michael

This is one of the powerful capabilities of Q&A that most people overlook: *You can retrieve records in Database A (or include them in a report) based on conditions in Database B.* First, both databases must have a matching Key field (your order number, for example) and the Key field in the external database must be a "Speedy" or indexed field. (Use the *File / Design / Customize / Speed-up Searches* screen.)
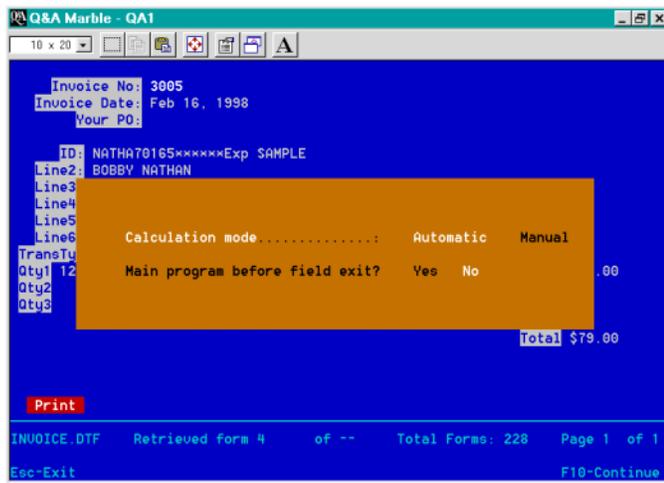
In your case, make sure that the order number in the **COD** database is a Speedy (S) or Speedy Unique (SU) field. Next, design your report in the **Orders** database with this retrieve specification:

In the Invoice No field:

```
={@XLu("COD", Invoice No, "Invoice No", "Status")
= "Open"}
```

This will retrieve only the records in **Orders** that have a matching record in **COD**, where the status in that record is marked "Open." You can also include other retrieval conditions that address the current database (such as a certain date range for your orders) along with the above external retrieval program. Then, you create the rest of your report the usual way in **Orders**. If you need a field from the **COD** database included, you can do an XLookup to that database using a report Derived Column (press F8 from the Column/Sort Spec screen). So, you can not only include records in a report based on the information in an external database, you can even include information from that external database in your report.

One caveat: If you are running this kind of report against a large (in terms of the number of records) external database and the files are located on a network, performance may be sluggish because Q&A needs to scan every external database record for each record in your current database. You can speed this up by copying everything to a local drive and running the report from there, or by running the report from the server itself.

Before you do this, keep in mind what you are trying to accomplish. If possible, design the report in the database that contains most of the information you need in order to minimize the need for Derived Column XLookups; or design it in the one with the largest number of records to maximize the speed.

Not Enough Disk Space to Backup File?

I run Q&A 4.0 for DOS from just my computer. I have a database named AR2000.DTF in C:\QA\DATA\, and a copy of the same file in F:\QA\DATA\.

After making updates to the AR2000 database on my local C:\ drive, I perform a *File / Utilities / Backup database* to the F:\ drive. But because there's already a copy of the file on the F:\ drive, I get a "Not enough disk space" error message. I first have to delete the file on the F:\ drive before the backup will work. What's your advice?

Vir

This is a new one. We know that Q&A is very sensitive to hard drive size and that it checks available disk space before it does anything. So, the solution to your problem is one of two things:

First, the obvious one: Is there *really* enough space on your F:\ drive for a copy of the file? (You should also be getting a message that the file will be overwritten.) Check this in Windows Explorer (right-click on the F:\ drive icon and select *Properties*) or run a simple dir command from the command line. If you *do* have enough space on the F:\ drive *and* the total drive size is more than 2GB, then you need to be running lesspace.com with any copy of Q&A that uses a large (greater than 2GB) hard drive. If it is larger than 2GB and you are crossing the 2GB barrier, Q&A will *not* be able to calculate the drive's available space. It can probably replace the file in a location on the drive where the old file was because it was sitting in a space below the 2GB barrier. Install lesspace.com with Q&A and run it via the appropriate batch file and try the backup again. As we have said in the past, the right kind of batch file for Q&A and lesspace is this:

```
@echo off
lesspace
qa.com
lesspace /u
exit
```

I bet that this will solve the problem.

P.S. Vir emailed us to report that this did indeed resolve the issue.

Problems Sharing a Database Opened on Server

I have a question regarding running Q&A on a network. One of my clients runs a peer-to-peer Windows-based network (all TCP/IP). They use Q&A 4.0 with the Network Pack and have an eight-user license. Anyway, everything works okay for the most part, everyone can work in the same database simultaneously, adding records, running their searches, and so forth.

The one exception is when someone opens a Q&A database locally from the "server" (where the Q&A program and data files reside). When someone does this, nobody else can get into that file. (The message the other users receive is, "*file in use by so and so*"). I'm assuming this is because, since the file is local to the server, Q&A opens it in "exclusive" mode instead of shared mode like it would over a network.

Is this just a matter of changing the sharing mode on the database (from "Automatic" to "Allow"), is there a command line switch we aren't using correctly—or what?

For your information, in the past we used Lantastic. While this caused many of its own problems (which is why we got rid of it and just use TCP/IP and Windows networking now), the one thing it did let you do is map a drive letter to itself. In other words, I could map the L:\ drive on the server to its own C:\ drive. This solved the problem I'm writing about. Windows networking doesn't let you do this, though. Thanks for any help you can provide.

Ben

It's About Time

JON MCLAUGHLIN

THERE are many design possibilities in creating a scheduling or day planner system with Q&A. I don't propose to cover all the details, but I would like to show how Q&A's Time formatting features can be streamlined and manipulated to serve several purposes.

The application I'll use to illustrate was designed to schedule weekly music lesson times. A student's lesson time might change from week to week, or a new lesson time might need to be added. This requires that the **Time** field be easily updatable. The application design also relies on Q&A's Report Module to print out a legible schedule.

Q&A and time

Q&A allows a field to be formatted as a Time field. The field can then be sorted in ascending order to produce a schedule of times. To do so, simply enter an "H" in the Time field at the Format Spec. Then, press F10 for Global Format Options. Here you have two basic options:

1. A 12-hour clock (4:30 pm) or

2. A 24-hour clock (16:30 or 16.30).

Q&A provides shortcuts for entering time values to help speed up data entry. The chart below summarizes those shortcuts for a 12-hour clock.

Typing this	Returns this
2	2:00 am
2p	2:00 pm
2.15	2:15 am
3.02p	3:02 pm
12	12:00 n
12p	12:00 n
12a	12:00 m

A simple schedule report

With a set of records containing time values, you can use Q&A's Report Module to create a schedule or day planner, with times listed down a column, in ascending order. For my application, I wanted such a column without the appended "am" or "pm", which I found unnecessary and visually "messy."

My solution seemed simple enough. I would create a Derived Column that copied the time column with "am" and "pm" deleted using the @DEL function.
At the Column/Sort Spec, I used this parameter:

```
Time: 1,AS, I
```

Then, at the Derived Column Spec:

```
Heading: Time
Formula: @DEL(#1,@INSTR(#1,":") + 3,4)
Column Spec: 2
```

Here's where I ran into trouble. Q&A will not convert the Time values to Text values, so deleting "am" or "pm" simply changes the Time values to their 24-hour clock equivalent—not what I wanted.
Here are two workarounds for this dilemma.

Method 1

This involves adding two additional Text fields to the database, one called Hour/Min and the other called am/pm. The time and "am" or "pm" are entered into separate fields, and a basic programming statement combines both in the Time field. Then, to create the Report, the Column/Sort Spec reads:

```
Time: 1,AS, I
Hour/Min: 2, H(Time)
```

I find this method cumbersome, since it requires additional fields and keystrokes. Below is what I think is a better workaround.

Method 2

This uses an interesting Derived Column that tricks Q&A into producing the desired result. It requires no additional fields.
The Column/Sort Spec reads:

```
Time: 1,AS, I
```

The Derived Column reads:

```
Heading: Time
Formula: @TEXT(#1 > 12:59 pm, #1 - 720) +
      @TEXT(#1 < 1:00 pm, #1)
Column Spec: 2
```

Q&A does time calculations using a 24-hour clock, and converts times to numbers representing the number of minutes. The statement:

soon as you clear that message, the DTF is zapped.

I discovered that these two thresholds are very close together—3K from my testing. Exactly what values the thresholds have depend on database factors (such as record size and number of fields), but I suspect that the window is always 3K.

The only solution I could come up with is to make sure there's enough memory.

I also see a brief note that with NOEMS it didn't delete the DTF file. (Of course, the conventional memory requirements would be different and you may not be able to run without it.)

I actually developed a program that can be run before launching Q&A to make sure that everything about the session is copasetic.

My client was using Q&A 4 for DOS, by the way. I don't know if it's the same for Q&A 5. Q&A for Windows would most likely not have the problem. (It does not Batch Post.)

Erika replies

I didn't check conventional memory on the Win2K box, but the Win98 box was around 585. Once I reduced the size of the target fields, the problem did go away. Is it possible that, with the large value already existing in the field, the Post-ADD operation didn't have enough resources to do the strcat-like [string concatenation —Ed.] operation it must be doing under the hood? This would be consistent with your findings on the conventional memory threshold. If so, then my problem was not the field size limit, but the attempt to manipulate a value too large for the available conventional memory using a Post. Is this correct?

John replies

That's my interpretation of things, that the large field in a particular record generated a memory requirement that just happened to fall within that window I mentioned.

For testing purposes, I used a program called EATMEM. It's a TSR that just consumes however much memory you ask it to. That way I could vary the amount of available conventional memory in 1K increments.

I suppose the thing to do would be to run the test on different operating systems to see if the problem really is governed strictly by the amount of conventional memory and is independent of the operating system. That's what I would expect—why should Q&A care which operating system it's running on? But I've not actually confirmed this.

Here's something else I've noticed when experimenting with conventional memory requirements. It takes more memory to "open" an existing record with a large amount of data than it does to create the record. I learned this attempting to determine the conventional memory requirement for a Q&A 4.0 for DOS application (to be run on a multitude of client's systems) and I found two things. First, that the amount of text I entered into a field significantly changed the conventional memory demands. Second, as I said, you can find yourself in the situation where you can enter data but you can't retrieve it unless you can free up some memory!

Strange.

Remember that the number of fields in a database is the determiner of "complexity" which has an impact on memory requirements. More fields, more memory needed. I have never found that other factors, such as the number of records or the number of reports, have any impact on memory requirements.

Erika replies

Actually, I heard about another case this morning. The person was running Q&A in Windows 98. The target database had no large text field values at all but the source DTF file was still disappearing after a Batch Post. It sounds like we have a lot of symptoms, but without the source code and a debugger it's going to be extremely difficult to determine actual cause. We may be looking at a problem caused by a combination of database size, complexity, field value size and available resources which can be arrived at through multiple routes.

*The bottom line is that Batch Posting— under certain not yet entirely isolated conditions—can severely damage the target database and even delete the source database. It is therefore crucial that you **make a backup of both undamaged databases prior to running any Batch Post.** John suggests: "A menu choice or macro that does the posting could first run a batch file to copy the databases." John adds that if you're using Q&A 5.0, you could post with his DTFDATA program and not run these risks. —Ed.*

```
If Print Report for = "Active Accounts" Then
{
Hospitals = @Replace(Hospitals, Selected Hospital
+ " ;",
"");
Print Selected Hospital = Selected Hospital;
@Macro("Print OPEN Hospital")
};

If Print Report for = "Closed Accounts" Then
{
Hospitals = @Replace(Hospitals, Selected Hospital
+ " ;", "");
Print Selected Hospital = Selected Hospital;
@Macro("Print CLOSED Hospital")
};
```

Remember that the **Selected Hospital** field is filled when you select a hospital from the pick-list. If it's empty, the first block of code lets you know that via the @Msgbox message, and prompts you to "Load Hospital List." It then sends the cursor to the **Initialize** field where Q&A awaits your next action.

If the program is still running, the second block of code executes. If the length of the value in the **Hospitals** field (the multiline field containing the list of hospital names) is less than 3—indicating there are no more hospital names in it—then the message displays telling you that there are no more hospitals for which to print a report. In this case, all the fields are cleared and the @Exit command saves the cleared record and exits to the File menu. You're done.

If there are still hospitals for which to print reports, the name of the hospital whose report was just-printed is removed from the list of hospitals in the **Hospitals** field. (The @Replace command simply replaces the hospital name, along with the semicolon that follows it, with nothing. The **Print Selected Hospital** field is then filled with the **Selected Hospital** value, and the appropriate macro is run. Shown here is the "Print OPEN Hospital macro," the one that runs the Active Cases report for the selected hospital:

```
<begdef><nokey><name>"Print<sp>OPEN<sp>Hospital"
<vidoff><f11><capsf10><esc>rp<capsf4><f10>cases<enter>
Open<sp>Hospital<sp>Summary<enter>y<tab><f12><f10>
<f10><f10>n<esc>fsreports<enter><f10><enddef>
```

When this macro is launched, the cursor is in the **Print Selected Hospital** field which contains the name of the hospital for which you are printing a report. The first thing the macro does is press F11 to copy the contents of that field (the hospital name) to the clipboard. The macro saves and exits the record, prints the Open Hospital Summary report from the CASES database (pasting the hospital name into the second field of the report Retrieve Spec in CASES), then returns to the record that launched it. The report is printing and you're now prompted to select the next hospital on the pick-list.

The macro for the Closed Accounts reports is identical except that it runs the "Closed Hospital Summary" report instead of the "Open Hospital Summary" report.

About the Sample Application

The sample application included in this month's download file is a severely abbreviated form of the original application, but it works in a way that enables you to see its features firsthand. The application includes the following files:

REPORTS.DTF—the automated report-running database. If your computer doesn't have a *C:\Windows\Temp* folder, be sure to edit the "Generate Hospital List" macro in the REPORTS.ASC macro file along with the program in this database's **Insert List** field. Change the path and filename to whatever you want to use that will work on your system.

REPORTS.ASC—the macro file containing the four macros mentioned in this article. Both report printing macros are able to handle a "No records found that met your Retrieve Spec" message without bombing. That is, if you print an Active Accounts report for Hospital X, for example, and there are no active cases for Hospital X in the CASES database, nothing strange will happen. You just won't get a printed report for Hospital X.

HOSPITLS.DTF—a one-field database of hospital names along with the report design that generates the hospital list.

CASES.DTF—the patient case file containing the "Open Hospital Summary" and "Closed Hospital Summary" reports. You can redesign these reports to suit your needs (such as choosing which fields to include in the report, the printer they print to, and so forth). Both reports are designed to print to an HP LaserJet or compatible printer in landscape mode. As-is, they're too wide for a portrait page.

Note that the Retrieve Specs for both reports use the **Closed Date** field to determine if a particular case is Active/Open (the **Closed Date** field is blank) or Inactive/Closed (the **Closed Date** field contains a date.). There is only one sample record in CASES. So, to get a realistic feel for the automatic report printing feature, you'll have to add some more records to CASES. The **Hospital** field in CASES displays an XUserselect pick-list of hospitals in HOSPITLS.DTF.

All the files are completely modifiable and you can adapt the application to meet your needs. One particularly appropriate use of the REPORTS database might be to print monthly billing statements using the Report module. (See my article, "Monthly Billing Statements in Report," in the July 1998 issue.) If you generate billing statements for more than a few customers each month, you could adapt this application to automate their printing.

Quit

The label in this field says "Quit and Exit." It's the field you click on when you're finished printing reports and want to exit the REPORTS database. Its program clears all the fields in the record, returning it to its starting state, then saves and exits via the @Exit command.

```
< Clear(Initialize, Print Report for, Selected
Hospital, Insert List, Hospitals, Print Selected
Hospital, Load Hospital List, Quit);
@Exit
```

Usage notes

To print an Active or Closed report for any hospital, you simply get the REPORTS record on screen, load the hospital list, select the hospital and print the report for it, then click on the "Quit and Exit" button.

To print the Active reports for all the hospitals (or a bunch of them), you start with the first hospital on the list (you can start with any hospital, actually) and continue selecting the next hospital and then the next one as the previous reports are printing. Each time the hospital pick-list redisplays, it will *not* include the names of the hospitals for which a report has already been printing during the current session. The next hospital in line (alphabetically) will be at the top of the pick-list and selected by default, so all you have to do is press Enter.

To re-initialize the database at any time, you can click "Load Hospital List" again.

When you're done printing, you should click "Quit and Exit" as this clears everything, setting the database back to its starting state.

If you've got "Spool MS DOS Print Jobs" turned on (*Control Panel / Printers / Properties / Details / Port Settings* in Windows 9x), as soon as you send one report to the printer, you'll be prompted to select the next one. If you don't have spooling turned on, you'll have to wait until your printer can accept all the data from the previous report before you can select the next hospital to print.

Tom Marcellus is editor of The Quick Answer.

It's About Time... cont'd from page 8

```
@TEXT(#1 > 12:59 pm, #1 - 720)
```

says "for all Time values after 12:59 pm, subtract 12 hours, thereby changing "pm" times to "am" times."

The statement relies on the fact that the time values will be converted to a 24-hour clock, where "am" times are simply the time without "am". The ascending Sort is unaffected, since that is taken care of by the Column/Sort Spec.

Conclusion

Designing a scheduling system has many aspects. It most certainly requires an efficient way to deal with Time values. I hope you'll find these techniques and tricks useful for your applications.

Jon McLaughlin is a professional musician and piano teacher living in Vancouver, Washington. He designs Q&A applications for himself and others. He can be reached at (360) 694-6869 or via email at hbark@juno.com.

Quick Tip — How to Underline Blank Report Fields

Have you ever wanted to underline or in some way mark blank fields in a Report? This method could be used to make blank fields easy to find, or to have blank fields underscored, so you could fill them in by hand later. In this example, the report output will replace all records missing a phone number with a line of underscores (___).

At the Column/Sort Spec, enter the following:

```
Phone: 5, H(15), I
```

The column width of 15 is forced using H(15), but any suitable number will work. The "I" makes this column invisible

Next, enter the following in a Derived Column:

```
Heading: 20: Phone
Formula: @LEFT(#5 + @TEXT(15, "_"), 15 – 15 * (#5
<> "") + @LEN(#5))
Column Spec: 6
```

The formula prints 15 underscores, *unless* the **Phone** field has a value entered, in which case only the value entered is printed. The expression (#5 <> "") returns the number 1 if the Phone field has a value, otherwise it returns the number 0.

Jon McLaughlin

You've got the answer: Change the setting on each database you access *at* the server station to "Allow" instead of "Automatic." This will permit someone on the server and on any/all of the workstations to access the database at the same time. Q&A looks to see if it is running on a network or a local drive for its file and record locking. When the file is on a network drive, the automatic setting is the correct one to use. However, when someone actually accesses a database from the server, Q&A thinks it is on a *local* drive instead of a network drive and it locks the file. Setting the database to "Allow" tells Q&A to share the file even if it is a local copy. Wherever possible, use the Automatic setting. Q&A will be faster and more efficient in this mode.

Now to your second issue—using a local drive as another drive letter name in Windows. You *can* do this. In Windows 9x, you can use the *Substitute* command in your autoexec.bat file and set any drive letter for any drive. The syntax of the substitute command is:

```
SUBST [drive1: [drive2:]path]
```

On a Windows 2000 or XP machine you can use the *net use* command to do the same thing. This can be done in the batch file that starts Q&A. The syntax for the net use command is:

```
Net Use [drive letter you want:]\\[name of your
machine]\[shared name of the drive to be
substituted for]
```

For example, *net use Z: \\Bill\C* will allow you to access the C:\ drive as either C:\ or Z:\. (The drive must be shared.)

More on Novell, Windows XP/2000, and Q&A

In the past few issues, We've mentioned problems associated with running Q&A on an XP machine with the files on a Novell server. We found that the Novell patch to the Windows XP driver appeared to fix the problem. (See the July 2002 issue.) *This is still the case for most people*.

However, in the past few weeks, we've received several reports of very poor performance with large Q&A databases being used in this configuration. We've had clients tell us that they find sorting and searching to be slow and that they have been getting lockups with the Q&A message "*Network Busy… Retrying, please be patient.*"

The problem is that Q&A does *not* "retry" in this situation. It simply gives up and the database is hard-locked for all users. It usually takes a Ctrl-Alt-Del combination to shut down Q&A, which can lead to having to Recover the database to reopen it, or worse, corrupting the file. We have also experienced the Novell server keeping the file locked, thus forcing a shut down and reboot of the offending workstation to unlock the file.

If—*and only if*—you are having the kind of experience described above, my suggestion is to move the database files to another workstation rather than the Novell server and run Q&A in a peer-to-peer configuration. In other words, do not access the files from the server. Map a new drive letter to the new workstation, set the sharing mode of the databases to "Allow," and use them in this manner. All problems then seem to disappear.

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.

Quick Tip — Q&A Time-Savers

Many Q&A users are unaware of Q&A's built-in shortcut keys. Here are a few you'll use often:

F7—In Add mode, saves changes and goes to Retrieve Spec.
Ctrl-F6—In Update mode, saves changes and goes to Add Data.
Shift-F7—undoes any changes made to a record.
F5—copies field value from last record into the current field.

Ctrl-F5—returns today's date.
Alt-F5—returns the current time.
Shift-F4—deletes the contents of a field.
Ctrl-Home—In Update mode, takes you to the first record.
Ctrl-End—In Update mode, takes you to the last record.

Jon McLaughlin