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

Inside Sesame.com

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

AND

LANTICA

SOFTWARE LLC.

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

Send Email from Your Database

TOM MARCELLUS

THE world has gone email. Companies large and small now hungrily forage for email addresses for their databases. And they want to *use* these things, which makes one of the most frequently-asked questions we receive at *The Quick Answer* these days (by email, of course): "How can I send email from my database?"

It strains reason a bit to expect a decade-old DOS program to do *anything* where email is concerned, but I got so tired of hearing the question and having to reply in the negative that I finally decided to see if there wasn't some way our old friend might be coaxed into doing something— *anything*— that could in some way simplify the process of sending email.

And what do you know?—it turns out that you actually *can* send email from a Q&A database.

No you can't

Well, you can't really. You need an email *program*—or email *programming* that Q&A doesn't have—to send email. Many of today's programs support various email-related *options*, but mainly what they do is start your email program for you and maybe do another thing or two to *help* you with the email. Microsoft products, for example, have a "*Send to mail recipient (as attachment)*" option on their File menus, which launches your email program and attaches the current file to a new outgoing message, but you still have to type in the recipient's address, subject and message.

You can actually do much more than this with Q&A. Heck, if you want, you can fully automate the process of preparing and sending emails using the email-related information that's right there in your database.

I'll show you how to do it using the sample Q&A 5.0 database included in this month's download file. This is all "just for the fun of it," you understand. With *Sesame* coming down the pike, and its ability to send email

Figure 1. LEFT: Sample record in the SENDMAIL.DTF database. RIGHT: Eudora new email message screen. The same data in both programs. No retyping necessary. One click passes the data from your Q&A database to your email program, and the message is ready to send.

Quick Tip — Is a Temp File Eating Up Your Free Space?

It was perplexing. I used to have plenty of free space on my 20GB hard disk, but all-of-sudden it shrunk to under 500MB. Searching around to find old folders to archive, I noticed in my Q&A 4.0 temp directory a Q&A temporary file—a big one—a HUGE one, in fact: 3.6GB! This file most likely remained from the crash of a corrupted database.

It's worth checking periodically for files like this. In Windows, you can search for files over a certain size:

Start/Search (or Find) Files or Folders, click on the "Size" checkbox for a new dropdown where you can specify files greater than, say 50,000 KB to find 50MB + files.

You might just be able to reclaim some lost hard disk real estate.

Alec Mulvey

Quick Tip — 'Added' When You Should have 'Retrieved'?

Virtually all the Q&A databases I work with from outside sources have this thing in common: they all contain at least several records—sometimes dozens of them—where only one field is filled, and that field contains something that looks like this:

```
MAR..
```

This happens as aresult of someone wanting to search the database for records with names that begin with "MAR" while mistakenly in Add Data mode instead of in Search/Update mode. In other words, they weren't at the Retrieve Spec for the database, but in a new record.

Of course when they type something like this in the field (it's often the first field in the record) and press F10, Q&A saves the spurious record to the database. The user gets no search results—just another blank form. But they don't think (or know) to press F9 to go back and delete the botched record.

I'm sure you've never made this mistake yourself, but if others use the database, you might want to check it periodically for such records.

First, you have to know which field or fields users normally search on. Let's say it's a **Last Name** field. At the Retrieve Spec, type the following in the **Last Name** field and press F10:

```
..\.\.
```

The "\" character when used in this fashion tells Q&A to search for the literal character that follows it. Accordingly, this retrieval parameter will find all the records in the database where the value in the **Last Name** field ends with two periods.

Another way to find the same bad records is to enter a retrieval expression in the **Last Name** field, like this:

```
{@Right(Last Name,
2) = ".."}
```

Depending on how the database is used, botched records like these can remain undiscovered for years. While they don't do much harm, they do add to the record count, so you can't get an accurate count of the number of good records in the database. They can also "mysteriously" show up in reports and other places where they might be ignored or "explained away" and never fixed. It's best just to find them and zap them.

Tom Marcellus

Database Emailing

with a built-in programming function (page 10), I don't expect folks to start overhauling their old databases to add some newfangled email feature. But the point is, you *can* do it. And *how* you do it makes, I think, an interesting read for those who enjoy pushing the envelope a bit.

So, what exactly do you want?

An email message consists of several things.

Two things you *must* have are the recipient's email address and a message.

One thing you'd *better* have is a subject line. I don't open emails that lack a straightforward plain-English subject line (and you shouldn't either). They go straight to the trash. A missing or malformed subject is your first clue that the message is probably from spamville—or Hell.

Three things you can *optionally* have in an outgoing email are a CC, a BCC, and a file attachment—or multiples of these.

So an email message consists of a minimum of three different elements, and a maximum of six.

When a Q&A user asks, "How can I send email from my database?"—just what is it they're really trying to accomplish?

Well, it could be something as extreme as wanting to run a kind of Mass Update or whatever where Q&A would somehow "generate" an email message to all the email addresses in the database. For such feats, you'll just have to wait for *Sesame*. (One nut wanted us to tell him how to design a Q&A 4.0 *report* to do this.)

On the other hand, they might be talking about something as basic as having Q&A launch Outlook from a database record, and have Outlook's new message screen come up with the **To:** address automatically filled with the email address from the record.

Or they might be looking for the above *plus* the ability to have the **Subject:** and **Message:** fields in their email program automatically filled as well. Or take it even a step further and be able to pass one or multiple **CC** and **BCC** email addresses to their email program from their database record.

Heaven forbid that they might want to include a *file attachment* on top of all this.

But even assuming that you *had* such information in your database, how do you get Q&A to launch your email program so that it starts a new message with everything already filled in and ready to go—including an optional file attachment?

The key is a little piece of technical know-how that only a Web developer could love.

Through the back door

In addition to your database, you'll need a two-line batch file and a Windows script file. If you don't know what a Windows script file is, see the June 2000 issue, "Extend

Q&A's Reach With Windows Scripts," along with the sidebar below.

A "self-contained" Q&A-based "send email" system could work along the following lines:

1. You could have some or all of the following fields in your database: **Email Address**, **CCs**, **BCCs** (each one supporting multiple email addresses), **Subject**, **Message**, and **File Attachment**. For email addresses, you could type them in or optionally select them from a pick-list (for interoffice emails, for example).

2. You'd type the subject in the **Subject** field. Or, you could auto-fill it with programming ("*Re: JAMISON, Rec# 3244*") or from a pick-list of canned subjects, possibly using the techniques described in "Simplify Data Entry with Wide DTFTALK Pick-Lists" in the October 2000 issue)

3. You'd then type the text of your message in the **Message** field, which could also be auto-filled initially with a canned message or message "template," say, from an XLookup-able "messages" database.

4. If you wanted to attach a file to the outgoing message, you'd either type its full path and filename in the **Attach** field, or select it from a pick-list of predefined attachments.

5. When all the particulars of your email have been specified, you click on the **Send** field. Programming in your database would validate the data, then assemble it all into a data string specially formatted for passing to an email program.

6. An @Shell command could then run the Windows script

Where to Get Windows Scripting Host

Windows Scripting Host is a powerful scripting engine that gives you a myriad of tools and ways to add automation to your databases and other applications in the Windows environment. It's built into Windows 2000 and XP, and versions of it are available for Windows 95 and 98. You can download it free at *http://msdn.microsoft.com/scripting*. Be sure to download the VBScript documentation separately. Copy the files to a new directory, then run them to complete the installation. If you're running Windows 95 Release 1 or you don't have Internet Explorer 4.0 or later, you'll also need to download and install a file named DCOM95.EXE available from the same area of the site.

file to do some final cleanup on the data string and fire up your email program, passing your data to it through the back door.

7. Your email program's new message screen would appear, containing all the particulars of the outgoing email. You'd simply click the "Send" button, then close or minimize the program (these steps could also be automated) to return to the Q&A record you were on.

This would make emailing from a Q&A 5.0 database a snap, particularly if its design and programming helped you select and assemble some or all of the six possible parts of the new message, so that you wouldn't have to type any more than is absolutely necessary.

SENDMAIL.DTF

Figure 1 on page 1 shows a sample SENDMAIL.DTF record. You can see all the fields except one. The red-colored field at the bottom of the screen is named **Qstring**. This is the field where the data string is assembled for passing to the email program. You click on this field to send the email. Just to the right of **Qstring** is an invisible field named **Run Script**.

The role these two fields play can be better explained by going through the main program in the **Qstring** field, followed by the short program in the **Run Script** field:

Qstring program:

```
<
If Email = "" Then {
@Msg("Email Address(es) missing.");
Goto Email
};

If Subject = "" Then {
@Msg("Subject line missing.");
Goto Subject
};

If Message = "" Then {
@Msg("Message missing.");
Goto Message
};

If @Instr(Email, ",") > 0 Or @Instr(Email, " ") >
0 Then {
@Msg("Separate multiple emails with a ';'. Spaces
& commas not allowed.");
Goto Email
};

If @Instr(CCs, ",") > 0 Or @Instr(CCs, " ") > 0
Then {
@Msg("Separate multiple CCs with a ';'. Spaces &
commas not allowed.");
Goto CCs
};

If @Instr(BCCs, ",") > 0 Or @Instr(BCCs, " ") > 0
Then {
@Msg("Separate multiple BCCs with a ';'. Spaces &
commas not allowed.");
Goto BCCs
};
```

```
If @Askuser(" ","Send the Email now?"," ") Then {

Date = @Date; Time = @Time;

qstring = Subject + "^" + Message;

qstring = @Replace(qstring, " ", "%20");
qstring = @Replace(qstring, ",", "%2C");
qstring = @Replace(qstring, ".", "%2E");
qstring = @Replace(qstring, "!", "%21");
qstring = @Replace(qstring, "?", "%3F");
qstring = @Replace(qstring, ":", "%3A");
qstring = @Replace(qstring, ";", "%3B");
qstring = @Replace(qstring, "
", "%0D%0A");
qstring = @Replace(qstring, "^", "&Body=");

qstring = "mailto:" + Email + "?cc=" + CCs
+ "&bcc=" + BCCs + "&Subject=" + qstring;

qstring = @Replace(qstring, "cc=&bcc=&", "");

qstring = @Replace(qstring, "&bcc=&", "&");

If Attach <> "" Then
qstring = qstring + "|" + Attach;

@Macro("SendMail")

}

Else {

@Msg("Send message aborted.");
Goto Contact Name
}
```

Starting with the **Qstring** program, you can see that it initially performs some rudimentary validation on the various fields. There must be data in the **Email**, **Subject** and **Message** fields, and no commas or spaces in any of the fields that contain email addresses. Multiple addresses need to be separated by semicolons (;). You could add further validation here, including checking to ensure that any email address contains an "@" sign and a period, and has a minimum length of, say, eight characters.

If the fields pass the validation, an @Askuser box prompts you to confirm that you want to send the email. If you hit "No," the process is cancelled and the cursor is sent to the home field. Otherwise, the fun starts.

The optional **Date** and **Time** fields are filled.

The contents of the **Subject** and **Message** fields are combined with a "^" character and stored in the **Qstring** field. This is so the programming can replace certain characters in this string with their ASCII hexadecimal equivalents.

This is "the heart of the beast," and we need to digress for a moment.

Email programs like Outlook and Eudora can be passed a particular kind of data string that opens them in "new message" mode and fills in the various fields with the data from the string. This string needs to be specially formatted—what's sometimes called *URL encoded*.

If you've ever clicked on a Web page email hyperlink like **Contact Us**, your email program should have fired up with the **To:** and **From:** fields automatically filled in.

Sometimes the **Subject:** field will also be auto-filled. Even the Message part of the email can be auto-filled from such a hyperlink, in case you (the owner of the Web site) wanted to provide some brief instructions or request specific information from your site's visitor.

The point is, an email program has to accommodate the normal process of you composing, sending and receiving email messages, as well respond to email hyperlinks on Web pages. That's why they're designed to accept URL-encoded strings as start-up parameters. And that's our key to the kingdom.

But certain characters, when used in these startup strings, can cause problems. Among these are spaces, commas, periods, question marks, exclamation points, colons and semicolons. So we have Q&A replace all such characters found in the **Subject** and **Message** fields with their ASCII hexidecimal equivalents preceded by the "%" character. This is the *URL encoding* part of it. These are then decoded back into the actual characters once they're read by the email program.

Email addresses don't need to be encoded because email programs all understand what they're looking at when they encounter one in a data string.

End of digression.

Once your database programming has applied the encoding to the **Subject** and **Message** values, it replaces the temporary "^" with the "*&Body=*" tag. You'll see how this figures into it just ahead.

At this point, you've got the contents of the **Subject** and **Message** fields in the **Qstring** field, properly encoded. You now need to finish it off by adding the other email fields, preceded by their tags, to the existing **Qstring** value.

```
qstring = "mailto:" + Email + "?cc=" + CCs
+ "&bcc=" + BCCs + "&Subject=" + qstring;
```

If there's an entry in the **Attach** field, then programming includes it at the end of the **Qstring** value, preceded by a "|" tag:

```
If Attach <> "" Then
qstring = qstring + "|" + Attach;
```

Finally, some cleanup:

```
qstring = @Replace(qstring, "cc=&bcc=&", "");
qstring = @Replace(qstring, "&bcc=&", "&");
```

If the **CC** and/or the **BCC** fields are empty, this part of the program removes their tags from the **Qstring** value and the string is now complete. A simple one might look something like this (in one line):

mailto:jjones@somwhere.com?cc=tsmith@somewhere
.com&Subject=More%20Cute%20Dogs&Body=Visit%20
http%3A//www%2Ecutepooches%2Ecom%20to%20to
%20see%20more%20cuddly%20curs%2E|c:\dogs\mypooch.jpg

Here, the message is addressed to *jjones@somewhere.com* with a CC to *tsmith@somwhere.com*.
the Subject is: *More Cute Dogs*
The Message or Body is: *Visit http://cutepooches.com to see more cuddly curs.*

And an image file named *mypooch.jpg* is attached to the outgoing message.

Look carefully and you'll see the tag that precedes each part of the email message (*mailto:*, *CC=*, *&Subject=* and *&Body=*). Notice that there are no spaces anywhere in the string. They've been replaced by "%20" and the periods have been replaced by "%2E" (except in the email addresses)—the hex equivalents of the two characters. This is an appropriately encoded startup string for passing to an email program like Eudora or Outlook.

After the Q&A programming has assembled and encoded the data string, we come to the *@Macro("SendMail")* command. Here's the macro:

```
<begdef><nokey><name>"SendMail"<vidoff>
<f6><f2><dn><dn><dn><dn><end><f10>I
<capsf4>c:\temp\qstring.txt<enter>y<f3>
<end><end><end><f10><f6><tab><enddef>
```

The macro "presses" F6 to expand the **Qstring** field (where the completed encoded data string is now stored), F2 for the Print Options screen, down-arrow four times to get to the *Print to* line, then End to select *DISK*. The F10 brings up the prompt for the filename, and Shift-F4 ("<capsf4>") erases the proposed path. The macro types in *c:\temp\qstring.txt*, presses Enter, then presses "Y" to confirm the overwrite. (The file *must exist* where the macro expects it.) The macro finishes up by deleting the contents of the **Qstring** field (no point in saving it), then pressing Tab to move to the adjacent **Run Script** field.

Run Script program:

```
< @Msg(@Shell("c:\temp\sendmail.bat"));
  @Msg("Message passed to email program.");
  CHome
```

The **Run Script** program's @Shell command runs the sendmail.bat batch file which, in turn, runs the *sendmail.wsf* Windows Script (see below). The program lets the user know what's happening via an @Msg message, then parks the cursor in the home field.

The batch file contains just these two commands:

```
Wscript c:\temp\sendmail.wsf
exit
```

In Windows 2000/XP, the "Wscript" at the beginning of the line is optional. In Windows 98, you need it, and you may need to use "Cscript" instead if you're having trouble getting "Wscript" to work.

@Help

Copying Mailing Labels to Another Computer

What I am looking for is how to copy my Q&A for DOS Mailing Label designs from one computer to another.

Andy

The answer is simple. Copy the file QALABEL.OVL from the machine that has the labels to the one that doesn't. The only issue is that this is an all or nothing proposal. All labels on a particular machine are stored in this file. So if you copy it to another machine, you will lose any that you created there and wind up with the ones from the other computer. It's best to design *all* your labels on one PC, then use that as the master QALABEL.OVL file.

Losing Original Label Definitions

I use multiple Q&A for DOS databases and I have to redesign my HP 5160 labels each time I use them because they're always pointing to the wrong one with the wrong fields. How can I have more than one copy of this label.

Bernadette

The magic word here is "copy." Before using *any* label design, you should *always* make a copy of the label to a new name. For example, before you set about customizing the HP 5160 label for the first time, highlight it, press the F5 key and copy the design to a name like *Customer HP5160* or *Vendor HP5160* or *[whatever] HP5160*. Get the idea? These new definitions will now be stored in the QALABEL.OVL file and will appear at the top of the list (by virtue of the "*"). You customize these copies of the label, *never* the originals, to make your changes and link to your databases. There is no practical limit to how many copies you can make.

Graduating From Labels to Envelopes

I can easily design address labels for my mailings but I want to be a little fancier now. Is there any way to make Q&A directly print onto envelopes instead of labels? Labels have to be attached and don't look as professional.

Gabrielle

Well, sure, it can be done. I presume we all know that Q&A for DOS has the ability to automatically print an envelope from a typed letter just by telling it to print to the envelope printer using envelope spacing rather than single or double line spacing. Oh, you *didn't* know that? Okay, then here's the secret. First, you should have one of the five printers slots in Q&A set up to be the envelope printer for your specific printer. (Usually, Printer E is a good choice as it stays out of everyone's way). Next, you type your letter with a return name and address at the top and at the left hand margin and make sure that there is at least one line skipped after the last line of the address. If you want to print the letter, simply press F2, F10. However, if you want to print the envelope for this letter, hit F2, select *Envelope* on the **Line Spacing** line, and *PrtrE* on the **Print to** line, then press F10. Voila! —an envelope.

How does this help with envelopes instead of labels? Well, Q&A works its envelope magic not only on fully typed letters, but also on merge letters. And, not only on complete letters but also on those that only have a return address and nothing else. So to make Q&A an envelope printer, create a merge letter something like this:

```
*Salutation*  *First Name* *Last Name*
*Address*
*City*, *State*  *Zip*
```

When you do the merge print, load your printer's envelope feeder with envelopes, press F2, Select *Envelope* on the **Line Spacing** line, and *PrtrE* on the **Print to** line, then hit F10. Mail-merged envelopes from your database!

Q&A on 2000 Server?

I am currently using Q&A 4.0 for DOS. I am about to set up a Windows 2000 server for my office and I am wondering if Q&A will work on the server correctly. We are currently using a peer to peer network and are over our connection limits. Any information you would give me would be appreciated.

Steven

Stumped?

Q&A works fine with Windows 2000 Server. It is what we ourselves run and suggest to all of our clients.

I do want to correct something you imply here. Q&A does not "work" on the server. Q&A is a client-based application that works fully within the memory of the workstation or client. No "work" is done on the server. The server simply stores the databases for use by people on the network. Q&A should not even be installed on the server unless you are using a Network Pack, in which case the program files reside on, but are not run on, the server. Just the data files (the .DTF and .IDX files) are kept there in a shared folder.

Two points:

- Make sure that users have complete read/write access and permissions to whatever folders on the server they will be using.

- If using a Network Pack installation, make sure that the program files directory on the server is also fully accessible by users for configuration changes and temporary files.

Calculate Daily Differences in a Report

I need some advice. I have a Q&A 5.0 database I use to record payments. How do I calculate the difference between many records for the same person entered day by day, to see the variances day by day in a report format—like this, for example:

```
Date      Name         Paid      Variance
May 10    John        $10.00
May 11    John        $15.00      +$5.00
May 12    John        $17.00      +$2.00
May 13    John        $27.00     +$10.00
May 14    John        $17.00     -$10.00
May 15    John        $10.00      -$7.00
```

Ian

The answer is in two parts. Remember, this is a *report*, not a calculation between records.

Assumptions:

1. One payment per person per day maximum, or it becomes more involved and needs to do differences in totals requiring seven derived columns instead of just three.

2. You want to see the difference from the last payment no matter how long ago it was made.

3. You want to see actual values, not absolute values.

4. You are requesting the difference from the last payment, not the statistical variance of all payments.

Requirements:

1. Done in a report.

2. Must be sorted first by person, then by date to address multiple people.

3. Must use derived columns that mimic the payments and names fields at the end of the report in order to reflect prior record values.

At the Column/Sort Spec:

```
Name: 10,AS,R
Date: 20,AS,CS
Paid: 30
```

Derived Columns:

```
Heading: Name Copy
Formula No. 1  #10
Column/Sort Spec: 100,I

Heading: Amount Copy
Formula No. 2  #30
Column/Sort Spec: 110,I

Heading: Difference
Formula No. 3  @txt(#100=#10,#30-#110)
Column/Sort Spec: 40,F(M,C)
```

Results:

```
Name     Date       Paid    Difference
——   ————   ———

Bill   05-11-2003   $5.00
Bill   05-12-2003  $13.00       $8.00
Bill   05-14-2003  $43.00      $30.00
Bill   05-15-2003  $12.00     -$31.00

John   05-10-2003  $10.00
John   05-11-2003  $15.00       $5.00
John   05-12-2003  $17.00       $2.00
John   05-13-2003  $27.00      $10.00
John   05-14-2003  $17.00     -$10.00
John   05-15-2003  $10.00      -$7.00
```

The trick here is that derived column 2 tells Q&A to subtract the last record's payment (#110) from this record's payment whenever the person is the same as the person in the last record. Remember: Only *one* payment per *person* per *day* or you'll need the following:

```
Name: 10,AS
Date: 20,AS,I
Paid: 30,I
```

Derived Columns:

```
Heading: Name Copy
Formula No. 1  #10
Column/Sort Spec: 100,I
```

Database Emailing

Following is the Outlook version of the *sendmail.wsf* Windows script file:

```
<job>

<object id="shell" progid="Wscript.Shell"/>
<script language="vbscript">

'Read in output file from Q&A.
'Launch Outlook with scrubbed startup string.
'Send keystrokes if there's a file attachment.

Set fileObj =
CreateObject("Scripting.FileSystemObject")
Set myFile =
fileObj.OpenTextFile("c:\temp\qstring.txt")

qstring = myFile.ReadAll

myFile.Close
Set fileObj = Nothing

qstring = Left(qstring, Len(qstring) -4)
qstring = Replace(qstring, vbCrLf, "")

If Instr(qstring, "|") > 0 Then

attachFile = Mid(qstring, Instr(qstring, "|") + 1)
qstring = Left(qstring, Instr(qstring, "|") -1)

End If

Return =
shell.Run("C:\Progra~1\Micros~2\Office\OUTLOOK.EXE
/c ipm.note /m " + qstring)
shell.AppActivate Return

If attachFile <> "" Then

Wscript.sleep(2000)

shell.SendKeys "%I" 'Insert menu
shell.SendKeys "f"  'Insert File
shell.SendKeys attachFile
'shell.SendKeys "{Enter}"

End If

</script>
</job>
```

This Windows script, which you can compose and edit in any text editor (Notepad will do), performs several important tasks.

First of all, when you expand a Q&A database field and print its contents to a DISK file (as the *Sendmail* macro does), Q&A adds its own "personality" to the mix:

• It replaces *soft* carriage returns (line wraps) in the expanded field with *hard* carriage returns in the output.

• It adds some garbage characters to the end of the file.

These have to be stripped out before the encoded string can be passed to the email program. The script's job is to read the *qstring.txt* file into a variable, lop off the

garbage characters, replace the hard carriage returns with nothing, then pass the scrubbed string as a single line to Outlook.

If you've included a file attachment, the script parses it out of the qstring and assigns it to an independent variable named *attachFile*.

After launching Outlook, if there's a file attachment, the script gives Outlook a couple of seconds to load (the *sleep* command), then begins sending keystrokes to it.

The SendKeys are required because you can't specify a file attachment in the startup string.

To attach a file to an email message in Outlook, you would press "Alt-I" to pull down the Insert menu, then "F" to open the "Insert File" box. The cursor is in the *File name* field when the Insert File box opens, so all you need to do is supply the path and filename and press enter to attach the file. What the script does, like a macro, is *send* these keystrokes to Outlook. The *attachFile* variable containes the keystrokes for the path and filename of the attachment. (The script supports only *one* attachment.)

All that's left to do is to send the Enter keystroke to complete the attachment process. In the above script, I've commented out that final {Enter}. Simply remove the apostrophe to activate it.

Eudora has different menus. Your SendKeys commands for Eudora would be:

```
shell.SendKeys "^h" '(shortcut for Attach File)
shell.SendKeys fileToAttach
shell.SendKeys "{Enter}"
```

For either program you can add the additional SendKeys required to actually Send the email (in the case of Eudora, *Queue* then *Send—Ctrl-E* followed by *Ctrl-T*— "*^e*" followed by "*^t*" in the script) and then even close the program if you like. I'm not in favor of sending an email message without first reviewing it.

With regard to the *Shell.Run* command in the script, use the abbreviated DOS-style folder names. The actual path to Outlook (2000) on my machine is:

```
C:\Program Files\Microsoft
Office\Office\Outlook.exe
```

But the script prefers this:

```
C:\Progra~1\Micros~2\Office\Outlook.exe
```

Another oddity. Even though *Microsoft Office* is the third (of four) *Microsoft…* folders under *C:/Program Files* on my computer, abbreviating it as *Micros~2* (rather than *Micros~3*) in the script file was the only way the script got the path to Outlook right. Go figure.

Outlook also requires the */c ipm.note /m* switch to precede the start-up data string when launching the program from the command line. If you're using Eudora, the script file is the same except for the Run command and the SendKeys. The Run command would be in this format:

```
"C:\Progra~1\Qualcomm\Eudora\Eudora.exe " +
qstring
```

So, what have we got here altogether? First, we have our Q&A program generating the encoded start-up string, saving it out to query.txt, then launching the *sendmail.wsf* Windows script via the *sendmail.bat* batch file. The script reads in query.txt, cleans it up, then launches the email program and passes it the start-up string. You simply fill in the fields in the database record and confirm that you want to send the email. The next thing you see is the Outlook (or Eudora) new message screen sitting on top of your Q&A window with the fields all filled in and the message ready to send.

Now, at least, when somebody asks, "How do I send email from my Q&A database," I won't immediately think that I'm dealing with a meathead.

Notes and further ideas

Adding recipients' real names
If you want your email addressed to the more friendly "Joe Blow" rather than just to "jblow@somewhere.com," you can modify your Q&A program to produce this:

```
mailto:Joe%20Blow%20<jblow@somewhere.com>...
```

Doing it this way has the added benefit of enabling you to search your outgoing emails by the recipient's name in the TO: field. This optional mailto: format works for both Outlook and Eudora.

Get faster sends
If you use Eudora, you can speed things up considerably by minimizing (not closing) Eudora after sending each email. This seems not to be the case with Outlook. If Outlook is minimized, the sendmail.wsf script (on my machine at least) starts a new instance of the program each time. On the other hand, Outlook loads much faster than Eudora.

Speaking of Eudora, there's a little bug in it. If you pass a question mark ("?") in the Body of the message, whether or not it's hex encoded, Eudora will truncate the message at that point. I've got a bug report in to them on it, but I'm not holding my breath.

Be brief
This little Q&A-based email system is not for long-winded email messages composed in an expanded field. Use it for brief messages where it's more convenient for you to select your recipients, subject, attachments and so forth from inside Q&A.

If your messages aren't canned (standardized) in some way and selectable from a pick-list or an XLookup to a "Messages" database, then why have a **Message** field in the database at all? Why not just compose your messages in the nice WYSIWYG editor provided by your email program?

Saving messages
Are you planning to save the text of your messages in the database itself? What for?—your email program will save your outgoing messages for you. And should you ever need to find a particular message, today's email programs have good search facilities.

What you *might* want to do, though, is save (in the database record) the date the email was sent along with the recipient's email address and perhaps even the subject. This would require a program that would store a reference for each email sent in a cumulative **Messages Sent** field:

```
Messages Sent = Messages Sent + "
" + @Str(@Month(@Date)) + "-"
+ @Str(@Dom(@Date)) + "-"
+ @Str(@Right(@Year(@Date)), 2) + " "
+ Email Address + " " + Subject
```

This way, if you wanted to review a sent email message at a later date, you could first go to the database to get the date, recipient and subject (assuming you knew which record it was in), then go to your email program to review the message. Knowing the date, email address and subject should make it easy to find.

Prospecting
If the database you're using is designed for marketing, with a record for each prospect, you could set it up to keep track of what you sent to each prospect by email, and when. This would enable you, for example, to send a different email message at set intervals to a prospect until he or she finally either told you to knock it off or broke down and called you to place an order.

Mass emailing
After poking around a bit, I found that it might be possible to work out a system whereby you could, essentially, run a Mass Update on a database that would pass a batch of email addresses to your email program (along with a fixed Subject and Message) and automatically queue them up for sending. But this is well beyond the scope of this article.

What about Q&A 4.0
If you're using Q&A 4.0 for DOS, you can't run a macro from programming. Neither do you have the @Shell command to launch the batch file. What to do? Well, in lieu of the @Macro command, you could add an @Msg message to your program to remind the user which hot key to press when they click on the **Send** field:

```
Press Alt-S from this field to send the email.
```

Your Alt-S macro would have to do more than generate the qstring.txt file; it would have to save the record, escape back to the Q&A Main Menu, and launch the Windows script file as an alternate program installed on the Main Menu. (See *Alternate Programs* in your Q&A

User Guide.) I haven't tested this, but I can't see why it shouldn't work.

I don't see any way to do any of this in Q&A for Windows without a specially-designed add-in that a chap like John Dow (www.johntdow.com) might be able to help you with.

Other email programs

I used Outlook 2000 SP-3 and Eudora 4.2 and 5.2, all running in Windows XP, for all my testing. Other versions of these email programs—and other email programs and operating systems—might not produce the same results. I see no reason why it shouldn't work in Windows 98 and 2000. Just remember that in Windows 98 your @Shell needs to specify the scripting host name and include the Windows script file filename extension.

A way to test your own email program is to pass a simple startup string to it via the Start/Run box. Enter a little test string like the following (make sure you specify the correct path to the executable file), click OK, and see what happens:

```
"c:\program files\microsoft office\office\
outlook.exe" /c ipm.note /m ailto:bsmith@blah.com?
cc=fjones@blah.com &subject=blah &body=blahblah
```

If it works, your email program's new message screen should appear, all filled out.

To test with Eudora, you'd use the same format but without the */c ipm.note /m* switch.

Manual running

If you run your sendmail.wsf script manually (by double-clicking on it, for instance), Windows might warn you about the risks of doing so. Your anti-virus program might also make a stink ("Malicious Script Detected!"). But there are no risks with this benign script—all it does is read in a file and launch a program.

File locations

You can store the sendmail.wsf script file and qstring.txt file wherever you like—in your Q&A program files folder if you want. Just remember to set the paths in the script file (as well as in your @Shell programming command) accordingly.

Instant mail

See the tip on the following page for a technique that enables you to send email from your database in a slightly different way using Windows Automation.

Tom Marcellus is editor of *The Quick Answer*

The download file for this issue contains the Q&A 5.0 SENDMAIL.DTF database, two Windows Script files (one for Outlook, the other for Eudora), the SENDMAIL.BAT batch file, and a file containing the Sendmail macro.

Sesame Does Email Right

By popular demand, *Sesame's* SBasic programming language will feature a **SendMail** command that enables you to send email from your databases. Basically, it allows you to use form programming or Mass Update to send email using field values. Outgoing emails are not saved to any particular Outbox, but you can *cc* yourself if you need a copy in your main email program.

SendMail's syntax is:

```
@SendMail(SMTPServer as String, Subject as
String, FromAddress as String, ToAddress as
String, CcAddress as String, BCcAddress as
String, MessageText as String)
```

SMTPServer—The name of the outgoing mail server
Subject—Email subject
FromAddress—Sender's email address
ToAddress—Recipient's email address
CcAddress—Carbon copy email address
BCcAddress—Blind carbon copy email address
MessageText—The email message body

@SendMail returns "0" (zero) if the function succeeded (the email was sent). A non-zero return value indicates a failure.

A simple **SendMail** program with hard-coded parameters might go something like this:

```
var ii as Int

ii = @SendMail("mail.lantica.com",
"Greetings", "Erika Yoxall
<erika@lantica.com>",
"whalpern@lantica.com,amulvey@lantica.com",
"laser@HammerVE.com, "", "Hey guys, how are
you?")
```

All email addresses can be either the more personal "*Mark Lasersohn <laser@HammerVE.com>*" or simply "*laser@HammerVE.com.*"

Lists of addresses are separated by commas or semicolons.

Sesame will make sending email from your databases practically a no-brainer.

Quick Tip: Send 'Instant' Email from Your Q&A Database

The "send email" option covered in the "Send Email from Your Database" article in this issue essentially does this:

1. Starts your email program for you.

2. Passes your Q&A data to it, creating a new message.

3. Waits for you to click the "Send" button to transmit your message.

If you want your Q&A email "system" to just go ahead and send each email you compose without any further involvement on your part, you can do so with an imaginative Windows feature unimaginatively called "*Automation*."

Automation has been around for a long time. Programmers typically use it to get "hooks" into an existing Windows program (and thereby control it) from a "front end" they might have developed in a language such as Visual Basic or C++.

You can use Automation to control any Windows program that enables itself to be controlled in this manner. Outlook is such a program. Apparently, Eudora is, too, though the keys to that kingdom proved a bit beyond my reach at this writing.

At any rate, here's a Windows script (Emailer.wsf) that uses Windows Automation to pass your email-related information to Outlook as a new message to send. You never even see the Outlook screen. Depending on your Outlook options settings, the message can either be sent instantly, manually, or at the next opportunity (such as if you have Outlook set to check for new mail every hour, when it will send outgoing messages at the same time). This basic Windows script is designed to handle only the recipient's email address, a subject, and a message:

```
<job>

<object id="shell" progid="Wscript.Shell"/>
<script language="vbscript">

'#####################
'    For OUTLOOK     #
'#####################

'Reads in output file from Q&A database
'Sends email via Outlook using Automation

Set fileObj =
CreateObject("Scripting.FileSystemObject")
Set myFile =
fileObj.OpenTextFile("c:\temp\qstring.txt")
```

```
toWho = myFile.ReadLine
subject = myFile.Readline
message = myFile.Readall

message = Left(message, Len(message) -4)
message = Replace(message, vbCrLf, " ")
message = Replace(message, "  ", " ")

myFile.Close
Set fileObj = Nothing

Set WSHShell =
WScript.CreateObject("WScript.Shell")
    Set appOutl =
Wscript.CreateObject("Outlook.Application")
    Set maiMail = appOutl.CreateItem(0)

        maiMail.Recipients.Add(toWho)
        maiMail.Subject = subject
        maiMail.Body = message

        maiMail.Send

    Set appOutl = Nothing
    Set maiMail = Nothing
    Set recMessage = Nothing

</script>
</job>
```

You'd run this script from the batch file you launch from your Q&A database record, just like in the examples in the article.

But the Q&A programming that generates your qstring.txt file in this case can be a lot simpler. Here's really all you need in the **Qstring** field, at the Program Spec starting from the point where the @Askuser prompt is called:

```
If @Askuser(" ","Send the Email now?"," ") Then
{

Date = @Date; Time = @Time;

qstring = Emails + "
" + Subject + "
" + Message;

@Macro("SendMail")

}
Else
{
@Msg("Send message operation cancelled.");
Goto Name
}
```

The program creates a string whose first line is the recipient's email address, the second line is the subject, and the third and subsequent lines are the message.

—**Tom Marcellus**

Mail ('n' Email) Mergin' in *Sesame*

MARK LASERSOHN

OKAY, okay already—I heard you: Yes, a lot of people love Q&A's word processor. I mean, really love it. They don't just *like* it. They wooed it, dated it, married it, and had little applications with it. We (and when I say "we," I mean everyone else *besides* me) really wanted to write a word processor to go with *Sesame*. After all, what fun is it to dedicate every waking hour to a form-based, relational, fully programmable, report-writing, gunslinging database management system, if you don't get to write a word processor too?

But we (this time meaning *me*—yeah, its my fault) didn't write one. I'm sorry. They're really not that hard to write. A couple of man years on the outside.

Now a report writer, that's a whole 'nother kettle of monkeys. All of the programmers (except one: Erika— bless her sweet naive heart) on this project have written at least one report writer, and swore off doing another. Forever. But after tricking Erika into writing the report writer (hee hee), and now confessing to it, we (meaning— hey, where are you guys?) don't think we (meaning Erika) will have time to write a word processor.

Before you (meaning the lynch mob Erika is organizing) string me up and decide to ride on up yonder to that there general store to buy a passel of Access boxes, let me explain: Its okay. The grieving will pass. Life will go on. You can still send a form (merge) letter—even from *Sesame*.

Somewhere west of the Rio Grande

Let's set the scene. We're out west. It's night. The coyote's lonely howl punctuates the dark. A group of tough looking cowboys have a scrawny, poorly dressed programmer surrounded. The leader of the posse, Wild Bill Halpern, the local sheriff, approaches the hapless programmer, noose in hand, "So ya have any last words, ya varmint?"

"No sir..."

"Then let's get this hangin' done with. Promised these here consultants I'd have 'em back in the saloon by midnight." He starts towards the programmer, opening the noose.

"...but I really didn't think you'd miss W.P. (that's Word Processor, for all you city slickers) so much!"

"What?! I'd be blighted!" chimes in Alec "Ace" Mulvey, the fastest draw in all of London, "How are we going to correspond with clients and acquaintances, as kept in our records of such?"

"You can still do mail merge! Really, it's easy!"

"He's daft. Hang him," says Alec.

"Now hold yer horses there Ace," says Wild Bill. "Let's hear him out. So what d'ya mean—we kin still do a mail merge?"

"You can use Report Writer. It'll make a real purty mail merge!"

"But what if'n I jest have that there beta copy of *Sesame*—t'ain't got no *Re*-port *Wri*-ter not yet," exclaims one of the really angry looking consultants. "I ain't done bought no boxed *Sesame* as'n yet." He glares at Bill.

"You can use SBasic. The commands are built right in. You can use *PrintString*, and *NewPage*, and *FinishPage*!" A gasp rises from the lynch mob.

And they all lived happily ever after. [You just go on believing that—signed Erika]. The programmer explained how to use SBasic to create form letters, and when he was done, the consultants who hadn't fallen asleep or wandered on back to the saloon, made him explain all about using ODBC for the same purpose. The next day they tried to lynch a programmer from up Seattle ways— but he explained how to do a left outer join in SQL Server.

Howsabout an email merge?

Before I take the noose off completely, let me apologize once again: I'm sorry, you can't use *Sesame* to write a book. We haven't replaced a lot of the functionality that is in Q&A Write. But it is easy to create a form letter and do a mail merge with it—and it can even be done without using the Report Writer at all.

The easiest way is also the most modern. SBasic supports a new function called @*SendMail*. @SendMail allows the programmer to work things out so that email can be sent from inside *Sesame*. If you are on the "Keep Me Posted" list from Lantica, then you've received email from the @SendMail function. The @Sendmail function has the following syntax:

```
@sendmail( Server as string, Subject as string,
SenderUserId as string, DestUserIds as string,
CcUserIds as string, BccUserIds as string,
MessageBody as string) as int
```

The first argument is the SMTP server you want to use to send the email. This is the same as the outgoing server you set in your email program or web browser. It is usually something like: *smtp.your_isp.com* or *mail.your_isp.com*.

The second argument is the subject of the email, as would appear on the subject line of the email. The third argument is your email address. The fourth argument is the email address to which you want this letter sent. The fifth argument is optional (use an empty string) and specifies any "carbon copy" addresses you want to include. The sixth argument is also optional and contains any blind carbon copy addresses. The last argument is a string containing the "body" of the email itself. So, the following example would send an email through AT&T's email server from Erika to me with no carbon copies:

```
@SendMail("smtp.att.com", "Hello Mark",
"erika@lantica.com","laser@lantica.com","Hey Mark,
you better not be mentioning me in that article
you're writing!")
```

Of course, this isn't much use unless you put the programming into a Mass Update in a database full of email addresses. Assuming you have the following text fields: **Salutation**, **First Name**, **Last Name**, **Email Address**, the money field **Amount** and the date field **OwedSince**, you could then specify:

```
@SendMail("mail.isp.net", "Pay now!",
"tmarcellus@pools_R_us.com", Email Address, "",
"","Dear " + Salutation + " " + First Name + " " +
Last Name + "," + @NewLine() + @NewLine()
+ "Now look here deadbeat, You have owed me $" +
@Decimals(Amount, 2) + " since " +
@Str(@Month(OwedSince)) + "/" +
@Str(@Day(OwedSince)) + "/" +
@Str(@Year(OwedSince))
+ ". Pay now, or I'll repossess your swimming
pool!"
+ @NewLine() + @NewLine() + "Yours, Tom Marcellus"
+ @NewLine() + "Pools-R-Us")
```

This database would send an email that looks like this to one of its recipients:

```
From: tmarcellus@pools_R_us.com
To: "Bob DeadBeat" bdb@neverpays.org
Sent: Wednesday, June 11, 2003 2:05 PM
Subject: Pay Now!

Dear Mr. Bob Deadbeat,

Now look here deadbeat, You have owed me $62.95
since 4/22/2001. Pay now, or I'll repossess your
swimming pool!

Yours, Tom Marcellus
Pools-R-Us
```

But what if Tom doesn't want to send an email? He's an old fashioned guy and wants to send a real hold-it-in-your-hands letter to his deadbeat customers. Well, he's going to have to buy some stamps, but otherwise he doesn't have to leave SBasic or Mass Update to accomplish the same thing. SBasic has three routines for printing to a printer: NewPage(), PrintString(), and FinishPage(). NewPage takes two arguments:

```
NewPage(width, height)
```

The width and height are numeric arguments describing the paper you're using. For example, let's say you're using 8 1/2 by 11-inch paper, you could specify NewPage(850, 1100). That would tell SBasic that every one hundred units is equal to one inch. In other words, it divides the paper into 850 units across and 1100 up and down. If you specified NewPage(85, 110) then each inch would be ten units. If you specified NewPage(8500, 11000) then each inch would be divided into 1000 units. This is how finely you can control the printing on the page. The more units per inch, the finer the control. It also scales the text. A font specified as 10 units tall on a page divided thousands of units to the inch, would be very very small. The same font size on a page divided by 100 units to an inch, would be one tenth of an inch tall.

If you forget to call NewPage() before calling PrintString(), PrintString() will call it for you, specifying width and height as 850 by 1100, which is probably not right for you if, say, you're in Europe or normally prefer half inch margins.

PrintString() takes seven arguments:

```
PrintString(Val as string, Xpos as int, Ypos as
int, Width as int, Font as string, Fontsize as
int, MeasureOnly as int)
```

Val is the string you want to print. *Xpos* is the horizontal position where you want it to print based on the units specified in the NewPage() command. *Ypos* is the vertical position. *Width* is the optional width of a print column—set to zero (0) if you are not using print columns. *Font* is the name of the font you want to use. *Fontsize* is the size of the font—again, in the units specified using NewPage. *MeasureOnly* measures the size of your string in height and width but does not print it. Both Font and Fontsize will default to the "system font" face and size if you set Font as an empty string (i.e.: ""). If you have ever set a font face, then setting Font to an empty string will cause that font to be used again. Example:

```
NewPage(750, 1000)
PrintString("Wild Bill Halpern", 100, 200, 0,
"BArial", 33, 0)
FinishPage()
```

This would cause the string "Wild Bill Halpern" to print one inch in and two inches down on the page in the Bold Arial font one third of an inch tall. You'll note that instead of saying "Arial," I specified "Barial." Placing a space in front of the font name causes the "Normal" font to be used. Placing a "B" first causes bold. An "I" will cause italics. And "P" will cause both bold and italics.

I can hear Wild Bill's posse mounting up. "So pilgrim, what we gonna do if we donna know the length of that there field we'd be printin'?" asks T.J. "Cards" Shuflin, the meanest Tabasco drinking cowpoke this side of the Ole Mississip (but just barely on this side). Well T.J., the Xpos and Ypos can be set to negative one (-1). That causes the next bit of printing to start at the end of the last bit of

printing. Thus:

```
NewPage(750, 1000)
PrintString("Wild Bill Halpern", 100, 200, 0,
"BArial", 33, 0)
PrintString(" – The hardest typing consultant in
the Quaker state.", -1, -1, 0, " Times New Roman",
33, 0)
FinishPage()
```

would print:

Wild Bill Halpern – The hardest typing consultant in the Quaker state.

If the text doesn't fit inside the right margin and the Xpos has been set to –1, then it is printed on the next line. To determine the current Xpos and Ypos you can call @PageExtentX() and @PageExtentY(). these will return the next horizontal and vertical printing positions. If, for example, you want to start a new line, you can specify a new Ypos based on the last Y position used:

```
NewPage(750, 1000)
PrintString("Wild Bill Halpern", 100,
200, 0, "BArial", 33, 0)
PrintString(" – The hardest typing consultant in
the Quaker state.", -1, -1, 0, " Times new Roman",
33, 0)
PrintString("Have database will travel", 0,
PageExtentY() + 50, 0, "", 0, 0)
FinishPage()
```

would print:

Wild Bill Halpern – The hardest typing consultant in the Quaker state.

Have database will travel.

where the distance between the two lines is half an inch and the second line is using the last font specified (Times New Roman).

Of course a lot of those rambunctious cowpokes will want to print in some kind of column based format. That's where the fourth argument to PrintString—"Width"—comes into play. In the above examples, I have been setting this argument to zero (0), thus allowing the width to grow with the length of the string being printed. But I set it to a positive number that will limit the width of the string, possibly causing it to wrap onto the next line if it exceeds the specified width.

If for example, we have three fields to print: **Name**, **Description**, and **Cost** and we want these three fields to print in columns across the page:

```
NewPage(750, 1000)
PrintString(Name, -1, -1, 200, "", 0, 0)
PrintString(Description, -1, -1, 400, "", 0, 0)
```

```
PrintString("$" + @Decimals(Cost, 2), -1, -1,
100, "", 0, 0)
FinishPage()
```

where:

Name = "Antique Leather Saddle"
Description = "No cowboy or cowgirl should ever be without a truly fine handmade leather saddle. This saddle is made from the finest Spanish hand cured leather, dried at an altitude in excess of 10,000 feet, and hand oiled to gleaming perfection."
Cost = 1023.226

would print this:

Antique Leather Saddle	No cowboy or cowgirl should ever be without a truly fine handmade leather saddle. This saddle is made from the finest Spanish hand cured leather, dried at an altitude in excess of 10,000 feet, and hand oiled to gleaming perfection.	$1023.23

If you want to print more than a single row, you can use the @PageExtentY() after printing the first row. @PageExtentY() will return the next open vertical position below the row just printed, and can thereby be used as the Ypos argument for the next PrintString() call:

```
NewPage(750, 1000)
PrintString(Name, 0, 0, 200, "", 0, 0)
PrintString(Description, -1, -1, 400, "", 0, 0)
PrintString("$" + @Decimals(Cost, 2), -1, -1,
100, "", 0, 0)

PrintString(Name2, 0, @PageExtentY(), 200, "", 0,
0)
PrintString(Description2, -1, -1, 400, "", 0, 0)
PrintString("$" + @Decimals(Cost2, 2), -1, -1,
100, "", 0, 0)
FinishPage()
```

would print this:

Antique Leather Saddle	No cowboy or cowgirl should ever be without a truly fine handmade leather saddle. This saddle is made from the finest Spanish hand cured leather, dried at an altitude in excess of 10,000 feet, and hand oiled to gleaming perfection.	$1023.23
Old Tired Horse	No cowboy or cowgirl should be without a really old off-to-the-glue factory tired horse with a nasty disposition and propensity for throwing even the heaviest of riders.	$9.00

But what if one of your fields is especially long and may well extend beyond the bottom of the page? That's where the MeasureOnly argument to PrintString() comes into play. If you set this argument to one (1), then PrintString doesn't print your string, it merely measures

it, setting the PageExtentX and PageExtentY. These can then be retrieved before actually printing the string, so you can decide if you want to call FinishPage() and NewPage() first, and force that string onto the next page.

```
...
PrintString(Description, -1, -1, 0, "",  0, 1)
if(@PageExtentY() >= 1000)
{
    FinishPage()
    NewPage(750, 1000)
}
PrintString(Description, -1, -1, 0, "", 0, 0)
...
```

Using *Sesame* ODBC

Of course there are times when you really do need a full featured word processor to do a mail merge or form letter. And while I usually don't like to admit it, I hear tell that Microsoft's word processor isn't half bad. Combined with *Sesame's* ODBC driver, it's actually pretty easy to do a mail merge with *Sesame* from inside Word.

The first thing you need to do to use the ODBC driver is install it. Under the main *Sesame* installation directory you will (in the near future) find a directory called *SesODBC*. Under it there is a directory called *Setup32*. In that directory there's a file named "SETUP.EXE." If you go to the "Add/Remove Programs" dialog in the Control Panel and then browse to the SesODBC\Setup32 directory, you can select the SETUP.EXE file for installation. It's not installed by default.

The ODBC driver installation process is really pretty straightforward. You're first asked to select the *Sesame* driver from the list of available ODBC drivers in the installation kit. In the case of *Sesame* there is only the one. See Figure 1.
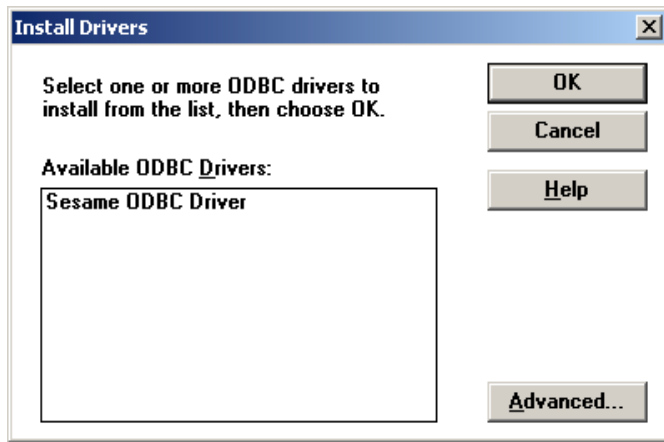

Figure 1. Installing the *Sesame* ODBC driver.

Then you're asked to either select to edit an existing data source, or press "Add" to create a new data source. Select "Add" and create a new *Sesame* data source. See Figure 2.

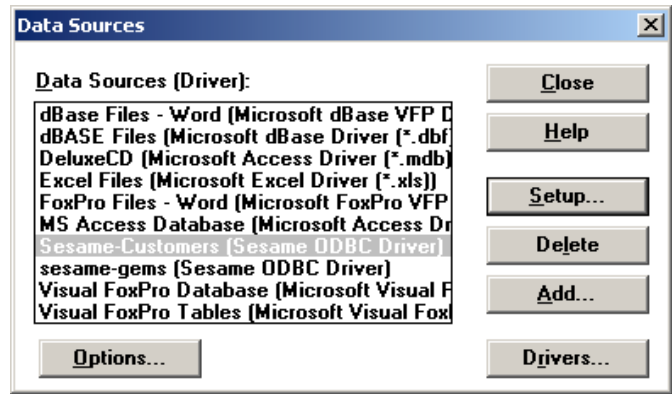Then it will ask you which driver you want to use for

Figure 2. Selecting the *Sesame* data source.

your new data source. Select "Sesame OBDC Driver" from the list. See Figure 3.
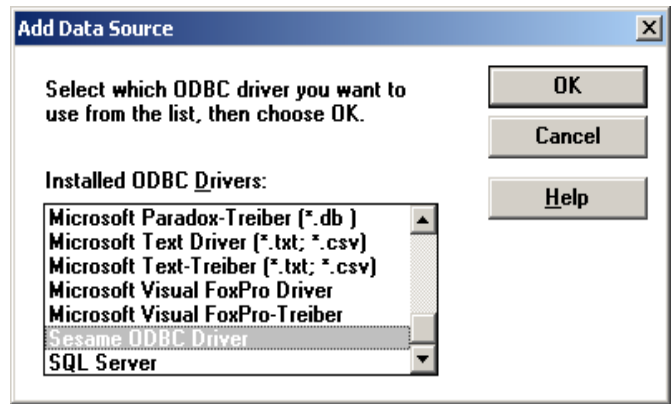

Figure 3. Selecting the *Sesame* ODBC driver.

Finally, you're asked for details about your data source. You are required to name your data source, specify the database (.db file) it lives in, the host name of your server, and the port number used to talk to the *Sesame* server. The answers shown in Figure 4 are generic. You may have to talk to your network administrator to get the correct answers for your system.

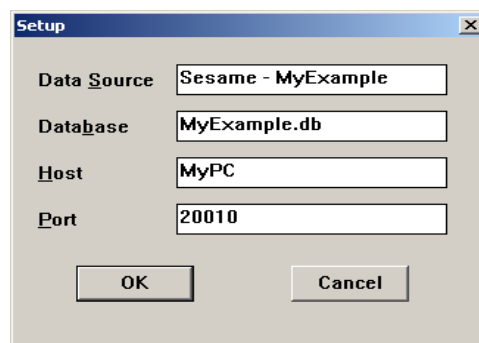Not too painful. You will have to repeat these steps for any other *Sesame* database you may want to access using ODBC. The next step is to start a *Sesame* server on the computer specified in the dialog

Figure 4. Filling in the details about the data source.

box. For this example, the command line to accomplish this is:

```
sesame –server MyPC:20010:20011 MyExample.db
```

This command tells *Sesame* to run as a server using the ports specified in the dialog box. The first port number has to match the port number ODBC is using, or the ODBC driver will not be able to talk to the *Sesame* server. The database MyExample.db is automatically opened and loaded by the *Sesame* server and will be there waiting for you when you try to access it using ODBC.

Next start Microsoft Word. From the Tools menu, select "Mail Merge." The dialog shown in Figure 5 appears.

Select "Create" from the first menu button. This will turn your ordinary Word document into a "special" mail merge document. Next, select "Open Data Source" from the "Data Source" menu. This will bring up a dialog
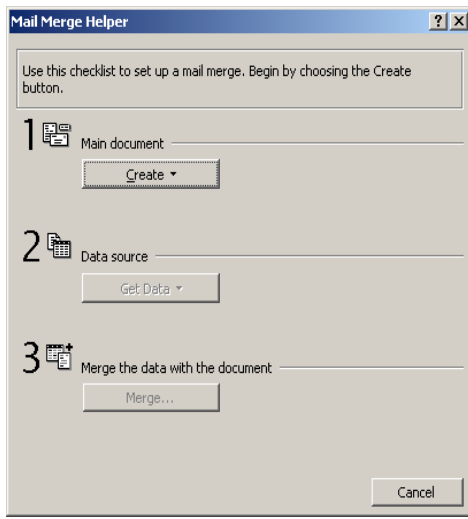
Figure 6. Using the MS Query button to select a data source.

allowing you to select from various types of files that are available data sources. Ignore these, and press the "MS Query" button near the bottom right corner of the dialog box. See Figure 6.

MS Query will then ask you to select a data source from its list of

Figure 5. Word's MailMerge Helper.

ODBC drivers. Select the name you specified when installing the *Sesame* ODBC driver. After a moment, MS Query will contact the *Sesame* server and request a list of fields. These will appear in the next dialog box as shown in Figure 7.

Select the fields you wish to include in your form letter. They will move from the left side of the dialog to the right. After selecting all of the fields that might be of use to your merge letter, click on next. You will then run through a series of dialogs for filtering and sorting the data involved in the merge letter. For now, simply click "Next" on each of these dialogs. Eventually you will get to one that asks if you would like to pass the data to Microsoft Word. Say "Yes."

At last, you can finally start adding the fields to the form letter. There is a menu just below the button bar near the left side of the window marked "Insert Merge Field." Each of the items on this menu corresponds to a field you
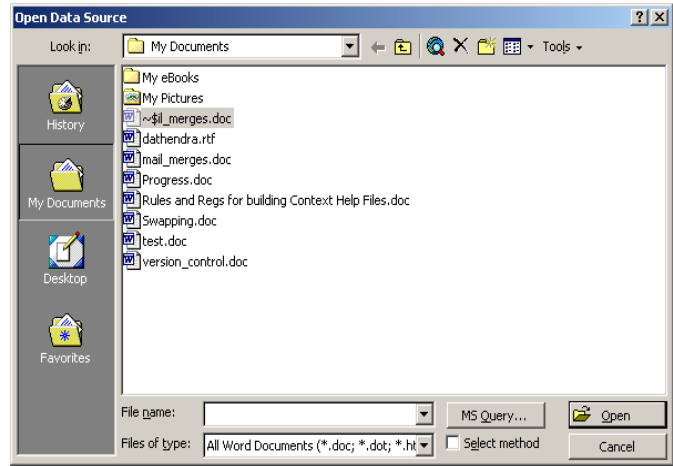
selected in MS Query. When you insert a field it will appear in your document at the current cursor position. The field name will be contained inside double angle braces: <<*field name*>>.

Further to the right, below the button bar is a button marked "Merge." If you hit this button, a dialog box will appear asking if you want to make a new document, send the merge to the printer, or send it to email. Select new document. Word will spin its wheels briefly and then begin making individual pages for each of the records in your database. Each page will feature the data from fields you selected in MS Query as they were inserted in your form letter. Once Word is finished retrieving the data, you can peruse the pages and send them to your favorite neighborhood printer.

Well, pardners, I suppose I may yet be lynched at the next Bash by a mob of heartbroken Q&A Write fans. I hope I have soothed away some of the tearful farewells, and maybe even sparked some imaginations. In any case, the trail calls.

Mark Lasersohn is a principal of Lantica Software, LLC and Technical Lead for the *Sesame* product development team. The hanging is currently off-calendar.
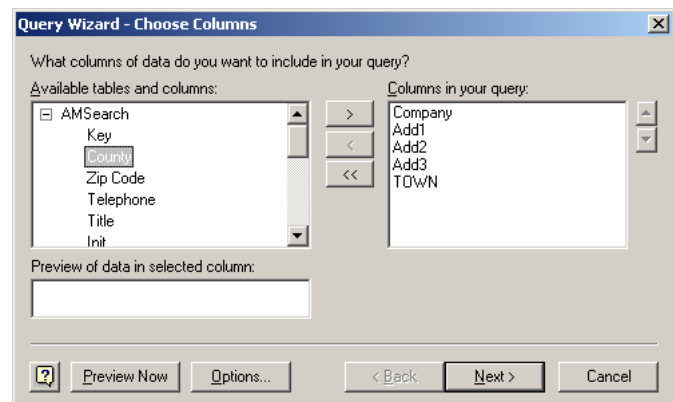
Figure 7. The list of fields from the selected *Sesame* database.

```
Heading: Amount Copy
Formula No. 2  #30
Column/Sort Spec: 110,I

Heading: Copy of Tot/Day
Formula No. 3  @total(#30,#20)
Column/Sort Spec: 150,I

Heading: Tot_Paid!by_Day
Formula No. 4  @txt(#20<>#160,@total(#30,#20))
Column/Sort Spec: 130,F(M,C)

Heading: Difference
Formula No. 5  @txt(#20<>#160,#130-#150)
Column/Sort Spec: 140,F(M,C)

Heading: Date
Formula No. 6  @txt(#20<>#160,#20)
Column/Sort Spec: 22,F(D16)

Heading: Date Copy
Formula No. 7  #20
Column/Sort Spec: 160,I
```

Results

```
                    Tot_Paid
Name    Date        by_Day   Difference
-----   -----------  --------  -----------
Bill    05-11-2003   $25.00      $25.00
        05-12-2003   $13.00     -$12.00
        05-14-2003   $63.00      $50.00
        05-15-2003   $12.00     -$51.00
John    05-10-2003   $10.00      -$2.00
        05-11-2003   $15.00       $5.00
        05-12-2003   $17.00       $2.00
        05-13-2003   $27.00      $10.00
        05-14-2003   $17.00     -$10.00
        05-15-2003   $10.00      -$7.00
```

Here, Q&A calculates the total each person pays each day (derived column #4), stores it for use in the next record (derived Column #3) and subtracts yesterday's total from today's total (Derived column #5). It prints each column *only* when the date changes (#20 <> #160).

Wrapping Information in Report Columns

I need a special emergency call list for my school that lists a lot of information for each student such as parent's names, multiple phone numbers, addresses, and so forth. The problem is that if I make too many columns in a report, the printed list is too wide. I would like to make it so it would print on a single page width for posting in the office. I need to be able to do something like this:

```
   Mother               Mom's Address
   Father                 Phones(s)
----------------  -------------------------------
Susanne Kohn       1234 Henry Hudson Parkway, #4W
Donald Kohn        Riverdale, NY 10463
                   718 888 0237(MH)
                   718 585 9321(MW)
                   877 608 8214(Mbeep)
```

Gabrielle

Yes, this is very doable with derived columns (new information created from data in the records). The only limiting factor is that you can still only work with a *total* of fifty columns in Q&A reports, whether they are visible or not. It seems to me that just the two columns above could be using as many as fourteen fields. With a few more contact numbers, the student's and dad's info could rapidly reach the fifty field limit. But with care and planning, we can get past even this.

The trick is to determine how big the widest column should be. It appears that 30 will do here. Then, select the fields that will be used to create the new columns (i.e. First Name and Last Name to make a Full Name) and make them invisible (10, I). Then create a derived column that will produce the wrapping effect:

```
Heading: Spaces
Formula: @Txt(30," ")
Column Spec: 2, I
```

This column, which just prints 30 spaces, is combined with one like this:

```
Heading: 20:Mother!Father
Formula: First NameM + " " + Last NameM + #2 +
First NameD + " " + Last nameD
Column Spec: 40
```

This, when limited to 20 characters wide, will wrap the Dad's name to the second line. The exact same procedure is used for the address and phone column except that care must be taken to:

- Use the @Str function whenever a number (like a zip code) is added to a text value. If you don't, the text will all disappear, leaving only a number behind.

- If you can't fit the entire line in one derived column, make it invisible and then use it as the beginning of the next derived column:

```
Heading: 30:Mom Address!Phone(1)
Formula: #45 + #2 + #46 + " " + #47 + " " +
@str(#48) + #2 + @str(#51) + #2 + @str(#52)
Column Spec: 60,I

Heading: 30:Mom's Address!Phones(s)
Formula: #60 + #2 + @str(#53)
Column Spec: 98,AS
```

Write Docs Too Large to Open

I don't even know how this happened. My documents, created in Q&A 5.0's Write word processor are now too large to open. There is some information in the document that is critical to me. How can I get it back?

Louis

This is a problem we've faced many times. Q&A will let you create and save documents that are too large to reopen. If the document size is close to the maximum, just

a little too large, something as simple as unloading your macro file may give you enough memory to open the document.

You could also make sure that you are not running lesspace or any other TSR and *temporarily* set the properties of the shortcut to *not* provide Expanded or Extended memory. (This will maximize your Conventional memory).

If all of the above still does not work, there is the old standby: Find someone with a copy of Q&A for Windows and open your document in that program. (There are essentially no memory limits in Q&A Write for Windows, and it recognizes the Q&A for DOS document format.) Then, break your large document into several separate, smaller documents and resave them as individual files.

Protecting Documents from Changes

This might be asking too much, but I have seen you guys do so much that was seemingly impossible that I will ask anyway. I create merge documents in Q&A write and then export them to MS Word so that I can attach them to an E-Mail. The issue is that these documents are Title Insurance descriptions created from very specific client data in the records. It is essential that the client see these but it is also essential that they not be able to change them. Even if I protect them in Word, the end user can open, modify and resave them. Is there any way I can make a non-changeable document using Q&A?

Brian

Almost, but you will need some help outside of Q&A. That help is provided by Adobe Acrobat. An Acrobat file is actually a kind of "picture" that can be attached to an email and viewed and printed by the recipient, but is extremely difficult, and somewhat costly, for the end-user to modify. The trick is to make Q&A create an Adobe Acrobat (*.pdf) file. This is really pretty straightforward but does require that you acquire a copy of the full Acrobat program, not just the free, ubiquitous Acrobat Reader.

Now the slight of hand. Use Windows to install a new *phantom* printer on your system. Use the Add Printer Wizard to install a new *Local* printer. Don't let Windows automatically search for the printer. When it asks what port, select *FILE* instead of the recommended LPT1. Select a standard HP laser printer from the Windows list, but make sure that it's a *postscript* printer (for example, HP LaserJet 2100 Series PS)! Do not attempt to print a Test Page! In Q&A, under *Utilities / Install Printer*, set up Printer D or E as:

```
Q&A PRINTER      PORT  PRINTER MODEL AND MODE
Printer E (PtrE) FILE  HP LaserJet 4M Postscript (Port)
```

Now, when you create that document you want to be an uneditable email attachment, simply print it to Printer E, giving the file name a *PRN* or *PS* extension. (This is important because it is what Acrobat will look for). Open Adobe Distiller (part of Acrobat), open the saved file from Q&A, and save the result as whatever you want with a *PDF* extension (*John Smith Title Policy NO12345.pdf*). There you have it—an Acrobat file from Q&A that can be sent anywhere.

While installing Acrobat, or afterwards, you can tell it to monitor a "watched" folder. This way, when you save a .ps file to this folder, Acrobat will automatically distill it and create a PDF file of it on-the-fly. If you frequently create PDF files, this can be a time-saver. See also "Make Your Q&A Write Documents Portable" in the June 1999 issue. —TJM

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.