

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**

L A N T I C A

S O F T W A R E L L C .

Makers of [Sesame Database Manager](#)  
Compatible with Symantec Q&A™  
<http://www.lantica.com>

# Forms Within Forms, Part I

You don't need a relational database to gain the powerful productivity boost subforms can provide.



**TOM MARCELLUS**

**O**NE advantage of relational databases is that you can place forms *inside* forms. In a Customer database, for example, you could have a *subform* for notes, and another for invoice activity. (See Figure 1.) This way, you could view, add, or delete notes and display invoice activity right from the customer's record, even though the subform data is stored in separate Notes and Invoice databases. It's the ultimate in convenience, and doesn't involve duplicate data storage.

Though Q&A 5.0 for DOS isn't relational, I'll show you how to create subforms to get relational-like functionality. I'll take an ordinary customer information database and add two subforms to it—one for notes, and another for invoice activity. You can adapt the techniques to fill your own subform needs. This month, I'll explain how to prep the subform databases (Notes and Invoices, in this case), then add the subform areas to the main database (Customer, in this case). Next month, in Part II, I'll show the programming.

(If you'd like to see how the subforms featured in this article work, but don't have the time to create the databases and type the programming, you can order the application ready-to-use. See "Resource" at the end of the article.)

## Scrolling subforms

A subform typically displays selected fields from external records in an area set aside within the "parent" database form. While in the subform, you can scroll these external records, update any editable fields, save any changes back to the external database, and return to the main part of the form at any time.

Take an Invoice Activity subform in a Customer database. There might be dozens of invoices for the customer whose record you're viewing. You'd want the subform to show certain fields from the invoices for that customer only. You might also want the option to mark an open invoice as *Paid* without having to go to the Invoice database to do it.

In a Notes subform in the same Customer database, you'd want to be able to scroll through the notes for the customer whose record you're viewing, and have the option to add or delete a note, or even search for one that contains a particular

word or phrase—without having to switch to the Notes database to perform any of these tasks.

## How the finished subforms will work

Before creating the sample Notes and Invoice Activity subforms, you'll want some idea of how they'll work in actual practice.

When you display a customer's record, you'll be in the main part of the form as usual. To activate the Notes subform, you'll simply click on the Notes button. (See Figure 1.) The first 34 characters of the most recent note will appear, with the date the note was created just above it. The first 80 characters of the note will appear on the message line. To view lengthier notes, you'll just click on the Note field, and press F6 to expand it.)

You'll click on the down arrow button to scroll

**Figure 1.** A sample Q&A 5.0 Customer database form design with subforms for Notes and Invoice activity. In this record, both subforms are active.

- 1 Forms Within Forms—Part I  
*Tom Marcellus*
- 2 Editorial—It Keeps on Going
- 2 *Tip: Easier Multiple Record Retrieval*  
*Gordon Meigs*
- 5 *Halt!—Who Goes There?*  
*Erika Yoxall*
- 6 *Tip: Shortcut Date Entry*  
*Gordon Meigs*
- 7 @Help  
*Edited by T.J. Shufflin*
- 9 *The Program Spec:*  
*Transfer Selected Records to Archive File*  
*Jeff Nitka*
- 11 *Quick-Search Files for Matching Names*  
*Gordon Meigs*



# It Keeps On Going...

TOM MARCELLUS

**W**Henever I start thinking I have Q&A 5.0 pretty well wired, something turns up to disabuse me of the idea. Case in point. The other day I was polishing up an application that auto-prints a succession of reports (the same report, but different groups of records) to disk and moves each one to a different subdirectory. (See "Tracking File Folders in the Law Office," February 1997.) The sticky part was that each report's header had to include database information that varied from one report to the next.

I remembered that the READ.ME file that came with Q&A 5.0 (it should be in your main Q&A directory) mentioned something about programming report headers. I checked it out, only to discover that such programs can't reference fields in the database. But it also described how to place information from a *derived column* in a report header—so I gave it a whirl.

I assigned columns #5 and #6 to the two fields that contained the information I needed in the report header. But since I didn't want them to appear in the report body, I made them invisible.

Next, I created an invisible derived column (the first one in the report), and filled its *Formula* line with an expression to concatenate (string together) the data from the invisible columns #5 and #6:

```
"H:\Library\Cases\" + @Str(@Num(#5)) + "." + #6 + "\Files"
```

I then placed **Run @Date @Time ! \*Pg {?1}\*** in the report header (the "1" specifies the first derived column out of a possible 16) and—*voilà!*—Q&A printed the variable information from the derived column (along with the run date and time) in each report's header. The result looked like this:

```
Run 5/22/97 3:44 pm H:\Library\Cases\89744.BIR\Files
```

Try this with your own reports. It's way cool! Works in crosstabs, too. Just expect the header data to be from the report's *first* record.

This month, **Erika Yoxall** describes a nifty technique that lets you control who can—and can't—run an application's "big" macros. **Jeff Nitka** shows how to select records to move to an archive file simply by checking them off a handy list. **Gordon Meigs** demonstrates a really clever way to auto-search any database (including the one you're in) for the person's name you just entered. And I'll show you how to build subforms *inside* your existing database forms, so you can conveniently view—even edit—data from external databases without leaving the record you're working on.

## The Quick Answer™

The Independent Guide to Q&A Expertise

Editor Tom Marcellus  
Publisher Michael Bell

The Quick Answer (ISSN 1052-3820) is published monthly (12 times per year) by Marble Publications, Inc., 9717 Delamere Ct., Rockville, MD 20850.

Cost of domestic subscriptions: 12 issues, \$79; 24 issues, \$142. Outside the U.S.: 12 issues, \$99; 24 issues, \$172. Single copy price: \$10; outside the U.S., \$12.50. All funds must be in U.S. currency. Back issues are available upon request, for the same price as a single copy.

Periodicals postage paid at Rockville, MD. POSTMASTER: Send address changes to **The Quick Answer, PO Box 9034, Gaithersburg, MD 20898-9034.**

Copyright © 1997 by Marble Publications, Inc. All rights reserved. No part of this periodical may be used or reproduced in any fashion (except in the case of brief quotations embodied in articles and reviews) without the prior written consent of Marble Publications, Inc.

Address editorial correspondence, @HELP questions, or requests for special permission to: Marble Publications, Inc., The Quick Answer, PO Box 9034, Gaithersburg, MD 20898-9034. Phone 800-780-5474 or 301-424-1658. Fax 301-424-1658. CompuServe 73370.1575.

For Q&A technical support, call Symantec 503-465-8600.

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

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



## Easier Multiple Record Retrieval



I've found that many Q&A users have never been shown one of the simplest record retrieval shortcuts there is. When preparing to update several records, they retrieve the first record, update it, press F10 to save it, then return to the Retrieve Spec to search for the next record. In a large database, particularly when searching on a non-indexed field, this can be very time consuming. An easier way is to retrieve all the pertinent records—for example, *Marcellus; Meigs; Washington; Lincoln; Roosevelt; Truman*—by typing them one after another in

the Retrieve Spec, and separating each with a semicolon. To have Q&A display them in order, press F8 for the Sort Spec, and enter *1as* in the Last Name field. Now, when you press F10, Q&A will perform *one* search, retrieve all the records you need to update, and display them in last name order. You can even mix your retrieval parameters. For example, you could type *Meigs; Washing.; Linc.; ..Roosev.; Truman* to run an exact search on Meigs and Truman, and a wildcard search on the others.

**Gordon Meigs**

# Forms within Forms

Continued from page 1

through progressively older notes, and the up arrow button to scroll back toward the most recent note.

You'll be able to delete a displayed note by clicking on the Delete button, and add a new note by clicking on the Add button, typing the note, then pressing Enter.

To search your notes, you'll click on the Search button, type the search word or phrase, and press Enter. The first matching note, if any, will appear, and you'll be prompted to search for the next match. During this process, you can press F6 from the Notes field to view any lengthy notes.

To reinitialize the subform and return to the most recent note, you'll double-click on the Notes subform button. To exit the Notes subform and return to the main part of the record, you'll click once on the Notes button.

To view the customer's invoice activity, you'll click on the Invoices button. The invoice date, invoice number, amount, and paid date (if paid) of the most recent invoice will appear. You can then click on the down arrow button to scroll through earlier invoices, and the up arrow button to scroll back through the more recent invoices. To mark an unpaid invoice paid (on the current date), you'll click on the empty Paid field and confirm the payment.

To reinitialize the subform and return to the most recent invoice, you can double-click on the Invoices button. To exit the subform and return to the main part of the form, you'll click once on the Invoices button.

Now, let's start building the subforms.

## Special key values essential

Retrieving external data into a subform involves XLookups. Saving new or edited subform data back to the external record involves XPosts. What controls this process is the index logic that links the *virtual* subform record to the *actual* external record. I'll explain.

A unique record identification scheme in the external Notes and Invoice databases isn't enough. A Speedy field containing 1997/05/1513:55 is fine for a note created on May 15, 1997 at 1:55 p.m., but it tells you nothing about for whom (which customer) the note was created.

Likewise, auto-incremented (@Numbered) invoices make sensible unique record numbers for the *Invoice* file, but they don't tell you to whom a given invoice belongs.

Key values for subform records must contain not only unique record numbers for the database in which they're stored, but also a unique value that links them to the current "parent" record of the database that contains

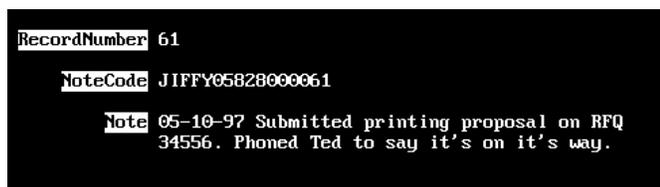


Figure 2. A typical Notes database record.

the subform. This way, the subform programming in the parent database—Customer, in this case—can find and display these external records rapidly and reliably.

Q&A's index logic is reliable only through 16 characters. My technique creates a combination key value comprised of the customer ID (or account number) up to 10 characters, and a sequential record number up to six digits. This tells you (and Q&A) *which* record it is and *whose* it is. This way, the Notes or Invoice database (or whatever the subform database might be) can contain record numbers through 999,999 before the subform record logic would break down.

## Preparing the subform databases

Creating subforms starts in the subform databases themselves. In this scenario, although Notes and Invoices are independent databases, they're treated as "child" databases in the Customer database in which they're subforms.

Notes.dtf will be totally controlled from the customer's record, where you'll add, search, and delete your notes. In other words, your Notes subform in Customer.dtf will be fully read/write. In contrast, the Invoice subform will be mainly read-only, since adding new invoices to an Invoice database from inside a customer's record wouldn't make sense.

A typical Notes.dtf record might look like Figure 2, with fields for **RecordNumber**, **NoteCode**, and **Note**. The RecordNumber field (Speedy) is a simple auto-incremented @Number field. NoteCode (Speedy) stores the unique 16-character key value that describes whose note it is (in this case, a 10-character customer ID number) and the record number (in this case 61), with the two values separated by zeroes. The Note field contains the text of the note.

For subform purposes, the only programming required in Notes.dtf is the following in the RecordNumber field:

```
< If RecordNumber = "" Then RecordNumber = @Number
```

The NoteCode and Note fields are filled by programming in the Customer database's Notes subform, as you'll see later.

To make an invoice database subform-ready, add two new text fields to it—**TransCode** (Speedy) and **TransData**. The following sample on-record-exit program in the TransCode field assumes fields named Cust ID, Invoice

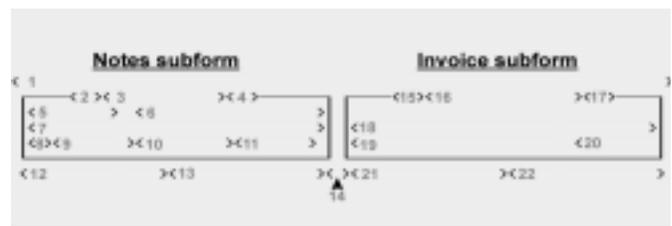


Figure 3. The bottom portion of the Customer form (the two subforms). See Table 1 for the field names and formatting.

**Table 1.** The field names corresponding to Figure 3's reference numbers. The "I" in the Other column means the field is visible only when the subform is active. (Unless otherwise noted, the form and all fields are set to black text on a white background, with the selected field white on black.) Dates are in the 05-15-97 format.

Ref	Field Name	Width/ Format	Other
1	SubformBar	78,T	Bright white on white
2	NoteUp	2,T	I
3	NoteBar	3,T,JC	
4	NoteDn	2,T	I
5	NoteDate	9,D	Bright white on white, I
6	NoteString	21,T	I
7	Note	34,T	I
8	NoteTemp	1,T	Read-only, always I
9	NoteSearch	8,T,JC	I
10	NoteAdd	10,T,JC	I
11	NoteDel	8,T,JC	I
12	NoteCode	16,T	Read-only, always I
13	NoteViewed	17,T	Read-only, always I
14	Hold	1,T	I
15	InvUp	2,T	I
16	InvBar	17,T,JC	
17	InvDn	2,T	I
18	InvCols	35,T	Read-only, Bright white
19	InvInfo	25,T	Read-only, I
20	InvPaid	8,D	Bright white on white
21	InvCode	16,T	Read-only, always I
22	InvViewed	17,T	Read-only, always I

No, Date, and Amount:

```
#100: TransCode =
  Cust ID + @Text(10 - @Len(Cust ID), "0")
  + @Right("000000" + @Str(Invoice No), 6);

TransData =
@Str(Invoice No) + " "
+ @Str(@Month(Date)) + "-"
+ @Str(@Dom(Date)) + "-" + @Right(@Year(Date), 2)
+ " $" + @Str(Amount);
If Not @Instr(Amount, ".") then
TransData = TransData + ".00"
```

The program first creates a 16-character key value for the TransCode field that identifies whose invoice it is and the invoice number. **JIFFY05828007322**, for example, is assigned to invoice 7322 for customer JIFFY05828.

Next, the TransData field is filled. If the invoice date is May 15, 1997, and the amount is \$1235.00, the TransData field is filled with the following combination value. (Each element is separated by two spaces):

```
7322 05-15-97 $1235.00
```

This way, only one XLookup will be required to retrieve this invoice information into the subform.

You can use this program to Mass Update existing invoices. When you've done so, your Invoice database will be subform-ready.

## Designing the subforms

You can now add the Notes and Invoice subforms to the Customer database. Figure 3 shows the bottom portion of the Figure 1 Customer database form (the subforms) during Design. Table 1 shows the corresponding field names, widths, formatting codes, and so forth.

After creating the subforms, go to the Palette Spec. For the subform fields, choose the field background color that matches the form's overall background color. This way, the subforms' fields won't show when the subforms aren't active, and the form as a whole will look cleaner.

The Hold field at the bottom of the subform area contains the following combination on-record-entry/on-record-exit program that serves both subforms. When you've typed the program, press F8, and type 10 (or whatever field ID number you want to use) on both the *On entry field ID* and *On exit field ID* lines:

```
#10: InvBar = "I n v o i c e s";
@Color(InvBar, 15,0);
Clear(InvCols, InvUp, InvDn, InvInfo,
  InvPaid, InvCode, InvViewed);
NoteBar = "N o t e s";
@Color(NoteBar, 15, 0);
Clear(NoteUp, NoteDn, NoteDate, Note, NoteTemp,
  NoteAdd, NoteDel, NoteSearch, NoteCode,
  NoteViewed, NoteString);
SubformBar = @Text(@Width(SubformBar), "-")
```

The program sets the text on the NoteBar and InvBar buttons — what you click on to activate the subforms — and clears the other subform fields. This way, the subforms look inactive when you're not working in them. By clearing the fields, the program ensures that the subform data isn't saved with the customer's record.

The SubformBar statement fills the screenwide field with a solid line (ASCII 196) to provide a visual break between the main form and the subform area. Program SubformBar with a <Chome command to keep the cursor from entering the subforms unless you deliberately click on the NoteBar or InvBar button fields.

Next month, I'll show you the programming for the rest of the subform fields.

*Tom Marcellus is editor of The Quick Answer and author of PC World Q&A Bible (IDG Books). Many of his databases featured in The Quick Answer are available from Marble Publications.*

## Resource

The subform application featured in this article—complete with the Customer, Notes, and Invoice databases—is available on disk for \$49 plus \$3 shipping. Send check or money order to Marble Publications, PO Box 9034, Gaithersburg, MD 20898. If ordering by chargecard, you can call 800-780-5474, or fax your order to 301-424-1658.

The Customer database (Custs2.dtf) contains the fully-programmed Notes and Invoice subforms ready for use with a mouse. (Next month's issue will include all the programming.) All three databases contain sample records, so you can install them in your default Q&A data subdirectory and immediately see how they work. The programming and other features aren't locked. You can freely modify the databases and copy their programs.

# Halt!— Who Goes There?

ERIKA YOXALL



In a multiuser environment, how do you control who can run an application's macros and who can't?



**R**UNNING macros from custom menus has its drawbacks. Aside from reliability issues, how do you automate tasks that perform password-protected operations? The approach I prefer is to create an application menu using a one-record menu database—a technique Bill Halpern discussed in “Who Needs Custom Menus?” in last June's *Quick Answer*, and at the Q&A Masters Seminar in Colorado Springs.

The problem with *any* macro-launching platform—be it custom menu or database—is that, well, it still involves *macros*. Q&A macros are “dumb”—they can't handle unexpected events, particularly password protection. If a user makes a menu selection that tries to perform an operation not allowed by the current password, the macro won't cancel itself, but will continue to pound out its keystrokes with potentially disastrous results.

To protect an application from its own macros, your menu system must have a way to verify that the current user has access to the functions a macro will perform before permitting that macro to be run. You can't do this with custom menus, but you can if you design your menu system in a database. I'll show you how.

## Security from scratch

Using Q&A's security features, you can grant or deny each user access rights to various database operations.

When someone attempts to perform a password-protected operation, Q&A checks his or her Access Control record to find out if it's okay.

Likewise, if you use a database as an application macro platform, you need a supplementary database with access control records to secure the selections.

Suppose you've created an application menu in a database named MENU.DTF. (See Figure 1.) You'll need a supplementary permissions database—let's call it PERMISS.DTF—that mirrors MENU.DTF. (See Figure 2.) In PERMISS.DTF, you'll have a record for each user with access to the application, and you'll use Q&A's security features to restrict access to PERMISS.DTF to the database administrator only.

In PERMISS.DTF, you'll need two fields for each MENU.DTF menu selection—a Yes/No permission field, and next to it a text field containing a description of the menu selection. You'll also need a Speedy text field named **UserID** at the top of the form. You can use an Initial Value or On Record Entry program to set the description fields to the various menu options.

You add a record to PERMISS.DTF for each user by typing their User ID in the UserID field, and entering Y or N in each Yes/No permission field to specify whether the user in question has access to the corresponding menu selection in MENU.DTF.

## Configuring the menu database

In MENU.DTF, you'll need a read-only **UserID** field to store the current User ID. Since the database consists of



**Figure 1.** A simple application menu designed in a Q&A database. Behind each selection is a macro that executes when the selection is clicked-on.



**Figure 2.** A permissions database that mirrors the menu database, and grants access to the menu selections on a user-by-user basis.

only the one record, you'll need an On Record Entry program like the following to prevent anyone from adding records:

```
If @Add Then
  {@Msgbox("You cannot add records",
    "to the Menu file.", "");
  @Exit
}
```

You'll also need to set the UserID field to the current User ID, like this:

```
UserID = @UserID
```

You can leave intact any programming that executes your menu selections (macros). Simply enclose these programs in French braces (“{ }”), and use If-Then-Else to conditionally execute them based on whether the user's PERMISS.DTF record contains a Y or N for the particular menu selection.

For example, if clicking on a *Post Sales* menu selection in MENU.DTF launches a macro named *Post Sales Macro*, then you'd construct your program for this

selection along the following lines:

```
< Post Sales:
If @XLookup("PERMISS", @UserID, "UserID", "Post
  Sales Permission") = Y Then
  { @Macro("Post Sales Macro") }

Else
  {@Msgbox("You are not authorized",
    "to post sales.", "");
  Goto Home
}
```

The **Home** field in the Goto statement is where the cursor “rests” while waiting for the user to make a menu selection.

Using this technique, you can make an application's menu system smarter and more reliable than a custom menu. And as the database administrator, you can restrict access to specific menu options without having to reprogram your databases.

Erika Yoxall owns Hammer Data Systems in Garrettsville, Ohio, specializing in Q&A and Microsoft Access. Phone/Fax 330-527-4018, emy103@worldnet.att.net.

## Bells 'n Whistles

You have lots of options when designing a database-controlled application menu system. One of my preferences is to prevent users from seeing any selections they can't access. (What they don't realize they can't access can't offend them.) Accordingly, I add On Record Entry programming to the MENU database that checks the PERMISS database and blanks out any MENU selections to which the user has no access. For example, if a selection is named *Post Sales*, my On Record Entry program includes the following:

```
If @XLookup("PERMISS", @UserID, "UserID", "Post Sales
  Permission") = Y Then
  Post Sales = "Post Sales"
Else Post Sales = ""
```

This way, the menu includes only the selections available to

the user. You can optionally use @Color to make unavailable options appear “dimmed.”

I also like to give users a friendly greeting, so I use a program like the following in MENU.DTF's UserID field:

```
If @Time < 12:00 then
  #10 = "Good morning " + @UserID + " ! Today is "
  + @Month$(@Date) + " " + @Str(@Dom(@Date)) + ", "
  + @Str(@Year(@Date)) + ".";

If @Time > 12:00 then
  #10 = "Good afternoon " + @UserID + " ! Today is "
  + @Month$(@Date) + " " + @Str(@Dom(@Date)) + ", "
  + @Str(@Year(@Date)) + "."
```

These are just a few examples of the kinds of bells and whistles you can add to your application menu when you design it in a database.



## Shortcut Date Entry



It surprises me to come across Q&A users who have never been shown the following data entry shortcut.

When you enter a date in a date-formatted field, you don't have to type the current year—Q&A will add it for you. Just type *5/12*, and press Tab or Enter, and Q&A will change it to *May 12, 1997* if that's the date format you've set at the Global Format Options screen for the database you're using. You can optionally type

*5-12*, *5 12*, or *may 12*, and get the same result. With the year 2000 approaching, it's wise to change your global date format to one that displays the full four-digit year—*2001*, not *01*. Then, Q&A will handle dates beyond 2000 without the need for a multi-billion-dollar federal reprogramming effort.

Gordon Meigs



EDITED BY T.J. SHUFLIN

## Postal Barcodes Needed



I need to include postal barcodes on my address labels. Can Q&A do this?

Robert Carter, Poughkeepsie, New York

Q&A doesn't have built-in facilities for postal barcoding, but there are options. Two back issues of *The Quick Answer* back issues should help you with your Q&A postal barcoding needs: September 1996, "How to Save Money on Mass Mailings," and November 1996, "Easy Postal Barcoding for Your Labels and Envelopes."

If you don't have these issues, you can order them from Marble Publications for \$10 each postpaid. Call 800-780-5474, or fax your order with charge card information to 301-424-1658.

## Don't "Balance" in a Report



I use a five-column checkbook database report that looks like a checkbook register. Here are my column headings:

Check #    Date    Payee    Deposit    Amount

Is it possible to add a *Balance* column, and have the report calculate a running balance and place it in this column?

Lee Weber, Klamath Falls, Oregon

Although it's possible to have a report derived column calculate a running balance, I don't recommend this approach in your case. Here's why. When you run the report, Q&A will need to know the starting or beginning balance. Unless you include *all* the records in the database, or have some other way to tell Q&A what the starting balance is, you won't get the results you're after.

A more practical approach is to add a **Balance** field to the database, and use a program to calculate the running balance from record to record. The first record would contain the starting balance. Subsequent records would retrieve the balance from the previous record, subtract or add the current payment or deposit, and update the balance. This way, every record in the database, including the last one, would reflect the balance as it existed at the time the record was created. A report for a selected group of records would then show the balance of the first record included in the report and, providing the *last* record is included in the report, the current balance as well.

You might find the following *Quick Answer* articles helpful: February and March 1993, "Home Finances Made Easy"; September 1993, "How to Track Multiple Accounts"; January 1995, "Automate Your Check Writing and Printing."

## Print Fields Aligned at Left Margin



Help! I need to be able to print my fields down the left margin. There are too many fields to print in columnar format across a page, and one of the fields might contain several sentences.

MaryEtta Kostelecky, Spanaway, Washington

If you're using Q&A for Windows, you can easily get what you need from a Freeform Report. In Q&A for DOS, you can use one of two methods. Both will print each selected record on a separate page.

The first method involves creating a Write merge document with the merge fields aligned down the left margin. You'll probably want to include the field labels as well. If so, then type the labels down the left margin, and place each merge field just to the right of its corresponding label. This merge document method makes it easy to enhance the output with line draw, additional text, boldface, and so forth. And you can create a variety of merge documents for even more flexibility. When you print a merge document, Q&A pauses at the Retrieve Spec for you to select the record(s) to include.

The second method involves creating a Print Spec. A Print Spec is often used with preprinted forms because it lets you tell Q&A where to print each field on the page. To create a Print Spec, select File / Print, and enter the name of the database. Then select Design/Redesign a Spec, and enter a name for the new spec. At the Retrieve Spec, select the records you want to print, then press F10 for the Fields Spec. Press F1 twice to display the Help screens, read up on the two styles of specs—*Free Form*, and *Coordinate*—and decide which will work best for you. When your new spec is finished, you can print one or multiple records. To do so, select File / Print, enter the database name, choose Print records, and select the Print Spec you designed. Answer *Yes* if you want to make temporary changes to the spec, such as selecting the record(s) to include, sorting them for output, or adding or removing a field from the Spec. If your output requirements differ from time to time, you can create several Print Specs, designing each one to give you the kind of output you need.

## Custom Label Codes in Reports



My salespeople take orders from clients who sometimes place several orders in a day. I've attempted to design what should be a fairly straightforward report—one that shows the data subtotaled first by Client, then by Salesperson. I don't have a problem until I try to add what I believe are called *custom labels* for the subtotals. When I add one to a field such as # of Boxes, it doesn't appear in the report. Can you tell me what I'm doing wrong?

Jim Ferguson, Little Rock, Arkansas

Your report was properly designed for the subtotals and totals. However, to make the custom labels work, you have to place their codes (STL for a custom Subtotal Label, and TL for a custom Total Label) in the columns where sorting breaks occur. Here, for example, is a Column Spec for the report shown in Figure 1:

```
Salesperson: 10,AS,STL(SUBTOTAL BY SALESPERSON),
              TL(GRAND TOTALS)
```

```
Client: 20,AS,STL(SUBTOTAL BY CLIENT)
```

```
# of orders per day: 30,H(# orders!per day),ST,T
```

```
# of Boxes: 40,ST,T
```

```
$ Generated: 50,ST,T
```

```
$ Profit: 60,ST,T
```

Salesperson	Client	# orders per day	# of Boxes	\$ Generated	\$ Profit
Jones	David	1	3	\$44.00	\$9.00
Jones	David	5	12	\$223.00	\$289.00
SUBTOTAL BY CLIENT		6	16	\$267.00	\$197.00
Jones	Mike	3	12	\$168.00	\$13.00
Jones	Mike	2	7	\$44.00	\$9.00
Jones	Mike	8	132	\$769.00	\$222.00
SUBTOTAL BY CLIENT		13	142	\$981.00	\$244.00
SUBTOTAL BY SALESPERSON		19	177	\$1410.00	\$341.00
Steen	Carla	1	1	\$18.00	\$1.00
SUBTOTAL BY CLIENT		1	1	\$18.00	\$1.00
Steen	Jimmy	4	14	\$274.00	\$23.00
Steen	Jimmy	2	13	\$44.00	\$9.00
SUBTOTAL BY CLIENT		6	27	\$318.00	\$32.00
SUBTOTAL BY SALESPERSON		7	28	\$402.00	\$22.00
GRAND TOTALS		26	205	\$2092.00	\$363.00

Figure 1. To print custom subtotal and total labels, be sure to place their codes and text in the proper columns.

## Transfer WordPerfect Merge File to Q&A



I use Q&A 4.0 for DOS and WordPerfect 6.1 for DOS. I have several WordPerfect mailing address lists with separate fields. These lists would be much easier to merge, update, and manage if they were in a Q&A database. Short of retyping 600-plus addresses into Q&A, what can I do?

Lisa Lonie, Philadelphia, Pennsylvania

You said your WordPerfect address lists contain *fields*, so I'll assume they're in what WordPerfect 6.x calls *Data* files (what WordPerfect 5.x referred to as *Secondary Merge Files*). The following procedure should do the job:

1. Open the merge (data) file in WordPerfect.
2. Select File / Save as.
3. Select *DOS Delimited Text* for the Format.
4. Choose *comma* as the Field Delimiter, *[CR][LF]* (carriage return, line feed) as the Record Delimiter, and *double quote mark* as the Field Encapsulate Character. (WordPerfect may propose these as the default export settings.)
5. Save the file to a name such as ADDRESS.TXT.
6. Display the ADDRESS.TXT ASCII file in Write to verify that there are quotes around the field values, a comma between each field, and each record starts on a new line.
7. Design a new Q&A database with as many fields as there are in the WordPerfect data file, plus a couple of additional fields just in case.
8. Select File / Utilities / Import / Standard ASCII. Enter ADDRESS.TXT as the file to import, and your new database as the file to receive the import. Press F10 at the Merge Spec and F10 at the ASCII Options screen.

A different kind of export procedure is required for WordPerfect 5.x for DOS. See "Quickly Convert WordPerfect Merge Files to Q&A" in the February, 1993 *The Quick Answer*. —TM.

*T.J. Shuflin is a forensic chemist and lab director of a crime laboratory in Alexandria, Louisiana. Since serving as the first President of the National Q&A User Group in 1991-92, he remains active with the group by maintaining its Internet web site at <http://www.qaug.com>.*

# Stumped?

Send Q&A questions to @Help, The Quick Answer, Marble Publications, Inc., PO Box 9034, Gaithersburg, MD 20898-9034 or fax to 301-424-1658. Please include your name, address, phone number, and your Q&A version number (and whether DOS or Windows) and a detailed description of the problem. We'll publish those questions we feel are of general reader interest; individual responses aren't possible.

# Auto-Transfer Selected Records to Archive Files

JEFF NITKA



Are your bulging databases slowing Q&A—and you—to a crawl? Here's the smart way to archive records.



A database with new records added regularly can grow large and inefficient. That's why it's a good idea to have a procedure in place that keeps such a database down to a manageable size. Most of my clients use one of the following methods to do this:

- They set up a new database each year to track just the current year's activity.
- Or, they transfer the older records to a duplicate history database, then remove the same records from the original database.

Depending on what's being tracked, I usually recommend the second approach. It lets you run reports that compare activity by time period, but doesn't limit you to a single year as the first option does. And you don't have to create a new database each year.

With this in mind, suppose you track orders in ORDERS.DTF. Periodically, when ORDERS.DTF grows large, you want to transfer selected records to an identically-designed database named OLDORDS.DTF.

The operative words here are *selected records*. If you always archive orders solely by date range (orders more than a year old, for example), the process would be easy. But suppose you want to keep some or all orders for some customers in the original database—at least for the time being. In this case, you'll want an easy way to select the orders you want moved to the archive file. I'll use an ORDERS database to illustrate, but you can adapt the technique to most any kind of database.

## How it works

You start by running a macro that creates a list (a report printed to a disk file) of all the Order IDs in ORDERS.DTF. You're then prompted to choose an action:

- 1 - Transfer Some
- 2 - Transfer All
- 3 - Quit

Option 3 cancels the procedure. Option 2 transfers every record in ORDERS to OLDORDS. Option 1 lets you review the orders in date order by customer, and type an X next to each order you want transferred. When you've done so, a count of the selected orders (up to 100, in this

case) appears, and you're prompted to begin the transfer to OLDORDS. When it's complete, the procedure deletes the same records from ORDERS. I use a database named TRANSFER.DTF as the interface for this automated procedure.

## Create the report

Design a columnar report for ORDERS named *Transfer Data*. Leave the Retrieve Spec blank. At the Column/Sort Spec, type 1AS in the Order Date field, 2AS,H(20) in the Customer Name field, and 4AS in the Order ID field. Column #3 is a derived column that places a box ("[ ]") just to the left of each Order ID number:

Heading: Box  
Formula: "[ ]"  
Column Spec: 3AS

At the Print Options screen, choose *Print to DISK*, then save and print the report to a file named *OUTPUT.RPT*. The resulting disk file will contain a list of all customers with orders in ORDERS.DTF, sorted by ascending date, then by customer name, and finally by Order ID number.

## Create TRANSFER.DTF

After you've run the initial macro that generates *OUTPUT.RPT*, the supplementary TRANSFER.DTF database takes over. It contains the following four fields:

Field Name	Format
List	Text
Count	Number
Process	Text, Read Only
Action	Text, Read Only

At TRANSFER's Navigation Spec, type > *Goto Count* in the **List** field. At the Program Spec, type the following programs in the indicated fields:

### List:

```
< if @add then
{ List = @insert("output.rpt");
  if List = ""
  then { List = "Quit"; @Macro("Escape") }
  else
  { Action = @lt( @Us("1-Transfer some," +
                 "2-Transfer All,3-Quit"),1);
    if Action = "" then Goto List else
    if Action = 1 then @Fedit else
    if Action = 2 then
      if @Askuser("Ready to transfer all printed",
                 "orders from NEW ORDERS " +
                 "to OLD ORDERS.", "Should I begin?")
      then {Msg("Transferring ALL orders. Please wait.");
```

```

        List = ""; @Macro("Transfer") }
    else Goto List
else
    if @Askuser("Are you sure you want to quit?","","")
    then @Macro("Escape") else Goto List } }
else { List = "Quit"; @Macro("Escape") }

```

In the program that follows, you'll see this line:

```

        Process = @replace(List,"
", "o") + "o";

```

When typing it, you must press Enter at the end of the first line, and use ASCII characters "o" (press Alt-F10, then Alt-9) and "•" (press Alt-F10, then Alt-7).

#### Count:

```

< if @add then
    if @in(List,"[x]") > 0
    then { Process = @replace(List,"
", "o") + "o"; Count = 0; Goto Process }
    else { @MsgBox("No orders have been marked.",
        "You will be returned to your list.", "");
        Goto List }

```

#### Process:

```

< if @in(Process,"[x]") > 0 then
{ Count = Count + 1;
  @Msg("Processing " + @str(Count)
    + " marked records.");
  Process = @rt( Process, @len(Process) -
    @in(Process,"[x]") - 2 );
  Process = @rt( Process, @len(Process) -
    @in(Process,"o") ) + "•" +
    @lt( Process, @in(Process,"o"));
  if Count = 100
  then { @MsgBox("When selecting orders for transfer,",
    "you can choose a maximum of 100",
    "orders at a time.");
    Process = @Replace(Process,"[x]","[ ]");
    Goto Process }
  else
  if @in(Process,"•") > 0
  then { Process = @replace(Process,"o","");
    Process = @rt(Process,@len(Process)-
    @in(Process,"•"));
    Process = @replace(Process,"•","");
    Goto Process }
  else
  if @lt(Process,1) = ";"
  then { Process = @rt(Process,@len(Process)-1);
    Goto Process }
  else
  if @rt(Process,1) = ";"
  then { Process = @lt(Process,@len(Process)-1);
    Goto Process }
  else
  if Process = ""
  then { @MsgBox("An error has occurred in processing.",
    "Returning you to the original list.", "");
    Goto List }
  else
  if @AskUser("Ready to transfer " + @str(Count) +
    " purchase order(s)",
    "from NEW ORDERS to OLD ORDERS.",
    "Should I begin?")
  then { @Msg("Transferring selected orders." +
    "Please wait."); @Macro("Transfer") }
  else Goto List

```

#### Action:

```
< Goto Process
```

### The programming explained

The routine inserts the OUTPUT.RPT file and expands the field, providing a convenient scrolling list to select the orders to archive. It then compiles the selected Order IDs into a list, separating each one with a semicolon. For example, if you made the following selections, the

Process field would contain 67890;24680:

Order Date	Customer Name	Box	Order ID
4/10/97	Nitka's Trivials	[x]	67890
	Widget Enterprises	[ ]	12345
4/11/97	Tom's Thumbs	[x]	24680

The routine then prompts you to begin the transfer. If you answer *Yes*, the *Transfer* macro copies the contents of the Process field (67890;24680, in this case) to Q&A's Clipboard, uses the value at the ORDERS Retrieve Spec to copy the records to OLDORDS, then uses it again to remove the copied records from ORDERS.

### Define the macros

You'll need the following two macros. Both assume that Automatic Execution is set to *Yes*, and that Order ID is the first field in the ORDERS and OLDORDS databases:

```

<begdef><altt><name>"Add<sp>to<sp>Transfer"<vidoff><esc>
<esc><esc>rp<home><sp><enter>orders<enter>transfer<sp>data
<enter><enter>ioutput.rpt<enter><left><enter><esc><esc>fa
transfer<enter><enddef>
*
<begdef><nokey><name>"Transfer"<vidoff><f6><f11><ctrlend>
<f10><f6><esc><left><enter>corders<enter>s<home><sp><enter>
oldords<enter><f6><f12><f6><f10><f10><n<esc><esc>fr<enter>
<enter><f6><f12><f6><f10><left><enter><esc>ur<enter><f10>
<esc><esc>wudd<home><sp><enter>output.rpt<enter><left>
<enter><esc><esc><esc><esc><esc><esc>m<enddef>

```

The *Add to Transfer* macro (with an Alt-T shortcut) prints the *Transfer Data* report to disk to OUTPUT.RPT, then opens TRANSFER.DTF. The *Transfer* macro performs these steps:

1. Copies the selected Order ID's to the Clipboard.
2. Abandons the record, and sets the Copy command to copy the records from ORDERS to OLDORDS.
3. At the ORDERS Retrieve Spec, expands the Order ID field, pastes the contents of the Clipboard (the Order IDs) into it, the closes the field.
4. Copies the selected records to OLDORDS.
5. Sets the Remove command to remove the transferred records from ORDERS.
6. At the Retrieve Spec, expands the Order ID field, pastes the contents of the Clipboard into it, then closes the field.
7. Removes the selected records from ORDERS.
8. Runs a Recovery on ORDERS.DTF. (Recovering a database where records have been mass deleted helps keep it in good condition.)
9. Deletes the temporary OUTPUT.RPT file.

When it comes to archiving selected records, you can't beat an automated procedure like this.

*Jeff Nitka works for a chemical manufacturer and develops Q&A applications part-time for Epoch Software, 908-874-3989. Jeff is the author of the Program Evaluator (a Q&A program debugging utility) and FaxMan (a Q&A faxing database), both of which are available from Marble Publications.*

# Quick-Search Other Files for Matching Names

GORDON MEIGS



Do you have different databases for leads, prospects, and clients, or for current stock items and non-stock items? You might need to check those other files to be sure the information you're about to enter isn't already in one of them. Here's a very cool way to do it.



**C**OMPANIES sometimes maintain a group of databases where, although the information in each is different, all of them contain some similar information such as people's names. When adding new records, it can be helpful to know if the names they're entering are already in one of the other databases.

Even if your company doesn't use databases in this fashion, just pretend it does. You're liable to discover a technique that could help fill a need you *do* have.

## Creating the capability

After typing the person's name, you'd like to run a quick check to see if they might already be in one of your other databases. Without having to exit the record and search the other databases, you'd like Q&A to pop-up a list of possible matches. How would this work? When you type in the last name and leave the field, you'd have Q&A ask you, *Do you want to search another database for this name?*

The first step is to add a field named **Key** to the last page of each database you might want to search for a matching name. It's Program Spec Field ID is #31 in all the databases. You then program these databases so that when you leave the Last Name field, the **Key** field (#31) is filled with a combination value consisting of the last and first name (*Smith,John*, for example). You make your new Key field Speedy so you can perform Lookups on it.

Next, you place a one-character field (#32) immediately following the Last Name field, and program it so that on-entry, when adding a new record, it displays an @Askuser dialog box. **Listing 1** shows the complete program, with line numbers added for reference.

**Listing 1.** Program for the one-character field that follows the last name field. The line numbers are for reference only. Don't include them in your program.

```
1. <#32: If @Add then {If @Askuser("Do you want to
2. search","other databases for","this name?") then
3. {#32 = @Shell("c:\search.bat");
4. #32 = @Insert("dblist");
5. #32 = @Replace (#32, ".dtf" + "
```

```
6. ",");
7. #32 = @Rep11as(#32,"","");
8. Usl(" --- Look In ---;" + #32, #32);
9. If @Instr(#32,"---") > 0 or #32 = ""
10. or @Len(#32) > 20 then {#32 = ""; goto #32};
11. XUsr(#32,"X#31","X#31",#30,#30,#32); #32 = "";};
12. If #48 = "" then {#48 = "Dear " + #20 + ",";
13. goto #48}
14. Else cnext
```

**Line 1:** If, during data entry, you answer *No* to the @Askuser prompt (indicating you *don't* want to search another database for this person), the program skips to line 9. There, the **Dear** field is checked. If it's empty, the program fills it with *Dear James* (or whatever the first name is) and moves the cursor to it so you can edit the entry. (You might want to change it to *Dear Jim*, for example). This **Dear** field business is, of course, optional.

**Line 2:** If you answer *Yes* to the @Askuser prompt, the @Shell command runs a DOS batch file named *search.bat*. In the system for which I designed this routine, new databases can be added at any time, so the list of databases to check for a matching name can change. Here's a sample search.bat batch file that produces an up-to-date list of databases:

```
dir /b c:\qa5\data\*.dtf >c:\qa5\data\dblist
```

Notice that the @Shell command specifies the path to search.bat—in this case, it's C:\. So for this to work on your computer, you'll need to place search.bat in the root directory of your C:\ drive. You'll also need to have the batch file specify the path to your default database files.

The batch file uses the DOS *Dir* command—the /b switch specifies *bare* format—to generate a list of the databases, in this case, in the \QA5\DATA subdirectory. The output is written to a file named *dblist* in the same directory. (The ">" sign tells DOS to print the output of the Dir command to the specified file rather than displaying it onscreen.)

Accordingly, when you answer *Yes* to the @Askuser prompt, a list of all databases in your data directory appears, *including the one you're in*. To search the file you're in, you may have to set it's Sharing Mode to *Allow* (File / Design / Secure / Declare sharing mode).

**Line 3:** The newly created list of database files—*dblist*—is inserted in field #32 courtesy of the @Insert command.

**Line 4:** All occurrences of *.dtf* followed by a carriage return are replaced by a semicolon, which serve as

separators for the items on the Userselect list to follow. Type this part of the program carefully. With the field editor open, type the double quote mark, press Enter, then type the other double quote mark on the next line.

**Line 5:** The semicolon at the end of the filename list is replaced with with nothing (a set of double quote marks). Otherwise, when the Userselect list is displayed, you'd wind up with a blank line at the top.

**Line 6:** The Userselect list is created and displayed. The first line reads --- *Look In* ---, followed by all the database filenames. (An annoyance with pop-up lists is that you can't always tell which field you're in. Adding a description line to the top of the list like this can tell you what purpose the selection list serves.)

**Line 7:** This part of the program enforces a valid selection from the list. For example, if you inadvertently select the top (information) line, or press Esc (no entry), the program clears field #32 and returns to it. The prompt, *Do you want to search other databases for this name?*, reappears.

**Line 8:** When you select a database name from the list, the program places it in field #32. The XUserselectR command then searches it for a match on the **Key** field (#31), using a starting and ending range controlled by the last name field (#30) in the current record. What appears on the resulting list are the people in the external database with the same last name as the current record.

The list displays the last name followed by the first, making it easy to spot a potential a match. If it's clear the name you're adding is new, just proceed. If it could be a match, you can take the necessary steps to verify it.

Should you want to check another database, simply press Up arrow to return to the the last name field, then press Tab to repeat the process. (You can optionally modify the program to include another prompt that could ask, *Do you want to check another file?*).

In any case, you can use this technique to run a quick check on one or more databases for potential matches without having to leave the record you're working on. By itself, the technique that displays the list of databases could be a useful feature in many applications.

Gordon Meigs is vice president and general manager of Professional Computer Technology Associates of Newtown, Pennsylvania. He teaches courses and does corporate training on Q&A, and has been designing and installing advanced Q&A business applications for more than nine years. 215-598-8440, CompuServe 71023,356.

## Inquiry Management Application

The complete lead-handling system  
for all business-to-business sales and  
marketing managers.

This 4-in-1 package doubles as the ideal  
contact management, telemarketing and  
database marketing tool.

Includes 340-page user's guide  
and free phone support.

Runs in Q&A 5.0 for DOS.

For information, please call or fax  
Walt Strasser at 310-573-4381.

**adLeads**<sup>™</sup>

A product of the Center for Strategic Communication

17065 Livorno Drive, Pacific Palisades, CA 90272

VAR / consultant inquiries also invited.



PO Box 9034  
Gaithersburg, MD 20898-9034

Periodicals  
Postage  
**PAID**  
at Rockville, MD