

# Record Categories and Pop-Up Pick-Lists

TOM MARCELLUS

For a database with different classes or categories of records, this handy point-and-click retrieval gizmo saves time and typing.

**D**ATABASES that contain *categories* of records aren't anything new. A Contacts database might store information on customers, vendors, and associates. A Products database for a bookseller might include categories for hardcover, paperback, audiobook, and magazine. If you staff combat air wings for a living, you might have a database with records for pilots, navigators, gunners, and bombardiers.

When the categories have much in common and the fields are the same (name, address, phone, notes and so on, in the case of a Contacts database), it can be far easier to work with one database rather than several.

One huge advantage of the single-database-with-categories approach is that you can swiftly change horses mid-stream. If you need a customer's address, then a vendor's fax number, then a friend's phone number, you don't have to type anything or switch databases. You can have Q&A display a pop-up list with options for Customers, Vendors, and Friends, then show another list with only the contacts in the selected category. It's like having a point-and-click Retrieve Spec built into every record.

Order entry can be simplified by first choosing the product category from a list, then getting a second list of just the products in that category. The list is "drilled-down," making it easier to find the item you want.

I'll show you how to make a database work this way. Though I'll use a simple Contacts database to illustrate, the database could be anything. If you already have a database with categories, you can add this feature to it.

One possible drawback to my approach is that it works only if each category contains fewer than 1,500 records (with an average 20-character key value). But you might find the feature so compelling that subdividing your larger categories presents no problem. A few ideas on this later.

## Designing the feature

The approach requires a supplementary database in addition to the main database. The main database for this example—let's call it A1.dtf—is a simple Contacts database (see Figure 1) containing any number of fields



# The Quick Answer

The Do-It-Yourself Guide to Q&A

April 1999

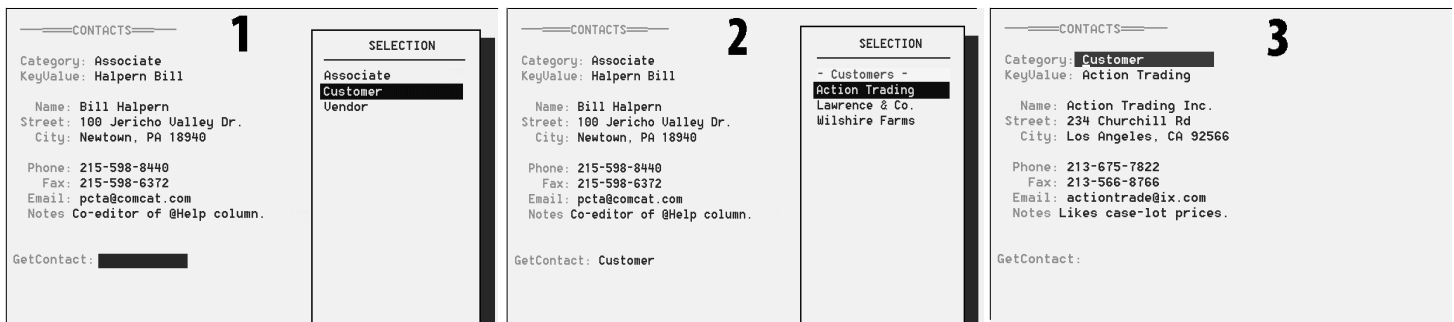
Volume 10, Number 4

- 1 Record Categories & Pick-Lists  
*Tom Marcellus*
- 2 *Tip*—Right Mouse Button  
*Tip*—LaserJet Junk Page
- 4 Required Fields and Data Integrity  
*John Dow*
- 6 @Help  
*Edited by Bill Halpern & Gordon Meigs*
  - Generic Date Calculations
  - Stamping Modified Records
  - Generate Export Files On-the-Fly
  - Network Database Path Essentials
- 7 What's New at quickanswer.com?
- 8 Pick-Lists with Default Selections  
*Tom Marcellus*
- 10 *Tip*—Create an Outlook 98 Email Address Book in Q&A
- 12 *Tip*—Keep Report Header When Printing to Disk (and Generate Pre-formatted HTML Files)

MARBLE PUBLICATIONS

and any number of records in three categories: Customer, Vendor, and Associate.

Continues on page 3



**Figure 1.** Jumping between records is as easy as 1-2-3. Simply click on the GetContact field (Left) and select the category and contact (Middle) to bring up the new record (Right). This point-and-click "Retrieve Spec" eliminates typing and database switching.

# Goof-Proof It

**F**AULTY data entry it can mess up an application something fierce. Even seemingly minor goofs like a typo or saving an incomplete record can make the information you worked so hard to gather and input unreliable.

But what can you do?

Pop-up pick-lists (Q&A 5.0) that bypass typing along with data-checking at the data entry and record-saving stages go a long way toward avoiding the "garbage-in, garbage out" problem.

This month we show you a couple of nifty pick-list techniques that cut down on the time and keystrokes it takes to complete routine tasks. You'll learn how creating classes or categories in your database can make record retrieval virtually goof-proof, and how to make a pick-list automatically respond to changing conditions.

In "Required Fields and Data Integrity" on page 4, John Dow offers an inventive technique that prevents records from being saved (and actions such as posting from occurring) until the records contain valid data.

## Quick Tip

### Don't Waste that Right Mouse Button

DOS

**Q**UICK—which Q&A key or keys do you most often press? I'm always pressing F-6 to open and close the field editor, but I still have to look at the keyboard to find it. So I assigned it to my right mouse button by typing the following macro in my QAMACRO.ASC file:

```
<begdef><nokey><name>"Right<sp>Button"<vidoff><f6><enddef>
```

This way I can *click-click* in any record—first with the left button on the field I want, then with the right button to expand it. When I'm done editing, I simply right-click again to close the editor.

If you're not using your right mouse button in Q&A to display the list of macros, why not give it a useful purpose? You can have it do anything from clearing a field or line of text (instead of pressing Shift-F4) to autotyping a canned paragraph or running a mini-procedure.

Tom Marcellus

## Quick Tip

### LaserJet 'Junk' Page

DOS

**I**N the January 1999 issue (p. 3), Alec Mulvey describes how to stop the HP LaserJet 6 from spitting out an unwanted "junk" page at the end of Q&A print jobs. Our LaserJet 4P here at Marble Publications had a tendency to do this when printing Q&A reports *with DOS Spooling turned on in Windows Printer Properties*, but we couldn't use Alec's LaserJet 6 fix. Then, Mike Bell (*The Quick Answer's* former publisher) called one day and mentioned Alec's article. Mike's busy title company uses a variety of LaserJet printers (4s, 4000s, and 5s) and he, too, had

experienced the same junk page problem. His cure was to set defaults of HIGH SPEED=OFF and ADV FUNCTIONS=OFF via his printers' control panels. (See your LaserJet printer manual.) We followed his lead with our 4P and it worked like a charm without affecting printing from Windows applications. Some of us need Windows to spool Q&A print jobs because of their lengths. See the QuickTip on page 2 of the January 1999 issue for more on the advantages and disadvantages of spooling.

# The Quick Answer™

Editor / Publisher  
Tom Marcellus

The Quick Answer (ISSN 1052-3820) is published monthly (12 times per year) by Marble Publications, Inc., 1927A Harbor Blvd., Costa Mesa, CA 92627 USA

Cost of domestic subscriptions: 12 issues, \$89; Outside the U.S., \$109; Online edition, \$59. Single copy price: \$11; outside the U.S., \$12.50. All funds must be in U.S. currency. Back issues are available upon request, for the same price as a single copy.

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

Copyright © 1999 by Marble Publications, Inc. All rights reserved. No part of this periodical may be used or reproduced in any fashion (except in the case of brief quotations embodied in articles and reviews) without the prior written consent of Marble Publications, Inc.

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

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.

### How to Reach Us

Phone 800-780-5474 / 949-722-9127  
Fax 949-722-9127  
Email mailbox@quickanswer.com  
Web <http://www.quickanswer.com>  
Mail Marble Publications  
The Quick Answer  
1927A Harbor Blvd.  
Costa Mesa, CA 92627

# Record Categories. . .continued from page 1

The supplementary database, A2.dtf (see Figure 2), contains just two fields and three records. The two fields are **Category** and **KeyArray**. The three records are Customer, Vendor, and Associate. A2.dtf's Customer record, for example, contains *Customer* in its Category field, while its KeyArray field contains a list of all the key values from the Customer category records in A1.dtf.

When you add a new *Customer* contact to A1.dtf, Q&A XPosts that record's KeyValue to the corresponding Customer record in A2.dtf. Thus the three category records in A2.dtf (Customer, Vendor, and Associate, in this case), taken as a whole, contain all the KeyValues for all the records in A1.dtf.

This arrangement lets you click on a **Get Contact** field in the main Contacts database (A1.dtf) to display a list of categories in A2.dtf, followed by a list of contact names within the category you select. Using it couldn't be easier. You choose the category then the contact and Q&A displays the record. When you add a new record to A1.dtf, you don't have to do anything—programming takes care of updating the corresponding A2.dtf record.

Suppose A1.dtf contains the following fields:

```
Category:
KeyValue:

Name:
Address:
CityStateZip:

GetContact:
```

The **Category** field in a record might contain *Associate*. If the **KeyValue** field contained *Blow Joe*, then the **Name** field would probably contain *Joe Blow*, and the **Address** and **CityStateZip** fields would be Joe Blow's address.

Now suppose you have (Associate) Joe Blow's record onscreen when you want to check something in (Vendor) Acme Supply's record. You click on the GetContact field, click on Vendor, and up pops a list of all your vendors. You click on Acme Supply, and there's the record.

The KeyValue entry can be structured differently for each category, but you should try to limit the length to fewer than 20 characters. (A Userselect list shows only the

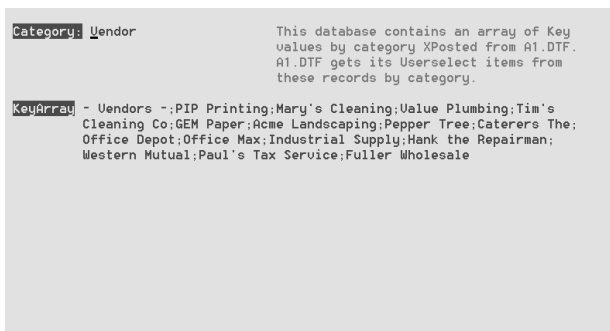


Figure 2. A typical supplementary database record (A2.DTF). This one contains the key values for the Vendors category.

## Why XUserselect Lists Can't Do the Job

You might think Q&A 5.0's XUserselectR *range* command would be ideal for displaying category pick-lists—but it isn't. Though it lets you specify a *range* of records for a selection list, that range is one-dimensional—limited to the external key field you specify. In other words, while XUserselectR can show you a list of companies that start with S, it *can't* restrict the list to just companies where the Category field contains *Customer*. (That's *two* dimensions.) To make use of XUserselect, you'd need a separate Speedy field in the database for *every* category of record.

first 19 characters.) For an Associates category you might use all personal names. For a Customers category you might have a mix of personal names and companies. (Some customers are persons, others are companies). The Vendor's category might be strictly company names. All you have to know is that the way you enter the KeyValue is how (and where) it will appear on the selection list. For companies, the KeyValue will be the company name; for individuals, you'll probably want the last name followed by the first name with or without a middle initial. Commas aren't permitted.

There are various ways you could program A1.dtf to auto-fill the KeyValue field during data entry, but that's outside my scope here. We're concerned solely with the action of selecting a category then an individual record from back-to-back pick-lists.

Here's the essential programming for the A1.dtf fields, followed by an explanation:

```
Category:
< If @Add and Category = "" Then {
  Category = @XUs("A2", "Category"); Cnext }

KeyValue: #100:
If @Instr(@XLookup("A2", Category, "Category",
"KeyArray"), KeyValue) = 0
then XPost("A2", Category, "Category",
@XLookup("A2", Category, "Category","KeyArray")
+ ";" + KeyValue, "KeyArray")

GetContact:
< If @Update then {
GetContact = @XUs("A2" , "Category");

If GetContact <> "" then GetContact =
@Us(@XLookup("A2", GetContact, "Category", "KeyArray"));

If GetContact <> "" then
if @XLookup(@Fn, GetContact, "KeyValue", "KeyValue")
<> "" then @Macro("GetContact") }

The GetContact Macro:
<begdef><nokey><name>"GetContact"<vidoff><f11><capsf4>
<f7><f3><dn><f12><f10><enddef>
```

Concludes on page 11

# Required Fields and Data Integrity

JOHN T. DOW

Don't let data entry goofs defeat your best laid plans. Proof your applications with this clever validating scheme

Application in  
**ONLINE Edition**



**D**ATABASES have features that help their designers ensure that all the data values entered into them are clean and trustworthy. After all, why go to great lengths to design the screens, figure out all the programming, and do daily backups when the information being added is wrong or incomplete?

Although Q&A has tools to ensure data integrity, you need a little cleverness to use them to your best advantage. For example, you might enter a date that's not in a valid format. Q&A will alert you to this, but will allow you to leave the bad data in the field. The basic problem is that *Q&A doesn't prevent you from saving a record that has an invalid value in some field.*

Here's a more complex problem. Suppose you're designing a scheduling system for a conference. To prevent overbooking, you want the system to post in real-time to another database to show that a conference seat has been booked. But you don't want the posting to occur until you have complete and valid information on the person attending the conference and have determined that there's still space available. Once posted, you want that person's record to show that he's booked for the conference, and you don't want the system to allow him to be booked twice.

The challenge here is to prevent the record from being saved until certain fields have certain values and an action (posting) has been satisfactorily completed. In this case, the database should be robust enough to prevent the following kinds of things from happening.

- The data entry clerk enters some of the information and tells Q&A to schedule the person for the conference. The clerk then continues to enter more information and finishes up by booking the same person again.
- The clerk brings up a person who was booked previously, changes some information, then inadvertently books him a second time.
- The clerk enters the information, saves the record, but forgets to book the person.

## What validation tools does Q&A offer?

Fortunately, Q&A has a feature that can be used to solve these problems. It's the *Required* or *Mandatory* field.

In Q&A DOS, you put a "!" (exclamation point) in the field at the Restrict Spec. Then, if you attempt to save the record without having a value in that field, Q&A will tell you, "*This field requires a value. Please enter a value before continuing.*"

The save action is terminated, and until you deal with the mandatory field, Q&A won't save the record. Programming control returns to the fields, whether you're in Add Data or Update mode.

In Q&A Windows, the *Restriction* column of the *Database Structure* table will say *Mandatory* for the field. If you attempt to save the record without having a value in the field, Q&A Windows tell you, "This field requires a value," and you won't be able to save the record until the field contains something.

With Q&A DOS, to leave the record, you have to escape out of it without saving it. In Q&A Windows, you have to delete the record.

Clearly, Q&A is very insistent about these required or mandatory fields. And you can use this insistence to your advantage.

## The challenge

Let's say we have a simple database with four data fields. One field is named *Required* because, in our example, it's a required field. However, we won't tell Q&A via the Restrict Spec it's required. Instead, we'll handle it with programming. Another field is named *NotRequired* because, well, it's not required.

There's also a *StartDate* field and an *EndDate* field. *StartDate* is a required field but *EndDate* isn't. However, if both are filled, then *EndDate* must not contain a date earlier than *StartDate*. Either date, if present, must be a valid date. This too will be handled with programming.

Notice that the conditions that determine when a record is "savable" can be complex and depend on interrelationships between fields. Whatever you can dream up and program can be the conditions that have to be met.

In this particular example, we want to set things up so that this record can't be saved unless all the following conditions are met:

- The *Required* field must have a value.
- *StartDate* must contain a valid date.
- *EndDate* can be blank or contain a valid date that isn't earlier than *StartDate*.

## The solution

In addition to the data fields describe above, I've added four more fields as follows:

1. A mandatory or required field named **Mandatory**. If this field is empty, the record isn't ready to be saved. If it isn't empty, Q&A will allow the record to be saved.  
You might want one or more other required fields. By making a field mandatory, you can't leave it without entering a value. However, it might confuse things if a user tries to save the record with multiple mandatory fields blank. Experiment to see what works best. Remember that you can make any field or any combination of fields and values mandatory through programming. They're simply all based on the one field (named Mandatory, in this case).
2. A read-only field named **ROStatus** (Read-Only Status). This field describes the status of the record based on whether the above criteria are being met. Strictly speaking, you don't need this field, only the Mandatory field. However, it can be very useful to have a field that tells you the status of the record. It should be a read-only field so it can't be altered, then you can trust that its value is accurate.
3. A field named **ValidationProcedure** programmed to set the ROStatus and Mandatory fields. This field contains a calc program that's executed by Q&A on leaving any changed field. I'm using #9998 as the field ID number for this field. (If you like, you can experiment with Gosubs instead of relying on automatic calc mode.)
4. A field named **RecordEntryAndExit** with programming identical to that in ValidationProcedure. This is made the on-record-entry and on-record-exit field. (You can't rely on just the ValidationProcedure field to work in calc mode because the calc mode can be changed from automatic to manual by the user. So to ensure a robust system, it's probably good to make it an on-record-exit field. It's nice to make it a record entry field, too, so you can be certain that your ROStatus field always displays the correct record status.) I'm using #9999 as the field ID number for this field.

The programs for the ValidationProcedure and RecordEntryAndExit fields follow. Unfortunately, the programs have to be in both fields. If one field is used as a calc field and for on-record-entry and on-record-exit as well, it won't work as a calc field anymore. The program shown here is for field #9998, but you need the same program for the other field, #9999.

```
#9998:
If Required = "" then {
  ROStatus = "REQUIRED is blank.";
} else if @Rest(StartDate,"/") then {
  ROStatus = "STARTDATE is blank.";
} else if StartDate = "" then {
```

```
  ROStatus = "STARTDATE has an invalid date value.";
} else if @Rest(EndDate,"/") and EndDate = "" then {
  ROStatus = "ENDDATE has an invalid date value.";
} else if EndDate <> "" and EndDate < StartDate Then {
  ROStatus = "ENDDATE is before STARTDATE.";
} else {
  ROStatus = ""; };
If ROStatus = "" Then {
  MANDATORY = "Ready";
} else {
  MANDATORY = ""; };
```

This uncomplicated bit of programming logic does all the required tests for the three fields, then sets both ROStatus and Mandatory.

Notice that if the tests are not met, Mandatory is cleared. This prevents you from saving the record. If the tests are met, Mandatory is set to "Ready," and Q&A will let you save the record.

At the same time Mandatory is set, ROStatus is also set. Why both? ROStatus is read-only, so there's no chance it will be altered somehow by the user. But you might want to use ROStatus to indicate other conditions that the record might be in. For example, if the record is part of a scheduling system I mentioned earlier, after deciding that all fields have valid values, it's then possible to post to another database. Having done that successfully, you could set ROStatus to indicate that posting has been done. This clearly shows the status of the record and can be used to prevent posting a second time as well.

Although the ROStatus field is a useful idea, it isn't necessary for this data integrity technique to work.

Another piece of programming you should have is field exit programming in Mandatory. This is so that if control is passed to the field somehow by mistake, it will be cleared and control will promptly go elsewhere. You want to prevent it from being treated as a normal data entry field. This is also the place to display a message to draw the user's attention to the problem that has been diagnosed and described in ROStatus. Finally, after displaying the message, the program can send control to the proper field so the user can correct the problem. Here's the program for Q&A 4.0 DOS and Windows. (In Q&A 5.0 you'd use @Msgbox instead of @Msg):

```
>
If MANDATORY <> "Ready" Then {
  MANDATORY = "";
  @msg(ROStatus);
  If @left(ROStatus,1) = "R" Then goto Required;
  If @left(ROStatus,1) = "S" Then goto StartDate;
  Goto EndDate; }
```

Notice that this program only tests the first letter of the text in ROStatus, since in this case it uniquely identifies the data field (Required, Startdate, or EndDate). In other cases a more complete test may be required.

You'll also want to use navigation programming to try to keep the user from entering the Mandatory field by mistake. You can have on-field-exit navigation in the preceding field to keep the cursor away from the Mandatory field.

*Concludes on page 9*



## Generic Date Calculations for Month DOS WIN

I have a record retrieval problem I've been trying to solve. A macro on a custom menu runs a report. It pauses at the Retrieve Spec after "pressing" Alt-F8 for saved programmed date retrieval specs that include such things such as *This Month*, *Last Month*, *Next Month*, and so on. However, for year end or year beginning, I have to include *Any December* and *Any January* for these months. Has anyone come up with general Retrieve Specs that would perform proper *Last Month* and *Next Month* record retrievals regardless of the month? It would certainly cut down on confusion.

Jim Tomsic, Via the Internet

This can be done with the help of a simple Lookup Table. The trick is to use actual date math rather than just the months so that you don't have to worry about January and December. The kicker is that it will be one day off for reports run in February of a leap year (but could be modified for that condition once every four years). The saved retrieve specs should look like these:

### Last month record retrieval:

```
#10: >={ @Date-@Dom(@Date) - @Tn(@Lu(@Month(@Date),1)) }  
.. <{ @Date-@Dom(@Date) + 1 }
```

### Next month record retrieval:

```
#10: >={@Date-(@Dom(@Date) + 1) + @Tn(@Lu(@Month(@Date),2)  
+ 2)} ..<={ @Date-(@Dom(@Date) + 1) +  
@Tn(@Lu(@Month(@Date),2) + 2) +  
@Tn(@Lu(@Month(@Date),3)) }
```

The Lookup Table looks like this:

Month	1-End This Month	Days	1-End Next Month
1	30	31	27
2	30	28	30
3	27	31	29
4	30	30	30
5	29	31	29
6	30	30	30
7	29	31	30
8	30	31	29
9	30	30	30
10	29	31	29
11	30	30	30
12	29	31	30

You can also use this approach in programming your database. Suppose you want to have Q&A calculate the first day of the next month for field #50 when you enter any date in field #5:

```
>#5:REM("First day of next month");  
#50=#5-(@Dom(#5)+1) + @Tn(@Lu(@Month(#5),2)+2)
```

Or the first day of the previous month:

```
>#5:REM("First day of last month");  
#50=#5-@Dom(#5) - @Tn(@Lu(@Month(#5),1))
```

Or even the last day of the next month:

```
>#5:REM("Last day of next month");  
#50=#5-(@Dom(#5)+1) + @Tn(@Lu(@Month(#5),2)+2) +  
@Tn(@Lu(@Month(#5),3))
```

Once you have the date, you can also get the day of the week, the number of days left in the month, the number of days from the first day of the month, and so on.

For more date calculation tricks like these, see "(Conceivably) Useful Date Calculations" in the June 1996 issue, and search *The Quick Answer* Topic Index (search on *date*) at [www.quickanswer.com](http://www.quickanswer.com).

## Stamping Modified Records DOS WIN

I need a Q&A 4.0 program that logs the User ID and date of anyone who updates a record. The problem with the program I have now is that it fills the UserID and date changed fields even if someone *just displays* a record. I want to know only if the record has actually been changed.

Linda Howard via the Internet

An often overlooked but valuable feature added to Q&A 5.0 is the @Modified function. An *If @Modified then . . .* program will execute only when the record is changed. Unfortunately, this is no help to you in Q&A 4.0. About all you can do in 4.0 is use a combination of programming and automatic calculation.

Suppose #10 is the User ID field and #20 is the date-



## Stumped?

Send your Q&A questions to @Help, The Quick Answer, Marble Publications, Inc., 1927A Harbor Blvd, Costa Mesa, CA 92627 or email to [mailbox@quickanswer.com](mailto:mailbox@quickanswer.com). Include your name, address, phone, and your Q&A version number (and whether DOS or Windows) and a detailed description of the problem. We'll publish those questions we feel are of general reader interest; individual responses aren't possible.

updated field. Place the following program in field #10. (Both fields should be read-only):

```
#10: If @Add and #10 = "" Then
{#10 = @Userid; #20 = @Date};
If @Update Then
{#10 = @Userid; #20 = @Date}
```

Next, display any record, press Shift-F8, and set Calculation Mode to *Automatic*. From now on, whenever anyone adds a record, the User ID and date will be placed in the fields. Whenever someone changes a record, the same fields will be updated.

If you wanted to log a change to any of just a few fields, on-record-exit @XLookups could compare the fields in the displayed record with the same ones in the record that exists on disk.

## Generate Export Files On-the-Fly



I have a database that contains the following fields:

```
Question:
Answer:
Export: >
```

What I want to do is build a text string and export it as an ASCII (text) file for use in a Microsoft Word document. Here's the program I'm using to generate the export file:

```
>#11: If Export = "Y" then
If @Askuser("Export this record to Word","")
Then Export = Question + ";" + Answer;
Export = @Shell("echo " + Export + " >>
D:\QA5\DATA\BTD1.txt ")
```

If the export string is less than 120 characters, everything works fine. But if the question plus the answer exceeds that, the text file isn't created. Any suggestions?

Dan Strickland via the Internet

The problem you're having here isn't with Q&A—you're simply running into a DOS command line limit. In the August 1998 issue, Tom Marcellus shows how to export practically an entire Q&A database record by assembling the text in a field like your Export field, then executing an @Macro command that runs an Alt-F9 External Program with the %g switch to create a temporary file. The External Program launches a batch file that reads the %g file as a %1 variable and saves its contents to a disk file. The technique places no limit on the amount of text you can save to an ASCII file for use elsewhere.

## Network Database Path Essentials

**DOS WIN**

On our "main" computer, shared databases are stored in the *d:\shared\database\* subdirectory. Another computer accesses these files using *f:\database\* (its "f" drive is actually the main computer's *d:\shared* drive. I've programmed XLookup statements to refer to *d:\shared\database*. They work fine—but only on the main computer. On the other computer, we get the error message, "Cannot open the

*lookup file d:\shared\database...*" If, on the secondary computer, I change the XLookup statements to refer to *f:\database\...*, it works. But it then doesn't work on the main computer. What do I need to do here?

Cheryl Lyon via the Internet

At the Utilities / Set Global Options screen, make sure the path on *each* computer is set to the directory where the database resides. Then, edit your programming statements to include *just* the database name. *Do not* include the path; Q&A will default to the path you've specified. *Never* include path names in your programming statements unless there's absolutely no way to avoid them. Keep all your databases in the same subdirectory and make sure it's the subdirectory you've set as the default for your database locations on each machine. It doesn't matter if it's drive F:\ on one computer and drive D:\ on another. The database default location is controlled by the default you've set in Q&A.

That said, the only other cause of the problem could be if you're using a Network Pack *without* using a *-p* personal path switch on each workstation. In this case, if your drives are mapped differently on the different workstations, each machine will pick up the path of the *last* person who quit Q&A. The personal path switch takes care of this. Run Q&A using the command line:

```
qa.com -pC:\mydirectory
```

Where *C:\mydirectory* is the location for *that* workstation (where it stores its own Q&A configuration files).

Bill Halpern and Gordon Meigs own Professional Computer Technology Associates in Newtown, Pennsylvania, 215-598-8440, [www.pcta-usa.com](http://www.pcta-usa.com), email [pcta@comcat.com](mailto:pcta@comcat.com).

—What's new at—

## www.quickanswer.com?

- **Quick Answer Online Index Database.** Search T.J. Shufflin's actual Q&A database of back issue topics using Keywords. Current as of March 1999. *Access from Home* page.
- **Area Codes Database.** Enter the area code to find the state or province and time zone. (Or enter the state abbreviation to find its area codes.) Use as a lookup file. Current as of March 1999. *Free Downloads* page.
- **Gender Database.** Why type "Mr." or "Ms." in that Salutation field when a Lookup to GENDER.DTF can do it for you? 4,000+ first names with salutations. *Free Downloads* page.
- **Y2K Patch** fixes imminent problem with Q&A file dates later than December 31, 1999. *Free Downloads* page.
- **The Q&A Year-2000 Pack** now includes a diskette with a utility that helps you locate potentially hazardous Date fields and fields that reference them. *Home* page *Y2K* link.
- **Latest QAPRINT.CFG and PRINTER.ZIP** (font) files for Q&A DOS. These Q&A 5.0 files work in 4.0. *Free Downloads* page.

# Pick-Lists with Dynamically Changing Default Selections

TOM MARCELLUS

Yep—your Userselect lists can display default selections that change with changing conditions.

**H**ERE at Marble Publications, we don't like pounding the data entry keys any more than the next guy, so we make liberal use of Userselect pick-lists. But we don't always use them as they come "out of the box." For some data entry tasks, we design "live" pick-lists that automatically change with changing conditions.

One of our databases requires separate fields for the month, day, and year. Since the correct date is often—but *not always*—the current date, we use a series of three Userselect lists to fill these fields, *and we make them default to the current month, day, and year.*

In other words, if today happens to be April 20, 1999, our **Month** list automatically defaults to *April*, our **Day** list defaults to *20*, and our **Year** list defaults to *1999*. This way, if the record requires the current date, pressing Enter three times completes the task. Should the record require a different month, day, and/or year, we simply select them from the lists in the usual way. It's a real time- and keystroke-saver—and it helps cut down on the errors.

But how do you create a Userselect list with a default selection that isn't the first item on the list? And how do you make that default selection automatically *change* with changing conditions? Let me first show the programming. Then I'll explain the principles that make this handy-dandy feature work.

## The database

The database contains text fields named Month, Day, Year, and Counter, and a date field named Date. The Month, Day and Year fields are filled by Userselect lists that default to the current month, day, and year, respectively. Here are the programs:

### Month:

```
< IF Month = "" then {
Counter = @Month(@Date); Gosub Counter;
Month = @Mid(@Userselect(
" 1 January, 2 February, 3 March, 4 April," +
" 5 May, 6 June, 7 July, 8 August, 9 September," +
"10 October,11 November,12 December"), 4,12); Cnext }
```

### Day:

```
< If Day = "" then {
Counter = @Dom(@Date); Gosub Counter;
Day = @Userselect(
" 1, 2, 3, 4, 5, 6, 7, 8, 9,10,11,12,13,14,15,16,
17,18,19,20,21,22,23,24,25,26,27,28,29,30,31");
Cnext }
```

### Year:

```
< If Year = "" then {
Counter = @Year(@Date) - 1997; Gosub Counter;
Year =
@Userselect("1998,1999,2000,2001,2002,2003,2004,2005");
Cnext }
```

### Counter:

```
< If Counter = "" then CNext;
If Counter < 2 then { Clear(Counter); Return }
else {
Counter = Counter -1;
@Macro("Default Select");
Goto Counter }
```

### Date:

```
< Date = @Todate(Month + " " + @Str(Day) + " " +
@Str(Year));
If Date = "" Then { Date = "ERR";
@Msgbox("", "No such date. Please correct it.", "") }
```

Here's how the Month program works. If the field is empty, the Counter field is set to *@Month(@Date)*, an expression that evaluates to the *number* of the current month. If that month is April, Counter is set to 4 and control passes to Counter courtesy of the Gosub.

Since Counter is greater than 2 at this point, 1 is subtracted from the 4, leaving 3, and the *Default Select* macro is run. As you can see here, all the macro does is execute one down arrow keypress:

```
<begdef><nokey><name>"Default<sp>Select"<vidoff><dn><enddef>
```

But the wild thing is that *the keypress doesn't occur at the point where the @Macro command appears in the program.* Q&A buffers (stores) it while the program is executing.

What happens instead is that control is *sent back* to the beginning of Counter's program. Counter is now 3, so 1 is subtracted from it, leaving 2, and the macro is executed again. Now there are *two* down arrow keystrokes in the buffer. Since the Counter value is still *not less than 2*, 1 is subtracted from it, leaving 1, and the macro is executed a *third* time. At this point Counter *is* less than 2, so Counter is cleared and control is passed back to the calling Month field via the Return statement.

The next command to execute in the Month field is the @Userselect. It displays the list with the months in chronological order since each one is preceded by its month number. At this point, the three down arrow keystrokes in the macro buffer are released, which highlights (selects) the fourth item on the list—*April*, the fourth month of the year. And that's how the current month becomes the default selection. (The @Mid function returns just the selected month name to the Month field.)



The default-setting Day and Year programs work on the same principal. The only difference is that for the Year you have to establish a baseline year that's one less than the first year on the selection list. In this case, 1998 is the first year on the list, so the baseline is `@Year(@Date) - 1997`. Thus if the current year is 1999, Counter is initialized to 2, resulting in the Counter program running the macro just once which selects 1999 (the second year on the list) as the default. If the current year were 2000, Counter would be set to 3 and the macro would execute twice, making 2000 the default.

The logic you employ to set the default selection on a Userselect list can be simply stated:

*To make the nth item on the list the default, initialize the Counter field with n.*

This has nothing to do with the list selections themselves, *only their relative positions on the list*.

The optional Date field is filled by programming, so you need the `@Todate` function to validate its entry. This traps the error if you were to select the date February 29, 1999 (not a leap year) or April 31 of any year, to mention just two examples.

So, as you can see, you need three things to make the feature work:

1. Programming logic that calculates the position of the default item on the list.
2. The macro that moves the default selection down one.

*Concludes on page 11*

## Required Fields ... continued from page 5

### How it all works

Suppose you enter only some of the data or you enter invalid data and then press F10 to save the record and start a new one. Alternatively, you're using Q&A Windows and you click on File / Exit. Or, as I mentioned above, you're using Q&A Windows and you tell Windows itself to shut down.

In any of these cases (and in others), Q&A will execute the record exit code and clear the Mandatory field because your program detects a problem. Because your programming clears the field, Q&A will go to the field and tell you that you can't save the record because the field is blank. If you continue to press Enter or F10, Q&A will continue to display the message that the field is required.

You can't save the record, exit from Q&A, or even shut down Windows! Talk about power!

At this point, unfortunately there's a brief gap in the process. When Q&A puts the cursor into the Mandatory field, it doesn't activate any on-entry programming or navigation. It just goes to that field and sits there. Unfortunately, users will have to be told what to do should that happen—namely, to type anything at all and press Enter. That will allow the cursor to leave the field, and the field exit programming will be invoked.

(This pause at the Mandatory field is avoided in Q&A Windows if you're using my DTFWIN program because it automatically presses Enter. I plan on having my DTFBOOST program provide the same feature for Q&A DOS databases.)

In any case, the record hasn't been saved, and control is back in the hands of your programming. Regardless of what the user enters in the Mandatory field, when exiting from it your field exit logic can determine the problem and send control to the field needing correction.

At this point, if you're using Q&A 5.0, you could display an `@Msgbox` message that says something like,

*"You cannot save this record for the following reason: StartDate is blank."* If you're using Q&A Windows with DTFWIN, with a slight variation in the programming you can display the same message.

The table below recaps the four fields I've discussed.

ValidationProcedure and RecordEntryAndExit should both test all the conditions you want met. If any is not met, clear Mandatory and set ROStatus to show which condition was not met. If all are met, set Mandatory to "Ready" and clear ROStatus.

On exit from Mandatory, test ROStatus to see what condition wasn't met. The ROStatus field should be visible so the user can see what condition wasn't met. However, if using Q&A 4.0 or Q&A Windows, you could also use `@Msg` to tell the user about the problem (or `@Msgbox` if you're using Q&A 5.0). Then, go to the proper field so the user can correct the problem.

With a few lines of code and a few extra fields, you can ensure that your critical data fields are clean. You can even prevent actions (such as posting) from taking place when the system isn't ready for them.

Fields Needed for Validation Technique	
<b>ValidationProcedure</b>	Read-only Hidden on back page Auto-calc programming Sets Mandatory and ROStatus
<b>RecordEntryAndExit</b>	Read-only Hidden on back page On-record-entry and -exit programming Sets Mandatory and ROStatus
<b>Mandatory</b>	Mandatory field Visible on front page Use navigation to skip over it Exit programming may display message Exit programming may go to proper field
<b>ROStatus</b>	Read only Visible on front page

John Dow is the creator of a host of useful utilities for Q&A, [jtd@pgh.net](mailto:jtd@pgh.net), [www.johndow.com](http://www.johndow.com). The download file for Online Edition subscribers contains the database featured in this article in Q&A 4.0 and 5.0 formats.



# Create an Outlook 98 Email Address Book from your Contacts Database

**M**ICROSOFT Outlook and Eudora Pro are today's most popular email programs. In the April 1998 issue, we showed how to create an address book for Eudora Pro in Q&A. Here, we'll show how to create one for Outlook 98.

The following procedure uses three database fields: **Last Name** (for sorting), **Full Name**, and **Email Address**. Your fields might differ. You might have Last Name and First Name fields but no Full Name field. In that case, you'll have to adapt the procedure for the fields you *do* have. The procedure is for Q&A DOS 4.0 and 5.0. You'll have to modify it for Q&A for Windows.

(For more on Outlook 98 email address book requirements, see Article Q179556, "How to Import and Export Text Data with Outlook," in the Microsoft Knowledge Base at microsoft.com.)

## In Q&A 4.0 or 5.0

First, get a count of the number of database records that contain an email address. Type `/=` (slash plus the equals sign—it means *not empty*) in the Email Address field at the Retrieve Spec. Press F10 to display the first record, then press Ctrl-End to display the last one. The count will appear at the bottom of the screen.

Next, follow these steps to design a reusable report that creates an Outlook-compatible email address book import file:

1. Select *Report / Design a Report* and enter the name of the database containing the email addresses.
2. Name the report something like *Email Info to Outlook* and choose *Columnar* as the report type.
3. At the Retrieve Spec, type `/=` in both the Email address and Full Name fields. Press F10.
4. At the Column/Sort Spec, type **1,AS,I** in the Last Name field, **2,I** in the Full Name field, and **3,I** in the Email address field.
5. Press F8 for Derived Columns, and fill in the first derived column exactly as shown below:

```
Heading: "Name", "E-Mail Address"
Formula: @Chr(34)+#2+@Chr(34)+", "+@Chr(34)+#3+@Chr(34)
Column/Sort Spec: 4,HS(\ )
```

6. Press F10 for the Print Options screen, then press F8 for the Define Page screen and make the following settings:

```
Page width.: 240      Page length.: 500
Left margin.: 0       Right margin.: 240
Top margin.: 0        Bottom margin.: 0
```

Adjust the *Page length* for the number of persons in the database with email addresses. (In other words, set the page length to a value that's just *greater* than the number of records you're including.)

7. Press F9 to return to the Print Options screen. Select print to *DISK*. The report should also be set to print single-spaced and justified Left.
8. Press F10 and print to a filename like QANAMES.CSV. Make a note of where you saved the file.

## In Outlook 98

Follow these steps to import the file into Outlook:

1. Select *File / New / Folder*.
2. Name the new folder something like *QANames* and select the location for it. Leave the "Folder Contains" field as is ("Contact Items").
3. Select the QANAMES folder from the Folder List.
4. Select *File / Import and Export*.
5. Select *Import from Another Program or File*.
6. Select *Comma Separated Values (DOS)* as the type of file.
7. Browse for the *QANAMES.CSV* file you created in Q&A and select it.
8. Select the destination folder (QANAMES or whatever you named it).
9. Click *Finish* to run the import.

When the import has finished, all the names and email addresses should appear in Last Name order in your new Outlook address book.

To update your email address book, rerun your *Email Info to Outlook* report, printing it to Disk to the same directory and filename. (This will save steps in Outlook.) In Outlook, start with step 3 above to import the updated Q&A file into the same address book. However, when prompted, tell Outlook to *Replace duplicates with items imported*.

## 'Live' Pick-Lists. . .continued from page 9

3. A program that executes the macro the number of times required to reach the item position determined in step 1.

The value for step 1 can be based on variable *external* or *internal* factors, or it can be a *constant*.

External factors could be the date (or current month or day or year) or time of day, whether a particular file exists (a batch file run via @Shell could determine it), or any of a host of other conditions outside of Q&A.

Internal factors could be the result of a programming calculation, the contents of a field in the record (perhaps an Initial Value), whether you're in Add or Update mode, who the user is, whether the record has been modified, or

any of a host of other conditions. A *constant* value could be a number hardcoded into the program or retrieved from a Lookup Table or external record.

You can even make it so that different users have different default selections (on the same pick-list) depending on the kinds of orders they're entering.

If you're getting lots orders for *Item X* due to some hot promo, you could temporarily make *Item X* the default pick-list selection.

The technique is useful across a wide range of list-driven data entry tasks where a specific item is more often than not the one that's needed.

Tom Marcellus is editor of *The Quick Answer*. The database featured in this article is in the April download file for Online Edition Subscribers.

---

## Record Categories. . .continued from page 3

### How the programming works

When you add a new record to the Contacts database (A1.dtf), you're prompted for the category. The XUserselect list displays options for Associate, Customer, and Vendor because those are the only records in A2.dtf.

The program in the KeyValue field executes on-record-exit. (Press F8 at the Program Spec, and type 100 in the *On exit field id* line.) When you save the record, it checks the corresponding category record in A2.dtf to see if its KeyArray field already contains the KeyValue entry from the new record. If it doesn't, then (1) the @XLookup places the entire KeyArray value in a temporary memory variable, (2) the new KeyValue entry, preceded by a semicolon, is appended to the end of it, and (3) the updated KeyArray value is XPosted back to the KeyArray field, overwriting the old value. Note that the second @XLookup and what immediately follows it is actually a parameter of the XPost command—the parameter that tells Q&A *what* to post.

The GetContact program executes if you move to the field in update mode. It displays the same three categories. The selected category then becomes a parameter in the Userselect command, which displays a list of all the names and/or companies in that category. Once GetContact contains a validated key value, the GetContact macro is triggered.

The macro copies the key value to the clipboard (F11), clears the GetContact field (Shift-F4), presses F7 for the Retrieve Spec (re-saving the record), clears the Spec with F3, moves down to the KeyValue field, pastes the key value (F12), and finally presses F10 to display the selected record.

This programming is skeletal. You can add other controls and features to it if you like.

If you delete a record from A1.dtf, good housekeeping

dictates that you should also remove it's key value from the pertinent A2.dtf record's KeyArray field. If you rarely delete records, this won't be an issue.

As I pointed out earlier, this feature isn't for huge databases containing just a few categories. It's designed for a database where the number of records in any one category doesn't exceed 1,500 or so. That said, if a category contains too many records, you might be able to break it down. For example, *Customers* could be broken down into *Customers Retail* and *Customers Wholesale* or *Customers Companies* and *Customers Persons*. In the same way, *Associates* might be broken down into *Associates Professional* and *Associates Personal*. You get the idea.

### Initial field entries

Before using the application, be sure to add the initial information to each A2.dtf record. In the Customer record, for example, type *Customer* in the **Category** field, and type - *Customers* - in the **KeyArray** field. This way, the list of customers will say - *Customers* - at the top, so there'll be no mistaking which list you're viewing.

The Category field in A2.dtf and the KeyValue field in A1.dtf must be Speedy/Unique (SU).

### Mass Updating an existing database

If you have an existing database of categories you'd like to update for use with this technique, add the GetContact or a similar field, make a list of all the categories, then add corresponding category records to the supplementary database. You can then use the KeyValue program shown earlier to Mass Update the main database. This will fill the supplementary database records with all the key values, and you'll be ready to rock.

Tom Marcellus is editor of *The Quick Answer*. The databases featured in this article are in the April download file for Online Edition subscribers.



# Keep Your Report Header When Printing to Disk

DOS

**W**HEN you print a Q&A report to a disk file, page margins, headers, footers, fonts and enhancements are all stripped out. To print a report to a disk file with its header intact, you can print it to *File* instead.

You'll need to install a Q&A FILE printer. Select Utilities / Install Printer. At the Printer Selection screen, choose an unused printer slot (A through E). In the Port column, select FILE, then choose *Basic Vanilla Non-Laser Printer* from the list of printers. This will generate output files with no printer control codes.

At the report's Define Page screen, set the Top Margin to 3 (for a three-line header) and type your header on the Header lines. If you want the header to appear just once at the top of the report, set the report's Page Length to a value greater than the number of lines the report will generate. Don't apply any enhancements to the report. (If you do, Q&A will add generic printer control codes for them to the output file.)

When printing, select your Print-to-FILE printer at the Print Options screen. Q&A will compile the report, prompt you for a filename, and save it to ASCII format with its header intact.

## How to generate pre-formatted HTML files

Q&A can print a disk report preformatted with the appropriate html (hypertext markup language) tags so you can immediately publish it to a Web site. This is how we generate *The Quick Answer's Topic Index* pages for [www.quickanswer.com](http://www.quickanswer.com).

We run a macro that prints a report to File for each alphabetical topic index range (a-b.htm, c-d.htm, and so forth) using the technique described earlier. The two header lines that define the html page for the a-b.htm report look like these:

```
<html><head><title>A-B Index</title></head><body>
<pre><strong><big>A-B Index</big></strong><small>
```

After the report is printed to File, the macro opens it in write, presses Ctrl-End to move to the end, inserts the following html page terminating codes, then resaves the file to ASCII:

```
</small></pre></body></html>
```

The report-cum-web page is now ready for upload to the Web site.



Visit *The Quick Answer's* Web Site at <http://www.quickanswer.com>

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

- Download a Free Sample Issue
- Q&A Tips and ideas
- Copy and Paste Web Site Programs
- Dozens of Free Files to Download
- Database Applications available
- Q&A Consultants Directory
- Links to Useful Web Sites & More!

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



1927A Harbor Blvd.  
Costa Mesa, CA 92627

Bulk-Rate  
U.S. Postage  
**PAID**  
Torrance CA  
Permit No. 333