

All Sorts

A sub-form displays data in a tabular form. Above each column is a button acting as a label. The user clicks the first button label, and the data is sorted in ascending order. They click again for descending order. Each button labels repeats the same behaviour.

Adrian Jones MBE, BA (Hons), June 2006

Adri@n10net.com

Ingredients

- 1 x One Level Higher or Form Open Related to 'pin' a subset of data to the form
- 2 x 'form-only' virtuals for sorting
- 1 x editbox (or button, if using DataEase 7) for each value on which to sort
- 1 x Get/setglobal (or get/setarray) custom functions to carry sort state
- 1 x ClearSelectionFilter CDF to refresh screen
- 1 x OML script per sort value to change sort state
- 1 x document properties set to display first record
- 3-D styling to imitate a button

In this recipe, we will imitate the sortable columns familiar in Windows Explorer, where you a click on the column label will sort first in ascending, then in descending order.

It demonstrates a number of techniques, including using virtual fields that only 'appear' in the data model.

Building the Sort Form

I start with the form I created in the article "[Filtering with One Level Higher](#)". This consists of a form built over Looper, with two sub-forms, one of which is Customer. Customer displays two fields: vNameComboR, which is the full customer name combined into a single string, starting with their last name, and vDOB, a virtual showing their date of birth.

	Name »	Date
1	Macchiaroli, Emma A	10/02/1959
2	MACDONALD, Louise M	17/08/1979
3	MACDONALD, Sue R	15/12/1950
4	MACIEJEWSKI, Maggie A	22/06/1960
5	MACKENZIE, Lynette T	09/05/1962
6	MACKEWICZ, Sarah X	11/12/1961
7	Mackey, Alison S	26/07/1947
8	Mackey, April Q	05/02/1947
9	Mackey, Claire O	07/12/1957
10	MACWAN, Christine W	29/10/1951
11	MADDEN, Elizabeth W	12/04/1973
12	MADDEN, Louise L	04/10/1947
13	MADERA, Pattie E	13/03/1986
14	MAGGET, Theresa M	26/02/1965
15	MAGGETTE, Katriona W	17/09/1949

Count: 436

1. Drag the Customer sub-form down a little and add to the Looper record object, a virtual called vNameSort. Make it text, 50, and derived:

```
if ( getarray ( 2 ) = blank , "Name >" ,  
if ( getarray ( 2 ) = "Date*" , "Name" , getarray ( 2 ) ) )
```

2. Add a second virtual called vDateSort. Make it text, 50, and derived:

```
if ( getarray ( 2 ) = "Name*" or getarray ( 2 ) = blank , "Date" , getarray ( 2 ) ) )
```

3. Change the visual control for both these fields to 3D, and style them so they look like buttons.
4. Add to the clicked event of vNameSort the following OML:

```
define "t" number .  
  
t := setarray ( 2 ,  
if ( getarray ( 2 ) = "Name >" or getarray ( 2 ) = blank , "Name <" , "Name >" ) ) .  
t:= clearselectionfilter ( ) .
```

5. Add to the clicked event of vDateSort the following OML:

```
define "t" number .  
  
t := setarray ( 2 ,  
if ( getarray ( 2 ) = "Date >" , "Date <" , "Date >" ) ) .  
t:= clearselectionfilter ( ) .
```

6. Now we need to add the form only virtuals that we will be sorting on. Drag vNameComboR down to expand the Customer record object, and add to this area a virtual called vSortAsc, text, 100, derived:

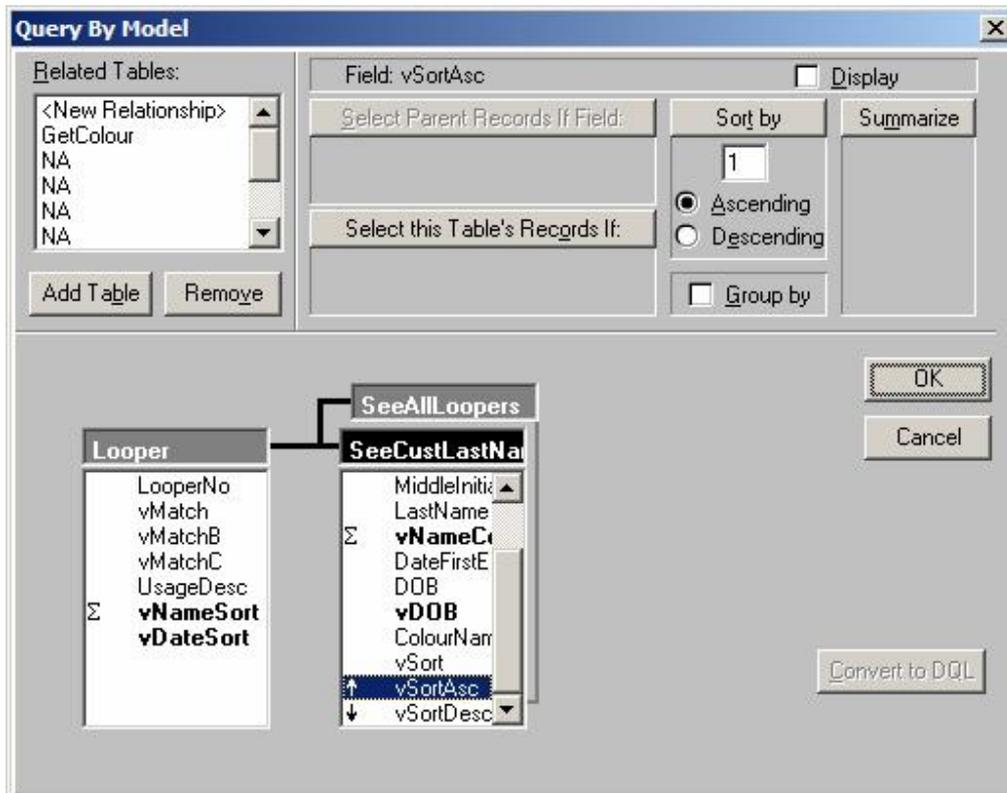
```
if ( getarray ( 2 ) = "Name >" or getarray ( 2 ) = blank , vNameComboR ,  
if ( getarray ( 2 ) = "Date >" ,  
concat ( year ( DOB ) , lastc ( concat ( "0" , month ( DOB ) ) , 2 ) , lastc ( concat ( "0" , day ( DOB ) ) , 2 ) ) , "ZZZZ" ) )
```

7. Add a second virtual called vSortDesc, text, 100, derived:

```
if ( getarray ( 2 ) = "Name <" , vNameComboR ,  
concat ( year ( DOB ) , lastc ( concat ( "0" , month ( DOB ) ) , 2 ) , lastc ( concat ( "0" , day ( DOB ) ) , 2 ) ) , "ZZZZ" )
```

The greater than arrow '>' means that the value appears in ascending order, and its reversed twin to mean we are sorting in descending order. If you choose other symbols, make sure that this is reflected in the derivation.

8. Open the Query By Model dialog, move to the SeeCustLastName box, and scroll down to see these two virtuals at the bottom of the list. Click on vSortAsc, then click on the Sort By button. Make sure its sort order is 1, and that the default Ascending is checked.



9. Move to vSortDesc, click on the Sort By button, make sure that the order is 2, and change the order to Descending.
10. Now highlight both vSortAsc and vSortDesc, and uncheck the display option at the top right of the dialog (or hit the spacebar).
11. Click OK. Accept the default of 'Modify' on the New layout dialog.

Discussion

It is easy enough to sort records in a sub-form. Simply open the QBM dialog, find the sub-form, select the field to sort on, click 'sort by', and then 'ascending'.

The problem is that this is a permanent setting; you cannot change it dynamically. We can neither swap the column on which we are sorting, nor the direction.

With my solution, we do both.

Clicking on a label button 'alerts' our form to the fact that the user wants to sort on a given field. The sort state (which column, which direction), is stored in memory using the setarray function (I'll refer to this as simply the array value).

Two virtual fields, vSortAsc and vSortDesc, are derived from the array value. Depending on the sort order, the value in the first field is either the value from the appropriate column, or a constant (in my example, a string of Z's). vSortDesc is simply derives the value of the selected column.

In the QBM, the sort order is set on these two fields, first in ascending order on vSortAsc, then in descending order on vSortDesc.

When the sort order is ascending, vSortAsc will contain the value of the desired column, and the data will simply sort in ascending order. The second sort in descending order does not affect the order on screen.

When the sort order is descending, vSortAsc will contain the same value for each record. The sorting on vSortDesc will now be effective, and so the records will appear in reverse order.

Here's a quick illustration of this:

Sort in Ascending Order

vSortAsc	vSortDesc
Adrian	Adrian
Barry	Barry
Charles	Charles

Sort in Descending Order

vSortAsc	vSortDesc
<i>ZZZZZ</i>	Charles
<i>ZZZZZ</i>	Barry
<i>ZZZZZ</i>	Adrian

Since the virtuals are text fields, we need to work a bit harder with date values, as they will otherwise sort alphabetically on the text values of the date ('02/02/2005' will appear before '01/01/2004'). To get round this, I reverse the dates so that they are formatted YYYYMMDD. DataEase has no easy functions to help with this, so that part of the derivation looks a bit involved.

Toggling

A key concept is that of toggling values, where a value is switched between two or more states. If this were a light switch, pseudo-code for toggling the switch would be:

```
Lightswitch := if ( Lightswitch = "off" , "on" , "off" ) .
```

The light has two states – on and off – and this code toggles between them.

(Actually, it has three: 'on', 'off' and 'blank'. If the state is other than 'on' or 'off' – for example, it hasn't yet been given a state – then the default with this code will be to set it to 'off'. Hence the second condition in vNameSort's OML.)

Variations of this concept are seen in the derivations of all the virtuals, and in the OML behind the column headings.

In Version 7 we can change the text on a button. This also means that we can navigate to the button via the keyboard. For the moment, I'm using a button-styled field.

Clearselectionfilter not only removes any user-entered selection criteria, it also refreshes the display, returning to the first main record in the set. My main record is simply the first one in the Looper table.

And that, more or less, is it. Run the form in user view. While #2 is blank, you should see that the Name button has the forward arrow after it, and that the names are in alphabetical order. Click the name button, and its label changes to show the backward arrow, with the names in reverse order.

Click date, and the forward arrow now appears after the word 'Date'; the records are in chronological date order, with the oldest first. Click the date button again, and the records are in reverse date order.

DataEase will struggle if you try to sort all 50 million records in your table, since the sort is on a virtual. But then again, Explorer will take its time if you try to do the same with a ridiculous number of files. You should make sure that you have filtered the set to a manageable amount first. Even so, you can probably easily get away with a couple of thousand at a time!

If you need to change the derivation in vSortAsc or vSortDesc, go to the Query By Model dialog, find the fields, highlight them and click 'display' (or hit the spacebar), okay this and the modify layout option dialog, and they will reappear in the sub-form. Make your changes, then go back to the Query By Model dialog to remove them from the display.

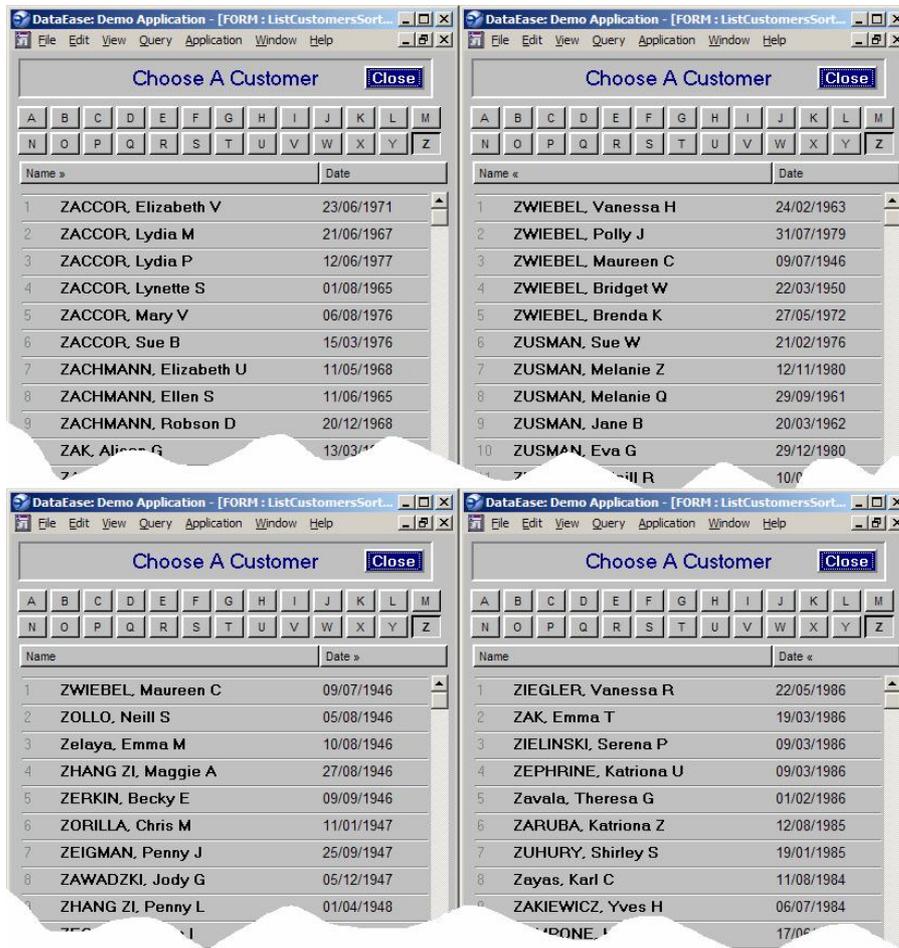


Figure 1: Top: the same list sorted first in ascending Name order, then descending by Name. Below: in date of birth in order, then in reverse order. Note the little arrow after the button label to indicate which field is being used to sort, and in which direction.

-
- *If you have any comments on this article, or suggestions for future material, please e-mail me at adri@n10net.com, visit www.n10net.com or call +44(0)7793 148660.*
 - *I offer consultancy, technical support and mentoring services and DataEase-related products, using DataEase for Windows and ASP.NET with SQL Server. I am based in London, UK.*
-

About The Author: Adrian Jones is the former editor of 'Dialogue, the DataEase magazine', the creator of the 'template' applications that accompany DataEase 6, and a freelance DataEase consultant. A leading thinker in the use of the Windows product, he started with DataEase DOS 2.53 to solve the needs of his publishing business in the 1980s. His consultancy skills have been used by manufacturing, banking and charity organizations, and most recently by the Office Of Chief Medical Examiner in New York as part of the post 9/11 recovery. In July 2004, he received an MBE for 'services to British and local families in New York'.