

## Re: Help needed on creating a subform

---

*Source:* <http://www.tech-archive.net/Archive/Access/microsoft.public.access.forms/2007-12/msg00493.html>

---

- *From:* Rod Plastow <[RodPlastow@xxxxxxxxxxxxxxxxxxxxxxxxxxxxxx](mailto:RodPlastow@xxxxxxxxxxxxxxxxxxxxxxxxxxxxxx)>
  - *Date:* Thu, 13 Dec 2007 00:29:09 -0800
- 

Hi Craig,

It isn't quite as simple as I thought – it never is, is it? This solution is a little involved and I would not recommend it if there are hundreds of keywords, probably not otherwise you would need keywords for keywords.

First the concept. Dispense with a subform for maintaining keywords as this is a little 'clunky' from the aspects of look and feel. Instead place a list box on your main form. This list box is 'selectable,' that is the user can select zero or more rows from the list at one time. Access offers two options for selection: simple and extended. I prefer 'simple' for this application as you select a row with a click of the mouse (or spacebar) and deselect the same way. Then I thought wouldn't it be nice to sort all the selected rows to the top of the list rather than have them scattered throughout the list and possibly not all visible due to scrolling.

So that's the concept. I know of no automated way that Access will do this for you so you must program all the features and functionality yourself. The rest of this post describes how to do this. As far as possible I have used names that will match yours but you must review the code and substitute your own names where applicable.

The first task is to implement a list box on your main form. My list box is called List7, yours will have a different name. In the properties box for the list box select the 'Other' tab and change the 'Selectable' entry from none to simple. Select the 'Format' tab and specify 3 columns with column widths of 0; 1 in;0 (this hides the first and third columns). If you don't think 1 in is wide enough for your keywords by all means make it wider. Check under the 'Data' tab that the bound column is 1 and the Row Source Type is Table/Query. You can alter the other properties to suit your design style.

Next you need to add some sub functions to the VBA code behind your form. The most convenient way to open the VBA coding window is to select your main form in design view, select the 'Event' tab in the properties box, double click in the 'On Current' row to display [Event Procedure] and then click on the elipsis button (...) at the end of the row. This creates an empty On Current procedure for the form.

Don't enter anything in this procedure yet but scroll down below the End Sub

## Re: Help needed on creating a subform

statement and enter the following. Don't worry this is by far the worst of them all. Watch out as this board has wrapped some of the lines.

```
Private Sub PopulateListBox()
```

```
Dim strSQL As String
```

```
strSQL = "SELECT tblKeyword.Id, tblKeyword.Keyword,  
tblKeywordLink.QuoteId " & _  
"FROM tblKeyword LEFT JOIN tblKeywordLink ON tblKeyword.Id =  
tblKeywordLink.KeywordId " & _  
"WHERE tblKeywordLink.QuoteId = " & Me.Id & " " & _  
"UNION " & _  
"SELECT tblKeyword.Id, tblKeyword.Keyword, 0 " & _  
"FROM tblKeyword " & _  
"WHERE tblKeyword.Id Not In (SELECT tblKeyword.Id " & _  
"FROM tblKeyword LEFT JOIN tblKeywordLink ON tblKeyword.Id =  
tblKeywordLink.KeywordId " & _  
"WHERE tblKeywordLink.QuoteId = " & Me.Id & " )" & _  
"ORDER BY QuoteId DESC, Keyword"
```

```
Me.List7.RowSource = strSQL
```

```
End Sub
```

What this procedure does is to construct a SQL string that is a UNION query based on the current value of the quote id. You can see that '& Me.Id &' occurs twice, once in each WHERE clause. I have called the field that contains the quote id, 'Id' for brevity. 'Me' is VBA shorthand for the current VBA object which, when the code executes is the main form. So this SQL string needs to be reconstructed every time you move to a new quote record; thus it will eventually be invoked from the On Current event of the form.

If you are not familiar with the '& \_' symbols they are simply the string concatenation symbol (&) and the new line continuation symbol (\_). It makes it much easier to read if you break up long strings across several coding lines.

A UNION query concatenates the results of two separate queries. In the above the first query (before the UNION keyword) retrieves existing keyword links for the current quote. It selects, in this sequence, the keyword id and the keyword text. I have also added the quote id to the selection purely for ordering the list box entries. The second query goes to the keyword table and retrieves all the keywords that are not in the result set of the first query. As there is no quote id in these cases the SQL includes the number zero in the equivalent position. The ORDER BY clause first sorts the active links to the top of the list – the assumption is that quote id is greater than zero – and then sorts alphanumerically within these divisions.

[A note here for any other posters: I played long and hard trying to get

Re: Help needed on creating a subform

## Re: Help needed on creating a subform

this to work in the Row Source property but Access failed to recognise Me.Id in whatever guise I tried. I also tried to avoid a UNION query but failed. Any suggestions?]

Finally there is a statement in the procedure that assigns the SQL to the row source of the list box. This assignment will trigger a requery of the list.

Now you need to enter another procedure.

```
Private Sub SynchroniseListBox()
```

```
Dim i As Long
```

```
For i = 0 To Me.List7.ListCount - 1
```

```
If Me.List7.Column(2, i) > 0 Then
```

```
Me.List7.Selected(i) = True
```

```
Else
```

```
Me.List7.Selected(i) = False
```

```
End If
```

```
Next
```

```
End Sub
```

A couple of explanations here: columns and rows are referenced relative to zero thus it is necessary to subtract 1 from the list count as the terminating condition for the loop. Column(2,i) is the value of quote id (from the first query) or zero (from the second query). If this value is greater than zero indicating an active keyword link then we must switch on the list box's selected property for that row.

Now back to that so far empty On Current procedure.

```
Private Sub Form_Current()
```

```
PopulateListBox
```

```
SynchroniseListBox
```

```
End Sub
```

Half the job is now done and I would strongly recommend that you test what you have done. You need to construct some keyword links by other methods. My next post will continue with how to keep your tables up-to-date.

Regards,

Rod

.