

SelectListboxOnFirstLetters

(Q) Can I make a search box where the proposed choices are dynamically limited to those starting with the letters typed by the user ?

tags:
[HowTo](#)

(R) There are several solutions. One of them is presented below.

The implemented module is triggered each time a character is entered.

Accepted characters are :

- Letters, either lower or upper cased, and the space character
- The BACKSPACE which cancels the last entered letter
- The ESCAPE key which cancels the complete string of characters.

Other keys are ignored.

The goal is to minimize the reload of the data from the database, as it is an operation that could take a long time while the user is keying letters in ...

Solution

It consists in:

- A **listbox**. It is reinitialized with database data at each *form loading*, and, while the focus is on the box, at each BACKSPACE key and at each ESCAPE key pressing.
- After each letter pressed by the user the box keeps only the subset of the initial data that corresponds with the entered characters, without database access. The control is temporarily converted from an SQL **RowSourceType** to a VALUelist one.
- The succession of letters is kept into the **Tag** property of the *listbox*.

Let's assume next table:

- Products table

Fields	Field Type	Primary
CategoryID	BigInt	
ProductName	Text	
QuantityPerUnit	Text	
ReorderLevel	Integer	
SupplierID	BigInt	
UnitPrice	Number	
UnitsInStock	Integer	
UnitsOnOrder	Integer	
Discontinued	Boolean	
ProductID	BigInt	Y
Picture	Image	

The *listbox* will help to converge as quickly as possible to the searched product name.

1. Associate next code with the *When loading* form event:

```
Sub InitList(Optional poEvent As Object) ' Optional because also called from Capt
'Activated when form opened and when ESC key is pressed in listbox

Const cstForm = "Products_FastSearch"
Const cstList = "FastList"
Dim ocList As Object, sSource As String

    Set ocList = Forms(cstForm).Controls(cstList)
    If Not IsMissing(poEvent) Then ocList.Tag = "" ' Initialized only when

    ocList.RowSourceType = com.sun.star.form.ListSourceType.SQL
    ocList.RowSource = "SELECT [ProductName] FROM [Products] ORDER BY [ProductName] ASC"

    sSource = Join(ocList.ItemData, ";")
    ocList.RowSourceType = com.sun.star.form.ListSourceType.VALUELIST ' For pe
    ocList.RowSource = sSource

End Sub
```

2. Associate next routine with the *Key pressed* event of the concerned *listbox* control :

```
Sub CaptureChar(poEvent As Object)

Dim oEvent As Object, ocList As Object, sTag As String, i As Integer
Dim vList() As Variant, vNewList() As Variant, iNew As Integer
    Set oEvent = Events(poEvent)
    If oEvent.EventType <> "KEYEVENT" Then Exit Sub

'Accepted keys: A-Z, a-z, BACKSPACE, ESCAPE. All other keys ignored
'Tag property of list box is used to keep sequence of entered characters
    With oEvent
        Set ocList = oEvent.Source
        Select Case True
            Case .KeyAlt, .KeyCtrl : Exit Sub
            Case .KeyCode = com.sun.star.awt.Key.ESCAPE
                ocList.Tag = ""
                Call InitList()
                Exit Sub
            Case .KeyCode = com.sun.star.awt.Key.BACKSPACE
                If Len(ocList.Tag) > 0 Then
                    sTag = ocList.Tag
                    ocList.Tag = Left(sTag, Len(sTag) - 1) ' Reduce
                    Call InitList()
                End If
            Case (UCase(.KeyChar) >= "A" And UCase(.KeyChar) <= "Z") Or (.KeyCode
                ocList.Tag = ocList.Tag & .KeyChar
            Case Else : Exit Sub
        End Select
    End With

    With ocList ' Process Tag
        sTag = .Tag
        vList() = Split(.RowSource, ";")
        .RowSource = ""
        vNewList() = Array() ' Otherwise Redim protests
        If UBound(vList) >= 0 Then
            ReDim vNewList(0 To UBound(vList)) ' Resized later
            iNew = 0 ' Counts valid entries in vList
        End If
    End With

End Sub
```

```

        For i = 0 To UBound(vList)
            If Len(vList(i)) >= Len(sTag) Then
                If UCase(Left(vList(i), Len(sTag))) = UCase(sTag) Then
                    vNewList(iNew) = vList(i)
                    iNew = iNew + 1
                End If
            End If
        Next i
        If iNew > 0 Then
            ReDim Preserve vNewList(0 To iNew - 1)
            .RowSource = Join(vNewlist, ";")           'Apply new list after
            .ListIndex = 0                             'Select first element
        End If
    End If
End With

End Sub

```

See also

Event

ItemData

ListIndex

RowSource

RowSourceType

Tag

Refer to ...

Basic module	Form	Form event	Control	Control event	Comments
FastSearch	Products_FastSearch	When loading	FastList	Key pressed	Select the listbox, enter characters => the listbox content changes.

Bookmark this page » » [SelectListboxOnFirstLetters](#)