

v0.0 - clsSearchableDropdown_20230319_1734.cls

VERSION 1.0 CLASS

BEGIN

MultiUse = -1 'True

END

Attribute VB_Name = "clsSearchableDropdown"

Attribute VB_GlobalNameSpace = False

Attribute VB_Creatable = False

Attribute VB_PredeclaredId = False

Attribute VB_Exposed = False

Option Explicit

'clsSearchableDropdown - 2023-03-01

,

'Author: Paul Kelly - <https://ExcelMacroMastery.com/>

'Version: 2.1 Updated 30-01-2023

'YouTube video: https://youtu.be/gkLB-xu_JTU

'Description: Searchable Dropdown list

,

The code in this class allows the user to

,

create a searchable dropdown list of items.

,

'HOW TO USE - See READ ME File - 2023-03-01

,

'LICENSE ETC.

' You can use this code free in any of your software applications and distribute as you wish.

' Please do not alter or remove the authors name or website.

Private Const m_conMaxRows As Long = 6

'Configurable Settings

Private m_compareMethod As VbCompareMethod 'Determines case sensitivity in the search

Private m_listOfItems As Variant 'This is the array of items that is filtered

Private m_maxRows As Long 'The number of rows to be displayed in the listbox.

Private m_startText As String 'Start text in the textbox

Private m_showAllMatches As Boolean 'True: shows all matches. False show the number of rows specified by m_MaxRows

'This is used to prevent events running when changes are being made

Private m_UpdateControl As Boolean

Private m_textboxStartingState As Boolean 'Used to decide when to remove the starting text

v0.0 - clsSearchableDropdown_20230319_1734.cls

```
Private m_textboxStartingState As Boolean    'Used to decide when
to remove the starting text
```

```
Private WithEvents myListBox As MSForms.listbox
Attribute myListBox.VB_VarHelpID = -1
Private WithEvents myTextBox As MSForms.TextBox
Attribute myTextBox.VB_VarHelpID = -1
```

```
'-----
---- PROPERTIES
```

```
Public Property Let CompareMethod(ByVal value As VbCompareMethod)
```

```
    m_compareMethod = value
    Call FilterListBox
```

```
End Property
```

```
Public Property Get SelectedItem() As String
```

```
    SelectedItem = IIf(m_textboxStartingState = True, "",
myTextBox.value)
```

```
End Property
```

```
Public Property Let List(ByVal value As Variant)
```

```
    m_listOfItems = value
```

```
End Property
```

```
'The number of rows that will be visible in the listbox
```

```
Public Property Let MaxRows(ByVal value As Long)
```

```
    m_maxRows = value
    Call FilterListBox
```

```
End Property
```

```
'Set the text to be displayed in the textbox before the search
```

```
Public Property Let StartText(ByVal text As String)
```

```
    m_startText = text
    SetTextboxValue (m_startText)
```

```
End Property
```

```
'If true include all matches in the listbox. If false only show
```

End Property

'If true include all matches in the listbox. If false only show the rows

'specified by m_MaxRows

Public Property Let ShowAllMatches(ByVal state As Boolean)

 m_showAllMatches = state

 Call FilterListBox

End Property

Public Property Set SearchListBox(ByVal oListBox As MSForms.
listbox)

 Set myListBox = oListBox

 Call InitializeListBox

End Property

Public Property Set SearchTextBox(ByVal oTextBox As MSForms.
TextBox)

 Set myTextBox = oTextBox

 Call InitializeTextBox

End Property

'-----
-- CLASS EVENTS

'Description: Set the defaults

Private Sub Class_Initialize()

 m_compareMethod = vbTextCompare

 m_maxRows = m_conMaxRows

 m_startText = "Saisir le nom de la compagnie que vous
recherchez"

 m_showAllMatches = True 'False - 2023-03-01 - RMV

End Sub

'-----
LISTBOX EVENTS

Private Sub myListBox_DblClick(ByVal Cancel As MSForms.
ReturnBoolean)

```
Private Sub myListBox_DblClick(ByVal Cancel As MSForms.  
ReturnBoolean)
```

```
    'If the user clicks or presses enter then  
    'place the selected value in the textbox  
    If m_UpdateControl = False Then  
        SetTextboxValue myListBox.value  
        Call ShowListbox(False)  
    End If
```

```
End Sub
```

```
Private Sub myListBox_KeyDown(ByVal KeyCode As MSForms.  
ReturnInteger, ByVal Shift As Integer)
```

```
    'When the key is down in the listbox turn on  
    'm_UpdateControl to prevent the click event occurring  
    If KeyCode = vbKeyDown Then  
        m_UpdateControl = True  
    ElseIf KeyCode = vbKeyUp Then  
        m_UpdateControl = True  
        CheckListBoxFirstItem  
    ElseIf KeyCode = vbKeyReturn Then  
        'swallow the enter keycode as it passes on to the ok  
button  
        KeyCode = 0  
        SetTextboxValue myListBox.value  
        Call ShowListbox(False)  
    End If
```

```
End Sub
```

```
Private Sub myListBox_KeyUp(ByVal KeyCode As MSForms.  
ReturnInteger, ByVal Shift As Integer)
```

```
    'Turn update control off - turned on in KeyDown  
    If KeyCode = vbKeyDown Then  
        m_UpdateControl = False  
    ElseIf KeyCode = vbKeyUp Then  
        m_UpdateControl = False  
    End If
```

```
End Sub
```

```
Private Sub List_MoveDown()
```

```
    m_UpdateControl = True  
    If m_textboxStartingState = False Then
```

v0.0 - clsSearchableDropdown_20230319_1734.cls

```
Private Sub List_MoveDown()
```

```
    m_UpdateControl = True
    If m_textboxStartingState = False Then
        With myListBox
            .SetFocus
            If .ListIndex < .ListCount - 1 Then
                .ListIndex = .ListIndex + 1
                .Selected(.ListIndex) = True
            End If
        End With
    End If
    m_UpdateControl = False
```

```
End Sub
```

```
'-----
-- LISTBOX SUBS
```

```
Private Sub UpdateListbox(items As Variant)
```

```
    With myListBox

        'Reload listbox
        .Clear
        .ForeColor = rgbBlack

        'Set the listbox size
        If IsEmpty(items) Then
            'No items found
            .List = Array("Je n'ai rien trouvé!")
            .ForeColor = rgbRed
        Else
            'items found
            .List = items
            .ListIndex = 0
        End If

        'Resize the listbox
        Call SetListboxPosition

        'If show all matches then have a scrollbar
        If m_showAllMatches = True Then
            Call MakeAllMatchesAvailable
        Else
            .Height = ResizeListbox(myListBox, myTextBox.Font.
Size)
        End If
    End With
End Sub
```

```

v0.0 - clsSearchableDropdown_20230319_1734.cls
        .Height = ResizeListbox(myListBox, myTextBox.Font.
Size)
    End If

    End With

End Sub

Private Sub MakeAllMatchesAvailable()

    With myListBox

        'To get the scrollbar working correctly it is necessary
to
        'turn IntegralHeight off and on
        .IntegralHeight = False
        .Height = ResizeListbox(myListBox, myTextBox.Font.Size)
        .IntegralHeight = True

        'List index will not highlight to first unless the
second one is
        'highlighted first. It might be to do with the resizing
from the
        'Integral height
        If .ListCount > 1 Then .ListIndex = 1
        .ListIndex = 0

    End With

End Sub

Private Sub InitializeListBox()

    'Remove any automatic resizing of the listbox
myListBox.IntegralHeight = False

End Sub

Public Sub CheckListBoxFirstItem()

    If myListBox.ListIndex = 0 Then
        m_UpdateControl = False
        SelectTextBox
    End If

End Sub

Private Function ShowListbox(Optional ByVal show As Boolean =

```

End Sub

Private Function ShowListbox(Optional ByVal show As Boolean = True)

 myListBox.Visible = show

End Function

Private Function SetListboxPosition()

 With myListBox

 .Left = myTextBox.Left

 .Top = myTextBox.Top + (myTextBox.Height)

 .Width = myTextBox.Width

 Call ShowListbox(True)

 End With

End Function

Private Function ResizeListbox(myListBox As MSForms.listbox, fontSize As Double) As Double

 'Set listbox font to the same size as the textbox

 myListBox.Font.Size = fontSize

 Dim itemCount As Long

 itemCount = IIf(myListBox.ListCount > m_maxRows, m_maxRows, myListBox.ListCount)

 Dim itemSize As Double

 'the font size is itself plus a quarter for the space between rows

 itemSize = myListBox.Font.Size + (myListBox.Font.Size / 4)

 'Font 10 has different sizing

 Dim extraspace As Double

 If fontSize = 10 Then

 extraspace = 4

 Else

 'If 2 or less items then the listbox needs to be taller

 If myListBox.ListCount <= 2 Then

 extraspace = 3

 Else

 extraspace = 2

 End If

 End If

v0.0 - clsSearchableDropdown_20230319_1734.cls

```
        Else
            extraspace = 2
        End If
    End If

    ResizeListbox = (itemSize * ItemCount) + extraspace

End Function

'-----
TEXTBOX EVENTS

Private Sub myTextBox_Change()

    If m_UpdateControl = False Then
        If Trim(myTextBox.value) = "" Then
            Call InitializeTextBox
        Else
            If m_textboxStartingState = True Then
                m_textboxStartingState = False
                Call RemoveStartingText
            End If
            Call FilterListBox
        End If
    End If

End Sub

Private Sub myTextBox_KeyDown(ByVal KeyCode As MSForms.
ReturnInteger, ByVal Shift As Integer)

    'If the user presses keyup/down or enter while in the textbox
    If KeyCode = vbKeyDown Then
        List_MoveDown
    ElseIf KeyCode = vbKeyReturn Then
        'Swallow the enter keycode as it passes on to the ok
button
        KeyCode = 0
        If IsNull(myListBox.value) = False And
m_textboxStartingState = False Then
            SetTextboxValue myListBox.value
        End If
        Call ShowListbox(False)
    ElseIf KeyCode = vbKeyEscape Then
        'swallow the esc keycode
        Call InitializeTextBox
        KeyCode = 0
    End If

End If
```


v0.0 - clsSearchableDropdown_20230319_1734.cls

```
    Call InitializeTextBox
    KeyCode = 0
End If

End Sub

'-----
-- TEXTBOX SUBS

Private Sub SetTextboxValue(ByVal text As String)

    If m_UpdateControl = False Then

        With myListBox
            m_UpdateControl = True
            'Set the listbox selected value to the textbox
            'and hide the listbox
            myTextBox.value = text

            Call SelectTextBox

            m_UpdateControl = False

        End With

    End If

End Sub

Private Sub InitializeTextBox()

    'Set the starting text and position
    m_textboxStartingState = True
    SetTextboxValue m_startText
    myTextBox.SelStart = 0
    myTextBox.ForeColor = rgbBlue

    Call ShowListbox(False)

End Sub

'Description: Replace the default starting text with the letter
the user has typed.
'           This will work if the user types at any position
in the starting text.
Private Sub RemoveStartingText()

    m_UpdateControl = True
```

v0.0 - clsSearchableDropdown_20230319_1734.cls

```
Private Sub RemoveStartingText()
```

```
    m_UpdateControl = True
```

```
    With myTextBox
```

```
        .text = Mid(.value, .SelStart, 1)
```

```
        .ForeColor = rgbBlack
```

```
    End With
```

```
    m_UpdateControl = False
```

```
End Sub
```

```
Private Function SelectTextBox()
```

```
    With myTextBox
```

```
        '.SelStart = 0
```

```
        .SetFocus
```

```
    End With
```

```
End Function
```

```
'-----  
-- GENERAL SUBS
```

```
'Filters the Listbox
```

```
Public Sub FilterListBox()
```

```
    'In case a setting like MaxRows is used before the range is  
set
```

```
    If IsEmpty(m_listOfItems) Then Exit Sub
```

```
    m_UpdateControl = True
```

```
    Dim items As Variant
```

```
    items = FilterData
```

```
    Call UpdateListbox(items)
```

```
    m_UpdateControl = False
```

```
End Sub
```

```
Private Function FilterData() As Variant
```

```
    Dim textPattern As String
```

```
    textPattern = myTextBox.value
```

```
    'Create an array to store the filtered items
```

v0.0 - clsSearchableDropdown_20230319_1734.cls

```
Dim textPattern As String
textPattern = myTextBox.value

'Create an array to store the filtered items
Dim filteredItems() As String
ReDim filteredItems(0 To m_maxRows - 1)

'Read through all the items in the full list
Dim i As Long
Dim count As Long: count = 0
For i = LBound(m_listOfItems) To UBound(m_listOfItems)
    'Using Instr instead of Like so we can set the case
sensitivity
    If Instr(1, m_listOfItems(i, 1), textPattern,
m_compareMethod) > 0 Then
        filteredItems(count) = m_listOfItems(i, 1)
        count = count + 1
        If m_showAllMatches = False Then
            'Only show the max number of rows
            If count >= m_maxRows Then Exit For
        End If
    End If
Next

'Use variant so we can check later if the array is empty
Dim finalItems As Variant
If count > 0 Then
    ReDim finalItems(0 To count - 1)
    For i = 0 To count - 1
        finalItems(i) = filteredItems(i)
    Next i
End If

FilterData = finalItems

End Function
```