

Rem SelPartyDb - Tracks information about organizational constituents using _
concept of "party" and "SelCtl" class objects _
Copyright (C) 2004 Larry Francis Bednar

Rem This library is free software; you can redistribute it and/or _
modify it under the terms of the GNU Lesser General Public _
License as published by the Free Software Foundation; either _
version 2.1 of the License, or (at your option) any later version.

Rem This library is distributed in the hope that it will be useful, _
but WITHOUT ANY WARRANTY; without even the implied warranty of _
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU _
Lesser General Public License for more details.

Rem You should have received a copy of the GNU Lesser General Public _
License along with this library; if not, write to the Free Software _
Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Rem The author may be contacted at: _
3226 NE 24th Ave, Portland, OR 97212-2442 USA _
larry@quimdas.com _
www.quimdas.com

Option Compare Database

Option Base 1

Option Explicit

Rem Used to construct filter statements corresponding to values indicated _
by the user employing a mult-select list box. The object assumes that _
the selected values are a part of the base SELECT query and does not _
include criteria for joining tables as a part of its function

Rem Private module-level variables used for properties

Private mInUse As Long	'Indicates "in/not in" use of selections
Private mInListDescCol As Long	'Index of list box column containing values _ to be displayed in human-language desc of _ selection clause
Private mInValType As Long	'Indicates nbr/char type of target column
Private mIsList As Boolean	'Multi-select list box used on form
Private mSelClause As String	'Filter expression reflecting user choices
Private mSelDesc As String	'English version of filter expression for _ display on reports, etc.
Private mTrgtCol As String	'Group ID col name in target table
Private mTrgtTableAlias As String	'Name of target table

Rem Private module-level variables used for internal purposes

Private mPrevSel() As Variant	'Stores a previous set of selections from _ the associated multi-select list box
Private mReqsOK As Boolean	'Indicates whether base requirements are OK
Private mStoredSelCount As Long	'Count of stored list box selections
Private mStoredSelClause As String	'Stores previous value of SelClause
Private mStoredSelDesc As String	'Stores previous value of SelDesc

Rem Define enumeration of constants to indicate use of date value

```
Public Enum sclValType
    sclChar = 1           'Target col is character type
    sclDate = 2          'Target col is date type
    sclNbr = 3           'Target col is numeric
End Enum
```

```
Public Enum sclInUse
```

```

    sclIn = 6                'Select records that indicate membership
    sclNotIn = 7            'Select records that are not members
End Enum

Public Function CtlAssoc(ByVal ctl As Control) As Boolean
    Rem This function returns a boolean indicating whether the specified _
        control is a data-input control associated with this SelCtl object

    On Error GoTo Err_CtlAssoc

    Rem No two controls on the same form can have the same name
    If ctl.name = mlst.name Then
        CtlAssoc = True
    Else
        CtlAssoc = False
    End If

Exit_CtlAssoc:
    Exit Function

Err_CtlAssoc:
    MsgBox "SelCtlList.CtlAssoc Error Nbr " & Err.Number & " - " & _
        Err.Description
    Resume Exit_CtlAssoc

End Function

Private Function Delimiter() As String
    Rem This function returns the delimiter appropriate to the ValType _
        property value

    Dim lng As Long

    On Error GoTo Err_Delimiter

    lng = mlngValType

    Dim str As String

    Select Case lng
        Case sclChar
            str = "'"
        Case sclDate
            str = "#"
        Case sclNbr
            str = ""
        Case Else
            str = Null
    End Select

    Delimiter = str

Exit_Delimiter:
    Exit Function

Err_Delimiter:
    MsgBox "SelCtlTxt.Delimiter Error Nbr " & Err.Number & " - " & _
        Err.Description
    Resume Exit_Delimiter

```

```

End Function

Private Function InExpr() As String
    Rem Construct SQL comparison expressionf or group ID selection using _
        SQL "IN" comparison operator

    On Error GoTo Err_InExpr

    Dim str As String
    Dim strDelim As String
    Dim strOperator As String
    Dim strSelectedId As String
    Dim varRow As Variant

    str = ""

    strDelim = Delimiter()
    strOperator = Operator(mlngInUse)

    With mlst
        Rem Alter handling of listbox depending on multiselect setting
        If mlst.MultiSelect = 0 Then
            strSelectedId = strDelim & mlst.value & strDelim
        Else
            Rem Run OneIdTestClause for each selected row of the list box. Column 0 _
                is assumed to have the value to be tested
            For Each varRow In .ItemsSelected

                'ID value is assumed to be in column zero
                strSelectedId = strDelim & .Column(0, varRow) & strDelim

                Call basSelCtlShared.AddToCSVList(str, strSelectedId)

            Next varRow
        End If
    End With

    Rem If parsing of list box selections created a string, add column _
        specification, comparison operator and parentheses
    If Len(str) > 0 Then
        str = mstrTrgtTableAlias & "." & mstrTrgtCol & strOperator & "(" & str & ")"
    End If

    Rem Set function return value
    InExpr = str

Exit_InExpr:
    Exit Function

Err_InExpr:
    MsgBox "SelCtlList.InExpr Error Nbr " & Err.Number & " - " & Err.Description
    Resume Exit_InExpr

End Function

Private Function Operator(ByVal lng As sclInUse) As String
    Rem This function sets the operator corresponding to the TrgtValComparison _
        property value

    On Error GoTo Err_Operator

```

```

Dim str As String

Rem Set group selection comparison operator
Select Case lng
    Case sclNotIn
        str = " NOT IN "
    Case sclIn
        str = " IN "
    Case Else
        str = ""
End Select

Operator = str

Exit_Operator:
Exit Function

Err_Operator:
MsgBox "SelCtlList.Operator Error Nbr " & Err.Number & " - " & _
    Err.Description
Resume Exit_Operator

End Function

Private Sub CheckReqs()
Rem This procedure sets the value of the module-level boolean mbReqsOK _
to "true" if all properties required to calculate _
filter values are properly specified. Returns "false" and displays a _
window listing properties that still require values otherwise.

On Error GoTo Err_CheckReqs

Dim bOK As Boolean
Dim strUnspecProps As String

bOK = True
strUnspecProps = ""

If mlngInUse <> sclIn And mlngInUse <> sclNotIn Then
    bOK = False
    Call basSelCtlShared.AddToCSVList(strUnspecProps, "MbrUse")
End If

Select Case mlngValType
    Case sclChar, sclDate, sclNbr
    Case Else
        bOK = False
        Call basSelCtlShared.AddToCSVList(strUnspecProps, "ValType")
End Select

If Not IsObject(mlst) Then
    bOK = False
    Call basSelCtlShared.AddToCSVList(strUnspecProps, "LstCtl")
End If

If Len(mstrTrgtCol) = 0 Then
    bOK = False
    Call basSelCtlShared.AddToCSVList(strUnspecProps, "TrgtIdCol")
End If

```

```

If Len(mstrTrgtTableAlias) = 0 Then
    bOK = False
    Call basSelCtlShared.AddToCSVList(strUnspecProps, "TrgtTableAlias")
End If

If bOK = False Then
    MsgBox "SelCtlList properties " & strUnspecProps & " must have values
specified"
End If

    mbReqsOK = bOK

Exit_CheckReqs:
    Exit Sub

Err_CheckReqs:
    MsgBox "SelCtlList.CheckReqs Error Nbr " & Err.Number & " - " & Err.Description
    Resume Exit_CheckReqs

End Sub

Private Sub Class_Initialize()
    mlngInUse = sclIn
    mstrSelClause = ""

    ReDim mavarStoredLstSel(10)
End Sub

Public Sub Clear()
    Rem This procedure clears selection values stored in association with the _
    SelCtlList object

    Dim intI As Integer

    Rem Set stored values to "blank"
    mstrStoredSelClause = ""
    mstrStoredSelDesc = ""

    Rem Need to reset ListBox in more complicated approach.
    'mavarStoredListSel = ""
    For intI = 1 To UBound(mavarStoredLstSel)
        mavarStoredLstSel(intI) = 0
    Next
    mlngStoredLstSelCount = 0

    Rem After "StoredLst" variables are blanked, the RevertCtlVal procedure _
    may be used to revert the ListBox control on the associated form, as _
    well as the mstrSelClause and mstrSelDesc variables.
    Call RevertCtlVal

End Sub

Public Sub RevertCtlVal()
    Rem This procedure reverts the value of the associated text box control _
    to the saved value

    On Error GoTo Err_RevertCtlVal

    Rem Restore previous SelClause and SelDesc values

```

```

Call RevertSelVal

Rem Set value of list box selections to previously stored value
Call RevertLstSel

Exit_RevertCtlVal:
Exit Sub

Err_RevertCtlVal:
MsgBox "SelCtlList.RevertCtlVal Error Nbr " & Err.Number & " - " & _
      Err.Description
Resume Exit_RevertCtlVal

End Sub

Private Sub RevertLstSel()
Rem This procedure reverts the selections of the associated list box _
      control to the saved set

Dim lngI As Long
Dim lngPrevSelectedRow As Long

On Error GoTo Err_RevertLstSel

Rem Remove current selections
With mlst
  For lngI = 0 To .ListCount - 1
    .Selected(lngI) = False
  Next lngI
End With

Rem If count of stored selections is zero, do nothing - the earlier part _
      of the procedure has already reset to "no selections". If count is _
      positive, then reset stored values from array.
If mlngStoredLstSelCount > 0 Then

  Rem Set list box selections to previously stored set
  For lngI = 1 To mlngStoredLstSelCount
    lngPrevSelectedRow = mavarStoredLstSel(lngI)
    mlst.Selected(lngPrevSelectedRow) = True
  Next lngI

End If

Exit_RevertLstSel:
Exit Sub

Err_RevertLstSel:
MsgBox "SelCtlList.RevertLstSel Error Nbr " & Err.Number & " - " & _
      Err.Description
Resume Exit_RevertLstSel

End Sub

Private Sub RevertSelVal()
Rem Sets SelClause and SelDesc to previously stored values

mstrSelClause = mstrStoredSelClause
mstrSelDesc = mstrStoredSelDesc

```

End Sub

```
Public Sub SetProperties(ByVal LstCtl As ListBox, _
    ByVal lngListDescCol As Long, _
    ByVal lngInUse As sclInUse, _
    ByVal lngValType As sclValType, _
    ByVal strTrgtCol As String, _
    ByVal strTrgtTableAlias As String)
    Rem This procedure sets all required properties of the SelCtlObj. Used _
    in preference to individual property Let/Set approach to ensure that _
    all required properties are set correctly at time of object creation. _
    The only Let/Set procedures used should be for properties that are _
    altered by user update of control values on forms.

    Dim bListBox As Boolean
    Dim strListBoxName As String

    On Error GoTo Err_SetProperties

    Rem Set reference to listbox object
    bListBox = False
    strListBoxName = ""
    If Not LstCtl Is Nothing Then
        bListBox = True
        Set mlst = LstCtl
        strListBoxName = " (control '" & mlst.name & "'" )"
    Else
        MsgBox "SelCtlList.SetProperties - specified lstCtl parameter references " & _
            "'Nothing' or is not a list box control"
    End If

    Rem Check specified "description value" column against list box specs
    Rem NOTE that this parameter is zero-based, as differentiated from ListBox _
    control BoundColumn property, etc.
    If 0 <= lngListDescCol And lngListDescCol <= mlst.ColumnCount - 1 Then
        mlngListDescCol = lngListDescCol
    ElseIf lngListDescCol < 0 Then
        MsgBox "SelCtlList.SetProperties - specified lngListDescCol " & _
            "parameter indicates a column index < 0. List box column " & _
            "are always zero or greater " & strListBoxName
    ElseIf lngListDescCol > mlst.ColumnCount - 1 Then
        MsgBox "SelCtlList.SetProperties - specified lngListDescCol " & _
            "parameter indicates a column that is not defined in " & _
            "properties of the specified list box" & strListBoxName
    End If

    Rem Set value of property
    Rem SetOperator returns non-empty string if a valid InUse value _
    is provided
    If Operator(lngInUse) <> "" Then
        mlngInUse = lngInUse
    Else
        MsgBox "SelCtlTxt.SetProperties - Invalid value specified for " & _
            "lngInUse parameter" & strListBoxName
    End If

    Rem Set value of property
    Select Case lngValType
        Case sclChar, sclDate, sclNbr
            mlngValType = lngValType
```

```

        Case Else
            MsgBox "SelCtlTxt.SetProperties - Invalid value specified for " & _
                "lngValType parameter" & strListBoxName
    End Select

    Rem Set value of property
    If Len(strTrgtCol) > 0 Then
        mstrTrgtCol = strTrgtCol
    Else
        MsgBox "SelCtlList.SetProperties - invalid value (zero length string) " & _
            "specified for strTrgtCol parameter" & strListBoxName
    End If

    Rem Set value of property
    If Len(strTrgtTableAlias) > 0 Then
        mstrTrgtTableAlias = strTrgtTableAlias
    Else
        MsgBox "SelCtlList.SetProperties - invalid value (zero length string) " & _
            "specified for strTrgtTableAlias parameter" & strListBoxName
    End If

Exit_SetProperties:
    Exit Sub

Err_SetProperties:
    MsgBox "SelCtlList.SetProperties Error Nbr " & Err.Number & " - " & _
        Err.Description & strListBoxName
    Resume Exit_SetProperties

End Sub

Public Sub SetSelClause()
    Rem Construct SQL clause used for record filter. Typically called by _
        a controls "after update" event procedure in a form. Also builds _
        selection description used in report subtitles, etc. to summarize _
        the record selection criteria used to generate the report.

    On Error GoTo Err_SetSelClause

    Dim str As String
    Dim strFinal As String
    Dim strSelectedId As String
    Dim varRow As Variant

    str = ""
    strFinal = ""

    Rem Run procedure to check for required user specifications
    Call CheckReqs

    Rem Set value only if no gross violations of requirements are found
    If mbReqsOK = True Then
        str = InExpr()
    End If

    mstrSelClause = str

    Rem Set SelDesc property to match newly constructed SelClause value
    Call SetSelDesc

```

```

Exit_SetSelClause:
    Exit Sub

Err_SetSelClause:
    MsgBox "SelCtlList.SetSelClause Error Nbr " & Err.Number & " - " &
Err.Description
    Resume Exit_SetSelClause

End Sub

Private Sub SetSelDesc()
    Rem Construct value of SelDesc property corresponding to value of SelClause _
    property. Typically called by SetSelClause method, since the SelDesc _
    property should be set to correspond to the SelClause property each time _
    a value is constructed.

    On Error GoTo Err_SetSelDesc

    Dim strInteractDesc As String      'Desc of "all/any one" use of selections
    Dim strSelAbbr As String           'Abbreviation used to describe selection
    Dim strSelsDesc As String          'Overall comma-separated list of all selections
    Dim strTempDesc As String
    Dim varRow As Variant              'Used to represent a single element of _
                                        ItemsSelected collection associated with _
                                        list box

    strSelsDesc = ""
    strTempDesc = ""

    Rem Construct description only if base requirements are OK and if a _
    matching SelClause exists
    If mbReqsOK = True And Len(mstrSelClause) > 0 Then

        Rem Create "id interaction" part of overall selection description
        If mlngInUse = sclIn Then
            strInteractDesc = " in at least one of  "
        ElseIf mlngInUse = sclNotIn Then
            strInteractDesc = " not in at least one of  "
        Else
            strInteractDesc = ""
        End If

        Rem Build list of selected group abbreviations
        With mlst
            Rem Handle construction differently depending on multiselect setting
            If .MultiSelect = 0 Then
                strSelsDesc = .Column(mlngListDescCol)
            Else
                For Each varRow In .ItemsSelected
                    Rem Human-language description of value is taken from list box _
                    column indicated in ListDescCol property
                    strSelAbbr = .Column(mlngListDescCol, varRow)
                    Call basSelCtlShared.AddToCSVList(strSelsDesc, strSelAbbr)
                Next varRow
            End If
        End With

        strTempDesc = "'" & mstrTrgtCol & "' " & strInteractDesc & _
            "(" & strSelsDesc & ")"
    End If

```

```

End If

mstrSelDesc = strTempDesc

Exit_SetSelDesc:
Exit Sub

Err_SetSelDesc:
MsgBox "SelCtlList.SetSelDesc Error Nbr " & Err.Number & " - " & Err.Description
Resume Exit_SetSelDesc

End Sub

Public Sub SetStoredCtlVal()
Rem Sets module-level variables used to revert values _
if a user entry results in zero records being selected

On Error GoTo Err_SetStoredCtlVal

'Debug.Print "scl.SetStoredCtlVal - started"

Rem Store current SelClause and SelDesc values
Call SetStoredSelVal

Rem Store current list box selections
Call SetStoredLstSel

Exit_SetStoredCtlVal:
Exit Sub

Err_SetStoredCtlVal:
MsgBox "SelCtlList.SetStoredCtlVal Error Nbr " & Err.Number & " - " & _
Err.Description
Resume Exit_SetStoredCtlVal

End Sub

Private Sub SetStoredLstSel()
Rem Sets module-level variables used to revert values _
if a user entry results in zero records being selected

Dim intI As Integer
Dim varRow As Variant

On Error GoTo Err_SetStoredLstSel

'Debug.Print "scl.SetStoredLstSel - started"
'Debug.Print "scl.SetStoredLstSel - selection count is " & _
mlst.ItemsSelected.Count
mlngStoredLstSelCount = mlst.ItemsSelected.Count

Rem Enlarge storage array if needed, and clear all values indicating _
selected rows
If mlngStoredLstSelCount > UBound(mavarStoredLstSel) Then
ReDim mavarStoredLstSel(mlngStoredLstSelCount)
Else
For intI = 1 To UBound(mavarStoredLstSel)
mavarStoredLstSel(intI) = 0
Next
End If

```

```

Rem If any rows are selected, store their indexes, otherwise the _
    cleared array and the zero count of selected items already stored _
    are all that is needed
If mlngStoredLstSelCount > 0 Then

    Rem Store current list box selections
    intI = 0
    With mlst
        For Each varRow In .ItemsSelected
            intI = intI + 1
            Rem Array element stores row number of selected row
            mavarStoredLstSel(intI) = varRow
            'Debug.Print "    Row '" & varRow & "' is selected"
        Next varRow
    End With

End If

Exit_SetStoredLstSel:
Exit Sub

Err_SetStoredLstSel:
MsgBox "SelCtlList.SetStoredLstSel Error Nbr " & Err.Number & " - " & _
    Err.Description
Resume Exit_SetStoredLstSel

End Sub

Private Sub SetStoredSelVal()
    Rem Stores current values of SelClause and SelDesc to allow for a fast _
    and simple reversion of values without need to rerun calculations

    mstrStoredSelClause = mstrSelClause
    mstrStoredSelDesc = mstrSelDesc

End Sub

Property Get ctl() As ListBox
    Rem Return listbox object used to select group ids
    ctl = mlst
End Property

Property Get InUse() As sclInUse
    Rem Return selection type ("In", "NotIn" of selections)
    InUse = mlngInUse
End Property

Property Let InUse(ByVal lng As sclInUse)
    Rem Set selection type ("In", "NotIn" of selections)
    mlngInUse = lng
End Property

Property Get ListDescCol() As Long
    Rem Return column index of column designated to contain human-language _
    descriptions of selections
    ListDescCol = mlngListDescCol
End Property

Property Get SelClause() As String

```

```
    Rem Return filter clause representing control selections
    SelClause = mstrSelClause
End Property

Property Get SelDesc() As String
    Rem Return filter clause representing control selections
    SelDesc = mstrSelDesc
End Property

Property Get TrgtCol() As String
    Rem Return name of group id column in target group membership table
    TrgtCol = mstrTrgtCol
End Property

Property Get TrgtTableAlias() As String
    Rem Return name of target group membership table
    TrgtTableAlias = mstrTrgtTableAlias
End Property

Property Get ValueType() As sclValType
    Rem Return group interaction type ("char", "date", "nbr")
    ValueType = mlngValType
End Property
```