Home x Install x Full Index x Tutorial x EnumerateControls x FindOutTableExists x UseVariablesInSQL x CreateRecordsetFrom x AddRecordToRecordset x CountRecordsRecordset x LimitsRecordset x MixAccess2baseAndUNO ! User's Guide x AllForms x DatabaseWindow x ShortcutNotationMore x DLookupSamples x CalculatedField x MultiSelectListboxSelectForm x FillAutoControlValue x CarryToNewRecord x BrowseThruControls x TipTextForLongValues x AskBeforeSaving x Sync2Combos x ZoomOnImage x AddAllToBox x KeepFormsSynchro x SelectListboxOnFirstLetters x MoveItemsBetweenListboxes x SimulateTabbed x SearchStandalone x CalculatorDialog x

CalculatorDialog

(Q) Can a numeric field be edited thru a Calculator widget like Base forms propose a Calendar widget for date fields ?

tags:
HowTo

(R) A calculator can be implemented as a **Dialog**. When the dialog is closed the computed result is copied into the concerned form field.

The solution must implement a *as easy as possible* programmatic AND user interface from the initial form event activating the calculator up to the final copy of the result into the numeric form field.
See the suggested implementation in the *Calculator* form of the provided examples database: a button with a nice calculator icon opens the dialog.

Steps:

Make a form containing the targeted numeric fields and one button by such field.

Link the *Execute action* event of each button with a macro similar to next one. One macro by each field/button pair.

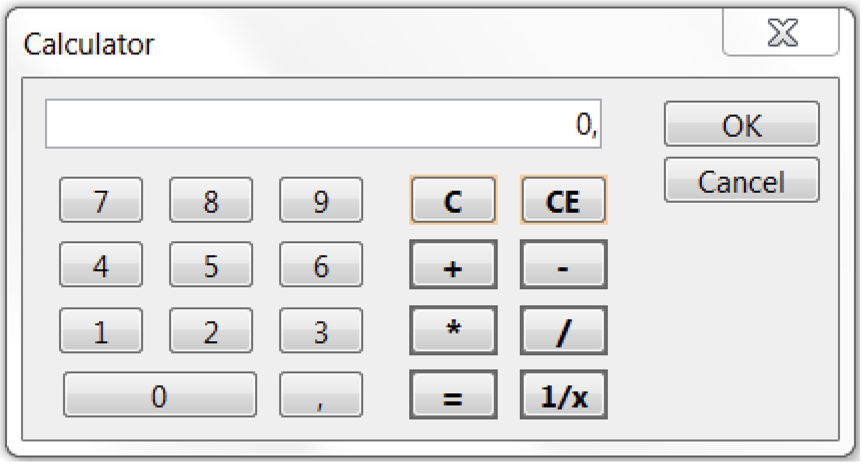
```
Sub StartCalculator(poEvent As Object)
'       StartCalculator should be adapted to user's need:
'       - it is assumed to be activated from an appropriate button
'       - the field to compute is assumed in the same form or subform as the button
'       - the field to compute must be of type NUMERICFIELD or CURRENCYFIELD
'       - the name of the field to compute should be set in next line
'       - The 3rd argument is optional. True = initialize calculator with field current value

        Call StartCalcDialog(poEvent, "TargetCalcField", True)

End Sub
```

Of course adapt the code to your needs !

~~Design a calculator such as~~ ... Export and re-import next calculator dialog ...



The code behind the *StartCalcDialog* Sub

The code is stored in the *Calculator* module of the provided example file. It is inspired by an old sample code from Microsoft. It is generic enough to adapt the display of the decimal point to the locale settings.

1. The current status of the Calculator is constantly maintained in next variables

```
Type CalcBuffer
        DisplayField            As Object               '       Display window on calculator
        Operand1                As Double               '       1st Operand
        Operand2                As Double               '       2nd Operand
        DisplayText             As String               '       To display in dialog
        NumberOfOperands        As Integer              '       1 or 2
        DecimalPoint            As Boolean              '       Decimal point present yet ?
        LastInput               As String               '       Indicate type of last keypress
        PendingOperation        As String               '       Indicate pending operation (+,
        LocalePoint             As String               '       Local decimal point
End Type

Public gCalc As CalcBuffer
Const cstStdFormat = "General Number"          '       Format of display in calculator
```

2. Next Sub (the core Sub) checks the arguments, manages the display of the dialog window, and copies the result in the original numeric field if the OK button is pressed:

```
Sub StartCalcDialog(poEvent, ByVal psFieldName As String, Optional ByVal pbCopy As Boolean)

Dim ocFieldToCompute As Object, ocButton As Object, ofForm As Object, oDialog As Object
Dim i As Integer, bFound As Boolean, iDialog As Integer
        Set ocButton = Application.Events(poEvent).Source
        If ocButton.ObjectType <> "CONTROL" Then Exit Sub

        If IsMissing(pbCopy) Then pbCopy = True

        'Check field name exists
        Set ofForm = ocButton.Parent    '       ofForm could be a form or a subform !
        bFound = False
        For i = 0 To ofForm.Controls().Count - 1
                Set ocFieldToCompute = ofForm.Controls(i)
                If UCase(ocFieldToCompute.Name) = UCase(psFieldName) Then
                        bFound = True
                        Exit For
                End If
        Next i
        If Not bFound Then
                TraceLog("ERROR", "Field name " & psFieldName & " not found in form or subfor
                Exit Sub
        End If

        'Check field is of admitted types
        If ocFieldToCompute.SubType <> "NUMERICFIELD" And ocFieldToCompute.SubType <> "CURRENC
                        TraceLog("ERROR", "Field " & psFieldName & " is not numeric")
                        Exit Sub
        End If

Const dlgOK = 1                         '       OK button pressed
Const dlgCancel = 0                     '       Cancel button pressed
        Set oDialog = Application.AllDialogs("dlgCalc")
        oDialog.Start

        '       Initialize gCalc
        gCalc.DisplayField = oDialog.Controls("CalcDisplay")
        gCalc.LocalePoint = Right(Format(0,"General Number"),1)
        If pbCopy Then
                gCalc.Operand1 = ocFieldToCompute.Value
                gCalc.Operand2 = gCalc.Operand1
                gCalc.DisplayText = Join(Split(Format(gCalc.Operand1, cstStdFormat), ","), ".
                gCalc.DecimalPoint = ( Abs(gCalc.Operand1 - Fix(gCalc.Operand1)) > 0 )
                gCalc.NumberOfOperands = 1
                gCalc.LastInput = "OPS"
                gCalc.PendingOperation = "="
        Else
                gCalc.LastInput = "NONE"
                gCalc.NumberOfOperands = 0
                gCalc.PendingOperation = " "
                gCalc.DecimalPoint = False
                gCalc.Operand1 = 0
                gCalc.Operand2 = 0
        End If

        'Load dialog
        gCalc.DisplayField.Value = Format(gCalc.Operand1, cstStdFormat)
        oDialog.Controls("BtnPoint").Caption = gCalc.LocalePoint                '       Set d
        iDialog = oDialog.Execute
        Select Case iDialog
                Case dlgOK
                        ocFieldToCompute.Value = gCalc.Operand1
                Case dlgCancel
        End Select

        oDialog.Terminate
        Exit Sub
End Sub
```

3. The calculator should react to each activation of any button in the dialog box. Link all buttons - except *OK* and *Cancel* - *Execute action* event to next Sub:

```
Sub CalcButtonPressed(poEvent As Object)

Dim oEvent As Object, sName As String, oButton As Object, oDisplay As Object, sChar As String
        Set oEvent = Application.Events(poEvent)
        If oEvent.EventType <> "ACTIONEVENT" Then Exit Sub

        Set oButton = oEvent.Source
        sName = UCase(oButton.Name)
        Select Case sName
                Case "BTNADD"                   :       sChar = "+"
                Case "BTNSUB"                   :       sChar = "-"
                Case "BTNMULT"                  :       sChar = "*"
                Case "BTNDIV"                   :       sChar = "/"
                Case "BTNENTER"                 :       sChar = "="
                Case "BTNCLEAR"                 :       sChar = "C"
                Case "BTNCE"                    :       sChar = "CE"
                Case "BTNINVERT"                :       sChar = "1/x"
                Case "BTNPOINT"                 :       sChar = "."
                Case Else                       :       sChar = Right(sName, 1)         '
        End Select
        Call ProcessKey(sChar)
        Exit Sub

End Sub
```

4. In addition, the caculator should react in the same way when the equivalent key is pressed. Link the *Key pressed* event of the *CalcDisplay* field (i.e. the display of the calculator) to next Sub:

```
Sub CalcKeyPressed(poEvent As Object)

Dim oEvent As Object, oDisplay As Object, sChar As String
        Set oEvent = Application.Events(poEvent)
        If oEvent.EventType <> "KEYEVENT" Then Exit Sub

'Accepted keys: 0-9, BACKSPACE, C, ESCAPE, dot, comma, +, -, *, /, %, =, ENTER.
'All other keys ignored
        With oEvent
                Select Case True        '       Both KeyCode and KeyChar used to be generic a
                        Case .KeyAlt, .KeyCtrl  :       Beep    :       Exit Sub        '
                        Case .KeyCode = com.sun.star.awt.Key.ESCAPE Or UCase(.KeyChar) = "C"
                        Case .KeyCode = com.sun.star.awt.Key.BACKSPACE
                        Case .KeyCode = com.sun.star.awt.Key.RETURN Or .KeyCode = com.sun.sta
                                sChar = "="
                        Case .KeyCode = com.sun.star.awt.Key.ADD Or .KeyChar = "+"
                        Case .KeyCode = com.sun.star.awt.Key.SUBTRACT Or .KeyChar = "-"
                        Case .KeyCode = com.sun.star.awt.Key.MULTIPLY Or .KeyChar = "*"
                        Case .KeyCode = com.sun.star.awt.Key.DIVIDE Or .KeyChar = "/" Or .Key
                                sChar = "/"
                        Case .KeyChar = "_"
                        Case .KeyCode = com.sun.star.awt.Key.DECIMAL Or .KeyCode = com.sun.st
                                Or .KeyCode = com.sun.star.awt.Key.COMMA Or .KeyChar = "." Or
                                sChar = "."
                        Case .KeyChar >= "0" And .KeyChar <= "9"
                        Case .KeyCode >= com.sun.star.awt.Key.NUM0 And .KeyCode <= com.sun.st
                                sChar = Trim(Str(.KeyCode - com.sun.star.awt.Key.NUM0))
                        Case Else               :       Beep    :       Exit Sub
                End Select
        End With

        Call ProcessKey(sChar)
        Exit Sub

End Sub
```

5. Now the real processing of the entered key or the clicked button:

```
Sub ProcessKey(ByVal psChar As String)
'       Process gCalc structure based on argument

Dim sDisplayText As String
Const cstMax = 999999999999

        Select Case psChar
                Case "C"                '       Cancel
                        gCalc.DisplayText = Format(0, "0.")
                        gCalc.Operand1 = 0
                        gCalc.Operand2 = 0
                        gCalc.NumberOfOperands = 0
                        gCalc.PendingOperation = " "
                        gCalc.LastInput = "NONE"
                Case "CE"               '       Cancel entry
                        gCalc.DisplayText = Format(0, "0.")
                        gCalc.DecimalPoint = False
                        gCalc.LastInput = "CE"
                Case "."                '       Decimal point
                ' If last keypress was an operator, initialize DisplayText to "0."
                ' Otherwise, append a decimal point to the display.
                        If gCalc.DecimalPoint Then
                                Beep
                        Else
                                If gCalc.LastInput = "NEG" Then
                                        gCalc.DisplayText = Format(0, "-0.")
                                ElseIf gCalc.LastInput <> "NUMS" Then
                                        gCalc.DisplayText = Format(0, "0.")
                                End If
                                gCalc.DecimalPoint = True
                                gCalc.LastInput = "NUMS"
                        End If
                Case "+", "-", "*", "/", "="    '       Arithmetic operators
                ' If the immediately preceeding keypress was part of a number, increment Deci
                ' set Operand1. If two are present, set Operand1 equal to the result of the o
                ' input string, and display the result.
                        sDisplayText = gCalc.DisplayText
                        If gCalc.LastInput = "NUMS" Then
                                gCalc.NumberOfOperands = gCalc.NumberOfOperands + 1
                        End If
                        Select Case gCalc.NumberOfOperands
                                Case 0
                                        If psChar = "-" And gCalc.LastInput <> "NEG" Then
                                                gCalc.DisplayText = "-" & gCalc.DisplayText
                                                gCalc.LastInput = "NEG"
                                        End If
                                Case 1
                                        gCalc.Operand1 = Val(gCalc.DisplayText)
                                        If psChar = "-" And gCalc.LastInput <> "NUMS" And gCa
                                                gCalc.DisplayText = "-"
                                                gCalc.LastInput = "NEG"
                                        End If
                                Case 2
                                        gCalc.Operand2 = Val(sDisplayText)
                                        Select Case gCalc.PendingOperation
                                                Case "+"
                                                        gCalc.Operand1 = gCalc.Operand1 + gCa
                                                Case "-"
                                                        gCalc.Operand1 = gCalc.Operand1 - gCa
                                                Case "*"
                                                        gCalc.Operand1 = gCalc.Operand1 * gCa
                                                Case "/"
                                                        If Sgn(gCalc.Operand2) = 0 Then
                                                                gCalc.Operand1 = cstMax * Sgn
                                                        Else
                                                                gCalc.Operand1 = gCalc.Operan
                                                        End If
                                                Case "="
                                                        gCalc.Operand1 = gCalc.Operand2
                                        End Select
                                        gCalc.DisplayText = Join(Split(Format(gCalc.Operand1,
                                        gCalc.NumberOfOperands = 1
                        End Select
                        If gCalc.LastInput <> "NEG" Then
                                gCalc.LastInput = "OPS"
                                gCalc.PendingOperation = psChar
                        End If
                Case "1/x"      '       Invert result
                        If gCalc.LastInput = "NUMS" Then gCalc.Operand1 = Val(gCalc.DisplayTex
                        If Sgn(gCalc.Operand1) = 0 Then
                                gCalc.Operand1 = cstMax
                        Else
                                gCalc.Operand1 = 1 / gCalc.Operand1
                        End If
                        gCalc.LastInput = "OPS"
                        gCalc.DisplayText = Join(Split(Format(gCalc.Operand1, cstStdFormat), '
                Case Else               '       DIGIT
                ' Append new number to the number in the display.
                        If gCalc.LastInput <> "NUMS" Then
                                gCalc.DisplayText = Format(0, ".")
                                gCalc.DecimalPoint = False
                        End If
                        If gCalc.DecimalPoint Then
                                gCalc.DisplayText = gCalc.DisplayText & psChar
                        Else
                                gCalc.DisplayText = Left(gCalc.DisplayText, InStr(gCalc.Displa
                        End If
                        If gCalc.LastInput = "NEG" Then gCalc.DisplayText = "-" & gCalc.Displa
                        gCalc.LastInput = "NUMS"
        End Select

        'DebugPrint psChar, gCalc.Operand1, gCalc.Operand2, gCalc.PendingOperation, gCalc.Numb
        gCalc.DisplayField.Value = Join(Split(gCalc.DisplayText, "."), gcalc.LocalePoint)
        Exit Sub

End Sub
```

See also

Dialog
Control
Events Handler

Refer to ...

Basic module	Form	Form event	Control	Control event	Comments
Calculator	Calculator		Calculator	Execute action	Enter an optional initial value and click on the icon.
	dlgCalc (dialog)		cmd0...9 btnClear btnCE btnAdd btnSub btnMult btnDiv btnEnter btnInvert	Execute action	
			CalcDisplay	Key pressed	

Bookmark this page » » CalculatorDialog