

Spread Sheet

ActiveX

ActiveX

100

ActiveX
가

MSFlexGrid
MSFlexGrid

Spread sheet

Spread sheet
MDI 가

sheet 가
1 Spread sheet MDI

가



1 Spread sheet MDI

| | (Name) | | |
|--------------|--------------|---------|------------|
| MDI From | frmMiniSheet | Caption | Mini sheet |
| CommonDialog | cdlFile | | |

1 Spread sheet MDI

| Caption | (Name) | | |
|-------------------|-----------|----------|--------|
| &File | mnuFile | | |
| &New sheet | mnuNew | | |
| &Open sheet... | mnuOpen | | |
| - | mnuSep1 | | |
| &Save sheet | mnuSave | | |
| Save sheet &as... | mnuSaveAs | | |
| - | mnuSep2 | | |
| &Exit | mnuExit | | |
| &Edit | mnuEdit | | |
| &Copy | mnuCopy | Shortcut | Ctrl+C |
| &Paste | mnuPaste | Shortcut | Ctrl+V |
| &Tools | mnuTools | | |
| &Graph | mnuGraph | | |

2 Menu

1 Spread sheet MDI
CommonDialog

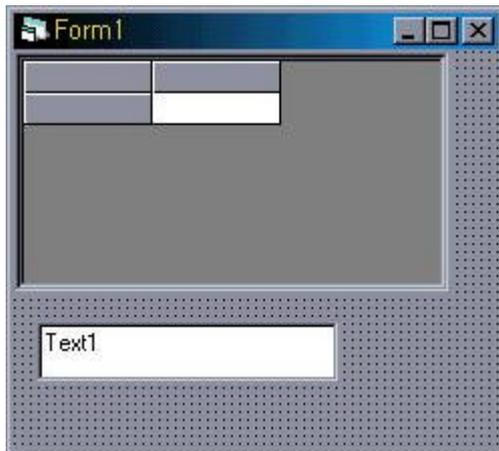
2

Ctrl-T

“Microsoft Common Dialog Control”

sheet

MDIChild



2 Sheet MDIChild

| | (Name) | | |
|------------|----------|----------|------|
| Form | frmSheet | MDIChild | True |
| MSFlexGrid | grdSheet | Top | 0 |
| | | Left | 0 |
| Text Box | txtRec | | |

3 Spread sheet MDI

2 Sheet MDIChild
3 MSFlexGrid
Spread sheet 가
MSFlexGrid Sheet

가

가

Project-Add module
가

ExpCalc

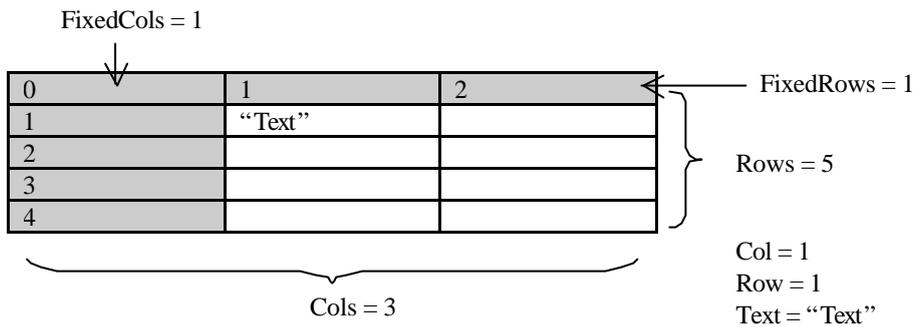
Project

MiniSheet

SaveAs . shtFilename Sheet MDIChild SaveSheet New SaveAs
 Sheet 가 Sheet
 Exit MDI Unload . Copy Paste Sheet
 MDIChild Copy Paste Sheet

frmSheet

Mini Sheet frmSheet MDIChild
 MSFlexGrid . MSFlexGrid 가



3 MSFlexGrid

MSFlexGrid 3 가 Cols Rows
 FixedCols FixedRows
 Col Row Text
 Row ColSel, RowSel Clip Col,
 CellForeColor, CellBackColor CellAlignment

3 MDIChild

3 MDIChild

Option Explicit

```
Const shtNewName = "Untitled"
Const shtMaxCol = 30
Const shtMAXRow = 1000
```

```
Dim shtNotSaved As Boolean
Public shtFilename As String
```

```
Private Sub Form_Load()
  Dim i As Long
```

```
  shtNotSaved = False
  shtFilename = ""
  frmMiniSheet.SheetNo = frmMiniSheet.SheetNo + 1
  Me.Caption = shtNewName & " - " & frmMiniSheet.SheetNo & " [Sheet]"
```

```
  grdSheet.Redraw = False
```

```

grdSheet.AllowUserResizing = flexResizeColumns
grdSheet.AllowBigSelection = True
grdSheet.Cols = shtMaxCol
grdSheet.Rows = shtMAXRow

grdSheet.Row = 0
For i = 1 To shtMaxCol - 1
    grdSheet.Col = i
    grdSheet.Text = ColNo2ColName(i)
    grdSheet.FixedAlignment(i) = flexAlignCenterCenter
Next
grdSheet.Col = 0
For i = 1 To shtMAXRow - 1
    grdSheet.Row = i
    grdSheet.Text = i
Next

grdSheet.Col = 1
grdSheet.Row = 1
grdSheet.Redraw = True

txtRec.BorderStyle = vbBSNone
txtRec.Visible = False
End Sub

Private Sub Form_Resize()
    grdSheet.Width = Me.ScaleWidth
    grdSheet.Height = Me.ScaleHeight
End Sub

Private Sub Form_Unload(Cancel As Integer)
    Dim Canceled As Integer

    If shtNotSaved Then
        Canceled = MsgBox("Sheet is not saved. Do you want to save the sheet?",
            vbYesNoCancel + vbInformation, "Not saved")
        If Canceled = vbYes Then
            frmMiniSheet.mnuSave_Click
        ElseIf Canceled = vbCancel Then
            Cancel = 1
        End If
    End If
End Sub

Private Function ColNo2ColName(ByVal Col As Long) As String
    Dim name As String
    Dim digit As Integer

    name = ""
    Col = Col - 1
    Do
        digit = Col Mod 26
        Col = Col \ 26
        name = Chr(Asc("A") + digit) & name
        If Col = 0 Then Exit Do
        Col = Col - 1
    Loop
    ColNo2ColName = name
End Function

```

```

shtMaxCol    shtMaxRow    Sheet
shtFilename  Sheet
.shtNotSaved sheet 가
Public
frmSheet
frmSheet    Load    가
MSFlexGrid
    MSFlexGrid
    MSFlexGrid
    Redraw    False
    AllowUserResizing
가
    . AllowBigSelection
    grdSheet    가
ColNo2ColName
1    "A", 2    "B", 3    "C"
ColNo2ColName
    . FixedAlignment
    txtRec    가
BorderStyle    vbBSNoe
Visible    False
frmSheet    가
    MSFlexGrid    가
    Resize
frmSheet 가 Unload    sheet 가
    shtNotSaved 가 True    MsgBox
    sheet    가
    MsgBox    vbYes(6), vbNo(7), vbCancel(2)    . Yes
frmMiniSheet MDI    Save_Click
    가 Save
    . No    frmSheet 가
Cancel    Cancel    0    frmSheet
가
    Sheet 가
    가
    MSFlexGrid
    MSFlexGrid

```

5.0 MSFlexGrid

4 grdSheet, txtRec

```

Private Sub grdSheet_DblClick()
    grdEdit 32
End Sub

Private Sub grdSheet_GotFocus()
    grdText2Cell
End Sub

Private Sub grdSheet_KeyPress(KeyAscii As Integer)
    grdEdit KeyAscii
End Sub

Private Sub grdSheet_LeaveCell()
    grdText2Cell
End Sub

Private Sub txtRec_KeyDown(KeyCode As Integer, Shift As Integer)
    EditKeyCode KeyCode, Shift
End Sub

```

```

Private Sub txtRec_KeyPress(KeyAscii As Integer)
    If KeyAscii = Asc(vbCr) Then KeyAscii = 0
End Sub

Private Sub txtRec_LostFocus()
    grdText2Cell
End Sub

4   grdSheet   txtRec   .   grdSheet   KeyPress
    .   txtRec
    .   grdEdit   grdSheet   가   txtRec   txtRec
txtRec   가   가   txtRec   KeyDown
    .   txtRec   grdSheet
    .   txtRec   가
    (LostFocus)   txtRec   grdSheet
grdText2Cell   .   grdSheet   가
(GotFocus),   (LeaveCell)   grdText2Cell   .
txtRec   KeyPress   가   KeyAscii=0   가
    가
5   4   .

```

5

```

Private Sub grdEdit(KeyAscii As Integer)
    Select Case KeyAscii
        Case 32
            txtRec.Text = grdSheet.Text
            txtRec.SelStart = 1
            txtRec.SelLength = Len(txtRec.Text)
        Case Else
            txtRec.Text = Chr(KeyAscii)
            txtRec.SelStart = 1
            shtNotSaved = True
    End Select

    txtRec.Move grdSheet.CellLeft, grdSheet.CellTop + grdSheet.Top, _
                grdSheet.CellWidth, grdSheet.CellHeight
    txtRec.Visible = True
    txtRec.SetFocus
End Sub

```

```

Private Sub EditKeyCode(KeyCode As Integer, Shift As Integer)
    Select Case KeyCode
        Case vbKeyEscape
            txtRec.Visible = False
            grdSheet.SetFocus
        Case vbKeyReturn
            grdSheet.SetFocus
            grdMoveNextCell
        Case vbKeyUp
            grdSheet.SetFocus
            If grdSheet.Row > grdSheet.FixedRows Then _
                grdSheet.Row = grdSheet.Row - 1
        Case vbKeyDown
            grdSheet.SetFocus
            If grdSheet.Row < grdSheet.Rows - 1 Then _
                grdSheet.Row = grdSheet.Row + 1
    End Select
End Sub

```

```

Private Sub grdText2Cell()
    Dim s As String

    If Not txtRec.Visible Then Exit Sub
    s = Trim(txtRec.Text)
    If Mid(s, 1, 1) = "=" Then
        s = Mid(s, 2, Len(s) - 1)
        GetExpCalc Me, s
    End If
    If IsNumeric(s) Then
        grdSheet.CellAlignment = flexAlignRightCenter
    Else
        grdSheet.CellAlignment = flexAlignLeftCenter
    End If
    grdSheet.Text = s
    txtRec.Visible = False
End Sub

Private Sub grdMoveNextCell()
    grdText2Cell
    If grdSheet.Col < grdSheet.Cols - 1 Then grdSheet.Col = grdSheet.Col + 1
End Sub

```

```

grd2Text2Cell    txtRec        Sheet                                grdText2Cell
                                                "=="
    txtRec.Text 가 "="                GetExpCalc
    . GetExpCalc    ExpCalc
    CellAlignment
grdSheet.Text    s                txtRec                Sheet
가

```

Copy & Paste

Copy & Paste

6

6 Copy & Paste

```

Public Sub Copy()
    grdSheet.Redraw = False
    Clipboard.SetText grdSheet.Clip
    grdSheet.Redraw = True
End Sub

Public Sub Paste()
    Dim ColSel As Long, RowSel As Long

    grdSheet.Redraw = False
    ColSel = grdSheet.ColSel
    RowSel = grdSheet.RowSel

    grdSheet.ColSel = grdSheet.Cols - 1
    grdSheet.RowSel = grdSheet.Rows - 1
    grdSheet.Clip = Clipboard.GetText

    grdSheet.ColSel = ColSel
    grdSheet.RowSel = RowSel
    grdSheet.Redraw = True
End Sub

```



```

Loop

grdSheet.Row = 1
grdSheet.Col = 1
grdSheet.Redraw = True
Close #1

shtFilename = filename
End Sub

```

가 가 , 3 Open

Open *pathname* **For** *mode* **As** [#]*filenumber*

pathname *path* . *mode* Append,
 Binary, Input, Output, Random 가 . 4 .

| Mode | mode | access | Text/Binary | |
|--------|------|--------|-------------|--|
| Append | | | Text | |
| Binary | / | | Binary | |
| Input | | | Text | |
| Output | | | Text | |
| Random | / | | Text/Binary | |

4

. filenumber
 . 1 511 Open 가 .
 Mini Sheet Input
 Output Input #, Write# Line Input #, Print #
 가 가 , .
 SaveSheet ColSel Clip sheet shtFilename
 . OpenSheet sheet

ExecCalc 가 . .
 8 가 . GetExpCalc 가 .
 , “ ”, “ ”, “ ”, “ ” . GetToken “ ”, “ ”

EvalExp 가
 가 가 가 , . EvalExp
 4

8 가

```
Public Sub GetExpCalc(ExpStr)
  Buff = ExpStr
  GetToken dOP
  ExpStr = EvalExp(dOP)
End Sub
```

```
Private Function EvalExp(dOP)
  Opr1 = EvalExp2(dOP)
  Do While (dOP = idAdd) Or (dOP = idSub)
    OP = dOP
    GetToken dOP
    Opr1 = Calc(Opr1, EvalExp2(dOP), OP)
  Loop
  EvalExp = Opr1
End Function
```

```
Private Function EvalExp2(dOP)
  Opr1 = EvalExp3(dOP)
  Do While (dOP = idMul) Or (dOP = idDiv)
    OP = dOP
    GetToken dOP
    Opr1 = Calc(Opr1, EvalExp3(dOP), OP)
  Loop
  EvalExp2 = Opr1
End Function
```

```
Private Function EvalExp3(dOP)
  Opr1 = EvalExp4(dOP)
  Do While dOP = idPow
    OP = dOP
    GetToken dOP
    Opr1 = Calc(Opr1, EvalExp4(dOP), OP)
  Loop
  EvalExp3 = Opr1
End Function
```

```
Private Function EvalExp4(dOP)
  Select Case dOP
    Case idMinus
      GetToken dOP
      Opr1 = - EvalExp4(dOP)
    Case idSin
      GetToken dOP
      Opr1 = Sin(EvalExp4(dOP))
    Case idLpar
      GetToken dOP
      Opr1 = EvalExp(dOP)
      If dOP = idRpar Then GetToken dOP
    Case idNumber
      Opr1 = CSng(dOP)
      GetToken dOP
  End Select
  EvalExp4 = Opr1
End Function
```

| | | | |
|---------|----------------|--------|----------|
| EvalExp | | | . dOP |
| | EvalExp2 | dOP | Opr1 |
| dOP | . | 가 | GetToken |
| | . Calc(A,B,OP) | A OP B | |


```

Const idNum = "0", idErr = "ERR", idDot = ".", idEOL = "EOL"
Const idAdd = "+", idSub = "-", idMul = "*", idDiv = "/"
Const idPow = "^", idMinus = "-"
Const idSin = "sin", idCos = "cos", idTan = "tan"
Const idSqrt = "sqrt", idExp = "exp"
Const idLn = "ln"
Const idLpar = "(", idRpar = ")"
Const CellNameStr = "ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789"
Const CellNameStr2 = "ABCDEFGHIJKLMNOPQRSTUVWXYZ"
Const NoOfCmd = 17

```

```

Dim cmdList(NoOfCmd) As String
Dim frmCallSheet As Form
Dim Buff As String, idx As Integer

```

```

Public Sub InitCalc()
    cmdList(1) = idAdd: cmdList(2) = idSub
    cmdList(3) = idMul: cmdList(4) = idDiv
    cmdList(5) = idPow: cmdList(6) = idMinus
    cmdList(7) = idSin: cmdList(8) = idCos
    cmdList(9) = idTan: cmdList(10) = idSqrt
    cmdList(11) = idExp: cmdList(12) = idLn
    cmdList(13) = idLog: cmdList(14) = idAsin
    cmdList(15) = idAcos: cmdList(16) = idLpar
    cmdList(17) = idRpar
End Sub

```

```

Public Sub GetExpCalc(frm As Form, ExpStr As String)
    Dim Attr As String
    Dim Num As Single

    Set frmCallSheet = frm

    Buff = Trim(ExpStr)
    idx = 1
    GetToken Attr
    If Attr = idErr Then
        ExpStr = "Error"
    Else
        Num = EvalExp(Attr)
        If Attr = idErr Then
            ExpStr = "Error"
        Else
            ExpStr = Num
        End If
    End If
End Sub

```

```

Private Sub GetToken(ByRef Attr As String)
    Dim s As String, ss As String
    Dim i As Integer

    If idx > Len(Buff) Then
        Attr = idEOL
        Exit Sub
    End If

    Do While Mid(Buff, idx, 1) = " "
        idx = idx + 1
    Loop

```

```

s = Mid(Buff, idx, 1)
If IsNumeric(s) Or (s = idDot) Then
    ss = ""
    Do
        ss = ss + s
        idx = idx + 1
        If idx > Len(Buff) Then Exit Do
        s = Mid(Buff, idx, 1)
    Loop Until Not (IsNumeric(s) Or (s = idDot))
    If IsNumeric(ss) Then
        Attr = ss
    Else
        Attr = idErr
    End If
Else
    For i = 1 To NoOfCmd
        If InStr(idx, UCase(Buff), UCase(cmdList(i))) = idx Then
            Attr = cmdList(i)
            idx = idx + Len(Attr)
            Exit Sub
        End If
    Next
    If InStr(1, CellNameStr2, UCase(s)) > 0 Then
        ss = ""
        Do
            ss = ss + s
            idx = idx + 1
            If idx > Len(Buff) Then Exit Do
            s = Mid(Buff, idx, 1)
        Loop While InStr(1, CellNameStr, UCase(s)) > 0
        Attr = ss
    Else
        Attr = idErr
    End If
End If
End Sub

Private Function Calc(OP1 As Single, OP2 As Single, OP As String) As Single
    Dim sOP1 As Single, sOP2 As Single

    sOP1 = CSng(OP1)
    sOP2 = CSng(OP2)
    Select Case OP
        Case idAdd
            Calc = sOP1 + sOP2
        Case idSub
            Calc = sOP1 - sOP2
        Case idMul
            Calc = sOP1 * sOP2
        Case idDiv
            Calc = sOP1 / sOP2
        Case idPow
            Calc = sOP1 ^ sOP2
    End Select
End Function

Private Function EvalExp(ByRef Attr As String) As Single
    Dim Num As Single
    Dim OP As String

```

```

    Num = EvalExp2(Attr)
    Do While (Attr = idAdd) Or (Attr = idSub)
        OP = Attr
        GetToken Attr
        Num = Calc(Num, EvalExp2(Attr), OP)
    Loop
    EvalExp = Num
End Function

Private Function EvalExp2(ByRef Attr As String) As Single
    Dim Num As Single
    Dim OP As String

    Num = EvalExp3(Attr)
    Do While (Attr = idMul) Or (Attr = idDiv)
        OP = Attr
        GetToken Attr
        Num = Calc(Num, EvalExp3(Attr), OP)
    Loop
    EvalExp2 = Num
End Function

Private Function EvalExp3(ByRef Attr As String) As Single
    Dim Num As Single
    Dim OP As String

    Num = EvalExp4(Attr)
    Do While Attr = idPow
        OP = Attr
        GetToken Attr
        Num = Calc(Num, EvalExp4(Attr), OP)
    Loop
    EvalExp3 = Num
End Function

Private Function EvalExp4(ByRef Attr As String) As Single
    Dim Num As Single
    Dim Canceled As Boolean

    Select Case Attr
        Case idMinus
            GetToken Attr
            Num = -EvalExp4(Attr)
        Case idSin
            GetToken Attr
            Num = Sin(EvalExp4(Attr))
        Case idCos
            GetToken Attr
            Num = Cos(EvalExp4(Attr))
        Case idTan
            GetToken Attr
            Num = Tan(EvalExp4(Attr))
        Case idExp
            GetToken Attr
            Num = Exp(EvalExp4(Attr))
        Case idSqrt
            GetToken Attr
            Num = Sqr(EvalExp4(Attr))
        Case idln
            GetToken Attr
            Num = Log(EvalExp4(Attr))
    End Select
End Function

```

```

Case idLpar
  GetToken Attr
  Num = EvalExp(Attr)
  If Attr = idRpar Then
    GetToken Attr
    EvalExp4 = Num
    Exit Function
  End If
Case Else
  If IsNumeric(Attr) Then
    Num = CSng(Attr)
    GetToken Attr
  Else
    frmCallSheet.GetCellVal Num, Attr, Canceled
    If Canceled Then
      Attr = idErr
    Else
      GetToken Attr
    End If
  End If
End Select
EvalExp4 = Num
End Function

'      : GetCellVal      frmSheet MDIChild      .
'
Public Sub GetCellVal(Num As Single, CellName As String, _
                    Canceled As Boolean)

  Dim ColName As String, RowName As String
  Dim s As String, idx As Integer
  Dim ColNo As Long, RowNo As Long
  Dim j As Integer

  ColName = ""
  RowName = ""

  idx = 1
  Do
    s = UCase(Mid(CellName, idx, 1))
    If (s < "A") Or (s > "Z") Then Exit Do
    idx = idx + 1
    ColName = ColName + s
  Loop Until idx > Len(CellName)

  If idx > Len(CellName) Then
    Canceled = True
    Exit Sub
  End If

  Do
    s = UCase(Mid(CellName, idx, 1))
    If (s < "0") Or (s > "9") Then Exit Do
    idx = idx + 1
    RowName = RowName + s
  Loop Until idx > Len(CellName)

  ColNo = Asc(Right(ColName, 1)) - Asc("A") + 1
  j = 1
  For idx = Len(ColName) - 1 To 1 Step -1
    ColNo = ColNo + (Asc(Mid(ColName, idx, 1)) - Asc("A") + 1) * 26 ^ j
  Next idx

```

```
    j = j + 1
Next

RowNo = CLng(RowName)

s = grdSheet.TextMatrix(RowNo, ColNo)

If IsNumeric(s) Then
    Num = CSng(s)
    Canceled = False
Else
    Canceled = True
End If

End Sub
```