VERSION 5.00

Object = "{F9043C88-F6F2-101A-A3C9-08002B2F49FB}#1.2#0"; "COMDLG32.OCX"

Begin VB.Form frmCalc

BorderStyle = 3 'Fixed Dialog

Caption = "AGDISP Calculations"

ClientHeight = 6705

ClientLeft = 1755

ClientTop = 1980

ClientWidth = 6150

ForeColor = &H80000008&

Icon = "CALC.frx":0000

LinkTopic = "Form1"

LockControls = -1 'True

MaxButton = 0 'False

PaletteMode = 1 'UseZOrder

ScaleHeight = 6705

ScaleWidth = 6150

Begin VB.Frame fraFiles

Caption = "Files to Process"

ForeColor = &H80000008&

Height = 1815

Left = 120

TabIndex = 10

Top = 0

Width = 5895

Begin VB.CommandButton cmdRemove

Caption = "&Remove"

Height = 375

HelpContextID = 1050

Left = 2880

TabIndex = 3

Top = 1320

Width = 975

End

Begin VB.CommandButton cmdAdd

Caption = "&Add"

Height = 375

HelpContextID = 1050

Left = 1800

TabIndex = 2

Top = 1320

Width = 975

End

Begin VB.ListBox lstNames

Height = 1035

HelpContextID = 1050

Left = 120

TabIndex = 5

Top = 240

Width = 5655

End

End

Begin VB.Frame fraMessage

Caption = "Messages"

ForeColor = &H80000008&

Height = 2775

Left = 120

TabIndex = 9

Top = 1800

Width = 5895

Begin VB.ListBox lstCalcStat

BeginProperty Font

Name = "Courier New"

Size = 8.25

Charset = 0

Weight = 400

Underline = 0 'False

Italic = 0 'False

Strikethrough = 0 'False

EndProperty

Height = 2370

Left = 120

TabIndex = 4

Top = 240

Width = 5655

End

End

Begin VB.CommandButton cmdClose

Cancel = -1 'True

Caption = "&Close"

Height = 375

Left = 3120

TabIndex = 1

Top = 6240

Width = 855

End

Begin VB.CommandButton cmdStartStop

Caption = "&Start"

Height = 375

Left = 2160

TabIndex = 0

Top = 6240

Width = 855

End

Begin VB.Frame fraStatus

Caption = "Status"

ForeColor = &H80000008&

Height = 1575

Left = 120

TabIndex = 6

Top = 4560

Width = 5895

Begin VB.PictureBox picTherm

AutoRedraw = -1 'True

DrawMode = 14 'Copy Pen

Height = 255

Left = 360

ScaleHeight = 195

ScaleWidth = 4995

TabIndex = 8

Top = 240

Width = 5055

End

Begin VB.Label lblStatusMessage

Alignment = 2 'Center

Caption = "Status message 2"

ForeColor = &H80000008&

Height = 495

Index = 1

Left = 120

TabIndex = 11

Top = 960

Width = 5655

WordWrap = -1 'True

End

Begin VB.Label lblStatusMessage

Alignment = 2 'Center

Caption = "Status message 1"

ForeColor = &H80000008&

Height = 255

Index = 0

Left = 120

TabIndex = 7

Top = 600

Width = 5655

End

End

Begin VB.Timer Timer1

Enabled = 0 'False

Left = 0

Top = 480

End

Begin MSComDlg.CommonDialog CMDialog1

Left = 0

Top = 0

\_ExtentX = 847

\_ExtentY = 847

\_Version = 393216

End

End

Attribute VB\_Name = "frmCalc"

Attribute VB\_GlobalNameSpace = False

Attribute VB\_Creatable = False

Attribute VB\_PredeclaredId = True

Attribute VB\_Exposed = False

' $Id: CALC.FRM,v 1.10 2016/09/13 13:19:56 tom Exp $

'Calculations form

'

'Tag property: Ouput: Status flag: "ok"=success

' "fail"=problems

'

Dim StartDate As Variant

Private Sub BatchCalcs()

'Perform calculations on all the files listed

Dim strErrLocation As String

On Error GoTo Error\_Handler

'turn off Add/Remove buttons

cmdAdd.Enabled = False

cmdRemove.Enabled = False

'cycle through all the files

AddToLog lstCalcStat, "Batch calculations starting..."

For i = 0 To lstNames.ListCount - 1

lstNames.ListIndex = i

UI.FileName = lstNames.List(i)

AddToLog lstCalcStat, "File: " & UI.FileName

If UserDataRead(UI.FileName, UD, UC, False) Then

UI.DataNeedsChecking = True

ClearUserCalc UC 'clear out previous calcs

calcstat = Calculate()

'check for user-requested halt

If calcstat = 2 Then

AddToLog lstCalcStat, "Batch calculations halted."

cmdAdd.Enabled = True

cmdRemove.Enabled = True

Exit Sub

End If

'Calcs went okay, save the file

If UserDataWrite(UI.FileName, UD, UC, False) Then

UpdateDataChangedFlag False 'we just saved the data

Else

AddToLog lstCalcStat, "Error writing file."

End If

Else

AddToLog lstCalcStat, "Error reading file."

End If

Next

AddToLog lstCalcStat, "Batch calculations complete."

'turn on Add/Remove buttons

cmdAdd.Enabled = True

cmdRemove.Enabled = True

'====================================================

'Exit Point for BatchCalcs

'====================================================

Exit\_BatchCalcs:

Exit Sub

'====================================================

' ERROR HANDLER ROUTINE(S)

'====================================================

Error\_Handler:

gobjErrors.Append Err, "BatchCalcs", "frmCalc", strErrLocation

gobjErrors.UserMessage

gobjErrors.WriteToErrorLog

gobjErrors.Clear

Resume Exit\_BatchCalcs

End Sub

Private Function Calculate() As Integer

'Perform the AGDISP calculations on the current data

'Returns:

' 0 = success

' 1 = errors

' 2 = halted

'

'This routine calls the fortran code in the DLL

'that performs the actual calculations

Dim strErrLocation As String

On Error GoTo Error\_Handler

Dim Msg As String

Dim fn As String

Dim ndiam As Long

Dim ntherm As Integer

ReDim Diam(2 \* MAX\_DROPS - 1) As Single

ReDim Compl(2 \* MAX\_DROPS - 1) As Single

Dim NXY As Long

Dim X(MAX\_CALCDATA - 1) As Single

Dim Y(MAX\_CALCDATA - 1) As Single

Dim dum As Single

' Change the form mouse pointer

Me.MousePointer = vbHourglass 'hourglass

UI.CalcsInProgress = True 'the calcs have begun!

cmdStartStop.Caption = "&Stop"

cmdStartStop.Enabled = True

lblStatusMessage(0).Caption = "Starting calculations..."

'Reset PlotVar so that settings from the prev calcs do not linger

UI.PlotVar = 0

'record the date and time locally

StartDate = Now

'Enable the elapsed timer

Timer1.Interval = 1000 'milliseconds

Timer1.Enabled = True

'reset the calc flag here because we are about to

'call the fortran routines and alter their COMMON

'(the DataChanged flag is set further down)

UC.Valid = False

UC.CalpuffCalcsAvailable = False

'If this flag ever goes false, stop calculating

UI.OkToDoCalcs = True

'fill in the header data

UC.CodeVersion = AGDISPVERSION

UC.StartDate = Format$(StartDate, "mm-dd-yyyy")

UC.StartTime = Format$(StartDate, "hh:mm:ss")

'Time stamp the calc log

Msg = ""

AppendStr Msg, UC.StartDate & " " & UC.StartTime, False

AppendStr Msg, " Calculations starting", False

AddToLog lstCalcStat, Msg

AppendStr UC.MessageLog, Msg, True

'Check the data

If UI.DataNeedsChecking Then

If Not CheckData(lstCalcStat) Then GoTo CalculateError

End If

'Process all the drop categories

'The compl array tells how much work has been done after

'each drop category is complete.

DoEvents: If Not UI.OkToDoCalcs Then GoTo CalculateHalt

Call aglims(ndiam, Diam(0), Compl(0)) 'get computed drop dist

UpdateTherm 0, 1 'Init thermometer

For i = 0 To ndiam - 1

lblStatusMessage(0).Caption = "Initial Drop Size: " & AGFormat$(Diam(i)) & " �m"

DoEvents: If Not UI.OkToDoCalcs Then GoTo CalculateHalt

Call agdrop(i + 1)

UpdateTherm Compl(i), Compl(ndiam - 1)

Next

'we are about to change the saved calculation data, so

'set the data changed flag.

DoEvents: If Not UI.OkToDoCalcs Then GoTo CalculateHalt

UpdateDataChangedFlag True

lblStatusMessage(0).Caption = "Completing calculations..."

'recover various results

'note: call agends ONLY ONCE for each flag value!

'use agends to retrieve depos

Call agends(AGENDS\_DEPOS, NXY, X(0), Y(0)) 'depos

UC.NumDep = NXY

ReDim UC.DepDist(NXY)

ReDim UC.DepVal(NXY)

CopyMemory UC.DepDist(0), X(0), Len(X(0)) \* NXY

CopyMemory UC.DepVal(0), Y(0), Len(Y(0)) \* NXY

DoEvents: If Not UI.OkToDoCalcs Then GoTo CalculateHalt

'use agends to retrieve number depos

Call agends(AGENDS\_DEPDROPS, NXY, X(0), Y(0)) 'number depos

UC.NumDrp = NXY

ReDim UC.DrpDist(NXY)

ReDim UC.DrpVal(NXY)

CopyMemory UC.DrpDist(0), X(0), Len(X(0)) \* NXY

CopyMemory UC.DrpVal(0), Y(0), Len(Y(0)) \* NXY

DoEvents: If Not UI.OkToDoCalcs Then GoTo CalculateHalt

'use agends to retrieve pond-integrated depos

Call agends(AGENDS\_PID, NXY, X(0), Y(0)) 'pond-int depos

UC.NumPID = NXY

ReDim UC.PIDDist(NXY)

ReDim UC.PIDVal(NXY)

CopyMemory UC.PIDDist(0), X(0), Len(X(0)) \* NXY

CopyMemory UC.PIDVal(0), Y(0), Len(Y(0)) \* NXY

DoEvents: If Not UI.OkToDoCalcs Then GoTo CalculateHalt

'use agends to retrieve vertical flux

Call agends(AGENDS\_FLUX, NXY, X(0), Y(0)) 'vert depos

UC.NumFlux = NXY

ReDim UC.FluxDist(NXY)

ReDim UC.FluxVal(NXY)

CopyMemory UC.FluxDist(0), X(0), Len(X(0)) \* NXY

CopyMemory UC.FluxVal(0), Y(0), Len(Y(0)) \* NXY

DoEvents: If Not UI.OkToDoCalcs Then GoTo CalculateHalt

'use agends to retrieve Concentration

Call agends(AGENDS\_1HRCON, NXY, X(0), Y(0)) 'Concentration

UC.NumConc = NXY

ReDim UC.ConcDist(NXY)

ReDim UC.ConcVal(NXY)

CopyMemory UC.ConcDist(0), X(0), Len(X(0)) \* NXY

CopyMemory UC.ConcVal(0), Y(0), Len(Y(0)) \* NXY

DoEvents: If Not UI.OkToDoCalcs Then GoTo CalculateHalt

'use agends to retrieve COV/ESW

Call agends(AGENDS\_COV, NXY, X(0), Y(0)) 'COV

UC.NumCOV = NXY

ReDim UC.COVESW(NXY)

ReDim UC.COVVal(NXY)

CopyMemory UC.COVESW(0), X(0), Len(X(0)) \* NXY

CopyMemory UC.COVVal(0), Y(0), Len(Y(0)) \* NXY

DoEvents: If Not UI.OkToDoCalcs Then GoTo CalculateHalt

'use agends to retrieve COV mean deposition

Call agends(AGENDS\_MEAN, NXY, X(0), Y(0)) 'COV

UC.NumCOVM = NXY

ReDim UC.COVMDist(NXY)

ReDim UC.COVMVal(NXY)

CopyMemory UC.COVMDist(0), X(0), Len(X(0)) \* NXY

CopyMemory UC.COVMVal(0), Y(0), Len(Y(0)) \* NXY

DoEvents: If Not UI.OkToDoCalcs Then GoTo CalculateHalt

'use agends to retrieve fraction aloft

Call agends(AGENDS\_ALOFT, NXY, X(0), Y(0)) 'fraction aloft

UC.NumFA = NXY

ReDim UC.FADist(NXY)

ReDim UC.FAVal(NXY)

CopyMemory UC.FADist(0), X(0), Len(X(0)) \* NXY

CopyMemory UC.FAVal(0), Y(0), Len(Y(0)) \* NXY

DoEvents: If Not UI.OkToDoCalcs Then GoTo CalculateHalt

'use agends to retrieve single-swath depos

Call agends(AGENDS\_SGLDEP, NXY, X(0), Y(0)) 'single-swath depos

UC.NumSgl = NXY

ReDim UC.SglDist(NXY)

ReDim UC.SglVal(NXY)

CopyMemory UC.SglDist(0), X(0), Len(X(0)) \* NXY

CopyMemory UC.SglVal(0), Y(0), Len(Y(0)) \* NXY

DoEvents: If Not UI.OkToDoCalcs Then GoTo CalculateHalt

'use agends to retrieve single swath upwind half boom

Call agends(AGENDS\_SGLHAF, NXY, X(0), Y(0)) 'half boom

UC.NumHalf = NXY

ReDim UC.HalfDist(NXY)

ReDim UC.HalfVal(NXY)

CopyMemory UC.HalfDist(0), X(0), Len(X(0)) \* NXY

CopyMemory UC.HalfVal(0), Y(0), Len(Y(0)) \* NXY

DoEvents: If Not UI.OkToDoCalcs Then GoTo CalculateHalt

'use agends to retrieve spray block deposition

Call agends(AGENDS\_SBLOCK, NXY, X(0), Y(0))

UC.NumSBD = NXY

ReDim UC.SBDDist(NXY)

ReDim UC.SBDVal(NXY)

CopyMemory UC.SBDDist(0), X(0), Len(X(0)) \* NXY

CopyMemory UC.SBDVal(0), Y(0), Len(Y(0)) \* NXY

DoEvents: If Not UI.OkToDoCalcs Then GoTo CalculateHalt

'use agends to retrieve canopy deposition

Call agends(AGENDS\_CANOPY, NXY, X(0), Y(0))

UC.NumCAN = NXY

ReDim UC.CANDist(NXY)

ReDim UC.CANVal(NXY)

CopyMemory UC.CANDist(0), X(0), Len(X(0)) \* NXY

CopyMemory UC.CANVal(0), Y(0), Len(Y(0)) \* NXY

DoEvents: If Not UI.OkToDoCalcs Then GoTo CalculateHalt

'use agends to retrieve time accountancy aloft

Call agends(AGENDS\_TAALOFT, NXY, X(0), Y(0))

UC.NumTAA = NXY

ReDim UC.TAATime(NXY)

ReDim UC.TAAVal(NXY)

CopyMemory UC.TAATime(0), X(0), Len(X(0)) \* NXY

CopyMemory UC.TAAVal(0), Y(0), Len(Y(0)) \* NXY

DoEvents: If Not UI.OkToDoCalcs Then GoTo CalculateHalt

'use agends to retrieve time accountancy vapor

Call agends(AGENDS\_TAVAPOR, NXY, X(0), Y(0))

UC.NumTAV = NXY

ReDim UC.TAVTime(NXY)

ReDim UC.TAVVal(NXY)

CopyMemory UC.TAVTime(0), X(0), Len(X(0)) \* NXY

CopyMemory UC.TAVVal(0), Y(0), Len(Y(0)) \* NXY

DoEvents: If Not UI.OkToDoCalcs Then GoTo CalculateHalt

'use agends to retrieve time accountancy canopy

Call agends(AGENDS\_TACANOPY, NXY, X(0), Y(0))

UC.NumTAC = NXY

ReDim UC.TACTime(NXY)

ReDim UC.TACVal(NXY)

CopyMemory UC.TACTime(0), X(0), Len(X(0)) \* NXY

CopyMemory UC.TACVal(0), Y(0), Len(Y(0)) \* NXY

DoEvents: If Not UI.OkToDoCalcs Then GoTo CalculateHalt

'use agends to retrieve time accountancy ground

Call agends(AGENDS\_TAGROUND, NXY, X(0), Y(0))

UC.NumTAG = NXY

ReDim UC.TAGTime(NXY)

ReDim UC.TAGVal(NXY)

CopyMemory UC.TAGTime(0), X(0), Len(X(0)) \* NXY

CopyMemory UC.TAGVal(0), Y(0), Len(Y(0)) \* NXY

DoEvents: If Not UI.OkToDoCalcs Then GoTo CalculateHalt

'use agends to retrieve height accountancy aloft

Call agends(AGENDS\_HAALOFT, NXY, X(0), Y(0))

UC.NumHAA = NXY

ReDim UC.HAAVal(NXY)

ReDim UC.HAAHgt(NXY)

CopyMemory UC.HAAHgt(0), X(0), Len(X(0)) \* NXY

CopyMemory UC.HAAVal(0), Y(0), Len(Y(0)) \* NXY

DoEvents: If Not UI.OkToDoCalcs Then GoTo CalculateHalt

'use agends to retrieve height accountancy vapor

Call agends(AGENDS\_HAVAPOR, NXY, X(0), Y(0))

UC.NumHAV = NXY

ReDim UC.HAVVal(NXY)

ReDim UC.HAVHgt(NXY)

CopyMemory UC.HAVHgt(0), X(0), Len(X(0)) \* NXY

CopyMemory UC.HAVVal(0), Y(0), Len(Y(0)) \* NXY

DoEvents: If Not UI.OkToDoCalcs Then GoTo CalculateHalt

'use agends to retrieve height accountancy canopy

Call agends(AGENDS\_HACANOPY, NXY, X(0), Y(0))

UC.NumHAC = NXY

ReDim UC.HACVal(NXY)

ReDim UC.HACHgt(NXY)

CopyMemory UC.HACHgt(0), X(0), Len(X(0)) \* NXY

CopyMemory UC.HACVal(0), Y(0), Len(Y(0)) \* NXY

DoEvents: If Not UI.OkToDoCalcs Then GoTo CalculateHalt

'use agends to retrieve spray block dsd

Call agends(AGENDS\_SBLOCKDSD, NXY, X(0), Y(0))

UC.NumSBDSD = NXY

ReDim UC.SBDSDDiam(NXY)

ReDim UC.SBDSDFrac(NXY)

CopyMemory UC.SBDSDDiam(0), X(0), Len(X(0)) \* NXY

CopyMemory UC.SBDSDFrac(0), Y(0), Len(Y(0)) \* NXY

DoEvents: If Not UI.OkToDoCalcs Then GoTo CalculateHalt

'use agends to retrieve downwind dsd

Call agends(AGENDS\_DWINDDSD, NXY, X(0), Y(0))

UC.NumDWDSD = NXY

ReDim UC.DWDSDDiam(NXY)

ReDim UC.DWDSDFrac(NXY)

CopyMemory UC.DWDSDDiam(0), X(0), Len(X(0)) \* NXY

CopyMemory UC.DWDSDFrac(0), Y(0), Len(Y(0)) \* NXY

DoEvents: If Not UI.OkToDoCalcs Then GoTo CalculateHalt

'use agends to retrieve vertical flux dsd

Call agends(AGENDS\_FLUXDSD, NXY, X(0), Y(0))

UC.NumFXDSD = NXY

ReDim UC.FXDSDDiam(NXY)

ReDim UC.FXDSDFrac(NXY)

CopyMemory UC.FXDSDDiam(0), X(0), Len(X(0)) \* NXY

CopyMemory UC.FXDSDFrac(0), Y(0), Len(Y(0)) \* NXY

DoEvents: If Not UI.OkToDoCalcs Then GoTo CalculateHalt

'use agends to retrieve distance accountancy aloft

Call agends(AGENDS\_DAALOFT, NXY, X(0), Y(0))

UC.NumDAA = NXY

ReDim UC.DAADist(NXY)

ReDim UC.DAAVal(NXY)

CopyMemory UC.DAADist(0), X(0), Len(X(0)) \* NXY

CopyMemory UC.DAAVal(0), Y(0), Len(Y(0)) \* NXY

DoEvents: If Not UI.OkToDoCalcs Then GoTo CalculateHalt

'use agends to retrieve distance accountancy vapor

Call agends(AGENDS\_DAVAPOR, NXY, X(0), Y(0))

UC.NumDAV = NXY

ReDim UC.DAVDist(NXY)

ReDim UC.DAVVal(NXY)

CopyMemory UC.DAVDist(0), X(0), Len(X(0)) \* NXY

CopyMemory UC.DAVVal(0), Y(0), Len(Y(0)) \* NXY

DoEvents: If Not UI.OkToDoCalcs Then GoTo CalculateHalt

'use agends to retrieve distance accountancy canopy

Call agends(AGENDS\_DACANOPY, NXY, X(0), Y(0))

UC.NumDAC = NXY

ReDim UC.DACDist(NXY)

ReDim UC.DACVal(NXY)

CopyMemory UC.DACDist(0), X(0), Len(X(0)) \* NXY

CopyMemory UC.DACVal(0), Y(0), Len(Y(0)) \* NXY

DoEvents: If Not UI.OkToDoCalcs Then GoTo CalculateHalt

'use agends to retrieve distance accountancy ground

Call agends(AGENDS\_DAGROUND, NXY, X(0), Y(0))

UC.NumDAG = NXY

ReDim UC.DAGDist(NXY)

ReDim UC.DAGVal(NXY)

CopyMemory UC.DAGDist(0), X(0), Len(X(0)) \* NXY

CopyMemory UC.DAGVal(0), Y(0), Len(Y(0)) \* NXY

DoEvents: If Not UI.OkToDoCalcs Then GoTo CalculateHalt

'use agends to retrieve canopy dsd

Call agends(AGENDS\_CANDSD, NXY, X(0), Y(0))

UC.NumCNDSD = NXY

ReDim UC.CNDSDDiam(NXY)

ReDim UC.CNDSDFrac(NXY)

CopyMemory UC.CNDSDDiam(0), X(0), Len(X(0)) \* NXY

CopyMemory UC.CNDSDFrac(0), Y(0), Len(Y(0)) \* NXY

DoEvents: If Not UI.OkToDoCalcs Then GoTo CalculateHalt

'use agends to retrieve spray block area coverage

Call agends(AGENDS\_SBCOVER, NXY, X(0), Y(0))

UC.NumSBAC = NXY

ReDim UC.SBACRate(NXY)

ReDim UC.SBACFrac(NXY)

CopyMemory UC.SBACRate(0), X(0), Len(X(0)) \* NXY

CopyMemory UC.SBACFrac(0), Y(0), Len(Y(0)) \* NXY

DoEvents: If Not UI.OkToDoCalcs Then GoTo CalculateHalt

'use agends to retrieve application layout

Call agends(AGENDS\_LAYOUT, NXY, X(0), Y(0))

UC.NumLAY = NXY

ReDim UC.LAYDist(NXY)

ReDim UC.LAYFrac(NXY)

CopyMemory UC.LAYDist(0), X(0), Len(X(0)) \* NXY

CopyMemory UC.LAYFrac(0), Y(0), Len(Y(0)) \* NXY

DoEvents: If Not UI.OkToDoCalcs Then GoTo CalculateHalt

'use agends to retrieve point DSD

Call agends(AGENDS\_POINTDSD, NXY, X(0), Y(0))

UC.NumPTDSD = NXY

ReDim UC.PTDSDDiam(NXY)

ReDim UC.PTDSDFrac(NXY)

CopyMemory UC.PTDSDDiam(0), X(0), Len(X(0)) \* NXY

CopyMemory UC.PTDSDFrac(0), Y(0), Len(Y(0)) \* NXY

DoEvents: If Not UI.OkToDoCalcs Then GoTo CalculateHalt

'call agreps to update the spray line reps

Call agreps(UD.CTL.NumLines, UD.CTL.LineReps(0))

'call agnums to get the numerics

Call agnums(UC.SwathDisp, UC.SBCOV, UC.SBMeanDep, UC.AppEff, \_

UC.DownwindDep, UC.AirborneDrift, UC.EvapFrac, UC.CanopyDep)

'Check Swath Displacement

If UC.EvapFrac < 0 Then

Msg = ""

AppendStr Msg, "Warning!", True

AppendStr Msg, "Swath Displacement could not be calculated.", False

'Add message to calc log

AddToLog lstCalcStat, Msg

AppendStr UC.MessageLog, Msg, True

'Let the user know about it

If Not UI.CalcsBatchMode Then

MsgBox Msg, vbCritical + vbOKOnly

End If

End If

GoTo CalculateSuccess

'Success! reset the calc flags and set return value

CalculateSuccess:

UC.Valid = True

UI.CalcsInProgress = False

'save these new calculations in the revert file

If Not UI.CalcsBatchMode Then

fn = App.Path & Chr$(92) & App.EXEName & ".rvt"

If UserDataWrite(fn, UD, UC, False) Then UI.RevertCalcsAvailable = True

End If

'update the CALPUFF flag. If we just did calcs with

'CALPUFF saving enabled, set a flag

If UD.CALPUFFFLAG Then UC.CalpuffCalcsAvailable = True

'Log the completion

Msg = ""

AppendStr Msg, Date$ & " " & Time$, False

AppendStr Msg, " Calculations complete", False

AddToLog lstCalcStat, Msg

AppendStr UC.MessageLog, Msg, True

lblStatusMessage(0).Caption = "Calculations complete."

picTherm.Cls 'clear the thermometer bar

Me.MousePointer = vbDefault 'reset the mouse pointer

Timer1.Enabled = False

Me.Tag = "ok"

cmdStartStop.Caption = "&Start"

cmdStartStop.Enabled = False 'Prevent accidental restart

Calculate = 0

Exit Function

CalculateError:

'stop the calculations because the checking routine an error

UI.CalcsInProgress = False

Msg = ""

AppendStr Msg, Date$ & " " & Time$, False

AppendStr Msg, " Calculations halted due to input errors", False

AddToLog lstCalcStat, Msg

AppendStr UC.MessageLog, Msg, True

picTherm.Cls 'clear the thermometer bar

lblStatusMessage(0).Caption = "Calculations halted due to input errors."

Me.MousePointer = vbDefault

Timer1.Enabled = False

Me.Tag = "fail"

cmdStartStop.Caption = "&Start"

Calculate = 1

GoTo Exit\_Calculate

CalculateHalt:

'stop the calculations and exit

UI.CalcsInProgress = False

Msg = ""

AppendStr Msg, Date$ & " " & Time$, False

AppendStr Msg, " Calculations halted", False

AddToLog lstCalcStat, Msg

AppendStr UC.MessageLog, Msg, True

picTherm.Cls 'clear the thermometer bar

lblStatusMessage(0).Caption = "Calculations halted."

Me.MousePointer = vbDefault

Timer1.Enabled = False

Me.Tag = "fail"

cmdStartStop.Caption = "&Start"

Calculate = 2

GoTo Exit\_Calculate

'====================================================

'Exit Point for Calculate

'====================================================

Exit\_Calculate:

Exit Function

'====================================================

' ERROR HANDLER ROUTINE(S)

'====================================================

Error\_Handler:

gobjErrors.Append Err, "Calculate", "frmCalc", strErrLocation

gobjErrors.UserMessage

gobjErrors.WriteToErrorLog

gobjErrors.Clear

Resume Exit\_Calculate

End Function

Private Sub cmdAdd\_Click()

'add a file to the list box

Dim strErrLocation As String

On Error GoTo Error\_Handler

Dim fn As String

If PickFile(fn) Then AddToLog lstNames, fn

'====================================================

'Exit Point for cmdAdd\_Click

'====================================================

Exit\_cmdAdd\_Click:

Exit Sub

'====================================================

' ERROR HANDLER ROUTINE(S)

'====================================================

Error\_Handler:

gobjErrors.Append Err, "cmdAdd\_Click", "frmCalc", strErrLocation

gobjErrors.UserMessage

gobjErrors.WriteToErrorLog

gobjErrors.Clear

Resume Exit\_cmdAdd\_Click

End Sub

Private Sub cmdClose\_Click()

'halt any calculations and hide the form

Dim strErrLocation As String

On Error GoTo Error\_Handler

UI.OkToDoCalcs = False

Me.Hide

'====================================================

'Exit Point for cmdClose\_Click

'====================================================

Exit\_cmdClose\_Click:

Exit Sub

'====================================================

' ERROR HANDLER ROUTINE(S)

'====================================================

Error\_Handler:

gobjErrors.Append Err, "cmdClose\_Click", "frmCalc", strErrLocation

gobjErrors.UserMessage

gobjErrors.WriteToErrorLog

gobjErrors.Clear

Resume Exit\_cmdClose\_Click

End Sub

Private Sub cmdRemove\_Click()

'remove a name from the list

Dim strErrLocation As String

On Error GoTo Error\_Handler

If lstNames.ListCount > 0 And lstNames.ListIndex >= 0 Then

lstNames.RemoveItem lstNames.ListIndex

End If

'====================================================

'Exit Point for cmdRemove\_Click

'====================================================

Exit\_cmdRemove\_Click:

Exit Sub

'====================================================

' ERROR HANDLER ROUTINE(S)

'====================================================

Error\_Handler:

gobjErrors.Append Err, "cmdRemove\_Click", "frmCalc", strErrLocation

gobjErrors.UserMessage

gobjErrors.WriteToErrorLog

gobjErrors.Clear

Resume Exit\_cmdRemove\_Click

End Sub

Private Sub cmdStartStop\_Click()

'start calcs if stopped, stop them if started

Dim strErrLocation As String

On Error GoTo Error\_Handler

If UI.CalcsInProgress Then '"Stop" button

UI.OkToDoCalcs = False

Else '"Start" button

If UI.CalcsBatchMode Then

BatchCalcs

Else

dum% = Calculate()

End If

End If

'====================================================

'Exit Point for cmdStartStop\_Click

'====================================================

Exit\_cmdStartStop\_Click:

Exit Sub

'====================================================

' ERROR HANDLER ROUTINE(S)

'====================================================

Error\_Handler:

gobjErrors.Append Err, "cmdStartStop\_Click", "frmCalc", strErrLocation

gobjErrors.UserMessage

gobjErrors.WriteToErrorLog

gobjErrors.Clear

Resume Exit\_cmdStartStop\_Click

End Sub

Private Sub Form\_Activate()

Dim strErrLocation As String

On Error GoTo Error\_Handler

If Not UI.CalcsInProgress Then

'general calculation setup

UI.DataNeedsChecking = True 'init the checking flag

lstCalcStat.Clear 'clear the display control

lblStatusMessage(1).Caption = ""

' "Note: Screen savers may adversely affect calculation performance."

'do different things for batch and immediate modes

Select Case UI.CalcsBatchMode

Case True 'batch mode

lblStatusMessage(0).Caption = \_

"Select files to process and press Start."

Case False 'non-batch

'start the calcs if required, or just do a check, if not

ClearUserCalc UC 'clear out previous calcs

If UI.StartCalcsOnLoad Then

dum% = Calculate()

Me.Hide

Else

dum% = CheckData(lstCalcStat)

lblStatusMessage(0).Caption = "Press Start to begin calculations."

End If

End Select

End If

'====================================================

'Exit Point for Form\_Activate

'====================================================

Exit\_Form\_Activate:

Exit Sub

'====================================================

' ERROR HANDLER ROUTINE(S)

'====================================================

Error\_Handler:

gobjErrors.Append Err, "Form\_Activate", "frmCalc", strErrLocation

gobjErrors.UserMessage

gobjErrors.WriteToErrorLog

gobjErrors.Clear

Resume Exit\_Form\_Activate

End Sub

Private Sub Form\_Load()

'initialize this form

Dim strErrLocation As String

On Error GoTo Error\_Handler

Const HELPID\_RUN = 1250

Const HELPID\_BATCH = 1050

Me.Tag = "fail" 'set default return status

'Size the form according to the mode

Select Case UI.CalcsBatchMode

Case True

fraFiles.Visible = True

fraFiles.Top = 120

fraFiles.Left = 120

fraMessage.Top = fraFiles.Top + fraFiles.Height

fraMessage.Left = 120

lstCalcStat.HelpContextID = HELPID\_BATCH

cmdStartStop.HelpContextID = HELPID\_BATCH

cmdClose.HelpContextID = HELPID\_BATCH

Case False

fraFiles.Visible = False

fraFiles.Top = 0

fraFiles.Left = 0

fraMessage.Top = 120

fraMessage.Left = 120

lstCalcStat.HelpContextID = HELPID\_RUN

cmdStartStop.HelpContextID = HELPID\_RUN

cmdClose.HelpContextID = HELPID\_RUN

End Select

fraStatus.Top = fraMessage.Top + fraMessage.Height

fraStatus.Left = 120

cmdStartStop.Top = fraStatus.Top + fraStatus.Height + 120

cmdStartStop.Left = Me.ScaleWidth \ 2 - cmdStartStop.Width - 60

cmdClose.Top = cmdStartStop.Top

cmdClose.Left = Me.ScaleWidth \ 2 + 60

'Size the form according to the Start/Stop button

HeightDiff = Me.Height - Me.ScaleHeight

Me.Height = HeightDiff + cmdStartStop.Top + cmdStartStop.Height + 120

CenterForm Me 'center the form

lblStatusMessage(0).Caption = "" 'clear out the calc message

lblStatusMessage(1).Caption = ""

'====================================================

'Exit Point for Form\_Load

'====================================================

Exit\_Form\_Load:

Exit Sub

'====================================================

' ERROR HANDLER ROUTINE(S)

'====================================================

Error\_Handler:

gobjErrors.Append Err, "Form\_Load", "frmCalc", strErrLocation

gobjErrors.UserMessage

gobjErrors.WriteToErrorLog

gobjErrors.Clear

Resume Exit\_Form\_Load

End Sub

Private Sub Form\_Unload(Cancel As Integer)

'halt any calculations

Dim strErrLocation As String

On Error GoTo Error\_Handler

UI.OkToDoCalcs = False

'====================================================

'Exit Point for Form\_Unload

'====================================================

Exit\_Form\_Unload:

Exit Sub

'====================================================

' ERROR HANDLER ROUTINE(S)

'====================================================

Error\_Handler:

gobjErrors.Append Err, "Form\_Unload", "frmCalc", strErrLocation

gobjErrors.UserMessage

gobjErrors.WriteToErrorLog

gobjErrors.Clear

Resume Exit\_Form\_Unload

End Sub

Private Function PickFile(fn As String) As Integer

'Dialog box for obtaining a file name

' -returns true on OK, false on Cancel

'

Dim strErrLocation As String

On Error GoTo Error\_Handler

'Set default return value

PickFile = False

'Turn on CancelError

CMDialog1.CancelError = True

'Set Default Extension

'added if a file name is entered without an extension

CMDialog1.DefaultExt = "agd"

'Set filter list

CMDialog1.Filter = "All Files (\*.\*)|\*.\*|Data Files (\*.ag)|\*.ag"

'Specify current filter

CMDialog1.FilterIndex = 2

'Set the default file name

'CMDialog1.FileName =

'Set dialog flags

CMDialog1.Flags = cdlOFNHideReadOnly

'Display the dialog box

'1=Open 2=Save As

CMDialog1.Action = 1 '"Open" dialog

'full file path is CMDialog1.FileName

'file name only is CMDialog1.FileTitle

fn = CMDialog1.FileName

PickFile = True

'====================================================

'Exit Point for PickFile

'====================================================

Exit\_PickFile:

Exit Function

'====================================================

' ERROR HANDLER ROUTINE(S)

'====================================================

Error\_Handler:

Select Case Err.Number

Case 32755 'user selected cancel

PickFile = False

Case Else

gobjErrors.Append Err, "PickFile", "frmCalc", strErrLocation

gobjErrors.UserMessage

gobjErrors.WriteToErrorLog

gobjErrors.Clear

End Select

Resume Exit\_PickFile

End Function

Private Sub Timer1\_Timer()

Dim strErrLocation As String

On Error GoTo Error\_Handler

lblStatusMessage(1).Caption = "Elapsed Time: " & Format$(CDbl(Now) - CDbl(StartDate), "hh:mm:ss")

'====================================================

'Exit Point for Timer1\_Timer

'====================================================

Exit\_Timer1\_Timer:

Exit Sub

'====================================================

' ERROR HANDLER ROUTINE(S)

'====================================================

Error\_Handler:

gobjErrors.Append Err, "Timer1\_Timer", "frmCalc", strErrLocation

gobjErrors.UserMessage

gobjErrors.WriteToErrorLog

gobjErrors.Clear

Resume Exit\_Timer1\_Timer

End Sub

Private Sub UpdateTherm(curr\_val As Single, max\_val As Single)

'Update the Percent Completed Thermometer bar

Dim strErrLocation As String

On Error GoTo Error\_Handler

If max\_val = 0 Then

frac = 0

Else

frac = curr\_val / max\_val

End If

s$ = Format$(Int(frac \* 100 + 0.5)) + "%"

X = picTherm.Width \* frac

picTherm.Cls

picTherm.CurrentX = (picTherm.Width - picTherm.TextWidth(s$)) / 2

picTherm.CurrentY = (picTherm.Height - picTherm.TextHeight(s$)) / 2

picTherm.Print s$

picTherm.Line (0, 0)-(X, picTherm.Height), RGB(0, 135, 0), BF

picTherm.Refresh

'

' If the window is iconized, treat the whole form as a thermometer bar

If Me.WindowState = 1 Then

X = Me.Width \* frac

Me.Cls

Me.CurrentX = (Me.Width - Me.TextWidth(s$)) / 2

Me.CurrentY = (Me.Height - Me.TextHeight(s$)) / 2

SaveDrawMode = Me.DrawMode

Me.DrawMode = 14 'Merge Pen Not

Me.Print s$

Me.Line (0, 0)-(X, Me.Height), QBColor(12), BF

Me.DrawMode = SaveDrawMode

DoEvents ' Allow other events.

End If

'====================================================

'Exit Point for UpdateTherm

'====================================================

Exit\_UpdateTherm:

Exit Sub

'====================================================

' ERROR HANDLER ROUTINE(S)

'====================================================

Error\_Handler:

gobjErrors.Append Err, "UpdateTherm", "frmCalc", strErrLocation

gobjErrors.UserMessage

gobjErrors.WriteToErrorLog

gobjErrors.Clear

Resume Exit\_UpdateTherm

End Sub