APPENDIX

SUGARCANE FLOWERING MODEL SOURCE CODE

Main Form (Screen 1)

```
'Determine the Constant Value
Private Const Tbaseemerge As Single = 8 'Tbase for Planting to Emergence
Private Const Tbaseleaf As Single = 16     'Tbase for Leaf development
Private Const Tbasepi As Single = 16       'Tbase for panicle induction
Private Const Tbasepe As Single = 16       'Tbase for panicle emergence
Private Const Tbasei As Single = 16           'Tbase for calculate daily GDD after PE


'Click the Simulation button
Private Sub cmdSim_Click()
'Check "Are the required values ready?"

'First, "Are the management date ready"
If Pyyjdate = Empty Or Hyyjdate = Empty Then
    MsgBox "You must determine planting date and harvesting date of the crop", _
    0, "Missing field"
ElseIf SDEPTH = Empty And CaneType = 1 Then
    MsgBox "Sowing depth in centimetre are required", 0, "Missing field"

'Is a variety be selected? and 'Does it contain all require value?
ElseIf FoundVar = False Or FoundVarPar = False Then
    MsgBox "A variety may not be selected.", 0, "Missing field"

'Is weather data of the determinated duration ready?
ElseIf FoundLat = False Or FoundWthdate = False Then
    MsgBox "Your weather file may miss a required value.", 0, "Missing field"

'Is an output file name be selected?
ElseIf Outputfile = Empty Then
    MsgBox "You must select or create a file name as output file.", 0, "Missing field"

'Is a DLmodel and a CaneType be selected?
ElseIf DLmodel = 0 Or CaneType = 0 Then
    MsgBox "You must select an simulation option.", 0, "Missing field"
ElseIf DLmodel = 2 And Not FoundDegree = True Then
    MsgBox "Daylength definition may missing.", 0, "Missing field"

Else
    'After all require values are ready then
    'Write the selected file and path of selected variety,
    'weather file and output file.
    Open App.Path & "\" & "FilePath.Txt" For Output As #2
    Print #2, SelVarfile
    Print #2, WthDirSelect
    Print #2, Outputpath
    Close #2

    'Write the weater station information and the variety information
    Open Outputfile For Output As #1
    Print #1, "*Flowering : "  ' Print text to file.
    Print #1,    ' Print blank line to file.
    Print #1,    ' Print blank line to file.
    Print #1, headerweather         'print header of weather

    'Write the weather station  information
    Print #1, WthDescription
    Print #1,    ' Print blank line to file.
    Print #1,    ' Print blank line to file.
    Print #1, headergenotype         'print header of genotype

    'Write the selected genotype information
    Print #1, VarArray(SelVar).VarDescription
    Print #1,    ' Print blank line to file.
    Print #1,    ' Print blank line to file.
```

```
'write the header for each variable
Print #1, "@"; "DATE"; Tab(8); "DLen"; Tab(15); "CumGDD"; _
Tab(25); "Stage", Tab(33); "#Leaf", Tab(40); "GDDi"

'then "RUN" the program after "reading those coefficient"
Dim ISTAGE As Integer    'ISTAGE of development
Dim DGDDemerge As Single     'base on Tbaseemerge
Dim DGDDleaf As Single     'base on Tbaseleaf
Dim DGDDpi As Single     'base on Tbasepi
Dim DGDDpe As Single     'base on Tbasepe
Dim DGDDi As Single          'base on Tbasei
Dim DGDD As Single     'Daily GDD
Dim CGDD As Single     'CUMGDD of Tbaseleaf(Emergence to PE)
Dim SGDD1 As Single
Dim SGDD2 As Single
Dim SGDD3 As Single
Dim SGDD9 As Single
Dim XSTAGE As Single         'development rate of ISTAGE0
Dim DL As Single                 'daylength
Dim DLyest As Single         'daylength of yesterday
Dim LDR As Single                'leaf development rate
Dim LN As Single                 'leaf number at each day
Dim k As Long                     'count

'Start at ISTAGE = 7
ISTAGE = 7

'Change "Date" to be the number (like in Excel)
'To count number of days form Planting to Harvesting
k = (CLng(CDate(txtHd.Text)) - CLng(CDate(txtPd.Text)))

'Check that "Is  #Days(k)  = #Days in WthArray(daysNdx) ?"
If k = daysNdx Then

    'if it's true then the simulation can start.
    For eachdate = 0 To daysNdx

        ReDim Preserve ResultArray(eachdate)
        DGDDemerge = 0
        DGDDleaf = 0
        DGDDpi = 0
        DGDDi = 0
        DGDD = 0
        LDR = 0

        'Calculate the daylength
        If DLmodel = 1 Then
            DL = CERES(Latitude, jdate(WthArray(eachdate).yydoy))
        ElseIf DLmodel = 2 Then
            DL = CBM(Latitude, DldefArray(SelDef).inDegrees, _
            jdate(WthArray(eachdate).yydoy))
        End If
        ResultArray(eachdate).Daylength = Format(CStr(DL), "#0.00")

        'Development stage
        Select Case ISTAGE

            'Start simulation at planting
            Case Is = 7
                DGDDemerge = GDD(WthArray(eachdate).Tmax, WthArray(eachdate).Tmin,
                Tbaseemerge)
                DGDD = DGDDemerge
                XSTAGE = 7
                ISTAGE = 8

            'Planting to Root germiation
            Case Is = 8
                DGDDemerge = GDD(WthArray(eachdate).Tmax, WthArray(eachdate).Tmin,
                Tbaseemerge)
                DGDD = DGDDemerge
                XSTAGE = 8
                ISTAGE = 9
```

```
'Root germination to Emergence
Case Is = 9
    DGDDemerge = GDD(WthArray(eachdate).Tmax, WthArray(eachdate).Tmin,
    Tbaseemerge)
    SGDD9 = SGDD9 + DGDDemerge
    DGDD = DGDDemerge

    If CaneType = 1 Then
        RDEP = RDEP + (0.15 * DGDDemerge)
        P9 = 170 + (10 * SDEPTH)
        XSTAGE = 8 + (SGDD9 / P9)
        If XSTAGE > 9 Then
            ISTAGE = 1
            XSTAGE = 0
            SGDD1 = SGDD9 - P9    'SGDD exceed the required for this
            phase
            DGDDleaf = SGDD9 - P9   'Start calculate leaf development
            LDR = DGDDleaf / VarArray(SelVar).PI1
        End If
    ElseIf CaneType = 2 Then
        XSTAGE = 8 + (SGDD9 / 200)
        If XSTAGE > 9 Then
            ISTAGE = 1
            XSTAGE = 0
            SGDD1 = SGDD9 - 200     'SGDD exceed the required for this
            phase
            DGDDleaf = SGDD9 - 200   'Start calculate leaf development
            LDR = DGDDleaf / VarArray(SelVar).PI1
        End If
    End If

'Emergence to End of Juvenile
Case Is = 1
    DGDDleaf = GDD(WthArray(eachdate).Tmax, WthArray(eachdate).Tmin,
    Tbaseleaf)
    DGDD = DGDDleaf
    LDR = DGDDleaf / VarArray(SelVar).PI1
    SGDD1 = SGDD1 + DGDDleaf
    XSTAGE = 0 + ((SGDD1 / VarArray(SelVar).PI1) / 14)

    If XSTAGE >= 1 Then
        ISTAGE = 2
        XSTAGE = 1
        SGDD2 = SGDD1 - (VarArray(SelVar).PI1 * 14)  'SGDD exceed the
        required for this phase
    End If

'End of Juvenile to Panicle Initiation
Case Is = 2
    DGDDpi = GDD(WthArray(eachdate).Tmax, WthArray(eachdate).Tmin,
    Tbasepi)
    DGDDleaf = GDD(WthArray(eachdate).Tmax, WthArray(eachdate).Tmin,
    Tbaseleaf)
    LDR = DGDDleaf / VarArray(SelVar).PI2

    If DLyest >= DL Then
        RATEIN = RATEINgddrate(DGDDpi, VarArray(SelVar).P22, DL, _
        VarArray(SelVar).P20, VarArray(SelVar).PS)
        If RATEIN > 0 Then
            DGDD = DGDDpi
        Else
            DGDD = DGDDleaf
        End If
    End If

    SGDD2 = SGDD2 + DGDD
    SIND = SIND + RATEIN
    XSTAGE = 1 + SIND

    If SIND >= 1 Then
        ISTAGE = 3
        XSTAGE = 2
    End If
```

```
                'Panicle Initiation to Panicle Emergence
                Case Is = 3
                    DGDDleaf = GDD(WthArray(eachdate).Tmax, WthArray(eachdate).Tmin,
                    Tbaseleaf)
                    LDR = DGDDleaf / VarArray(SelVar).PI2
                    DGDDpe = GDD(WthArray(eachdate).Tmax, WthArray(eachdate).Tmin,
                    Tbasepe)
                    DGDD = DGDDpe
                    SGDD3 = SGDD3 + DGDDpe
                    XSTAGE = 2 + (SGDD3 / VarArray(SelVar).PE)

                    If XSTAGE >= 3 Then
                        XSTAGE = 3
                        ISTAGE = 4
                        DGDDleaf = DGDDleaf - (SGDD3 - VarArray(SelVar).PE)
                        LDR = DGDDleaf / VarArray(SelVar).PI2
                    End If

                'After Panicle Emergence
                Case Is = 4
                    DGDDi = GDD(WthArray(eachdate).Tmax, WthArray(eachdate).Tmin,
                    Tbasei)
                    DGDD = DGDDi

                'End at XSTAGE 3 (still calculate daily GDD until harvest)
            End Select

            'The daily gdd of plant is represented by GDDi.
            ResultArray(eachdate).GDDi = Format(CStr(DGDD), "##00.00")

            'Calculate Cumumlative GDD after emergence
            CGDD = CGDD + DGDDleaf + DGDDi
            ResultArray(eachdate).CUMGDDleaf = Format(CStr(CGDD), "##00.00")

            'Calculate #Leaf after emergence to PE
            LN = LN + LDR
            ResultArray(eachdate).Nleaf = Format(CStr(LN), "##00.00")

            'The stage of plant is represented by XSTAGE variable.
            ResultArray(eachdate).Stage = Format(CStr(XSTAGE), "##00.00")

            ' write daily "CGDD , DL, PINR, A" into the output file
            Print #1, WthArray(eachdate).yydoy; Tab(8);
            ResultArray(eachdate).Daylength; _
            Tab(15); ResultArray(eachdate).CUMGDDleaf; Tab(25); ResultArray
            (eachdate).Stage; _
            Tab(33); ResultArray(eachdate).Nleaf; Tab(40); ResultArray(eachdate).GDDi

            'After write the DL to file it become DL of yesterday
            Dlyest = DL

        Next eachdate
        Close #1    ' Close file.

        'After simulation finish, simulation result is shown.
        frmMain.Hide
        frmSimResult.Show


    Else
        The #Days(k) <> #Days in WthArray
        MsgBox "The selected weather file may not match the crop duration.", _
        0, "Missing field"
    End If
End If
End Sub ' cmdSim_Click
```

```
Private Sub txtPd_LostFocus()
' The txtPd_LostFocus procedure ensures that
' the value entered into the Date text box is a valid date.
If IsDate(txtPd.Text) Then
    Pd = txtPd.Text
    Pyyjdate = yyjdate(txtPd.Text)
    txtPd.Text = Cdatetolongdate(txtPd.Text)
ElseIf txtPd.Text <> "" Then
    MsgBox "Your input is not date.", _
    0, "Missing date"
End If
End Sub  ' txtPd_LostFocus

Private Sub txtHd_LostFocus()
' The txtHd_LostFocus procedure ensures that
' the value entered into the Date text box is a valid date.
If IsDate(txtHd.Text) Then
    Hd = txtHd.Text
    Hyyjdate = yyjdate(txtHd.Text)
    txtHd.Text = Cdatetolongdate(txtHd.Text)
ElseIf txtPd.Text <> "" Then
    MsgBox "Your input is not date.", _
    0, "Missing date"
End If
End Sub  ' txtHd_LostFocus

Private Sub txtSd_LostFocus()
' The txtHd_LostFocus procedure ensures that
' the value entered into the Date text box is a valid date.
If txtSd.Text <> "" Then
    SDEPTH = CSng(txtSd.Text)
End If
End Sub  ' txtHd_LostFocus

'After change the management date
'Wth file must be selected again.
Private Sub txtHd_Change()
FoundWthdate = False
End Sub 'txtHd_Change
Private Sub txtPd_Change()
FoundWthdate = False
End Sub 'txtPd_Change

'Confirm Exiting
Private Sub cmdExit_Click()
    Dim Msg, Style, Title, Help, Response
    Msg = "Do you want to exit ?"   ' Define message.
    Style = vbYesNo + vbCritical + vbDefaultButton2 ' Define buttons.
    Title = "Sugarcane Flowering Model"  ' Define title.
    Response = MsgBox(Msg, Style, Title, Help, Ctxt)
    If Response = vbYes Then    ' User chose Yes.
        End    ' Perform some action.
    Else    ' User chose No.
    ' Not perform any action.
    End If
End Sub

Private Sub cmdVar_Click()
    frmMain.Hide
    frmVar.Show
End Sub

Private Sub cmdWth_Click()
If Pyyjdate = Empty Or Hyyjdate = Empty Then
    MsgBox "You must determine planting and harvesting date before selecting weather
    file", 0, "Missing field"
Else
    frmMain.Hide
    frmWth.Show
End If
End Sub
```

```
Private Sub cmdOut_Click()
    frmMain.Hide
    frmOut.Show
End Sub

Private Sub cmdSimopt_Click()
    frmMain.Hide
    frmSimopt.Show
End Sub

'Explain "How to use, and each part information by List Box.

Private Sub cmdSim_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As
Single)
    lblDes.Caption = "Click here to run simulation"
End Sub

Private Sub cmdSimopt_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As
Single)
    lblDes.Caption = "Click here to select optional simulation for daylength and
    sugarcane type."
End Sub

Private Sub cmdVar_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As
Single)
    lblDes.Caption = "Click here to select a variety that its development will be
    simulated."
End Sub

Private Sub cmdWth_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As
Single)
    lblDes.Caption = "Click here to select weather data files (or a file) of the crop
    duration and location."
End Sub

Private Sub Form_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As
Single)
    lblDes.Caption = "First of all you need to put Planting, Harvesting date and
    Sowing depth.  Then, select weather data, a variety, and sink of this simulation
    output result.  Moreover, this model provide optional of daylength and panicle
    initiation model."
End Sub

Private Sub fraMan_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As
Single)
    lblDes.Caption = "Input date as date formats according to the locale setting of
    your system and the result is in dd-mmm-yyyy format  and sowing depth is in unit
    of centimeter."
End Sub

Private Sub cmdOut_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As
Single)
    lblDes.Caption = "Click here to select or create a file name for storage
    simulation result."
End Sub
```

Variety Form (Screen 2)

```
Dim defaultfile As String

'ReOpen the default file, which is the previous selected file
Private Sub cmdDef_Click()
    If FileExists(defaultfile) = True Then
        SelVarfile = defaultfile
    Else
        MsgBox "Default File not found", _
        0, "Missing field"
    End If
    Form_Load
End Sub

'Open a new file
Private Sub cmdNew_Click()
    cdlNewfile.ShowOpen
    SelVarfile = cdlNewfile.fileName
    Form_Load
End Sub

'Re-read the variety parameter in order to allow parameter edition in the *.cul
Private Sub form_activate()
    Form_Load
End Sub

Private Sub Form_Load()
    Dim Readline1 As Variant
    Dim Readline2 As Variant
    cboVar.Clear

    'Open the "FilePath" file and read 1st line which is the default *.cul.
    Open App.Path & "\" & "FilePath.Txt" For Input As #2
        Line Input #2, Readline2
        defaultfile = Readline2
    Close #2

    'Are the default file name exist in the computer?
    'If yes, open the file
    If FileExists(defaultfile) = True Then
        Open defaultfile For Input As #1
        lblpath.Caption = defaultfile
        If SelVarfile <> "" Then    'If a new file was selected.
            Close #1       'Close the default file, then open the selected file
            Open SelVarfile For Input As #1
                lblpath.Caption = SelVarfile
        End If

    'If the default file not found, Are the selected file exist?
    ElseIf FileExists(SelVarfile) = True Then
        Open SelVarfile For Input As #1
            lblpath.Caption = SelVarfile

    Else    'In case of neither Default nor Selected file was found.
        lblpath.Caption = "Default File not found"
        GoTo EndIT:     'Do not need to find any parameter from file
    End If

    'Set as False to Check to True
    'If a variety was selected and the required parameter are ready.
    FoundVar = False
    FoundVarPar = False

    'Reads variety parameters and its contents into VarArray, an array of records.
    Also create the list of variety in the lstVar list box.

    'each variable is set for Count VarArray Index
    eachVar = 0
    Do While Not EOF(1)
        Line Input #1, Readline1
        If Left(Readline1, 1) = "@" Then
            headergenotype = Readline1          'print header of genotype

        ElseIf Left(Readline1, 1) <> "" And Left(Readline1, 1) <> "@" And Left
        (Readline1, 1) <> "!" And Left(Readline1, 1) <> "*" Then
            ReDim Preserve VarArray(eachVar)
            FoundVar = True     'A variety was found

            ' Then check "parameter
            If (IsNumeric(LTrim(Mid(Readline1, 56, 5))) And IsNumeric(LTrim(Mid
            (Readline1, 62, 5))) And IsNumeric(LTrim(Mid(Readline1, 74, 5))) And
            IsNumeric(LTrim(Mid(Readline1, 81, 4))) And IsNumeric(LTrim(Mid(Readline1,
            98, 5)))) And IsNumeric(LTrim(Mid(Readline1, 86, 5))) Then
                FoundVarPar = True      'All required parameter was found

                ' Then store the record number in the array.
                VarArray(eachVar).VarDescription = Readline1
                VarArray(eachVar).Varname = RTrim(Mid(Readline1, 8, 15))
                VarArray(eachVar).PE = CSng(LTrim(Mid(Readline1, 98, 5)))
                VarArray(eachVar).PI1 = CSng(LTrim(Mid(Readline1, 56, 5)))
                VarArray(eachVar).PI2 = CSng(LTrim(Mid(Readline1, 62, 5)))
                VarArray(eachVar).PS = CSng(LTrim(Mid(Readline1, 74, 5)))
                VarArray(eachVar).P20 = CSng(LTrim(Mid(Readline1, 81, 4)))
                VarArray(eachVar).P22 = CSng(LTrim(Mid(Readline1, 86, 5)))

                ' Build the variety list. and Keep only variety row
                cboVar.AddItem VarArray(eachVar).Varname
                eachVar = eachVar + 1
            End If
        End If
    Loop
    Close #1

    ' Keep a record of the last variety index.
    VarNdx = eachVar - 1

    EndIT:
End Sub  ' Form_Load

Private Sub cboVar_Click()
' Read the user's selected variety
' and display the variety name in the selected variety
    Dim i As Integer
    For i = 0 To VarNdx
        If cboVar.Text = VarArray(i).Varname Then
            SelVar = i
            Exit For
        End If
    Next i
End Sub

Private Sub cmdOK_Click()
    If cboVar.Text = "" Then
        MsgBox "You must select a variety.", _
        0, "Variety Selection"
    Else
        SelVarfile = lblpath.Caption
        frmVar.Hide
        frmMain.Show
    End If
End Sub

Private Sub cmdCancel_Click()
    If cboVar.Text = "" Then
        MsgBox "You must select a variety.", _
        0, "Variety Selection"
    Else
        frmVar.Hide
        frmMain.Show
    End If
End Sub
```

Weather (Screen 3)

```
'weather file form
'weather.frm
' Produces weather data array
' Reads weather data from the default path in FilePath
Dim DefaultPath As String
Dim k As Long

Private Sub cmdCancel_Click()
    If FoundLat = False Or FoundWthdate = False Then
        MsgBox "Your weather file may miss a required value.", _
        0, "Missing field"
    Else
        frmWth.Hide
        frmMain.Show
    End If
End Sub 'cmdCancel_Click

Private Sub cmdOK_Click()
' Reads weather data from the default path in FilePath
  ' Open the *.wth file and
  ' read its contents into WthArray,
  ' an array of records. consist of Tmax, Tmin, YYDOY
    Dim j As Integer
    Dim eachdate As Integer
    Dim Readline As Variant
    FoundLat = False
    FoundWthdate = False
    eachdate = 0

    If lstWth.ListCount = 0 Then
        MsgBox "You must select weather file that match the crop duration.", _
        0, "Weather file"

    Else
        WthDirSelect = filWth.Path
        For j = 0 To lstWth.ListCount - 1
            Open filWth.Path & "\" & lstWth.List(j) For Input As #1 + j

            'read "LAT" of the "weather file"
            Do While Not EOF(1 + j)
                Line Input #1 + j, Readline
                If Left(Readline, 1) = "@" Then
                    headerweather = Readline
                    Line Input #1 + j, Readline
                    WthDescription = Readline
                    Latitude = CSng(LTrim(Mid(Readline, 10, 6)))     'got LAT
                    FoundLat = True     'Found latitude
                    Exit Do
                End If
            Loop

            ' Read the file.
            Do While Not EOF(1 + j)
                Line Input #1 + j, Readline
                If Left(Readline, 5) >= Pyyjdate And Left(Readline, 5) <= Hyyjdate Then
                    ReDim Preserve WthArray(eachdate)
                    FoundWthdate = True     'found all required parameter

                    ' Then store the record number in the array.
                    WthArray(eachdate).yydoy = Left(Readline, 5)          'got yydoy
                    list
                    WthArray(eachdate).Tmax = CSng(LTrim(Mid(Readline, 14, 4)))
                    WthArray(eachdate).Tmin = CSng(LTrim(Mid(Readline, 20, 4)))
                    eachdate = eachdate + 1
                End If
            Loop
            Close #j + 1
        Next j

        ' Keep a record of the last WthArray index.
        daysNdx = eachdate - 1
        k = CLng(DateValue(Hd)) - CLng(DateValue(Pd))
        'Check that all needed parameter are read

        If k <> daysNdx Then
            MsgBox “The selected weather file not match the crop duration.”, _
            0, “Missing field”

        ElseIf FoundLat = False Or FoundWthdate = False Then
            MsgBox “the selected weather file may miss a required value.”, _
            0, “Missing field”

        Else
            frmWth.Hide
            frmMain.Show
        End If

    End If
End Sub  ' cmdOK (wth files were selected)

Private Sub cmdAdd_Click()
    If Not AlreadyListed(lstWth, filWth.fileName) _
        Then lstWth.AddItem filWth.fileName
End Sub

Private Sub cmdRemove_Click()
    If lstWth.ListCount <> 0 Then _
        lstWth.RemoveItem (lstWth.ListCount - 1)
End Sub

Private Sub cmdSetnewpath_Click()
    frmNewpath.Show
End Sub

Private Sub filWth_DblClick()
    If Not AlreadyListed(lstWth, filWth.fileName) _
        Then lstWth.AddItem filWth.fileName
End Sub

Private Sub form_activate()
    lstWth.Clear
    If WthDirSelect <> “” Then
        filWth.Path = WthDirSelect
        lblpath.Caption = WthDirSelect

    Else
        If filePathExists(filWth, DefaultPath) = True Then
            filWth.Path = DefaultPath
            lblpath.Caption = DefaultPath
        End If
    End If
End Sub

Private Sub Form_Load()
    Dim Readline2 As Variant
    lstWth.Clear

    Open App.Path & “\” & “FilePath.Txt” For Input As #2
        Line Input #2, Readline2
        Line Input #2, Readline2
        DefaultPath = Readline2
    Close #2

    If filePathExists(filWth, DefaultPath) = True Then
        filWth.Path = DefaultPath
        lblpath.Caption = DefaultPath
    End If
End Sub
```

SET NEW PATH

```
Dim DefaultPath As String

Private Sub cmdCancel_Click()
    frmNewpath.Hide
    frmWth.Show
End Sub

Private Sub cmdDefault_Click()
    If dirPathExists(Dir1, DefaultPath) = True Then
        Drive1.Drive = Left(DefaultPath, 3)
        Dir1.Path = DefaultPath
    Else
        MsgBox "Default path not found", _
        0, "Missing field"
    End If
End Sub

Private Sub cmdOK_Click()
    WthDirSelect = Dir1.Path
    frmNewpath.Hide
    frmWth.Show
End Sub

Private Sub Drive1_Change()
    Dir1.Path = Drive1.Drive
End Sub

Private Sub Form_Load()
    Open App.Path & "\" & "FilePath.Txt" For Input As #2
        Line Input #2, Readline2
        Line Input #2, Readline2
        DefaultPath = Readline2
    Close #2
End Sub
```

Simulation Option Form (Screen 4)

```
Private Sub cmdCancel_Click()
    If Dlmodel = 0 Or CaneType = 0 Then
        MsgBox “You must select an simulation option.”, _
        0, “Missing field”

    ElseIf Dlmodel = 2 And Not FoundDegree = True Then
        MsgBox “Daylength definition may missing.”, _
        0, “Missing field”

    Else
        frmSimopt.Hide
        frmMain.Show
    End If
End Sub

Private Sub cmdOK_Click()
    If optCERES.Value = True Then
        Dlmodel = 1

    ElseIf optCBM.Value = True And FoundDegree = True Then
        Dlmodel = 2

    ElseIf optCBM.Value = True And Not FoundDegree = True Then
        MsgBox “Daylength definition may missing.”, _
        0, “Missing field”
    End If

    If optPlanted.Value = True Then
        CaneType = 1

    ElseIf optRatooned.Value = True Then
        CaneType = 2
    End If

    If Dlmodel <> 0 And CaneType <> 0 Then
        frmSimopt.Hide
        frmMain.Show
    End If
End Sub

‘ Reads daylength definition  from
‘ a text file named “Application Path & \Dldefinition.txt”
Private Sub Form_Load()
    Dlmodel = 0
    CaneType = 0
    FoundDegree = False
‘ Open the Dldefinition.Txt file and
‘ read its contents into DldefArray,
‘ an array of records. Also create
‘ the list of definition in the
‘ cboDef combo box.
    Dim i As Integer

    Open App.Path & “\Dldefinition.Txt” For Input As #1
        i = 0
        ‘ Read the file.

        Do While Not EOF(1)
            Input #1, DldefArray(i).Dlname
            Input #1, DldefArray(i).inDegrees
            Input #1, DldefArray(i).definition
            ‘ Build the daylength definition list.
            cboDef.AddItem DldefArray(i).Dlname
            i = i + 1
        Loop
    Close #1

‘ Keep a record of the last definition index.
    DldefNdx = i - 1
End Sub  ‘ Form_Load
```

```
Private Sub cboDef_Click()
' Read the user's daylength definition choice
' and display the definition description
' for the selected definition
    Dim i As Integer
    For i = 0 To DldefNdx
        If cboDef.Text = DldefArray(i).Dlname Then
            lblDes.Caption = "Daylength definition  = " & DldefArray(i).definition
            SelDef = i
            FoundDegree = True
            Exit For
        End If
    Next i
End Sub  ' cboDef_Click
Private Sub optCERES_Click()
    lblDes.Caption = "Daylength definition of CERES model is fixed.  CERES model
    defines daylength as including the periods of civil twilight."
    optCERES.Value = True
    optCBM.Value = False
    cboDef.Enabled = False
End Sub
Private Sub optCBM_Click()
    lblDes.Caption = "CBM model allows user to select the daylength definition which
    defined by position of the sun,... Select the user's daylength definition from the
    definition list"
'Check either "check CBM" or "non-check CBM", if "check CBM", the "Combo1" is
available
    MsgBox "please, put in your daylength definition", 0, "Next needed parameter"
    optCBM.Value = True
    cboDef.Enabled = True
    optCERES.Value = False
End Sub
Private Sub optPlanted_Click()
    lblDes.Caption = "The sugarcane plant type is planted cane."
    optPlanted.Value = True
    optRatooned.Value = False
End Sub
Private Sub optRatooned_Click()
    lblDes.Caption = "The sugarcane plant type is ratooned cane."
    optRatooned.Value = True
    optPlanted.Value = False
End Sub
Private Sub frmSimopt_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As
Single)
    lblDes.Caption = "Select the desire daylength calculation model and the sugarcane
    plant type from the available options"
End Sub
Private Sub fraType_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As
Single)
    lblDes.Caption = "Select a plant type to simulate the sugarcane phenological
    development."
End Sub
Private Sub optCBM_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As
Single)
    lblDes.Caption = "CBM model allows user to select the daylength definition which
    defined by position of the sun,... Select the user's daylength definition from the
    definition list"
End Sub
Private Sub optCERES_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As
Single)
    lblDes.Caption = "Daylength definition of CERES model is fixed.  CERES model
    defines daylength as including the periods of civil twilight."
End Sub
Private Sub optPlanted_MouseMove(Button As Integer, Shift As Integer, X As Single, Y
As Single)
    lblDes.Caption = "Select this option when planting of node sugarcane stalk is
    growing method."
End Sub
Private Sub optRatooned_MouseMove(Button As Integer, Shift As Integer, X As Single, Y
As Single)
    lblDes.Caption = "Select this option when the sugarcane plant is cutted and let to
    grow up as a next crop."
End Sub
```

Output File (Screen 5)

```
Dim DefaultPath As String

Private Sub cmdCancel_Click()
    If Outputfile = Empty Then
        MsgBox "You must select or create a file name as output file.", _
        0, "Missing field"
    Else
        frmOut.Hide
        frmMain.Show
    End If
End Sub

Private Sub cmdDefault_Click()
    If dirPathExists(dirOut, DefaultPath) = True Then
        drvOut.Drive = Left(DefaultPath, 3)
        dirOut.Path = DefaultPath
        filOut.Path = dirOut.Path
    Else
        MsgBox "Default path not found", _
        0, "Missing field"
    End If
    txtOut.Text = "flower.out"
End Sub

Private Sub cmdOK_Click()
    Dim Msg, Style, Title, Response
    If txtOut.Text = "" Or txtOut.Text = "*.out" Then
        MsgBox "You must select or create a file name as output file.", _
        0, "Missing field"
    Else
        Outputfile = filOut.Path & "\" & txtOut.Text
        Outputpath = filOut.Path

        If FileExists(Outputfile) = True Then
            Msg = "The file   " & Outputfile & "   aready exists.  Do you want to
            replace the existing file ?"   ' Define message.
            Style = vbYesNo  ' Define buttons.
            Title = "Output filename"  ' Define title.
            ' context.
            ' Display message.
            Response = MsgBox(Msg, Style, Title)
            If Response = vbYes Then    ' User chose Yes.
                frmOut.Hide
                frmMain.Show
            Else    ' User chose No.
                Outputfile = Empty
                txtOut.Text = "*.out" ' Perform some action.
            End If
        Else
            frmOut.Hide
            frmMain.Show
        End If
    End If
End Sub

Private Sub drvOut_Change()
    lblpath.Caption = dirOut.Path
    dirOut.Path = drvOut.Drive ' Set Dir list box
End Sub

Private Sub dirOut_Change()
    lblpath.Caption = dirOut.Path
    filOut.Path = dirOut.Path ' Set file list box path.
End Sub

Private Sub filOut_Click()
    txtOut.Text = filOut.fileName
    lblpath.Caption = filOut.Path & "\" & filOut.fileName
End Sub
```

```
Private Sub form_activate()
    If Outputpath <> "" Then
        drvOut.Drive = Left(Outputpath, 3)
        dirOut.Path = Outputpath
        filOut.Path = dirOut.Path
        filOut.Refresh
        lblpath.Caption = Outputpath
    Else
    End If
    txtOut.Text = "*.out"
End Sub

Private Sub Form_Load()
    Dim Readline2 As Variant
    Open App.Path & "\" & "FilePath.Txt" For Input As #2
        Line Input #2, Readline2
        Line Input #2, Readline2
        Line Input #2, Readline2
        DefaultPath = Readline2
    Close #2

    If dirPathExists(dirOut, DefaultPath) = True Then
        drvOut.Drive = Left(DefaultPath, 3)
        dirOut.Path = DefaultPath
        filOut.Path = dirOut.Path
    Else
        lblpath.Caption = "Default path not found"
    End If
    txtOut.Text = "*.out"
End Sub
```

Simulation result (Screen 6)

```
Private Sub cmdback_Click()
    frmSimResult.Hide
    frmMain.Show
End Sub

Private Sub cmdChart_Click()
    frmChart.Show
    frmSimResult.Hide
End Sub

Private Sub cmdPrint_Click()
    Printer.Print rtbResult.Text
    Printer.EndDoc
End Sub

Private Sub form_activate()
    Form_Load
End Sub

Private Sub Form_Load()
    rtbResult.fileName = Outputfile
End Sub
```

Chart (Screen 7)

```
Private Sub cmdClose_Click()
    frmChart.Hide
    frmSimResult.Show
End Sub

Private Sub cboYaxis_Click()
    Dim i As Integer
    If cboYaxis.Text = "All" Then
        ReDim X(1 To (daysNdx + 2), 1 To 5)

        'set the data
        For i = 2 To UBound(X, 1)
            X(i, 2) = CSng(ResultArray(i - 2).Daylength)
            X(i, 3) = CSng(ResultArray(i - 2).CUMGDDleaf)
            X(i, 4) = CSng(ResultArray(i - 2).Stage)
            X(i, 5) = CSng(ResultArray(i - 2).Nleaf)
        Next

        'Set row lebel
        For i = 2 To UBound(X, 1)
            X(i, 1) = Cyyjdatetodate(WthArray(i - 2).yydoy)
        Next

        'Set column lebel
        X(1, 2) = "Daylength"
        X(1, 3) = "CUMGDD"
        X(1, 4) = "Stage"
        X(1, 5) = "#Leaf"

        ' set the chart data
        chartResult.ChartData = X
    Else
        ReDim X(1 To (daysNdx + 2), 1 To 2)
        For i = 2 To UBound(X, 1)
            X(i, 1) = Cyyjdatetodate(WthArray(i - 2).yydoy)
            Select Case cboYaxis.Text
                Case "Daylength"
                    X(i, 2) = ResultArray(i - 2).Daylength
                    X(1, 2) = "Daylength"
                Case "CUMGDD"
                    X(i, 2) = ResultArray(i - 2).CUMGDDleaf
                    X(1, 2) = "CUMGDD"
                Case "Stage"
                    X(i, 2) = ResultArray(i - 2).Stage
                    X(1, 2) = "Stage"
                Case "#Leaf"
                    X(i, 2) = ResultArray(i - 2).Nleaf
                    X(1, 2) = "#Leaf"
            End Select
        Next
        ' set the chart data
        chartResult.ChartData = X
    End If
    'chartResult.Plot.WidthToHeightRatio = 1.25
End Sub

Private Sub form_activate()
    Form_Load
End Sub
```

```
Private Sub cmdPrintWindow_Click()
    cmdPrintWindow.Visible = False
    cmdClose.Visible = False
    Line1.Visible = False
    Line2.Visible = False
    Label1.Visible = False
    Label2.Visible = False
    txtXaxis.Visible = False
    cboYaxis.Visible = False
    frmChart.Visible = False
    Shape1.Visible = False
    Shape2.Visible = False
PrintForm
    cmdPrintWindow.Visible = True
    cmdClose.Visible = True
    Line1.Visible = True
    Line2.Visible = True
    Label1.Visible = True
    Label2.Visible = True
    txtXaxis.Visible = True
    cboYaxis.Visible = True
    frmChart.Visible = True
    Shape1.Visible = True
    Shape2.Visible = True
End Sub

Private Sub Form_Load()
    cboYaxis.Clear
    txtXaxis.Text = "date"
    cboYaxis.AddItem "All"
    cboYaxis.AddItem "Daylength"
    cboYaxis.AddItem "CUMGDD"
    cboYaxis.AddItem "Stage"
    cboYaxis.AddItem "#Leaf"
    cboYaxis.ListIndex = 0

    ReDim X(1 To (daysNdx + 2), 1 To 5)
    'set the data
    For i = 2 To UBound(X, 1)
        X(i, 2) = CSng(ResultArray(i - 2).Daylength)
        X(i, 3) = CSng(ResultArray(i - 2).CUMGDDleaf)
        X(i, 4) = CSng(ResultArray(i - 2).Stage)
        X(i, 5) = CSng(ResultArray(i - 2).Nleaf)
    Next

    'Set row lebel
    For i = 2 To UBound(X, 1)
        X(i, 1) = Cyyjdatetodate(WthArray(i - 2).yydoy)
    Next

    'Set column lebel
    X(1, 2) = "Daylength"
    X(1, 3) = "CUMGDD"
    X(1, 4) = "Stage"
    X(1, 5) = "#Leaf"

    ' set the chart
    With chartResult
        .ChartData = X
        .TextLengthType = VtTextLengthTypeDevice
    End With
End Sub
```

Module

```
' Cane Phenology
' Module: Phenology.BAS
'
'These variables are publiced
'for communication between forms
Public Pd As Date
Public Hd As Date
Public Pyyjdate As String   'Change format for read wth file
Public Hyyjdate As String
Public SDEPTH As Single

Public SelVarfile As String
Public headergenotype As Variant

' VarArray is a dynamic array of
'    record numbers, representing the
'    numbers of Varieties in genotype file
Public VarArray() As VarRec
Public VarNdx As Integer
Public SelVar As Integer

Public WthDirSelect As String   'New Directory of wth
Public headerweather As Variant
Public WthDescription As Variant
Public Latitude As Single          'L is latitude

' WthArray is a dynamic array of
'    record numbers, representing the
'    numbers of days(k) from Pdate to Hdate
Public WthArray() As WthRec
Public daysNdx As Integer

'DLmodel = 1 is CERESmodel
'DLmodel = 2 is CBM model
Public DLmodel As Byte

'DldefArray contain
'DldefArray(SelDef).inDegrees" for cal daylength
Public DldefArray(8) As DldefRec
Public DldefNdx As Integer
Public SelDef As Integer

'CaneType = 1 is plant cane
'CaneType = 2 is ratooned cane
Public CaneType As Byte

Public Outputfile As String        'output path and file name
Public Outputpath As String     'output path

'ResultArray is a dynamic array of record numbers,
'   representing the numbers of runtimes of the model,
'  which is equal to numbers of days(k) from Pdate to Hdate
Public ResultArray() As ResultRec

'Xarray is a dynamic array of coordinate (x,y) on chart
Public X() As Variant

'For check that "Was the parameter found?"
Public FoundWthdate As Boolean
Public FoundLat As Boolean
Public FoundVar As Boolean
Public FoundVarPar As Boolean
Public FoundDegree As Boolean
```

```
' Type definition for variety and its parameter records.
Type VarRec
    PS As Single
    P20 As Single
    PE As Single
    PI1 As Single
    PI2 As Single
    P22 As Single
    Varname As String
    VarDescription As Variant
End Type

'Type definition of weather data set (Tmax, Tmin, yydoy)
Type WthRec
    Tmax As Single
    Tmin As Single
    yydoy As String
End Type

'Type definition for daylength and its degree records.
Type DldefRec
    Dlname As String
    inDegrees As Single
    definition As String
End Type

'Type simulation result and its records
Type ResultRec
    Daylength As String
    CUMGDDleaf As String
    Stage As String
    Nleaf As String
    GDDi As String
End Type

Function GDD(Tmax As Single, Tmin As Single, Tbase As Single) As Single
'Growing Degree Day (Daily)
    GDD = ((Tmax + Tmin) / 2) - Tbase
    If GDD < 0 Then
        GDD = 0
    End If
End Function

Function RATEINgddrate(DGDD As Single, P22 As Single, DL As Single, P20 As Single, PS As
Single) As Single
 ' RATEIN is function of daylength and temperature
 'inversion of the duration to initiation(rate)
    ' DL = calculated daylength
    ' P20 = threshold photoperiod or maximum optimum photoperiod
    ' P22 = in this case, it is phyllochron (PI2)
    ' PS = photoperiod sensitivity ((1/days)/hour)
    If DL <= P20 Then
        RATEINgddrate = (DGDD / P22) * 1
    'daylength function = 1 under optimal photoperiod
    ElseIf P20 = -99 Then
    'for Non-photoperiod sensitive cultivar
        RATEINgddrate = 0   'No photoinduction
    Else
        If (1 + (PS * (P20 - DL))) > 0 Then
            RATEINgddrate = (DGDD / P22) * (1 + (PS * (P20 - DL)))
            'daylength function have negative slope
            'So give PS as positive value because already minus(-)
        Else    'In case of DL beyond the critical photoperiod, photo-induction not occur
            RATEINgddrate = 0
        End If
    End If
End Function

Function Arccos(number As Double) As Double
    Arccos = Atn(-number / Sqr(-number * number + 1)) + 2 * Atn(1)
    'calculation method is from "derived math function" of VB5
End Function
```

```
Function Arcsin(number As Double) As Double
    Arcsin = Atn(number / Sqr(-number * number + 1))
    'calculation method is from "derived math function" of VB5
End Function

Function CBM(L As Single, p As Single, jdate As Integer) As Single
'L as latitude is in degree
'p as daylength coefficient is in degree
'jdate is the day of the year
    Dim pi As Double
        pi = 4 * Atn(1)      ' Calculate the value of pi.
    Dim Rev As Double   'Rev is the revolution angle
        Rev = 0.2163108 + 2 * Atn(0.9671396 * Tan(0.0086 * (jdate - 186)))
    Dim Dec As Double   'Dec is the sun's declination angle
        Dec = Arcsin(0.39795 * Cos(Rev))
    Dim DL As Double    'DL as the length of the day is in hours
        DL = 24 - 24 / pi * Arccos((Sin(p * pi / 180) + (Sin(L * pi / 180) * Sin(Dec))) /
        (Cos(L * pi / 180) * Cos(Dec)))
    CBM = CSng(DL)
End Function

Function CERES(L As Single, jdate As Integer)
'CERES function us for civil twilight only
'L as latitude is in degree
'jdate is the day of the year
    Dim pi As Double
        pi = 4 * Atn(1)      'Calculate the value of pi.

    Dim Dec As Double   'Dec is the sun's declination angle
        Dec = 0.4093 * Sin(0.0172 * (jdate - 82.2))

    Dim DL As Double    'DL as the length of the day is in hours
        DL = 7.639 * Arccos(((-Sin(L * pi / 180) * Sin(Dec)) - 0.1047) / (Cos(L * pi /
        180) * Cos(Dec)))
    CERES = CSng(DL)
End Function

Function jdate(yyjdate As String) As Integer
' change from yyjdate to be jdate
    jdate = CInt(Right(yyjdate, 3))
End Function

Function yyjdate(Pdate As Date) As String
'Change from DATE to be YYJDATE
'yyjdate format is yyddd (98001)
    Dim Julian As Integer
    Dim dd, mm, yyyy As Integer

    dd = Day(Pdate)
    mm = Month(Pdate)
    yyyy = Year(Pdate)
    If yyyy Mod 4 = 0 Then
        Select Case mm
            Case Is = 1
                Julian = dd
            Case Is = 2
                Julian = dd + 31
            Case Is = 3
                Julian = dd + 60
            Case Is = 4
                Julian = dd + 91
            Case Is = 5
                Julian = dd + 121
            Case Is = 6
                Julian = dd + 152
            Case Is = 7
                Julian = dd + 182
            Case Is = 8
                Julian = dd + 213
            Case Is = 9
                Julian = dd + 244
            Case Is = 10
                Julian = dd + 274
            Case Is = 11
                Julian = dd + 305
            Case Is = 12
                Julian = dd + 335
        End Select
    Else: Select Case mm
            Case Is = 1
                Julian = dd
            Case Is = 2
                Julian = dd + 31
            Case Is = 3
                Julian = dd + 59
            Case Is = 4
                Julian = dd + 90
            Case Is = 5
                Julian = dd + 120
            Case Is = 6
                Julian = dd + 151
            Case Is = 7
                Julian = dd + 181
            Case Is = 8
                Julian = dd + 212
            Case Is = 9
                Julian = dd + 243
            Case Is = 10
                Julian = dd + 273
            Case Is = 11
                Julian = dd + 304
            Case Is = 12
                Julian = dd + 334
        End Select
    End If
    yyjdate = Right(CStr(yyyy), 2) + Format(Julian, "000")
End Function

Function Cdatetolongdate(Pdate As Date) As String
'Change from DATE to be long date format
'Long date format is  "5 January 1997"
    Dim mmm As String
    Dim dd, mm, yyyy As Integer

    dd = Day(Pdate)
    mm = Month(Pdate)
    yyyy = Year(Pdate)
    Select Case mm
        Case 1
            mmm = "January"
        Case 2
            mmm = "February"
        Case 3
            mmm = "March"
        Case 4
            mmm = "April"
        Case 5
            mmm = "May"
        Case 6
            mmm = "June"
        Case 7
            mmm = "July"
        Case 8
            mmm = "August"
        Case 9
            mmm = "September"
        Case 10
            mmm = "October"
        Case 11
            mmm = "November"
        Case 12
            mmm = "December"
    End Select
    Cdatetolongdate = CStr(dd) & " " & mmm & " " & CStr(yyyy)
End Function
```

```
Function Cyyjdatetodate(yyjdate As String) As String
' change from yyjdate to be "dd-mm-yy"
    Dim yy As String
    Dim doy As Integer
    Dim dd As String
    Dim mm As String
    yy = Left(yyjdate, 2)
    doy = CInt(Right(yyjdate, 3))

    If yy Mod 4 = 0 Then
        Select Case doy
            Case 1 To 31
                dd = doy
                mm = "Jan"
            Case 32 To 60
                dd = doy - 31
                mm = "Feb"
            Case 61 To 91
                dd = doy - 60
                mm = "Mar"
            Case 92 To 121
                dd = doy - 91
                mm = "Apr"
            Case 122 To 152
                dd = doy - 121
                mm = "May"
            Case 153 To 182
                dd = doy - 152
                mm = "Jun"
            Case 183 To 213
                dd = doy - 182
                mm = "Jul"
            Case 214 To 244
                dd = doy - 213
                mm = "Aug"
            Case 245 To 274
                dd = doy - 244
                mm = "Sep"
            Case 275 To 305
                dd = doy - 274
                mm = "Oct"
            Case 306 To 335
                dd = doy - 305
                mm = "Nov"
            Case 336 To 366
                dd = doy - 335
                mm = "Dec"
        End Select
    Else: Select Case doy
            Case 1 To 31
                dd = doy
                mm = "Jan"
            Case 32 To 59
                dd = doy - 31
                mm = "Feb"
            Case 60 To 90
                dd = doy - 59
                mm = "Mar"
            Case 91 To 120
                dd = doy - 90
                mm = "Apr"
            Case 121 To 151
                dd = doy - 120
                mm = "May"
            Case 152 To 181
                dd = doy - 151
                mm = "Jun"
            Case 182 To 212
                dd = doy - 181
                mm = "Jul"
            Case 213 To 243
                dd = doy - 212
                mm = "Aug"
```

```
      Case 244 To 273
        dd = doy - 243
        mm = "Sep"
      Case 274 To 304
        dd = doy - 273
        mm = "Oct"
      Case 305 To 334
        dd = doy - 304
        mm = "Nov"
      Case 335 To 365
        dd = doy - 334
        mm = "Dec"
    End Select
  End If
  Cyyjdatetodate = dd & "-" & mm & "-" & yy
End Function

Function AlreadyListed(listControl As Control, newItem As String) As Boolean
 ' The AlreadyListed function checks an existing
 ' list to see if a potential new entry (newItem)
 ' is currently in the list or not. If the item
 ' is in the list, the function returns true; if
 ' not, false.

  Dim i As Integer
  AlreadyListed = False
 ' Go through the current list and
 ' search for a match.
  For i = 0 To listControl.ListCount - 1
    If Lcase$(listControl.List(i)) = LCase$(newItem) Then
      AlreadyListed = True
  Next i

End Function  ' AlreadyListed

Function FileExists(fileName As String) As Boolean
 ' Checks to see if a file exists on disk.
 ' Returns True if the file is found, or
 ' False if it is not.

 ' Set up an error trap.
  On Error GoTo noFile
   ' Attempt to open the file.
    Open fileName For Input As #1
    Close #1
  ' Return True if no error occurs.
    FileExists = True
    Exit Function

 noFile:
 ' If the file can't be opened,
 ' return False.
    FileExists = False
End Function  ' FileExists

Function filePathExists(file As FileListBox, PathName As String) As Boolean
 ' Checks to see if a path exists on disk.
 ' Returns True if the path is found, or
 ' False if it is not.

 ' Set up an error trap.
  On Error GoTo noPath
    file.Path = PathName
    ' Return True if no error occurs.
    filePathExists = True
    Exit Function

 noPath:
  ' If the path can't be accessed,
  ' return False.
  filePathExists = False
End Function  ' filePathExists
```

```
Function dirPathExists(Dir1 As DirListBox, PathName As String) As Boolean

  ' Checks to see if a directory exists on disk.
  ' Returns True if it is found, or
  ' False if it is not.

  ' Set up an error trap..
    If PathName = "" Then GoTo noPath Else
    On Error GoTo noPath
        Dir1.Path = PathName
    ' Return True if no error occurs.
        dirPathExists = True
        Exit Function

    noPath:

    ' If the path can't be accessed,
    ' return False.
    dirPathExists = False

End Function   ' dirPathExists
```

CURRICULUM VITAE

Name:	Sarinthip Promrit
Date of Birth	December 6, 1974
Place of Birth	Sukhothai, Thailand.

EDUCATION BACKGROUND:

1992-1995	B.Sc. Agriculture (Horticulture) Faculty of Agriculture, Chiang Mai University, Chiang Mai, Thailand
1996-2000	M.S. Agriculture (Agricultural Systems) Graduate School, Chiang Mai University, Chiang Mai, Thailand

SCHOLARSHIP:

1996-2000 Thailand Research Fund (TRF), through the Development and Validation of Sugarcane Model in Thailand Research Project, Multiple Cropping Center, Chiang Mai University, Thailand

WORK EXPERIENCE:

1996-2000 Research Assistant in the Development and Validation of Sugarcane Model in Thailand Research Project, Multiple Cropping Center, Chiang Mai University, Thailand

HOME:

48/1 Moo 9, Srinakhon Sub-District, Srinakhon District,
Sukhothai, Thailand, 64180 Tel (055) 651082