

TGIS code

[illegible]

Fig A.1 - Flow diagram illustrating the simplified relationships between the modules of the TGIS.

PROGRAM CODE

Public Declarations

This is where variables that are used in multiple forms are declared.

Option Explicit

```
'Variables used by multiple forms etc.
Public pMxDocument As IMxDocument
Public pMap As IMap
Public pActiveView As IActiveView
Public pLayer As ILayer
Public pFeatureLayer As IFeatureLayer
Public pTable As ITable
Public pField As IField
Public pRow As IRow
Public pLayerFields As ILayerFields
Public LayerArray() As Variant
Public intArrayIndex As Integer
Public intArrayCurrent As Integer
Public intListCount As Long
Public strLayerSelect As String
Public strDbTMin As String
Public strDbTMax As String
Public lngDbTMin As Long
Public lngDbTMax As Long
Public lngDispTMin As Long
Public lngDispTMax As Long
Public lngNoOfRows As Long
Public lngFieldMin As Long
Public lngFieldMax As Long
Public lngFieldProb As Long
Public intComplexProb As Integer

'IMxDocument
'IMap interface to get at layers
'IActiveView interface to get at refresh methods
'ILayer
'IFeatureLayer interface to select layer by name
'ITable interface to get into layer table
'IField interface to deal with fields
'IRow interface to get at the rows in the layer table
'ILayerFields interface to get to layer's fields
'Array to hold details of layers loaded in
'Integer to use as array index (next level to add)
'Integer to hold current array level
'Number of fields in list boxes
'String to hold name of currently selected layer
'Tmin data field
'Tmax data field
'Min value in data field
'Max value in data field
'Min time selected
'Max time selected
'Number of rows in the table
'Field index of min field
'Field index of max field
'Field index of prob field
'Index for complex probability (0 or 1)

'Variables for normal probability
Public sngItemMean As Single
Public sngItemStDev As Single
+/-2 St.Dev.)
Public sngWidth As Single
approximating normal curve
Public sngCurrent As Single
Public sngGraphConstant As Single
Public sngGraphPower As Single
Public sngGraphResult As Single
Public pi As Single
Public sngSqr2Pi As Single
Public sngStart As Single
Public sngEnd As Single

'Mean value of item
'Standard deviation of item (assuming min & max are
'Width in fractions of a StDev of rectangles

'Current stage in calculation of normal probabilities
'Graph constant for normal probability
'Graph power for normal probability
'Result achieved on graph
'Pi
'Square root of 2 Pi
'Where to begin calculation of normal probability
'Where to end calculation of normal probability

'Variables for OxCal probability
Public pRelClassColl As IRelationshipClassCollection
interface
Public pRelClass As IRelationshipClass
Public pFeatureClass As IFeatureClass
Public pEnumRelClass As IEnumRelationshipClass
Public pRelRow As IRow
Public pRelObjSet As ISet
Public pRelField As IField
Public dblFieldWidth As Double
Public lngDateField As Long
Public lngOxcalField As Long
Public strDateField As String
Public strOxcalField As String
Public lngEntryCount As Long
Public lngEntry As Long
Public dblMinYear As Double
Public dblMaxYear As Double
Public booDoNotShow As Boolean
Public booOxCalReset As Boolean
dates

'IRelationshipClassCollection
'IRelationshipClass interface
'IFeatureClass interface
'IEnumRelationshipClass interface
'IRow interface
'ISet interface
'IField interface
'Double to hold width of each probability band
'Long integer to hold index of date field
'Long integer to hold index of probability field
'String for date field name
'String for prob field name
'Number of entries for each item
'Counter
'Min year for particular item
'Max year for particular item
'To avoid error when trying to show unloaded form
'To force reset when changing layer and using OxCal
```

```

'Variables used by analyst forms
Public intAnaWhich As Integer
Public intAnaSpace As Integer
Public intAnaProb As Integer
Public lngAnaScale As Long
Public lngAnaMin As Long
Public lngAnaMax As Long
Public lngAnaRange As Long
Public AnaArray() As Double
Public AnaCatsArray() As Long
Public AnaPercArray() As Double
Public lngAnaDivisions As Long
Public strAnaResults As String
Public dblPeriodPercent As Double
Public AnaPeriodArray() As Long
Public lngPeriodCount As Long
Public dblMaxWeight As Double
Public booAnalyst As Boolean
Public strNormAnalystField As String
Public lngNormAnalystField As Long

'Index for which stage (0 or 1)
'Index for spatial scope (0, 1, or 2)
'Index for complex probability (0, 1, or 2)
'Specified scale
'Minimum time
'Maximum time
'Min-max time range
'Array to hold results
'Array to hold temporal divisions
'Array to hold results as a percentage
'Number of divisions (long integer)
'String for result output
'Rate of change % to form periods
'Array to hold period values
'Count of periods
'Maximum weighting
'To record whether to normalise analytical process
'Field name for any normalisation field
'Field number for any normalisation field

```

ThisDocument

This is the code contained within the root of the template document that handles methods associated with launching the TGIS when clicking on the “Launch TGIS” button.

Option Explicit

```

Private Sub TGIS_Launch_Click()

    'Button to launch T-GIS module

    'Will check a layer is selected, save that layer to a variable and then launch loader
    frame

    'Initialise error handler
    On Error GoTo ErrorLaunchClick

    'Firstly, reset all if OxCal probabilities have been used
    Dim intOkAnswer As Integer
    If booOxCalReset = True Then
        intOkAnswer = MsgBox("WARNING: clicking this will cause a data reset", vbOKCancel,
                               "T-GIS")

        If intOkAnswer = 1 Then
            Call frmTgisPanel.ResetProcess
        Else
            Exit Sub
        End If
    End If
    booOxCalReset = False

    'Initialises variables
    Set pMxDocument = ThisDocument
    Set pLayer = pMxDocument.SelectedLayer

    If intArrayIndex = 0 Then
        ReDim LayerArray(9, 0)
        intComplexProb = 0
    End If

    'Obtains name of layer and checks if in array. If it is then loads main frame
    strLayerSelect = pLayer.Name
    'Error 91 will occur if no (or multiple) layer selected

    Dim intCount As Integer
    For intCount = 0 To intArrayIndex
        If LayerArray(0, intCount) = strLayerSelect Then
            'Load main frame
            intArrayCurrent = intCount
            If frmTgisPanel.Visible = False Then
                Load frmTgisPanel
                frmTgisPanel.InitialiseMainForm
                frmTgisPanel.Show vbModeless
            End If
        End If
    Next intCount

    'Sets level of current layer in array
    'If main form is not visible...
    'Loads main form

```

```

        Else
            frmTgisPanel.InitialiseMainForm      'Else reinitialises main form
        End If
    Exit Sub
End If
Next

'Else load loader frame
intArrayCurrent = intArrayIndex      'Sets current array level to array index
Load frmTgisLoader
frmTgisLoader.Show vbModal

Exit Sub

'Error handler
ErrorLaunchClick:

'Error selector
Select Case Err.Number
Case 91      'Triggered if no layer / multiple layers selected
    MsgBox "Select a layer then click again", , "T-GIS"
    Err.Clear
    Exit Sub
Case Else
    MsgBox "Error: " & Err.Number & " - " & Err.Description, , "T-GIS"
    Err.Clear
End Select

Exit Sub

End Sub

```

frmTgisLoader

This is the form called by the “Launch TGIS” method, assuming all checks are positive. It is used to select the two fields in the layer’s attribute table that relate to the minimum and maximum date of each object.

The screenshot shows a Windows-style dialog box titled "Launch T-GIS". Inside, there's a section labeled "Bayesian". Below this, there are two columns of radio buttons. The left column is titled "Select minimum field:" and the right column is titled "Select maximum field:". Both columns have a list of fields: Shape, LABCODE, EASTING, NORTHING, BAYESSTART, BAYESEND, and PHASE. In the left column, "BAYESSTART" is selected. In the right column, "BAYESEND" is selected. At the bottom of the dialog, there is a "Launch" button and an empty text box.

Fig A.2 - Launch T-GIS form.

```

Option Explicit
'General declarations

Private Sub cmdLaunch_Click()

    'Launches the TGIS

    'Checks fields selected in panel, take them, checks they're not the same, checks they
    'contain appropriate data, then launches the main panel, sending on the selected layer
    name

    'Initialise error handler
    On Error GoTo ErrorLaunchClick

    'Declare variables
    Dim strDbTMin As String      'Tmin field
    Dim strDbTMax As String      'Tmax field

    'Assign variables
    Set pTable = pMxDocument.SelectedLayer      'Sets table to current layer

    'Obtains and checks if fields selected are applicable
    Dim intList As Long          'Counter variable
    For intList = 0 To intListCount
        If lstMinField.Selected(intList) = True Then      'If the item is selected...
            strDbTMin = CStr(lstMinField.List(intList))    'write it to the min field
                                                            attribute
        End If
    Next intList

    intList = 0
    For intList = 0 To intListCount
        If lstMaxField.Selected(intList) = True Then      'If the item is selected...
            strDbTMax = CStr(lstMaxField.List(intList))    'write it to the max field
                                                            attribute
        End If
    Next intList

    'Checks fields aren't null or the same
    If strDbTMin = "" Or strDbTMax = "" Then
        MsgBox "Select two fields, try again.", vbOKOnly, "T-GIS"
        Exit Sub

    ElseIf strDbTMin = strDbTMax Then
        MsgBox "No fields selected or selected fields are the same, try again.", vbOKOnly, "
        T-GIS"
        Exit Sub
    End If

    'Checks if values in two fields are all numeric, that values in min are all less than
    'those in max, and obtains the minimum value in min field and maximum value in max field

    lngFieldMin = pTable.FindField(strDbTMin)      'Obtains field index of min field
    lngFieldMax = pTable.FindField(strDbTMax)      'Obtains field index of max field

    'Obtains number of rows in table
    Dim pQueryFilter As IQueryFilter      'Need an empty query to obtain the number of rows
                                          in the table

    Set pQueryFilter = New QueryFilter
    lngNoOfRows = CLng(pTable.RowCount(pQueryFilter))      'Obtains the number of rows

    Dim lngCurrentRow As Long

    Call ProgressBar(lngCurrentRow, (lngNoOfRows - 1))      'Updates progress bar

    Set pRow = pTable.GetRow(lngCurrentRow)      'Gets first row
    'Checks if values are numeric
    If IsNumeric(pRow.Value(lngFieldMin)) = False Or IsNumeric(pRow.Value(lngFieldMax)) =
        False Then
        MsgBox "One of the selected fields contains non-numeric data, try again.", , "T-GIS"
        Exit Sub
    'Then that the min field is lower than the max field

```

```

ElseIf pRow.Value(lngFieldMax) < pRow.Value(lngFieldMin) Then
    MsgBox "Data in maximum field of lower value than minimum field at row " &
        lngCurrentRow & ".", , "T-GIS"
    Exit Sub
'Then sets the initial values for min & max to the values in the first row
Else
    lngDbTMin = pRow.Value(lngFieldMin)
    lngDbTMax = pRow.Value(lngFieldMax)
End If

'Then loops through other rows to do the same checks
lngCurrentRow = 1
For lngCurrentRow = 1 To lngNoOfRows - 1      'Have to subtract one as they start from
                                                0 not 1

    Call ProgressBar(lngCurrentRow, (lngNoOfRows - 1)) 'Updates progress bar

    Set pRow = pTable.GetRow(lngCurrentRow)      'Gets next row
    'Checks if values are numeric
    If IsNumeric(pRow.Value(lngFieldMin)) = False Or IsNumeric(pRow.Value(lngFieldMax)) =
                                                False Then
        MsgBox "One of the selected fields contains non-numeric data, try again.", , "T-
            GIS"
        Exit Sub
    'Then that the min field is lower than the max field
    ElseIf pRow.Value(lngFieldMax) < pRow.Value(lngFieldMin) Then
        MsgBox "Data in maximum field of lower value than minimum field at row " &
            lngCurrentRow & ".", , "T-GIS"
        Exit Sub
    'Then re-sets min & max values if value in row is less / greater than current value
    Else
        If lngDbTMin > pRow.Value(lngFieldMin) Then
            lngDbTMin = pRow.Value(lngFieldMin)
        ElseIf lngDbTMax < pRow.Value(lngFieldMax) Then
            lngDbTMax = pRow.Value(lngFieldMax)
        End If
    End If
Next lngCurrentRow

'Loads data into array
LayerArray(0, intArrayCurrent) = strLayerSelect      'Adds layer name to array
LayerArray(1, intArrayCurrent) = strDbTMin            'Adds min field to array
LayerArray(2, intArrayCurrent) = strDbTMax            'Adds max field to array
LayerArray(3, intArrayCurrent) = lngDbTMin            'Adds min value to array
LayerArray(4, intArrayCurrent) = lngDbTMax            'Adds max value to array
LayerArray(5, intArrayCurrent) = lngDbTMin            'Adds min value to array as
                                                    initial min setting
LayerArray(6, intArrayCurrent) = lngDbTMax            'Adds max value to array as
                                                    initial max setting
LayerArray(7, intArrayCurrent) = "NULL"                'To later hold last calculated
                                                    values
LayerArray(8, intArrayCurrent) = "NULL"                'To later hold last calculated
                                                    values
LayerArray(9, intArrayCurrent) = 0                    'Type of probability calculation
                                                    to use

'Unload frame
Unload Me

'Increase size of array for next layer to be added (i.e. array is always 1 layer bigger
                                                    than used)
intArrayIndex = intArrayIndex + 1                    'Increments array index
ReDim Preserve LayerArray(9, UBound(LayerArray, 2) + 1) 'Resizes array to allow new
                                                    layer

'Load main frame
If frmTgisPanel.Visible = False Then                'If main form is not visible...
    Load frmTgisPanel                                'Loads main form
    frmTgisPanel.Show vbModeless
    frmTgisPanel.InitialiseMainForm
Else
    frmTgisPanel.InitialiseMainForm                    'Else reinitialises main form
End If

Exit Sub

```

```

'Error handler
ErrorLaunchClick:

    ErrorHandler Err.Number
    Err.Clear

Exit Sub

End Sub

Private Sub UserForm_Initialize()

    'Takes the layer selected variable and populate the panels

'Initialise error handler
On Error GoTo ErrorInitialise

    'Obtains variable names associated with selected layer and populate the panels with them

    'Assign variables
    Set pFeatureLayer = pMxDocument.SelectedLayer      'Sets FeatureLayer to current layer
    Set pLayerFields = pFeatureLayer                  'QueryInterface

    'Sets layer caption
    lblLayerSelected1.Caption = strLayerSelect

    'Adds layer fields to list boxes
    intListCount = pLayerFields.FieldCount 'Gets the number of fields belonging to the layer

    Dim intList As Long
    For intList = 0 To intListCount - 1      'Have to subtract one as we start from 0 not 1
        Set pField = pLayerFields.Field(intList) 'Sets the current field to the IField
        lstMinField.AddItem pField.Name        'Assigns field name to min listbox
        lstMaxField.AddItem pField.Name       'Assigns field name to max listbox
    Next intList

Exit Sub

'Error handler
ErrorInitialise:

    ErrorHandler Err.Number
    Err.Clear

Exit Sub

End Sub

'Progress bar subroutine
Private Sub ProgressBar(lngValue As Long, lngTotal As Long)

'Initialise error handler
On Error GoTo ErrorProgress

    'Repaint form
    Me.Repaint

    'Set new bar width
    lblBar.Width = (216 / lngTotal) * lngValue

    'Repaint frame
    fraBar.Repaint

Exit Sub

'Error handler
ErrorProgress:

    ErrorHandler Err.Number
    Err.Clear

Exit Sub

End Sub

```

```

'External subroutine to handle errors
Private Sub ErrorHandler(ErrorNumber As Long)

    'Error selector
    Select Case ErrorNumber
    'Insert specific error cases to catch here
    Case Else
        MsgBox "Error: " & Err.Number & " - " & Err.Description, , "T-GIS"
    End Select

End Sub

```

frmTgisPanel

This is the main user interface for the TGIS: all functionality is accessed from here.

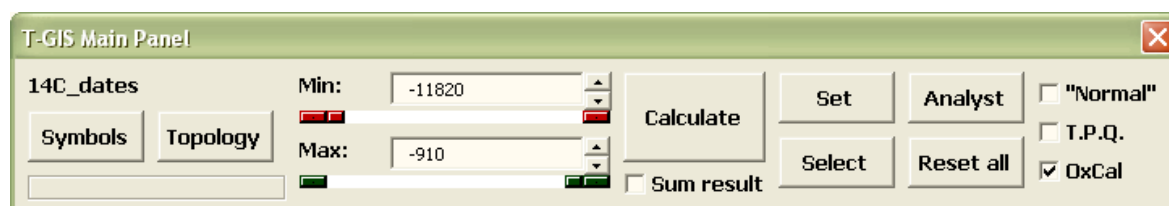


Fig A.3 - Main TGIS user interface.

```

Option Explicit
'General declarations

Private lngFieldTopol As Long           'Field index of topology field
Private booInitialised As Boolean       'Boolean variable to stop adjust methods
                                         'being called before time
Private booCheckBoxes As Boolean        'Boolean variable to stop checkbox click
                                         'methods being called
Private pFeatureLayer As FeatureLayer   'FeatureLayer variable to select layer by
                                         'name
Private pGeoFeatureLayer As IGeoFeatureLayer 'IGeoFeatureLayer interface to access
                                         'renderer
Private pSymbol As ISymbol              'ISymbol interface to set symbol properties
Private pMarkerSymbol As ISimpleMarkerSymbol 'ISimpleMarkerSymbol interface to set symbol
                                         'properties
Private pColor As IRgbColor             'IRgbColor interface to create colour

Private Sub chkComplexProb_Click()
    'Method to deal with complexity tick box

'Initialise error handler
On Error GoTo ErrorChkComplex

    'Abort this method if called from the other checkbox method
    If booCheckBoxes = False Then
        Exit Sub
    End If

    'If box is unticked, set complexity state to 0; if ticked, set to 1
    If chkComplexProb.Value = False Then

        intComplexProb = 0
        LayerArray(9, intArrayCurrent) = 0
    End If
End Sub

```



```

ElseIf chkComplexProb.Value = True Then

    intComplexProb = 1
    LayerArray(9, intArrayCurrent) = 1
    booCheckBoxes = False
    chkTPQ.Value = False
    chkOxcal.Value = False
    booCheckBoxes = True

End If

Exit Sub

'Error handler
ErrorChkComplex:

    ErrorHandler Err.Number
    Err.Clear

Exit Sub

End Sub

Private Sub chkOxcal_Click()

    'Method to deal with Oxcal tick box

'Initialise error handler
On Error GoTo ErrorChkOxcal

'Abort this method if called from the other checkbox method
If booCheckBoxes = False Then
    Exit Sub
End If

'If box is unticked, set complexity state to 0; if ticked, set to 3
If chkOxcal.Value = False Then

    intComplexProb = 0
    LayerArray(9, intArrayCurrent) = 0

ElseIf chkOxcal.Value = True Then

    intComplexProb = 3
    LayerArray(9, intArrayCurrent) = 0      'Don't save it, as need to load in table each
                                           time

    booCheckBoxes = False
    chkComplexProb.Value = False
    chkTPQ.Value = False
    booCheckBoxes = True

    'Then load in table
    booDoNotShow = False
    Load frmTgisOxCal
    If booDoNotShow = False Then frmTgisOxCal.Show      'i.e. show OxCal form if a relate
                                                         exists
    If booDoNotShow = True Then Unload frmTgisOxCal      'i.e. unload OxCal form if no
                                                         relate exists

End If

Exit Sub

'Error handler
ErrorChkOxcal:

    ErrorHandler Err.Number
    Err.Clear

Exit Sub

End Sub

```

```

Private Sub chkTPQ_Click()
    'Method to deal with T.P.Q. tick box

    'Initialise error handler
    On Error GoTo ErrorChkTPQ

    'Abort this method if called from the other checkbox method
    If booCheckBoxes = False Then
        Exit Sub
    End If

    'If box is unticked, set complexity state to 0; if ticked, set to 2
    If chkTPQ.Value = False Then

        intComplexProb = 0
        LayerArray(9, intArrayCurrent) = 0

    ElseIf chkTPQ.Value = True Then

        intComplexProb = 2
        LayerArray(9, intArrayCurrent) = 2
        booCheckBoxes = False
        chkComplexProb.Value = False
        chkOxcal.Value = False
        booCheckBoxes = True

    End If

Exit Sub

'Error handler
ErrorChkTPQ:

    ErrorHandler Err.Number
    Err.Clear

Exit Sub

End Sub

Private Sub cmdAnalyst_Click()

    'Launches analytical tool

    'Initialise error handler
    On Error GoTo ErrorCmdAnalyst

    'Load the spatial selection form of the Analyst tool
    intAnaWhich = 0
    Load frmAnaSpace
    frmAnaSpace.Show

Exit Sub

'Error handler
ErrorCmdAnalyst:

    ErrorHandler Err.Number
    Err.Clear

Exit Sub

End Sub

Private Sub cmdCalculate_Click()

    'Launches probability calculator when clicked

    'Initialise error handler
    On Error GoTo ErrorCmdCalculate

    booAnalyst = False        'Make sure any sum process knows that it isn't being called for
                                the analyst tool

```

```

        'Call probability calculator
        Call ProbCalc

Exit Sub

'Error handler
ErrorCmdCalculate:

    ErrorHandler Err.Number
    Err.Clear

Exit Sub

End Sub

Private Sub cmdResetAll_Click()

    'Resets all relevant variables and closes form

'Initialise error handler
On Error GoTo ErrorCmdReset

    'Declare variables
    Dim intCheck As Integer                                'Integer to obtain result of message box

    intCheck = MsgBox("Are you sure you wish to reset?", vbYesNo, "T-GIS")

    If intCheck = 6 Then

        'Call reset process
        Call ResetProcess

    Else
        Exit Sub
    End If

    'Refresh display and unload main form
    pMxDocument.UpdateContents
    pActiveView.Refresh
    Unload Me
Exit Sub

'Error handler
ErrorCmdReset:

    ErrorHandler Err.Number
    Err.Clear

Exit Sub

End Sub

Private Sub cmdSelect_Click()

    'Selects a group of items over a certain probability

'Initialise error handler
On Error GoTo ErrorCmdSelect

    Load frmTgisSelect
    frmTgisSelect.Show

Exit Sub

'Error handler
ErrorCmdSelect:

    ErrorHandler Err.Number
    Err.Clear

Exit Sub

End Sub

```

```

Private Sub cmdSetSelect_Click()

    'Sets min and max to min and max of current selection

    'Initialise error handler
    On Error GoTo ErrorCmdSet

        'Declare variables
        Dim lngNoOfRowsSelect As Long
        Dim lngCurrentRowSelect As Long
        Dim pTableSelection As ITableSelection

        Dim pSelectionSet As ISelectionSet
        Dim pEnumIDs As IEnumIDs

        Dim lngID As Long
        Dim lngSelectMin As Long
        Dim lngSelectMax As Long

        'Number of rows in selection
        'Current row in selection
        'ITableSelection interface to get selected rows
        'ISelectionSet interface to get selected rows
        'IEnumIDs interface to get OIDs of selected rows
        'OID of current row
        'Minimum value of selection
        'Maximum value of selection

        'Initialise variables
        Set pTableSelection = pTable
        Set pSelectionSet = pTableSelection.SelectionSet
        lngNoOfRowsSelect = pSelectionSet.Count

        'Set selection to current layer
        'Set selection set to that of table
        'Obtain size of selection

        'Exit the subroutine if no features selected
        If lngNoOfRowsSelect = 0 Then
            MsgBox "No items selected", , "T-GIS"
            Exit Sub
        End If

        Set pEnumIDs = pSelectionSet.IDs

        lngID = pEnumIDs.Next

        'Obtain initial values
        Set pRow = pTable.GetRow(lngID)
        lngSelectMin = pRow.Value(lngFieldMin)
        lngSelectMax = pRow.Value(lngFieldMax)

        'Gets row

        lngID = pEnumIDs.Next

        'Then check them against the rest, if any
        Do Until lngID = -1

            Set pRow = pTable.GetRow(lngID)

            'Gets row

            'Sets new min and max if lower/higher than current min and max
            If pRow.Value(lngFieldMin) < lngSelectMin Then lngSelectMin = pRow.Value(lngFieldMin)
            If pRow.Value(lngFieldMax) > lngSelectMax Then lngSelectMax = pRow.Value(lngFieldMax)

            lngID = pEnumIDs.Next

        Loop

        'Then set min and max on main panel
        scrMinT.Value = lngSelectMin
        scrMaxT.Value = lngSelectMax

    Exit Sub

    'Error handler
    ErrorCmdSet:

        ErrorHandler Err.Number
        Err.Clear

    Exit Sub

End Sub

```

```

Private Sub cmdSymbols_Click()

    'Sets symbology to recommended settings for probability

    'Initialise error handler
    On Error GoTo ErrorCmdSymbols

    'Declare variables
    Dim pClassBreaksRenderer As IClassBreaksRenderer      'IClassBreaksRenderer interface
                                                         to set symbology
    Dim pNullColor As IColor                               'IColor Interface to get null
                                                         colour
    Dim pBlackColor As IRgbColor                          'IRgbColor Interface to create
                                                         black

    'Initialise variables
    Set pClassBreaksRenderer = New ClassBreaksRenderer
    Set pGeoFeatureLayer = pFeatureLayer                  'Set IGeoFeatureLayer to current
                                                         layer (QueryInterface)
    Set pTable = pFeatureLayer                            'Set ITable to current layer
                                                         (QueryInterface)

    'Set symbols to use colour gradient over TGIS_Prob field
    pClassBreaksRenderer.Field = "TGIS_Prob" 'Set renderer field to TGIS_Prob

    pClassBreaksRenderer.BreakCount = 11 'Set number of breaks then breaks
    pClassBreaksRenderer.MinimumBreak = 0
    pClassBreaksRenderer.Break(0) = 0
    pClassBreaksRenderer.Break(1) = 0.1
    pClassBreaksRenderer.Break(2) = 0.2
    pClassBreaksRenderer.Break(3) = 0.3
    pClassBreaksRenderer.Break(4) = 0.4
    pClassBreaksRenderer.Break(5) = 0.5
    pClassBreaksRenderer.Break(6) = 0.6
    pClassBreaksRenderer.Break(7) = 0.7
    pClassBreaksRenderer.Break(8) = 0.8
    pClassBreaksRenderer.Break(9) = 0.9
    pClassBreaksRenderer.Break(10) = 1

    Set pMarkerSymbol = New SimpleMarkerSymbol            'Create new simple marker symbol
    Set pNullColor = New RgbColor
    pNullColor.NullColor = True
    pMarkerSymbol.Color = pNullColor                     'Apply colour
    pMarkerSymbol.size = 6                               'Set size
    pMarkerSymbol.Style = esriSMSCircle                  'Set symbol
    pMarkerSymbol.Outline = True                         'Add outline
    Set pBlackColor = New RgbColor                       'Create new colour (black)
    pBlackColor.Red = 0                                  'Set red channel
    pBlackColor.Green = 0                                'Set green channel
    pBlackColor.Blue = 0                                 'Set blue channel
    pMarkerSymbol.OutlineColor = pBlackColor             'Set outline to black
    Set pSymbol = pMarkerSymbol                          'Queryinterface

    pClassBreaksRenderer.Symbol(0) = pSymbol            'Sets symbol
    pClassBreaksRenderer.Label(0) = "0"                 'Sets label

    Dim lngCount As Long
    For lngCount = 1 To 10

        Set pColor = New RgbColor                       'Create new RGB colour
                                                         (Yellow -> Red)
        'If the user wishes to adjust the standard symbology colours, they should adjust
                                                         these values here
        pColor.Red = 255
        pColor.Green = 255 - ((255 / 100) * lngCount ^ 2) 'Set red channel
        pColor.Blue = 0                                  'Set green channel
                                                         'Set blue channel
        Set pMarkerSymbol = New SimpleMarkerSymbol      'Create new simple marker
                                                         symbol
        pMarkerSymbol.Color = pColor                    'Apply colour
        pMarkerSymbol.size = 6                          'Set size
        pMarkerSymbol.Style = esriSMSCircle             'Set symbol
        pMarkerSymbol.Outline = True                   'Add outline
        pMarkerSymbol.OutlineColor = pBlackColor        'Set outline to black
        Set pSymbol = pMarkerSymbol                    'Queryinterface

        pClassBreaksRenderer.Symbol(lngCount) = pSymbol 'Sets symbol
    
```

```

        pClassBreaksRenderer.Label(lngCount) = ">" & ((lngCount - 1) / 10) & " to " &
            (lngCount / 10) 'Sets label

    Next lngCount

    Set pGeoFeatureLayer.Renderer = pClassBreaksRenderer 'Set current layer's
                                                            renderer to class breaks renderer

    'Refresh display
    pMxDocument.UpdateContents
    pActiveView.Refresh

Exit Sub

'Error handler
ErrorCmdSymbols:

    ErrorHandler Err.Number
    Err.Clear

Exit Sub

End Sub

Private Sub cmdTopology_Click()

    'Sets symbology to recommended settings for topology

    'Initialise error handler
    On Error GoTo ErrorCmdTopology

        'Declare variables
        Dim pUniqueValueRenderer As IUniqueValueRenderer 'IUniqueValueRenderer interface
                                                            to set symbology
        Dim pNullColor As IColor 'IColor Interface to get null
                                                            colour
        Dim pBlackColor As IRgbColor 'IRgbColor Interface to create
                                                            black

        'Initialise variables
        Set pUniqueValueRenderer = New UniqueValueRenderer
        Set pGeoFeatureLayer = pFeatureLayer 'Set IGeoFeatureLayer to current
                                                            layer (QueryInterface)
        Set pTable = pFeatureLayer 'Set ITable to current layer
                                                            (QueryInterface)

        'Set symbols to use colour categories for TGIS_Topol field
        pUniqueValueRenderer.FieldCount = 1
        pUniqueValueRenderer.Field(0) = "TGIS_Topol" 'Set renderer field to TGIS_Prob

        Set pMarkerSymbol = New SimpleMarkerSymbol 'Create new simple marker symbol
        Set pNullColor = New RgbColor
        pNullColor.NullColor = True
        pMarkerSymbol.Color = pNullColor 'Apply colour
        pMarkerSymbol.size = 6 'Set size
        pMarkerSymbol.Style = esriSMSCircle 'Set symbol
        pMarkerSymbol.Outline = True 'Add outline
        Set pBlackColor = New RgbColor 'Create new colour (black)
        pBlackColor.Red = 0 'Set red channel
        pBlackColor.Green = 0 'Set green channel
        pBlackColor.Blue = 0 'Set blue channel
        pMarkerSymbol.OutlineColor = pBlackColor 'Set outline to black
        Set pSymbol = pMarkerSymbol 'Queryinterface

        pUniqueValueRenderer.DefaultSymbol = pSymbol 'Sets default symbol
        pUniqueValueRenderer.UseDefaultSymbol = True
        pUniqueValueRenderer.AddValue "Unknown", "", pSymbol 'Sets symbol

        Set pColor = New RgbColor 'Create new RGB colour
        'If the user wishes to adjust the standard symbology colours, they should adjust these
        'values here
        pColor.Red = 200 'Set red channel
        pColor.Green = 0 'Set green channel
        pColor.Blue = 0 'Set blue channel
        Set pMarkerSymbol = New SimpleMarkerSymbol 'Create new simple marker symbol

```

```

pMarkerSymbol.Color = pColor          'Apply colour
pMarkerSymbol.size = 6                 'Set size
pMarkerSymbol.Style = esriSMSCircle   'Set symbol
pMarkerSymbol.Outline = True           'Add outline
pMarkerSymbol.OutlineColor = pBlackColor 'Set outline to black
Set pSymbol = pMarkerSymbol            'Queryinterface

pUniqueValueRenderer.AddValue "Before", "", pSymbol 'Sets symbol

Set pColor = New RgbColor               'Create new RGB colour
'If the user wishes to adjust the standard symbology colours, they should adjust these
                                         values here
pColor.Red = 255                        'Set red channel
pColor.Green = 155                      'Set green channel
pColor.Blue = 0                        'Set blue channel
Set pMarkerSymbol = New SimpleMarkerSymbol 'Create new simple marker symbol
pMarkerSymbol.Color = pColor            'Apply colour
pMarkerSymbol.size = 6                 'Set size
pMarkerSymbol.Style = esriSMSCircle     'Set symbol
pMarkerSymbol.Outline = True           'Add outline
pMarkerSymbol.OutlineColor = pBlackColor 'Set outline to black
Set pSymbol = pMarkerSymbol            'Queryinterface

pUniqueValueRenderer.AddValue "Overlaps Min", "", pSymbol 'Sets symbol

Set pColor = New RgbColor               'Create new RGB colour
'If the user wishes to adjust the standard symbology colours, they should adjust these
                                         values here
pColor.Red = 155                        'Set red channel
pColor.Green = 255                      'Set green channel
pColor.Blue = 155                      'Set blue channel
Set pMarkerSymbol = New SimpleMarkerSymbol 'Create new simple marker symbol
pMarkerSymbol.Color = pColor            'Apply colour
pMarkerSymbol.size = 6                 'Set size
pMarkerSymbol.Style = esriSMSCircle     'Set symbol
pMarkerSymbol.Outline = True           'Add outline
pMarkerSymbol.OutlineColor = pBlackColor 'Set outline to black
Set pSymbol = pMarkerSymbol            'Queryinterface

pUniqueValueRenderer.AddValue "Overlaps All", "", pSymbol 'Sets symbol

Set pColor = New RgbColor               'Create new RGB colour
'If the user wishes to adjust the standard symbology colours, they should adjust these
                                         values here
pColor.Red = 0                          'Set red channel
pColor.Green = 225                      'Set green channel
pColor.Blue = 0                        'Set blue channel
Set pMarkerSymbol = New SimpleMarkerSymbol 'Create new simple marker symbol
pMarkerSymbol.Color = pColor            'Apply colour
pMarkerSymbol.size = 6                 'Set size
pMarkerSymbol.Style = esriSMSCircle     'Set symbol
pMarkerSymbol.Outline = True           'Add outline
pMarkerSymbol.OutlineColor = pBlackColor 'Set outline to black
Set pSymbol = pMarkerSymbol            'Queryinterface

pUniqueValueRenderer.AddValue "Within", "", pSymbol 'Sets symbol

Set pColor = New RgbColor               'Create new RGB colour
'If the user wishes to adjust the standard symbology colours, they should adjust these
                                         values here
pColor.Red = 0                          'Set red channel
pColor.Green = 155                      'Set green channel
pColor.Blue = 255                      'Set blue channel
Set pMarkerSymbol = New SimpleMarkerSymbol 'Create new simple marker symbol
pMarkerSymbol.Color = pColor            'Apply colour
pMarkerSymbol.size = 6                 'Set size
pMarkerSymbol.Style = esriSMSCircle     'Set symbol
pMarkerSymbol.Outline = True           'Add outline
pMarkerSymbol.OutlineColor = pBlackColor 'Set outline to black
Set pSymbol = pMarkerSymbol            'Queryinterface

pUniqueValueRenderer.AddValue "Overlaps Max", "", pSymbol 'Sets symbol

Set pColor = New RgbColor               'Create new RGB colour
'If the user wishes to adjust the standard symbology colours, they should adjust these
                                         values here

```

```

pColor.Red = 0
pColor.Green = 0
pColor.Blue = 200
Set pMarkerSymbol = New SimpleMarkerSymbol
pMarkerSymbol.Color = pColor
pMarkerSymbol.size = 6
pMarkerSymbol.Style = esriSMSCircle
pMarkerSymbol.Outline = True
pMarkerSymbol.OutlineColor = pBlackColor
Set pSymbol = pMarkerSymbol

pUniqueValueRenderer.AddValue "After", "", pSymbol

Set pGeoFeatureLayer.Renderer = pUniqueValueRenderer

'Refresh display
pMxDocument.UpdateContents
pActiveView.Refresh

Exit Sub

'Error handler
ErrorCmdTopology:

    ErrorHandler Err.Number
    Err.Clear

Exit Sub

End Sub

Private Sub scrMaxT_Change()

    'Take the new maxima value

'Initialise error handler
On Error GoTo ErrorScrMax

    If booInitialised = False Then
        Exit Sub
    End If

    lngDispTMax = CLng(scrMaxT.Value)

    'Call method to adjust UI
    Call AdjustMax

Exit Sub

'Error handler
ErrorScrMax:

    ErrorHandler Err.Number
    Err.Clear

Exit Sub

End Sub

Private Sub scrMinT_Change()

    'Take the new minima value

'Initialise error handler
On Error GoTo ErrorScrMin

    If booInitialised = False Then
        Exit Sub
    End If

    lngDispTMin = CLng(scrMinT.Value)

    'Call method to adjust UI
    Call AdjustMin

```



```
Exit Sub

'Error handler
ErrorScrMin:

    ErrorHandler Err.Number
    Err.Clear

Exit Sub

End Sub

Private Sub spnMaxT_Change()

    'Take the new maxima value and rewrite the value in txtMaxT

    'Initialise error handler
    On Error GoTo ErrorSpnMax

    If booInitialised = False Then
        Exit Sub
    End If

    lngDispTMax = CLng(spnMaxT.Value)

    'Call method to adjust UI
    Call AdjustMax

Exit Sub

'Error handler
ErrorSpnMax:

    ErrorHandler Err.Number
    Err.Clear

Exit Sub

End Sub

Private Sub spnMinT_Change()

    'Take the new minima value and rewrite the value in txtMinT

    'Initialise error handler
    On Error GoTo ErrorSpnMin

    If booInitialised = False Then
        Exit Sub
    End If

    lngDispTMin = CLng(spnMinT.Value)

    'Call method to adjust UI
    Call AdjustMin

Exit Sub

'Error handler
ErrorSpnMin:

    ErrorHandler Err.Number
    Err.Clear

Exit Sub

End Sub
```

```
Private Sub txtMaxT_AfterUpdate()  
    'Take the new maxima value  
  
    'Initialise error handler  
    On Error GoTo ErrorTxtMax  
  
    If booInitialised = False Then  
        Exit Sub  
    End If  
  
    'Reset value if non-numeric input entered  
    If IsNumeric(txtMaxT.Value) = False Then  
        txtMaxT.Value = lngDispTMax  
        Exit Sub  
    End If  
  
    lngDispTMax = CLng(txtMaxT.Value)  
  
    'Call method to adjust UI  
    Call AdjustMax  
  
Exit Sub  
  
'Error handler  
ErrorTxtMax:  
  
    ErrorHandler Err.Number  
    Err.Clear  
  
Exit Sub  
  
End Sub  
  
Private Sub txtMinT_AfterUpdate()  
    'Take the new minima value  
  
    'Initialise error handler  
    On Error GoTo ErrorTxtMin  
  
    If booInitialised = False Then  
        Exit Sub  
    End If  
  
    'Reset value if non-numeric input entered  
    If IsNumeric(txtMinT.Value) = False Then  
        txtMinT.Value = lngDispTMin  
        Exit Sub  
    End If  
  
    lngDispTMin = CLng(txtMinT.Value)  
  
    'Call method to adjust UI  
    Call AdjustMin  
  
Exit Sub  
  
'Error handler  
ErrorTxtMin:  
  
    ErrorHandler Err.Number  
    Err.Clear  
  
Exit Sub  
  
End Sub
```

```

'External (public as called from outside form) subroutine to initialise the main form
Public Sub InitialiseMainForm()

    'Take the selected layer and fields, write the layer label and set the minima / maxima
    'as the initial values, then write the prob field to the layer table

'Initialise error handler
On Error GoTo ErrorInitialise

    'Declare variables
    Dim pFieldEdit As IFieldEdit                                'Interface to edit field

    'Sets variables from LayerArray (prob. not necessary but belt & braces)
    strLayerSelect = CStr(LayerArray(0, intArrayCurrent))
    strDbTMin = CStr(LayerArray(1, intArrayCurrent))
    strDbTMax = CStr(LayerArray(2, intArrayCurrent))
    lngDbTMin = CLng(LayerArray(3, intArrayCurrent))
    lngDbTMax = CLng(LayerArray(4, intArrayCurrent))
    lngDispTMin = CLng(LayerArray(5, intArrayCurrent))
    lngDispTMax = CLng(LayerArray(6, intArrayCurrent))
    intComplexProb = CInt(LayerArray(9, intArrayCurrent))

    'Initialise other variables
    booInitialised = False
    Set pMap = pMxDocument.FocusMap                            'Sets map to current map
    Dim intLayer As Long
    For intLayer = 0 To (pMap.LayerCount - 1)
        If UCase(pMap.Layer(intLayer).Name) = UCase(strLayerSelect) Then
            Set pFeatureLayer = pMap.Layer(intLayer)
            Exit For
        End If
    Next intLayer
    Set pTable = pFeatureLayer                                  'Sets table to current layer
    Set pActiveView = pMxDocument.ActiveView                   'Sets current active view
    lngFieldMin = pTable.FindField(strDbTMin)
    lngFieldMax = pTable.FindField(strDbTMax)

    'Populates fields / label
    lblLayerSelected2.Caption = strLayerSelect

    'Initialises checkbox, scrollbars and spinners
    booCheckBoxes = True

    If intComplexProb = 0 Then

        chkComplexProb.Value = False
        chkTPQ.Value = False
        chkOxcal.Value = False

    ElseIf intComplexProb = 1 Then

        chkComplexProb.Value = True

    ElseIf intComplexProb = 2 Then

        chkTPQ.Value = True

    End If

    scrMinT.Min = lngDbTMin
    scrMinT.Max = lngDbTMax
    scrMinT.SmallChange = 1

    scrMaxT.Min = lngDbTMin
    scrMaxT.Max = lngDbTMax
    scrMaxT.SmallChange = 1

    spnMinT.Min = lngDbTMin
    spnMinT.Max = lngDbTMax
    spnMinT.SmallChange = 1

    spnMaxT.Min = lngDbTMin
    spnMaxT.Max = lngDbTMax
    spnMaxT.SmallChange = 1

```

```

'Adds probability field to table
lngFieldProb = pTable.FindField("TGIS_Prob")
If lngFieldProb = -1 Then
    Set pField = New Field
    Set pFieldEdit = pField
    pFieldEdit.Name = "TGIS_Prob"
    pFieldEdit.Type = esriFieldTypeSingle
    pTable.AddField pField
    lngFieldProb = pTable.FindField("TGIS_Prob")
End If

'Obtain field number of TGIS_Prob
'If field does not exist
'Create new field
'Queryinterface
'Set name of field
'And data type
'Add new field
'And set field number to that field

'Adds topology field to table
lngFieldTopol = pTable.FindField("TGIS_Topol")
If lngFieldTopol = -1 Then
    Set pField = New Field
    Set pFieldEdit = pField
    pFieldEdit.Name = "TGIS_Topol"
    pFieldEdit.Type = esriFieldTypeString
    pTable.AddField pField
    lngFieldTopol = pTable.FindField("TGIS_Topol")
End If

'Obtain field number of TGIS_Topol
'If field does not exist
'Create new field
'Queryinterface
'Set name of field
'And data type
'Add new field
'And set field number to that field

Dim pQueryFilter As IQueryFilter
'Need an empty query to obtain the number of rows
'           in the table

Set pQueryFilter = New QueryFilter
lngNoOfRows = CLng(pTable.RowCount(pQueryFilter))
lngNoOfRows = lngNoOfRows - 1
'Obtains the number of rows
'Then subtract one as we start from
'           row 0

'Finally, sets initial text values
booInitialised = True
scrMinT.Value = CStr(lngDispTMin)
scrMaxT.Value = CStr(lngDispTMax)

Exit Sub

'Error handler
ErrorInitialise:

    ErrorHandler Err.Number
    Err.Clear

Exit Sub

End Sub

'External subroutine to adjust the graphical elements of the UI when min changed
Public Sub AdjustMin()

    'Adjusts text in text boxes (and scrollbars / graphical display of time?) if adjustments
    'made to dates
    'then calls probability calculator

'Initialise error handler
On Error GoTo ErrorAdjustMin

'Performs checks on selected values
If lngDispTMin < lngDbTMin Then
    lngDispTMin = lngDbTMin
End If

If lngDispTMin >= lngDbTMax Then
    lngDispTMin = lngDbTMax - 1
End If

If lngDispTMin >= lngDispTMax Then
    lngDispTMax = lngDispTMin + 1
End If

'Sets new values in text boxes to main min & max values
LayerArray(5, intArrayCurrent) = lngDispTMin
LayerArray(6, intArrayCurrent) = lngDispTMax
txtMinT.Value = CStr(lngDispTMin)
txtMaxT.Value = CStr(lngDispTMax)
scrMinT.Value = lngDispTMin
scrMaxT.Value = lngDispTMax

```

```

        spnMinT.Value = lngDispTMin
        spnMaxT.Value = lngDispTMax

Exit Sub

'Error handler
ErrorAdjustMin:

        ErrorHandler Err.Number
        Err.Clear

Exit Sub

End Sub

'External subroutine to adjust the graphical elements of the UI when max changed
Public Sub AdjustMax()

        'Adjusts text in text boxes (and scrollbars / graphical display of time?) if adjustments
                                                made to dates
        'then calls probability calculator

'Initialise error handler
On Error GoTo ErrorAdjustMax

        'Performs checks on selected values

        If lngDispTMax <= lngDbTMin Then
                lngDispTMax = lngDbTMin + 1
        End If

        If lngDispTMax > lngDbTMax Then
                lngDispTMax = lngDbTMax
        End If

        If lngDispTMax <= lngDispTMin Then
                lngDispTMin = lngDispTMax - 1
        End If

        'Sets new values in text boxes to main min & max values
        LayerArray(5, intArrayCurrent) = lngDispTMin
        LayerArray(6, intArrayCurrent) = lngDispTMax
        txtMinT.Value = CStr(lngDispTMin)
        txtMaxT.Value = CStr(lngDispTMax)
        scrMinT.Value = lngDispTMin
        scrMaxT.Value = lngDispTMax
        spnMinT.Value = lngDispTMin
        spnMaxT.Value = lngDispTMax

Exit Sub

'Error handler
ErrorAdjustMax:

        ErrorHandler Err.Number
        Err.Clear

Exit Sub

End Sub

'External subroutine to rewrite the prob field
Public Sub ProbCalc()

        'Calculates probability for each item in the layer and writes to its table

'Initialise error handler
On Error GoTo ErrorProbCalc

        'Declare variables
        'General variables
        Dim sngDispRange As Single           'Display range
        Dim sngDateRange As Single          'Date range
        Dim sngProb As Single                'Probability calculation
        Dim lngCurrentRow As Long            'Row counter

```

```

Dim sngItemTMin As Single      'TMin of item
Dim sngItemTMax As Single     'TMax of item
Dim sngDispTMin As Single     'TMin of display
Dim sngDispTMax As Single     'TMax of display
Dim strTopology As String     'Temporal topology of date
'Variables for T.P.Q. probability
Dim sngFactor1 As Single      'First factor
Dim sngFactor2 As Single      'Second factor
'Variables for Oxcal probability
Dim dblCurrentYear As Double  'Current year
Dim dblCurrentProb As Double  'Current probability
Dim dblRemainder As Double    'Remainder when category overlaps

'Assign variables
sngDispTMin = CSng(txtMinT.Value)
sngDispTMax = CSng(txtMaxT.Value)
sngDispRange = CSng(lngDispTMax) - lngDispTMin
LayerArray(7, intArrayCurrent) = sngDispTMin      'Writes min value to array to record last
                                                    calculated values
LayerArray(8, intArrayCurrent) = sngDispTMax      'Writes max value to array to record last
                                                    calculated values

sngWidth = 0.05                '0.05 StDev seems to achieve accurate probability to > 2 s.f.
pi = 4 * Atn(1)                'Define pi
sngSqr2Pi = Sqr(2 * pi)        'Define the square root of 2 Pi

'Cycle through layer items and set probability field for each
For lngCurrentRow = 0 To lngNoOfRows

    Call ProgressBar(lngCurrentRow, lngNoOfRows)  'Updates progress bar

    Set pRow = pTable.GetRow(lngCurrentRow)        'Gets row
    sngItemTMin = CSng(pRow.Value(lngFieldMin))
    sngItemTMax = CSng(pRow.Value(lngFieldMax))
    sngDateRange = sngItemTMax - sngItemTMin

    If intComplexProb = 0 Then                    'i.e. use simple probability

        If sngItemTMin = sngItemTMax Then
            If sngItemTMin >= sngDispTMin And sngItemTMin <= sngDispTMax Then
                sngProb = 1
                strTopology = "Within"
            ElseIf sngItemTMin < sngDispTMin Then
                sngProb = 0
                strTopology = "Before"
            ElseIf sngItemTMin > sngDispTMax Then
                sngProb = 0
                strTopology = "After"
            Else
                sngProb = 0
                strTopology = "Unknown"
            End If
        ElseIf sngDispTMin <= sngItemTMin And sngDispTMax >= sngItemTMax Then
            sngProb = 1
            strTopology = "Within"
        ElseIf sngDispTMin > sngItemTMax Then
            sngProb = 0
            strTopology = "Before"
        ElseIf sngDispTMax < sngItemTMin Then
            sngProb = 0
            strTopology = "After"
        ElseIf sngDispTMin <= sngItemTMin And sngDispTMax < sngItemTMax Then
            sngProb = (sngDispTMax - sngItemTMin) / sngDateRange
            strTopology = "Overlaps Max"
        ElseIf sngDispTMin > sngItemTMin And sngDispTMax >= sngItemTMax Then
            sngProb = (sngItemTMax - sngDispTMin) / sngDateRange
            strTopology = "Overlaps Min"
        ElseIf sngDispTMin > sngItemTMin And sngDispTMax < sngItemTMax Then
            sngProb = sngDispRange / sngDateRange
            strTopology = "Overlaps All"
        Else
            sngProb = 0
            strTopology = "Unknown"
        End If

    ElseIf intComplexProb = 1 Then                'i.e. use complex probability

```

```

'Calculate item mean and standard deviation
sngItemStDev = sngDateRange / 4           'if +/-2 St.Dev, this is thus
                                           1/4 of the date range
sngItemMean = sngItemTMin + (sngDateRange / 2) 'if +/-2 St.Dev, half the
                                           date range above the min

'Then calculate the probability from sngDispTMin to sngDispTMax
If sngItemTMin = sngItemTMax Then
    If sngItemTMin >= sngDispTMin And sngItemTMin <= sngDispTMax Then
        sngProb = 1
        strTopology = "Within"
    ElseIf sngItemTMin < sngDispTMin Then
        sngProb = 0
        strTopology = "Before"
    ElseIf sngItemTMin > sngDispTMax Then
        sngProb = 0
        strTopology = "After"
    Else
        sngProb = 0
        strTopology = "Unknown"
    End If
ElseIf sngDispTMin > (sngItemMean + (3 * sngItemStDev)) Then
    'i.e. date falls outside the ±3StDev (99.7%) range
    '(as we are only recording probabilities to 2 s.f.
    sngProb = 0
    strTopology = "Before"
ElseIf sngDispTMax < (sngItemMean - (3 * sngItemStDev)) Then
    'i.e. date falls outside the ±3StDev (99.7%) range
    '(as we are only recording probabilities to 2 s.f.
    sngProb = 0
    strTopology = "After"
Else
    'Define variables
    sngGraphConstant = 1 / (sngSqr2Pi * sngItemStDev)

    If sngDispTMin > (sngItemMean - (sngItemStDev * 3)) Then
        sngStart = sngDispTMin
    Else
        sngStart = (sngItemMean - (sngItemStDev * 3))
    End If
    If sngDispTMax < (sngItemMean + (sngItemStDev * 3)) Then
        sngEnd = sngDispTMax
    Else
        sngEnd = (sngItemMean + (sngItemStDev * 3))
    End If

    sngCurrent = sngStart
    sngProb = 0

    'Then calculate probability from sngStart to sngEnd
    Do Until sngCurrent >= sngEnd
        'Calculate graph power
        sngGraphPower = ((sngCurrent - sngItemMean) ^ 2) / (2 * (sngItemStDev ^
                                                                2))

        'Then the result
        sngGraphResult = sngGraphConstant * Exp(-sngGraphPower)
        'Then add to cumulative probability
        sngProb = sngProb + (sngGraphResult * (sngWidth * sngItemStDev))
        'Then increment loop position
        sngCurrent = sngCurrent + (sngItemStDev * sngWidth)
    Loop

    'Assign topology value
    If sngDispTMin > (sngItemMean - (sngItemStDev * 3)) And sngDispTMax <
        (sngItemMean + (sngItemStDev * 3)) Then
        strTopology = "Overlaps All"
    ElseIf sngDispTMin > (sngItemMean - (sngItemStDev * 3)) And sngDispTMax >
        (sngItemMean + (sngItemStDev * 3)) Then
        strTopology = "Overlaps Min"
    ElseIf sngDispTMin < (sngItemMean - (sngItemStDev * 3)) And sngDispTMax <
        (sngItemMean + (sngItemStDev * 3)) Then
        strTopology = "Overlaps Max"
    ElseIf sngDispTMin < (sngItemMean - (sngItemStDev * 3)) And sngDispTMax >
        (sngItemMean + (sngItemStDev * 3)) Then
        strTopology = "Within"

```

```

Else
    strTopology = "Unknown"
End If

End If

ElseIf intComplexProb = 2 Then          'i.e. use terminus post quem probability

    'Calculate the probability from sngDispTMin to sngDispTMax
    If sngItemTMin = sngItemTMax Then
        If sngItemTMin >= sngDispTMin And sngItemTMin <= sngDispTMax Then
            sngProb = 1
            strTopology = "Within"
        ElseIf sngItemTMin < sngDispTMin Then
            sngProb = 0
            strTopology = "Before"
        ElseIf sngItemTMin > sngDispTMax Then
            sngProb = 0
            strTopology = "After"
        Else
            sngProb = 0
            strTopology = "Unknown"
        End If
    ElseIf sngDispTMin <= sngItemTMin And sngDispTMax >= sngItemTMax Then
        sngProb = 1
        strTopology = "Within"
    ElseIf sngDispTMin > sngItemTMax Then
        sngProb = 0
        strTopology = "Before"
    ElseIf sngDispTMax < sngItemTMin Then
        sngProb = 0
        strTopology = "After"
    ElseIf sngDispTMin <= sngItemTMin And sngDispTMax < sngItemTMax Then
        'i.e. lefthand portion of triangle is probability
        sngFactor1 = sngItemTMax - sngDispTMax
        sngProb = 1 - (sngFactor1 ^ 2) * (1 / (sngDateRange ^ 2))
        strTopology = "Overlaps Max"
    ElseIf sngDispTMin > sngItemTMin And sngDispTMax >= sngItemTMax Then
        'i.e. righthand portion of triangle is probability
        sngFactor1 = sngItemTMax - sngDispTMin
        sngProb = (sngFactor1 ^ 2) * (1 / (sngDateRange ^ 2))
        strTopology = "Overlaps Min"
    ElseIf sngDispTMin > sngItemTMin And sngDispTMax < sngItemTMax Then
        'i.e. middle portion of triangle is probability
        sngFactor1 = sngItemTMax - sngDispTMin
        sngFactor2 = sngItemTMax - sngDispTMax
        sngProb = ((sngFactor1 ^ 2) * (1 / (sngDateRange ^ 2))) - ((sngFactor2 ^ 2) * (1 / (sngDateRange ^ 2)))
        strTopology = "Overlaps All"
    Else
        sngProb = 0
        strTopology = "Unknown"
    End If

ElseIf intComplexProb = 3 Then          'i.e. use OxCal probability

    sngProb = 0                        'Start from an assumption of 0
    dblMinYear = 3.1415                'Set to Pi so we know if it's new or not (as year
                                      unlikely to be Pi!)
    dblMaxYear = 3.1415                'Set to Pi so we know if it's new or not

    Set pRelObjSet = pRelClass.GetObjectsRelatedToObject(pRow)
    lngEntryCount = pRelObjSet.Count
    lngEntry = 0

    For lngEntry = 0 To lngEntryCount - 1

        Set pRelRow = pRelObjSet.Next
        dblCurrentYear = pRelRow.Value(lngDateField)
        dblCurrentProb = pRelRow.Value(lngOxcalField)
    
```



```

'If the year fits within the current date range
If dblCurrentYear > sngDispTMin - (dblFieldWidth / 2) And dblCurrentYear <
    sngDispTMax + (dblFieldWidth / 2) Then

    If dblCurrentYear < sngDispTMin Then

        dblRemainder = sngDispTMin - dblCurrentYear
        dblCurrentProb = dblCurrentProb * (dblRemainder / dblFieldWidth)

    ElseIf dblCurrentYear > sngDispTMax Then

        dblRemainder = dblCurrentYear - sngDispTMax
        dblCurrentProb = dblCurrentProb * (dblRemainder / dblFieldWidth)

    ElseIf (dblCurrentYear - (dblFieldWidth / 2)) < sngDispTMin Then

        'i.e. if +- date width / 2 overlaps our displayed min
        dblRemainder = dblCurrentYear - sngDispTMin + (dblFieldWidth / 2)
        dblCurrentProb = (dblCurrentProb / dblFieldWidth) * dblRemainder

    ElseIf (dblCurrentYear + (dblFieldWidth / 2)) > sngDispTMax Then

        'i.e. if +- date width / 2 overlaps our displayed max
        dblRemainder = sngDispTMax - dblCurrentYear + (dblFieldWidth / 2)
        dblCurrentProb = (dblCurrentProb / dblFieldWidth) * dblRemainder

    End If

    'Add on the cumulative probability
    sngProb = sngProb + dblCurrentProb

Else
    'Do nothing as it is outside the date range
End If

'Then check if this is the current min or max year
If dblMinYear = 3.1415 And dblMaxYear = 3.1415 Then
    'i.e. populate variables if this is the first field
    dblMinYear = dblCurrentYear
    dblMaxYear = dblCurrentYear
ElseIf dblCurrentYear < dblMinYear Then
    dblMinYear = dblCurrentYear
ElseIf dblCurrentYear > dblMaxYear Then
    dblMaxYear = dblCurrentYear
Else
    'Do nothing
End If

Next lngEntry

'Then multiply by date width
sngProb = sngProb * dblFieldWidth

'Then adjust min & max by half the date span
dblMinYear = dblMinYear - (dblFieldWidth / 2)
dblMaxYear = dblMaxYear + (dblFieldWidth / 2)

'Then work out the temporal topology
If sngDispTMax < dblMinYear Then
    strTopology = "After"
ElseIf sngDispTMin > dblMaxYear Then
    strTopology = "Before"
ElseIf sngDispTMin > dblMinYear And sngDispTMax < dblMaxYear Then
    strTopology = "Overlaps All"
ElseIf sngDispTMin < dblMinYear And sngDispTMax > dblMaxYear Then
    strTopology = "Within"
ElseIf dblMinYear < sngDispTMin And dblMaxYear > sngDispTMin Then
    strTopology = "Overlaps Min"
    If sngProb < 0.01 Then strTopology = "Before" 'To counter OxCal's output of
                                                0 prob years
    If sngProb > 0.99 Then strTopology = "Within" 'To counter OxCal's output of
                                                0 prob years
ElseIf dblMinYear < sngDispTMax And dblMaxYear > sngDispTMax Then
    strTopology = "Overlaps Max"
    If sngProb < 0.01 Then strTopology = "After" 'To counter OxCal's output of
                                                0 prob years

```

```

        If sngProb > 0.99 Then strTopology = "Within" 'To counter OxCal's output of
                                                    0 prob years
        Else
            strTopology = "Unknown"
        End If

    End If

    pRow.Value(lngFieldProb) = FormatNumber(sngProb, 2)
    pRow.Value(lngFieldTopol) = strTopology
    pRow.Store

Next lngCurrentRow

lblBar2.Width = 0 'Resets progress bar

'Refresh Display
pMxDocument.UpdateContents
pActiveView.Refresh

'If sum checkbox is clicked
If chkProbSum.Value = True Then

    'Launch summed probability layer form
    Load frmTgisProbSum
    frmTgisProbSum.Show

End If

Exit Sub

'Error handler
ErrorProbCalc:

    ErrorHandler Err.Number
    Err.Clear

Exit Sub

End Sub

'External (public) subroutine to handle selection
Public Sub SelectItems(sngSelect As Single)

'Initialise error handler
On Error GoTo ErrorSelect

    'Declare variables
    Dim pQueryFilter As IQueryFilter 'IQueryFilter interface to query layer
    Dim pFeatureSelection As IFeatureSelection 'IFeatureSelection interface to select
                                                    features
    Dim strQuery As String

    'Create QueryFilter to assess layer
    Set pQueryFilter = New QueryFilter 'New query filter
    strQuery = "TGIS_Prob >= " & sngSelect 'Writes desired query to string
    pQueryFilter.SubFields = "TGIS_Prob" 'Sets query field
    pQueryFilter.WhereClause = strQuery 'Sets query clause

    'Create FeatureSelection to select items
    Set pFeatureSelection = pFeatureLayer 'New selection
    pFeatureSelection.SelectFeatures pQueryFilter, esriSelectionResultNew, False 'Selects
                                                    according to query

    'Refresh Display
    pMxDocument.UpdateContents
    pActiveView.Refresh

Exit Sub

```

```

'Error handler
ErrorSelect:

    ErrorHandler Err.Number
    Err.Clear

Exit Sub

End Sub

'Progress bar subroutine
Private Sub ProgressBar(lngValue As Long, lngTotal As Long)

'Initialise error handler
On Error GoTo ErrorProgress

'Repaint form
'Me.Repaint

'Set new bar width
lblBar2.Width = (120 / lngTotal) * lngValue

'Repaint frame
fraBar2.Repaint

Exit Sub

'Error handler
ErrorProgress:

    ErrorHandler Err.Number
    Err.Clear

Exit Sub

End Sub

'External subroutine to handle reset process
Public Sub ResetProcess()

'Initialise error handler
On Error GoTo ErrorReset

'Declare variables
Dim pSimpleRenderer As ISimpleRenderer      'ISimpleRenderer interface to set single
                                             symbol for layer

'Remove prob field from layer attributes
intArrayCurrent = 0

For intArrayCurrent = 0 To (intArrayIndex - 1)      'Loops through each layer in
                                                    array

    strLayerSelect = LayerArray(0, intArrayCurrent) 'Obtains name of layer

    Dim intLayer As Long
    For intLayer = 0 To (pMap.LayerCount - 1)      'Loops through layers in map
        If UCase(pMap.Layer(intLayer).Name) = UCase(strLayerSelect) Then
            Set pFeatureLayer = pMap.Layer(intLayer) 'Selects layer by name
            Exit For                                'Then exits loop
        End If
    Next intLayer

    Set pTable = pFeatureLayer                    'Sets table to current
    Dim pFields As IFields                        'IFields interface
    Set pFields = pTable.Fields                  'Set to fields in layer table
    Set pField = pFields.Field(pFields.FindField("TGIS_Prob")) 'Finds probability field
    pTable.DeleteField pField                    'And deletes it
    Set pField = pFields.Field(pFields.FindField("TGIS_Topol")) 'Finds topologyfield
    pTable.DeleteField pField                    'And deletes it

    'Then "reset" symbology
    Dim intColor As Integer                      'Integer variable to calculate
                                                    colour

```

```

Set pColor = New RgbColor                                'Create new RGB colour
If intArrayIndex > 1 Then
    intColor = intArrayCurrent * (255 / (intArrayIndex - 1))    'Adjustment factor
                                                                for colour
Else
    intColor = 100
End If
If intColor > 255 Then intColor = 255                    'Simple checks to avoid <0 and
                                                         >255
If intColor < 1 Then intColor = 0                        'Simple checks to avoid <0 and
                                                         >255

pColor.Red = 255 - (intColor)                            'Set red channel
pColor.Green = 175                                       'Set green channel
pColor.Blue = 0 + (intColor)                             'Set blue channel
Set pMarkerSymbol = New SimpleMarkerSymbol              'Create new simple marker symbol
pMarkerSymbol.Color = pColor                            'Apply colour
pMarkerSymbol.size = 6                                  'Set size
pMarkerSymbol.Style = esriSMSCircle                    'Set symbol
pMarkerSymbol.Outline = True                            'Add outline
Set pColor = New RgbColor                               'Create new colour (black)
pColor.Red = 0                                           'Set red channel
pColor.Green = 0                                         'Set green channel
pColor.Blue = 0                                          'Set blue channel
pMarkerSymbol.OutlineColor = pColor                    'Set outline to black
Set pSymbol = pMarkerSymbol                             'Queryinterface

Set pSimpleRenderer = New SimpleRenderer                'Create new simple renderer
pSimpleRenderer.Label = ""                              'Set label to null
pSimpleRenderer.Description = ""                        'Set description to null
Set pSimpleRenderer.Symbol = pSymbol                   'Set symbol to pSymbol

Set pGeoFeatureLayer = pFeatureLayer                    'Set GeoFeatureLayer to current
                                                         layer (QueryInterface)
Set pGeoFeatureLayer.Renderer = pSimpleRenderer        'Set current layer's renderer to
                                                         simple renderer

Next intArrayCurrent

ReDim LayerArray(9, 0)    'Reset layer array
intArrayIndex = 0         'Reset array index
booOxCalReset = False    'And switch off the auto-reset

'Refresh display and unload main form
pMxDocument.UpdateContents
pActiveView.Refresh
Unload Me

Exit Sub

'Error handler
ErrorReset:

    ErrorHandler Err.Number
    Err.Clear

Exit Sub

End Sub

'External subroutine to handle errors
Private Sub ErrorHandler(ErrorNumber As Long)

    'Error selector
    Select Case ErrorNumber
    'Insert specific error cases to catch here
    Case Else
        MsgBox "Error: " & Err.Number & " - " & Err.Description, , "T-GIS"
    End Select

End Sub

```

frmTgisOxCal

This is the form called when the OxCal checkbox is ticked and allows the user to select the fields that contain OxCal probabilities from the table related to the main layer table.

Fig A.4 - OxCal probabilities selection form.

```

Option Explicit
'General declarations
Private booCancel As Boolean           'Boolean to tell if form is cancelled or successfully
closed
Private booSuccess As Boolean          'Boolean to tell if selection successfully made
Private intRelFieldCount As Integer    'Integer to hold number of fields in related table

Private Sub cmdEnter_Click()
    'Subroutine triggered by Enter button

    'Initialise error handler
    On Error GoTo ErrorEnter

    'Declare variables
    Dim dblThisYear As Double           'Current year when iterating through particular item

    'Initialise variables
    booSuccess = False

    'Obtains and checks fields selected
    Dim intRelField As Long
    For intRelField = 0 To intRelFieldCount - 1
        If lstDateField.Selected(intRelField) = True Then      'If the item is selected...
            strDateField = CStr(lstDateField.List(intRelField)) 'write it to the min field
                                                                attribute
        End If
        If lstProbField.Selected(intRelField) = True Then      'If the item is selected...
            strOxcalField = CStr(lstProbField.List(intRelField)) 'write it to the max
                                                                field attribute
        End If
    Next intRelField

    'Checks fields aren't null or the same
    If strDateField = "" Or strOxcalField = "" Then
        MsgBox "Select two fields, try again.", vbOKOnly, "T-GIS"
        Exit Sub
    ElseIf strDateField = strOxcalField Then
        MsgBox "No fields selected or selected fields are the same, try again.", vbOKOnly,
            "T-GIS"
        Exit Sub
    End If

    'Obtain first row in layer table
    Set pRow = pTable.GetRow(0)

```

```

'Then get the first row associated with it in related table
Set pRelObjSet = pRelClass.GetObjectsRelatedToObject(pRow)
Set pRelRow = pRelObjSet.Next

lngDateField = pRelRow.Fields.FindField(strDateField)      'Obtains field index of date
                                                            field
lngOxcalField = pRelRow.Fields.FindField(strOxcalField)    'Obtains field index of prob
                                                            field

'Then calculate the width of each item in the date field (in years)
lngEntryCount = pRelObjSet.Count

dblThisYear = pRelRow.Value(lngDateField)
dblMinYear = dblThisYear
dblMaxYear = dblThisYear

For lngEntry = 1 To lngEntryCount - 1

    Set pRelRow = pRelObjSet.Next
    dblThisYear = pRelRow.Value(lngDateField)

    If dblThisYear < dblMinYear Then dblMinYear = dblThisYear
    If dblThisYear > dblMaxYear Then dblMaxYear = dblThisYear

Next lngEntry

dblFieldWidth = (dblMaxYear - dblMinYear) / (lngEntryCount - 1)

'NB: No checks to be run on whether the fields are numeric, as they will be rather huge

'Then close this form
strDateField = ""
strOxcalField = ""
booCancel = False
booSuccess = True
booOxCalReset = True    'To force reset when changing layer and using OxCal dates
Call CloseForm

Exit Sub

'Error handler
ErrorEnter:

    ErrorHandler Err.Number
    Err.Clear

Exit Sub

End Sub

Private Sub UserForm_Terminate()
    'Subroutine called when form closed

    'Initialise error handler
    On Error GoTo ErrorTerminate

    Call CloseForm

Exit Sub

'Error handler
ErrorTerminate:

    ErrorHandler Err.Number
    Err.Clear

Exit Sub

End Sub

```

```

Private Sub UserForm_Initialize()
    'Subroutine used for form initialisation

    'Initialise error handler
    On Error GoTo ErrorInitialise

        'Initialise variables
        booCancel = True
        Set pRelClassColl = pFeatureLayer
        Set pEnumRelClass = pRelClassColl.RelationshipClasses
        Set pRelClass = pEnumRelClass.Next 'Takes the first relationship available: i.e. Oxcal
                                           dates have to be within that

        'If no relationship applies to current layer, warn user and exit subroutine
        If pRelClass Is Nothing Then
            MsgBox "OxCal probability function requires a relationship table to exist for current
                layer.", , "T-GIS"

            Call CloseForm
            Exit Sub
        End If

        'Obtain first row in layer table
        Set pRow = pTable.GetRow(0)

        'Then get the first row associated with it in related table
        Set pRelObjSet = pRelClass.GetObjectsRelatedToObject(pRow)
        Set pRelRow = pRelObjSet.Next

        'Get count of fields in related table
        intRelFieldCount = pRelRow.Fields.FieldCount

        'Then populate listboxes with fields
        Dim intRelField As Long
        For intRelField = 0 To intRelFieldCount - 1 'Have to subtract one as we start from 0
                                                    not 1
            Set pRelField = pRelRow.Fields.Field(intRelField) 'Sets the current field to the
                                                                IField
            lstDateField.AddItem pRelField.Name 'Assigns field name to min listbox
            lstProbField.AddItem pRelField.Name 'Assigns field name to max listbox
        Next intRelField

    Exit Sub

    'Error handler
    ErrorInitialise:

        ErrorHandler Err.Number
        Err.Clear

    Exit Sub

End Sub

Private Sub CloseForm()
    'Subroutine called when form closed

    'Initialise error handler
    On Error GoTo ErrorClose

        If booSuccess = True Then
            Unload Me 'Unload this form if all successful
            Exit Sub
        End If
        If booCancel = True Then frmTgisPanel.chkOxcal.Value = False
        booCancel = False
        booDoNotShow = True 'Make sure the main form does not try to load the
                            OxCal form

    Exit Sub

```

```

'Error handler
ErrorClose:

    ErrorHandler Err.Number
    Err.Clear

Exit Sub

End Sub

'External subroutine to handle errors
Private Sub ErrorHandler(ErrorNumber As Long)

    'Error selector
    Select Case ErrorNumber
    'Insert specific error cases to catch here
    Case Else
        MsgBox "Error: " & Err.Number & " - " & Err.Description, , "T-GIS"
    End Select

End Sub

```

frmTgisSelect

This is the form launched by the “Select” button, the use of which enables the selection of objects in the current layer with a probability greater than the set value.

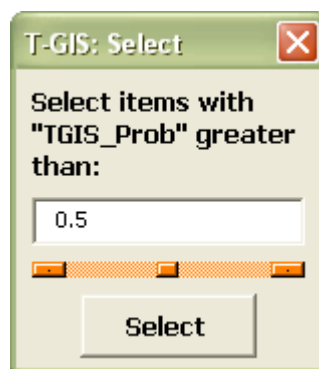


Fig A.5 - Selection based on probability form

```

Option Explicit
'General declarations

Private Sub cmdSelect_Click()
    'Adds items to selection with probability greater than that set in textbox / on scrollbar

    'Initialise error handler
    On Error GoTo ErrorCmdSelect

    'Declare variables
    Dim sngSelect As Single

    'Checks value in text box is numeric
    If IsNumeric(txtProb.Value) = False Then
        MsgBox "Enter a numeric value between 0 and 1.", , "T-GIS"
        txtProb.Value = 0
        Exit Sub
    End If

    'Obtains selection value
    sngSelect = txtProb.Value

    'Closes form
    Unload Me

```



```

        'Launches item select method in main panel
        frmTgisPanel.SelectItems (sngSelect)

Exit Sub

'Error handler
ErrorCmdSelect:

    ErrorHandler Err.Number
    Err.Clear

Exit Sub

End Sub

Private Sub scrProb_Change()
    'Adjusts text box

'Initialise error handler
On Error GoTo ErrorScrChange

    'Sets text box value
    txtProb.Value = scrProb.Value / 10

Exit Sub

'Error handler
ErrorScrChange:

    ErrorHandler Err.Number
    Err.Clear

Exit Sub

End Sub

Private Sub txtProb_Change()
    'Checks validity and adjusts scrollbars

'Initialise error handler
On Error GoTo ErrorTxtChange

    'Checks if value in field is numeric
    If IsNumeric(txtProb.Value) = False Then
        Exit Sub
    End If

    'Limits value in field to between 0 and 1
    If txtProb.Value > 1 Then
        txtProb.Value = 1
    ElseIf txtProb.Value < 0 Then
        txtProb.Value = 0
    End If

    'Sets scrollbar value
    scrProb.Value = txtProb.Value * 10

Exit Sub

'Error handler
ErrorTxtChange:

    ErrorHandler Err.Number
    Err.Clear

Exit Sub

End Sub

```

```

Private Sub UserForm_Initialize()
    'Initialises selection form

    'Initialise error handler
    On Error GoTo ErrorInitialise

    'Set initial value of text box
    txtProb.Value = 0

Exit Sub

'Error handler
ErrorInitialise:

    ErrorHandler Err.Number
    Err.Clear

Exit Sub

End Sub

'External subroutine to handle errors
Private Sub ErrorHandler(ErrorNumber As Long)

    'Error selector
    Select Case ErrorNumber
    'Insert specific error cases to catch here
    Case Else
        MsgBox "Error: " & Err.Number & " - " & Err.Description, , "T-GIS"
    End Select

End Sub

```

frmTgisProbSum

This form is called when the user clicks “Calculate” if they have the “Sum result” checkbox ticked. It will allow the user to select a field by which to weight any probabilities, and will then sum probabilities based upon spatial concurrence. A new shapefile layer will then be created and added to the map containing these summed probabilities. It is also used to select the weighting field for the Analyst tool.

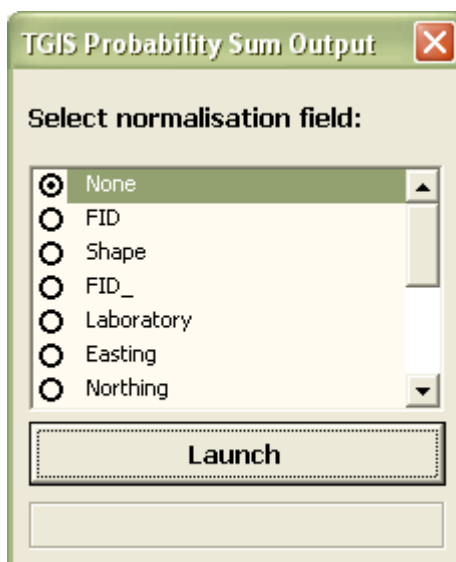


Fig A.6 - Summed results form.

```

Option Explicit
'General Declarations
Private pNewFeatureClass As IFeatureClass

Private Sub cmdSumProb_Click()
    'Calculates sum of probabilities at each point and creates new layer

'Initialise error handler
On Error GoTo ErrorLaunch

    'Declare variables
    Dim pFeature As IFeature           'IFeature interface to obtain point details
    Dim pPoint As IPoint               'IPoint interface to obtain point x & y
    Dim booNormalise As Boolean         'Boolean to record whether to normalise
    Dim booPointPresent As Boolean      'Boolean to record whether point already present
    Dim dblMultiplier As Double        'Normalisation figure
    Dim strNormalField As String        'Normalisation field name
    Dim lngNormalField As Long          'Normalisation field number
    Dim strNormalSumField As String     'Sum of normalisation field name
    Dim lngCurrentRowSum As Long        'Counter for field
    Dim lngCurrentArraySum As Long      'Counter for array
    Dim lngSumArraySize As Long         'Size of array
    Dim ProbSumArray() As Double        'Array to hold easting, northing, and sum of
                                         TGIS_Prob

    Dim dblCurrentProb As Double        'TGIS_Prob of current object
    Dim dblCurrentEasting As Double     'Easting of current object
    Dim dblCurrentNorthing As Double    'Northing of current object
    Dim strNewLayerName As String       'Name for new layer
    Dim intList As Long                 'Counter variable

    'First, run alternative process when called from Analyst tool
    If booAnalyst = True Then

        If lstNormalField.Selected(0) = True Then

            booAnalyst = False

        Else

            'Obtain field for normalisation
            intList = 1
            For intList = 1 To intListCount
                If lstNormalField.Selected(intList) = True Then
                    strNormAnalystField = CStr(lstNormalField.List(intList))
                    'If field is selected...
                    'Write it to the normal field attribute
                End If
            Next intList

            'Then check it is numeric
            lngNormAnalystField = pTable.FindField(strNormAnalystField)
            'Obtains field index of normal field

            lngCurrentRowSum = 0
            For lngCurrentRowSum = 0 To lngNoOfRows
                'Have to subtract one as they start from 0 not 1

                Call ProgressBar(lngCurrentRowSum, lngNoOfRows) 'Updates progress bar

                Set pRow = pTable.GetRow(lngCurrentRowSum)      'Gets next row
                'Checks if values are numeric
                If IsNumeric(pRow.Value(lngNormAnalystField)) = False Then
                    MsgBox "The selected field contains non-numeric data, try again.", , "T-GIS"

                    Exit Sub
                End If

            Next lngCurrentRowSum

            lblBar4.Width = 0

        End If
    End If

```

```

        'Then load next form
        Unload Me
        Load frmAnaProc
        frmAnaProc.Show

    Exit Sub

End If

'Initialise variables
lngCurrentArraySum = 0
lngSumArraySize = 0
ReDim ProbSumArray(4, 0)

'Obtain selection from normalisation field and check if numeric
If lstNormalField.Selected(0) = True Then
    'i.e. no normalisation to be performed
    booNormalise = False
    dblMultiplier = 1
Else
    'Obtain field for normalisation
    intList = 1
    For intList = 1 To intListCount
        If lstNormalField.Selected(intList) = True Then
            strNormalField = CStr(lstNormalField.List(intList))
            'If field is selected...
            'Write it to the normal
            'field attribute
        End If
    Next intList

    'Then check it is numeric
    lngNormalField = pTable.FindField(strNormalField) 'Obtains field index of normal
                                                    field

    lngCurrentRowSum = 0
    For lngCurrentRowSum = 0 To lngNoOfRows
        'Have to subtract one as they
        'start from 0 not 1

        Call ProgressBar(lngCurrentRowSum, lngNoOfRows) 'Updates progress bar

        Set pRow = pTable.GetRow(lngCurrentRowSum) 'Gets next row
        'Checks if values are numeric
        If IsNumeric(pRow.Value(lngNormalField)) = False Then
            MsgBox "The selected field contains non-numeric data, try again.", , "T-GIS"
            Exit Sub
        End If

    Next lngCurrentRowSum

    lblBar4.Width = 0
    booNormalise = True

End If

'Populate array with eastings, northings, and summed probability

'Assigning IRow, IFeature, and IPoint interfaces
Set pRow = pTable.GetRow(0)
Set pFeature = pTable.GetRow(0)
If pFeature.Shape.GeometryType = esriGeometryPoint Then
    Set pPoint = pFeature.Shape
Else
    MsgBox "The selected layer contains non-point data, feature does not function for
    non-point data.", , "T-GIS"

    Unload Me
    Exit Sub
End If

'Obtain values for current point
dblCurrentProb = CDbl(pRow.Value(lngFieldProb))
If booNormalise = True Then dblMultiplier = CDbl(pRow.Value(lngNormalField))
dblCurrentEasting = CDbl(pPoint.X)
dblCurrentNorthing = CDbl(pPoint.Y)

ProbSumArray(0, 0) = dblCurrentEasting
ProbSumArray(1, 0) = dblCurrentNorthing
ProbSumArray(2, 0) = (dblCurrentProb * dblMultiplier)

```

```

If booNormalise = True Then ProbSumArray(3, lngSumArraySize) = dblMultiplier

lngCurrentRowSum = 1
For lngCurrentRowSum = 1 To lngNoOfRows

    Call ProgressBar(lngCurrentRowSum, lngNoOfRows)      'Updates progress bar

    'Assigning IRow, IFeature, and IPoint interfaces
    Set pRow = pTable.GetRow(lngCurrentRowSum)
    Set pFeature = pTable.GetRow(lngCurrentRowSum)
    If pFeature.Shape.GeometryType = esriGeometryPoint Then
        Set pPoint = pFeature.Shape
    Else
        MsgBox "The selected layer contains non-point data, feature does not function for
            non-point data.", , "T-GIS"
        Unload Me
        Exit Sub
    End If

    'Obtain values for current point
    dblCurrentProb = CDBl(pRow.Value(lngFieldProb))

    If booNormalise = True Then dblMultiplier = CDBl(pRow.Value(lngNormalField))
    dblCurrentEasting = CDBl(pPoint.X)      'Obtain Easting
    dblCurrentNorthing = CDBl(pPoint.Y)     'Obtain Northing

    'Check values against array and either add them in or add new point
    booPointPresent = False
    lngCurrentArraySum = 0
    For lngCurrentArraySum = 0 To lngSumArraySize

        'Check point x & y against array values
        If dblCurrentEasting = ProbSumArray(0, lngCurrentArraySum) And _
            dblCurrentNorthing = ProbSumArray(1, lngCurrentArraySum) Then
            'Add on new value
            ProbSumArray(2, lngCurrentArraySum) = ProbSumArray(2, lngCurrentArraySum) +
                (dblCurrentProb * dblMultiplier)

            If booNormalise = True Then
                If dblCurrentProb > 0 Then ProbSumArray(3, lngCurrentArraySum) =
                    ProbSumArray(3, lngCurrentArraySum) + dblMultiplier
            End If
            booPointPresent = True
            Exit For
        End If

    Next lngCurrentArraySum

    If booPointPresent = False Then

        'Resize array and add new point if not found
        lngSumArraySize = lngSumArraySize + 1
        ReDim Preserve ProbSumArray(4, lngSumArraySize)
        ProbSumArray(0, lngSumArraySize) = dblCurrentEasting
        ProbSumArray(1, lngSumArraySize) = dblCurrentNorthing
        ProbSumArray(2, lngSumArraySize) = (dblCurrentProb * dblMultiplier)
        If booNormalise = True Then
            If dblCurrentProb > 0 Then ProbSumArray(3, lngSumArraySize) = dblMultiplier
        End If

    End If

Next lngCurrentRowSum

'Create new layer
strNewLayerName = "Sum of TGIS_Prob for " & strLayerSelect & " " & lngDispTMin & " to " &
    lngDispTMax

If booNormalise = True Then
    strNewLayerName = strNewLayerName + " normalised by " & strNormalField
End If

'Declare variables
Dim pNewLayer As IFeatureLayer
Dim pNewTable As ITable
Dim pNewField As IField
Dim pNewFieldEdit As IFieldEdit
Dim pNewRow As IRow

```

```

Dim pNewFeature As IFeature
Dim pNewPoint As IPoint

'Create new shapefile
Call CreateShapefile(strNewLayerName)

'Initialise variables
Set pNewLayer = New FeatureLayer
Set pNewTable = pNewLayer
Set pNewLayer.FeatureClass = pNewFeatureClass

'Set layer name
pNewLayer.Name = strNewLayerName

'Add points to layer shapefile
lngCurrentArraySum = 0
For lngCurrentArraySum = 0 To lngSumArraySize

    Call ProgressBar(lngCurrentArraySum, lngSumArraySize)      'Updates progress bar

    Set pNewPoint = New Point

    pNewPoint.X = ProbSumArray(0, lngCurrentArraySum)
    pNewPoint.Y = ProbSumArray(1, lngCurrentArraySum)

    Set pNewFeature = pNewFeatureClass.CreateFeature
    Set pNewFeature.Shape = pNewPoint
    pNewFeature.Store

Next lngCurrentArraySum

'Add new fields to layer table
Set pNewField = New Field
Set pNewFieldEdit = pNewField
pNewFieldEdit.Name = "Easting"
pNewFieldEdit.Type = esriFieldTypeDouble
pNewTable.AddField pNewField

Set pNewField = New Field
Set pNewFieldEdit = pNewField
pNewFieldEdit.Name = "Northing"
pNewFieldEdit.Type = esriFieldTypeDouble
pNewTable.AddField pNewField

Set pNewField = New Field
Set pNewFieldEdit = pNewField
pNewFieldEdit.Name = "SumProb"
pNewFieldEdit.Type = esriFieldTypeDouble
pNewTable.AddField pNewField

If booNormalise = True Then
    Set pNewField = New Field
    Set pNewFieldEdit = pNewField
    pNewFieldEdit.Name = "SumOf" & strNormalField
    pNewFieldEdit.Type = esriFieldTypeDouble
    pNewTable.AddField pNewField
End If

Dim lngNewRows As Long
Dim pQueryFilter As IQueryFilter      'Need an empty query to obtain the number of
                                      rows in the table

Set pQueryFilter = New QueryFilter
lngNewRows = CLng(pNewTable.RowCount(pQueryFilter))  'Obtains the number of rows
lngNewRows = lngNewRows - 1                        'Then subtract one as we start
                                                    from row 0

'Add Easting, Northing, and summed probability to layer table
lngCurrentRowSum = 0
For lngCurrentRowSum = 0 To lngNewRows

    Call ProgressBar(lngCurrentRowSum, lngNewRows)      'Updates progress bar

    Set pNewRow = pNewTable.GetRow(lngCurrentRowSum)
    Set pNewFeature = pNewTable.GetRow(lngCurrentRowSum)
    Set pNewPoint = pNewFeature.Shape

```

```

lngCurrentArraySum = 0
For lngCurrentArraySum = 0 To lngSumArraySize

    'Check point x & y against array values
    If pNewPoint.X = ProbSumArray(0, lngCurrentArraySum) And pNewPoint.Y =
        ProbSumArray(1, lngCurrentArraySum) Then

        pNewRow.Value(3) = ProbSumArray(0, lngCurrentArraySum)
        pNewRow.Value(4) = ProbSumArray(1, lngCurrentArraySum)
        pNewRow.Value(5) = FormatNumber(ProbSumArray(2, lngCurrentArraySum), 2)
        If booNormalise = True Then pNewRow.Value(6) = ProbSumArray(3,
            lngCurrentArraySum)

        pNewRow.Store

    Exit For
End If

Next lngCurrentArraySum

Next lngCurrentRowSum

'Then set symbology
pNewLayer.DisplayField = "SumProb"

'Declare variables
Dim pGeoFeatureLayer As IGeoFeatureLayer          'IGeoFeatureLayer interface to access
                                                    renderer
Dim pClassBreaksRenderer As IClassBreaksRenderer  'IClassBreaksRenderer interface to
                                                    set symbology
Dim pNullColor As IColor                          'IColor Interface to get null colour
Dim pBlackColor As IRgbColor                      'IRgbColor Interface to create black
Dim pSymbol As ISymbol                            'ISymbol interface to set symbol
                                                    properties
Dim pMarkerSymbol As ISimpleMarkerSymbol          'ISimpleMarkerSymbol interface to set
                                                    symbol properties
Dim pColor As IRgbColor                          'IRgbColor interface to create colour
Dim dblMaxSum As Double                          'Max summed probability
Dim dblSumStep As Double                        '10% of above

'Initialise variables
Set pClassBreaksRenderer = New ClassBreaksRenderer
Set pGeoFeatureLayer = pNewLayer                  'Set IGeoFeatureLayer to current
                                                    layer (QueryInterface)

'Find max summed probability
Set pNewRow = pNewTable.GetRow(0)
dblMaxSum = pNewRow.Value(5)

lngCurrentRowSum = 1
For lngCurrentRowSum = 1 To lngNewRows

    Set pNewRow = pNewTable.GetRow(lngCurrentRowSum)
    If pNewRow.Value(5) > dblMaxSum Then dblMaxSum = pNewRow.Value(5)

Next lngCurrentRowSum

dblSumStep = dblMaxSum / 10

'Set symbols to use colour gradient over SumProb field
pClassBreaksRenderer.Field = "SumProb"            'Set renderer field
pClassBreaksRenderer.BreakCount = 11              'Set number of breaks then breaks
pClassBreaksRenderer.MinimumBreak = 0
pClassBreaksRenderer.Break(0) = 0
pClassBreaksRenderer.Break(1) = 1 * dblSumStep
pClassBreaksRenderer.Break(2) = 2 * dblSumStep
pClassBreaksRenderer.Break(3) = 3 * dblSumStep
pClassBreaksRenderer.Break(4) = 4 * dblSumStep
pClassBreaksRenderer.Break(5) = 5 * dblSumStep
pClassBreaksRenderer.Break(6) = 6 * dblSumStep
pClassBreaksRenderer.Break(7) = 7 * dblSumStep
pClassBreaksRenderer.Break(8) = 8 * dblSumStep
pClassBreaksRenderer.Break(9) = 9 * dblSumStep
pClassBreaksRenderer.Break(10) = dblMaxSum

```

```

Set pMarkerSymbol = New SimpleMarkerSymbol      'Create new simple marker symbol
Set pNullColor = New RgbColor
pNullColor.NullColor = True
pMarkerSymbol.Color = pNullColor                'Apply colour
pMarkerSymbol.size = 6                         'Set size
pMarkerSymbol.Style = esriSMSCircle             'Set symbol
pMarkerSymbol.Outline = True                   'Add outline
Set pBlackColor = New RgbColor                  'Create new colour (black)
pBlackColor.Red = 0                             'Set red channel
pBlackColor.Green = 0                           'Set green channel
pBlackColor.Blue = 0                           'Set blue channel
pMarkerSymbol.OutlineColor = pBlackColor        'Set outline to black
Set pSymbol = pMarkerSymbol                    'Queryinterface

pClassBreaksRenderer.Symbol(0) = pSymbol        'Sets symbol
pClassBreaksRenderer.Label(0) = "0"             'Sets label

Dim lngCount As Long
For lngCount = 1 To 10

    Set pColor = New RgbColor                    'Create new RGB colour
                                                (Yellow -> Red)
    pColor.Red = 235                             'Set red channel
    pColor.Green = 255 - ((255 / 100) * lngCount ^ 2) 'Set green channel
    pColor.Blue = 10                             'Set blue channel
    Set pMarkerSymbol = New SimpleMarkerSymbol    'Create new simple marker
                                                symbol
    pMarkerSymbol.Color = pColor                  'Apply colour
    pMarkerSymbol.size = 6                       'Set size
    pMarkerSymbol.Style = esriSMSCircle           'Set symbol
    pMarkerSymbol.Outline = True                  'Add outline
    pMarkerSymbol.OutlineColor = pBlackColor      'Set outline to black
    Set pSymbol = pMarkerSymbol                  'Queryinterface

    pClassBreaksRenderer.Symbol(lngCount) = pSymbol 'Sets symbol
    pClassBreaksRenderer.Label(lngCount) = ">" & ((lngCount - 1) * dblSumStep) & " to " & (lngCount * dblSumStep) 'Sets label

Next lngCount

Set pGeoFeatureLayer.Renderer = pClassBreaksRenderer 'Set current layer's
                                                        renderer to class breaks renderer

'Then add layer and refresh display
pMap.AddLayer pNewLayer

pMxDocument.UpdateContents
pActiveView.Refresh

'Then unload form
Unload Me

Exit Sub

'Error handler
ErrorLaunch:

    ErrorHandler Err.Number
    Err.Clear

Exit Sub

End Sub

Private Sub UserForm_Initialize()

    'Initialises new form, populating panel with field names

    'Initialise error handler
    On Error GoTo ErrorInitialise

    'Assign variables
    Set pFeatureLayer = pLayer                    'Sets FeatureLayer to current layer
    Set pLayerFields = pFeatureLayer              'QueryInterface

```



```

'Adds layer fields to list boxes
intListCount = pLayerFields.FieldCount

'Gets the number of fields belonging to
the layer

'Adds "None" option to normal listbox
lstNormalField.AddItem "None"

Dim intList As Long
For intList = 0 To intListCount - 1
    Set pField = pLayerFields.Field(intList)
    lstNormalField.AddItem pField.Name
Next intList

'Have to subtract one as we start from 0
not 1
'Sets the current field to the IField
'Assigns field name to normal listbox

lstNormalField.Selected(0) = True
'Sets initial selection to "None"

Exit Sub

'Error handler
ErrorInitialise:

    ErrorHandler Err.Number
    Err.Clear

Exit Sub

End Sub

'Progress bar subroutine
Private Sub ProgressBar(lngValue As Long, lngTotal As Long)

'Initialise error handler
On Error GoTo ErrorProgress

    'Repaint form
    Me.Repaint

    'Set new bar width
    lblBar4.Width = (156 / lngTotal) * lngValue

    'Repaint frame
    fraBar4.Repaint

Exit Sub

'Error handler
ErrorProgress:

    ErrorHandler Err.Number
    Err.Clear

Exit Sub

End Sub

'External subroutine to create a shapefile
'Modified from:
'http://edndoc.esri.com/arcobjects/8.3/?URL=/ArcObjectsOnline/Samples/Geodatabase/Creating%25
20Data/CreateNewShapefile.htm

'With thanks!
Public Sub CreateShapefile(strFileName As String)

    Dim strName As String
    strName = strFileName ' Dont include .shp extension
    Const strShapeFieldName As String = "Shape"
    Dim strDirectory As String
    Dim pWorkspace As IWorkspace
    Dim pDataSet As IDataset
    Set pDataSet = pMap.Layer(0)
    Set pWorkspace = pDataSet.Workspace
    strDirectory = pWorkspace.PathName

```

```

' Open the folder to contain the shapefile as a workspace
Dim pFWS As IFeatureWorkspace
Dim pWorkspaceFactory As IWorkspaceFactory
Set pWorkspaceFactory = New ShapefileWorkspaceFactory
Set pFWS = pWorkspaceFactory.OpenFromFile(strDirectory, 0)

' Set up a simple fields collection
Dim pSFields As IFields
Dim pSFieldsEdit As IFieldsEdit
Set pSFields = New Fields
Set pSFieldsEdit = pSFields

Dim pSField As IField
Dim pSFieldEdit As IFieldEdit

' Make the shape field
' it will need a geometry definition, with a spatial reference
Set pSField = New Field
Set pSFieldEdit = pSField
pSFieldEdit.Name = strShapeFieldName
pSFieldEdit.Type = esriFieldTypeGeometry

Dim pGeomDef As IGeometryDef
Dim pGeomDefEdit As IGeometryDefEdit
Set pGeomDef = New GeometryDef
Set pGeomDefEdit = pGeomDef
With pGeomDefEdit
    .GeometryType = esriGeometryPoint
    Set .SpatialReference = New UnknownCoordinateSystem
End With
Set pSFieldEdit.GeometryDef = pGeomDef
pSFieldsEdit.AddField pSField

' Create the shapefile
' (some parameters apply to geodatabase options and can be defaulted as Nothing)
Set pNewFeatureClass = pFWS.CreateFeatureClass(strName, pSFields, Nothing, _
    Nothing, esriFTSimple, strShapeFieldName, "")
End Sub

'External subroutine to handle errors
Private Sub ErrorHandler(ErrorNumber As Long)

'Error selector
Select Case ErrorNumber
'Insert specific error cases to catch here
Case -2147467259
    MsgBox "New layer already exists. Delete from computer and try again.", , "T-GIS"
    Exit Sub
Case Else
    MsgBox "Error: " & Err.Number & " - " & Err.Description, , "T-GIS"
End Select
End Sub

```

frmAnaSpace

This is the first of a series of forms called when the Analyst tool is launched. Using it, the desired spatial scope for analysis is selected. The temporal selection form is then called.

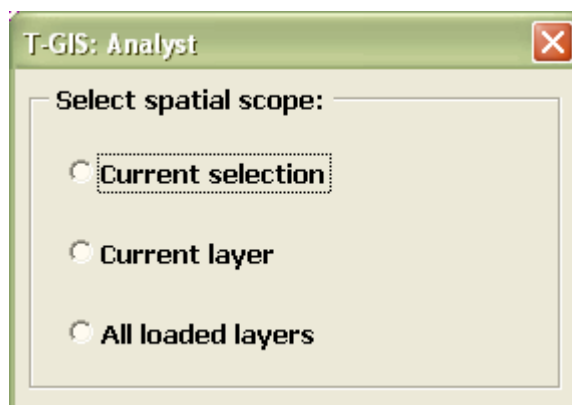


Fig A.7 - Spatial scope selection form for Analyst tool.

```

Option Explicit
'General declarations

Private Sub optAllLayers_Click()
    'Method for All Layers button click

    'Initialise error handler
    On Error GoTo ErrorAllLayers

    'Set integer recording option selected to 2
    intAnaSpace = 2
    booAnalyst = False

    'Then load next form
    Unload Me
    Load frmAnaTime
    frmAnaTime.Show

Exit Sub

'Error handler
ErrorAllLayers:

    ErrorHandler Err.Number
    Err.Clear

Exit Sub

End Sub

Private Sub optLayer_Click()
    'Method for Current Layer button click

    'Initialise error handler
    On Error GoTo ErrorCurrent

    'Set integer recording option selected to 1
    intAnaSpace = 1
    booAnalyst = True

    'Then load next form
    Unload Me
    Load frmAnaTime
    frmAnaTime.Show

Exit Sub

```

```

'Error handler
ErrorCurrent:

    ErrorHandler Err.Number
    Err.Clear

Exit Sub

End Sub

Private Sub optSelection_Click()
    'Method for Selection button click

'Initialise error handler
On Error GoTo ErrorSelection

    'Set integer recording option selected to 0
    intAnaSpace = 0
    booAnalyst = True

    'Then load next form
    Unload Me
    Load frmAnaTime
    frmAnaTime.Show

Exit Sub

'Error handler
ErrorSelection:

    ErrorHandler Err.Number
    Err.Clear

Exit Sub

End Sub

'External subroutine to handle errors
Private Sub ErrorHandler(ErrorNumber As Long)

    'Error selector
    Select Case ErrorNumber
    'Insert specific error cases to catch here
    Case Else
        MsgBox "Error: " & Err.Number & " - " & Err.Description, , "T-GIS"
    End Select

End Sub

```

frmAnaTime

This is the next form in the Analyst sequence. It is used to select the temporal scope of the analysis, then calls the next form in the sequence - spatial resolution.

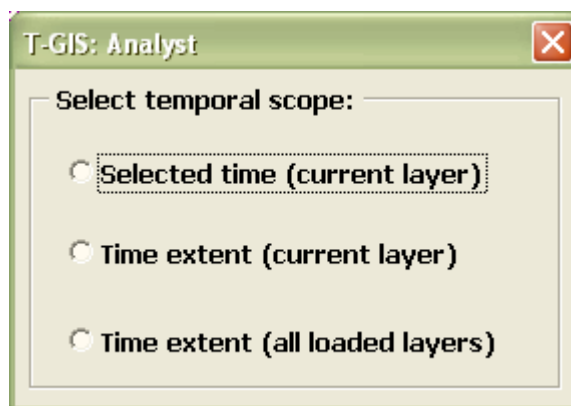


Fig A.8 - Temporal scope selection form for Analyst tool.

```

Option Explicit
'General declarations

Private Sub optExtentAll_Click()
    'Method for Time Extent (all layers) button click

    'Initialise error handler
    On Error GoTo ErrorExtentAll

    'Set min, max and range variables to min and max of all layers
    lngAnaMin = LayerArray(3, 0)
    lngAnaMax = LayerArray(4, 0)

    Dim intCount As Integer
    For intCount = 1 To (intArrayIndex - 1)
        If LayerArray(3, intCount) < lngAnaMin Then lngAnaMin = LayerArray(3, intCount)
        If LayerArray(4, intCount) > lngAnaMax Then lngAnaMax = LayerArray(4, intCount)
    Next intCount

    lngAnaRange = lngAnaMax - lngAnaMin

    'Then load next form
    Unload Me
    Load frmAnaScale
    frmAnaScale.Show

Exit Sub

'Error handler
ErrorExtentAll:

    ErrorHandler Err.Number
    Err.Clear

Exit Sub

End Sub

Private Sub optExtentTime_Click()
    'Method for Time Extent button click

    'Initialise error handler
    On Error GoTo ErrorExtent

    'Set min, max and range variables
    lngAnaMin = LayerArray(3, intArrayCurrent)
    lngAnaMax = LayerArray(4, intArrayCurrent)
    lngAnaRange = lngAnaMax - lngAnaMin

```

```

        'Then load next form
        Unload Me
        Load frmAnaScale
        frmAnaScale.Show

Exit Sub

'Error handler
ErrorExtent:

        ErrorHandler Err.Number
        Err.Clear

Exit Sub

End Sub

Private Sub optSelectTime_Click()
    'Method for Time Selection button click

'Initialise error handler
On Error GoTo ErrorTimeSelect

        'Set min, max and range variables
        lngAnaMin = LayerArray(5, intArrayCurrent)
        lngAnaMax = LayerArray(6, intArrayCurrent)
        lngAnaRange = lngAnaMax - lngAnaMin

        'Then load next form
        Unload Me
        Load frmAnaScale
        frmAnaScale.Show

Exit Sub

'Error handler
ErrorTimeSelect:

        ErrorHandler Err.Number
        Err.Clear

Exit Sub

End Sub

'External subroutine to handle errors
Private Sub ErrorHandler(ErrorNumber As Long)

    'Error selector
    Select Case ErrorNumber
        'Insert specific error cases to catch here
    Case Else
        MsgBox "Error: " & Err.Number & " - " & Err.Description, , "T-GIS"
    End Select

End Sub

```

frmAnaScale

This is the next form in the Analyst tool sequence, and is used to select the temporal resolution (in years) of the analysis. The processing form is then called. This form is also used to select the rate of change for the “Form periods” method (see below).

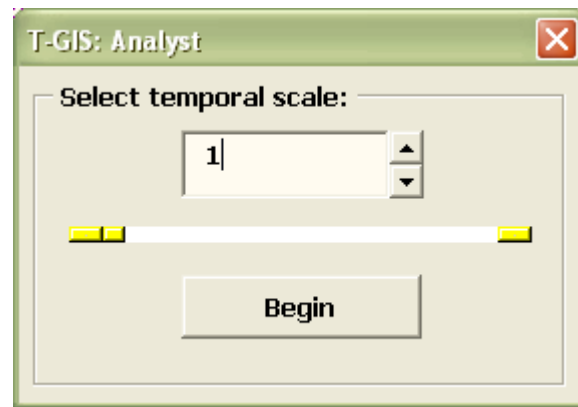


Fig A.9 - Temporal resolution selection form for Analyst tool.

```
Option Explicit
'General declarations

Private Sub cmdBegin_Click()
    'Launches analysis process

    'Initialise error handler
    On Error GoTo ErrorBegin

    If intAnaWhich = 0 Then

        'Obtain desired scale value (from scrollbar, as textbox might be non-numeric)
        lngAnaScale = scrScale.Value

    ElseIf intAnaWhich = 1 Then

        dblPeriodPercent = CDbl(scrScale.Value) / 100

    End If

    'Then loads next form
    Unload Me

    If booAnalyst = False Then
        Load frmAnaProc
        frmAnaProc.Show
    Else
        Load frmTgisProbSum
        frmTgisProbSum.Show
    End If

Exit Sub

'Error handler
ErrorBegin:

    ErrorHandler Err.Number
    Err.Clear

Exit Sub

End Sub
```

```

Private Sub scrScale_Change()
    'Method for scrollbar change

    'Initialise error handler
    On Error GoTo ErrorScroll

        'Set spinner and textbox values
        spnScale.Value = scrScale.Value
        txtScale.Value = scrScale.Value

Exit Sub

    'Error handler
ErrorScroll:

        ErrorHandler Err.Number
        Err.Clear

Exit Sub

End Sub

Private Sub spnScale_Change()
    'Method for spinner button change

    'Initialise error handler
    On Error GoTo ErrorSpin

        'Set scrollbar and textbox values
        scrScale.Value = spnScale.Value
        txtScale.Value = txtScale.Value

Exit Sub

    'Error handler
ErrorSpin:

        ErrorHandler Err.Number
        Err.Clear

Exit Sub

End Sub

Private Sub txtScale_AfterUpdate()
    'Method for textbox change

    'Initialise error handler
    On Error GoTo ErrorText

        'If value in textbox is numeric, set spinner and scrollbar values
        If IsNumeric(txtScale.Value) = True Then

            'First make sure value is within accepted range
            If txtScale.Value < 1 Then txtScale.Value = 1

            If intAnaWhich = 0 Then

                If txtScale.Value > lngAnaRange Then txtScale.Value = lngAnaRange

            ElseIf intAnaWhich = 1 Then

                If txtScale.Value > 100 Then txtScale.Value = 100

            End If

            'Then set values
            scrScale.Value = txtScale.Value
            spnScale.Value = txtScale.Value

```



```

        'If non-numeric, set it back to the previous value
        ElseIf IsNumeric(txtScale.Value) = False Then

            txtScale.Value = scrScale.Value

        End If

Exit Sub

'Error handler
ErrorText:

    ErrorHandler Err.Number
    Err.Clear

Exit Sub

End Sub

Private Sub UserForm_Initialize()
    'Initialises form

    'Initialise error handler
    On Error GoTo ErrorInitialise

    If intAnaWhich = 0 Then

        'Initialise label, spinner, scrollbar and text box
        fraScale.Caption = "Select temporal scale:"

        scrScale.Min = 1
        scrScale.Max = lngAnaRange
        scrScale.SmallChange = 1

        spnScale.Min = 1
        spnScale.Max = lngAnaRange
        spnScale.SmallChange = 1

        txtScale.Value = 1

    ElseIf intAnaWhich = 1 Then

        'Initialise label, spinner, scrollbar and text box
        fraScale.Caption = "Select rate of change %:"

        scrScale.Min = 1
        scrScale.Max = 100
        scrScale.SmallChange = 1

        spnScale.Min = 1
        spnScale.Max = 100
        spnScale.SmallChange = 1

        txtScale.Value = 1

    End If

Exit Sub

'Error handler
ErrorInitialise:

    ErrorHandler Err.Number
    Err.Clear

Exit Sub

End Sub

```

```

'External subroutine to handle errors
Private Sub ErrorHandler(ErrorNumber As Long)

    'Error selector
    Select Case ErrorNumber
    'Insert specific error cases to catch here
    Case Else
        MsgBox "Error: " & Err.Number & " - " & Err.Description, , "T-GIS"
    End Select

End Sub

```

frmAnaProc

This form displays a progress bar while the calculations are undertaken by the Analyst tool. When it is complete, the results are written to a text variable and the results form launched. There are also methods in here relating to the “Form periods” tool.

```

Option Explicit
'General declarations
Private strAnaDbTMin As String           'Min field name
Private strAnaDbTMax As String           'Max field name
Private lngAnaFieldMin As Long            'Min field number
Private lngAnaFieldMax As Long            'Max field number
Private intAnaLayer As Integer            'Number of current layer
Private dblAnaDivisions As Double         'Number of divisions
Private dblAnaDivRemainder As Double      'Remainder over integer divisions
Private lngCount As Long                  'Counter
Private sngAnalystMultiplier As Single   'Multiplier for any normalisation

'Note: although this module purports to simply regulate the progress bar,
'in fact all the analytical processing takes place herein

Private Sub UserForm_Activate()
    'Initialises form, then processes analysis, then exits

    'Initialise error handler
    On Error GoTo ErrorInitialise

    'First, set multiplier to 1 if no normalisation
    If booAnalyst = False Then
        sngAnalystMultiplier = 1
    End If

    If intAnaWhich = 0 Then

        Me.Repaint

        'Initialise division and results arrays
        dblAnaDivisions = CDBl(lngAnaRange) / CDBl(lngAnaScale)
        lngAnaDivisions = Int(dblAnaDivisions)
        dblAnaDivRemainder = dblAnaDivisions - CDBl(lngAnaDivisions)
        If dblAnaDivRemainder > 0 Then lngAnaDivisions = lngAnaDivisions + 1

        ReDim AnaArray(lngAnaDivisions - 1)
        ReDim AnaCatsArray(lngAnaDivisions - 1)
        ReDim AnaPercArray(lngAnaDivisions - 1)

        'Then populate categories array
        For lngCount = 1 To (lngAnaDivisions - 1)

            AnaCatsArray((lngCount - 1)) = lngAnaMin + (lngCount * lngAnaScale)

        Next lngCount

        AnaCatsArray(lngAnaDivisions - 1) = lngAnaMax

        'Then launch analysis
        Select Case intAnaSpace
        Case 0, 2            'i.e. perform analysis on selection or all layers

```

```

Dim strProcessedLayer As String

'For each layer in the array:
For intAnaLayer = 0 To (intArrayIndex - 1)

    'Call layer analysis module
    strProcessedLayer = LayerArray(0, intAnaLayer)
    intAnaProb = LayerArray(9, intAnaLayer)
    lblCurrentProcess.Caption = "Processing layer: " & strProcessedLayer
    Call AnalyseLayer(strProcessedLayer)

    'Then reset progress bar
    lblBar3.Width = 0
    fraBar3.Repaint

Next intAnaLayer

Case 1                'i.e. perform analysis on current layer

    'Call layer analysis module
    intAnaLayer = intArrayCurrent
    lblCurrentProcess.Caption = "Processing current layer"
    Call AnalyseLayer(strLayerSelect)

    'Then reset progress bar
    lblBar3.Width = 0
    fraBar3.Repaint

End Select

'Obtain highest value in AnaArray()
dblMaxWeight = AnaArray(0)

For lngCount = 1 To (lngAnaDivisions - 1)

    If AnaArray(lngCount) > dblMaxWeight Then dblMaxWeight = AnaArray(lngCount)

Next lngCount

'Then populate percentage array
For lngCount = 0 To (lngAnaDivisions - 1)

    AnaPercArray(lngCount) = FormatNumber(AnaArray(lngCount) / dblMaxWeight, 2)

Next lngCount

'Write array data to string variable
lblCurrentProcess.Caption = "Writing results table"

strAnaResults = "Weighted timeline between " & lngAnaMin & " and " & lngAnaMax & "
                for:" & vbCrLf

If intAnaSpace = 0 Then

    strAnaResults = strAnaResults & "Current selection" & vbCrLf

ElseIf intAnaSpace = 1 Then

    strAnaResults = strAnaResults & "Layer: " & strLayerSelect & vbCrLf

ElseIf intAnaSpace = 2 Then

    strAnaResults = strAnaResults & "Layers: " & vbCrLf

    For intAnaLayer = 0 To (intArrayIndex - 1)

        strAnaResults = strAnaResults & LayerArray(0, intAnaLayer) & vbCrLf

    Next intAnaLayer

End If

If booAnalyst = True Then

    strAnaResults = strAnaResults & "Normalised by " & strNormAnalystField & vbCrLf

End If

```

```

strAnaResults = strAnaResults & "At scale of " & lngAnaScale & " years " & vbCrLf
strAnaResults = strAnaResults & "Period:" & vbTab & "Mean:" & vbTab & "Weighting:" &
vbTab & "%Weight:" & vbCrLf

strAnaResults = strAnaResults & lngAnaMin & " to " & AnaCatsArray(0) & vbTab &
(lngAnaMin + (lngAnaScale / 2)) _
& vbTab & AnaArray(0) & vbTab & AnaPercArray(0) & vbCrLf

For lngCount = 0 To (lngAnaDivisions - 2)

    Call ProgressBar(lngCount, (lngAnaDivisions - 2)) 'Updates progress bar

    strAnaResults = strAnaResults & (AnaCatsArray(lngCount) + 1) & " to " &
AnaCatsArray(lngCount + 1) & vbTab _
& (AnaCatsArray(lngCount) + ((AnaCatsArray(lngCount + 1) -
AnaCatsArray(lngCount)) / 2)) & vbTab & _
AnaArray(lngCount + 1) & vbTab & AnaPercArray(lngCount + 1) & vbCrLf

Next lngCount

'Then loads results form
Unload Me
Load frmAnaResults
frmAnaResults.Show

ElseIf intAnaWhich = 1 Then

    'Declare variables specific to this process
    Dim dblWeightPercent As Double '% of max. weighting to form groups from
    Dim dblCurrentValue As Double 'Current value of period
    Dim dblGroupValue As Double 'Value of current group

    'Reset array from previous use
    ReDim AnaPeriodArray(0)

    Me.Repaint

    lblCurrentProcess.Caption = "Calculating periods"

    'Calculate percentage for change
    dblWeightPercent = dblMaxWeight * dblPeriodPercent

    'Move through AnaArray(), populating AnaPeriodArray() with year values for each break
    lngPeriodCount = 1
    ReDim Preserve AnaPeriodArray(lngPeriodCount - 1)
    AnaPeriodArray(0) = lngAnaMin

    dblCurrentValue = AnaArray(0)

    For lngCount = 1 To (lngAnaDivisions - 2)

        'Update progress bar
        Call ProgressBar(lngCount, (lngAnaDivisions - 2))

        'Obtain current value
        dblGroupValue = AnaArray(lngCount)

        If dblGroupValue = 0 And dblCurrentValue = 0 Then

            'Do nothing!

        ElseIf dblGroupValue = 0 And dblCurrentValue > 0 Then

            'Set new current value
            dblCurrentValue = dblGroupValue

            'Increment period count and resize array
            lngPeriodCount = lngPeriodCount + 1
            ReDim Preserve AnaPeriodArray(lngPeriodCount - 1)

            'Add new period value to array
            AnaPeriodArray(lngPeriodCount - 1) = AnaCatsArray(lngCount - 1)

```

```

ElseIf dblCurrentValue = 0 And dblGroupValue > 0 Then

    'Set new current value
    dblCurrentValue = dblGroupValue

    'Increment period count and resize array
    lngPeriodCount = lngPeriodCount + 1
    ReDim Preserve AnaPeriodArray(lngPeriodCount - 1)

    'Add new period value to array
    AnaPeriodArray(lngPeriodCount - 1) = AnaCatsArray(lngCount - 1)

ElseIf dblGroupValue < (dblCurrentValue - dblWeightPercent) Then

    'Set new current value
    dblCurrentValue = dblGroupValue

    'Increment period count and resize array
    lngPeriodCount = lngPeriodCount + 1
    ReDim Preserve AnaPeriodArray(lngPeriodCount - 1)

    'Add new period value to array
    AnaPeriodArray(lngPeriodCount - 1) = AnaCatsArray(lngCount - 1)

ElseIf dblGroupValue > (dblCurrentValue + dblWeightPercent) Then

    'Set new current value
    dblCurrentValue = dblGroupValue

    'Increment period count and resize array
    lngPeriodCount = lngPeriodCount + 1
    ReDim Preserve AnaPeriodArray(lngPeriodCount - 1)

    'Add new period value to array
    AnaPeriodArray(lngPeriodCount - 1) = AnaCatsArray(lngCount - 1)

Else

    'Do nothing!

End If

Next lngCount

'Increment period count and resize array
lngPeriodCount = lngPeriodCount + 1
ReDim Preserve AnaPeriodArray(lngPeriodCount - 1)

'Add new period value to array
AnaPeriodArray(lngPeriodCount - 1) = lngAnaMax

'Then loads periods form
Me.Hide
Unload Me
Load frmTgisPeriods
frmTgisPeriods.Show

End If

Exit Sub

'Error handler
ErrorInitialise:

    ErrorHandler Err.Number
    Err.Clear

Exit Sub

End Sub

```

```

'Layer process subroutine
Private Sub AnalyseLayer(strLayerName As String)

'Initialise error handler
On Error GoTo ErrorAnalyseLayer

    'Initialise variables: could re-use the public versions of first two as used in the main
                                form,
    'but this will likely cause bugs when we return to the main form
    Dim pAnaLayer As IFeatureLayer      'IFeatureLayer interface to select layer by
                                        name
    Dim pAnaTable As ITable              'ITable interface to get into layer table
    Dim pQueryFilter As IQueryFilter     'Need an empty query to obtain the number of
                                        rows in the table

    Dim lngCurrentRow As Long            'Row counter
    Dim lngNoOfRowsAna As Long           'Total number of rows
    Dim pTableSelection As ITableSelection 'ITableSelection interface to get selected
                                        rows
    Dim pSelectionSet As ISelectionSet    'ISelectionSet interface to get selected rows
    Dim pEnumIDs As IEnumIDs              'IEnumIDs interface to get OIDs of selected
                                        rows

    Dim lngID As Long                    'OID of current row

    'Set the current layer to be analysed
    Dim intLayer As Long
    For intLayer = 0 To (pMap.LayerCount - 1)
        If UCase(pMap.Layer(intLayer).Name) = UCase(strLayerName) Then
            Set pAnaLayer = pMap.Layer(intLayer)
            Exit For
        End If
    Next intLayer
    Set pAnaTable = pAnaLayer            'Sets table to current layer
    Set pQueryFilter = New QueryFilter

    'Obtains layer min and max fields
    strAnaDbTMin = CStr(LayerArray(1, intAnaLayer))
    strAnaDbTMax = CStr(LayerArray(2, intAnaLayer))
    lngAnaFieldMin = pAnaTable.FindField(strAnaDbTMin)
    lngAnaFieldMax = pAnaTable.FindField(strAnaDbTMax)

    If intAnaSpace = 0 Then

        Set pTableSelection = pAnaTable      'Set selection to current layer
        Set pSelectionSet = pTableSelection.SelectionSet 'Set selection set to that of
                                                        table
        lngNoOfRowsAna = pSelectionSet.Count 'Obtain size of selection

        'Exit the subroutine if no features selected
        If lngNoOfRowsAna = 0 Then
            Exit Sub
        End If

        Set pEnumIDs = pSelectionSet.IDs
        lngCurrentRow = 0

        lngID = pEnumIDs.Next
        Do Until lngID = -1

            Call ProgressBar(lngCurrentRow, lngNoOfRowsAna) 'Updates progress bar

            Set pRow = pAnaTable.GetRow(lngID)              'Gets row

            'Call calculation method for row
            Call CalculateRow

            lngID = pEnumIDs.Next
            lngCurrentRow = lngCurrentRow + 1

        Loop

    Else

        lngNoOfRowsAna = CLng(pAnaTable.RowCount(pQueryFilter)) 'Obtains the number of
                                                                    rows
        lngNoOfRowsAna = lngNoOfRowsAna - 1                      'Then subtract one as we
                                                                    start from row 0
    End If

```

```

    For lngCurrentRow = 0 To lngNoOfRowsAna

        Call ProgressBar(lngCurrentRow, lngNoOfRowsAna)      'Updates progress bar

        Set pRow = pAnaTable.GetRow(lngCurrentRow)           'Gets row

        'Call calculation method for row
        Call CalculateRow

    Next lngCurrentRow

End If

Exit Sub

'Error handler
ErrorAnalyseLayer:

    ErrorHandler Err.Number
    Err.Clear

Exit Sub

End Sub

'Calculation subroutine
Private Sub CalculateRow()

'Initialise error handler
On Error GoTo ErrorCalculate

    'Declare variables
    Dim sngAnaItemMin As Single      'Item Tmin
    Dim sngAnaItemMax As Single      'Item TMax
    Dim sngAnaItemRange As Single    'Item date range
    Dim sngAnaProb As Single         'Item probability
    Dim sngAnaDivMin As Single       'Division min
    Dim sngAnaDivMax As Single       'Division max
    Dim sngAnaFactor1 As Single      'Factor variable for t.p.q. probability
    Dim sngAnaFactor2 As Single      'Factor variable for t.p.q. probability

    'Initialise variables
    sngAnaItemMin = CSng(pRow.Value(lngAnaFieldMin))
    sngAnaItemMax = CSng(pRow.Value(lngAnaFieldMax))
    sngAnaItemRange = sngAnaItemMax - sngAnaItemMin
    sngItemStDev = sngAnaItemRange / 4      'if +/-2 St.Dev, this is thus 1/4 of
                                           the date range

    'Firstly, skip item if it falls outside temporal range of interest
    If intAnaProb = 0 Or 2 Then
        If sngAnaItemMin > lngAnaMax Or sngAnaItemMax < lngAnaMin Then
            'Exit subprocedure
            Exit Sub
        End If
    ElseIf intAnaProb = 1 Then
        If (sngAnaItemMin - sngItemStDev) > lngAnaMax Or (sngAnaItemMax + sngItemStDev) <
            lngAnaMin Then
            'Exit subprocedure
            Exit Sub
        End If
    End If

    'Then check where remaining items in current row fall on the "graph" and make
    calculations
    sngAnaDivMin = CSng(lngAnaMin)
    sngAnaDivMax = CSng(AnaCatsArray(0))

```

```

If intAnaProb = 0 Then      'i.e. use simple probability

    If sngAnaItemMin = sngAnaItemMax Then
        If sngAnaItemMin >= sngAnaDivMin And sngAnaItemMin <= sngAnaDivMax Then
            sngAnaProb = 1
        Else
            sngAnaProb = 0
        End If
    ElseIf sngAnaDivMin <= sngAnaItemMin And sngAnaDivMax >= sngAnaItemMax Then
        sngAnaProb = 1
    ElseIf sngAnaDivMin > sngAnaItemMax Or sngAnaDivMax < sngAnaItemMin Then
        sngAnaProb = 0
    ElseIf sngAnaDivMin <= sngAnaItemMin And sngAnaDivMax < sngAnaItemMax Then
        sngAnaProb = (sngAnaDivMax - sngAnaItemMin) / sngAnaItemRange
    ElseIf sngAnaDivMin > sngAnaItemMin And sngAnaDivMax >= sngAnaItemMax Then
        sngAnaProb = (sngAnaItemMax - sngAnaDivMin) / sngAnaItemRange
    ElseIf sngAnaDivMin > sngAnaItemMin And sngAnaDivMax < sngAnaItemMax Then
        sngAnaProb = (sngAnaDivMax - sngAnaDivMin) / sngAnaItemRange
    Else
        sngAnaProb = 0
    End If

    'Perform any normalisation
    If booAnalyst = True Then
        sngAnalystMultiplier = CSng(pRow.Value(lngNormAnalystField))
    End If
    sngAnaProb = sngAnaProb * sngAnalystMultiplier

    AnaArray(0) = FormatNumber(AnaArray(0) + CDBl(sngAnaProb), 2)      'Adds current
                                                                    date probability to cumulative array

ElseIf intAnaProb = 1 Then 'i.e. use complex probability

    'Calculate variables
    sngItemMean = sngAnaItemMin + (sngAnaItemRange / 2) 'if +/-2 St.Dev, half the date
                                                         range above the min
    sngWidth = 0.05                                     '0.05 StDev seems to achieve
                                                         accurate probability to > 2 s.f.
    pi = 4 * Atn(1)                                     'Define pi
    sngSqr2Pi = Sqr(2 * pi)                             'Define the square root of 2 Pi

    If sngAnaItemMin = sngAnaItemMax Then
        If sngAnaItemMin >= sngAnaDivMin And sngAnaItemMin <= sngAnaDivMax Then
            sngAnaProb = 1
        Else
            sngAnaProb = 0
        End If
    ElseIf sngAnaDivMin <= (sngAnaItemMin - sngItemStDev) And sngAnaDivMax >=
        (sngAnaItemMax + sngItemStDev) Then
        sngAnaProb = 1
    ElseIf sngAnaDivMin > (sngAnaItemMax + sngItemStDev) Or sngAnaDivMax < (sngAnaItemMin
        - sngItemStDev) Then
        sngAnaProb = 0
    Else

        If sngAnaItemMin > (sngItemMean - (sngItemStDev * 3)) Then
            sngStart = sngAnaDivMin
        Else
            sngStart = (sngItemMean - (sngItemStDev * 3))
        End If
        If sngAnaItemMax < (sngItemMean + (sngItemStDev * 3)) Then
            sngEnd = sngAnaDivMax
        Else
            sngEnd = (sngItemMean + (sngItemStDev * 3))
        End If

        sngCurrent = sngStart
        sngAnaProb = 0
        sngGraphConstant = 1 / (sngSqr2Pi * sngItemStDev)

```



```

        'Then calculate probability from sngStart to sngEnd
    Do Until sngCurrent >= sngEnd
        'Calculate graph power
        sngGraphPower = ((sngCurrent - sngItemMean) ^ 2) / (2 * (sngItemStDev ^ 2))
        'Then the result
        sngGraphResult = sngGraphConstant * Exp(-sngGraphPower)
        'Then add to cumulative probability
        sngAnaProb = sngAnaProb + (sngGraphResult * (sngWidth * sngItemStDev))
        'Then increment loop position
        sngCurrent = sngCurrent + (sngItemStDev * sngWidth)
    Loop

End If

'Perform any normalisation
If booAnalyst = True Then
    sngAnalystMultiplier = CSng(pRow.Value(lngNormAnalystField))
End If
sngAnaProb = sngAnaProb * sngAnalystMultiplier

AnaArray(0) = FormatNumber(AnaArray(0) + CDBl(sngAnaProb))      'Adds current date
                                                                probability to cumulative array

ElseIf intAnaProb = 2 Then      'i.e. use t.p.q. probability

    If sngAnaItemMin = sngAnaItemMax Then
        If sngAnaItemMin >= sngAnaDivMin And sngAnaItemMin <= sngAnaDivMax Then
            sngAnaProb = 1
        ElseIf sngAnaItemMin < sngAnaDivMin Then
            sngAnaProb = 0
        ElseIf sngAnaItemMin > sngAnaDivMax Then
            sngAnaProb = 0
        Else
            sngAnaProb = 0
        End If
    ElseIf sngAnaDivMin <= sngAnaItemMin And sngAnaDivMax >= sngAnaItemMax Then
        sngAnaProb = 1
    ElseIf sngAnaDivMin > sngAnaItemMax Then
        sngAnaProb = 0
    ElseIf sngAnaDivMax < sngAnaItemMin Then
        sngAnaProb = 0
    ElseIf sngAnaDivMin <= sngAnaItemMin And sngAnaDivMax < sngAnaItemMax Then
        'i.e. lefthand portion of triangle is probability
        sngAnaFactor1 = sngAnaItemMax - sngAnaDivMax
        sngAnaProb = 1 - (sngAnaFactor1 ^ 2) * (1 / (sngAnaItemRange ^ 2))
    ElseIf sngAnaDivMin > sngAnaItemMin And sngAnaDivMax >= sngAnaItemMax Then
        'i.e. righthand portion of triangle is probability
        sngAnaFactor1 = sngAnaItemMax - sngAnaDivMin
        sngAnaProb = (sngAnaFactor1 ^ 2) * (1 / (sngAnaItemRange ^ 2))
    ElseIf sngAnaDivMin > sngAnaItemMin And sngAnaDivMax < sngAnaItemMax Then
        'i.e. middle portion of triangle is probability
        sngAnaFactor1 = sngAnaItemMax - sngAnaDivMin
        sngAnaFactor2 = sngAnaItemMax - sngAnaDivMax
        sngAnaProb = ((sngAnaFactor1 ^ 2) * (1 / (sngAnaItemRange ^ 2))) -
                     ((sngAnaFactor2 ^ 2) * (1 / (sngAnaItemRange ^ 2)))
    Else
        sngAnaProb = 0
    End If

    'Perform any normalisation
    If booAnalyst = True Then
        sngAnalystMultiplier = CSng(pRow.Value(lngNormAnalystField))
    End If
    sngAnaProb = sngAnaProb * sngAnalystMultiplier

    AnaArray(0) = FormatNumber(AnaArray(0) + CDBl(sngAnaProb), 2)      'Adds current
                                                                date probability to cumulative array

End If

Dim lngCount As Long

```

```

For lngCount = 1 To (lngAnaDivisions - 1)

    sngAnaDivMin = CSng(AnaCatsArray(lngCount - 1)) + 1
    sngAnaDivMax = CSng(AnaCatsArray(lngCount))

    If intAnaProb = 0 Then      'i.e. use simple probability

        If sngAnaItemMin = sngAnaItemMax Then
            If sngAnaItemMin >= sngAnaDivMin And sngAnaItemMin <= sngAnaDivMax Then
                sngAnaProb = 1
            Else
                sngAnaProb = 0
            End If
        ElseIf sngAnaDivMin <= sngAnaItemMin And sngAnaDivMax >= sngAnaItemMax Then
            sngAnaProb = 1
        ElseIf sngAnaDivMin > sngAnaItemMax Or sngAnaDivMax < sngAnaItemMin Then
            sngAnaProb = 0
        ElseIf sngAnaDivMin <= sngAnaItemMin And sngAnaDivMax < sngAnaItemMax Then
            sngAnaProb = (sngAnaDivMax - sngAnaItemMin) / sngAnaItemRange
        ElseIf sngAnaDivMin > sngAnaItemMin And sngAnaDivMax >= sngAnaItemMax Then
            sngAnaProb = (sngAnaItemMax - sngAnaDivMin) / sngAnaItemRange
        ElseIf sngAnaDivMin > sngAnaItemMin And sngAnaDivMax < sngAnaItemMax Then
            sngAnaProb = (sngAnaDivMax - sngAnaDivMin) / sngAnaItemRange
        Else
            sngAnaProb = 0
        End If

        'Perform any normalisation
        If booAnalyst = True Then
            sngAnalystMultiplier = CSng(pRow.Value(lngNormAnalystField))
        End If
        sngAnaProb = sngAnaProb * sngAnalystMultiplier

        AnaArray(lngCount) = FormatNumber(AnaArray(lngCount) + CDb1(sngAnaProb))
    'Adds current date probability to cumulative array

    ElseIf intAnaProb = 1 Then 'i.e. use complex probability

        'Calculate variables
        sngItemMean = sngAnaItemMin + (sngAnaItemRange / 2) 'if +/-2 St.Dev, half the
                                                             date range above the min
        sngWidth = 0.05                                     '0.05 StDev seems to achieve
                                                             accurate probability to > 2 s.f.
        pi = 4 * Atn(1)                                     'Define pi
        sngSqr2Pi = Sqr(2 * pi)                             'Define the square root of 2
                                                             Pi

        If sngAnaItemMin = sngAnaItemMax Then
            If sngAnaItemMin >= sngAnaDivMin And sngAnaItemMin <= sngAnaDivMax Then
                sngAnaProb = 1
            Else
                sngAnaProb = 0
            End If
        ElseIf sngAnaDivMin <= (sngAnaItemMin - sngItemStDev) And sngAnaDivMax >=
            (sngAnaItemMax + sngItemStDev) Then
            sngAnaProb = 1
        ElseIf sngAnaDivMin > (sngAnaItemMax + sngItemStDev) Or sngAnaDivMax <
            (sngAnaItemMin - sngItemStDev) Then
            sngAnaProb = 0
        Else
            If sngAnaItemMin > (sngItemMean - (sngItemStDev * 3)) Then
                sngStart = sngAnaDivMin
            Else
                sngStart = (sngItemMean - (sngItemStDev * 3))
            End If
            If sngAnaItemMax < (sngItemMean + (sngItemStDev * 3)) Then
                sngEnd = sngAnaDivMax
            Else
                sngEnd = (sngItemMean + (sngItemStDev * 3))
            End If

            sngCurrent = sngStart
            sngAnaProb = 0
            sngGraphConstant = 1 / (sngSqr2Pi * sngItemStDev)

```

```

        'Then calculate probability from sngStart to sngEnd
        Do Until sngCurrent >= sngEnd
            'Calculate graph power
            sngGraphPower = ((sngCurrent - sngItemMean) ^ 2) / (2 * (sngItemStDev ^ 2))

            'Then the result
            sngGraphResult = sngGraphConstant * Exp(-sngGraphPower)
            'Then add to cumulative probability
            sngAnaProb = sngAnaProb + (sngGraphResult * (sngWidth * sngItemStDev))
            'Then increment loop position
            sngCurrent = sngCurrent + (sngItemStDev * sngWidth)
        Loop

    End If

    'Perform any normalisation
    If booAnalyst = True Then
        sngAnalystMultiplier = CSng(pRow.Value(lngNormAnalystField))
    End If
    sngAnaProb = sngAnaProb * sngAnalystMultiplier

    AnaArray(lngCount) = FormatNumber(AnaArray(lngCount) + CDb1(sngAnaProb))
    'Adds current date probability to cumulative array

    ElseIf intAnaProb = 2 Then      'i.e. use t.p.q. probability

        If sngAnaItemMin = sngAnaItemMax Then
            If sngAnaItemMin >= sngAnaDivMin And sngAnaItemMin <= sngAnaDivMax Then
                sngAnaProb = 1
            ElseIf sngAnaItemMin < sngAnaDivMin Then
                sngAnaProb = 0
            ElseIf sngAnaItemMin > sngAnaDivMax Then
                sngAnaProb = 0
            Else
                sngAnaProb = 0
            End If
        ElseIf sngAnaDivMin <= sngAnaItemMin And sngAnaDivMax >= sngAnaItemMax Then
            sngAnaProb = 1
        ElseIf sngAnaDivMin > sngAnaItemMax Then
            sngAnaProb = 0
        ElseIf sngAnaDivMax < sngAnaItemMin Then
            sngAnaProb = 0
        ElseIf sngAnaDivMin <= sngAnaItemMin And sngAnaDivMax < sngAnaItemMax Then
            'i.e. lefthand portion of triangle is probability
            sngAnaFactor1 = sngAnaItemMax - sngAnaDivMax
            sngAnaProb = 1 - (sngAnaFactor1 ^ 2) * (1 / (sngAnaItemRange ^ 2))
        ElseIf sngAnaDivMin > sngAnaItemMin And sngAnaDivMax >= sngAnaItemMax Then
            'i.e. righthand portion of triangle is probability
            sngAnaFactor1 = sngAnaItemMax - sngAnaDivMin
            sngAnaProb = (sngAnaFactor1 ^ 2) * (1 / (sngAnaItemRange ^ 2))
        ElseIf sngAnaDivMin > sngAnaItemMin And sngAnaDivMax < sngAnaItemMax Then
            'i.e. middle portion of triangle is probability
            sngAnaFactor1 = sngAnaItemMax - sngAnaDivMin
            sngAnaFactor2 = sngAnaItemMax - sngAnaDivMax
            sngAnaProb = ((sngAnaFactor1 ^ 2) * (1 / (sngAnaItemRange ^ 2))) -
                ((sngAnaFactor2 ^ 2) * (1 / (sngAnaItemRange ^ 2)))
        Else
            sngAnaProb = 0
        End If

        'Perform any normalisation
        If booAnalyst = True Then
            sngAnalystMultiplier = CSng(pRow.Value(lngNormAnalystField))
        End If
        sngAnaProb = sngAnaProb * sngAnalystMultiplier

        AnaArray(lngCount) = FormatNumber(AnaArray(lngCount) + CDb1(sngAnaProb), 2)
        'Adds current date probability to cumulative array

    End If

Next lngCount

Exit Sub

```

```

ErrorCalculate:
    ErrorHandler Err.Number
    Err.Clear

Exit Sub

End Sub

'Progress bar subroutine
Private Sub ProgressBar(lngValue As Long, lngTotal As Long)

'Initialise error handler
On Error GoTo ErrorProgress

'Repaint form
Me.Repaint

'Set new bar width
lblBar3.Width = (186 / lngTotal) * lngValue

'Repaint frame
fraBar3.Repaint

Exit Sub

'Error handler
ErrorProgress:

    ErrorHandler Err.Number
    Err.Clear

Exit Sub

End Sub

'External subroutine to handle errors
Private Sub ErrorHandler(ErrorNumber As Long)

'Error selector
Select Case ErrorNumber
'Insert specific error cases to catch here
Case Else
    MsgBox "Error: " & Err.Number & " - " & Err.Description, , "T-GIS"
End Select

End Sub

```

frmAnaResults

This form contains a text box in which are presented the results of the Analyst tool's analysis. Adjustments can be made to the temporal resolution of the analysis (re-calling the form frmAnaScale and then making new calculations from that point). The results can also be copied to the clipboard for pasting into a spreadsheet (or other document). Further, the "Form periods" button will call a method used to form periods based upon the rate of change in the current summed probability curve for study in the main TGIS.

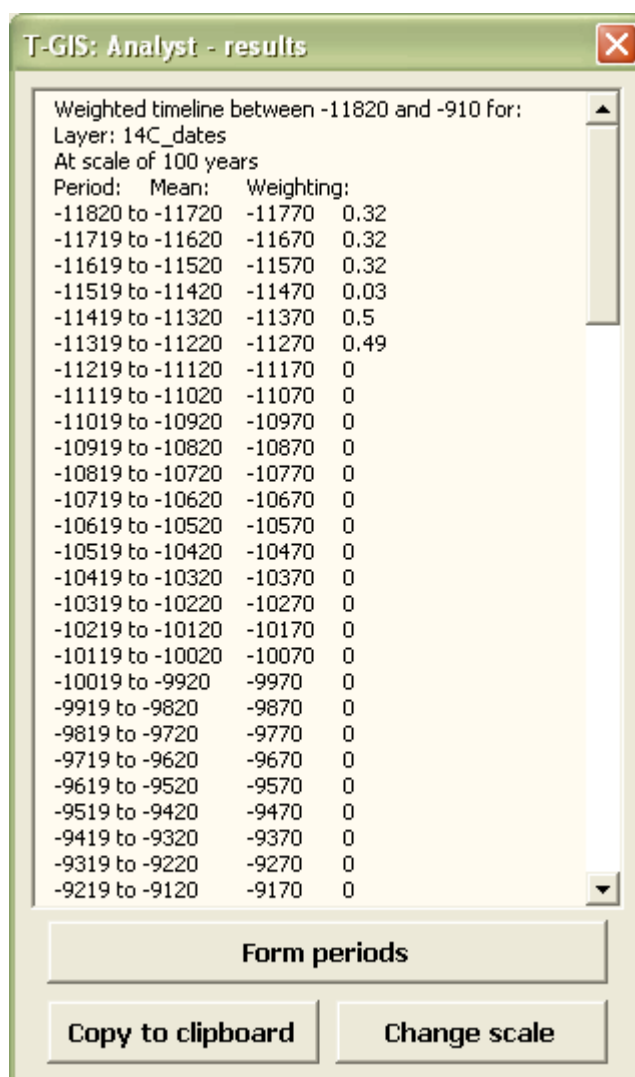


Fig A.10 - Analyst results form.

```

Option Explicit
'General declarations

Private Sub cmdLaunchPeriods_Click()
    'Launches period calculator

    'Initialise error handler
    On Error GoTo ErrorPeriods

    'Unloads this frame and launches periods frame (temp)
    intAnaWhich = 1
    Unload Me
    Load frmAnaScale
    frmAnaScale.Show

Exit Sub

'Error handler
ErrorPeriods:

    ErrorHandler Err.Number
    Err.Clear

Exit Sub

End Sub

```

```
Private Sub cmdRescale_Click()  
    'Launches Scale form  
  
    'Initialise error handler  
    On Error GoTo ErrorRescale  
  
        'Unloads this frame and launches scale frame  
        Unload Me  
        Load frmAnaScale  
        frmAnaScale.Show  
  
Exit Sub  
  
    'Error handler  
ErrorRescale:  
  
        ErrorHandler Err.Number  
        Err.Clear  
  
Exit Sub  
  
End Sub  
  
Private Sub cmdClipboard_Click()  
    'Copies results to clipboard  
  
    'Initialise error handler  
    On Error GoTo ErrorSave  
  
        'Declare variables  
        Dim pDataObject As DataObject  
  
        'Initialise variables  
        Set pDataObject = New DataObject  
  
        'Copy results to clipboard  
        pDataObject.SetText txtResultsOutput.Text  
        pDataObject.PutInClipboard  
  
Exit Sub  
  
    'Error handler  
ErrorSave:  
  
        ErrorHandler Err.Number  
        Err.Clear  
  
Exit Sub  
  
End Sub  
  
Private Sub UserForm_Initialize()  
    'Initialises form  
  
    'Initialise error handler  
    On Error GoTo ErrorInitialise  
  
        'Apply string to text window  
        txtResultsOutput.Text = strAnaResults  
  
Exit Sub  
  
    'Error handler  
ErrorInitialise:  
  
        ErrorHandler Err.Number  
        Err.Clear  
  
Exit Sub  
  
End Sub
```

```

'External subroutine to handle errors
Private Sub ErrorHandler(ErrorNumber As Long)

    'Error selector
    Select Case ErrorNumber
    'Insert specific error cases to catch here
    Case Else
        MsgBox "Error: " & Err.Number & " - " & Err.Description, , "T-GIS"
    End Select

End Sub

```

frmTgisPeriods

This is the form that results when the “Form periods” tool is launched from the Analyst tool. It relates to the main TGIS UI, but seizes control of the system due to problems with not being permitted to have more than one dominant form open at any time. The user may scroll through or select periods and calculations will be made automatically by the TGIS.

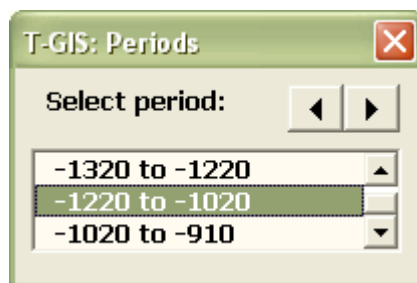


Fig A.11 - Period selection form.

```

Option Explicit
'General declarations
Private lngCount As Long      'Counter

Private Sub lstPeriod_Click()
    'Method for listbox

    'Initialise error handler
    On Error GoTo ErrorList

    'Iterate through list items
    For lngCount = 0 To (lngPeriodCount - 1)
        If lstPeriod.Selected(lngCount) = True Then      'If the item is selected...

            spnPeriod.Value = lngCount                    'Set the spinner value

        End If
    Next lngCount

Exit Sub

'Error handler
ErrorList:

    ErrorHandler Err.Number
    Err.Clear

Exit Sub

End Sub

```

```

Private Sub spnPeriod_Change()
    'Method for spinner buttons

    'Initialise error handler
    On Error GoTo ErrorSpinner

        'Set list selection then repaint form
        lstPeriod.Selected(spnPeriod.Value) = True
        Me.Repaint

        'Then call method to adjust main panel
        Call AdjustMainPanel

Exit Sub

'Error handler
ErrorSpinner:

    ErrorHandler Err.Number
    Err.Clear

Exit Sub

End Sub

Private Sub UserForm_Activate()
    'Method called when userform launched

    'Initialise error handler
    On Error GoTo ErrorInitialise

        'Populate listbox from AnaPeriodArray()
        lstPeriod.AddItem AnaPeriodArray(0) & " to " & AnaPeriodArray(1)

        If lngPeriodCount > 2 Then
            For lngCount = 1 To (lngPeriodCount - 2)

                lstPeriod.AddItem AnaPeriodArray(lngCount) & " to " & AnaPeriodArray(lngCount + 1)

            Next lngCount
        End If

        lstPeriod.Selected(0) = True

        'Set min / max / step / value for spinner
        spnPeriod.Min = 0
        spnPeriod.Max = (lngPeriodCount - 2)
        spnPeriod.SmallChange = 1
        spnPeriod.Value = 0

        'Repaint form
        Me.Repaint

        'Then call method to adjust main panel
        Call AdjustMainPanel

Exit Sub

'Error handler
ErrorInitialise:

    ErrorHandler Err.Number
    Err.Clear

Exit Sub

End Sub

```



```
Private Sub AdjustMainPanel()  
    'External subroutine to adjust the values on the main panel and calculate probability  
  
    'Initialise error handler  
    On Error GoTo ErrorAdjust  
  
    'Obtain values from AnaPeriodArray() & set new values for min & max in main panel  
    lngDispTMin = AnaPeriodArray(spnPeriod.Value)  
    lngDispTMax = AnaPeriodArray(spnPeriod.Value + 1)  
    frmTgisPanel.AdjustMin  
    frmTgisPanel.AdjustMax  
    frmTgisPanel.Repaint  
  
    'Then call probability calculator  
    Call frmTgisPanel.ProbCalc  
  
Exit Sub  
  
    'Error handler  
ErrorAdjust:  
  
    ErrorHandler Err.Number  
    Err.Clear  
  
Exit Sub  
  
End Sub  
  
'External subroutine to handle errors  
Private Sub ErrorHandler(ErrorNumber As Long)  
  
    'Error selector  
    Select Case ErrorNumber  
    'Insert specific error cases to catch here  
    Case Else  
        MsgBox "Error: " & Err.Number & " - " & Err.Description, , "T-GIS"  
    End Select  
  
End Sub
```