

Create_output - 21

```
' Arr is an unallocated array, or  
' Dimension is greater than the number of dimension of Arr, or  
' Dimension is less than 1.  
' This function does not support arrays of user-defined Type variables.  
' .....
```

```
Dim NumDimensions As Long
```

```
' if Arr is not an array, return 0 and get out.  
If IsArray(Arr) = False Then  
    NumElements = 0  
    Exit Function  
End If
```

```
' ensure that Dimension is at least 1.  
If Dimension < 1 Then  
    NumElements = 0  
    Exit Function  
End If
```

```
' get the number of dimensions  
NumDimensions = NumberofArrayDimensions(Arr)  
If NumDimensions < Dimension Then  
    NumElements = 0  
    Exit Function  
End If
```

```
' returns the number of elements in the array  
NumElements = UBound(Arr, Dimension) - LBound(Arr, Dimension) + 1  
End Function
```

frmExportchofix - 1

```
' In this form the User chooses to save the matrix or to continue with the statistical analysis (FIX  
matrix creation process)  
' Subroutine: cmdContinue_Click  
' Author: M.Ferretti  
' Date: May 2010  
' Description: This subroutine ends the matrix creation process and start the statistical analysis
```

```
Private Sub cmdContinue_Click()
```

```
MkDir "C:\TATfolder2"  
MkDir "C:\TATfolder2\fix"  
MkDir "C:\TATfolder2\fix\split"  
MkDir "C:\TATfolder2\fix\singlefeatures"  
MkDir "C:\TATfolder2\fix\singletables"
```

```
' GP library (modified)  
Dim GP As IGeoProcessor  
Set GP = New GeoProcessor  
GP.OverwriteOutput = True
```

```
' Declare and set a variant array to hold the parameters  
Dim parameters As IVariantArray  
Set parameters = New VarArray
```

```
' Populate the variant array with the parameters  
parameters.Add "c:\TATfolder\Fixanalysis.shp" '(in_features)  
parameters.Add "c:\TATfolder\Fixanalysis.shp" '(join_field)
```

```
' Now execute the Copy tool  
GP.Execute "copy", parameters, Nothing  
deleteTATfoldersfix
```

```
Dim pMxDoc As IMxDocument  
Dim pMap As IMap  
Dim pWorkspaceFactory As IWorkspaceFactory  
Dim pFeatureWorkspace As IFeatureWorkspace  
Dim pFeatureLayer As IFeatureLayer  
Dim pFeatureClass As IFeatureClass
```

```
' Specify the workspace and the feature class  
Set pWorkspaceFactory = New ShapefileWorkspaceFactory  
Set pFeatureWorkspace = pWorkspaceFactory.OpenFromFile("c:\TATfolder2", 0)  
Set pFeatureClass = pFeatureWorkspace.OpenFeatureClass("Fixanalysis")
```

```
' Prepare a feature layer  
Set pFeatureLayer = New FeatureLayer  
Set pFeatureLayer.FeatureClass = pFeatureClass  
pFeatureLayer.Name = pFeatureLayer.FeatureClass.AliasName
```

```
' Add the feature layer to the active map  
Set pMxDoc = ThisDocument  
Set pMap = pMxDoc.FocusMap  
pMap.AddLayer pFeatureLayer
```

```
' Refresh the active view  
pMxDoc.ActiveView.Refresh  
fixanalysisstatot
```

```
End Sub
```

```
Private Sub CommandExport_Click ()
```

```
SaveDialog 'call the export process
```

```
deleteTATfoldersfix
```

```
End
```

```
End Sub
```

```
Private Sub cmdCancel_Click ()
```

```
deleteTATfoldersfix
```

End

End Sub

```

' In this form the User chooses to save the matrix or to continue with the statistical analysis (HR
matrix creation process)

' Subroutine: cmdContinue_Click
' Author: M.Ferretti
' Date: May 2010
' Description: This subroutine ends the matrix creation process and start the statistical analysis

Private Sub cmdContinue_Click()

Mkdir "C:\TATfolder2"
Mkdir "C:\TATfolder2\hr"
Mkdir "C:\TATfolder2\hr\split"
Mkdir "C:\TATfolder2\hr\singlefeatures"
Mkdir "C:\TATfolder2\hr\singletables"

' GP library (modified)
Dim GP As IGeoProcessor
Set GP = New GeoProcessor

GP.OverwriteOutput = True

' Declare and set a variant array to hold the parameters
Dim parameters As IVariantArray
Set parameters = New VarArray

' Populate the variant array with the parameters
parameters.Add "c:\TATfolder\Hranalysis.shp" '(in_features)
parameters.Add "c:\TATfolder\Hranalysis.shp" '(join_field)

' Now execute the Copy tool
GP.Execute "copy", parameters, Nothing

deleteTATfoldershr

Dim pMxDoc As IMxDocument
Dim pMap As IMap
Dim pWorkspaceFactory As IWorkspaceFactory
Dim pFeatureWorkspace As IFeatureWorkspace
Dim pFeatureLayer As IFeatureLayer
Dim pFeatureClass As IFeatureClass

' Specify the workspace and the feature class
Set pWorkspaceFactory = New ShapefileWorkspaceFactory
Set pFeatureWorkspace = pWorkspaceFactory.OpenFromFile("c:\TATfolder2\", 0)
Set pFeatureClass = pFeatureWorkspace.OpenFeatureClass("Hranalysis")

' Prepare a feature layer.
Set pFeatureLayer = New FeatureLayer
Set pFeatureLayer.FeatureClass = pFeatureClass
pFeatureLayer.Name = pFeatureLayer.FeatureClass.AliasName

' Add the feature layer to the active map.
Set pMxDoc = ThisDocument
Set pMap = pMxDoc.FocusMap
pMap.AddLayer pFeatureLayer

' Refresh the active view.
pMxDoc.ActiveView.Refresh

hranalysisstottot

End Sub

Private Sub CommandExport_Click ()

SaveDialog 'call the export process

deleteTATfoldershr

End

End Sub

Private Sub cmdCancel_Click ()

deleteTATfoldershr

```

End

End Sub

frmFix - 1

```

' In this form the User chooses the fixes field code (FIX matrix creation process)
' Subroutine: UserForm_Initialize
' Author: M.Ferretti
' Date: May 2010
' Description: This subroutine populates the combobox with the fields present in the layer selected
from the User
Private Sub UserForm_Initialize()
' Add items to the dropdown list
Dim pMxDoc As IMxDocument
Set pMxDoc = ThisDocument
Dim pFeatureLayer As IFeatureLayer
Set pFeatureLayer = pMxDoc.FocusMap.Layer(0) '1st layer
Dim i As Integer

For i = 0 To pFeatureLayer.FeatureClass.Fields.FieldCount - 1
    cboUnits.AddItem pFeatureLayer.FeatureClass.Fields.Field(i).Name
Next
End Sub

' Subroutine: cmdRun_Click
' Author: M.Ferretti
' Date: May 2010
' Description: This subroutine copies the field selected from the User in a new file named "Animals
"
' AddeleteField and CalculateField from Programming ArcObjects with VBA: a task-oriented approach
- Kang-Tsung Chang (modified)
Private Sub cmdRun_Click()
' Part 1: Define the feature class
Dim pMxDoc As IMxDocument
Dim pFeatureLayer As IFeatureLayer
Dim pFeatureClass As IFeatureClass
Set pMxDoc = ThisDocument
Set pFeatureLayer = pMxDoc.FocusMap.Layer(0)
Set pFeatureClass = pFeatureLayer.FeatureClass

' Part 2: Add new field
Dim pField1 As IFieldEdit

' Define the new field
Set pField1 = New Field
pField1.Name = "Animals"
pField1.Type = esriFieldTypeDouble
pField1.Length = 8
pFeatureClass.AddField pField1

' Part 3: Calculate the new field values.
Dim pCursor As ICursor
Dim pCalculator As ICalculator
Dim sSelectedField As String

' Prepare a cursor with all records.
Set pCursor = pFeatureClass.Update(Nothing, True)

' Define a calculator.
Set pCalculator = New Calculator
Set pCalculator.Cursor = pCursor
sSelectedField = cboUnits.Text

' Eliminate the double quotes
Dim newsSelectedField As String
newsSelectedField = "[" & sSelectedField & "]"

' Calculate the field values
pCalculator.Expression = newsSelectedField
pCalculator.Field = "Animals"
pCalculator.Calculate

' Close the combobox
frmFix.Hide
End Sub

```

```

Private Sub cmdCancel_Click()
' End the Appennino Tool
Const MyPath = "C:\TATFolder\orfilesmod\" ' Set the path.
Const FileType = "*.shp"
ProcessFilesdelete MyPath, FileType
Kill "C:\TATFolder\orfilesmod*"
Rmdir "C:\TATFolder\orfilesmod"
Rmdir "C:\TATFolder\intersectfiles"
Rmdir "C:\TATFolder\hr"
Rmdir "C:\TATFolder\fix"
Rmdir "C:\TATFolder"
End

End Sub

Sub ProcessFilesdelete(strFolder As String, strFilePattern As String)
' Delete the create files
' http://www.vbaexpress.com/kb/getarticle.php?kb_id=245 (modified)

Dim strFileName As String
Dim strFolders() As String
Dim iFolderCount As Integer
Dim i As Integer

' Collect child folders
strFileName = dir$(strFolder & "\", vbDirectory)
Do Until strFileName = ""
If (GetAttr(strFolder & "\" & strFileName) And vbDirectory) = vbDirectory Then
If Left$(strFileName, 1) <> "." Then
ReDim Preserve strFolders(iFolderCount)
strFolders(iFolderCount) = strFolder & "\" & strFileName
iFolderCount = iFolderCount + 1
End If
End If
Loop

' Process files in current folder
strFileName = dir$(strFolder & "\" & strFilePattern)
Do Until strFileName = ""
'Do things with files here*****
Dim Filex As String
Dim Filex2 As String
Dim Filex3 As String
Dim Filex4 As String
Filex = strFolder & "\" & strFileName
Filex2 = strFileName
Filex3 = Len(Filex2) ' count the lenght and cancel 4 characters ".shp"
Filex4 = Left(Filex2, Filex3 - 4)

' GP library (modified)
Dim GP As IGeoProcessor
Set GP = New GeoProcessor

GP.OverwriteOutput = True

' Declare and set a variant array to hold the parameters
Dim parameters As IVariantArray
Set parameters = New VarArray

' Populate the variant array with the parameters
parameters.Add Filex4

' Now execute the Delete tool
GP.Execute "delete_management", parameters, Nothing

```

```

strFileName = dir$()
Loop
' Look through child folders
For i = 0 To iFolderCount - 1
ProcessFilesdelete strFolders(i), strFilePattern
Next i
End Sub

```

frmFix_layer - 1

```
' In this form the User chooses the animal's fixes shape file (FIX matrix creation process)
' Subroutine: UserForm_Initialize
' Author: M.Ferretti
' Date: May 2010
' Description: This subroutine, and the following function, populates the combobox with the layers
present in the TOC
' http://forums.esri.com/Thread.asp?c=93&f=992&t=212356&mc=2#msgid640862 (modified)

Public Sub UserForm_Initialize()
Dim pMxDoc As IMxDocument
Dim pID As New UID
Dim pEnumLyr As IEnumLayer
Dim pLyr As ILayer

' Set the UID for GeoFeatureLayers
pID.Value = "[E156D7B5-22AF-11D3-9F99-00C04F6BC78E]"

' Get an enumeration on ALL the feature layers (TRUE = iterate through group layers)
Set pMxDoc = ThisDocument
Set pEnumLyr = pMxDoc.FocusMap.Layers(pID, True)

' Loop thru the returned layers and add the names to the combo box.
pEnumLyr.Reset
Set pLyr = pEnumLyr.Next
While Not pLyr Is Nothing
    ComboBox1.AddItem pLyr.Name
    Set pLyr = pEnumLyr.Next
Wend

End Sub

Public Function FindFPlayerByName(LayerName As String) As IFeatureLayer

Dim pEnumLayer As IEnumLayer
Dim pCompositeLayer As ICompositeLayer
Dim i As Integer
Dim pMxDoc As IMxDocument
Dim pMap As IMap

Set pMxDoc = Application.Document
Set pMap = pMxDoc.FocusMap

Set pEnumLayer = pMap.Layers
pEnumLayer.Reset

Dim pLayer As ILayer
Set pLayer = pEnumLayer.Next

Do While Not pLayer Is Nothing
    If TypeOf pLayer Is ICompositeLayer Then
        Set pCompositeLayer = pLayer
        For i = 0 To pCompositeLayer.count - 1
            With pCompositeLayer
                If .Layer(i).Name = LayerName Then
                    If TypeOf .Layer(i) Is IFeatureLayer Then
                        Set FindFPlayerByName = pCompositeLayer.Layer(i)
                    End Function
                End If
            End If
        End With
    Next i
    ElseIf pLayer.Name = LayerName And TypeOf pLayer Is IFeatureLayer Then
        Set FindFPlayerByName = pLayer
    End Function
Exit Function
End If
Set pLayer = pEnumLayer.Next
Loop
End Function

' Subroutine: cmdRun_Click
' Author: M.Ferretti
' Date: May 2010
' Description: This subroutine copies and renames the User selected fixes shape file (to preserve t
he original User's file), add the file in the TOC and move the file to the "0" position
```

frmFix_layer - 2

```
' GP Library (modified)
Private Sub cmdRun_Click()
Dim pFPlayer As IFeatureLayer
Set pFPlayer = FindFPlayerByName(frmFix_layer.ComboBox1.Text)

' GP library (modified)
Dim GP As IGeoProcessor
Set GP = New GeoProcessor

GP.OverwriteOutput = True

' Declare and set a variant array to hold the parameters
Dim parameters As IVariantArray
Set parameters = New VarArray

' Populate the variant array with the parameters
parameters.Add pFPlayer '(in_features)
parameters.Add "c:\TATfolder\orfilesmod\fixmod.shp"

' Now execute the Copy tool
GP.Execute "copyfeatures", parameters, Nothing

' Sub MoveLayer()
' http://forums.esri.com/Thread.asp?c=93&f=993&t=118360&mc=9#msgid340233

Dim pMxDocument As IMxDocument
Dim pMap As IMap
Dim pLayer As ILayer
Dim x As Long

Set pMxDocument = Application.Document
Set pMap = pMxDocument.FocusMap
Set pLayer = pMxDocument.SelectedLayer

pMap.MoveLayer pLayer, pMap.Layercount - pMap.Layercount

' Close the combobox
frmFix_layer.Hide

End Sub

Private Sub cmdCancel_Click()

' End the Appennino Tool

Const MyPath = "C:\TATfolder\orfilesmod\" ' Set the path.
Const FileType = "*.shp"
ProcessFilesDelete MyPath, FileType

Kill "C:\TATfolder\orfilesmod*"

Rmdir "C:\TATfolder\orfilesmod"
Rmdir "C:\TATfolder\intersectfiles"
Rmdir "C:\TATfolder\hr"
Rmdir "C:\TATfolder\fix"
Rmdir "C:\TATfolder"
End

End Sub

Sub ProcessFilesDelete(strFolder As String, strFilePattern As String)

' Delete the create files
' http://www.vbaexpress.com/kb/getarticle.php?kb_id=245 (modified)

Dim strFileName As String
Dim strFolders() As String
Dim iFolderCount As Integer
Dim i As Integer

' Collect child folders
strFileName = dir$(strFolder & "\", vbDirectory)
Do Until strFileName = ""
    If (GetAttr(strFolder & "\" & strFileName) And vbDirectory) = vbDirectory Then
        If Left$(strFileName, 1) <> "." Then

```

frmFix_layer - 3

```
ReDim Preserve strFolders(iFolderCount)
strFolders(iFolderCount) = strFolder & "\" & strFileName
iFolderCount = iFolderCount + 1
End If
End If
strFileName = dir$( )
Loop

'process files in current folder
strFileName = dir$(strFolder & "\" & strFilePattern)
Do Until strFileName = ""
'Do things with files here*****
Dim Filex As String
Dim Filex2 As String
Dim Filex3 As String
Dim Filex4 As String
Filex = strFolder & "\" & strFileName
Filex2 = strFileName
Filex3 = Len(Filex2) ' count the lenght and cancel 4 caracthers ".shp"
Filex4 = Left(Filex2, Filex3 - 4)

' GP library (modified)
Dim GP As IGeoProcessor
Set GP = New GeoProcessor

GP.OverwriteOutput = True

' Declare and set a variant array to hold the parameters
Dim parameters As IVariantArray
Set parameters = New VarArray

' Populate the variant array with the parameters
parameters.Add Filex4

' Now execute the Dissolve tool
GP.Execute "delete_management", parameters, Nothing

'*****
strFileName = dir$( )
Loop

' Look through child folders
For i = 0 To iFolderCount - 1
ProcessFilesdelete strFolders(i), strFilePattern
Next i
End Sub
```

frmHome_range - 1

```
' In this form the User chooses the home range field code(HR and FIX matrix creation process)
' Subroutine: UserForm_Initialize
' Author: M.Ferretti
' Date: May 2010
' Description: This subroutine populates the combobox with the fields present in the layer selected
from the User

Private Sub UserForm_Initialize()
' Add items to the dropdown list
Dim pMxDoc As IMxDocument
Set pMxDoc = ThisDocument
Dim pFeatureLayer As IFeatureLayer
Set pFeatureLayer = pMxDoc.FocusMap.Layer(0) '1st layer
Dim i As Integer

For i = 0 To pFeatureLayer.FeatureClass.Fields.FieldCount - 1
cbOnUnits.AddItem pFeatureLayer.FeatureClass.Fields.Field(i).Name
Next

End Sub

' Subroutine: cmdRun_Click
' Author: M.Ferretti
' Date: May 2010
' Description: This subroutine copies the field selected from the User in a new file named "Animals"
"
' AddeleteField and CalculateField from Programming ArcObjects with VBA: a task-oriented approach
- Kang-tsung Chang (modified)

Private Sub cmdRun_Click()
' Part 1: Define the feature class
Dim pMxDoc As IMxDocument
Dim pFeatureLayer As IFeatureLayer
Dim pFeatureClass As IFeatureClass
Set pMxDoc = ThisDocument
Set pFeatureLayer = pMxDoc.FocusMap.Layer(0)
Set pFeatureClass = pFeatureLayer.FeatureClass

' Part 2: Add new field
Dim pField1 As IFieldEdit

' Define the new field
Set pField1 = New Field
pField1.Name = "Animals"
pField1.Type = esriFieldTypeDouble
pField1.Length = 8
pFeatureClass.AddField pField1

' Part 3: Calculate the new field values
Dim pCursor As ICursor
Dim pCalculator As ICalculator
Dim sSelectedField As String

' Prepare a cursor with all records
Set pCursor = pFeatureClass.Update(Nothing, True)

' Define a calculator
Set pCalculator = New Calculator
Set pCalculator.Cursor = pCursor
sSelectedField = cbOnUnits.Text

' Eliminate the double quotes
Dim newsSelectedField As String
newsSelectedField = "[" & sSelectedField & "]"

' Calculate the field values
pCalculator.Expression = newsSelectedField
pCalculator.Field = "Animals"
pCalculator.Calculate

' Close the form
frmHome_range.Hide
End Sub
```

```

Private Sub cmdCancel_Click()
' End the Appennino Tool
Const MyPath = "C:\TATFolder\orfilesmod\" ' Set the path.
Const FileType = "*.shp"
ProcessFilesdelete MyPath, FileType
Kill "C:\TATFolder\orfilesmod*"
Rmdir "C:\TATFolder\orfilesmod"
Rmdir "C:\TATFolder\intersectfiles"
Rmdir "C:\TATFolder\hr"
Rmdir "C:\TATFolder\fix"
Rmdir "C:\TATFolder"
End
End Sub
Sub ProcessFilesdelete(strFolder As String, strFilePattern As String)
' Delete the create files
' http://www.vbaexpress.com/kb/getarticle.php?kb_id=245 (modified)
Dim strFileName As String
Dim strFolders() As String
Dim iFolderCount As Integer
Dim i As Integer
' Collect child folders
strFileName = dir$(strFolder & "\", vbDirectory)
Do Until strFileName = ""
If (GetAttr(strFolder & "\" & strFileName) And vbDirectory) = vbDirectory Then
If Left$(strFileName, 1) <> "." Then
ReDim Preserve strFolders(iFolderCount)
strFolders(iFolderCount) = strFolder & "\" & strFileName
iFolderCount = iFolderCount + 1
End If
End If
strFileName = dir$()
Loop
' Process files in current folder
strFileName = dir$(strFolder & "\" & strFilePattern)
Do Until strFileName = ""
' Do things with files here*****
Dim Filex As String
Dim Filex2 As String
Dim Filex3 As String
Dim Filex4 As String
Filex = strFolder & "\" & strFileName
Filex2 = strFileName
Filex3 = Len(Filex2) ' count the lenght and cancel 4 caracethers ".shp"
Filex4 = Left(Filex2, Filex3 - 4)
' GP library (modified)
Dim GP As IGeoProcessor
Set GP = New GeoProcessor
GP.OverwriteOutput = True
' Declare and set a variant array to hold the parameters
Dim parameters As IVariantArray
Set parameters = New VariantArray
' Populate the variant array with the parameters
parameters.Add Filex4
' Now execute the Delete tool
GP.Execute "delete_management", parameters, Nothing
*****
strFileName = dir$()

```

```

Loop
' Look through child folders
For i = 0 To iFolderCount - 1
ProcessFilesdelete strFolders(i), strFilePattern
Next i
End Sub

```

```
frmHome_range_layer - 2
rve the original User's file), add the file in the TOC and move the file to the "0" position
' GP Library (modified)
Private Sub cmdRun_Click()
Dim pFLayer As IFeatureLayer
Set pFLayer = FindFLayerByName(frmHome_range_layer.ComboBox1.Text)
' GP library (modified)
Dim GP As IGeoProcessor
Set GP = New GeoProcessor
GP.OverrideOutput = True
' Declare and set a variant array to hold the parameters
Dim parameters As IVariantArray
Set parameters = New VarArray
' Populate the variant array with the parameters
parameters.Add pFLayer '(in features)
parameters.Add "c:\TATfolder\orfilesmod\home_rangemod.shp"
' Now execute the Copy tool
GP.Execute "copyfeatures", parameters, Nothing
' Sub MoveLayer()
'http://forums.esri.com/Thread.asp?c=93&f=993&t=118360&nc=9#msgid340233
Dim pMXDoc As IMXDocument
Dim pMap As IMap
Dim pLayer As ILayer
Dim x As Long
Set pMXDoc = Application.Document
Set pMap = pMXDoc.FocusMap
Set pLayer = pMXDoc.SelectedLayer
pMap.MoveLayer pLayer, pMap.LayerCount - pMap.LayerCount
' Close the combobox
frmHome_range_layer.Hide
End Sub
Private Sub cmdCancel_Click()
' End the Appennino Tool
Const MyPath = "C:\TATfolder\orfilesmod\" ' Set the path.
Const FileType = "*.shp"
ProcessFilesdelete MyPath, FileType
Kill "C:\TATfolder\orfilesmod\*"
Rmdir "C:\TATfolder\orfilesmod"
Rmdir "C:\TATfolder\intersectfiles"
Rmdir "C:\TATfolder\hr"
Rmdir "C:\TATfolder\fix"
Rmdir "C:\TATfolder"
End
End Sub
Sub ProcessFilesdelete(strFolder As String, strFilePattern As String)
' Delete the create files
' http://www.vbaexpress.com/kb/getarticle.php?kb_id=245 (modified)
Dim strFileName As String
Dim strFolders() As String
Dim iFolderCount As Integer
Dim i As Integer
' Collect child folders
strFileName = dir$(strFolder & "\", vbDirectory)
Do Until strFileName = ""
frmHome_range_layer - 1
' In this form the User chooses the animal's home ranges shape file (HR and FIX matrix creation process)
' Subroutine: UserForm_Initialize
' Author: M.Ferretti
' Date: May 2010
' Description: This subroutine, and the following function, populates the combobox with the layers present in the TOC
' http://forums.esri.com/Thread.asp?c=93&f=992&t=212356&nc=2#msgid640862 (modified)
Public Sub UserForm_Initialize()
Dim pMXDoc As IMXDocument
Dim pID As New UID
Dim pEnumLyr As IEnumLayer
Dim pLyr As ILayer
' Set the UID for GeoFeatureLayers
pID.Value = "[E156D7E5-22AF-11D3-9F99-00C04F6BC78E]"
' Get an enumeration on ALL the feature layers (TRUE = iterate through group layers)
Set pMXDoc = ThisDocument
Set pEnumLyr = pMXDoc.FocusMap.Layers(pID, True)
' Loop thru the returned layers and add the names to the combo box.
pEnumLyr.Reset
Set pLyr = pEnumLyr.Next
While Not pLyr Is Nothing
ComboBox1.AddItem pLyr.Name
Set pLyr = pEnumLyr.Next
Wend
End Sub
Public Function FindFLayerByName(LayerName As String) As IFeatureLayer
Dim pEnumLayer As IEnumLayer
Dim pCompositeLayer As ICompositeLayer
Dim i As Integer
Dim pMXDoc As IMXDocument
Dim pMap As IMap
Set pMXDoc = Application.Document
Set pMap = pMXDoc.FocusMap
Set pEnumLayer = pMap.Layers
pEnumLayer.Reset
Dim pLayer As ILayer
Set pLayer = pEnumLayer.Next
Do While Not pLayer Is Nothing
If TypeOf pLayer Is ICompositeLayer Then
Set pCompositeLayer = pLayer
For i = 0 To pCompositeLayer.count - 1
With pCompositeLayer
If .Layer(i).Name = LayerName Then
If TypeOf .Layer(i) Is IFeatureLayer Then
Set FindFLayerByName = pCompositeLayer.Layer(i)
Exit Function
End If
End If
End If
Next i
ElseIf pLayer.Name = LayerName And TypeOf pLayer Is IFeatureLayer Then
Set FindFLayerByName = pLayer
Exit Function
End If
Set pLayer = pEnumLayer.Next
Loop
End Function
' Subroutine: cmdRun_Click
' Author: M.Ferretti
' Date: May 2010
' Description: This subroutine copies and renames the User selected home range shape file (to prese
```


frmHome_range_layer - 3

```
' Process files in current folder
strFileName = dir$(strFolder & "\" & strFilePattern)
Do Until strFileName = ""
    'Do things with files here*****
    Dim Filex As String
    Dim Filex2 As String
    Dim Filex3 As String
    Dim Filex4 As String
    Filex = strFolder & "\" & strFileName
    Filex2 = strFilePattern
    Filex3 = Len(Filex2)
    Filex4 = Left(Filex2, Filex3 - 4)

    ' GP library (modified)
    Dim GP As IGeoProcessor
    Set GP = New GeoProcessor

    GP.OverwriteOutput = True

    'Declare and set a variant array to hold the parameters
    Dim parameters As IVariantArray
    Set parameters = New VarArray

    'Populate the variant array with the parameters
    parameters.Add Filex4

    'Now execute the Dissolve tool
    GP.Execute "delete_management", parameters, Nothing

    *****
    strFileName = dir$(
Loop
    'Look through child folders
    For i = 0 To iFolderCount - 1
        ProcessFilesdelete strFolders(i), strFilePattern
    Next i
End Sub
```

frmHrOrFix - 1

```
'In this form the User chooses the Home Range (polygon) analysis or the Fix (point) analysis
Public Sub OptionButton1_Click()
    ' Open the Subroutine Appennino_tool_general_hr in the "Create_matrix" module
    Appennino_tool_general_hr

    ' Close the Choose Form
    frmHrOrFix.Hide

End Sub

Private Sub OptionButton2_Click()

    ' Open the Subroutine Appennino_tool_general_fix in the "Create_matrix" module
    Appennino_tool_general_fix

    ' Close the Choose Form
    frmHrOrFix.Hide

End Sub

Private Sub cmdCancel_Click()

    'End the Appennino Tool
    End

End Sub
```

```

frmLand_use - 1
' In this form the User chooses the land use field code(HR and FIX matrix creation process)
' Subroutine: UserForm_Initialize
' Author: M.Ferretti
' Date: May 2010
' Description: This subroutine populates the combobox with the fields present in the layer selected
from the User

Private Sub UserForm_Initialize()
' Add items to the dropdown list
Dim pMxDoc As IMxDocument
Set pMxDoc = ThisDocument
Dim pFeatureLayer As IFeatureLayer
Set pFeatureLayer = pMxDoc.FocusMap.Layer(0) '1st layer
Dim i As Integer

For i = 0 To pFeatureLayer.FeatureClass.Fields.FieldCount - 1
cboUnits.AddItem pFeatureLayer.FeatureClass.Fields.Field(i).Name
Next

End Sub

' Subroutine: cmdRun_Click
' Author: M.Ferretti
' Date: May 2010
' Description: This subroutine copies the field selected from the User in a new file named "land_us
e"
' AddDeleteField and CalculateField from Programming ArcObjects with VBA: a task-oriented approach
- Kang-Tsung Chang (modified)

Private Sub cmdRun_Click()
' Part 1: Define the feature class
Dim pMxDoc As IMxDocument
Dim pFeatureLayer As IFeatureLayer
Dim pFeatureClass As IFeatureClass
Set pMxDoc = ThisDocument
Set pFeatureLayer = pMxDoc.FocusMap.Layer(0)
Set pFeatureClass = pFeatureLayer.FeatureClass

' Part 2: Add new field
Dim pField1 As IFieldEdit
' Define the new field.
Set pField1 = New Field
pField1.Name = "Land use"
pField1.Type = esriFieldTypeDouble
pField1.Length = 8
pFeatureClass.AddField pField1

' Part 3: Calculate the new field values
Dim pCursor As ICursor
Dim pCalculator As ICalculator
Dim sSelectedField As String

' Prepare a cursor with all records
Set pCursor = pFeatureClass.Update(Nothing, True)

' Define a calculator
Set pCalculator = New Calculator
Set pCalculator.Cursor = pCursor
sSelectedField = cboUnits.Text

' Eliminate the double quotes
Dim newsSelectedField As String
newsSelectedField = "[" & sSelectedField & "]"

'Calculate the field values
pCalculator.Expression = newsSelectedField
pCalculator.Field = "Land_use"
pCalculator.Calculate

'Close the form
frmLand_use.Hide

End Sub

frmLand_use - 2
Private Sub cmdCancel_Click()
' End the Appennino Tool

Const MyPath = "C:\TATFolder\orfilesmod\" ' Set the path.
Const FileType = "*.shp"
ProcessFilesdelete MyPath, FileType

Kill "C:\TATFolder\orfilesmod\"

Rmdir "C:\TATFolder\orfilesmod"
Rmdir "C:\TATFolder\intersectfiles"
Rmdir "C:\TATFolder\hr"
Rmdir "C:\TATFolder\fix"
Rmdir "C:\TATFolder"

End

End Sub

Sub ProcessFilesdelete(strFolder As String, strFilePattern As String)
' Delete the create files
' http://www.vbaexpress.com/kb/getarticle.php?kb_id=245 (modified)

Dim strFileName As String
Dim strFolders() As String
Dim iFolderCount As Integer
Dim i As Integer

' Collect child folders
strFileName = dir$(strFolder & "\\", vbDirectory)
Do Until strFileName = ""
If (GetAttr(strFolder & "\" & strFileName) And vbDirectory) = vbDirectory Then
If Left$(strFileName, 1) <> "." Then
ReDim Preserve strFolders(iFolderCount)
strFolders(iFolderCount) = strFolder & "\" & strFileName
iFolderCount = iFolderCount + 1
End If
End If
strFileName = dir$()
Loop

' Process files in current folder
strFileName = dir$(strFolder & "\" & strFilePattern)
Do Until strFileName = ""
'Do things with files here*****
Dim Filex As String
Dim Filex2 As String
Dim Filex3 As String
Dim Filex4 As String
Filex = strFolder & "\" & strFileName
Filex2 = strFilePattern
Filex3 = Len(Filex2)
Filex4 = Left(Filex2, Filex3 - 4)

' GP append (GP Library)
Dim GP As IGeoprocessor
Set GP = New Geoprocessor

GP.OverwriteOutput = True

' Declare and set a variant array to hold the parameters
Dim parameters As IVariantArray
Set parameters = New VarArray

' Populate the variant array with the parameters
parameters.Add Filex4

' Now execute the Dissolve tool
GP.Execute "delete_management", parameters, Nothing

```

frmLand_use - 3

```
'*****  
strFileName = dir$(  
Loop  
' Look through child folders  
For i = 0 To iFolderCount - 1  
ProcessFilesdelete strFolders(i), strFilePattern  
Next i
```

End Sub

frmLand_use_layer - 1

```
' In this form the User chooses the Land Use shape file (HR and FIX matrix creation process)  
' Subroutine: UserForm_Initialize  
' Author: M.Ferretti  
' Date: May 2010  
' Description: This subroutine, and the following function, populates the combobox with the layers  
present in the TOC  
' http://forums.esri.com/Thread.asp?c=93&f=992&t=212356&mc=2#msgid640862 (modified)  
  
Public Sub UserForm_Initialize()  
  
Dim pMxDoc As IMxDocument  
Dim pID As New UID  
Dim pEnumLyr As IEnumLayer  
Dim pLyr As ILayer  
  
' Set the UID for GeoFeatureLayers  
pID.value = "[E156D7B5-22AF-11D3-9F99-00C04F6BC78E]"  
  
' Get an enumeration on ALL the feature layers (TRUE = iterate through group layers)  
Set pMxDoc = ThisDocument  
Set pEnumLyr = pMxDoc.FocusMap.Layers(pID, True)  
  
' Loop thru the returned layers and add the names to the combo box.  
pEnumLyr.Reset  
Set pLyr = pEnumLyr.Next  
While Not pLyr Is Nothing  
    ComboBox1.AddItem pLyr.Name  
    Set pLyr = pEnumLyr.Next  
Wend
```

End Sub

```
Public Function FindFLayerByName(LayerName As String) As IFeatureLayer  
Dim pEnumLayer As IEnumLayer  
Dim pCompositeLayer As ICompositeLayer  
Dim i As Integer  
Dim pMxDoc As IMxDocument  
Dim pMap As IMap  
  
Set pMxDoc = Application.Document  
Set pMap = pMxDoc.FocusMap  
  
Set pEnumLayer = pMap.Layers  
pEnumLayer.Reset  
  
Dim pLayer As ILayer  
Set pLayer = pEnumLayer.Next  
  
Do While Not pLayer Is Nothing  
    If TypeOf pLayer Is ICompositeLayer Then  
        Set pCompositeLayer = pLayer  
        For i = 0 To pCompositeLayer.count - 1  
            With pCompositeLayer  
                If .Layer(i).Name = LayerName Then  
                    Set FindFLayerByName = pCompositeLayer.Layer(i)  
                    Exit Function  
                End If  
            End If  
            End With  
        Next i  
    ElseIf pLayer.Name = LayerName And TypeOf pLayer Is IFeatureLayer Then  
        Set FindFLayerByName = pLayer  
        Exit Function  
    End If  
    Set pLayer = pEnumLayer.Next  
Loop  
End Function
```

' Subroutine: cmdRun_Click

```
' Author: M.Ferretti  
' Date: May 2010  
' Description: This subroutine copies and renames the User selected Land Use shape file (to preserv  
e the original User's file), add the file in the TOC and move the file to the "0" position  
' GP Library (modified)
```

frmPres - 1

```
Private Sub Continuebutton_Click()  
frmPres.Hide 'This command closes the Presentation Form (frmPres)  
End Sub  
Private Sub Exitbutton_Click()  
End  
'This command ends the Appennino Tool  
End Sub
```

frmLand_use_layer - 2

```
Private Sub cmdRun_Click()  
' Select the layer selected in the combobox  
Dim pFLayer As IFeatureLayer  
Set pFLayer = FindFLayerByName(FrmLand_use_layer.ComboBox1.Text)  
  
' Copy the selected layer in the tool folder (GP library)  
  
Dim GP As IGeoProcessor  
Set GP = New GeoProcessor  
GP.OverwriteOutput = True  
  
' Declare and set a variant array to hold the parameters  
Dim parameters As IVariantArray  
Set parameters = New VarArray  
  
' Populate the variant array with the parameters  
parameters.Add pFLayer '(in_features)  
parameters.Add "c:\TATfolder\orfilesmod\land_usemod.shp"  
  
' Now execute the Copy tool  
GP.Execute "copyfeatures", parameters, Nothing  
  
' Sub MoveLayer ()  
' http://forums.esri.com/Thread.asp?c=93&f=993&t=118360&mc=9#msgid340233  
Dim pMXDocument As IMXDocument  
Dim pMap As IMap  
Dim pLayer As ILayer  
Dim x As Long  
  
Set pMXDocument = Application.Document  
Set pMap = pMXDocument.FocusMap  
Set pLayer = pMXDocument.SelectedLayer  
  
' Move to the "0" position in the TOC  
pMap.MoveLayer pLayer, pMap.Layercount - pMap.Layercount  
  
' Close the combobox  
FrmLand_use_layer.Hide  
End Sub  
  
Private Sub cmdCancel_Click()  
End the Appennino Tool  
  
Rmdir "C:\TATfolder\orfilesmod"  
Rmdir "C:\TATfolder\intersectfiles"  
Rmdir "C:\TATfolder\hr"  
Rmdir "C:\TATfolder\fix"  
Rmdir "C:\TATfolder"  
End  
End Sub
```

```

' In this form the tool products the One-way Anova analysis file, a table with mean and standard de
viation and a graph (FIX process)

' Subroutine: UserForm_Initialize
' Author: M.Ferretti
' Date: May 2010
' Description: This subroutine prepares the two dimensional array for the One-way Anova analysis

Private Sub UserForm_Initialize()
Dim pMxDoc As IMxDocument
Dim pMap As IMap
Dim pFeatureLayer As IFeatureLayer
Dim pFeatClass As IFeatureClass
Dim pRow As IRow

Set pMxDoc = ThisDocument
Set pMap = pMxDoc.FocusMap

' Define the input feature class
Set pFeatureLayer = pMap.Layer(0)
Set pFeatClass = pFeatureLayer.FeatureClass

' Count the record number
Dim pLayerDef As IFeatureLayerDefinition2
Set pLayerDef = pFeatureLayer
Dim pQFilter As IQueryFilter
Set pQFilter = New QueryFilter
pQFilter.WhereClause = pLayerDef.DefinitionExpression
Dim count As Long
count = pFeatureLayer.FeatureClass.FeatureCount(pQFilter)
countfields = pFeatureLayer.FeatureClass.Fields.FieldCount

' Set the array
Dim pCursor As ICursor
Dim NameArray() As Double
ReDim NameArray(count - 1, countfields - 3) As Double

'Add a the Array two field values
Set pCursor = pFeatClass.Update(Nothing, True)
Set pRow = pCursor.NextRow

i = 0 ' i and j dimension of the array
j = 0

Do While Not pRow Is Nothing
z = countfields - 2 '

Do While Not z = 0 '
NameArray(i, j) = pRow.value(pRow.Fields.FieldCount - z)
Debug.Print pRow.value(pRow.Fields.FieldCount - z)

z = z - 1
j = (countfields - 3) - z
Loop

Set pRow = pCursor.NextRow
i = i + 1
Loop

IsArrayAllocated(NameArray)
Debug.Print IsArrayAllocated(NameArray)

NumberOfArrayDimensions(NameArray)
Debug.Print NumberOfArrayDimensions(NameArray)

NumElements(NameArray)
Debug.Print NumElements(NameArray, 1)

```

```

Debug.Print NumElements(NameArray, 2)

ANOVAmod NameArray
GPStatistics
OneTypemeanststat
OneTypemeanststat2
DrawGraphmod

End Sub

' Function: ANOVAmod
' Author: M.Ferretti
' Date: May 2010
' Description: This function performs the the One-way Anova analysis
' http://www.ybcode.com/asp/showsn.asp?theID=867 (modified)

Function ANOVAmod(x() As Double) As Double

Dim VarTot As Double
Dim VarTread As Double
Dim VarE As Double
Dim QSTot As Double
Dim QStreat As Double
Dim QSE As Double
Dim dfTot As Integer
Dim dfTread As Integer
Dim dfE As Integer

' Einfaktorielle Varianzanalyse / Analysis of Variance

' Inputs:
' x()...2 dimensional array
' Outputs:
' VarTot...Variance total
' VarTreat...Treatmentvariance
' VarE...Errorvariance
' QSTot...Squaresum total
' QStreat...Squaresum treatment
' QSE Squaresum error
' dfTot...Degrees of Freedom total
' dfTreat...Degrees of Freedom treatment
' dfE...Degrees of Freedom error

Dim MwTot As Double
Dim SummenTest() As Double
Dim MweTest() As Double
Dim tmpQSE() As Double
Dim tmpSumTot As Double
Dim t As Integer
Dim N As Integer
Dim i As Integer, j As Integer

ReDim SummenTest(UBound(x, 2))
ReDim MweTest(UBound(x, 2))

N = UBound(x, 1) + 1
t = UBound(x, 2) + 1
For i = 0 To UBound(x, 2)
For j = 0 To UBound(x, 1)
SummenTest(i) = SummenTest(i) + x(j, i) ' Summen der Tests
Next j
MweTest(i) = SummenTest(i) / N
Next i

For i = 0 To UBound(SummenTest)
tmpSumTot = tmpSumTot + SummenTest(i)
Next i
MwTot = tmpSumTot / (N * t)
dfE = t * (N - 1)

ReDim tmpQSE(UBound(x, 2))
For i = 0 To UBound(x, 2)

```

```

' Mittelwert total
' Summen der Tests
' Mittelwerte der Tests
' Fehlerquadratsumme
' Summe total
' t...Anzahl Tests numero di gruppi o campioni o c
' n...Anzahl Werte numero di osservazioni (numero

' SummenTest(i) = SummenTest(i) + x(j, i) ' Summen der Tests
Next j
MweTest(i) = SummenTest(i) / N
Next i

For i = 0 To UBound(SummenTest)
tmpSumTot = tmpSumTot + SummenTest(i)
Next i
MwTot = tmpSumTot / (N * t)
dfE = t * (N - 1)

ReDim tmpQSE(UBound(x, 2))
For i = 0 To UBound(x, 2)

```

```

Public Function GetPval(ByVal F As Double, ByVal dfN As Double, ByVal dfD As Double, ByVal dFD As Double) As Double
'Input parameters:
' F = the F-value;
' dfN = the numerator degrees of freedom;
' dfD = the denominator degrees of freedom;

Dim PiD2 As Double
Dim x As Double
Dim th As Double
Dim A As Double
Dim sth As Double
Dim cth As Double
Dim res As Double

PiD2 = 3.14159265358979 / 2
x = dfD / (dfN * F + dfD)
th = Math.Atn(Math.Sqr(dfN * F / dfD))
A = th / PiD2
sth = Math.Sin(th)
cth = Math.Cos(th)

'PRIMA ERA
'If (IEERRemainder(dfN, 2)) = 0 Then
'ORA E'
If Math.Abs(IEERRemainder(dfN, 2)) = 0 Then
res = StatCom(1 - x, dfD, dfN + dfD - 4, dfD - 2) * ((x ^ (dfD / 2)))
'Debug.Print StatCom(1 - x, dfD, dfN + dfD - 4, dfD - 2)
ElseIf Math.Abs(IEERRemainder(dfD, 2)) = 0 Then
res = 1 - StatCom(x, dfN, dfN + dfD - 4, dfN - 2) * (1 - x ^ (dfN / 2))
End If

If dfD > 1 Then
A = A + sth * cth * StatCom(cth * cth, 2, dfD - 3, -1) / PiD2
'Debug.Print StatCom(cth * cth, 2, dfD - 3, -1)
End If

If dfN = 1 Then
res = 1 - A
End If

Dim c As Double
c = 4 * StatCom(sth * sth, dfD + 1, dfN + dfD - 4, dfD - 2) * sth * cth ^ (dfD) / 3.14159265
358979

If dfD = 1 Then
res = 1 - A + c / 2
End If

Dim k As Double
k = 2

While k <= (dfD - 1) / 2
c = c * k / (k - 0.5)
k = 1 + k
Wend

res = 1 - A + c
GetPval = res

Debug.Print "Probability Value (p): " & res

End Function

' Daniel Soper http://www.danielsoper.com/statcalc/default.aspx#c07

Public Function StatCom(ByVal q As Double, ByVal i As Double, ByVal j As Double, ByVal B As Double) As Double
As Double
'Utility function for computing p-values for the t and F-distributions

Dim zz As Double
Dim z As Double
Dim k As Integer

zz = 1
z = zz

```

```

For j = 0 To UBound(x, 1)
QSTot = QSTot + ((x(j, i) - MwTot) ^ 2) ' Quadratsumme total
tmpQSE(i) = tmpQSE(i) + ((x(j, i) - MwTest(i)) ^ 2)
Next j
QSTreat = QSTreat + ((MwTest(i) - MwTot) ^ 2)
QSE = QSE + tmpQSE(i) ' Fehlervarianz

Next i

VarE = QSE / dFE

QSTreat = QSTreat * N
dfTread = t - 1
VarTread = QSTreat / dfTread ' Treatment Varianz

dfTot = N * t - 1
VarTot = QSTot / dfTot

ANOVAmod = VarTread / VarE

Dim res As Double
res = GetPval(ANOVAmod, dfTread, dFE)
Debug.Print res

'Dim Fcritic As Double 'l'Fcritical non funziona
'Fcritic = GetFcritical(dfTread, dFE, res)
'Debug.Print Fcritic

ListBox1.AddItem dfTread
ListBox2.AddItem dFE
ListBox3.AddItem dfTot
ListBox4.AddItem QSTreat
ListBox5.AddItem QSE
ListBox6.AddItem QSTot
ListBox7.AddItem VarTread
ListBox8.AddItem VarE
ListBox9.AddItem VarTot
ListBox10.AddItem ANOVAmod
ListBox11.AddItem res

End Function

' Daniel Soper http://www.danielsoper.com/statcalc/default.aspx#c07

Public Function GetFcritical(ByVal df_Numerator As Double, ByVal df_Denominator As Double, ByVal p
As Double) As Double
Dim v As Double
Dim dv As Double
Dim F As Double

v = 0.5
dv = 0.5
F = 0#

While dv > 0.00000000000001
F = 1 / v - 1
dv = dv / 2
If GetPval(F, df_Numerator, df_Denominator) > p Then
v = v - dv
Else
v = v + dv
End If
Wend

Debug.Print "F = " & F

End Function

Public Function IEERRemainder(ByVal dividend As Double, ByVal divisor As Double) As Double
'http://msdn.microsoft.com/en-us/library/system.math.ieceerremainder.aspx
Dim A As Double
IEERRemainder = dividend - (divisor * Math.Round(dividend / divisor))
'Debug.Print "IEERRemainder = " & IEERRemainder

End Function

' Daniel Soper http://www.danielsoper.com/statcalc/default.aspx#c07

```

frmResponsefix - 5

```
k = i
While k <= j
  ZZ = ZZ * q * k / (k - B)
  Z = ZZ + Z
  k = 2 + k
Wend

'Debug.Print "z = " & z
StatCom = z
End Function

' Subroutine: GPStatistics
' Author: M.Ferretti
' Date: May 2010
' Description: This subroutine prepares the mean and standard deviation for each field (land use code) of Statfixanalysis
Sub GPStatistics() '(strIn, strOut, disField As String)
Dim GP As IGeoProcessor
Set GP = New GeoProcessor
GP.OverwriteOutput = True
Dim res As String
' Declare and set a variant array to hold the parameters
Dim parameters As IVariantArray
Set parameters = New VarArray
' Part 1: Get the feature class and its fields
Dim pMxDoc As IMXDocument
Dim pMap As IMap
Dim pFeatureLayer As IFeatureLayer
Dim pFeatureClass As IFeatureClass
Dim pFields As IFields
Dim count As Long
Set pMxDoc = ThisDocument
Set pMap = pMxDoc.FocusMap
Set pFeatureLayer = pMap.Layer(0)
Set pFeatureClass = pFeatureLayer.FeatureClass
Set pFields = pFeatureClass.Fields
' Get the number of fields
count = pFields.FieldCount
' Part 2: Prepare a list of fields and display the list.
Dim ii As Long
Dim aField As IField2
Dim fieldName As String
' Loop through each field, and add the field name to a list.
For ii = 2 To pFields.FieldCount - 1
  Set aField = pFields.Field(ii)
  res = aField.Name
Dim x As String
x = res & " MEAN; " & res & " STD"
' Now execute the Statistic tool
' Populate the variant array with the parameters
parameters.Add "c:\TATFolder2\fix\Statfixanalysis.shp" '(in_features)
parameters.Add "c:\TATFolder2\fix\singletables\" & res & ".dbf" '(in_features) '(out_feature_class)
parameters.Add x
GP.Execute "Statistics", parameters, Nothing
parameters.RemoveAll

```

frmResponsefix - 6

```
Next
Exit Sub
End Sub

' Subroutine: OneTypemeanstdstat and OneTypemeanstdstat2
' Author: M.Ferretti
' Date: May 2010
' Description: This subroutine products the table with mean and standard deviation of each land use code (tipology)
Sub OneTypemeanstdstat() ' Process the first file in the folder (without do/loop) and rename it to Tablemeanstd2
Const MyPath = "c:\TATFolder2\fix\singletables" ' Set the path.
Const FileType = "*.dbf"
Createmeanstdstat MyPath, FileType
End Sub

Sub Createmeanstdstat(strFolder As String, strFilePattern As String)
Dim strFileName As String
Dim strFolders() As String
Dim iFolderCount As Integer
Dim i As Integer
' Collect child folders
strFileName = dir$(strFolder & "\", vbDirectory)
Do Until strFileName = ""
  If (GetAttr(strFolder & "\" & strFileName) And vbDirectory) = vbDirectory Then
    If Left$(strFileName, 1) <> "." Then
      ReDim Preserve strFolders(iFolderCount)
      strFolders(iFolderCount) = strFolder & "\" & strFileName
      iFolderCount = iFolderCount + 1
    End If
  End If
Loop
' Process files in current folder
strFileName = dir$(strFolder & "\\" & strFilePattern)
'Do Until strFileName = ""
'Do things with files here*****
Dim Filex As String
Dim Filex2 As String
Filex = strFolder & "\" & strFileName
Filex2 = strFileName
Dim GP2 As IGeoProcessor
Set GP2 = New GeoProcessor
GP2.OverwriteOutput = True
' Declare and set a variant array to hold the parameters
Dim parameters2 As IVariantArray
Set parameters2 = New VarArray
' Populate the variant array with the parameters
parameters2.Add "c:\TATFolder2\fix\singletables\" & Filex2 '(in_features)
parameters2.Add "c:\TATFolder2\fix\Tablemeanstd2"
'Now execute the copy tool
GP2.Execute "copy", parameters2, Nothing
'*****
strFileName = dir$( )
'Loop
' Look through child folders
For i = 0 To iFolderCount - 1
  ProcessFileshr strFolders(i), strFilePattern
Next i
' Erase two filed, remain OID and FREQUENCY field
Dim filePath As String
filePath = "c:\TATFolder2\fix"

```

```

Dim tableName As String
tableName = "Tablemeanstd2.dbf"

Dim pTableWindow As ITableWindow
Dim pExistingTableWindow As ITableWindow
Dim pStandaloneTableCollection As IStandaloneTableCollection
Dim pTable As ITable
Dim pFields As IFields
Dim pField As IField2
Dim pField2 As IField2

Dim pWorkspace As IWorkspace
Dim pFact As IWorkspaceFactory
Dim pFact2 As IWorkspaceFactory
Set pFact = New shapefileWorkspaceFactory

Set pWorkspace = pFact.OpenFromFile(filePath, 0)

Set pWorkspace = pWorkspace
Set pTable = pWorkspace.OpenTable(tableName)
Set pFields = pTable.Fields
Set pField = pFields.Field(pFields.FieldCount - 1) 'std field

pTable.DeleteField pField

Set pField2 = pFields.Field(pFields.FieldCount - 1) 'mean field
pTable.DeleteField pField2

End Sub

Sub OneTypemeanstdstat2() ' Join every single dbf file with Tablemeanstd2, erase the "frequency" f
ield
Const MyPath = "C:\TATFolder2\fix\singletables" ' Set the path.
Const FileType = "*.dbf"
Createmeanstdstat2 MyPath, FileType

End Sub

Sub Createmeanstdstat2(strFolder As String, strFilePattern As String)
Dim strFileName As String
Dim strFolders() As String
Dim iFolderCount As Integer
Dim i As Integer

' Collect child folders
strFileName = dir$(strFolder & "\", vbDirectory)
Do Until strFileName = ""
If (GetAttr(strFolder & "\" & strFileName) And vbDirectory) = vbDirectory Then
If Left$(strFileName, 1) <> "." Then
ReDim Preserve strFolders(iFolderCount)
strFolders(iFolderCount) = strFolder & "\" & strFileName
iFolderCount = iFolderCount + 1
End If
End If
strFileName = dir$()

Loop

' Process files in current folder
strFileName = dir$(strFolder & "\" & strFilePattern)
Do Until strFileName = ""
'Do things with files here*****
Dim Filex As String
Dim Filex2 As String
Filex = strFolder & "\" & strFileName
Filex2 = strFileName

' Delete the frequency field

Dim filePath As String
filePath = "C:\TATFolder2\fix\singletables"
Dim tableName As String
tableName = Filex2

Dim pTableWindow As ITableWindow
Dim pExistingTableCollection As IStandaloneTableCollection

```

```

Dim pStandaloneTable As IStandaloneTable
Dim pTable As ITable
Dim pFields As IFields
Dim pField As IField2

Dim pWorkspace As IWorkspace
Dim pFact As IWorkspaceFactory
Dim pFact2 As IFeatureWorkspaceFactory
Set pFact = New shapefileWorkspaceFactory

Set pWorkspace = pFact.OpenFromFile(filePath, 0)

Set pWorkspace = pWorkspace
Set pTable = pWorkspace.OpenTable(tableName)
Set pFields = pTable.Fields
Set pField = pFields.Field(pFields.FieldCount - 1) 'FREQUENCY')

pTable.DeleteField pField

' Perform the join
Dim GP1 As IGeoProcessor
Set GP1 = New GeoProcessor

GP1.OverwriteOutput = True

' Declare and set a variant array to hold the parameters
Dim parameters As IVariantArray
Set parameters = New VarArray

' Populate the variant array with the parameters
parameters.Add "C:\TATFolder2\fix\Tablemeanstd2.dbf" '(in_features)
parameters.Add "OID" '(join_field)
parameters.Add "C:\TATFolder2\fix\singletables\" & Filex2 '(in_features) '(out_feature_class)
parameters.Add "OID" ',RMSIID FIRST;CHECK_NAME FIRST;STATUS FIRST"

'Now execute the join tool
GP1.Execute "joinfield", parameters, Nothing

parameters.RemoveAll

'*****
strFileName = dir$()
Loop

' Look through child folders
For i = 0 To iFolderCount - 1
ProcessFiles strFolders(i), strFilePattern
Next i

' Open the table

' Public Sub OpenTable()
' http://forums.esri.com/Thread.asp?c=93&f=992&t=105591&mc=3#msgid299132
Dim filePath2 As String
filePath2 = "C:\TATFolder2\fix"
Dim tableName2 As String
tableName2 = "Tablemeanstd2.dbf"

Dim pMxDoc As IMXDocument
Set pMxDoc = ThisDocument

Dim pWorkspace2 As IWorkspace
Dim pFact2 As IWorkspaceFactory
Set pFact2 = New shapefileWorkspaceFactory

Set pWorkspace2 = pFact2.OpenFromFile(filePath2, 0)
Dim pWorkspace2 As IFeatureWorkspace
Set pWorkspace2 = pWorkspace2

Dim pTable1 As ITable
Set pTable1 = pWorkspace2.OpenTable(tableName2)

Dim pStTab As IStandaloneTable
Set pStTab = New StandaloneTable
Set pStTab.table = pTable1
Dim pStTabColl As IStandaloneTableCollection
Set pStTabColl = pMxDoc.FocusMap

```


frmResponsefix - 10

```
Next
' Update data graph.

Dim pCancelTracker As ITrackCancel
Set pCancelTracker = New CancelTracker
pDataGraph1.Update pCancelTracker

' Create data graph window within ArcMap
Dim pDGWin As IDataGraphWindow2
Set pDGWin = New DataGraphWindow
Set pDGWin.DataGraphBase = pDataGraphBase
Set pDGWin.Application = ThisDocument.Parent
pDGWin.Show (True)

Dim pDataGraphs As IDataGraphCollection
Set pDataGraphs = pMxDoc
pDataGraphs.AddDataGraph pDataGraphBase

End Sub

Private Sub Exitbutton_Click()

ListBox1.Clear
ListBox2.Clear
ListBox3.Clear
ListBox4.Clear
ListBox5.Clear
ListBox6.Clear
ListBox7.Clear
ListBox8.Clear
ListBox9.Clear
ListBox10.Clear

Kill "C:\TATfolder2\fix\singlfeatures\*"
Rmdir "C:\TATfolder2\fix\singlfeatures"
Kill "C:\TATfolder2\fix\split\*"
Rmdir "C:\TATfolder2\fix\split"
Kill "C:\TATfolder2\fix\singletables\*"
Rmdir "C:\TATfolder2\fix\singletables"
Kill "C:\TATfolder2\fix\*"
Rmdir "C:\TATfolder2\fix"
Kill "C:\TATfolder2\*"
Rmdir "C:\TATfolder2"

Dim GP As IGeoProcessor
Set GP = New GeoProcessor

GP.OverwriteOutput = True

' Declare and set a variant array to hold the parameters
Dim parameters As IVariantArray
Set parameters = New VarArray

' Populate the variant array with the parameters
parameters.Add "Statfixanalysis"

' Now execute the delete tool
GP.Execute "delete_management", parameters, Nothing

Dim GP2 As IGeoProcessor
Set GP2 = New GeoProcessor

GP2.OverwriteOutput = True

' Declare and set a variant array to hold the parameters
Dim parameters2 As IVariantArray
Set parameters2 = New VarArray

' Populate the variant array with the parameters
parameters2.Add "Fixanalysis"

'Now execute the Dissolve tool
GP2.Execute "delete_management", parameters2, Nothing

End
```

frmResponsefix - 9

```
pStTabColl.AddStandaaloneTable pStTab

' Refresh the TOC
pMxDoc.UpdateContents

Dim pTableWindow1 As ITableWindow
Set pTableWindow1 = New TableWindow

Set pTableWindow1.Application = Application
Set pTableWindow1.table = pTable1

pTableWindow1.Show True

End Sub

' Subroutine: DrawGraphmod
' Author: M.Ferretti
' Date: May 2010
' Description: This subroutine provides to draw a graph box-plot type
' http://resources.esri.com/help/9.3/ArcGISDesktop/dotnet/60f8f525-ade9-4f79-9e32-0835f736f30c.htm

Public Sub DrawGraphmod()

Dim pMxDoc As IMxDocument
Set pMxDoc = ThisDocument

' Create graph
Dim pDataGraphBase As IDataGraphBase
Dim pDataGraph As IDataGraph
Set pDataGraphBase = New dataGraphT
Set pDataGraphT = pDataGraphBase

' Graph and legend titles
pDataGraphT.GeneralProperties.Title = ""
pDataGraphT.LegendProperties.Title = "Land Use code"
pDataGraphBase.Name = "Land use location of the animal fixes in respect to the land use incidence i
n the Home range"
pDataGraphT.AxisProperties(2).Visible = False

' Insert the loop for each field of the table

Dim pMap As IMap
Dim pFeatureLayer As IFeatureLayer
Dim pFeatureClass As IFeatureClass
Dim pFields As IFields
Dim count As Long
Set pMap = pMxDoc.FocusMap
Set pFeatureLayer = pMap.Layer(0)
Set pFeatureClass = pFeatureLayer.FeatureClass
Set pFields = pFeatureClass.Fields

' Get the number of fields
count = pFields.FieldCount

' Prepare a list of fields and display the list.
Dim ii As Long
Dim aField As IField2
Dim fieldName As String

' Loop through each field, and add the field name to a list
For ii = 2 To pFields.FieldCount - 1
Set aField = pFields.Field(ii)
res = aField.Name

' Create box plot series
Dim pSP As ISeriesProperties
Set pSP = pDataGraphT.AddSeries("box_plot")
pSP.ColorType = esriGraphColorMatch
pSP.InLegend = True 'nascondo legenda
pSP.Name = res ' Each box-plot have the name of the origin field

Dim pLayer As ILayer
Set pLayer = pMxDoc.FocusMap.Layer(0)
pSP.SourceData = pLayer
pSP.SetField 0, res
```

```
frmResponsefix - 1
' In this form the tool products the One-way Anova analysis file, a table with mean and standard de
viation and a graph (HR process)
' Subroutine: UserForm_Initialize
' Author: M.Ferretti
' Date: May 2010
' Description: This subroutine prepares the two dimensional array for the One-way Anova analysis
Private Sub UserForm_Initialize()
Dim pMxDoc As IMxDocument
Dim pMap As IMap
Dim pFeatureLayer As IFeatureLayer
Dim pFeatClass As IFeatureClass
Dim pRow As IRow
Set pMxDoc = ThisDocument
Set pMap = pMxDoc.FocusMap
' Define the input feature class
Set pFeatureLayer = pMap.Layer(0)
Set pFeatClass = pFeatureLayer.FeatureClass
' Count the record number
Dim pLayerDef As IFeatureLayerDefinition2
Set pLayerDef = pFeatureLayer
Dim pQFilter As IQeryFilter
Set pQFilter = New QueryFilter
pQFilter.WhereClause = pLayerDef.DefinitionExpression
Dim count As Long
count = pFeatureLayer.FeatureClass.FeatureCount(pQFilter)
countfields = pFeatureLayer.FeatureClass.Fields.FieldCount
' Set the array
Dim pCursor As ICursor
Dim NameArray() As Double
ReDim NameArray(count - 1, countfields - 3) As Double
'Add a the Array two field values
Set pCursor = pFeatClass.Update(Nothing, True)
Set pRow = pCursor.NextRow
i = 0 ' i and j dimension of the array
j = 0
Do While Not pRow Is Nothing
z = countfields - 2 '
Do While Not z = 0 '
NameArray(i, j) = pRow.value(pRow.Fields.FieldCount - z)
Debug.Print pRow.value(pRow.Fields.FieldCount - z)
z = z - 1
j = (countfields - 3) - z ' il primo valore di j nell'array è 12-3-(10-1)=0
Loop
Set pRow = pCursor.NextRow
i = i + 1
Loop
IsArrayAllocated (NameArray)
Debug.Print IsArrayAllocated (NameArray)
NumberOfArrayDimensions (NameArray)
Debug.Print NumberOfArrayDimensions (NameArray)
NumElements (NameArray)
Debug.Print NumElements (NameArray, 1)
Debug.Print NumElements (NameArray, 2)
```

```

frmResponsehr - 3
Next j
QSTreat = QSTreat + ((MweTest(i) - MwTot) ^ 2)
QSE = QSE + tmpQSE(i)
Next i
VarE = QSE / dfE
QSTreat = QSTreat * N
dfTreat = t - 1
VarTreat = QSTreat / dfTreat
dfTot = N * t - 1
VarTot = QSTot / dfTot
ANOVAmod = VarTreat / VarE
Dim res As Double
res = GetPval(ANOVAmod, dfTreat, dfE)
Debug.Print res
'Dim Fcritic As Double 'Fcritical non funziona
'Fcritic = GetFcritical(dfTreat, dfE, res)
Debug.Print Fcritic
ListBox1.AddItem dfTreat
ListBox2.AddItem dfE
ListBox3.AddItem dfTot
ListBox4.AddItem QSTreat
ListBox5.AddItem QSE
ListBox6.AddItem QSTot
ListBox7.AddItem VarTreat
ListBox8.AddItem VarE
ListBox9.AddItem VarTot
ListBox10.AddItem ANOVAmod
ListBox11.AddItem res
End Function
' Daniel Soper http://www.danielsoper.com/statcalc/default.aspx#c07
Public Function GetFcritical(ByVal df_Numerator As Double, ByVal df_Denominator As Double, ByVal df_Df As Double) As Double
As Double) As Double
Dim v As Double
Dim dv As Double
Dim F As Double
v = 0.5
dv = 0.5
F = 0#
While dv > 0.000000000001
F = 1 / v - 1
dv = dv / 2
If GetPval(F, df_Numerator, df_Denominator) > p Then
v = v - dv
Else
v = v + dv
End If
Wend
Debug.Print "F = " & F
End Function
Public Function IEERRemainder(ByVal dividend As Double, ByVal divisor As Double) As Double
Dim A As Double
IEERRemainder = dividend - (divisor * Math.Round(dividend / divisor))
Debug.Print "IEERRemainder = " & IEERRemainder
End Function
' Daniel Soper http://www.danielsoper.com/statcalc/default.aspx#c07
Public Function GetPval(ByVal F As Double, ByVal dfN As Double, ByVal dfD As Double) As Double

```

```

frmResponsehr - 2
ANOVAmod NameArray
GPstatistics
OneTypemeanststat
OneTypemeanststat2
DrawGraphmod
End Sub
' Function: ANOVAmod
' Author: M.Ferretti
' Date: May 2010
' Description: This function performs the One-way Anova analysis
' http://www.vbcode.com/asp/showsn.asp?theID=867 (modified)
Function ANOVAmod(x() As Double) As Double
Dim VarTot As Double
Dim VarTreat As Double
Dim VarE As Double
Dim QSTot As Double
Dim QSTreat As Double
Dim QSE As Double
Dim dfTot As Integer
Dim dfTreat As Integer
Dim dfE As Integer
' Einfaktorielle Varianzanalyse / Analysis of Variance
' Inputs:
' x()...2 dimensional array
' Outputs:
' VarTot...Variance total
' VarTreat...Treatmentvariance
' VarE...Errorvariance
' QSTot...Squaresum total
' QSTreat...Squaresum treatment
' QSE Squaresum error
' dtTot...Degrees of Freedom total
' dfTreat...Degrees of Freedom treatment
' dfE...Degrees of Freedom error
' Mittelwert total
' Summen der Tests
' Mittelwerte der Tests
' Fehlerquadratsumme
' Summe total
' t... Anzahl Tests numero di gruppi o campioni o c
' n...Anzahl Werte numero di osservazioni (numero
di animali)
Dim i As Integer, j As Integer
ReDim SummenTest(UBound(x, 2))
ReDim MweTest(UBound(x, 2))
N = UBound(x, 1) + 1
t = UBound(x, 2) + 1
For i = 0 To UBound(x, 1)
For j = 0 To UBound(x, 2)
SummenTest(i) = SummenTest(i) + x(j, i) ' Summen der Tests
Next j
MweTest(i) = SummenTest(i) / N
Next i
For i = 0 To UBound(SummenTest)
tmpSumTot = tmpSumTot + SummenTest(i) ' Summe total
Next i
MwTot = tmpSumTot / (N * t) ' Mittelwert total
dfE = t * (N - 1)
ReDim tmpQSE(UBound(x, 2))
For i = 0 To UBound(x, 2)
For j = 0 To UBound(x, 1)
QSTot = QSTot + ((x(j, i) - MwTot) ^ 2) ' Quadratsumme total
tmpQSE(i) = tmpQSE(i) + ((x(j, i) - MweTest(i)) ^ 2)

```

```

' Input parameters:
' F = the F-value;
' dfN = the numerator degrees of freedom;
' dfD = the denominator degrees of freedom;

Dim pID2 As Double
Dim x As Double
Dim th As Double
Dim A As Double
Dim sth As Double
Dim cth As Double
Dim res As Double

pID2 = 3.14159265358979 / 2
x = dfD / (dfN * F + dfD)
th = Math.AtN(Math.Sqrt(dfN * F / dfD))
A = th / pID2
sth = Math.Sin(th)
cth = Math.Cos(th)

If Math.Abs(IEEERemainder(dfN, 2)) = 0 Then
    res = StatCom(1 - x, dfD, dfN + dfD - 4, dfD - 2) * ((x ^ (dfD / 2)))
' Debug.Print StatCom(1 - x, dfD, dfN + dfD - 4, dfD - 2)
ElseIf Math.Abs(IEEERemainder(dfD, 2)) = 0 Then
    res = 1 - StatCom(x, dfN, dfN + dfD - 4, dfN - 2) * (1 - x ^ (dfN / 2))
End If

If dfD > 1 Then
    A = A + sth * cth * StatCom(cth * cth, 2, dfD - 3, -1) / pID2
' Debug.Print StatCom(cth * cth, 2, dfD - 3, -1)
End If

If dfN = 1 Then
    res = 1 - A
End If

Dim c As Double
c = 4 * StatCom(sth * sth, dfD + 1, dfN + dfD - 4, dfD - 2) * sth * cth ^ (dfD) / 3.14159265

358979

If dfD = 1 Then
    res = 1 - A + c / 2
End If

Dim k As Double
k = 2

While k <= (dfD - 1) / 2
    c = c * k / (k - 0.5)
    k = 1 + k
Wend

res = 1 - A + c

GetPval = res

Debug.Print "Probability Value (p): = " & res

End Function

' Daniel Soper http://www.danielsoper.com/statcalc/default.aspx#t07

Public Function StatCom(ByVal q As Double, ByVal i As Double, ByVal j As Double, ByVal B As Double)
    As Double
' Utility function for computing p-values for the t and F-distributions

Dim zz As Double
Dim z As Double
Dim k As Integer

zz = 1
z = zz
k = i

While k <= j
    zz = zz * q * k / (k - B)

```

```

z = zz + z
k = 2 + k
Wend

' Debug.Print "z = " & z
StatCom = z

End Function

' Subroutine: GPstatistics
' Author: M.Ferretti
' Date: May 2010
' Description: This subroutine prepares the mean and standard deviation for each field (land use co
de) of Stathranalysis

Sub GPstatistics() '(strIn, strOut, disField As String)

Dim GP As IGeoProcessor
Set GP = New GeoProcessor

GP.OverrideOutput = True

Dim res As String

' Declare and set a variant array to hold the parameters
Dim parameters As IVariantArray
Set parameters = New VarArray

' Part 1: Get the feature class and its fields
Dim pMxDoc As IMxDocument
Dim pMap As IMap
Dim pFeatureLayer As IFeatureLayer
Dim pFeatureClass As IFeatureClass
Dim pFields As IFields
Dim count As Long
Set pMxDoc = ThisDocument
Set pMap = pMxDoc.FocusMap
Set pFeatureLayer = pMap.Layer(0)
Set pFeatureClass = pFeatureLayer.FeatureClass
Set pFields = pFeatureClass.Fields

' Get the number of fields
count = pFields.FieldCount

' Part 2: Prepare a list of fields and display the list
Dim ii As Long
Dim aField As IField2
Dim fieldName As String

' Loop through each field, and add the field name to a list
For ii = 2 To pFields.FieldCount - 1
    Set aField = pFields.Field(ii)
    res = aField.Name

Dim x As String
x = res & " MEAN; " & res & " STD"

' Now execute the Statistic tool

' Populate the variant array with the parameters
parameters.Add "C:\TATFolder2\hr\Stathranalysis.shp" '(in_features)
parameters.Add "C:\TATFolder2\hr\singletables\" & res & ".dbf" '(in_features) '(out_feature_cl
ass)
parameters.Add x
GP.Execute "Statistics", parameters, Nothing
parameters.RemoveAll
Next
Exit Sub

End Sub

```

```

' Subroutine: OneTypeMeanstDstat and OneTypeMeanstDstat2
' Author: M.Ferretti
' Date: May 2010
' Description: This subroutine products the table with mean and standard deviation of each land use
code (tipology)

Sub OneTypeMeanstDstat() ' Process the first file in the folder (without do/loop) and rename it to
TableMeanstD1
    Const MyPath = "C:\TATFolder2\hr\singletables" ' Set the path.
    Const FileType = "*.dbf"
    CreatemeanstDstat MyPath, FileType
End Sub

Sub CreatemeanstDstat(strFolder As String, strFilePattern As String)

Dim strFileName As String
Dim strFolders() As String
Dim iFolderCount As Integer
Dim i As Integer

' Collect child folders
strFileName = dir$(strFolder & "\", vbDirectory)
Do Until strFileName = ""
    If (GetAttr(strFolder & "\\" & strFileName) And vbDirectory) = vbDirectory Then
        If Left$(strFileName, 1) <> "." Then
            ReDim Preserve strFolders(iFolderCount)
            strFolders(iFolderCount) = strFolder & "\" & strFileName
            iFolderCount = iFolderCount + 1
        End If
    End If
    strFileName = dir$()
Loop

' Process files in current folder
strFileName = dir$(strFolder & "\\" & strFilePattern)
'Do Until strFileName = ""
'Do things with files here*****

Dim Filex As String
Dim Filex2 As String
Filex = strFolder & "\" & strFileName
Filex2 = strFileName

Dim GP2 As IGeoProcessor
Set GP2 = New GeoProcessor

GP2.OverwriteOutput = True

' Declare and set a variant array to hold the parameters
Dim parameters2 As IVariantArray
Set parameters2 = New VariantArray

' Populate the variant array with the parameters
parameters2.Add "c:\TATFolder2\hr\singletables" & Filex2 '(in_features)
parameters2.Add "c:\TATFolder2\hr\tablemeanstD1"

'Now execute the copyDissolve tool
GP2.Execute "copy", parameters2, Nothing

'*****
strFileName = dir$()
'Loop

'Look through child folders
For i = 0 To iFolderCount - 1
    ProcessFileshr strFolders(i), strFilePattern
Next i

' Erase two field, remain OID and FREQUENCY field
Dim filePath As String
filePath = "C:\TATFolder2\hr"
Dim tableName As String
tableName = "tablemeanstD1.dbf"

Dim pTableWindow As ITableWindow
Dim pExistingTableWindow As ITableWindow

```

```

Dim pStandAloneTableCollection As IStandAloneTableCollection
Dim pStandAloneTable As IStandAloneTable
Dim pTable As ITable
Dim pFields As IFields
Dim pField As IField2
Dim pField2 As IField2

Dim pWorkspace As IWorkspace
Dim pFact As IWorkspaceFactory
Dim pFact2 As IWorkspaceFactory
Set pFact = New shapefileWorkspaceFactory

Set pWorkspace = pFact.OpenFromFile(filePath, 0)

Set pWorkspace = pWorkspace
Set pTable = pWorkspace.OpenTable(tableName)
Set pFields = pTable.Fields
Set pField = pFields.Field(pFields.FieldCount - 1) 'std field
pTable.DeleteField pField

Set pField2 = pFields.Field(pFields.FieldCount - 1) 'mean field
pTable.DeleteField pField2

End Sub

Sub OneTypeMeanstDstat2() ' Join every single dbf file with TableMeanstD1, erase the "frequency" f
ield
    Const MyPath = "C:\TATFolder2\hr\singletables" ' Set the path.
    Const FileType = "*.dbf"
    CreatemeanstDstat2 MyPath, FileType
End Sub

Sub CreatemeanstDstat2(strFolder As String, strFilePattern As String)

Dim strFileName As String
Dim strFolders() As String
Dim iFolderCount As Integer
Dim i As Integer

' Collect child folders
strFileName = dir$(strFolder & "\", vbDirectory)
Do Until strFileName = ""
    If (GetAttr(strFolder & "\\" & strFileName) And vbDirectory) = vbDirectory Then
        If Left$(strFileName, 1) <> "." Then
            ReDim Preserve strFolders(iFolderCount)
            strFolders(iFolderCount) = strFolder & "\" & strFileName
            iFolderCount = iFolderCount + 1
        End If
    End If
    strFileName = dir$()
Loop

' Process files in current folder
strFileName = dir$(strFolder & "\\" & strFilePattern)
Do Until strFileName = ""
    ' Do things with files here*****

Dim Filex As String
Dim Filex2 As String
Filex = strFolder & "\" & strFileName
Filex2 = strFileName

' Delete the frequency field

Dim filePath As String
filePath = "C:\TATFolder2\hr\singletables"
Dim tableName As String
tableName = Filex2

Dim pTableWindow As ITableWindow
Dim pExistingTableWindow As ITableWindow
Dim pStandAloneTableCollection As IStandAloneTableCollection
Dim pTable As ITable
Dim pFields As IFields
Dim pField As IField2

```

```

Dim pWorkspace As IWorkspace
Dim pFact As IWorkspaceFactory
Dim pFWorkspace As IFeatureWorkspace
Set pFact = New shapefileWorkspaceFactory

Set pWorkspace = pFact.OpenFromFile(filePath, 0)

Set pFWorkspace = pWorkspace
Set pTable = pFWorkspace.OpenTable(tableName)
Set pFields = pTable.Fields
Set pField = pFields.Field(pFields.FindField("FREQUENCY"))

pTable.DeleteField pField

' Perform the join
Dim GP1 As IGeoProcessor
Set GP1 = New GeoProcessor

GP1.OverwriteOutput = True

' Declare and set a variant array to hold the parameters
Dim parameters1 As IVariantArray
Set parameters1 = New VarArray

' Populate the variant array with the parameters
parameters1.Add "C:\ATFolder2\hr\tablemeanstd1.dbf" ' (in_features)
parameters1.Add "OID" ' (join field)
parameters1.Add "C:\ATFolder2\hr\singletables" & Filex2 ' (in_features) ' (out_feature_class)
parameters1.Add "OID" ' ,RMSIID FIRST,CHECK_NAME FIRST,STATUS FIRST"

' Now execute the join tool
GP1.Execute "joinfield", parameters1, Nothing

parameters1.RemoveAll
' *****
strFileName = dir$( )
Loop

' Look through child folders
For i = 0 To iFolderCount - 1
    ProcessFileshr strFolders(i), strFilePattern
Next i

' Open the table

' Public Sub OpenTable()
'     http://forums.esri.com/Thread.asp?c=93&f=992&t=105591&mc=3#msgid299132
Dim filePath2 As String
filePath2 = "C:\ATFolder2\hr"
Dim tableName2 As String
tableName2 = "tablemeanstd1.dbf"

Dim pMxDoc As IMXDocument
Set pMxDoc = ThisDocument

Dim pWorkspace2 As IWorkspace
Dim pFact2 As IWorkspaceFactory
Set pFact2 = New shapefileWorkspaceFactory

Set pWorkspace2 = pFact2.OpenFromFile(filePath2, 0)
Dim pFWorkspace2 As IFeatureWorkspace
Set pFWorkspace2 = pWorkspace2

Dim pTable1 As ITable
Set pTable1 = pFWorkspace2.OpenTable(tableName2)

Dim pStTab As IStandAloneTable
Set pStTab = New StandAloneTable
Dim pStTabColl As IStandAloneTableCollection
Set pStTabColl = pMxDoc.FocusMap
pStTabColl.AddStandAloneTable pStTab

```

```

' Refresh the TOC
pMxDoc.UpdateContents

Dim pTableWindow1 As ITableWindow
Set pTableWindow1 = New TableWindow

Set pTableWindow1.Application = Application
Set pTableWindow1.table = pTable1

pTableWindow1.Show True

End Sub

' Subroutine: DrawGraphmod
' Author: M.Ferretti
' Date: May 2010
' Description: This subroutine provides to draw a graph box-plot type
' http://resources.esri.com/help/9.3/ArcGISDesktop/dotnet/60f8f525-ade9-4f79-9e32-0835f736f30c.htm

Public Sub DrawGraphmod()

Dim pMxDoc As IMXDocument
Set pMxDoc = ThisDocument

' Create graph
Dim pDataGraphBase As IDataGraphBase
Dim pDataGraph As IDataGraph
Set pDataGraphBase = New dataGraph
Set pDataGraph = pDataGraphBase

' Graph and legend titles
pDataGraph.GeneralProperties.Title = ""
pDataGraph.LegendProperties.Title = "Land Use code"
pDataGraphBase.Name = "Land use partition in the animal home range in respect to the overall land use partition"

pDataGraph.AxisProperties(2).Visible = False

' Insert the loop for each field of the table
Dim pMap As IMap
Dim pFeatureLayer As IFeatureLayer
Dim pFeatureClass As IFeatureClass
Dim pFields As IFields

Dim count As Long
Set pMap = pMxDoc.FocusMap
Set pFeatureLayer = pMap.Layer(0)
Set pFeatureClass = pFeatureLayer.FeatureClass
Set pFields = pFeatureClass.Fields

' Get the number of fields
count = pFields.FieldCount

' Prepare a list of fields and display the list
Dim ii As Long
Dim aField As IField2
Dim fieldName As String

' Loop through each field, and add the field name to a list
For ii = 2 To pFields.FieldCount - 1
    Set aField = pFields.Field(ii)
    res = aField.Name

' Create box plot series
Dim pSP As ISeriesProperties
Set pSP = pDataGraph.AddSeries("box_plot")
pSP.ColorType = esriGraphColorMatch
pSP.InLegend = True 'nascondo legenda
pSP.Name = res ' Each box-plot have the name of the origin field

Dim pLayer As ILayer
Set pLayer = pMxDoc.FocusMap.Layer(0)
pSP.SourceData = pLayer
pSP.SetField 0, res

Next

' Update data graph

```

```

frmResponsehr - 10
Dim pCancelTracker As ITrackCancel
Set pCancelTracker = New CancelTracker
pDataGraphT.Update pCancelTracker

' Create data graph window within ArcMap
Dim pdGWin As IDataGraphWindow2
Set pdGWin = New DataGraphWindow
Set pdGWin.DataGraphBase = pDataGraphBase
Set pdGWin.Application = ThisDocument.Parent
pdGWin.Show (True)

Dim pDataGraphs As IDataGraphCollection
Set pDataGraphs = pMxDoc
pDataGraphs.AddDataGraph pDataGraphBase

End Sub

Private Sub Exitbutton_Click()
ListBox1.Clear
ListBox2.Clear
ListBox3.Clear
ListBox4.Clear
ListBox5.Clear
ListBox6.Clear
ListBox7.Clear
ListBox8.Clear
ListBox9.Clear
ListBox10.Clear
ListBox11.Clear

Kill "C:\TATFolder2\hr\singlefeatures\*"
Rmdir "C:\TATFolder2\hr\singlefeatures"
Kill "C:\TATFolder2\hr\split\*"
Rmdir "C:\TATFolder2\hr\split"
Kill "C:\TATFolder2\hr\singleables\*"
Rmdir "C:\TATFolder2\hr\singleables"
Kill "C:\TATFolder2\hr\*"
Rmdir "C:\TATFolder2\hr"
Kill "C:\TATFolder2\*"
Rmdir "C:\TATFolder2"

' GP append, da dissolve modificato
Dim GP As IGeoProcessor
Set GP = New GeoProcessor

GP.OverwriteOutput = True

' Declare and set a variant array to hold the parameters
Dim parameters As IVariantArray
Set parameters = New VarArray

' Populate the variant array with the parameters
parameters.Add "Stathranalysis"

' Now execute the Dissolve tool
GP.Execute "delete_management", parameters, Nothing

Dim GP2 As IGeoProcessor
Set GP2 = New GeoProcessor

GP2.OverwriteOutput = True

' Declare and set a variant array to hold the parameters
Dim parameters2 As IVariantArray
Set parameters2 = New VarArray

' Populate the variant array with the parameters
parameters2.Add "Hranalysis"

' Now execute the delete tool
GP2.Execute "delete_management", parameters2, Nothing

End

```


frmStudy_area_layer - 1

```
' In this form the User chooses the study area shape file (HR and FIX matrix creation process)
' Subroutine: UserForm_Initialize
' Author: M.Ferretti
' Date: May 2010
' Description: This subroutine, and the following function, populates the combobox with the layers
present in the TOC
' http://forums.esri.com/Thread.asp?c=93&f=992&t=212356&mc=2&msgid=640862 (modified)

Public Sub UserForm_Initialize()
Dim pMxDoc As IMxDocument
Dim pID As New UID
Dim pEnumLyr As IEnumLayer
Dim pLyr As ILayer

' Set the UID for GeoFeatureLayers
pID.Value = "[E156D7B5-22AF-11D3-9F99-00C04F6BC78E]"

' Get an enumeration on ALL the feature layers (TRUE = iterate through group layers)
Set pMxDoc = ThisDocument
Set pEnumLyr = pMxDoc.FocusMap.Layers(pID, True)

' Loop thru the returned layers and add the names to the combo box.
pEnumLyr.Reset
While Not pLyr Is Nothing
    Set pLyr = pEnumLyr.Next
    ComboBox1.AddItem pLyr.Name
    Set pLyr = pEnumLyr.Next
Wend

End Sub
```

```
Public Function FindLayerByName(LayerName As String) As IFeatureLayer 'collegata a rendere disponi
bile il file selezionato nel combobox
Dim pEnumLayer As IEnumLayer
Dim pCompositeLayer As ICompositeLayer
Dim i As Integer
Dim pMxDoc As IMxDocument
Dim pMap As IMap

Set pMxDoc = Application.Document
Set pMap = pMxDoc.FocusMap

Set pEnumLayer = pMap.Layers
pEnumLayer.Reset

Dim pLayer As ILayer
Set pLayer = pEnumLayer.Next

Do While Not pLayer Is Nothing
    If TypeOf pLayer Is ICompositeLayer Then
        Set pCompositeLayer = pLayer
        For i = 0 To pCompositeLayer.count - 1
            With pCompositeLayer
                If .Layer(i).Name = LayerName Then
                    If TypeOf .Layer(i) Is IFeatureLayer Then
                        Set FindLayerByName = pCompositeLayer.Layer(i)
                    Exit Function
                End If
            End If
        End With
    ElseIf pLayer.Name = LayerName And TypeOf pLayer Is IFeatureLayer Then
        Set FindLayerByName = pLayer
    Exit Function
End If
Set pLayer = pEnumLayer.Next
Loop

End Function

' Subroutine: cmdRun_Click
' Author: M.Ferretti
' Date: May 2010
' Description: This subroutine copies the selected study area file from the User, to preserve the o
riginal User's file
```

frmStudy_area_layer - 2

```
' AddDeleteField and CalculateField from Programming ArcObjects with VBA: a task-oriented approach
- Kang-Tsung Chang (modified)

Private Sub cmdRun_Click()
Dim pFLayer As IFeatureLayer
Set pFLayer = FindLayerByName (frmStudy_area_layer.ComboBox1.Text)

' GP Librari (modified)
Dim GP As IGeoProcessor
Set GP = New GeoProcessor

GP.OverwriteOutput = True

' Declare and set a variant array to hold the parameters
Dim parameters As IVariantArray
Set parameters = New VarArray

' Populate the variant array with the parameters
parameters.Add pFLayer '(in features)
parameters.Add "c:\TATfolder\orfilesmod\1study_areamod.shp"

'Now execute the Copy tool
GP.Execute "copyfeatures", parameters, Nothing

frmStudy_area_layer.Hide ' chiuso il combobox

' Dissolve the study area (more polygons become a unique polygon)
' Uses new GeoProcessor library to perform dissolve
' strIn, strOut = full path strings for FC or shp input and output
' disField = field in input table to dissolve

Dim GP1 As IGeoProcessor
Set GP1 = New GeoProcessor

GP1.OverwriteOutput = True

' Declare and set a variant array to hold the parameters
Dim parameters1 As IVariantArray
Set parameters1 = New VarArray

' Populate the variant array with the parameters
parameters1.Add "c:\TATfolder\orfilesmod\1study_areamod.shp" '(in features)
parameters1.Add "c:\TATfolder\orfilesmod\study_areamod.shp" '(in_features) '(out_feature_class)

' Now execute the Dissolve tool
GP1.Execute "Dissolve_management", parameters1, Nothing

' Eliminate 1study_areamod.shp from the TOC

Dim pMxDoc As IMxDocument
Dim pMap As IMap
Dim pFeatureLayer As IFeatureLayer
Dim pFeatureClass As IFeatureClass
Dim pDataset As IDataset
Dim pActiveView As IActiveView
Set pMxDoc = ThisDocument
Set pMap = pMxDoc.FocusMap

' Define the dataset to be deleted
Set pFeatureLayer = pMap.Layer(1)
Set pFeatureClass = pFeatureLayer.FeatureClass

' Remove the layer from the active map
pMap.DeleteLayer pFeatureLayer

' Delete the dataset
Set pDataset = pFeatureClass
pDataset.Delete

' Refresh the map
Set pActiveView = pMap
pActiveView.Refresh

End Sub
```



```

Private Sub cmdCancel_Click()
    ' End the Appennino Tool
    Loop
    'Look through child folders
    For i = 0 To iFolderCount - 1
        ProcessFilesdelete strFolders(i), strFilePattern
    Next i
End Sub

```

```

Private Sub cmdCancel_Click()
    ' End the Appennino Tool
    Const MyPath = "C:\TATFolder\orfilesmod\" ' Set the path.
    Const FileType = "*.shp"
    ProcessFilesdelete MyPath, FileType
    Kill "C:\TATFolder\orfilesmod\*"
    Rmdir "C:\TATFolder\orfilesmod"
    Rmdir "C:\TATFolder\intersectfiles"
    Rmdir "C:\TATFolder\hr"
    Rmdir "C:\TATFolder\fix"
    Rmdir "C:\TATFolder"
End
End Sub
Sub ProcessFilesdelete(strFolder As String, strFilePattern As String)
    ' Delete the create files
    ' http://www.vbaexpress.com/kb/getarticle.php?kb_id=245 (modified)
    Dim strFileName As String
    Dim strFolders() As String
    Dim iFolderCount As Integer
    Dim i As Integer
    'Collect child folders
    strFileName = dir$(strFolder & "\", vbDirectory)
    Do Until strFileName = ""
        If (GetAttr(strFolder & "\\" & strFileName) And vbDirectory) = vbDirectory Then
            If Left$(strFileName, 1) <> "." Then
                ReDim Preserve strFolders(iFolderCount)
                strFolders(iFolderCount) = strFolder & "\" & strFileName
                iFolderCount = iFolderCount + 1
            End If
        End If
    Loop
    strFileName = dir$()
    'process files in current folder
    strFileName = dir$(strFolder & "\" & strFilePattern)
    Do Until strFileName = ""
        'Do things with files here*****
        Dim Filex As String
        Dim Filex2 As String
        Dim Filex3 As String
        Dim Filex4 As String
        Filex = strFolder & "\" & strFileName
        Filex2 = strFileName
        Filex3 = Len(Filex2) ' Calculate how is long and eliminate 4 character ".shp"
        Filex4 = Left(Filex2, Filex3 - 4)
        'GP Library (modified)
        Dim GP As IGeoProcessor
        Set GP = New GeoProcessor
        GP.OverwriteOutput = True
        'Declare and set a variant array to hold the parameters
        Dim parameters As IVariantArray
        Set parameters = New VarArray
        'Populate the variant array with the parameters
        parameters.Add Filex4
        'Now execute the Dissolve tool
        GP.Execute "delete_management", parameters, Nothing
    Loop
    *****
    strFileName = dir$()

```

Option Explicit

```
' Subroutine: Appennino_tool_Click  
' Author: M.Ferretti  
' Date: May 2010  
' Description: This subroutine starts when the user pushes the Appennino Tool icon. This subroutine  
opens the Presentation Form (frmPres) and the Choose Form (frmHrOrFix)
```

```
Private Sub Appennino_tool_Click()  
frmPres.Show  
frmHrOrFix.Show  
End Sub
```

```
Private Function Appennino_tool_ToolTip() As String
```

```
Appennino_tool_ToolTip = "The Appennino Tool"
```

```
End Function
```