John S. Wood, Research Associate

GIS at the Harte Research Institute

Abstract

The Harte Research Institute for Gulf of Mexico Studies (HRI) is a new face on the rapidly expanding horizon of marine research. The mission of HRI, an endowed research institute at Texas A&M University-Corpus Christi, is to support and advance the long-term sustainable use and conservation of the Gulf of Mexico, focusing on coastal and marine policy, biodiversity, and conservation through an ecosystem-level approach. The goals of the HRI GIS department are to provide researchers and collaborators a cutting-edge research facility with capabilities unmatched in the marine research arena and serve as an information portal for researchers, industry, education, state and federal agencies, and conservation interests from the U.S., Mexico, and Cuba. Collaborating with existing research facilities and assuming a leadership role to the ecosystem approach require the use of enterprise-like technology in a project-based realm, and HRI GIS is developing tools to enable it to process and store data on an enterprise scale.

In 2001, Mr. Edward Harte announced a $46 million dollar endowment to Texas A&M University-Corpus Christi, with the intent of establishing a world-class, international research center for Gulf of Mexico studies. His vision was to bring to the Gulf of Mexico a research facility which would serve to function much as Woods Hole Oceanographic Institution (WHOI) functions for the Atlantic coast, and Scripps Institution of Oceanography functions for the Pacific Coast. The mission of this newly endowed and developing institute at Texas A&M University-Corpus Christi is to support and advance the long-term sustainable use and conservation of the Gulf of Mexico. Cooperating and collaborating with research facilities and researchers in United States, Mexico, and Cuba, it will promote excellence and innovation in interdisciplinary scientific research, public policy initiatives, and education of the public. A 19-member Advisory Council of renowned scientists, policy makers and conservationists was formed, and planning meetings have been held in Corpus Christi, Houston, Havana, Florida, Monterrey and Veracruz.

This generous endowment will fund six endowed chair positions: Coastal and Marine Policy, Biodiversity, Ecosystem Modeling, Socio-Economics, Human and Ocean Health, and Geographic Information Systems. The advisory council recognizes the important role that GIS will play in the structure of this cutting-edge research facility, demonstrating the belief that this technology is a vital component for collaboration and cooperation between the other departments, as well as other collaborating research facilities.

An additional $18 million from the State of Texas to house the Institute supplemented this $46 million dollar endowment. Building construction started in 2003, and should be complete in 2005. Although wide-ranging activity and research are planned after HRI officially "opens," several early projects are already underway: Veracruz reef studies; a 50-year update of 'Bulletin 89', an initiative to determine the total biodiversity of the Gulf of Mexico; the State of the Gulf Symposium, the Gulf of Mexico GIS; and, GulfBase, a Gulfwide research website.

Figure 1 : Harte Research Institute for Gulf of Mexico Studies

---

Top-Notch Facilities…

The GIS lab will be located on the second floor of the new building, which is expected to be completed in mid-summer, 2005. This approximately 3000 sq. ft. laboratory will house as many as 25 GIS-capable workstations, with complete LAN and wireless capabilities. Preliminary plans call for two servers; the first will serve data throughout the institute and campus, the second server will serve as a data portal, connecting collaborating researchers to the GIS database. Currently, this database contains almost a terabyte of data, covering the Gulf Coast from Key West, Florida to Brownsville, Texas. Some data has been collected for Mexico and Cuba, but this data is very hard to come by. In addition to the servers, plans call for a sixty inch plotter, a forty-two inch plotter-scanner, and other peripherals, as well as data collection instruments. Other equipment, including

The Environmental Systems Research Institute (ESRI) has generously offered their expertise in GIS Systems design, to ensure that all hardware, software and networking infrastructure will be compatible, efficient and top-of-the-line. Actual purchase of most equipment and software will take place after final construction is complete, with considerable influence from the Endowed Chair, who has not been selected yet.

GIS Data…

The data collection at HRI is at this time primarily base data layers, collected from state information portals, such as the Texas Natural Resources Information System (TNRIS), Texas General Land Office, Texas Commission for Environmental Quality's

External Geographic Information Systems Page, Alabama Water Quality Information System, Louisiana Geographic Information Systems Council, Florida Department of Environmental Protection (FDEP), Florida's Land Boundary Information System (LABINS), the Mississippi Automated Resource Information System (MARIS), Natural Resources Conservation Service (NRCS), the National Agricultural Imagery Program, EPA's CMAP Program, The National Atlas, Instituto Nacional de Estadística Geografía e Informática (INEGI), GIS Data Depot, NASA's GES Distributed Active Archive Center, The World Data Center, Office of Coast Survey, National Marine Sanctuaries Program, NOAA Electronic Navigational Charts, and other state and federal agencies, as well as many Non-Governmental Organizations such as The Nature Conservancy, Pronatura, Surf Rider Foundation, the Socio-Economic Data And Application Center (SEDAC), Audubon Society, and NatureServe.

By collecting data from these expansive and diverse portals into one library, HRI hopes to become the premier site for GIS data for the Gulf Coast. The data sets have been organized following the State of Texas Department of Information Resources guidelines, with some alterations; since these guidelines were developed, it is no longer necessary for datasets to be in one specified projection. The suggested folder names were altered so that there are no spaces. Numerous sub-headings have been developed to make navigating to a specific type of data easier.

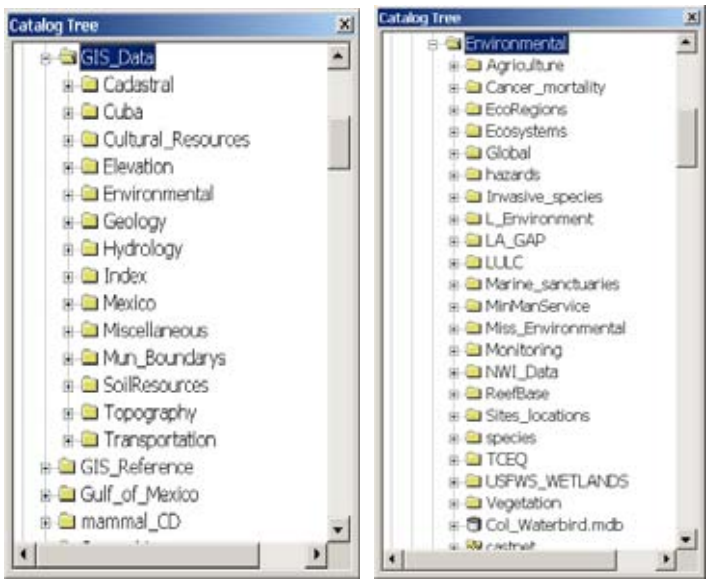


Figure 2 : ArcCatalog file structure

Digital Ortho Quarter-Quads (DOQQs), National Elevation Dataset (NED) DEMs and Digital Raster Graphics (DRGs) are stored by degree block, and image catalogs are built for each block and data type, facilitating rapid loading and ease of use.

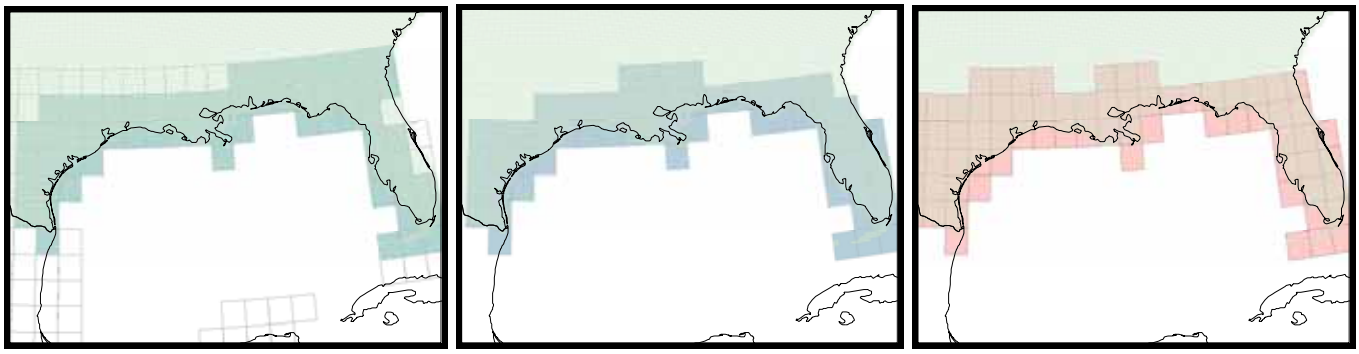
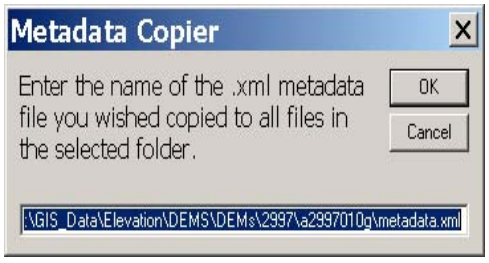


Figure 3 : Data holdings; National Elevation Datasets, Digital Ortho Quarter Quads, and Digital Raster Graphics.

GIS Data Processing Tools…

There are 59 degree blocks of DOQQs, 71 degree blocks of DRGs, with approximately 64 datasets in each degree block, as well as 64 degree blocks of NED. Although it varies from state to state, most of the data is delivered from the state agencies in compressed format. There is also a variety of metadata formats, varying by agency, and often varying within an agency. On rare occasions, the metadata has already been imported and attached as an ArcCatalog .xml file. More often, there is one metadata file that applies to an entire set of data. Several tools were developed as Visual Basics Application macros to attach metadata, and build pyramids and thumbnails for this massive amount of data. The source code for these macros is included at the end of this paper, or by contacting the author.

Metadata Copier



This macro copies the .xml file entered by the user to all the files in the selected folder. This simple routine uses the FileSystem.filecopy function, using the filename entered in the GUI as the input file, and copying it to each filename in the selected directory.

Figure 4 : Metadata Copier Input GUI.

Metadata Wizard

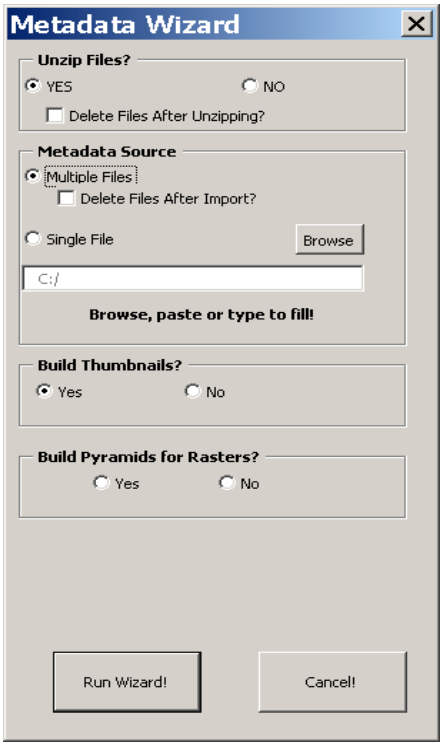


Figure 5: Metadata Wizard Interface

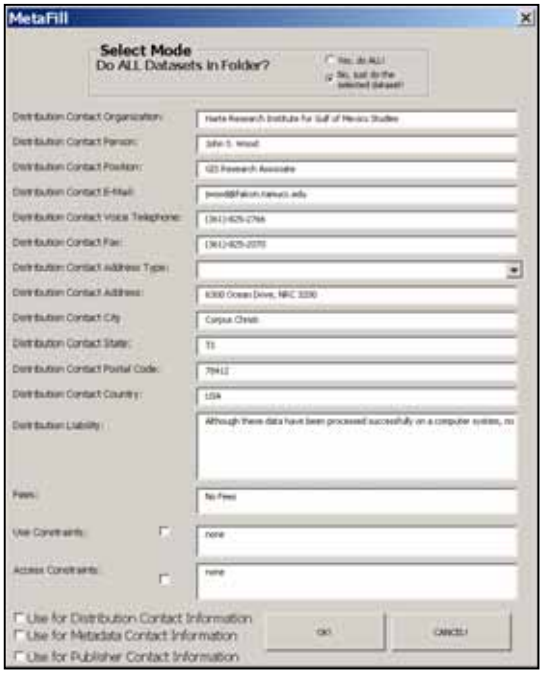
This macro, dubbed the Metadata Wizard, cuts processing time tremendously. The user first selects the directory containing the downloaded data files. If the files are in a WinZip compatible compressed format, the user can specify if the macro should decompress them. If so, the macro then writes and calls a Winzip 9.0 command line batch file, which automatically unzips the files into the same directory. The user may select an option to delete the compressed files after decompression.

After the files are decompressed, the user then selects to import either individual metadata files, or a single metadata file for the entire directory. For individual files, the metadata must have the same base filename; anyfile.shp must have an associated anyfile.met. The macro will recognize numerous metadata file extensions (.met, .fgd, .txt, .xml, .mtd, .sgml), and by accessing the macro code in the Visual Basics macro editor, others can be added easily. It then uses the appropriate ArcCatalog Metadata Import tool to import the metadata.

The user may also select options to build thumbnails for most data types, and/or pyramids for any rasters is the directory.

The Metadata Wizard is actually a combination of several separate macros. The first macro was adapted from a macro written by Rob Crumlin, available through ArcScripts, that recursively built pyramids for a specified raster type. This macro was then adapted to import metadata, the decompression feature was later added, and then the macro was adapted to build thumbnails for all rasters in a directory.

MetaFill Macro



MetaFill is another macro developed to speed processing metadata for large datasets. In the FGDC standard metadata, there are at least three different places where contact information is required: in the publisher contact section, the distribution contact section and in the metadata contact section. Filling all these blanks is time-consuming and tedious…and quite often, the same information is required in multiple locations.

This macro will fill in any or all the information, in the specified locations, for either a single metadata file or all the metadata files in a specified directory. The user may also use it to fill in those critical but often repetitive use and access constraints, distribution liability and fees sections. Use CAUTION when employing this macro…it will overwrite any section with text in the GUI! The user may change or delete default values by accessing the VBA code.

Figure 6: MetaFill GUI

Multiple Spatial Reference

Although this macro was not developed at HRI, it has been a great help in processing downloaded datasets. It was first written by Vicki Magnis and Jennifer McCollom, and later ported to ArcCatalog 9x by Duane Cranford. The user selects the file to assign a common projection to, and then calls this macro. After the user selects the proper coordinate system, the macro will assign the projection file to all the selected files. This macro is available at http://arcscripts.esri.com/details.asp?dbid=13664.

WriteDirectory Macro

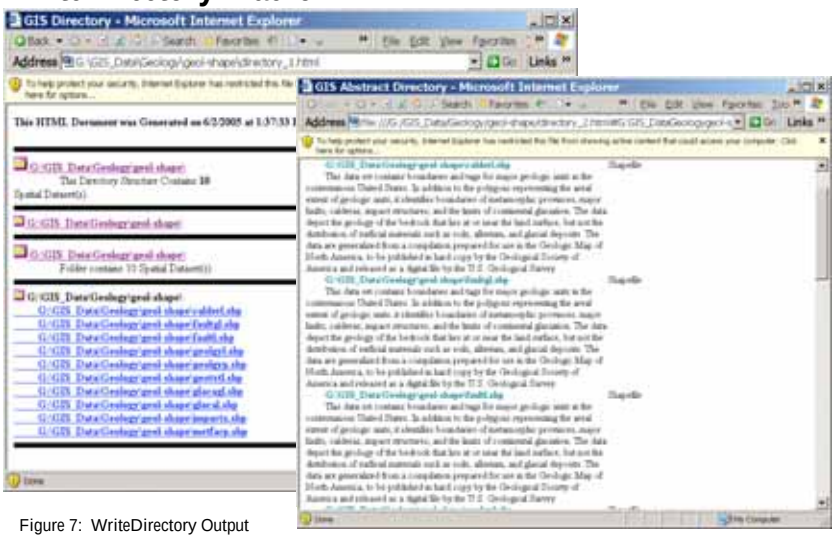


Figure 7: WriteDirectory Output

Another great tool developed at HRI, the WriteDirectory macro, will write the directory structure, filename, and file type to an HTML page, which is then linked to another page that contains the abstract, if available in the ESRI metadata. It will begin at the specified directory and will write and link HTML for that directory and any subdirectories. Until a full IMS site is developed, these HTML pages serve as a user's catalog to the HRI library.

Because it will progress into subdirectories, this macro has proven extremely useful. With minor modifications, it can be used to count specific datasets of a specific file type, or find directories where metadata is either non-existent or incomplete.

Cautions, Questions and Suggestions…

These macros should be used with caution. A test run on expendable datasets is highly suggested, both to familiarize the user with required inputs and outputs, and to find any bugs that might exist. When using these on computers other than the one they were written on, the user is often required to add references in VBA. Some of the macros require Visual Basics to be installed on the computer, in order to access some of the functions, but the macros themselves can be adapted to eliminate this. Please contact the author with any suggestions, questions, or fixes that you find useful.

The Future…

The future for the Harte Research Institute for Gulf of Mexico Studies is looking very bright: a search has been initiated for the Endowed Chair in GIS, and several candidates have been interviewed. The brand-new facilities should be complete within a matter of weeks; shortly after move-in, an IMS site will be installed, enabling HRI to distribute the data that is already collected, and hopefully receive even more from other research facilities and collaborators around the Gulf. Several research centers at the Texas A&M University-Corpus Christi will come under the HRI umbrella, and a

new doctoral program in Coastal and Marine Systems Sciences, with a strong emphasis in Marine GIS, will begin classes in the fall of 2005. HRI looks forward to serving as an active member of the Marine GIS community!

Appendix…Source Code

Metadata Copier Macro

```
Option Explicit
Public Sub metadata_Copier()
'Purpose: For all data items contained in a folder, list in .html file
    Dim pGxApp As esriCatalogUI.IGxApplication
    Dim pGxObjectCont As esriCatalog.IGxObjectContainer
    Dim pEnumGxObj As esriCatalog.IEnumGxObject
    Dim pGxObj As esriCatalog.IGxObject
    Dim pGxObj1 As esriCatalog.IGxObject
    Dim log_filename As String
    Dim fs As Scripting.FileSystemObject
    Dim write_File As String
    Dim log As Scripting.TextStream
    Dim pcounter As Integer
    Dim nDir As Integer
    Dim fileCount As Integer
    Dim read_File As String
    read_File = InputBox("Enter the name of the .xml metadata file you wished copied to all files in the selected folder.", "Metadata Copier",
"G:\GIS_Data\Elevation\DEMS\DEMs\2997\a2997010g\metadata.xml")
    Set pGxApp = esriArcCatalog.Application
    Set pGxObj = pGxApp.SelectedObject
    If Not TypeOf pGxObj Is IGxFolder Then
        MsgBox "Please select a Folder", vbExclamation
        Exit Sub
    Else
        Set pGxObjectCont = pGxObj
        Set pEnumGxObj = pGxObjectCont.Children      'Check to see if there are children
        If pEnumGxObj Is Nothing Then
            'MsgBox "Nothing was found in the  " & pGxObj.FullName & " Folder"
            Exit Sub
        Else
       ' Set fs = CreateObject("Scripting.FileSystemObject")
       ' Dim metadata_read As TextStream
       ' Set metadata_read = fs.OpenTextFile(readFile, ForReading, True)
       ' Dim meta_Text As String
       ' meta_Text = metadata_read.ReadAll
       ' MsgBox ("Here is what it said" & meta_text)
        Set pGxObj1 = pEnumGxObj.Next  ' For each object that is in the Folder
        Dim intCount As Integer
              Do While Not pGxObj1 Is Nothing
                If pGxObj1.Category = "Folder" Then
                  subfolder pGxObj1, read_File, write_File
                End If
                'Set fs = CreateObject("Scripting.File")
                If Not pGxObj1.Category = "Folder" Then
                    write_File = pGxObj1.FullName & ".xml"
                FileSystem.FileCopy read_File, write_File
                End If
                Set pGxObj1 = pEnumGxObj.Next
              Loop
```

```
      End If
   End If
MsgBox "Done!  "
End Sub
Private Sub subfolder(ByVal pGxObj As esriCatalog.IGxObject, read_File As String, write_File As String)
   Dim pGxObjectCont As esriCatalog.IGxObjectContainer
   Dim pEnumGxObj As esriCatalog.IEnumGxObject
   Dim pGxObj1 As esriCatalog.IGxObject
   Dim fs As FileSystemObject

   Set pGxObjectCont = pGxObj
   Set pEnumGxObj = pGxObjectCont.Children      'Check to see if there are children
   Set fs = CreateObject("Scripting.FileSystemObject")
   If pEnumGxObj Is Nothing Then
      MsgBox "Nothing was found in the  " & pGxObj.FullName & " Folder"
      'Exit Sub
   Else
   MsgBox "In Function"
   Set pGxObj1 = pEnumGxObj.Next  ' For each object that is in the Folder,
   Do While Not pGxObj1 Is Nothing
      If pGxObj1.Category = "Folder" Then
        subfolder pGxObj1, read_File, write_File
      Else
         FileCopy read_File, write_File
      End If
     Set pGxObj1 = pEnumGxObj.Next
   Loop
   End If
End Sub
```

Metadata Wizard

```
Sub RecursiveTN_MD()
'Methods: IMetadata.Metadata, IMetadataImport.Import
'Purpose: For all data items contained in a folder, import a specified XML document
'as metadata. Checks to make sure the selected object is a Folder and also that
'the folder is not empty

   Dim pGxApp As esriCatalogUI.IGxApplication
   Dim pGxSel As esriCatalog.IGxSelection
   Dim pGxObjectCont As esriCatalog.IGxObjectContainer
   Dim pEnumGxObj As esriCatalog.IEnumGxObject
   Dim pGxObj As esriCatalog.IGxObject
   Dim pGxObj1 As esriCatalog.IGxObject
   Dim pMD As esriGeoDatabase.IMetadata
   Dim m_pImportTXT As esriCatalog.ImportMPTXT
   Dim m_pImportMPTXT As esriCatalog.ImportMPTXT
   Dim m_pImportXML As esriCatalog.ImportXML
   Dim m_pImportSGML As esriCatalog.ImportMPSGML
   Dim pImport As esriCatalog.IMetadataImport
   Dim metaExtension As String
   Dim fs As Scripting.FileSystemObject
   Dim sFile As String
   Dim RetVal As Variant
   Dim batch As Scripting.TextStream
   Dim sPath As String
   Dim sPos As Long
   Dim multifileFlag As Boolean
   Dim strName As String
   Dim metaTypeFlag As String
   Dim deletefiles
   Dim buildTN As Boolean
   Dim BuildPyramids As Boolean

   Set pGxApp = esriArcCatalog.Application
   'pGxApp.RefreshWindow
```

```
Set pGxObj = pGxApp.SelectedObject
' MsgBox ("pGxObj = " & pGxObj.FullName)
deletefiles = vbNo
multifileFlag = True
MetaWiz.Show

responseMeta = MetaWiz.MetaButtonYes
'MsgBox response & " multi-metadata"
If responseMeta = False Then
   strName = MetaWiz.tbMetaFileName
  ' MsgBox strName & "STRNAM"
End If
buildTN = MetaWiz.Buildthumbnails
BuildPyramids = MetaWiz.obPyramYes



responseZip = MetaWiz.Unzip_YES
If responseZip = True Then
   Set fs = CreateObject("Scripting.FileSystemObject")
   Set batch = fs.CreateTextFile("C:\batch_unzip.bat", True)
   'sPos = InStrRev(pGxObj.FullName, "\") 'get position of \ character from beginning of string
   sPath = pGxObj.FullName & "\"
   ' MsgBox ("Spath = " & sPath)
   Dim myChar As String
   myChar = Strings.Chr(34)
   batch.WriteLine (myChar & "C:\program files\winzip\wzunzip" & myChar & " -o -yp " _
        & sPath & "*.zip " & sPath)
   batch.Close
   RetVal = Shell("C:\batch_unzip.bat", vbMaximizedFocus)
   MsgBox "Please Wait for Files to be Unzipped"

   If MetaWiz.chb1 = True Then
      'MsgBox "MetaWiz.chb1 =  True...deleting zip files"
      Set fs = CreateObject("Scripting.FileSystemObject")
      Set batch = fs.CreateTextFile("C:\batch_deletezip.bat", True)
      batch.WriteLine ("del " & sPath & "*.zip")
      batch.Close
      RetVal = Shell("C:\batch_deletezip.bat", vbMaximizedFocus)
         'MsgBox "Please Wait for Files to be Deleted"
   End If
      'deletefiles = MsgBox("Delete Metadata Files After Import?", vbYesNo, "Delete Metadata Files")
End If

multifileFlag = responseMeta
'MsgBox multifileFlag

pGxObj.Refresh

If Not TypeOf pGxObj Is esriCatalog.IGxFolder Then
   MsgBox "Please select a Folder", vbExclamation
   Exit Sub
Else
   Set pGxObjectCont = pGxObj
   'Check to see if there are children
   Set pEnumGxObj = pGxObjectCont.Children
   If pEnumGxObj Is Nothing Then
      MsgBox "Nothing was found in the  " & pGxObj.FullName & " Folder"
      Exit Sub
   Else
      'For each object that is in the Folder, import the text document as metadata
      Set pGxObj1 = pEnumGxObj.Next
      If multifileFlag = False Then
        '  MsgBox "Metadata File Name is  " & tbMetaFileName
        metaExtension = Strings.Right(MetaWiz.tbMetaFileName, 4)
        '  MsgBox "MetaExtension is " & metaExtension
        metaExtension = ".unknown"  'InputBox("What Extension is the metadata ?...", "Extension?", extName)
      End If
```

```
Do While Not pGxObj1 Is Nothing
    Set fs = CreateObject("Scripting.FileSystemObject")

    sPos = InStrRev(pGxObj1.FullName, ".") 'get position of \ character from beginning of string
    sPath = Strings.Left(pGxObj1.FullName, sPos)  'get path
    If fs.FileExists(Strings.Left(pGxObj1.FullName, Len(pGxObj1.FullName) - 4) & ".txt") = True Then
       metaTypeFlag = "TXT"
       metaExtension = ".txt"
    End If
    If fs.FileExists(Strings.Left(pGxObj1.FullName, Len(pGxObj1.FullName) - 4) & ".fgd") = True Then
       metaExtension = ".fgd"
       metaTypeFlag = "TXT"
       'MsgBox pGxObj1.FullName & "   " & extName & "    " & metaTypeFlag
    End If

    If fs.FileExists(Strings.Left(pGxObj1.FullName, Len(pGxObj1.FullName) - 4) & ".met") = True Then
       metaTypeFlag = "TXT"
       metaExtension = ".met"
    End If

    If fs.FileExists(Strings.Left(pGxObj1.FullName, Len(pGxObj1.FullName) - 4) & ".mtd") = True Then
       metaTypeFlag = "TXT"
       metaExtension = ".mtd"
    End If

    If fs.FileExists(Strings.Left(pGxObj1.FullName, Len(pGxObj1.FullName) - 4) & ".xml") = True Then
       metaTypeFlag = "XML"
       metaExtension = ".xml"
       '  MsgBox "File is XML"
    End If

    If fs.FileExists(Strings.Left(pGxObj1.FullName, Len(pGxObj1.FullName) - 4) & ".sgml") = True Then
       '  extName = ".sgml"
       metaTypeFlag = "SGML"
       metaExtension = ".sgml"
       'MsgBox "SGML found for " & pGxObj1.FullName
    End If

    sFile = pGxObj1.FullName
    If multifileFlag = True Then
       Dim strLength As Integer
       strLength = Len(sPath) - 1
       strName = Strings.Left(pGxObj1.FullName, strLength) & metaExtension
    Else
       strName = MetaWiz.tbMetaFileName

       '*********TEMPORARY*********
       Set m_pImportMPTXT = New esriCatalog.ImportMPTXT
       Set pImport = m_pImportMPTXT
       '*********TEMPORARY*********
    End If
    Set pMD = pGxObj1
    If metaTypeFlag = "TXT" Then
       Set m_pImportMPTXT = New esriCatalog.ImportMPTXT
       Set pImport = m_pImportMPTXT
    End If
    If metaTypeFlag = "XML" Then
       Set m_pImportXML = New esriCatalog.ImportXML
       Set pImport = m_pImportXML
    End If
    If metaTypeFlag = "SGML" Then
       Set m_pImportSGML = New esriCatalog.ImportMPSGML
       Set pImport = m_pImportSGML
    End If


    'Import a local txt document to metadata
    Set fs = CreateObject("Scripting.FileSystemObject")
   ' MsgBox "strName = " & strName
    If fs.FileExists(strName) = True Then
```

```
            pImport.Import strName, pMD
            '    MsgBox "Importing " & strName
            If MetaWiz.chb2 = True Then
               Set fs = CreateObject("Scripting.FileSystemObject")
               Set batch = fs.CreateTextFile("C:\batch_delete_meta.bat", True)
               batch.WriteLine ("del " & strName)
               batch.Close
               RetVal = Shell("C:\batch_delete_meta.bat", vbMaximizedFocus)
               'MsgBox "Please Wait for Files to be Deleted"
                  End If

            End If
            If Not pGxObj1.Category = "Folder" Then
              If buildTN = True Then
                 BuildThumbnail pGxObj1, sFile
              End If
              If pGxObj1.Category = "Raster Dataset" Then
                 BuildPyramidX pGxObj1, sFile
                ' MsgBox "Raster Dataset = True"
             End If

            End If
            Set pGxObj1 = pEnumGxObj.Next

          Loop
      End If
   End If
   MsgBox "Done!!!"

End Sub
Private Function BuildThumbnail(ByVal pGxObj1 As esriCatalog.IGxObject, sFile As String)

      Dim qApp As esriCatalogUI.IGxApplication
      Dim bUID As New esriSystem.UID
      Dim m_pGxCatalog As esriCatalog.IGxCatalog
      Dim m_pAoGxApp As esriCatalogUI.IGxApplication          'Get a refence to the application

      Set qApp = esriArcCatalog.Application 'Create a UID Object to hold the ClassID of the view
      Set m_pAoGxApp = esriArcCatalog.Application
      Set m_pGxCatalog = m_pAoGxApp.Catalog

      m_pGxCatalog.Location = sFile
      bUID = "esriCatalogUI.gxpreview"            'used to open preview view

      qApp.ViewClassID = bUID 'open the view
          'To open geographic or table view the preview must be opened
          'then open the type of preview required
      If (TypeOf qApp.View Is esriCatalogUI.IGxPreview) Then
        Dim pview As esriCatalogUI.IGxView
        Dim ppreview As esriCatalogUI.IGxPreview
        Set pview = qApp.View
        Set ppreview = pview
        bUID.Value = "esriCatalogUI.GxGeoGraphicView"
        On Error Resume Next
        ppreview.ViewClassID = bUID
      End If

      Dim pCmdItem As esriFramework.ICommandItem
      Set pCmdItem = esriArcCatalog.Application.Document.CommandBars.Find(ArcID.Geography_CreateThumbnail)
      pCmdItem.Execute

End Function
Private Function BuildPyramidX(ByVal pGxObj1 As esriCatalog.IGxObject, sFile As String)
   Dim pApp As esriCatalogUI.IGxApplication
   Dim pRWS As esriDataSourcesRaster.IRasterWorkspace
   Dim pWs As esriGeoDatabase.IWorkspace
   Dim pWSF As esriGeoDatabase.IWorkspaceFactory
   Dim pRasterDs As esriGeoDatabase.IRasterDataset
   Dim pPyramid As esriDataSourcesRaster.IRasterPyramid
   Dim sPath As String
```

```
    Dim lFilePathLength As Long
    Dim sPos As Long
    Dim sName As String
    Dim lFileNameLength As Long


    Set pApp = esriArcCatalog.Application
    Set pWSF = New esriDataSourcesRaster.RasterWorkspaceFactory

    sPos = InStrRev(sFile, "\") 'get position of \ character from beginning of string
    sPath = Strings.Left(sFile, sPos) 'get path
    lFilePathLength = Len(sFile)
    lFileNameLength = (lFilePathLength - sPos)
    sName = Strings.Right(sFile, lFileNameLength) 'gets just the file name; no path included

    If pWSF.IsWorkspace(sPath) = True Then 'get rasterworkspace
       Set pRWS = pWSF.OpenFromFile(sPath, 0)
            'Get RasterDataset
       Set pRasterDs = pRWS.OpenRasterDataset(sName)
       Set pPyramid = pRasterDs
            'Create pyramid
       If pPyramid.Present = False Then
         pPyramid.Create
       End If
    End If

End Function
FORM CODE

Private Sub cbBrowse_Click()
Dim metafile As String
    CommonDialog1.ShowOpen
    tbMetaFileName.Text = CommonDialog1.FileName
    metafile = tbMetaFileName.Text
End Sub
Private Sub cbCancel_Click()
    MsgBox "LATER!"
    MetaWiz.Hide
    Exit Sub
End Sub
Private Sub chb1_Click()
    If chb1.Value = True Then
     '   MsgBox "Delete zips in form"
    Else
      ' MsgBox "Do not delete Zips in form"
    End If
End Sub
Private Sub cbRun_Click()
   MetaWiz.Hide
  ' MsgBox deletemeta & vbNewLine & metaFile & vbNewLine & multipleMeta
   'RecursiveTN_MD_GUI(metafile, deleteZip, deletemeta, multipleMeta)
End Sub
Private Sub meta_source_Click()
'meta_source = multipleMeta
End Sub

Private Sub MetaButtonYes_Click()
    multipleMeta = True
    chb2.Enabled = True
    cbBrowse.Enabled = False
    tbMetaFileName.Enabled = False
End Sub

Private Sub metaSingleBttn_Click()
        multipleMeta = False
        chb2.Enabled = False
        cbBrowse.Enabled = True
        tbMetaFileName.Cut
        tbMetaFileName.Enabled = True
        tbMetaFileName.SetFocus
```

```
End Sub

Private Sub obPyramYes_Click()
   Dim pyraYes As Boolean
   pyraYes = True
End Sub

Private Sub obPyraNo_Click()
   Dim pyraYes As Boolean
   pyraYes = False
End Sub

Private Sub tbMetaFileName_Change()
   singleMeta = tbMetaFileName.Text
End Sub
Private Sub Unzip_NO_Click()
   response = vbNo
   chb1.Enabled = False
End Sub
Private Sub Unzip_YES_Click()
   response1 = vbYes
  chb1.Enabled = True
End Sub
```

MetaFill Macro

```
Public Sub fillMeta()
'Purpose: For all data items contained in a folder, Create metatdata
   Dim pGxApp As IGxApplication



   Dim pGxObj As IGxObject
   Dim strSimple As String
   Dim contact As Boolean
   Dim pGxObjectCont As esriCatalog.IGxObjectContainer
   Dim pEnumGxObj As esriCatalog.IEnumGxObject

   Dim pGxCatalog As IGxCatalog


   Set pGxApp = Application
   Set pGxCatalog = pGxApp.Catalog

   Set pGxObj = pGxApp.SelectedObject
   'MsgBox pGxObj.Name

   metaFill.Show

   Do While metaFill.obRecursiveNo = False And metaFill.obRecursiveYes = False
      MsgBox "You must Select a Mode!"
      metaFill.Show
   Loop

   If metaFill.obRecursiveYes = True Then
     '
      Set pGxObjectCont = pGxObj
      Set pEnumGxObj = pGxObjectCont.Children      'Check to see if there are children
      If pEnumGxObj Is Nothing Then
         'MsgBox "Nothing was found in the  " & pGxObj.FullName & " Folder"
         Exit Sub
      End If
      Set pGxObj = pEnumGxObj.Next

      Do While Not pGxObj Is Nothing
            fillMetaBlanks pGxObj
            Set pGxObj = pEnumGxObj.Next
      Loop
```

```
    Else
        fillMetaBlanks pGxObj
    End If
    'MsgBox strSimple
    Dim pGxView As IGxView
    Set pGxView = pGxApp.View
    If TypeOf pGxView Is IGxDocumentationView Then pGxView.Refresh

MsgBox "Done!  This macro brought to you by John S. Wood  "
End Sub
Private Function fillMetaBlanks(ByVal pGxObj As esriCatalog.IGxObject)

    Dim pGxObjectCont As esriCatalog.IGxObjectContainer
    Dim pEnumGxObj As esriCatalog.IEnumGxObject
    Dim pMD As IMetadata
    Dim pPS As IPropertySet
    Dim pXPS As IXmlPropertySet2

    Dim DistCountry As Boolean
    DistCountry = True

    Dim DistCity As Boolean
    DistCity = True

    Dim D_AddressType As Boolean
    D_AddressType = True

    Dim Dist_Address As Boolean
    Dist_Address = True

    Dim DistLiability As Boolean
    DistLiability = True

    Dim DistPerson As Boolean
    DistPerson = True

    Dim DistPost As Boolean
    DistPost = True

    Dim DistState As Boolean
    DistState = True

    Dim DistPos As Boolean
    DistPos = True

    Dim DistOrg As Boolean
    DistOrg = True

    Dim DistVoice As Boolean
    DistVoice = True

    Dim DistFax As Boolean
    DistFax = True

    Dim DistEmail As Boolean
    DistEmail = True

    Dim fees As Boolean
    fees = True

    Dim metaUse As Boolean
    metaUse = False

    Dim distUse As Boolean
    distUse = False

    Dim recurse As Boolean
    recurse = False

    Dim pubUse As Boolean
```

```
pubUse = True


metaUse = metaFill.cbMetaContactUse
distUse = metaFill.cbDistContactUse
pubUse = metaFill.cbPubContactuse
  '  MsgBox "pubUse is " & pubUse



If Not metaFill.tbDistFees.Text = "" Then
    Dim DistributionFees As String
    DistributionFees = metaFill.tbDistFees.Text
    ' MsgBox "DistributionLiability is  " & DistributionLiability
Else
    fees = False
End If

If Not metaFill.tbDistributionLiability.Text = "" Then
    Dim DistributionLiability As String
    DistributionLiability = metaFill.tbDistributionLiability.Text
' MsgBox "DistributionLiability is  " & DistributionLiability
Else
    DistLiability = False
End If

If Not metaFill.tbDistContPerson.Text = "" Then
    Dim DistContPerson As String
    DistContPerson = metaFill.tbDistContPerson.Text
' MsgBox "DistContPerson is  " & DistContPerson
Else
    DistPerson = False
End If

If Not metaFill.tbDistcontactVoice.Text = "" Then
    Dim DistcontactVoice As String
    DistcontactVoice = metaFill.tbDistcontactVoice.Text
   ' MsgBox "DistcontactVoice is  " & DistcontactVoice
Else
    DistVoice = False
End If

If Not metaFill.tbDistcontactState.Text = "" Then
    Dim DistcontactState As String
    DistcontactState = metaFill.tbDistcontactState.Text
   ' MsgBox "DistcontactState is  " & DistcontactState
Else
    DistState = False
End If


If Not metaFill.tbDistcontactPost.Text = "" Then
    Dim DistcontactPost As String
    DistcontactPost = metaFill.tbDistcontactPost.Text
   ' MsgBox "DistcontactPost is  " & DistcontactPost
Else
    DistPost = False
End If

If Not metaFill.tbDistcontactPos.Text = "" Then
    Dim DistcontactPos As String
    DistcontactPos = metaFill.tbDistcontactPos.Text
   'MsgBox "DistcontactPos is  " & DistcontactPos
Else
    DistPos = False
End If


If Not metaFill.tbDistcontactOrg.Text = "" Then
    Dim DistcontactOrg As String
```

```
   DistcontactOrg = metaFill.tbDistcontactOrg.Text
'   MsgBox "DistcontactOrg is  " & DistcontactOrg
Else
   DistOrg = False
End If


If Not metaFill.tbDistcontactFax.Text = "" Then
   Dim DistcontactFax As String
   DistcontactFax = metaFill.tbDistcontactFax.Text
 '  MsgBox "DistcontactFax is  " & DistcontactFax
Else
   DistFax = False
End If

If Not metaFill.tbDistcontactEmail.Text = "" Then
   Dim DistcontactEmail As String
   DistcontactEmail = metaFill.tbDistcontactEmail.Text
  '  MsgBox "DistcontactEmail is  " & DistcontactEmail
Else
    DistEmail = False
End If


If Not metaFill.tbDistcontactCountry.Text = "" Then
   Dim DistcontactCountry As String
   DistcontactCountry = metaFill.tbDistcontactCountry.Text
  '  MsgBox "DistcontactCountry is  " & DistcontactCountry
Else

   DistCountry = False
End If

If Not metaFill.tbDistcontactCity.Text = "" Then
   Dim DistcontactCity As String
   DistcontactCity = metaFill.tbDistcontactCity.Text
  '  MsgBox "DistcontactCity is  " & DistcontactCity
Else

   DistCity = False
End If


If Not metaFill.cbDistAddressType.Text = "" Then
Dim DistAddressType As String
   DistAddressType = metaFill.cbDistAddressType.Text
 '  MsgBox "DistAddressType is  " & DistAddressType
Else

   D_AddressType = False
End If

If Not metaFill.tbDistcontactAddress.Text = "" Then
   Dim DistcontactAddress As String
   DistcontactAddress = metaFill.tbDistcontactAddress.Text
 '  MsgBox "DistcontactAddress is  " & DistcontactAddress
Else

   Dist_Address = False
End If

Set pMD = pGxObj
Set pXPS = pMD.Metadata

  'MsgBox "should be false" & D_AddressType



  '  strSimple = pXPS.SimpleGetProperty("distinfo/distrib/cntinfo/cntorgp/cntorg")
   ' MsgBox strSimple
```

```
If metaFill.cbAccess = True Then
   pXPS.SetPropertyX "idinfo/accconst", _
      metaFill.tbAccess.Text, esriXPTText, esriXSPAAddOrReplace, False
End If
If metaFill.cbUse = True Then
      pXPS.SetPropertyX "idinfo/useconst", _
      metaFill.tbUse.Text, esriXPTText, esriXSPAAddOrReplace, False
End If

If distUse = True Then
   If DistPerson = True Then
      pXPS.SetPropertyX "distinfo/distrib/cntinfo/cntorgp/cntper", _
      DistContPerson, esriXPTText, esriXSPAAddOrReplace, False
   End If
   If DistPerson = True Then
      pXPS.SetPropertyX "distinfo/distrib/cntinfo/cntorg/cntper", _
      DistContPerson, esriXPTText, esriXSPAAddOrReplace, False
   End If
   If DistOrg = True Then
      pXPS.SetPropertyX "distinfo/distrib/cntinfo/cntorgp/cntorg", _
      DistcontactOrg, esriXPTText, esriXSPAAddOrReplace, False
   End If
   If DistOrg = True Then
      pXPS.SetPropertyX "distinfo/distrib/cntinfo/cntorg/cntorg", _
      DistcontactOrg, esriXPTText, esriXSPAAddOrReplace, False
   End If
   If DistPos = True Then
      pXPS.SetPropertyX "distinfo/distrib/cntinfo/cntpos", _
      DistcontactPos, esriXPTText, esriXSPAAddOrReplace, False
   End If
   If D_AddressType = True Then
      pXPS.SetPropertyX "distinfo/distrib/cntinfo/cntaddr/addrtype", _
      DistAddressType, esriXPTText, esriXSPAAddOrReplace, False
   End If
   If Dist_Address = True Then
      pXPS.SetPropertyX "distinfo/distrib/cntinfo/cntaddr/address", _
         DistcontactAddress, esriXPTText, esriXSPAAddOrReplace, False
   End If
   If DistCity = True Then
      pXPS.SetPropertyX "distinfo/distrib/cntinfo/cntaddr/city", _
         DistcontactCity, esriXPTText, esriXSPAAddOrReplace, False
   End If
   If DistState = True Then
      pXPS.SetPropertyX "distinfo/distrib/cntinfo/cntaddr/state", _
         DistcontactState, esriXPTText, esriXSPAAddOrReplace, False
   End If
   If DistPost = True Then
      pXPS.SetPropertyX "distinfo/distrib/cntinfo/cntaddr/postal", _
        DistcontactPost, esriXPTText, esriXSPAAddOrReplace, False
   End If
   If DistCountry = True Then
      pXPS.SetPropertyX "distinfo/distrib/cntinfo/cntaddr/country", _
        DistcontactCountry, esriXPTText, esriXSPAAddOrReplace, False
   End If
   If DistVoice = True Then
      pXPS.SetPropertyX "distinfo/distrib/cntinfo/cntvoice", _
         DistcontactVoice, esriXPTText, esriXSPAAddOrReplace, False
   End If

   If DistEmail = True Then
      pXPS.SetPropertyX "distinfo/distrib/cntinfo/cntemail", _
         DistcontactEmail, esriXPTText, esriXSPAAddOrReplace, False
   End If
   If fees = True Then
      pXPS.SetPropertyX "distinfo/stdorder/fees", _
         DistributionFees, esriXPTText, esriXSPAAddOrReplace, False
   End If
   If DistFax = True Then
      pXPS.SetPropertyX "distinfo/distrib/cntinfo/cntfax", _
         DistcontactFax, esriXPTText, esriXSPAAddOrReplace, False
```

```
    End If
    If DistLiability = True Then
       pXPS.SetPropertyX "distinfo/distliab", _
          DistributionLiability, esriXPTText, esriXSPAAddOrReplace, False
    End If
End If  'distuse

If metaUse = True Then
    If DistPerson = True Then
       pXPS.SetPropertyX "metainfo/metc/cntinfo/cntorgp/cntper", _
          DistContPerson, esriXPTText, esriXSPAAddOrReplace, False
    End If

    If DistOrg = True Then
       pXPS.SetPropertyX "metainfo/metc/cntinfo/cntorgp/cntorg", _
       DistcontactOrg, esriXPTText, esriXSPAAddOrReplace, False
    End If

    If D_AddressType = True Then
       pXPS.SetPropertyX "metainfo/metc/cntinfo/cntaddr/addrtype", _
       DistAddressType, esriXPTText, esriXSPAAddOrReplace, False
    End If
    If Dist_Address = True Then
       pXPS.SetPropertyX "metainfo/metc/cntinfo/cntaddr/address", _
          DistcontactAddress, esriXPTText, esriXSPAAddOrReplace, False
    End If
    If DistVoice = True Then
       pXPS.SetPropertyX "metainfo/metc/cntinfo/cntvoice", _
          DistcontactVoice, esriXPTText, esriXSPAAddOrReplace, False
    End If
    If DistFax = True Then
       pXPS.SetPropertyX "metainfo/metc/cntinfo/cntfax", _
          DistcontactFax, esriXPTText, esriXSPAAddOrReplace, False
    End If
    If DistEmail = True Then
       pXPS.SetPropertyX "metainfo/metc/cntinfo/cntemail", _
          DistcontactEmail, esriXPTText, esriXSPAAddOrReplace, False
    End If
    If DistCountry = True Then
       pXPS.SetPropertyX "metainfo/metc/cntinfo/cntaddr/country", _
         DistcontactCountry, esriXPTText, esriXSPAAddOrReplace, False
    End If
    If DistCity = True Then
       pXPS.SetPropertyX "metainfo/metc/cntinfo/cntaddr/city", _
          DistcontactCity, esriXPTText, esriXSPAAddOrReplace, False
    End If
    If DistState = True Then
       pXPS.SetPropertyX "metainfo/metc/cntinfo/cntaddr/state", _
          DistcontactState, esriXPTText, esriXSPAAddOrReplace, False
    End If
    If DistPost = True Then
       pXPS.SetPropertyX "metainfo/metc/cntinfo/cntaddr/postal", _
         DistcontactPost, esriXPTText, esriXSPAAddOrReplace, False
    End If
    If DistPos = True Then
       pXPS.SetPropertyX "metainfo/metc/cntinfo/cntpos", _
       DistcontactPos, esriXPTText, esriXSPAAddOrReplace, False
    End If
End If 'metause

If pubUse = True Then
    If DistPerson = True Then
       pXPS.SetPropertyX "idinfo/ptcontac/cntinfo/cntorgp/cntper", _
       DistContPerson, esriXPTText, esriXSPAAddOrReplace, False
    End If
    If DistPerson = True Then
       pXPS.SetPropertyX "idinfo/ptcontac/cntinfo/cntorg/cntper", _
       DistContPerson, esriXPTText, esriXSPAAddOrReplace, False
    End If
    If DistOrg = True Then
```

```
            pXPS.SetPropertyX "idinfo/ptcontac/cntinfo/cntorgp/cntorg", _
            DistcontactOrg, esriXPTText, esriXSPAAddOrReplace, False
        End If
        If DistOrg = True Then
            pXPS.SetPropertyX "idinfo/ptcontac/cntinfo/cntorg/cntorg", _
            DistcontactOrg, esriXPTText, esriXSPAAddOrReplace, False
        End If
        If D_AddressType = True Then
            pXPS.SetPropertyX "idinfo/ptcontac/cntinfo/cntaddr/addtype", _
                DistAddressType, esriXPTText, esriXSPAAddOrReplace, False
        End If
            pXPS.SetPropertyX "idinfo/ptcontac/cntinfo/cntaddr/address", _
                DistcontactAddress, esriXPTText, esriXSPAAddOrReplace, False
        If DistCity = True Then
            pXPS.SetPropertyX "idinfo/ptcontac/cntinfo/cntaddr/city", _
                DistcontactCity, esriXPTText, esriXSPAAddOrReplace, False
        End If
        If DistState = True Then
            pXPS.SetPropertyX "idinfo/ptcontac/cntinfo/cntaddr/state", _
                DistcontactState, esriXPTText, esriXSPAAddOrReplace, False
        End If
        If DistPost = True Then
            pXPS.SetPropertyX "idinfo/ptcontac/cntinfo/cntaddr/postal", _
              DistcontactPost, esriXPTText, esriXSPAAddOrReplace, False
        End If
        If DistCountry = True Then
            pXPS.SetPropertyX "idinfo/ptcontac/cntinfo/cntaddr/country", _
              DistcontactCountry, esriXPTText, esriXSPAAddOrReplace, False
         End If
        If DistVoice = True Then
            pXPS.SetPropertyX "idinfo/ptcontac/cntinfo/cntvoice", _
                DistcontactVoice, esriXPTText, esriXSPAAddOrReplace, False
        End If
        If DistEmail = True Then
            pXPS.SetPropertyX "idinfo/ptcontac/cntinfo/cntemail", _
                DistcontactEmail, esriXPTText, esriXSPAAddOrReplace, False
        End If
        If DistPos = True Then
            pXPS.SetPropertyX "idinfo/ptcontac/cntinfo/cntpos", _
            DistcontactPos, esriXPTText, esriXSPAAddOrReplace, False
        End If
        If DistFax = True Then
            pXPS.SetPropertyX "idinfo/ptcontac/cntinfo/cntfax", _
                DistcontactFax, esriXPTText, esriXSPAAddOrReplace, False
        End If

    End If 'pubUse

    pMD.Metadata = pXPS

End Function\
FORM CODE:
Private Sub cbAccess_Click()
    accessConstraints = True
End Sub

Private Sub cbCancel_Click()
    Unload Me
    Exit Sub
End Sub
Private Sub cbDistContactUse_Click()
    Dim distContactUse As Boolean
   ' distContactUse = True
End Sub
Private Sub cbMetaContactUse_Click()
    Dim metaContactUse As Boolean
  '  metaContactUse = True
End Sub

Private Sub cbOkay_Click()
```

```
    metaFill.Hide
End Sub
Private Sub cbPubContactuse_Click()
    Dim pubContactUse As Boolean
   ' pubContactUse = True
   '  MsgBox pubContactUse
End Sub

Private Sub cbUse_Click()
    useConstraints = True
End Sub

Private Sub obRecursiveNo_Click()
   ' Recursive = False
    'MsgBox "Recursive = No!"
End Sub
Private Sub obRecursiveYes_Click()
    Recursive = True
    'MsgBox "Recursive = Yes!"
End Sub

Private Sub UserForm_Initialize()
    cbDistAddressType.AddItem "mailing and physical address"
    cbDistAddressType.AddItem "mailing address"
    cbDistAddressType.AddItem "physical address"
End Sub
```

Multiple Spatial Reference

```
'Created by Vicki Magnis and ifer Mcollom 5/2004
'Edited by Duane Cranford to work in ArcCatalog 9 - 8/2004
'This tool will set the same projection for selected raster, shapefiles and coverages
Public Sub MultiSpatRef()
    Dim pApp As esriCatalogUI.IGxApplication
    Dim pGxObjectCont As esriCatalog.IGxObjectContainer
    Set pApp = esriArcCatalog.Application
    Dim pGxSelection As esriCatalog.IGxSelection
    Set pGxSelection = pApp.Selection
    Dim plist As esriCatalog.IEnumGxObject
    Set plist = pGxSelection.SelectedObjects
    Dim pGxObject As esriCatalog.IGxObject
    Dim pName As esriSystem.IName
    Dim pDS As esriGeoDatabase.IDataset
    Dim pSpatRef As esriGeometry.ISpatialReference
    Set pSpatRef = SelectSpatRef
    Dim pGDSE As esriGeoDatabase.IGeoDatasetSchemaEdit
    'Exit out of sub if you "Cancelled" the Coordinate System dialog box
    If pSpatRef Is Nothing Then
        Exit Sub
    End If
    Set pGxObject = plist.Next
    Do Until pGxObject Is Nothing
        If TypeOf pGxObject Is esriCatalog.IGxDataset Then
            Set pName = pGxObject.InternalObjectName
            Set pDS = pName.Open
            Set pGDSE = pDS
            With pGDSE
                If .CanAlterSpatialReference Then
                    .AlterSpatialReference pSpatRef
                Else: MsgBox "error"
                End If
            End With
        End If
        Set pGxObject = plist.Next
    Loop
MsgBox "Done!"
CleanUp:
    Set pSpatRef = Nothing
    'Set pRasterDs = Nothing
```

```
    Set pGDSE = Nothing
End Sub
Function SelectSpatRef() As esriGeometry.ISpatialReference
Set SelectSpatRef = Nothing
    Dim pGxDlg As esriCatalogUI.IGxDialog
    Set pGxDlg = New esriCatalogUI.GxDialog
    Dim pGxFilter As esriCatalog.IGxObjectFilter
    Set pGxFilter = New esriCatalog.GxFilterSpatialReferences
    Set pGxDlg.ObjectFilter = pGxFilter
    pGxDlg.Title = "Select Coordinate System"
    pGxDlg.StartingLocation = "Coordinate Systems\Favorites"
    'pGxDlg.Name = "North American Datum 1983.prj"
    pGxDlg.AllowMultiSelect = False
    Dim pEnumGxObj As esriCatalog.IEnumGxObject
    pGxDlg.DoModalOpen 0, pEnumGxObj
    Dim pGxPrj As esriCatalog.IGxPrjFile
    pEnumGxObj.Reset
    Dim pGxObj As esriCatalog.IGxObject
    Set pGxObj = pEnumGxObj.Next
    If Not pGxObj Is Nothing Then
        Set pGxPrj = pGxObj
        Set SelectSpatRef = pGxPrj.SpatialReference
    End If
End Function
```

WriteDirectory Macro

```
Option Explicit

Public Sub WriteDirectory()
'Purpose: For all data items contained in a folder, Create HTML Directory
    Dim pGxApp As IGxApplication
    Dim pGxObjectCont As IGxObjectContainer
    Dim pEnumGxObj As IEnumGxObject
    Dim pGxObj As IGxObject
    Dim pGxObj1 As IGxObject
    Dim fs As FileSystemObject
    Dim log As TextStream
    Dim pcounter As Integer
    Dim pMD As IMetadata
    Dim pPS As IPropertySet
    Dim pXPS As IXmlPropertySet
    Dim strSimple As String
    Dim m_sFullName As String
    Dim sStart As String
    Dim response
    'Dim continue1 As Boolean
    Dim log_filename As String
    Dim log_filename2 As String
    Dim colDirs As New Collection
    Dim i As Integer



    Set pGxApp = Application

    '**********************
    Set fs = CreateObject("Scripting.FileSystemObject")
    'Set pApp = esriArcCatalog.Application
    Set pGxObj = pGxApp.SelectedObject
    sStart = pGxObj.FullName
    log_filename = InputBox("Enter the path of file you want to write to. Include file name and extension.  ", "Output File", sStart & "\directory_1.html")

    If fs.FileExists(log_filename) = False Then
        Set log = fs.CreateTextFile(log_filename, True)
    Else
        Set log = fs.OpenTextFile(log_filename, ForWriting)
    End If
```

```
    listDirectories colDirs, sStart
    log_filename2 = InputBox("Enter the path of file you want the abstracts written to. Include file name and extension.  ", "Output File", sStart &
"\directory_2.html")

    log.WriteLine ("<html><head><meta name=" & Chr(34) & "GENERATOR" & Chr(34) & " content=" & Chr(34) & "Microsoft FrontPage 5.0" & Chr(34) &
"><meta name=" & Chr(34) & "ProgId" & Chr(34) & " content=" & Chr(34) & "FrontPage.Editor.Document" & Chr(34) & "><title>GIS
Directory</title><style></style><script type=" & Chr(34) & "text/javascript" & Chr(34) & " src=" & Chr(34) & "/sourcescripts.js" & Chr(34) &
"></script></head><body><p><b> This HTML Document was Generated on " & Date & " at  " & Time & "</b></p><table border=0 cellpadding=0
cellspacing=0  bordercolor=#111111 width=100%>")
    log.WriteLine ("<script type=" & Chr(34) & "text/javascript" & Chr(34) & ">")
    log.WriteLine ("var newwindow;")
    log.WriteLine ("function popit(url){")
    log.WriteLine ("     newwindow=window.open(url)")
    log.WriteLine ("     if(window.focus) {newwindow.focus()}")
    log.WriteLine ("        } ")
    log.WriteLine ("</script>")
    log.WriteLine ("<hr color=#000000 size=7>")
    log.Close
   ' MsgBox (colDirs(1))
    For i = 1 To colDirs.Count
       sStart = colDirs(i)
       numWrite log_filename, sStart
    Next
    If fs.FileExists(log_filename) = False Then
       Set log = fs.CreateTextFile(log_filename, True)
    Else
       Set log = fs.OpenTextFile(log_filename, ForAppending)
    End If
    log.WriteLine ("<hr color=#000000 size=7>")
    log.Close
    'continue1 = True
    For i = 1 To colDirs.Count
       sStart = colDirs(i)
       FirstWrite log_filename, sStart
    Next
    'continue1 = True
     For i = 1 To colDirs.Count
       sStart = colDirs(i)
       SecondWrite log_filename, sStart
    Next
    For i = 1 To colDirs.Count
       sStart = colDirs(i)
       Thirdwrite log_filename, sStart
    Next

     '**********************
    Set fs = CreateObject("Scripting.FileSystemObject")

    If fs.FileExists(log_filename2) = False Then
       Set log = fs.CreateTextFile(log_filename2, True)
    Else
       Set log = fs.OpenTextFile(log_filename2, ForWriting)
    End If
    log.WriteLine ("<html><head><meta name=" & Chr(34) & "GENERATOR" & Chr(34) & " content=" & Chr(34) & "Microsoft FrontPage 5.0" & Chr(34) &
"><meta name=" & Chr(34) & "ProgId" & Chr(34) & " content=" & Chr(34) & "FrontPage.Editor.Document" & Chr(34) & "><title>GIS Abstract
Directory</title><style></style><script type=" & Chr(34) & "text/javascript" & Chr(34) & " src=" & Chr(34) & "/sourcescripts.js" & Chr(34) &
"></script></head><body><p><b> This HTML Document was Generated on " & Date & " at  " & Time & "</b></p><table border=0 cellpadding=0
cellspacing=0  bordercolor=#111111 width=100%>")
    log.Close
     For i = 1 To colDirs.Count
       sStart = colDirs(i)
       fourthwrite log_filename2, sStart
    Next
    Set fs = CreateObject("Scripting.FileSystemObject")
    Set log = fs.OpenTextFile(log_filename2, ForAppending)
    log.Write ("<INPUT type=" & Chr(34) & "button" & Chr(34) & " value=" & Chr(34) & "Close Window" & Chr(34) & "onClick=" & Chr(34) &
"window.close()" & Chr(34) & "> </FORM>")
    log.Close
    MsgBox "Done!   Brought to you by John S. Wood"
    End Sub
```

```
Private Function numWrite(log_filename As String, sStart As String)
    Dim pGxApp As IGxApplication
    Dim pGxObjectCont As IGxObjectContainer
    Dim pEnumGxObj As IEnumGxObject
    Dim pGxObj As IGxObject
    Dim pGxObj1 As IGxObject
    Dim fs As FileSystemObject
    Dim log As TextStream
    Dim pcounter As Integer
    Dim nDir As Integer
    Dim fileCount As Integer
    Dim sStartFlag As Boolean

    Dim pGxCatalog As IGxCatalog

    Set pGxApp = Application
    Set pGxCatalog = pGxApp.Catalog
    sStartFlag = False
    On Error GoTo ENDSUB
    sStartFlag = True
    'sstart = InputBox("Enter where to start Write .HTML function. It can simply be a drive letter, eg. x:\", "Write Number HTML", "D:\Projects\")
    pGxCatalog.Location = sStart



    Set pGxObj = pGxApp.SelectedObject
    Set pGxObjectCont = pGxObj
    Set pEnumGxObj = pGxObjectCont.Children      'Check to see if there are children

    If pEnumGxObj Is Nothing Then
        'MsgBox "Nothing was found in the  " & pGxObj.FullName & " Folder"
        Exit Function
    Else
        Set fs = CreateObject("Scripting.FileSystemObject")
        If fs.FileExists(log_filename) = False Then
            Set log = fs.CreateTextFile(log_filename, True)
        Else
            Set log = fs.OpenTextFile(log_filename, ForAppending)
        End If
        log.WriteLine ("<table border=0 cellpadding=0 cellspacing=0  bordercolor=#111111 width=100%>")
        'writeline A ********************
        log.WriteLine ("<tr><td width=41%><b><img border=0 src=http://www.sci.tamucc.edu/~jwood/folder_closed.bmp width=20 height=20><a name=" &
pGxObj.FullName & "\><a href=#" & pGxObj.FullName & "\a>  " & pGxObj.FullName & "\  " & "</a></b></td><td width=59%></td>")
        log.Close

        Set pGxObj1 = pEnumGxObj.Next  ' For each object that is in the Folder

        Do While Not pGxObj1 Is Nothing
            If pGxObj1.Category = "Folder" Then
                inFolderNumwrite pGxObj1, log_filename, log, fileCount, pcounter
            End If
            If Not pGxObj1.Category = "Folder" Then
                fileCount = fileCount + 1
                pcounter = pcounter + 1
            End If
            Set pGxObj1 = pEnumGxObj.Next
        Loop
    End If
    Set log = fs.OpenTextFile(log_filename, ForAppending)
    log.WriteLine ("<tr><td
width=41%>&amp;nbsp;&amp;nbsp;&amp;nbsp;&amp;nbsp;&amp;nbsp;&amp;nbsp;&amp;nbsp;&amp;nbsp;&amp;nbsp;&amp;nbsp;&amp;nbsp;&amp;nbsp;&amp;nbsp;&amp;nbsp;&amp;nbsp;&amp;nbsp;This Directory
Structure Contains <b>" & pcounter & "</b> Spatial Dataset(s)</td><td width=59%></td>")
    log.WriteLine (" </tr></table> ")
    log.Close
' **************************
ENDSUB:
'MsgBox "Directory " & sStart & "does not exist!"
 'If sStartFlag = True Then
  '   MsgBox "Directory " & sStart & "does not exist!"
' End If
```

```
' *************************

End Function
Private Function inFolderNumwrite(ByVal pGxObj As IGxObject, log_filename As String, log As TextStream, fileCount As Integer, pcounter As Integer)

   Dim pGxObjectCont As IGxObjectContainer
   Dim pEnumGxObj As IEnumGxObject
   Dim pGxObj1  As IGxObject
   Dim fs As FileSystemObject

   Set pGxObjectCont = pGxObj
   Set pEnumGxObj = pGxObjectCont.Children      'Check to see if there are children
   Set fs = CreateObject("Scripting.FileSystemObject")

   If pEnumGxObj Is Nothing Then
   'MsgBox "Nothing was found in the  " & pGxObj.FullName & " Folder"
   'Return
   Else
      fileCount = 0

      Set pGxObj1 = pEnumGxObj.Next  ' For each object that is in the Folder,

      Do While Not pGxObj1 Is Nothing
         If pGxObj1.Category = "Folder" Then
            inFolderNumwrite pGxObj1, log_filename, log, fileCount, pcounter
         End If
         If Not pGxObj1.Category = "Folder" Then
            fileCount = fileCount + 1
            pcounter = pcounter + 1
         End If
         Set pGxObj1 = pEnumGxObj.Next
      Loop
   End If
End Function
Private Function FirstWrite(log_filename As String, sStart As String)
 'Purpose: For all data items contained in a folder, write the first directory
   Dim pGxApp As IGxApplication
   Dim pGxObjectCont As IGxObjectContainer
   Dim pEnumGxObj As IEnumGxObject
   Dim pGxObj As IGxObject
   Dim pGxObj1 As IGxObject
   Dim fs As FileSystemObject
   Dim log As TextStream
   Dim pcounter As Integer
   Dim m_pGxCatalog As IGxCatalog


   Set pGxApp = Application
   Set m_pGxCatalog = pGxApp.Catalog
   'sstart = InputBox("Enter where to start Write .HTML function. It can simply be a drive letter, eg. x:\", "Write First HTML ", "D:\Projects\")
   On Error GoTo ENDSUB
   m_pGxCatalog.Location = sStart

   Set pGxObj = pGxApp.SelectedObject
   Set pGxObjectCont = pGxObj
   Set pEnumGxObj = pGxObjectCont.Children      'Check to see if there are children



   If Not TypeOf pGxObj Is IGxFolder Then
       MsgBox "Please select a Folder", vbExclamation
       Exit Function
   Else
       Set pGxObjectCont = pGxObj
       Set pEnumGxObj = pGxObjectCont.Children      'Check to see if there are children

       If pEnumGxObj Is Nothing Then
          'MsgBox "Nothing was found in the  " & pGxObj.FullName & " Folder"
           Exit Function
       Else
```

```
        Set fs = CreateObject("Scripting.FileSystemObject")
        If fs.FileExists(log_filename) = False Then
            Set log = fs.CreateTextFile(log_filename, True)
        Else
            Set log = fs.OpenTextFile(log_filename, ForAppending)
        End If
        log.WriteLine ("<table border=0 cellpadding=0 cellspacing=0  bordercolor=#111111 width=100%>")
        '  write line 1 *****************
        log.WriteLine ("<tr><td width=70%><b><img border=0 src=http://www.sci.tamucc.edu/~jwood/folder_closed.bmp " _
                    & "width=16 height=16><a name=" & pGxObj.FullName & "\a href=#" & pGxObj.FullName & "\a> " & pGxObj.FullName _
                    & "\    " & "</a></b></td><td width=30%>" & pGxObj.Category & "</td>")
        log.Close

        Set pGxObj1 = pEnumGxObj.Next  ' For each object that is in the Folder

        Do While Not pGxObj1 Is Nothing
            If pGxObj1.Category = "Folder" Then
                Set fs = CreateObject("Scripting.FileSystemObject")
                If fs.FileExists(log_filename) = False Then
                    Set log = fs.CreateTextFile(log_filename, True)
                Else
                  Set log = fs.OpenTextFile(log_filename, ForAppending)
                End If
                log.WriteLine (" ")
                '********write line 2
                log.WriteLine ("<tr><td width=47%><b><img border=0 src=http://www.sci.tamucc.edu/~jwood/folder_closed.bmp width=16 " _
                & "height=16><a href=#" & pGxObj1.FullName & "\>  " & pGxObj1.FullName _
                & "\  " & " </a></td><td width=53%>" & pGxObj1.Category & "</b></td></tr>")
                log.Close
                inFolder1 pGxObj1, log_filename, log
            End If
            Set pGxObj1 = pEnumGxObj.Next

          Loop
      End If
      Set fs = CreateObject("Scripting.FileSystemObject")
      If fs.FileExists(log_filename) = False Then
        Set log = fs.CreateTextFile(log_filename, True)
      Else
       Set log = fs.OpenTextFile(log_filename, ForAppending)
      End If
      log.WriteLine (" </tr></table><hr color=#000000 size=7>")
      log.Close
  End If
ENDSUB:
  'MsgBox "Error"
End Function
  Private Function inFolder1(ByVal pGxObj As IGxObject, log_filename As String, log As TextStream)

  Dim pGxObjectCont As IGxObjectContainer
  Dim pEnumGxObj As IEnumGxObject
  Dim pGxObj1 As IGxObject
  Dim fs As FileSystemObject

  Set pGxObjectCont = pGxObj
  Set pEnumGxObj = pGxObjectCont.Children      'Check to see if there are children
  If pEnumGxObj Is Nothing Then
    'MsgBox "Nothing was found in the  " & pGxObj.FullName & " Folder"
    Exit Function
  Else
    Set fs = CreateObject("Scripting.FileSystemObject")
    Set pGxObj1 = pEnumGxObj.Next  ' For each object that is in the Folder,

    Do While Not pGxObj1 Is Nothing
      If pGxObj1.Category = "Folder" Then
        Set fs = CreateObject("Scripting.FileSystemObject")
        If fs.FileExists(log_filename) = False Then
          Set log = fs.CreateTextFile(log_filename, True)
        Else
          Set log = fs.OpenTextFile(log_filename, ForAppending)
```

```
            End If
           '  write line 3 ******************
            log.WriteLine ("<tr><td width=70%><b><img border=0 src=http://www.sci.tamucc.edu/~jwood/folder_closed.bmp width=16  " _
                  & "width=16 height=16><a href=#" & pGxObj1.FullName & "\   > " & pGxObj1.FullName _
                  & "\   " & "</a></b></td><td width=30%>" & pGxObj1.Category & "</td>")
            log.Close
            inFolder1 pGxObj1, log_filename, log
          End If
        Set pGxObj1 = pEnumGxObj.Next
        Loop
    End If
End Function
    Private Function SecondWrite(log_filename As String, sStart As String)
    Dim pGxApp As IGxApplication
    Dim pGxObjectCont As IGxObjectContainer
    Dim pEnumGxObj As IEnumGxObject
    Dim pGxObj As IGxObject
    Dim pGxObj1 As IGxObject
    Dim fs As FileSystemObject
    Dim log As TextStream
    Dim pcounter As Integer
    Dim nDir As Integer
    Dim fileCount As Integer
    Dim pGxCatalog As IGxCatalog

    Set pGxApp = Application
    Set pGxCatalog = pGxApp.Catalog
     On Error GoTo ENDSUB
    'sstart = InputBox("Enter where to start Write .HTML function. It can simply be a drive letter, eg. x:\", "Write Second HTML ", "D:\Projects\")
    pGxCatalog.Location = sStart

    Set pGxObj = pGxApp.SelectedObject
    Set pGxObjectCont = pGxObj
    Set pEnumGxObj = pGxObjectCont.Children      'Check to see if there are children

    If pEnumGxObj Is Nothing Then
       'MsgBox "Nothing was found in the  " & pGxObj.FullName & " Folder"
       Exit Function
    Else
       Set fs = CreateObject("Scripting.FileSystemObject")
       If fs.FileExists(log_filename) = False Then
          Set log = fs.CreateTextFile(log_filename, True)
       Else
         Set log = fs.OpenTextFile(log_filename, ForAppending)
       End If
       log.WriteLine ("<table border=0 cellpadding=0 cellspacing=0  bordercolor=#111111 width=100%>")
       'writeline 4 ********************
       log.WriteLine ("<tr><td width=41%><b><img border=0 src=http://www.sci.tamucc.edu/~jwood/folder_closed.bmp width=20 height=20><a name=" &
pGxObj.FullName & "\a><a href=#" & pGxObj.FullName & "\1>  " & pGxObj.FullName & "\   " & "</a></b></td><td width=59%></td>")
       log.Close

       Set pGxObj1 = pEnumGxObj.Next  ' For each object that is in the Folder

       Do While Not pGxObj1 Is Nothing
          If pGxObj1.Category = "Folder" Then
             inFolder2 pGxObj1, log_filename, log, fileCount, pcounter
          End If
          If Not pGxObj1.Category = "Folder" Then
             fileCount = fileCount + 1
             pcounter = pcounter + 1
          End If
          Set pGxObj1 = pEnumGxObj.Next
       Loop
    End If
    Set fs = CreateObject("Scripting.FileSystemObject")
    If fs.FileExists(log_filename) = False Then
      Set log = fs.CreateTextFile(log_filename, True)
    Else
      Set log = fs.OpenTextFile(log_filename, ForAppending)
    End If
```

```
    log.WriteLine ("<tr><td
width=41%>                 Folder contains " &
fileCount & " Spatial Dataset(s) </td><td width=59%></td>")
    log.WriteLine (" </tr></table><hr color=#000000 size=7>")
    log.Close
ENDSUB:
    'MsgBox "Error!"
 End Function
 Private Function inFolder2(ByVal pGxObj As IGxObject, log_filename As String, log As TextStream, fileCount As Integer, pcounter As Integer)

    Dim pGxObjectCont As IGxObjectContainer
    Dim pEnumGxObj As IEnumGxObject
    Dim pGxObj1  As IGxObject
    Dim fs As FileSystemObject

    Set pGxObjectCont = pGxObj
    Set pEnumGxObj = pGxObjectCont.Children      'Check to see if there are children
    'Set fs = CreateObject("Scripting.FileSystemObject")

    If pEnumGxObj Is Nothing Then
    'MsgBox "Nothing was found in the  " & pGxObj.FullName & " Folder"
    'Return
    Else
       Set fs = CreateObject("Scripting.FileSystemObject")
       If fs.FileExists(log_filename) = False Then
          Set log = fs.CreateTextFile(log_filename, True)
       Else
         Set log = fs.OpenTextFile(log_filename, ForAppending)
       End If
       If Not fileCount = 0 Then
          log.WriteLine ("<tr><td
width=41%>                  Folder contains " &
fileCount & " Spatial Dataset(s)</td><td width=59%></td>")
       Else
            'write line 6 ******************
          log.WriteLine ("<tr><td width=41%>       Folder contains NO Spatial Datasets </td><td
width=59%></td>")
       End If
       log.WriteLine ("<tr><td width=41%><img border=0 src=http://www.sci.tamucc.edu/~jwood/folder_closed.bmp width=16
height=16><b>           <a name=" & pGxObj.FullName & "\><a href=#" & pGxObj.FullName
& "\1> " & pGxObj.FullName & "\" & "</a></b></td><td width=59%>")
       log.Close
       fileCount = 0

       Set pGxObj1 = pEnumGxObj.Next  ' For each object that is in the Folder,

       Do While Not pGxObj1 Is Nothing

          If pGxObj1.Category = "Folder" Then
             inFolder2 pGxObj1, log_filename, log, fileCount, pcounter
          End If
          If Not pGxObj1.Category = "Folder" Then
             fileCount = fileCount + 1
             pcounter = pcounter + 1
          End If
          Set pGxObj1 = pEnumGxObj.Next
       Loop
    End If
End Function

Private Function Thirdwrite(log_filename As String, sStart As String)
'Purpose: For all data items contained in a folder, Create metatdata

    Dim pGxApp As IGxApplication
    Dim pGxObjectCont As IGxObjectContainer
    Dim pEnumGxObj As IEnumGxObject
    Dim pGxObj As IGxObject
    Dim pGxObj1 As IGxObject
    Dim fs As FileSystemObject
    Dim log As TextStream
```

```
    Dim pcounter As Integer
    Dim nDir As Integer
    Dim fileCount As Integer
    Dim pGxCatalog As IGxCatalog


    'sstart = InputBox("Enter where to start Write .HTML function. It can simply be a drive letter, eg. x:\", "Write Third HTML ", "D:\Projects\")
    On Error GoTo ENDSUB
    Set pGxApp = Application
    Set pGxCatalog = pGxApp.Catalog
    pGxCatalog.Location = sStart


    Set pGxObj = pGxApp.SelectedObject
    If Not TypeOf pGxObj Is IGxFolder Then
       MsgBox "Please select a Folder", vbExclamation
       Exit Function
    Else
       Set pGxObjectCont = pGxObj
       Set pEnumGxObj = pGxObjectCont.Children      'Check to see if there are children

       If pEnumGxObj Is Nothing Then
          'MsgBox "Nothing was found in the  " & pGxObj.FullName & " Folder"
          Exit Function
       Else
          Set fs = CreateObject("Scripting.FileSystemObject")
          If fs.FileExists(log_filename) = False Then
             Set log = fs.CreateTextFile(log_filename, True)
          Else
             Set log = fs.OpenTextFile(log_filename, ForAppending)
          End If
          log.WriteLine ("<table border=0 cellpadding=0 cellspacing=0  bordercolor=#111111 width=100%>")
           '*************write line 8
          log.WriteLine ("<tr><td width=60%><b><img border=0 src=http://www.sci.tamucc.edu/~jwood/folder_closed.bmp width=16 height=16><a name=" &
pGxObj.FullName & "\1>   " & pGxObj.FullName & "\    " & "</b></td><td width=59%>" & pGxObj.Category & "</td>")
          log.Close

          Set pGxObj1 = pEnumGxObj.Next  ' For each object that is in the Folder

          Do While Not pGxObj1 Is Nothing
             If pGxObj1.Category = "Folder" Then
                Set fs = CreateObject("Scripting.FileSystemObject")
                If fs.FileExists(log_filename) = False Then
                   Set log = fs.CreateTextFile(log_filename, True)
                Else
                  Set log = fs.OpenTextFile(log_filename, ForAppending)
                End If
                log.WriteLine (" ")
                   '********write line 9
                log.WriteLine ("<tr><td width=47%><b><img border=0 src=http://sci.tamucc.edu/~jwood/folder_closed.bmp width=16 height=16><a name =" &
pGxObj1.FullName & "\1>       " & pGxObj1.FullName & "\" & "</td><td width=53%>" & pGxObj1.Category
& "</b></td></tr>")
                log.Close
                inFolder3 pGxObj1, log_filename, log
             Else
                Set fs = CreateObject("Scripting.FileSystemObject")
                If fs.FileExists(log_filename) = False Then
                   Set log = fs.CreateTextFile(log_filename, True)
                Else
                   Set log = fs.OpenTextFile(log_filename, ForAppending)
                End If
                   '************* writeline 10
                Dim replaceme As String
                replaceme = Replace(pGxObj1.FullName, "/", "")
                replaceme = Replace(replaceme, "\", "")
                replaceme = Replace(replaceme, ".", "")
                log.WriteLine ("<tr><td width=41%><a name=" & pGxObj1.FullName & "\1><A HREF=directory_2.html#" & replaceme & "abstract
TARGET=ABSTRACT ONCLICK=window.open('directory_2.html#" & replaceme & "', 'ABSTRACT', 'width=30,height=25'); return
false><b>          " & pGxObj1.FullName & " </b></font></td><td width=59%>  " & pGxObj1.Category &
"</td></tr>")
```

```
            log.Close
          End If
          Set pGxObj1 = pEnumGxObj.Next
        Loop
      End If
      Set fs = CreateObject("Scripting.FileSystemObject")
      If fs.FileExists(log_filename) = False Then
        Set log = fs.CreateTextFile(log_filename, True)
      Else
        Set log = fs.OpenTextFile(log_filename, ForAppending)
      End If
      log.WriteLine ("</tr></table><hr color=#000000 size=7><table border=0 cellpadding=0 cellspacing=0 bordercolor=#111111 width=100%>")
      log.Close
  End If
ENDSUB:
  'MsgBox "Error!"
  End Function
  Private Function inFolder3(ByVal pGxObj As IGxObject, log_filename As String, log As TextStream)

  Dim pGxObjectCont As IGxObjectContainer
  Dim pEnumGxObj As IEnumGxObject
  Dim pGxObj1 As IGxObject
  Dim fs As FileSystemObject
  Dim pMD As IMetadata
  Dim pPS As IPropertySet
  Dim pXPS As IXmlPropertySet
  Dim strSimple As String
  Dim replaceme As String

  Set pGxObjectCont = pGxObj
  Set pEnumGxObj = pGxObjectCont.Children      'Check to see if there are children

  If pEnumGxObj Is Nothing Then
     'MsgBox "Nothing was found in the  " & pGxObj.FullName & " Folder"
     'Exit Sub
  Else

  Set pGxObj1 = pEnumGxObj.Next  ' For each object that is in the Folder,
  Do While Not pGxObj1 Is Nothing
    If pGxObj1.Category = "Folder" Then
       replaceme = Replace(pGxObj1.FullName, "/", "")
       replaceme = Replace(replaceme, "\", "")
       replaceme = Replace(replaceme, ".", "")
       Set fs = CreateObject("Scripting.FileSystemObject")
       If fs.FileExists(log_filename) = False Then
          Set log = fs.CreateTextFile(log_filename, True)
       Else
          Set log = fs.OpenTextFile(log_filename, ForAppending)
       End If
       'WriteLine 11
       log.WriteLine ("<tr><td width=47%><b><img border=0 src=http://sci.tamucc.edu/~jwood/folder_closed.bmp width=16 height=16><a name =" &
pGxObj1.FullName & "\1>          " & pGxObj1.FullName & "\" & "</td><td width=53%>"
& pGxObj1.Category & "</b></td></tr>")
       log.Close
       inFolder3 pGxObj1, log_filename, log
    Else
       Set fs = CreateObject("Scripting.FileSystemObject")
       If fs.FileExists(log_filename) = False Then
          Set log = fs.CreateTextFile(log_filename, True)
       Else
          Set log = fs.OpenTextFile(log_filename, ForAppending)
       End If
       replaceme = Replace(pGxObj1.FullName, "/", "")
       replaceme = Replace(replaceme, "\", "")
       replaceme = Replace(replaceme, ".", "")
       ' writeline 12
       log.WriteLine ("<tr><td width=41%><a name=" & pGxObj1.FullName & "\1><A HREF=directory_2.html#" & replaceme & "abstract
TARGET=ABSTRACT ONCLICK=window.open('directory_2.html#" & replaceme & "', 'ABSTRACT', 'width=50,height=25'); return
false><b>          " & pGxObj1.FullName & " </b></font></td><td width=59%>  " & pGxObj1.Category &
"</td></tr>")
```

```
            log.Close
        End If
        Set pGxObj1 = pEnumGxObj.Next
    Loop
    End If
End Function
Public Function fourthwrite(log_filename2 As String, sStart As String)
'Purpose: For all data items contained in a folder, Create metatdata
    Dim pGxApp As IGxApplication
    Dim pGxObjectCont As IGxObjectContainer
    Dim pEnumGxObj As IEnumGxObject
    Dim pGxObj As IGxObject
    Dim pGxObj1 As IGxObject
    Dim fs As FileSystemObject
    Dim log As TextStream
    Dim pcounter As Integer
    Dim pMD As IMetadata
    Dim pPS As IPropertySet
    Dim pXPS As IXmlPropertySet
    Dim strSimple As String

    Dim pGxCatalog As IGxCatalog


   ' sstart = InputBox("Enter where to start Write .HTML function. It can simply be a drive letter, eg. x:\", "Write Fourth HTML ", "D:\Projects\")
    On Error GoTo ENDSUB
    Set pGxApp = Application
    Set pGxCatalog = pGxApp.Catalog
    pGxCatalog.Location = sStart


    Set pGxObj = pGxApp.SelectedObject

    If Not TypeOf pGxObj Is IGxFolder Then
        MsgBox "Please select a Folder", vbExclamation
        Exit Function
    Else
        Set pGxObjectCont = pGxObj
        Set pEnumGxObj = pGxObjectCont.Children      'Check to see if there are children

        If pEnumGxObj Is Nothing Then
           'MsgBox "Nothing was found in the  " & pGxObj.FullName & " Folder"
            Exit Function
        Else
            Set fs = CreateObject("Scripting.FileSystemObject")
            If fs.FileExists(log_filename2) = False Then
                Set log = fs.CreateTextFile(log_filename2, True)
            Else
                Set log = fs.OpenTextFile(log_filename2, ForAppending)
            End If
            log.WriteLine ("<table border=0 cellpadding=0 cellspacing=0  bordercolor=#111111 width=100%>")
            '************write line 13
            log.WriteLine ("<tr><td width=60%><b><img border=0 src=http://sci.tamucc.edu/~jwood/folder_closed.bmp width=16 height=16><a name=" &
pGxObj.FullName & "\1>   " & pGxObj.FullName & "\  " & "</b></td><td width=59%>" & pGxObj.Category & "</td>")
            log.Close

            Set pGxObj1 = pEnumGxObj.Next  ' For each object that is in the Folder

            Do While Not pGxObj1 Is Nothing
                If pGxObj1.Category = "Folder" Then
                    If fs.FileExists(log_filename2) = False Then
                        Set log = fs.CreateTextFile(log_filename2, True)
                    Else
                        Set log = fs.OpenTextFile(log_filename2, ForAppending)
                    End If
                    log.WriteLine (" ")
                    '********write line 14
                    log.WriteLine ("<tr><td width=47%><b><img border=0 src=http://sci.tamucc.edu/~jwood/folder_closed.bmp width=16 height=16><a name ="
& pGxObj1.FullName & "\1>         " & pGxObj1.FullName & "/" & " </td><td width=53%>" &
pGxObj1.Category & "</b></td></tr>")
```

```
                log.Close
                inFolder4 pGxObj1, log_filename2, log, pcounter
            Else
                Set fs = CreateObject("Scripting.FileSystemObject")
                If fs.FileExists(log_filename2) = False Then
                    Set log = fs.CreateTextFile(log_filename2, True)
                Else

                    Set log = fs.OpenTextFile(log_filename2, ForAppending)
                End If
                Dim replaceme As String
                replaceme = Replace(pGxObj1.FullName, "/", "")
                replaceme = Replace(replaceme, "\", "")
                replaceme = Replace(replaceme, ".", "")
                'writeline 15**********************
                log.WriteLine ("<tr><td width=41%><a name=" & replaceme & "abstract ><font
color=#008080><b>        " & pGxObj1.FullName & " </b></font></td><td width=59%>  " &
pGxObj1.Category & "</td></tr>")
                If Not pGxObj1.Category = "Personal Geodatabase" Then
                   Set pMD = pGxObj1
                   Set pPS = pMD.Metadata
                   Set pXPS = pPS
                   strSimple = pXPS.SimpleGetProperty("idinfo/descript/abstract")
                '  MsgBox strSimple     'Name further specify an an element from a list specifying the attribute value, "[@]"
                   If strSimple = "" Then
                     strSimple = "<font color=#FF0000>No Abstract Available </font>"
                     pcounter = pcounter + 1
                   End If
                   If Left(strSimple, 8) = "REQUIRED" Then
                     strSimple = "<font color=#FF0000>No Abstract Available </font>"
                     pcounter = pcounter + 1
                   End If
                   'write line 16 ************************
                   log.WriteLine ("<i><tr><td width=47%>        " & strSimple & " </tr></td></i>")
                Else
                  'write line 17 ************************
                  log.WriteLine ("<i><tr><td width=47%>        No Metadata Shown for Personal
Geodatabases</tr></td></i>")
                End If
                log.Close
            End If
            Set pGxObj1 = pEnumGxObj.Next
        Loop
        Set fs = CreateObject("Scripting.FileSystemObject")
        Set log = fs.OpenTextFile(log_filename2, ForAppending)
        log.WriteLine ("<i><tr><td width=47%>        Number of files w/o metadata = " & pcounter & "
</tr></td></i>")
        log.Close
      End If
   End If
ENDSUB:
   'MsgBox "Error!!!!"
  End Function
  Private Function inFolder4(ByVal pGxObj As IGxObject, log_filename2 As String, log As TextStream, pcounter As Integer)

  Dim pGxObjectCont As IGxObjectContainer
  Dim pEnumGxObj As IEnumGxObject
  Dim pGxObj1 As IGxObject
  Dim fs As FileSystemObject
  Dim pMD As IMetadata
  Dim pPS As IPropertySet
  Dim pXPS As IXmlPropertySet
  Dim strSimple As String


  Set pGxObjectCont = pGxObj
  Set pEnumGxObj = pGxObjectCont.Children      'Check to see if there are children
  Set fs = CreateObject("Scripting.FileSystemObject")

  If pEnumGxObj Is Nothing Then
```

```
        'MsgBox "Nothing was found in the  " & pGxObj.FullName & " Folder"
        'Exit Sub

    Else

    Set pGxObj1 = pEnumGxObj.Next  ' For each object that is in the Folder,
    Do While Not pGxObj1 Is Nothing
    Set log = fs.OpenTextFile(log_filename2, ForAppending)
        If pGxObj1.Category = "Folder" Then

            '*************writeline 18
            log.WriteLine ("<tr><td width=47%><b><img border=0 src=http://sci.tamucc.edu/~jwood/folder_closed.bmp width=16 height=16><a name =" &
pGxObj1.FullName & "\1>         " & pGxObj1.FullName & "\" & " </td><td width=53%>" &
pGxObj1.Category & "</b></td></tr>")
            log.Close
            inFolder4 pGxObj1, log_filename2, log, pcounter
        Else
            Dim replaceme As String
            replaceme = Replace(pGxObj1.FullName, "/", "")
            replaceme = Replace(replaceme, "\", "")
            replaceme = Replace(replaceme, ".", "")
            ' writeline 19 ************************
            log.WriteLine ("<tr><td width=41%><a name=" & replaceme & "abstract ><font
color=#008080><b>        " & pGxObj1.FullName & "  </b></font> </td><td width=59%>  " &
pGxObj1.Category & "</td></tr>")
            log.Close
            If Not pGxObj1.Category = "Personal Geodatabase" Then
                Set pMD = pGxObj1
                Set pPS = pMD.Metadata
                Set pXPS = pPS
                strSimple = pXPS.SimpleGetProperty("idinfo/descript/abstract")
                '  MsgBox strSimple     'Name further specify an an element from a list specifying the attribute value, "[@]"
                If strSimple = "" Then
                    strSimple = "<font color=#FF0000>No Abstract Available </font>"
                    pcounter = pcounter + 1
                End If
                If Left(strSimple, 8) = "REQUIRED" Then
                    strSimple = "<font color=#FF0000>No Abstract Available </font>"
                End If
                Set log = fs.OpenTextFile(log_filename2, ForAppending)
                log.WriteLine ("<i><tr><td width=47%>        " & strSimple & " </tr></td></i>")
                log.Close
            Else
                Set log = fs.OpenTextFile(log_filename2, ForAppending)
                log.WriteLine ("<i><tr><td width=47%>        No Metadata Shown for Personal
Geodatabases</tr></td></i>")
                log.Close
            End If

        End If
        Set pGxObj1 = pEnumGxObj.Next
    Loop
    End If
End Function
Private Function listDirectories(colDirs As Collection, sStart As String)
    Dim Directory As String

    Directory = InputBox("Enter the name of the directory you wish to write an html directory for.  ", "List Directories", sStart)
    While Len(Directory) <> 0
        colDirs.Add Directory
        Directory = InputBox("Enter the name of the directory you wish to write an html directory for.  **Cancel if done selecting**!", "List Directories", sStart)
    Wend
End Function
```

About the Author:

John S. Wood

John S. Wood is an Adjunct Professor of Geographic Information Sciences, and a Research Associate (GIS) for Harte Research Institute for Gulf of Mexico Studies. His research interests include the use of spatial information in environmental applications and research, coastal and wetlands management, habitat conservation, site selection and planning, and GIS education.

Texas A&M University-Corpus Christi