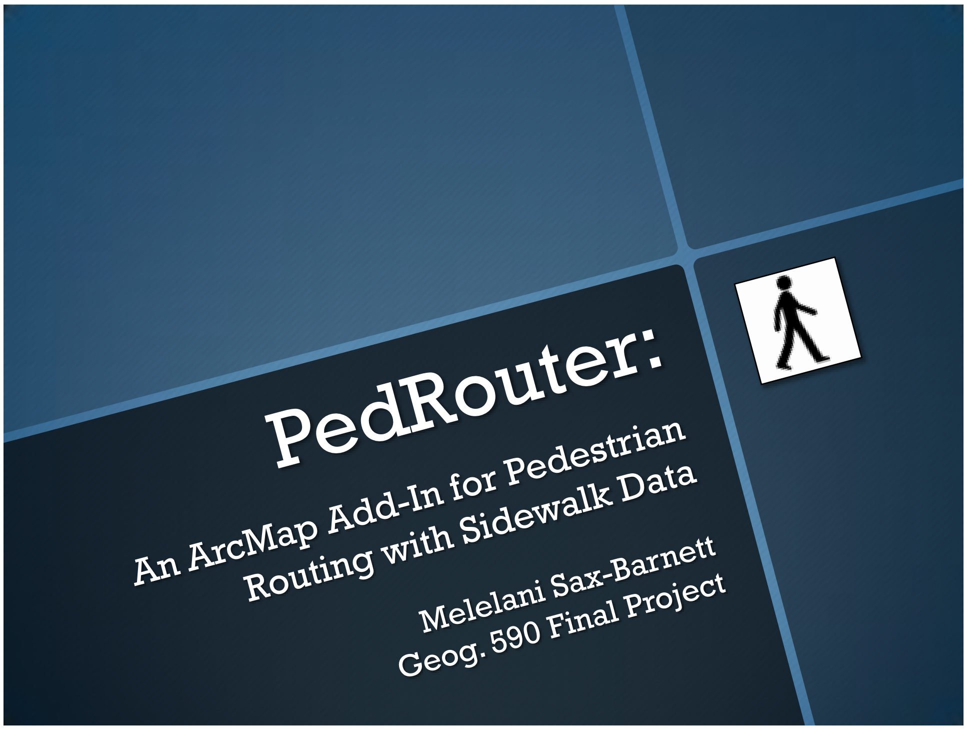
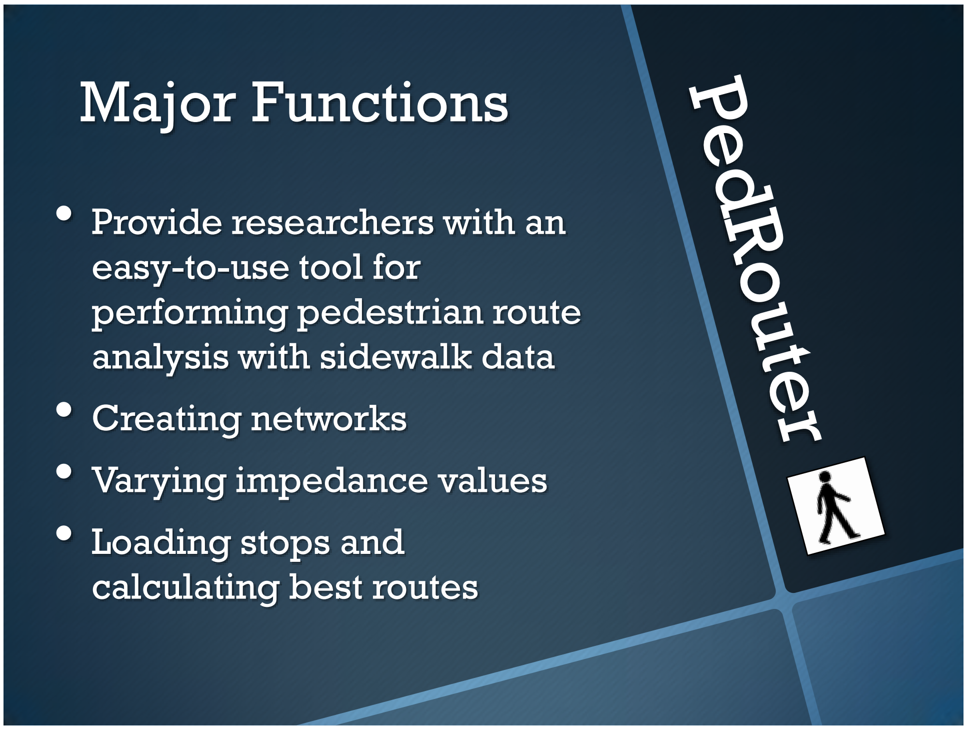


PedRouter:

An ArcMap Add-In for Pedestrian Routing with Sidewalk Data

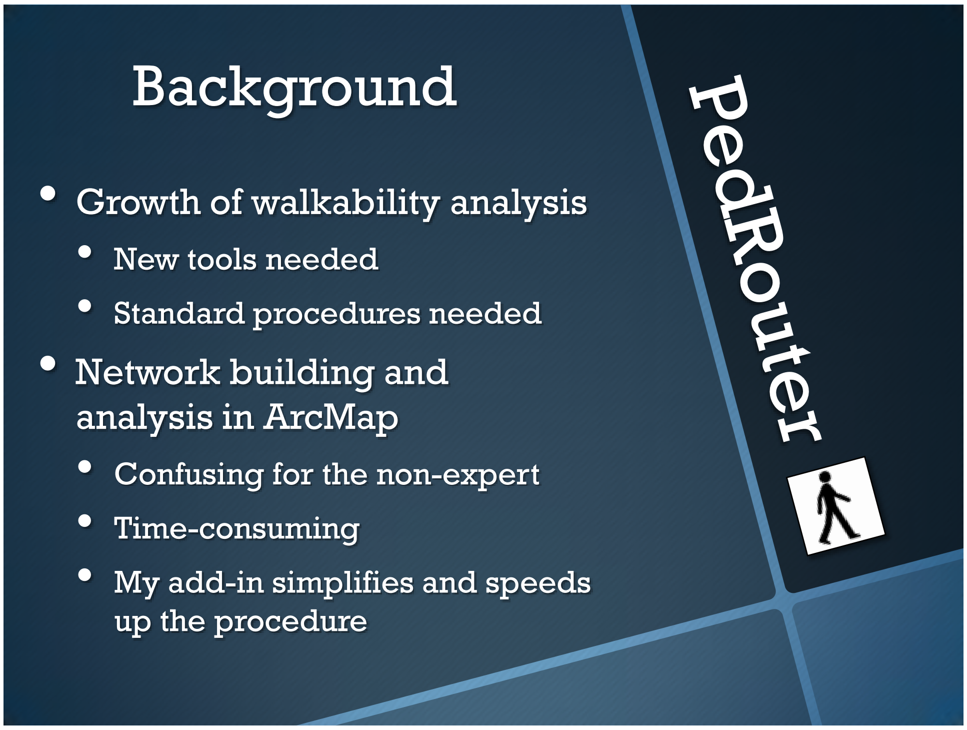
Melelani Sax-Barnett
Geog. 590 Final Project



Major Functions

- Provide researchers with an easy-to-use tool for performing pedestrian route analysis with sidewalk data
- Creating networks
- Varying impedance values
- Loading stops and calculating best routes

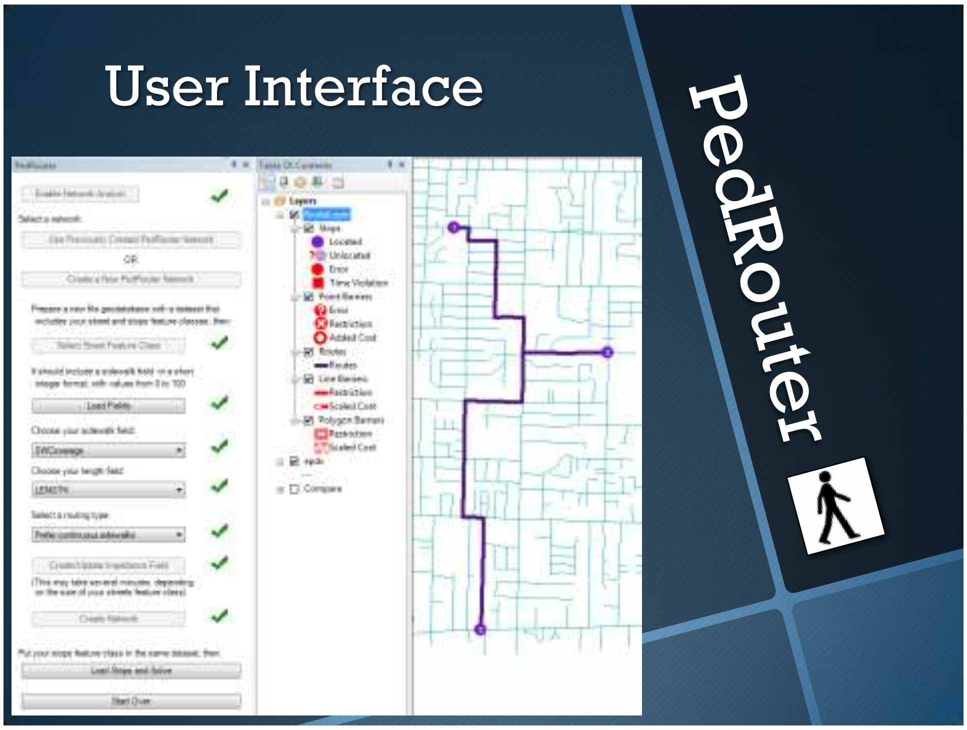
PedRouter



Background

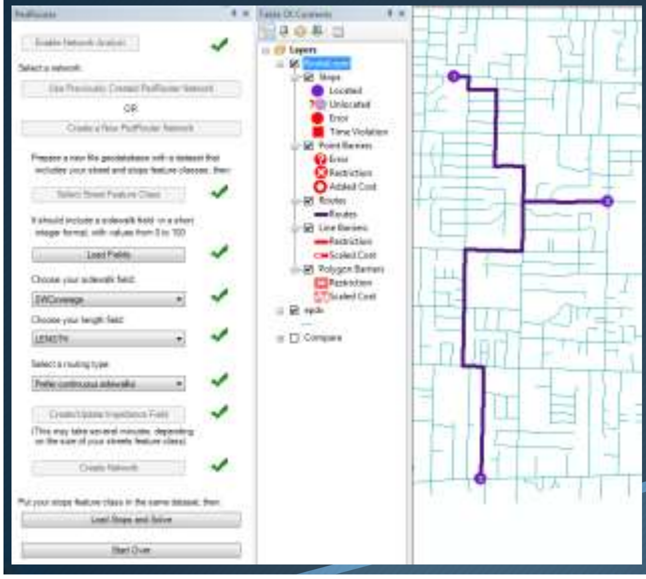
PedRouter

- Growth of walkability analysis
  - New tools needed
  - Standard procedures needed
- Network building and analysis in ArcMap
  - Confusing for the non-expert
  - Time-consuming
  - My add-in simplifies and speeds up the procedure



User Interface

PedRouter



PedRouter

Enable Network Analyst ✓

Select a network:

Use Previously Created PedRouter Network

OR

Create a New PedRouter Network

Prepare a new file geodatabase with a dataset that includes your street and stops feature classes, then:

Select Street Feature Class ✓

It should include a sidewalk field in a short integer format, with values from 0 to 100

Load Fields ✓

Choose your sidewalk field:

SWCoverage ✓

Choose your length field:

LENGTH ✓

Select a routing type:

Prefer continuous sidewalks ✓

Create/Update Impedance Field ✓

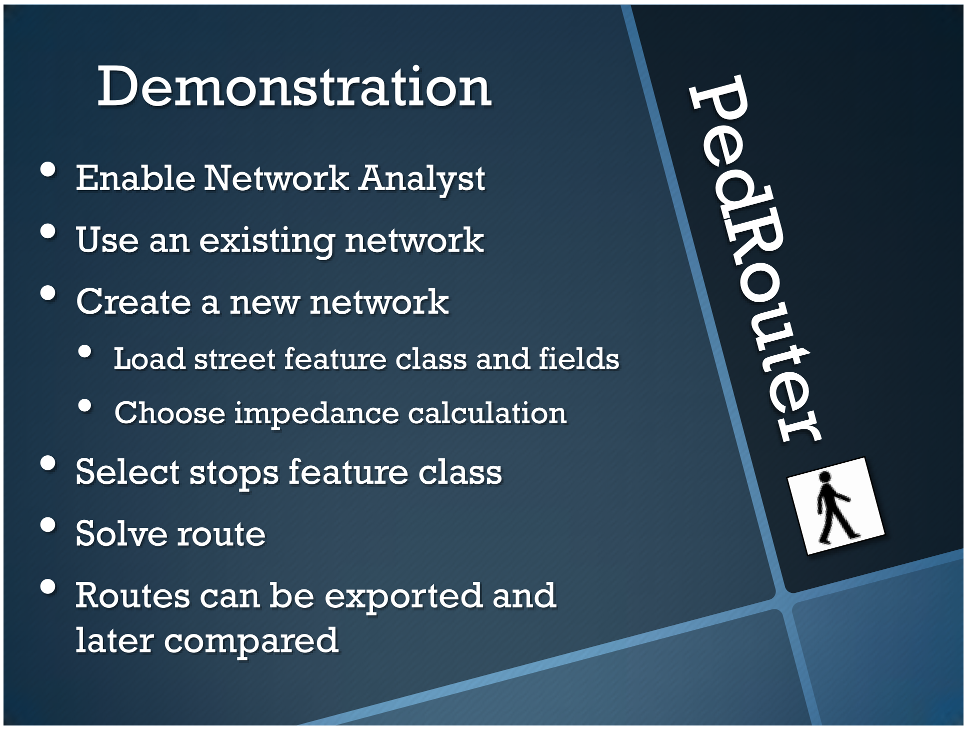
(This may take several minutes, depending on the size of your streets feature class)

Create Network ✓

Put your stops feature class in the same dataset, then:

Load Stops and Solve

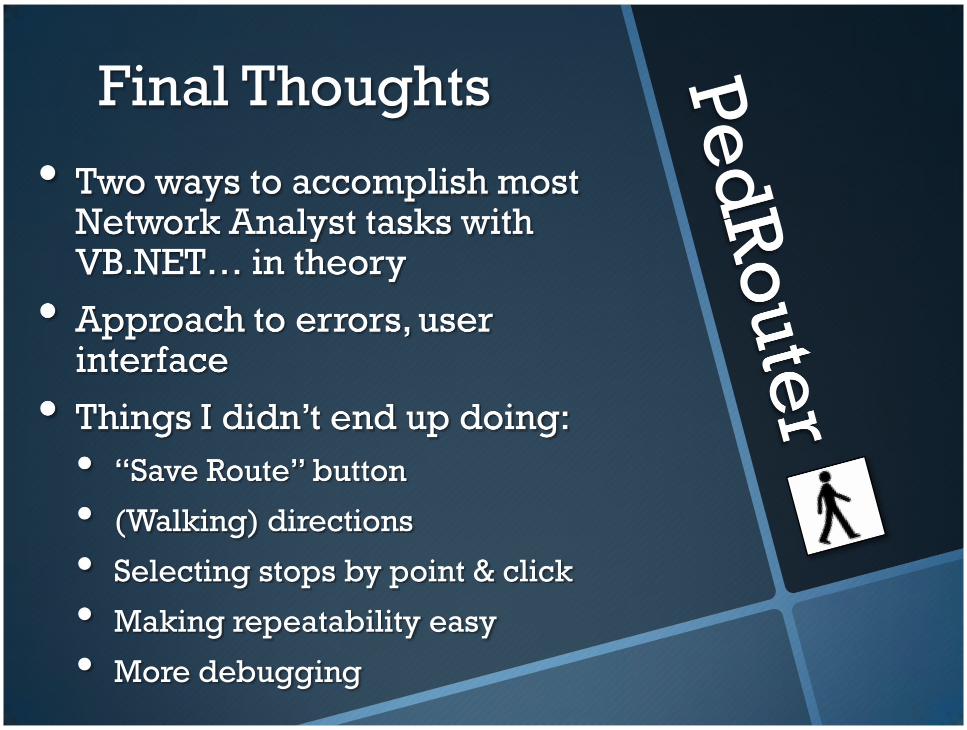
Start Over



Demonstration

- Enable Network Analyst
- Use an existing network
- Create a new network
  - Load street feature class and fields
  - Choose impedance calculation
- Select stops feature class
- Solve route
- Routes can be exported and later compared

PedRouter



Final Thoughts

- Two ways to accomplish most Network Analyst tasks with VB.NET… in theory
- Approach to errors, user interface
- Things I didn't end up doing:
  - "Save Route" button
  - (Walking) directions
  - Selecting stops by point & click
  - Making repeatability easy
  - More debugging

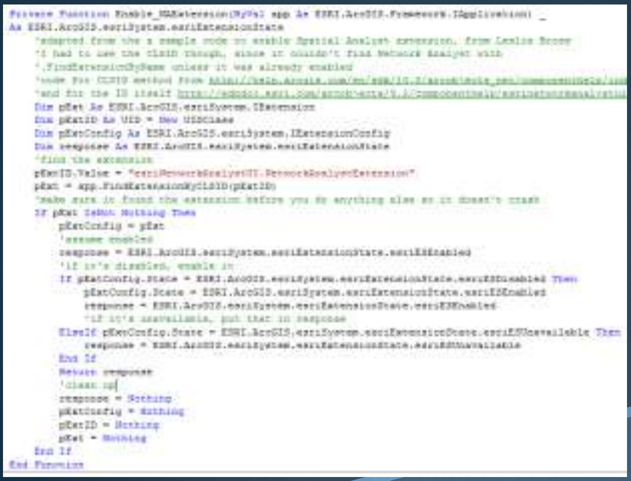
PedRouter

Questions?

The Code

PedRouter

- Enabling Network Analyst

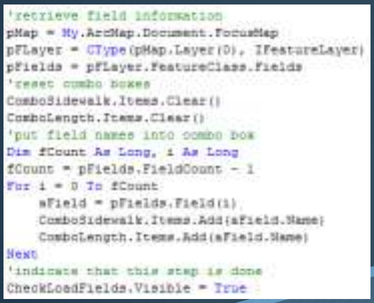


```
Private Function Enable_NAExtension(ByVal app As ESRI.ArcGIS.Framework.IApplication) _
As ESRI.ArcGIS.esriSystem.esriExtensionState
    'adapted from the a sample code to enable Spatial Analyst extension, from Leslie Bross
    'I had to use the CLSID though, since it couldn't find Network Analyst with
    '.FindExtensionByName unless it was already enabled
    'code for CLSID method from [illegible]
    'and for the ID itself [illegible]
    Dim pExt As ESRI.ArcGIS.esriSystem.IExtension
    Dim pExtID As UID = New UIDClass
    Dim pExtConfig As ESRI.ArcGIS.esriSystem.IExtensionConfig
    Dim response As ESRI.ArcGIS.esriSystem.esriExtensionState
    'find the extension
    pExtID.Value = "esriNetworkAnalystUI.NetworkAnalystExtension"
    pExt = app.FindExtensionByCLSID(pExtID)
    'make sure it found the extension before you do anything else so it doesn't crash
    If pExt IsNot Nothing Then
        pExtConfig = pExt
        'assume enabled
        response = ESRI.ArcGIS.esriSystem.esriExtensionState.esriESEnabled
        'if it's disabled, enable it
        If pExtConfig.State = ESRI.ArcGIS.esriSystem.esriExtensionState.esriESDisabled Then
            pExtConfig.State = ESRI.ArcGIS.esriSystem.esriExtensionState.esriESEnabled
            response = ESRI.ArcGIS.esriSystem.esriExtensionState.esriESEnabled
            'if it's unavailable, put that in response
        ElseIf pExtConfig.State = ESRI.ArcGIS.esriSystem.esriExtensionState.esriESUnavailable Then
            response = ESRI.ArcGIS.esriSystem.esriExtensionState.esriESUnavailable
        End If
        Return response
        'clean up
        response = Nothing
        pExtConfig = Nothing
        pExtID = Nothing
        pExt = Nothing
    End If
End Function
```

The Code

PedRouter

- Creating a new network
  - Selecting streets feature class: GxDialog with FGDB feature class filter and polyline check
  - Loading fields, based on Lab 4:

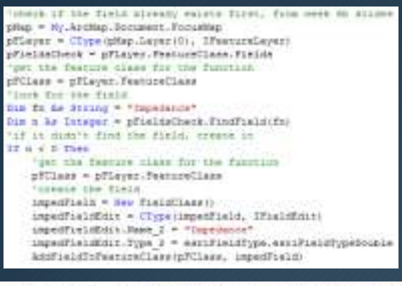


```
'retrieve field information
pMap = My.ArcMap.Document.FocusMap
pFLayer = CType(pMap.Layer(0), IFeatureLayer)
pFields = pFLayer.FeatureClass.Fields
'reset combo boxes
ComboSidewalk.Items.Clear()
ComboLength.Items.Clear()
'put field names into combo box
Dim fCount As Long, i As Long
fCount = pFields.FieldCount - 1
For i = 0 To fCount
    aField = pFields.Field(i)
    ComboSidewalk.Items.Add(aField.Name)
    ComboLength.Items.Add(aField.Name)
Next
'indicate that this step is done
CheckLoadFields.Visible = True
```

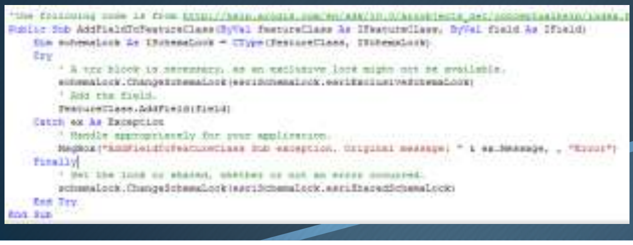
The Code

PedRouter

- Creating a new field



```
'check if the field already exists first, [illegible]
pMap = My.ArcMap.Document.FocusMap
pFLayer = CType(pMap.Layer(0), IFeatureLayer)
pFieldsCheck = pFLayer.FeatureClass.Fields
'get the feature class for the function
pFClass = pFLayer.FeatureClass
'look for the field
Dim fn As String = "Impedance"
Dim n As Integer = pFieldsCheck.FindField(fn)
'if it didn't find the field, create it
If n < 0 Then
    'get the feature class for the function
    pFClass = pFLayer.FeatureClass
    'create the field
    impedField = New FieldClass()
    impedFieldEdit = CType(impedField, IFieldEdit)
    impedFieldEdit.Name_2 = "Impedance"
    impedFieldEdit.Type_2 = esriFieldType.esriFieldTypeDouble
    AddFieldToFeatureClass(pFClass, impedField)
```

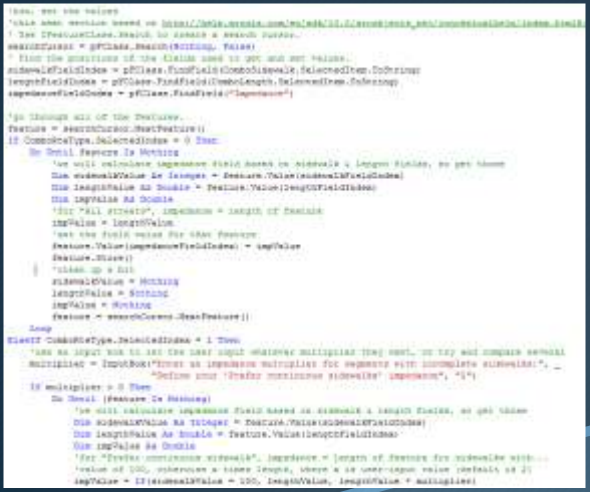


```
'the following code is from http://help.arcgis.com/en/sdk/10.0/arcobjects_net/conceptualhelp/index.h
Public Sub AddFieldToFeatureClass(ByVal featureClass As IFeatureClass, ByVal field As IField)
    Dim schemaLock As ISchemaLock = CType(featureClass, ISchemaLock)
    Try
        ' A try block is necessary, as an exclusive lock might not be available.
        schemaLock.ChangeSchemaLock(esriSchemaLock.esriExclusiveSchemaLock)
        ' Add the field.
        featureClass.AddField(field)
    Catch ex As Exception
        ' Handle appropriately for your application.
        MsgBox("AddFieldToFeatureClass Sub exception. Original message: " & ex.Message, , "Error")
    Finally
        ' Set the lock to shared, whether or not an error occurred.
        schemaLock.ChangeSchemaLock(esriSchemaLock.esriSharedSchemaLock)
    End Try
End Sub
```

The Code

PedRouter

- Updating field values

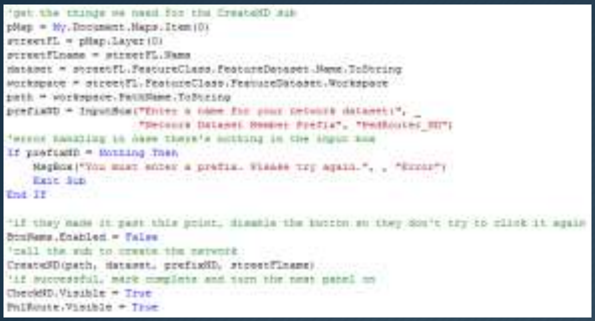


```
'now, set the values
'this next section based on http://help.arcgis.com/en/sdk/10.0/arcobjects_net/conceptualhelp/index.html#
' Use IFeatureClass.Search to create a search cursor.
searchCursor = pFClass.Search(Nothing, False)
' Find the positions of the fields used to get and set values.
sidewalkFieldIndex = pFClass.FindField(ComboSidewalk.SelectedItem.ToString)
lengthFieldIndex = pFClass.FindField(ComboLength.SelectedItem.ToString)
impedanceFieldIndex = pFClass.FindField("Impedance")

'go through all of the features...
feature = searchCursor.NextFeature()
If ComboRteType.SelectedIndex = 0 Then
    Do Until feature Is Nothing
        'we will calculate impedance field based on sidewalk & length fields, so get those
        Dim sidewalkValue As Integer = feature.Value(sidewalkFieldIndex)
        Dim lengthValue As Double = feature.Value(lengthFieldIndex)
        Dim impValue As Double
        'for "All streets", impedance = length of feature
        impValue = lengthValue
        'set the field value for that feature
        feature.Value(impedanceFieldIndex) = impValue
        feature.Store()
        'clean up a bit
        sidewalkValue = Nothing
        lengthValue = Nothing
        impValue = Nothing
        feature = searchCursor.NextFeature()
    Loop
ElseIf ComboRteType.SelectedIndex = 1 Then
    'use an input box to let the user input whatever multiplier they want.. or try and compare several
    multiplier = InputBox("Enter an impedance multiplier for segments with incomplete sidewalks:", _
                 "Define your 'Prefer continuous sidewalks' impedance", "2")
    If multiplier > 0 Then
        Do Until (feature Is Nothing)
            'we will calculate impedance field based on sidewalk & length fields, so get those
            Dim sidewalkValue As Integer = feature.Value(sidewalkFieldIndex)
            Dim lengthValue As Double = feature.Value(lengthFieldIndex)
            Dim impValue As Double
            'for "Prefer continuous sidewalk", impedance = length of feature for sidewalks with...
            'value of 100, otherwise n times length, where n is user-input value (default of 2)
            impValue = IIf(sidewalkValue = 100, lengthValue, lengthValue * multiplier)
```

The Code

PedRouter

- Creating a Network Dataset



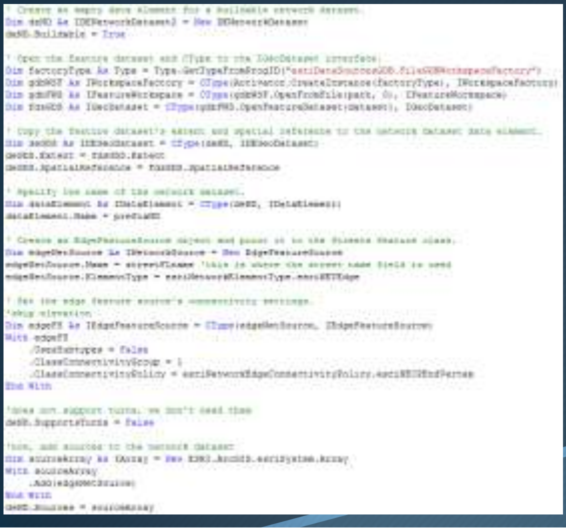
```
'get the things we need for the CreateND sub
pMap = My.Document.Maps.Item(0)
streetFL = pMap.Layer(0)
streetFLname = streetFL.Name
dataset = streetFL.FeatureClass.FeatureDataset.Name.ToString
workspace = streetFL.FeatureClass.FeatureDataset.Workspace
path = workspace.PathName.ToString
prefixND = InputBox("Enter a name for your network dataset:", _
                    "Network Dataset Name Prefix", "PedRouter_ND")
'error handling in case there's nothing in the input box
If prefixND = Nothing Then
    MsgBox("You must enter a prefix. Please try again.", , "Error")
    Exit Sub
End If

'if they made it past this point, disable the button so they don't try to click it again
BtnName.Enabled = False
'call the sub to create the network
CreateND(path, dataset, prefixND, streetFLname)
'if successful, mark complete and turn the next panel on
CheckND.Visible = True
PnlRoute.Visible = True
```

- CreateND sub adapted from:
  http://help.arcgis.com/en/sdk/10.0/arcobjects_net/conceptualhelp/index.html#/How_to_create_a_network_dataset/0001000000w7000000/

The Code

PedRouter

- Creating a Network Dataset

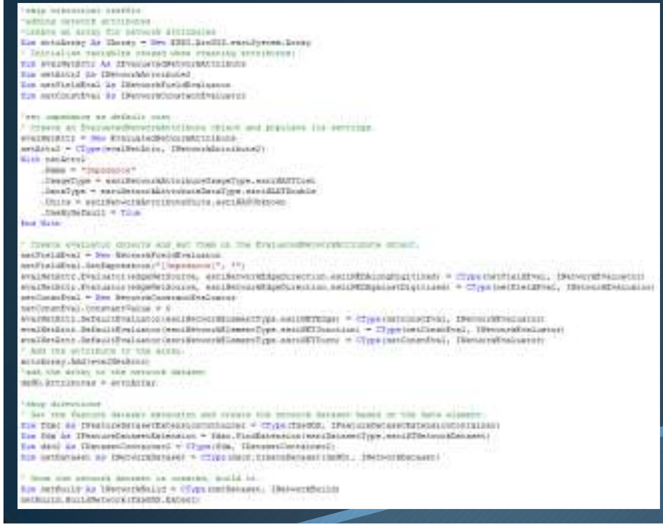


```
' Create an empty data element for a buildable network dataset.
Dim deND As IDENetworkDataset2 = New DENetworkDataset
deND.Buildable = True

' Open the feature dataset and CType to the IGeoDataset interface.
Dim factoryType As Type = Type.GetTypeFromProgID("esriDataSourcesGDB.FileGDBWorkspaceFactory")
Dim gdbWSF As IWorkspaceFactory = CType(Activator.CreateInstance(factoryType), IWorkspaceFactory)
Dim gdbFWS As IFeatureWorkspace = CType(gdbWSF.OpenFromFile(path, 0), IFeatureWorkspace)
Dim fdsGDS As IGeoDataset = CType(gdbFWS.OpenFeatureDataset(dataset), IGeoDataset)

' Copy the feature dataset's extent and spatial reference to the network dataset data element.
Dim deGDS As IDEGeoDataset = CType(deND, IDEGeoDataset)
deGDS.Extent = fdsGDS.Extent
deGDS.SpatialReference = fdsGDS.SpatialReference

' Specify the name of the network dataset.
Dim dataElement As IDataElement = CType(deND, IDataElement)
dataElement.Name = [illegible]

' Create an EdgeFeatureSource object and point it to the Streets feature class.
Dim edgeNetSource As INetworkSource = New EdgeFeatureSource
edgeNetSource.Name = streetFLname 'this is where the street name field is used
edgeNetSource.ElementType = esriNetworkElementType.esriNETEdge

' Set the edge feature source's connectivity settings.
'skip elevation
Dim edgeFS As IEdgeFeatureSource = CType(edgeNetSource, IEdgeFeatureSource)
With edgeFS
    .UsesSubtypes = False
    .ClassConnectivityGroup = 1
    .ClassConnectivityPolicy = esriNetworkEdgeConnectivityPolicy.esriNECPEndVertex
End With

'does not support turns, we don't need them
deND.SupportsTurns = False

'now, add sources to the network dataset
Dim sourceArray As IArray = New ESRI.ArcGIS.esriSystem.Array
With sourceArray
    .Add(edgeNetSource)
End With
deND.Sources = sourceArray
```

The Code

PedRouter

- Creating a Network Dataset

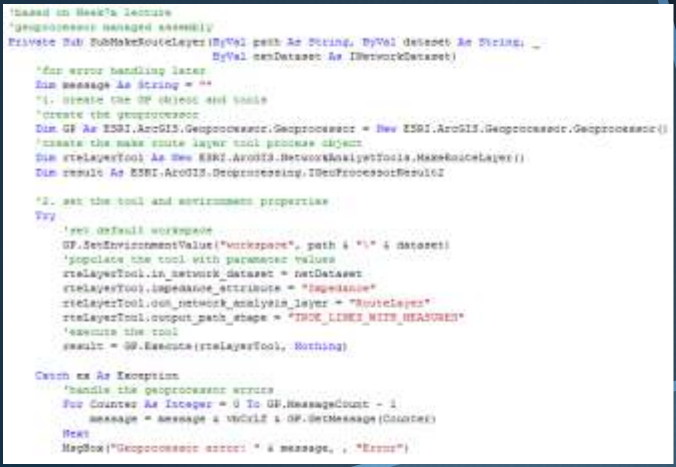


```
'skip [illegible]
'adding network attributes
'create an array for network attributes
Dim attributeArray As IArray = New ESRI.ArcGIS.esriSystem.Array
' Initialize variables reused when creating attributes.
Dim evalNetAttr As IEvaluatedNetworkAttribute
Dim netAttr2 As INetworkAttribute2
Dim netFieldEval As INetworkFieldEvaluator
Dim netConstEval As INetworkConstantEvaluator

'set impedance as default cost
' Create an EvaluatedNetworkAttribute object and populate its settings.
evalNetAttr = New EvaluatedNetworkAttribute
netAttr2 = CType(evalNetAttr, INetworkAttribute2)
With netAttr2
    .Name = "Impedance"
    .UsageType = esriNetworkAttributeUsageType.esriNAUTCost
    .DataType = esriNetworkAttributeDataType.esriNADTDouble
    .Units = esriNetworkAttributeUnits.esriNAUUnknown
    .UseByDefault = True
End With

' Create evaluator objects and set them on the EvaluatedNetworkAttribute object.
netFieldEval = New NetworkFieldEvaluator
netFieldEval.SetExpression("[Impedance]", "")
evalNetAttr.Evaluator(edgeNetSource, esriNetworkEdgeDirection.esriNEDAlongDigitized) = CType(netFieldEval, INetworkEvaluator)
evalNetAttr.Evaluator(edgeNetSource, esriNetworkEdgeDirection.esriNEDAgainstDigitized) = CType(netFieldEval, INetworkEvaluator)
netConstEval = New NetworkConstantEvaluator
netConstEval.ConstantValue = 0
evalNetAttr.DefaultEvaluator(esriNetworkElementType.esriNETEdge) = CType(netConstEval, INetworkEvaluator)
evalNetAttr.DefaultEvaluator(esriNetworkElementType.esriNETJunction) = CType(netConstEval, INetworkEvaluator)
evalNetAttr.DefaultEvaluator(esriNetworkElementType.esriNETTurn) = CType(netConstEval, INetworkEvaluator)
' Add the attribute to the array.
attributeArray.Add(evalNetAttr)
'set the array to the network dataset
deND.Attributes = attributeArray

'skip directions
' Get the feature dataset extension and create the network dataset based on the data element.
Dim fdxc As IFeatureDatasetExtensionContainer = CType(fdsGDS, IFeatureDatasetExtensionContainer)
Dim fdx As IFeatureDatasetExtension = fdxc.FindExtension(esriDatasetType.esriDTNetworkDataset)
Dim dsc2 As IDatasetContainer2 = CType(fdx, IDatasetContainer2)
Dim netDataset As INetworkDataset = CType(dsc2.CreateDataset(deND), INetworkDataset)

' Once the network dataset is created, build it.
Dim netBuild As INetworkBuild = CType(netDataset, INetworkBuild)
netBuild.BuildNetwork(fdsGDS.Extent)
```

The Code

PedRouter

- Preparing a route layer with a geoprocessing tool



```
'based on Week7a lecture
'geoprocessor managed assembly
Private Sub SubMakeRouteLayer(ByVal path As String, ByVal dataset As String, _
                              ByVal netDataset As INetworkDataset)
    'for error handling later
    Dim message As String = ""
    '1. create the GP object and tools
    'create the geoprocessor
    Dim GP As ESRI.ArcGIS.Geoprocessor.Geoprocessor = New ESRI.ArcGIS.Geoprocessor.Geoprocessor()
    'create the make route layer tool process object
    Dim rteLayerTool As New ESRI.ArcGIS.NetworkAnalystTools.MakeRouteLayer()
    Dim result As ESRI.ArcGIS.Geoprocessing.IGeoProcessorResult2

    '2. set the tool and environment properties
    Try
        'set default workspace
        GP.SetEnvironmentValue("workspace", path & "\" & dataset)
        'populate the tool with parameter values
        rteLayerTool.in_network_dataset = netDataset
        rteLayerTool.impedance_attribute = "Impedance"
        rteLayerTool.out_network_analysis_layer = "RouteLayer"
        rteLayerTool.output_path_shape = "TRUE_LINES_WITH_MEASURES"
        'execute the tool
        result = GP.Execute(rteLayerTool, Nothing)

    Catch ex As Exception
        'handle the geoprocessor errors
        For Counter As Integer = 0 To GP.MessageCount - 1
            message = message & vbCrLf & GP.GetMessage(Counter)
        Next
        MsgBox("Geoprocessor error: " & message, , "Error")
```

The Code

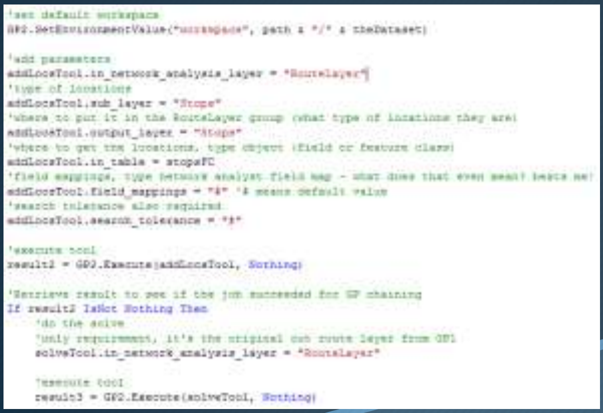
PedRouter

- Opening and preparing an existing PedRouter dataset
  - Used two GxDialogs
    - One for selecting and displaying streets feature class
    - One for selecting network dataset
    - Used appropriate filters and type checking
  - Gathered network dataset as INetworkDataset, workspace
  - Ran same sub as earlier to prepare route layer

The Code

PedRouter

- Using chained geoprocessing tools to add stops and solve
  - GxDialog to open stops FC, then:



```
'set default workspace
GP2.SetEnvironmentValue("workspace", path & "/" & theDataset)

'add parameters
addLocsTool.in_network_analysis_layer = "RouteLayer"
'type of locations
addLocsTool.sub_layer = "Stops"
'where to put it in the RouteLayer group (what type of locations they are)
addLocsTool.output_layer = "Stops"
'where to get the locations, type object (field or feature class)
addLocsTool.in_table = stopsFC
'field mappings, type network analyst field map - what does that even mean? beats me!
addLocsTool.field_mappings = "#" '# means default value
'search tolerance also required
addLocsTool.search_tolerance = "#"

'execute tool
result2 = GP2.Execute(addLocsTool, Nothing)

'Retrieve result to see if the job succeeded for GP chaining
If result2 IsNot Nothing Then
    'do the solve
    'only requirement, it's the original cub route layer from GP1
    solveTool.in_network_analysis_layer = "RouteLayer"

    'execute tool
    result3 = GP2.Execute(solveTool, Nothing)
```

Thank you!