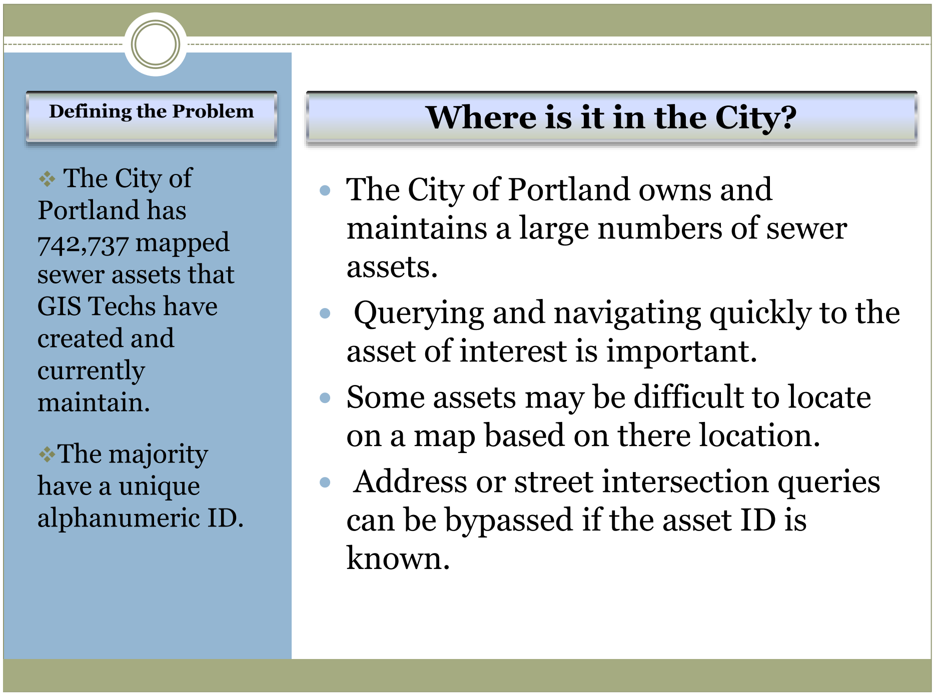
Collection System Asset Finder

Creating an ArcGIS Add-in with Visual Studio

ROBERT PYLE
GEOG590
GIS PROGRAMMING
PORTLAND STATE UNIVERSITY
WINTER
2011

City of Portland Sewer Collection System Background

- Owned by the City of Portland
- Managed by the Bureau of Environmental Services (BES)
- Maintained by the Portland Bureau of Transportation, (PBOT) Maintenance Operations
- 2323 miles of piped system
- 96,000 acres of service area

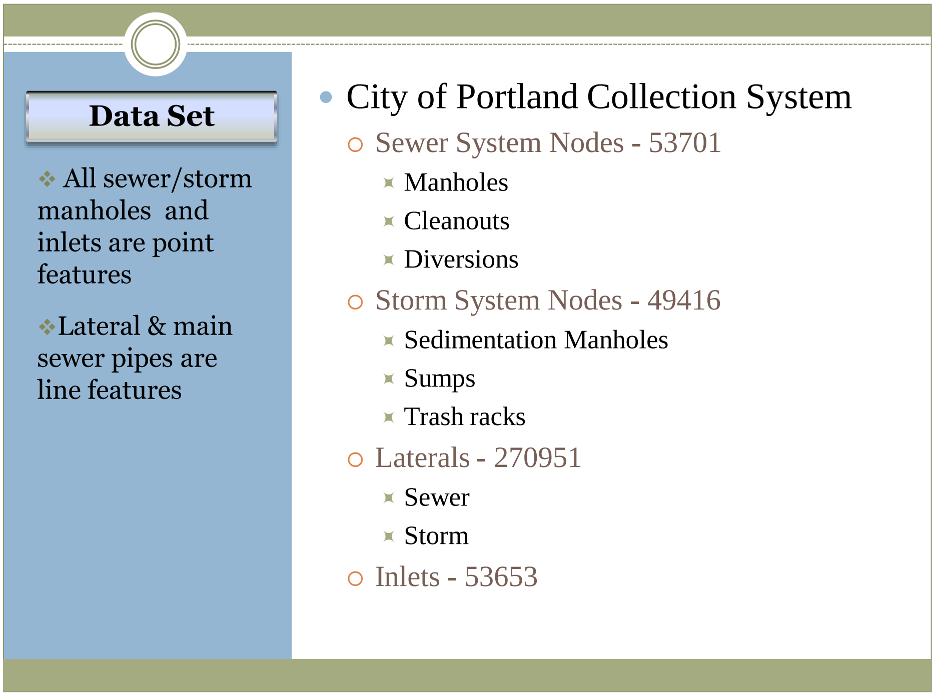


Defining the Problem

- The City of Portland has 742,737 mapped sewer assets that GIS Techs have created and currently maintain.
- The majority have a unique alphanumeric ID.

Where is it in the City?

- The City of Portland owns and maintains a large numbers of sewer assets.
- Querying and navigating quickly to the asset of interest is important.
- Some assets may be difficult to locate on a map based on there location.
- Address or street intersection queries can be bypassed if the asset ID is known.



Data Set

- All sewer/storm manholes and inlets are point features
- Lateral & main sewer pipes are line features

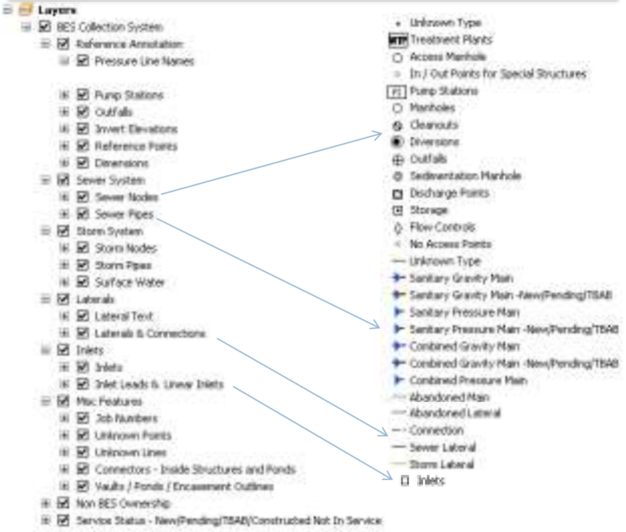
- City of Portland Collection System
  - Sewer System Nodes - 53701
    - Manholes
    - Cleanouts
    - Diversions
  - Storm System Nodes - 49416
    - Sedimentation Manholes
    - Sumps
    - Trash racks
  - Laterals - 270951
    - Sewer
    - Storm
  - Inlets - 53653

Data Set

❖ Each feature may have a unique alphanumeric code to identify it known as the Hansen Unit ID. *(ABC123)*

❖Mains have two alphanumeric IDs based on the up & downstream manhole points. *(ABC123 ABC456)*

City of Portland Collection System Layer



- Layers
  - ☑ BES Collection System
    - ☑ Reference Annotation
      - ☑ Pressure Line Names
      - ☑ Pump Stations
      - ☑ Outfalls
      - ☑ Invert Elevations
      - ☑ Reference Points
      - ☑ Dimensions
    - ☑ Sewer System
      - ☑ Sewer Nodes
      - ☑ Sewer Pipes
    - ☑ Storm System
      - ☑ Storm Nodes
      - ☑ Storm Pipes
      - ☑ Surface Water
    - ☑ Laterals
      - ☑ Lateral Text
      - ☑ Laterals & Connections
    - ☑ Inlets
      - ☑ Inlets
      - ☑ Inlet Leads & Linear Inlets
    - ☑ Misc Features
      - ☑ Job Numbers
      - ☑ Unknown Points
      - ☑ Unknown Lines
      - ☑ Connectors - Inside Structures and Ponds
      - ☑ Vaults / Ponds / Encasement Outlines
    - ☑ Non BES Ownership
    - ☑ Service Status - New/Pending/TBAB/Constructed Not In Service

Method

❖Keep solution simple and easy to use

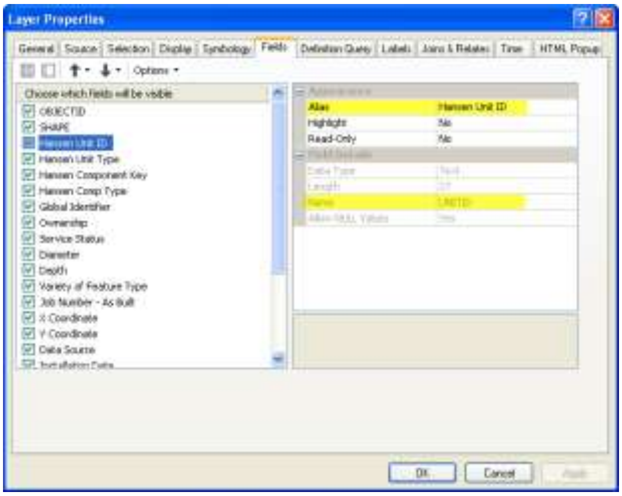
❖Allow user to go back to previously queried asset

- Create combobox
- Add combobox to a toolbar
- Access layers on the map
- Query the multiple layer’s attribute table for asset ID
- If asset ID is found add it to the combobox
- Then zoom to that selected feature at a fixed scale

Import Statements, Class, Constants and Function

```
Imports ESRI.ArcGIS.Carto
Imports ESRI.ArcGIS.Geodatabase
Imports ESRI.ArcGIS.esriSystem
Imports ESRI.ArcGIS.ArcMapUI
Imports System.Windows.Forms

Public Class Pyle_CSAFComboBox

    Inherits ESRI.ArcGIS.Desktop.AddIns.ComboBox
    'Set constant values for mapscale and UnitID
    Private Const FixedMapScale As Double = 600
    Private Const AssetIDColumnName As String = "UNITID"
    'This function will access the layer data
    Shared Function AccessLayersData(ByVal pLayer As ILayer) As IFeatureClass
        Dim pFeatureLayer As IFeatureLayer = pLayer
        If Not pFeatureLayer Is Nothing Then
            Return pFeatureLayer.FeatureClass
        Else
            Return Nothing
        End If
    End Function
```

Attribute Table - Alias vs. Name



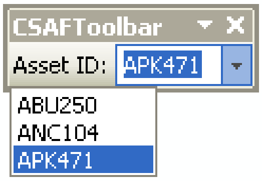
Choose which Fields will be visible

- ☑ OBJECTID
- ☑ SHAPE
- ☑ Hansen Unit ID
- ☑ Hansen Unit Type
- ☑ Hansen Component Key
- ☑ Hansen Comp Type
- ☑ Global Identifier
- ☑ Ownership
- ☑ Service Status
- ☑ Diameter
- ☑ Depth
- ☑ Variety of Feature Type
- ☑ Job Number - As Built
- ☑ X Coordinate
- ☑ Y Coordinate
- ☑ Data Source
- ☑ Installation Date

Appearance	
Alias	Hansen Unit ID
Highlight	No
Read-Only	No
Field Details	
Data Type	Text
Length	[illegible]
Name	UNITID
Allow NULL Values	Yes

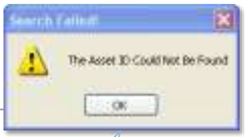
Add Value to Combo Box

```
Private Sub AddValuetoComboBox()

    Dim valueToAdd As String = Me.Value.Trim()
    Dim comboBoxItem As Item

    'Searches to see if value is already in it
    For Each comboBoxItem In Me.items
        If comboBoxItem.Caption = valueToAdd Then
            ' If the value is already in the combo box, do not add it again
            Exit Sub
        End If
    Next
    'Value from ComboBox add to ComboBox list
    Me.Add(valueToAdd)
End Sub
```



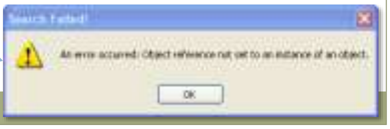
OnEnter, Sub ComboBoxAction

```
Protected Overrides Sub OnEnter()
    MyBase.OnEnter()
    ComboBoxAction()
End Sub

Private Sub ComboBoxAction()
    'this code performs the zoom to asset ID if ID is found.
    ' and displays error message if input can't be found
    Try
        If FindandZoomToID() = False Then
            MessageBox.Show("The Asset ID Could Not Be Found", "Search Failed!"
        Else
            AddValuetoComboBox()

        End If
        'Shows a message box if error occurs in any function
    Catch ex As Exception
        MessageBox.Show("An error occurred: " & ex.Message, "Search Failed!",
    End Try
End Sub
```



Search Failed!

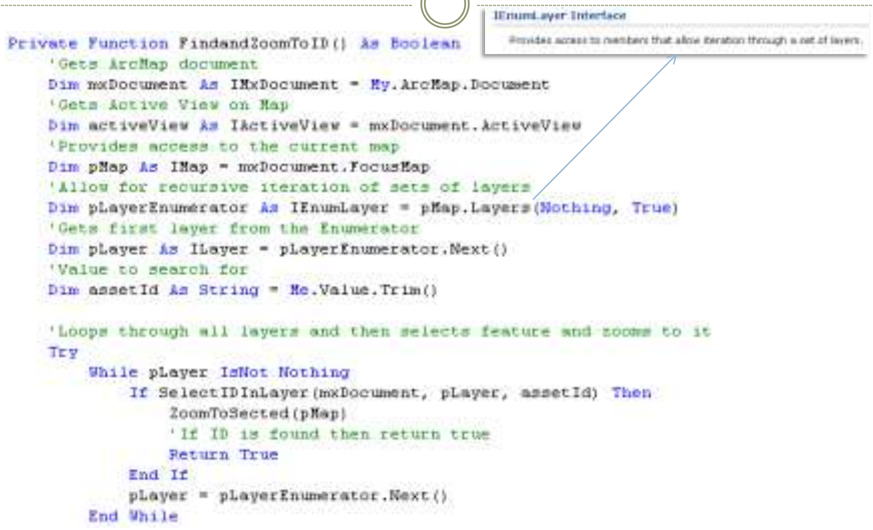
The Asset ID Could Not Be Found



Search Failed!

An error occurred: Object reference not set to an instance of an object.

Find and Zoom To ID

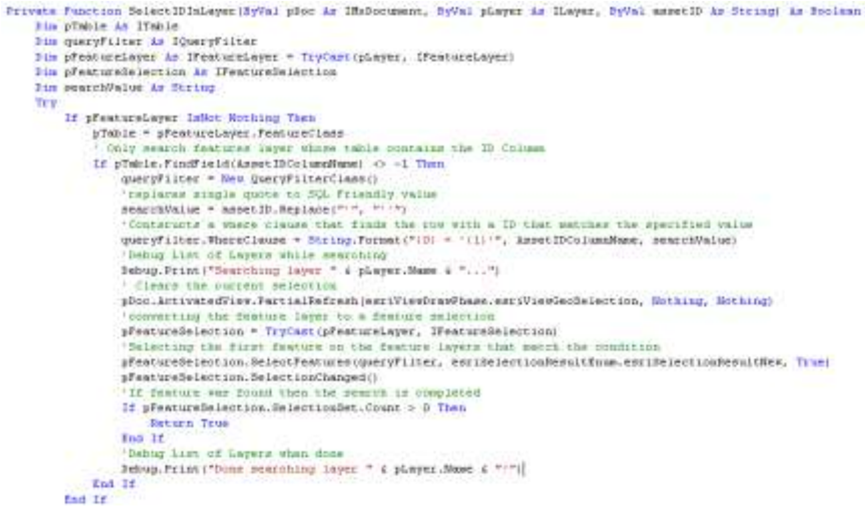


IEnumLayer Interface

Provides access to members that allow iteration through a set of layers.

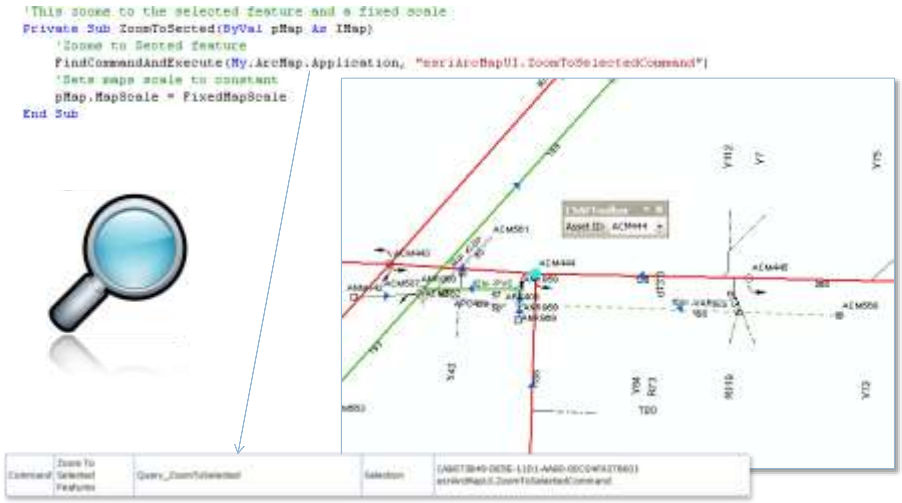
```
Private Function FindandZoomToID() As Boolean
    'Gets ArcMap document
    Dim mxDocument As IMxDocument = My.ArcMap.Document
    'Gets Active View on Map
    Dim activeView As IActiveView = mxDocument.ActiveView
    'Provides access to the current map
    Dim pMap As IMap = mxDocument.FocusMap
    'Allow for recursive iteration of sets of layers
    Dim pLayerEnumerator As IEnumLayer = pMap.Layers(Nothing, True)
    'Gets first layer from the Enumerator
    Dim pLayer As ILayer = pLayerEnumerator.Next()
    'Value to search for
    Dim assetId As String = Me.Value.Trim()

    'Loops through all layers and then selects feature and zooms to it
    Try
        While pLayer IsNot Nothing
            If SelectIDInLayer(mxDocument, pLayer, assetId) Then
                ZoomToSected(pMap)
                'If ID is found then return true
                Return True
            End If
            pLayer = pLayerEnumerator.Next()
        End While
```

Select ID In Layer Function



```
Private Function SelectIDInLayer(ByVal pDoc As IMxDocument, ByVal pLayer As ILayer, ByVal assetID As String) As Boolean
    Dim pTable As ITable
    Dim queryFilter As IQueryFilter
    Dim pFeatureLayer As IFeatureLayer = TryCast(pLayer, IFeatureLayer)
    Dim pFeatureSelection As IFeatureSelection
    Dim searchValue As String
    Try
        If pFeatureLayer IsNot Nothing Then
            pTable = pFeatureLayer.FeatureClass
            ' Only search features layer whose table contains the ID Column
            If pTable.FindField(AssetIDColumnName) <> -1 Then
                queryFilter = New QueryFilterClass()
                'replaces single quote to SQL friendly value
                searchValue = assetID.Replace("'", "''")
                'Contstructs a where clause that finds the row with a ID that matches the specified value
                queryFilter.WhereClause = String.Format("{0} = '{1}'", AssetIDColumnName, searchValue)
                'Debug List of Layers while searching
                Debug.Print("Searching layer " & pLayer.Name & "...")
                ' Clears the current selection
                pDoc.ActivatedView.PartialRefresh(esriViewDrawPhase.esriViewGeoSelection, Nothing, Nothing)
                'converting the feature layer to a feature selection
                pFeatureSelection = TryCast(pFeatureLayer, IFeatureSelection)
                'Selecting the first feature on the feature layers that match the condition
                pFeatureSelection.SelectFeatures(queryFilter, esriSelectionResultEnum.esriSelectionResultNew, True)
                pFeatureSelection.SelectionChanged()
                'If feature was found then the search is completed
                If pFeatureSelection.SelectionSet.Count > 0 Then
                    Return True
                End If
                'Debug list of Layers when done
                Debug.Print("Done searching layer " & pLayer.Name & ":")
            End If
        End If
```

Zoom to Selected w/ProgID



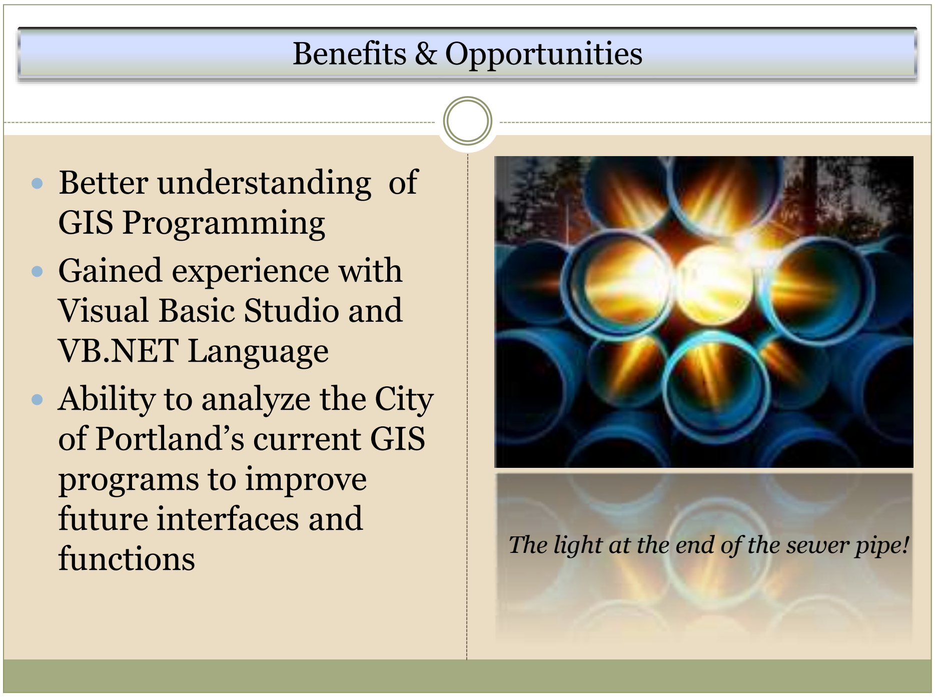
```
'This zooms to the selected feature and a fixed scale
Private Sub ZoomToSected(ByVal pMap As IMap)
    'Zooms to Sected feature
    FindCommandAndExecute(My.ArcMap.Application, "esriArcMapUI.ZoomToSelectedCommand")
    'Sets maps scale to constant
    pMap.MapScale = FixedMapScale
End Sub
```

Command	Zoom To Selected Features	Query_ZoomToSelected	Selection	{AB073B49-DE5E-11D1-AA80-00C04FA37860} esriArcMapUI.ZoomToSelectedCommand

Limitations & Pitfalls

- Alias vs. Name problem
- Query is case sensitive
- Unable to use any code that was already built for the current CGIS toolbar due to the complexity of framework

City of Portland vactor truck falls in sinkhole @ SE 16th & Oak on Dec 29, 2006



Benefits & Opportunities

- Better understanding of GIS Programming
- Gained experience with Visual Basic Studio and VB.NET Language
- Ability to analyze the City of Portland's current GIS programs to improve future interfaces and functions

The light at the end of the sewer pipe!

Demo

"Let Knowledge Serve the City" –*Portland State Motto*

"The City That Works" – *City of Portland Motto*