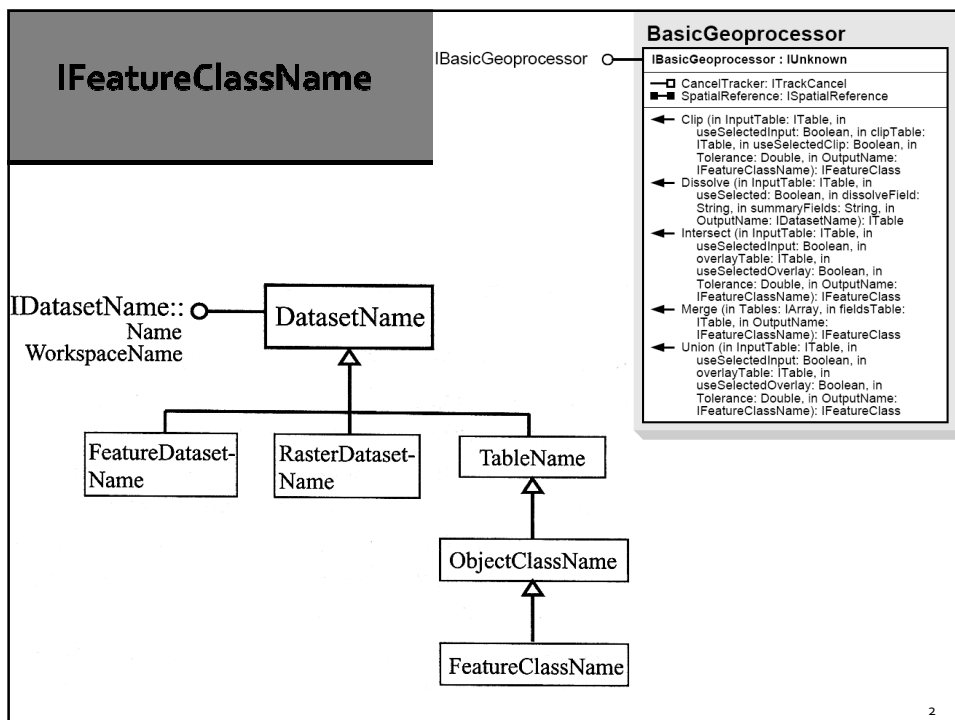


- * ArcObjects: Vector Data analysis II
- 1. Access GIS data via IFeatureClassName in analysis
- 2. Spatial Join
- 3. Vector Buffering
- 4. Vector Overlay

GEOG 4/590: GIS Programming

1



2

Setup a FeatureClassName

```

classDiagram
    class IWorkspaceName {
        BrowseName: String
        Category: String
        ConnectionProperties: IPropertySet
        PathName: String
        Type: esriWorkspaceType
        WorkspaceFactory: IWorkspaceFactory
        WorkspaceFactoryProgID: String
    }
    class IWorkspaceName2 {
        WorkspaceName: IWorkspaceName
        ConnectionString: String
    }
    class IDatasetName {
        Category: String
        Name: String
        SubsetName: IEnumDatasetName
        Type: esriDatasetType
        WorkspaceName: IWorkspaceName
    }
    class IFeatureClassName {
        FeatureDatasetName: IDatasetName
        FeatureType: esriFeatureType
        ShapeFieldName: String
        ShapeType: esriGeometryType
    }
    IWorkspaceName <|-- IWorkspaceName2
    IFeatureClassName --> IDatasetName
  
```

```

Dim pFeatClassName As IFeatureClassName = New FeatureClassName
Dim pDatasetName As IDatasetName = pFeatClassName `casting
Dim pWKSName As IWorkspaceName = New WorkspaceName
pWKSName.WorkspaceFactoryProgID = _
    "esriDataSourcesFile.ShapefileWorkspaceFactory"
pWKSName.PathName = "C:\temp\data"
pDatasetName.WorkspaceName = pWKSName
pDatasetName.Name = "Shapefile_Name"
  
```

3

Spatial Join

- A **spatial join** is a form of spatial analysis in which the attributes of features in two feature classes are joined together based on the relative locations of the features. Spatial joins are not dynamic and require the results to be saved to a new output layer.
- **Source table:** table to append attribute fields to
- **Join table:** table to append attribute fields from

ISpatialJoin Interface

ISpatialJoin performs a join operation based a spatial relationship between two feature classes.

Members

	All <input type="button" value="v"/>	Description
←	JoinAggregate	Join using aggregate. Only features within a distance of maxMapDist will be joined. A maxMapDist of -1 means infinity.
←	JoinNearest	Joins with the nearest feature in the join feature class. Only features within a distance of maxMapDist will be joined. A maxMapDist of -1 means infinity.
—□	JoinTable	The spatial table to append fields from.
←	JoinWithin	Joins a feature in the source feature class with the feature if it falls within in the join feature class.
—■	LeftOuterJoin	Indicates whether a match is required before adding a record from the source feature class to the result. If TRUE, all records in the source feature class are added regardless of whether there is a match.
—■	ShowProcess	Indicates whether to show update messages while processing join.
—□	SourceTable	The spatial table to append fields to.

CoClasses that implement ISpatialJoin

CoClasses and Classes	Description
SpatialJoin	Spatial Join two feature classes.

5

JoinWithin Example

```
Dim pPolygonFC As IFeatureClass = pPolygonLayer.FeatureClass
Dim pPointFC As IFeatureClass = pPointLayer.FeatureClass

'Define the output dataset
Dim pOutWorkspaceName As IWorkspaceName = New WorkspaceName
Dim pFCName As IFeatureClassName = New FeatureClassName
Dim pDatasetName As IDatasetName = pFCName
pOutWorkspaceName.WorkspaceFactoryProgID = _
    "esriDataSourcesFile.ShapefileWorkspaceFactory"
pOutWorkspaceName.PathName = "C:\temp\data\"
pDatasetName.WorkspaceName = pOutWorkspaceName
pDatasetName.Name = "Spatial_JoinWithin"

'Create and define a spatial join object
Dim pSpatialJoin As ISpatialJoin = New SpatialJoin
With pSpatialJoin
    .LeftOuterJoin = True
    .SourceTable = pPointFC 'target att table: point
    .JoinTable = pPolygonFC 'polygon
End With

'Perform spatial joinwithin - join attributes of polygons to points
'joinwith: join occurs when points fall within polygons
pSpatialJoin.JoinWithin(pFCName)
```

6

JoinAggregate Example

```
'joinaggregate
pDatasetName.Name = "Spatial_JoinAggregate"
```

```
With pSpatialJoin
    .LeftOuterJoin = True
    .SourceTable = pPolygonFC 'polygon
    .JoinTable = pPointFC 'point
End With
```

```
Dim pAggregate As IAggregateOptions = pSpatialJoin 'casting
```

```
With pAggregate
    .IsAverage = False
    .IsCount = True
    .IsMax = False
    .IsMin = False
    .IsStdDev = False
    .IsSum = False
    .IsVar = False
End With
```

```
pSpatialJoin.JoinAggregate(pFCName, 0)
'zero as the max distance, -1 indicates an infinite max distance
```

7

Buffering

FeatureCursorBuffer CoClass

Interfaces

Interfaces	Description
IBufferProcessingParameter	Provides access to members that set and retrieve parameters for the buffering process.
IFeatureCursorBuffer	Provides access to members that control the buffering of features.
IFeatureCursorBuffer2	Provides access to additional functions and configuration capabilities for the buffer generation.

IFeatureCursorBuffer2 Interface

All	Description
← Buffer	Buffers features to a new and existing feature class.
■ BufferedGeometry	Enumerator of buffered features.
→ BufferSpatialReference	Specifies in which spatial reference system should be buffered.
← BufferToGraphics	Buffers the selected features and stores them into a composite graphics layer.
→ CancelTrack	TrackCancel used when buffering.
→ DataFrameSpatialReference	Specifies the spatial reference system of the data frame.
■ Dissolve	Indicates if overlapping buffered features should be dissolved.
→ FeatureCursor	Feature cursor of features to buffer (overrides GraphicsLayer).
■ FieldDistance	Field specifying distance to buffer on.
← GraphicsLayer2	Graphics layer of elements to buffer (overrides FeatureCursor).
■ PolygonBufferType	Indicates how to buffer polygon features.
■ RingDistance	Multiple rings specifying distance to buffer on.
→ SourceSpatialReference	Specifies the spatial reference system of the source data (calls IFeatureCursorBuffer::SpatialReference()).
→ SpatialReference	Projection of buffered features.
→ TargetSpatialReference	Specifies the target spatial reference system.
■ Units	Conversion units, from map units to buffer units.
■ ValueDistance	Constant buffer distance.

IFeatureCursorBuffer2

```

Dim pFCursor As IFeatureCursor = pFClass.Search(Nothing, False)
Dim pSRef As ISpatialReference = pMap.SpatialReference

Dim pFCursorBuf As IFeatureCursorBuffer2 = New FeatureCursorBuffer
With pFCursorBuf
    .FeatureCursor = pFCursor
    .Dissolve = True
    .ValueDistance = 500
    .BufferSpatialReference = pSRef
    .DataFrameSpatialReference = pSRef
    .SourceSpatialReference = pSRef
    .TargetSpatialReference = pSRef
End With

'Set Output Shapefile Wkspce & Name
Dim pBufWSName As IWorkspaceName = New WorkspaceName
Dim pBufFCName As IFeatureClassName = New FeatureClassName
Dim pBufDatasetName As IDatasetName = pBufFCName
pBufWSName.WorkspaceFactoryProgID = "esriDataSourcesFile.ShapefileWorkspaceFactory"
pBufWSName.PathName = "c:\temp\data\"
pBufDatasetName.WorkspaceName = pBufWSName
pBufDatasetName.Name = "Buffer_result"

'Use the buffer method.
pFCursorBuf.Buffer(pBufFCName)

```

Vector Overlay - IBasicGeoprocessor

IBasicGeoprocessor

BasicGeoprocessor

IBasicGeoprocessor : IUnknown	
<input type="checkbox"/>	CancelTracker: ITrackCancel
<input checked="" type="checkbox"/>	SpatialReference: ISpatialReference
←	Clip (in InputTable: ITable, in useSelectedInput: Boolean, in clipTable: ITable, in useSelectedClip: Boolean, in Tolerance: Double, in OutputName: IFeatureClassName): IFeatureClass
←	Dissolve (in InputTable: ITable, in useSelected: Boolean, in dissolveField: String, in summaryFields: String, in OutputName: IDatasetName): ITable
←	Intersect (in InputTable: ITable, in useSelectedInput: Boolean, in overlayTable: ITable, in useSelectedOverlay: Boolean, in Tolerance: Double, in OutputName: IFeatureClassName): IFeatureClass
←	Merge (in Tables: IArray, in fieldsTable: ITable, in OutputName: IFeatureClassName): IFeatureClass
←	Union (in InputTable: ITable, in useSelectedInput: Boolean, in overlayTable: ITable, in useSelectedOverlay: Boolean, in Tolerance: Double, in OutputName: IFeatureClassName): IFeatureClass

```

Dim pBGP As IBasicGeoprocessor = New BasicGeoprocessor
pBGP.Intersect(pInputFC, False, pOverlayFC, False, o, pFeatClassName)

```

Intersect Example

```

Dim pInputLayer As IFeatureLayer = pMap.Layer(0)
Dim pInputFC As IFeatureClass = pInputLayer.FeatureClass
Dim pOverlayLayer As IFeatureLayer = pMap.Layer(1)
Dim pOverlayFC As IFeatureClass = pOverlayLayer.FeatureClass

'Define the feature class name and output location
Dim pNewWSName As IWorkspaceName = New WorkspaceName
Dim pFeatClassName As IFeatureClassName = New FeatureClassName
Dim pDatasetName As IDatasetName = pFeatClassName
pNewWSName.WorkspaceFactoryProgID = _
    "esriDataSourcesFile.ShapefileWorkspaceFactory"
pNewWSName.PathName = "c:\temp\data\"
pDatasetName.WorkspaceName = pNewWSName
pDatasetName.Name = "StreamBuffer_AOI_Intersect"

'Define a basic geoprocessor object
Dim pBGP As IBasicGeoprocessor = New BasicGeoprocessor
Dim tol As Double = 0 'Use default tolerance
Dim pOutFC As IFeatureClass = _
    pBGP.Intersect(pInputFC, False, pOverlayFC, False, tol, pFeatClassName)

```

Bonus Code Examples

Find a layer from the TOC

```

Dim pInputLayer As IFeatureLayer = pMap.Layer(GetLayerIndexByName("aoi_v"))
...
Public Shared Function GetLayerIndexByName(ByVal layerName As String) As Integer
    Dim pMxDoc As IMxDocument = My.ArcMap.Document
    Dim pMap As IMap = pMxDoc.FocusMap
    Dim pFLayer As IFeatureLayer

    If layerName.Trim.Length = 0 Then 'invalid input argument
        Return -1
    End If

    Dim i As Integer
    For i = 0 To pMap.LayerCount - 1
        pFLayer = pMap.Layer(i)
        If pFLayer.Name = layerName Then Exit For
    Next

    If i = pMap.LayerCount Then i = -1 'layer not found
    Return i
End Function

```

12

Delete a GIS data set (shapefile)

```
Public Shared Sub DeleteShapeFile(ByVal filePath As String, ByVal fileName As String)
    Dim pWSF As IWorkspaceFactory
    Dim pWS As IFeatureWorkspace
    Dim pFClass As IFeatureClass
    Dim pDataset As IDataset
    Try
        pWSF = New ShapefileWorkspaceFactory
        pWS = pWSF.OpenFromFile(filePath, 0)
        If Not fileName.Contains(".shp") Then fileName = fileName & ".shp"
        pFClass = pWS.OpenFeatureClass(fileName)
        pDataset = pFClass 'casting
        pDataset.Delete()
    Catch ex As Exception
        MsgBox("Unable to delete " & fileName)
    Finally
        pDataset = Nothing
        pFClass = Nothing
        pWS = Nothing
        pWSF = Nothing
    End Try
End Sub
```

13