Chapter 5: Working with Feature Service, Part I

FDO Providers, Feature Query, and Feature Selection

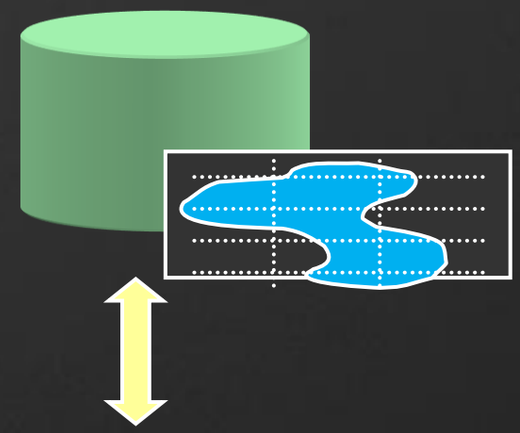
Autodesk

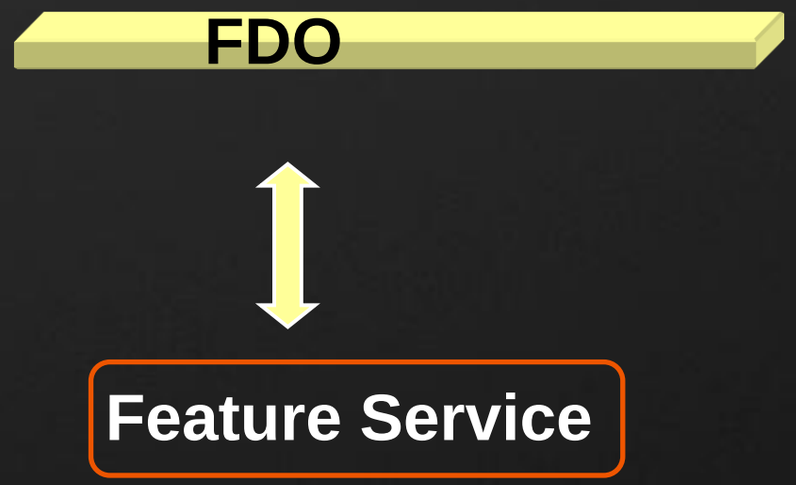
Chapter Overview

- Introduction to feature service
- FDO providers and their capabilities
- Feature source and schema
- Feature query
  - Tabular query
  - Spatial query
- Feature spatial operations
- Feature selection
  - Client side
  - Server side

Feature Service Overview

- Feature service provides APIs to store and retrieve features.
- Independent of the data storage technology by creating an abstraction layer.
- FDO (Feature Data Object) is used to construct this abstraction layer.
- Key concepts
  - Feature
  - Feature source
  - Feature class

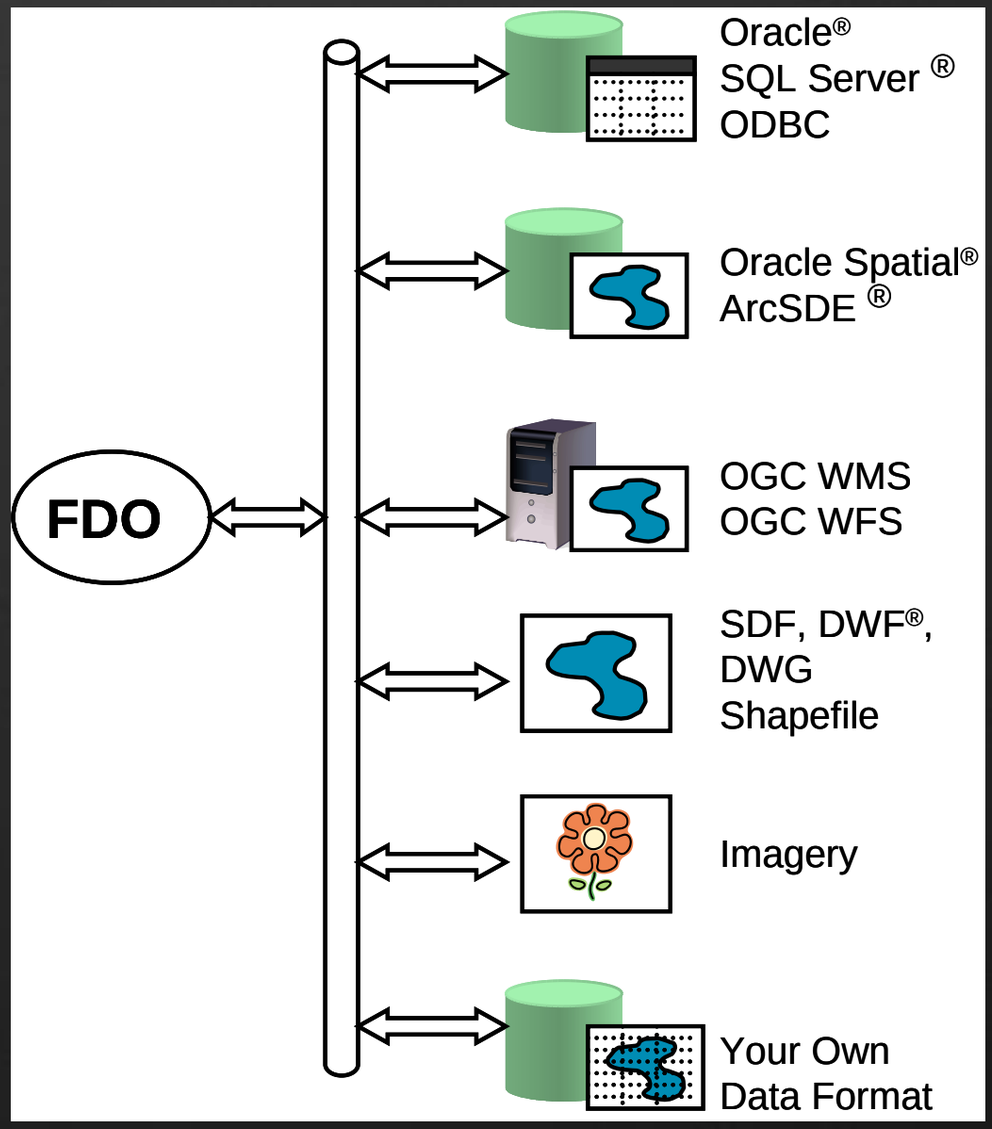




FDO Providers

FDO (Feature Data Objects)

- a set of API for manipulating, defining, and analyzing geographic data.
- Provides consistent access to different data storage with a common general purpose abstraction layer.
- You can implement your own custom FDO provider with FDO SDK



Autodesk

Get Feature Service

- Feature service can be created from MgSiteConnection object.
- It can restore the previous MapGuide session state because the MgSiteConnection object was created with session ID.

```
MgUserInformation userInfo = new MgUserInformation(sessionID);
siteConnection = new MgSiteConnection();
siteConnection.Open(userInfo);

MgFeatureService featureService =
 (MgFeatureService)siteConnection.CreateService(
    MgServiceType.FeatureService);
```

FDO Registry

- FDO providers are registered on MapGuide Enterprise server.
- Physically under this directory -
  C:\Program Files\Autodesk\MapGuideEnterprise2011\Server\Bin\FDO
- Programmatically get the registry and list the providers with this code snippet.

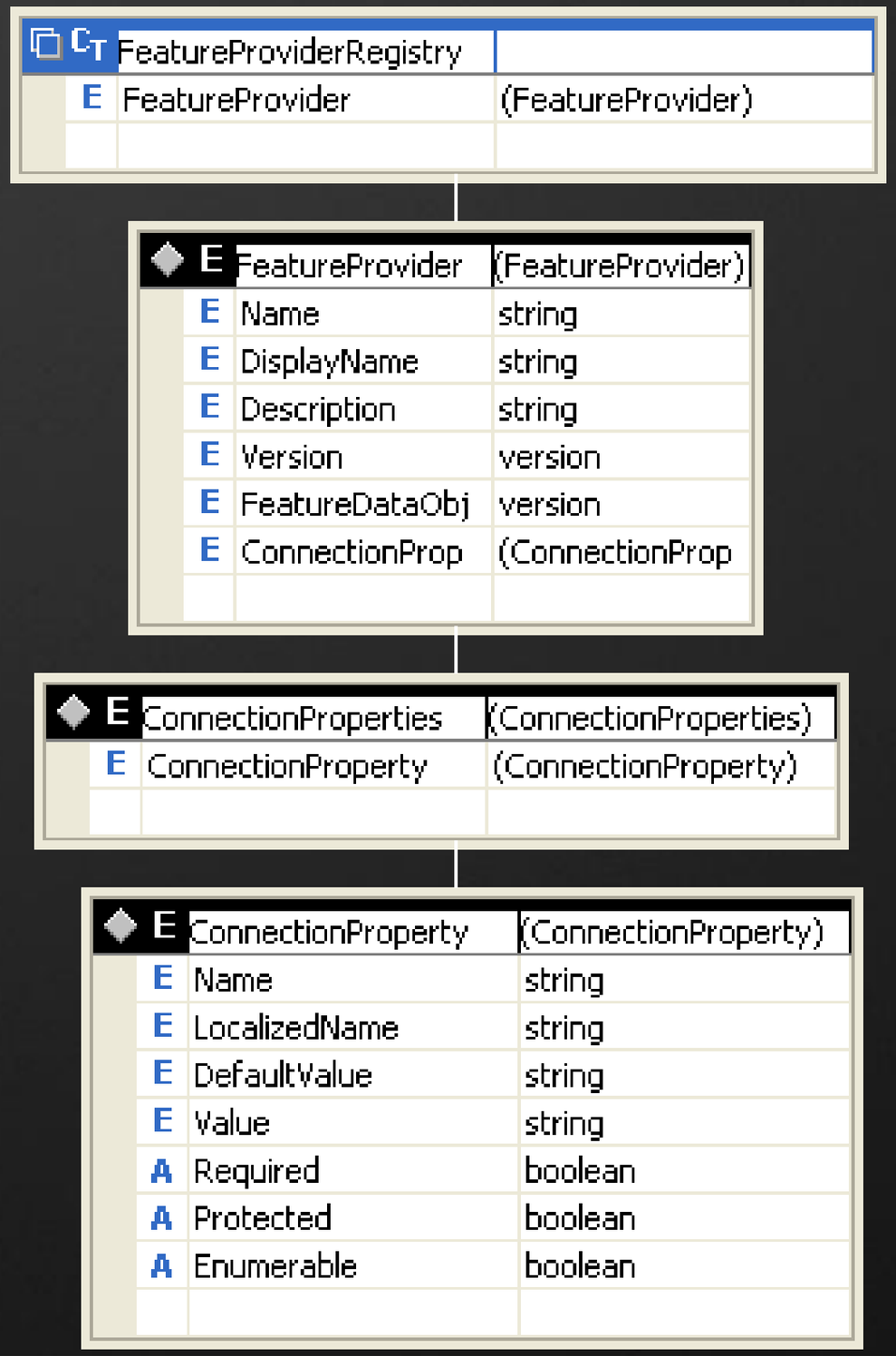
```
MgByteReader reader =
featureService.GetFeatureProviders();
String providers = reader.ToString();
```

```
<?xml version="1.0" encoding="UTF-8" standalone="no" ?>
- <FeatureProviderRegistry>
- <FeatureProvider>
    <Name>OSGeo.SDF.3.4</Name>
    <DisplayName>OSGeo FDO Provider for SDF</DisplayName>
    <Description>Read/write access to Autodesk's spatial database format, a file-ba
    <Version>3.4.0.0</Version>
    <FeatureDataObjectsVersion>3.4.0.0</FeatureDataObjectsVersion>
  - <ConnectionProperties>
    - <ConnectionProperty Enumerable="false" Protected="false" Required="true">
        <Name>File</Name>
        <LocalizedName>File</LocalizedName>
        <DefaultValue />
      </ConnectionProperty>
    - <ConnectionProperty Enumerable="true" Protected="false" Required="false">
        <Name>ReadOnly</Name>
        <LocalizedName>ReadOnly</LocalizedName>
        <DefaultValue>TRUE</DefaultValue>
        <Value>FALSE</Value>
        <Value>TRUE</Value>
      </ConnectionProperty>
    - <ConnectionProperty Enumerable="false" Protected="false" Required="false">
        <Name>MaxCacheSize</Name>
        <LocalizedName>MaxCacheSize</LocalizedName>
        <DefaultValue />
      </ConnectionProperty>
    </ConnectionProperties>
  </FeatureProvider>
- <FeatureProvider>
    <Name>OSGeo.SHP.3.4</Name>
    <DisplayName>OSGeo FDO Provider for SHP</DisplayName>
    <Description>Read/write access to spatial and attribute data in an ESRI SHP file.
    <Version>3.4.0.0</Version>
    <FeatureDataObjectsVersion>3.4.0.0</FeatureDataObjectsVersion>
```

Understand FDO Registry

- FDO registry lists all the FDO providers available to MapGuide Enterprise server.
- It also shows the connections properties on how to establish such a connection.
- For example
  - OSGeo.SDF.3.4
    - File
    - ReadOnly
  - Autodesk.Oracle.3.4
    - Username
    - Password
    - Service
    - Datastore

FDO Registry Schema



Autodesk

FDO Provider Capabilities

- Different FDO provider has different capabilities.
- For example
  - SHP and SDF don’t support topology.
  - Oracle and SQL Server support database transaction.
  - Oracle has much more spatial operations than SDF.
- Get a provider’s capability programmatically.

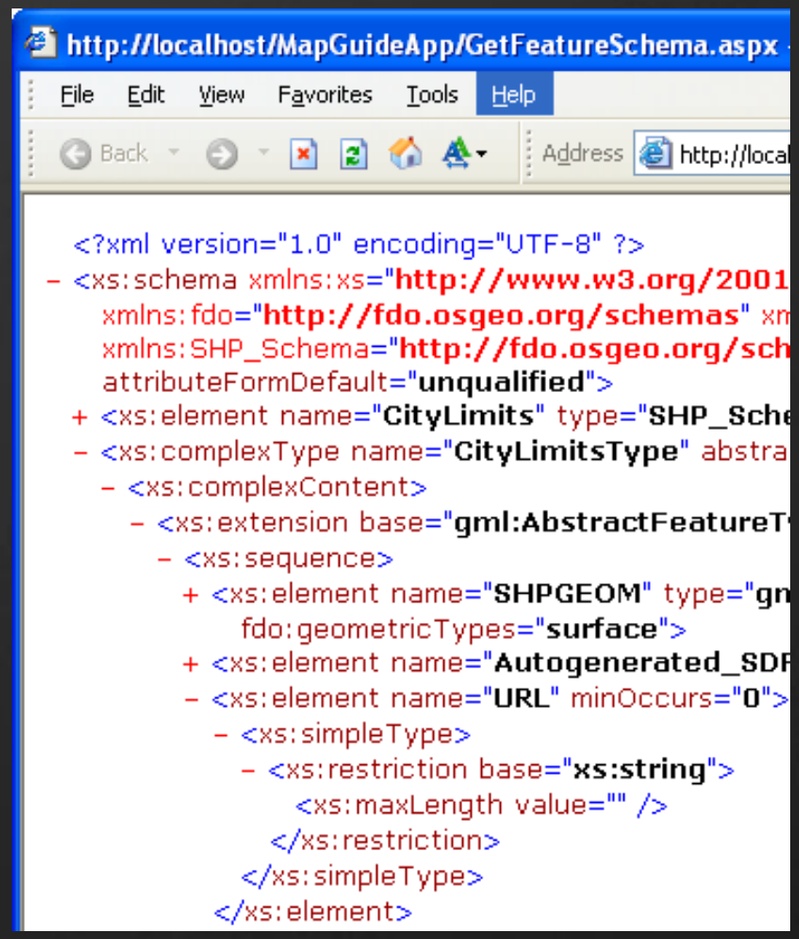
FDO Provider Capabilities

FdoProviderCapabilities	
E Provider	(Provider)
E Connection	(Connection)
E Schema	(Schema)
E Command	(Command)
E Filter	(Filter)
E Expression	(Expression)
E Raster	(Raster)
E Topology	(Topology)
E Geometry	(Geometry)

```
featureService.GetCapabilities(fullProviderName);
```

Understand Feature Schema

- Feature schema Defines the structure of data contained in a feature source.
- A feature class contains a set of properties.
- Each property has a set of attributes correspondent to the type of data it contains.
- Think feature class as a table and property as a column.
- Get the schema programmatically.




```
featureService.DescribeSchemaAsXml(resourceIdentifier, "");
```


Autodesk

Get Features from Source

- Feature service provides methods to retrieve features form a source.
- Retrieved features are stored in MgFeatureReader.
- Enumerate through the MgFeatureReader object to get individual feature.
- Get the property values based on the types defined in the feature schema.

```
MgFeatureReader featureReader = featureService.SelectFeatures(resId,
"VotingDistricts", null);

while (featureReader.ReadNext()) {
  int key = featureReader.GetInt32("Autogenerated_SDF_ID");
  MgByteReader byteReader = featureReader.GetGeometry("Data");

  MgGeometry geometry = geometryReaderWriter.Read(byteReader);
  MgPoint point = geometry.GetCentroid();
  double x = point.GetCoordinate().GetX();
  double y = point.GetCoordinate().GetY();
}
```

Query with Filter

- You can select a set of features from a source according to the criteria set in MgFeatureQueryOptions.
- Selection can be performed on both feature attributes and feature geometries.

```
MgFeatureQueryOptions query = new MgFeatureQueryOptions();
query.SetFilter("Year >= 1990");
query.SetSpatialFilter("SHPGEOM", geometry,
    MgFeatureSpatialOperations.Inside);
MgFeatureReader featureReader =
    featureService.SelectFeatures(resId, "feature_class_name",    query);
```

Basic Filter

Basic Filter

- Mainly used to perform queries on the attribute table.
- Conditions
  - Comparison
    >, <, <>, >=, <=
  - Like
  - In
- Expression functions
  - Avg
  - Sum
  - Count
  - Min, Max
  - Ceil, Floor
  - …

Filter String Examples

```
YEAR > 1990 and Year < 2000
NAME like “Richmond%”
RVALUE in (500000, 1000000)
DATE > ‘1995-3-15’
ADDRESS NULL
Ceil(PRICE1*1.2) < PRICE2*0.8
```

Geometry Spatial Relationship

Spatial Relationship Predicates

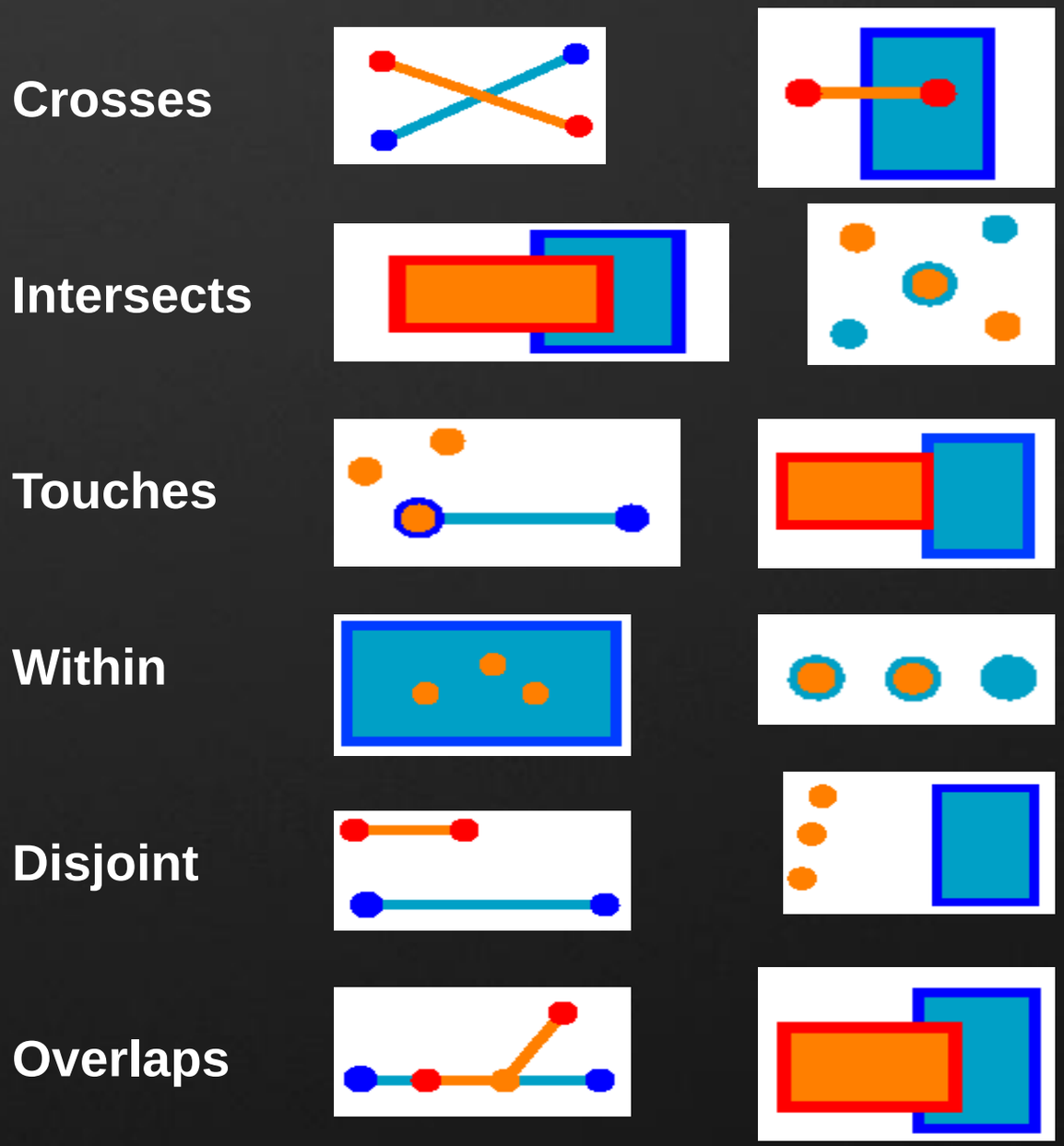
- Defined by the definition of boundary, interior, and exterior of two geometries.

OGC Common Predicates

- Contains
- Crosses
- Disjoint
- Equals
- Intersects
- Overlaps
- Touches
- Within

Note: please refer to the provider's capabilities for the actual predicts.

Spatial Relationship Predicate Samples



Autodesk

Spatial Filter

Spatial filters

- Operate queries on the relationship of two geometries.
- Based on spatial relationships
- Two ways to set spatial filters on MgFeatureQueryOptions -
  - SetSpatialFilter
  - SetFilter

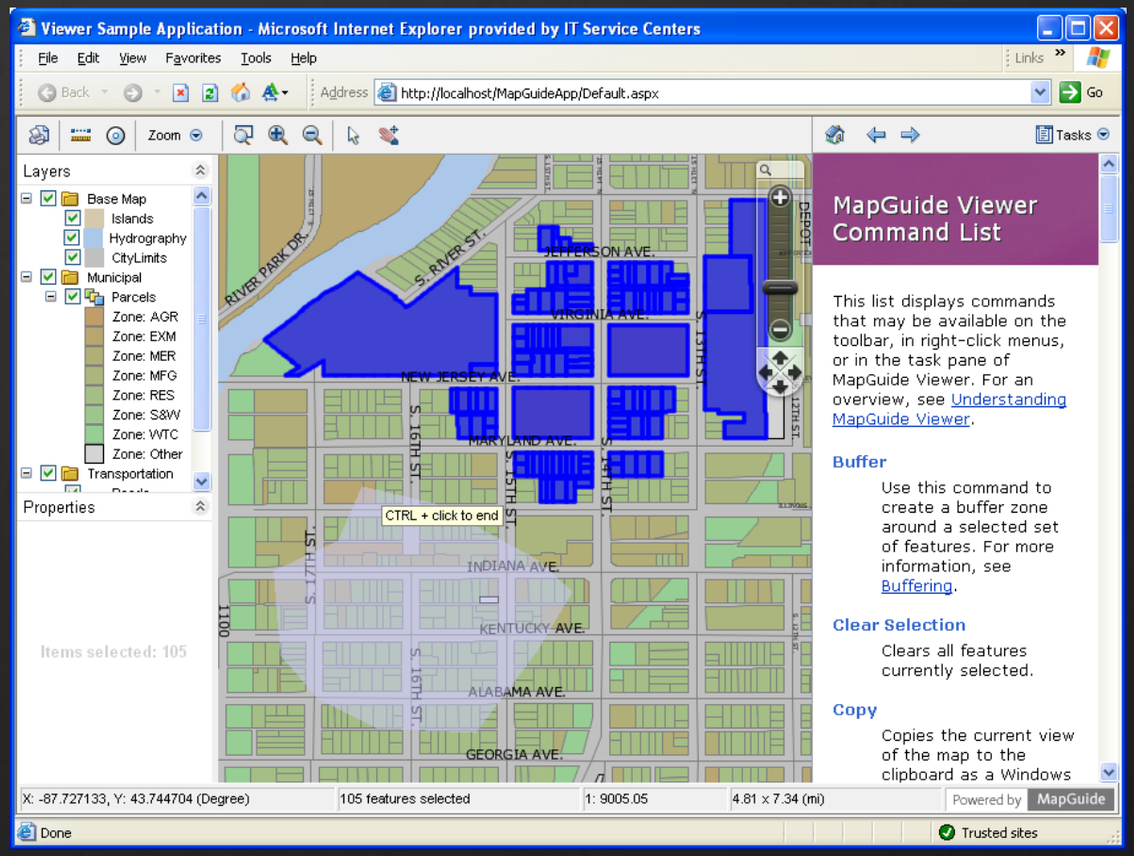
```
MgFeatureQueryOptions query = new MgFeatureQueryOptions();
String areaWkt = "POLYGON ((0 0, 2 0, 2 2, 0 2, 0 0))";
query.SetFilter("SHPGEOM inside GEOMFROMTEXT(" + areaWkt + ")");

query.SetSpatialFilter("SHPGEOM", aGeometryObject,
    MgFeatureSpatialOperations.Inside);
```

Work with Selections

- Selection is the operation to pick up features according to user's interests.
- Selection has two parts:
  - On the client side, user specify the area of interests with the viewer tool.
  - On the server side, the map display is updated accordingly to highlight the features within the area of interest.
- The communication between client and server is via XML.

Selections on viewer



Autodesk

Select on Client Side (AJAX/Fusion)

- Select on the client side with AJAX or Fusion viewer and get results on the server side
- Selection results can be accessed with server APIs.

```
MgMap map = new MgMap();
map.Open(resourceService, "map_name");
MgSelection selection = new MgSelection(map);
selection.Open(resourceService, "map_name");
String filter = selection.GenerateFilter(mgLayer,
    “feature_class_name");
MgFeatureQueryOptions option = new MgFeatureQueryOptions();
option.SetFilter(filter);
featureService.SelectFeatures(resId, "feature_class_name",
    option);
```

Select on Server Side

- Selection can also be made on the server side programmatically
- Selection results can be sent to the viewer to update the map display and highlight selected features.

```
featureReader = featureService.SelectFeatures(resourceID, "Parcel",
    option);
MgSelection newSelection = new MgSelection(map);
newSelection.AddFeatures(mgLayer, featureReader, 0);
String selectionXml = newSelection.ToXml();
```

```
<script language="javascript">
    function onPageLoad() {
        var selectionXML = '<%= selectionXML %>';
        parent.parent.SetSelectionXML(selectionXML);
    }
</script>
```

Questions

Questions ?

Autodesk

Exercise

- Highlight selection
- List selection

- Solution 4