DEVELOPER SUMMIT

March 10–13

Geometry and Data API

Thomas Emge

Editing Namespace

- EditingModule provides access to the templates and the sketch geometry.
- EditEvents capture creates, modifies, and deletes.
- Houses FeatureInspector.

Edit operations

- **All data manipulations should be executed as part of an edit operation.**
- **An edit operation can encapsulate multiple and various edits.**

```
// create an edit operation
var editOperation = EditingModule.CreateEditOperation();
editOperation.Name = "My first edit";

// queue the edit
editOperation.Create(CurrentTemplate, geometry);

//execute the operation
return editOperation.ExecuteAsync();
```

Reminder

- Recall the previous lecture on how to get code execution to the CIM thread
- Geometry and Data API are considered fine-grained APIs
- Can only be executed on the CIM thread

```
QueuingTaskFactory.StartNew(…)
```

Or

```
protected override void OnClick()
{
    MapPoint newPoint = new MapPoint();
}
```

ConstructedOnWrongThreadException was unhandled

An unhandled exception of type 'ArcGIS.Core.ConstructedOnWrongThreadException' occurred in ArcGIS.Core.dll

Troubleshooting tips:

Get general help for exceptions.

ArcGIS.Core.Data API

- **Provides access to the Geodatabase model incl. datasets, feature classes, schema definitions, relationship class and versioning information.**
- **Schema information is stored in definitions.**

```
string fileGDBLocation = @"c:\data\devsummit2015.gdb";

// open the file geodatabase from provided location
var fileGeodatabase = new Geodatabase(fileGDBLocation);

// open the dataset of type feature class by name
var featureClass = fileGeodatabase.OpenDataset<FeatureClass>("SimpleLines");

// get the schema information
var classDefinition = featureClass.Definition;
```

ArcGIS.Core.Data API (continued)

- **Table is the source object, containing a collection of rows that contain the data values.**
- **Data content is retrieved by constructing a search cursor.**

```
// search the feature class by creating a row cursor
var rowCursor = featureClass.Search(queryFilter, false);

// loop through the returned rows
while (rowCursor.MoveNext())
{
    // retrieve the current row from the cursor
    var row = rowCursor.Current;

    // do something with the content of the row
    long oid = (long)row[classDefinition.ObjectIDField];
}
```

ArcGIS.Core.Geometry API

- Similar to ArcGIS Runtime API
- Contains key classes and enumerations defining geometry types such as MapPoint, Envelope, Polyline, Polygon, and SpatialReference.
- Containers that house collections of the geometries as CoordinateCollection, PartCollection and SegmentCollection.

ArcGIS.Core.Geometry API (continued)

- **GeometryEngine static methods can perform numerous geometry manipulations such as buffer, cut, clip, offset, union, etc.**

```
var palmSprings = new MapPoint(-116.545, 33.83, new SpatialReference(4326));
var aroundConventionCenter = GeometryEngine.Buffer(palmSprings, 0.035);
```