



# ArcGIS Explorer 900 Developer: A truly managed experience

**Demo Theatre**

*Chris Davies*

*cdavies@esri.com*



# Summary

- **API Overview**
  - **Getting Started**
  - **.Net Framework Guidelines**
  - **Explorer API Patterns**
- 
- **Ask questions anytime**

# API OVERVIEW

# Explorer API Overview

- **Explorer 900 new features**
  - 2D and 3D display
  - Share layers with ArcMap/ArcGlobe
- **Usability Improvements**
  - New UI - Ribbon based
  - New .Net API
- **API Functionality matches the App**
  - Example: “Note” in UI is a Note class in API

# Explorer API Overview

- **API Divided into 5 Namespaces**
  - Application
  - Mapping
  - Data
  - Geometry
  - Threading
- **Customize Explorer with plug-in classes namespace**
  - Base classes in Application Namespace
  - Button, DockWindow, Gallery, Extension

# Explorer API Overview

- **Use classes from “lower” namespaces**
- **Mapping**
  - Map, MapDisplay , Symbol, Graphic
  - MapItems: Note, Layer, Folder, View, Link ...
- **Data**
  - Table, Row, Column, Raster, Geodatabase ...
- **Geometry**
  - Point, Polyline, Polygon, Coordinate System ...

# Explorer API Overview

- **Package plugin and its resources**
  - Zip file .eaz
  - Built from Visual Studio
- **Deploy using Application Configuration**
  - Centralised zip file (.ncfg)
  - Use Application Config tool
  - Add many plugins
  - Generates Xml to rearrange ribbon UI
  - Contains other resources map, basemaps, help system etc.



**GETTING STARTED**



# Getting Started with Development

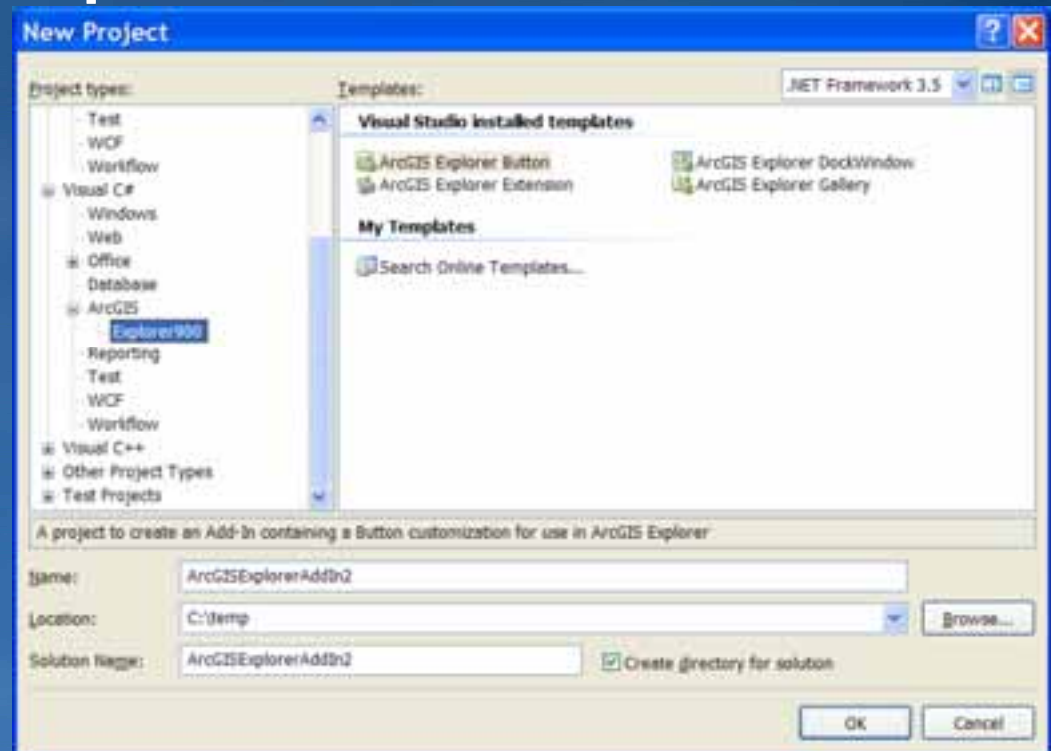
## 1. Download and Install Explorer 900 Application

- Runs alongside Explorer 500

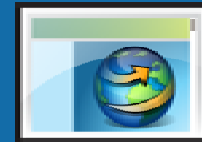
## 2. Download and Install Explorer SDK

- Visual Studio 2008
- Project Templates
- Help content

## 3. Create plug-in Button



*It's time for a*



**Demonstration**

# **.NET GUIDELINES**

# .Net Design Guidelines for Explorer API

- **Naming is crucial**
- **Avoid interfaces**
- **Keep inheritance shallow**
- **Avoid many small classes**
- **This all helps usability**
  - **intellisense lets you explorer the API**

# .Net Patterns

- Overloads

- Methods and constructors

```
Note note = new Note (  
    2 of 4 Note.Note (string name, Geometry geometry)
```

- Static methods

```
Point clickedPoint = Application.ActiveMapDisplay.TrackPoint();  
Geometry outGeom = GeometryOperations.Project(inGeom, inCoordSys);  
Symbol markerSym = Symbol.CreateMarker(@"c:\smile.png");
```

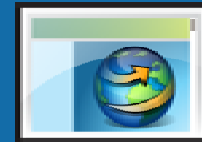
↑  
Class name

↑  
Property or method  
directly on class

- Collections

```
foreach (MapItem item in map.ChildItems)
```

*It's time for a*



**Demonstration**

# EXPLORER API PATTERNS

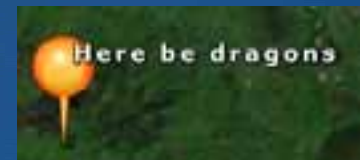
# Explorer API Patterns #1

- Coding against 2D or 3D largely the same
  - One MapDisplay with 2D or 3D ViewMode property
  - MapDisplay.ZoomTo to fly to a new place
  - Viewpoint class captures 2D or 3D

```
note.Viewpoint = mapDisplay.GetViewpoint();
```

- Defaults used where possible
  - Symbols defaulted on Notes and FeatureLayers
  - Geometries default WGS84 Coordinate System

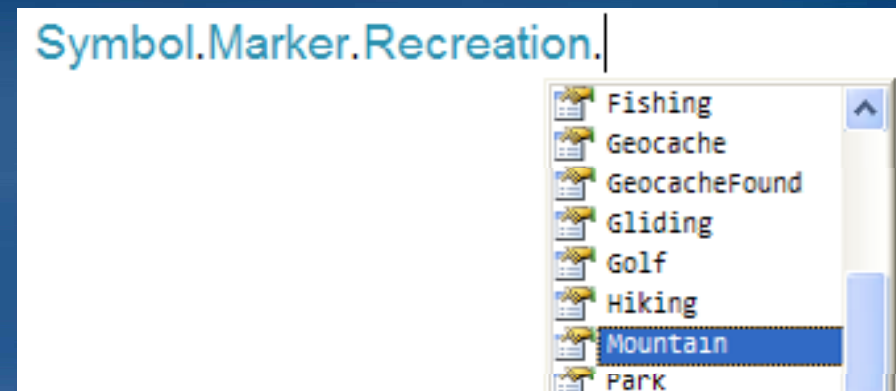
```
Point myPoint = new Point(-3, 53);
```





## Explorer API Patterns #2

- Static properties for access to installed resources
  - Symbols
  - Coordinate Systems



- Layers guide you through Connection

```
ServiceLayer.OpenMapServer("http://MyServer/ArcGIS/Services", "YellowstoneMap");
```

```
if (mountainsLayer.ConnectFileGeodatabase(@"C:\Data\Scotland.gdb", "mountains"))
```

## Explorer API Patterns #3

- Collapse key objects for simplicity
  - One Symbol class for all symbols
  - One Table class with IsSpatial property, one Row class too
  - Five Layer classes represent all layer types
  
- Changes to objects on a map automatically update UI
  - No need to call a Refresh
  - MapDisplay and Content update automatically

```
note.Graphic.Symbol = Symbol.Marker.Transportation.Bus;
```

```
mountainsLayer.Visible = false;
```

## Explorer API Patterns #4

- **MapItem hierarchical collection**
  - Map and Folder have ChildItems
  - Parent property on every map item
  - Map.Descendants to enumerate all children, grandchildren etc
  - GetMapItems<> to return collection of specific type

```
foreach (MapItem item in map.Descendants)
{
    if (item.Name == "Hiking Routes" && item is Folder)
        // Do Stuff
}
```

```
// returns all notes in the map in any folder
var notes = map.GetMapItems<Note>();
```

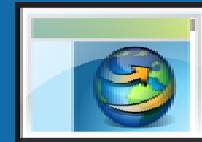
## Explorer API Patterns #5

- **Threading**
  - Threads checks to throw exceptions for incorrect usage
  - Explorer BackgroundWorker for background processing
- **.Net XML serialization used for persistence**
  - Helper methods to save and load xml to string
  - Use to transfer state to Background Worker threads

```
string xmlString = note.ToXmlString();
```

```
Note.CreateFromXmlString(xmlString);
```

*It's time for a*



**Demonstration**

# SUMMARY

# Summary

- **Getting started**
  - **Design principles**
  - **.Net Guidelines**
  - **Explorer patterns**
- 
- **Any API feedback is welcome, no matter how small**
    - **Explore 900 Beta forums**

## Useful Info

- Chris Davies: [cdavies@esri.com](mailto:cdavies@esri.com)
- [Download: API Evaluator](#)
  - Helps you and ESRI understand API usage
- [Book: .Net Framework Design Guidelines](#)



# Explorer Sessions

- **Tuesday 5:00pm**
  - Explorer Meet the teams Oasis 1
- **Wednesday 10:00am**
  - Demo Theater: Using application configurations to roll out a customized ArcGIS Explorer 900 experience
- **Wednesday 2:45pm (part 1) + 4:30pm (part 2)**
  - **Technical Session: Introducing and Implementing ArcGIS Explorer 900**