



ArcGIS Pro .NET SDK: The Road Ahead

Rob Elkins Jr - Esri

Charlie Macleod - Esri

Technical Workshop

ArcGIS Pro .NET SDK: The Road Ahead

- **This is the new .Net SDK for the ArcGIS Pro Application**
 - **Component of a larger SDK for ArcGIS Pro that includes Python and arcpy.mapping**
 - **For extending ArcGIS Pro via the ArcGIS Pro .Net API**
 - **Leverages .Net features and programming patterns**
 - **Application UI state integrated with async .Net API (TAP pattern)**
 - **Geometry API consistent with Runtime**
 - **ArcGIS Pro Application APIs under development**

ArcGIS Pro .NET SDK: The Road Ahead

- **Extensibility is provided by the ArcGIS Pro Framework**
 - **ArcGIS.Desktop.Framework.dll**
- **“Plug in” to Framework Extensibility via Addins in the SDK**
 - **Visual Studio templates provided for many different Framework elements**
 - **Currently C#**
 - **VBNet to follow**
- **ArcGIS Pro Functionality provided via its API. No separate PIAs required**

ArcGIS Pro .NET SDK: The Road Ahead

Timeframe

- **July 2014**
- **ArcGIS Pro Application is at Beta**
 - “Beta 4”
 - .Net SDK is at Beta4. Currently available within the EAP only
- **November 2014**
- **ArcGIS Pro Application Final 1.0**
 - .Net SDK is at 1.0. Available as a Beta to all ArcGIS Pro users
- **March 2015**
- **ArcGIS Pro Application Final 1.1**
 - .Net SDK is at 1.1. Available as a final release to all ArcGIS Pro users

ArcGIS Pro .NET SDK: The Road Ahead - Addins Defined

- **Addins defined:**
- **Declarative definition (in “daml” – declarative xml – config file), “Metadata”**
- **.Net Assembly containing the implementation**
 - **Can contain 1 or more framework UI extensions**
 - **Implement Framework Plugin Contract**
- **Resources, images, strings, data**
- **References to ArcGIS Pro Assemblies (“the” API)**
- **Registry free**

- **Addins deployed to a well-known folder under <user>\MyDocuments**

ArcGIS Pro .NET SDK: The Road Ahead - **Addins Defined**

- **Demo (button)**
- **Demo (custom control)**
- **Demo (checkbox – events)**
- **Demo (edit tool)**
- **Demo (dockpane)**
- **Demo (layer context menu)**

ArcGIS Pro .NET SDK: The Road Ahead - Addin Architecture

- DAML Definition
- The excerpt below shows the beginnings of an extension. In this case a new Module is being added and the Module has a button

```
<?xml version="1.0" encoding="utf-8"?>
<ArcGIS defaultAssembly="UCSamples.dll" defaultNamespace="UCSamples"
xmlns="http://schemas.esri.com/DADF/Registry"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://schemas.esri.com/DADF/Registry
file:///C:/Program%20Files/ArcGIS%20Pro/bin/ArcGIS.Desktop.Framework.xsd">

  <modules>
    <insertModule id="UCSamples_Module1_id" className="ForTheUcModule" autoLoad="false"
caption="Module1">
      <groups>
        <group id="UCSamples.FixedZoom_id" caption="Fixed Zoom" appearsOnAddInTab="true">
          <!-- host controls within groups -->
          <button refID="UCSamples.FixedZoomIn" size="large" />
        </group>
      </groups>

      <controls>
        <!-- add your controls here -->
        <button id="UCSamples.FixedZoomIn" caption="FixedZoomIn"
className="UCSamples.FixedZoom.FixedZoomIn" loadOnClick="true"
smallImage="Images\ZoomFixedZoomIn16.png" largeImage="Images\ZoomFixedZoomIn32.png"
condition="esri_mapping_mapPane">
          <tooltip heading="Tooltip Heading">
            Fixed Zoom In<disabledText />
          </tooltip>
        </button>
      </controls>
    </insertModule>
  </modules>
</ArcGIS>
```

ArcGIS Pro .NET SDK: The Road Ahead - Addin Architecture

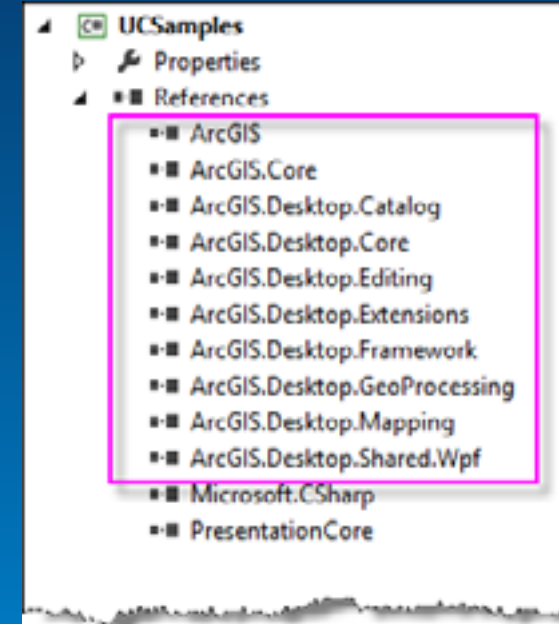
- Implementation, Compilation

```
using ArcGIS.Core.CIN;
using ArcGIS.Core.Geometry;
using ArcGIS.Desktop.Extensions;
using ArcGIS.Desktop.Framework;
using ArcGIS.Desktop.Framework.Contracts;
using ArcGIS.Desktop.Framework.Threading.Tasks;
using ArcGIS.Desktop.Internal.Mapping;
using ArcGIS.Desktop.Mapping;

namespace UCSamples.FixedZoom
{
    /// <summary>
    /// The button shows zoom-in functionality by manipulating the camera.
    /// In 2D view mode we are changing the scale property whereas in 3D the z-value
    /// of the observer (eye).
    /// </summary>
    internal class FixedZoomIn : Button {}
        private static readonly double MinimumScale = 10000.0;
        private static readonly double MinimumElevation = 1000;

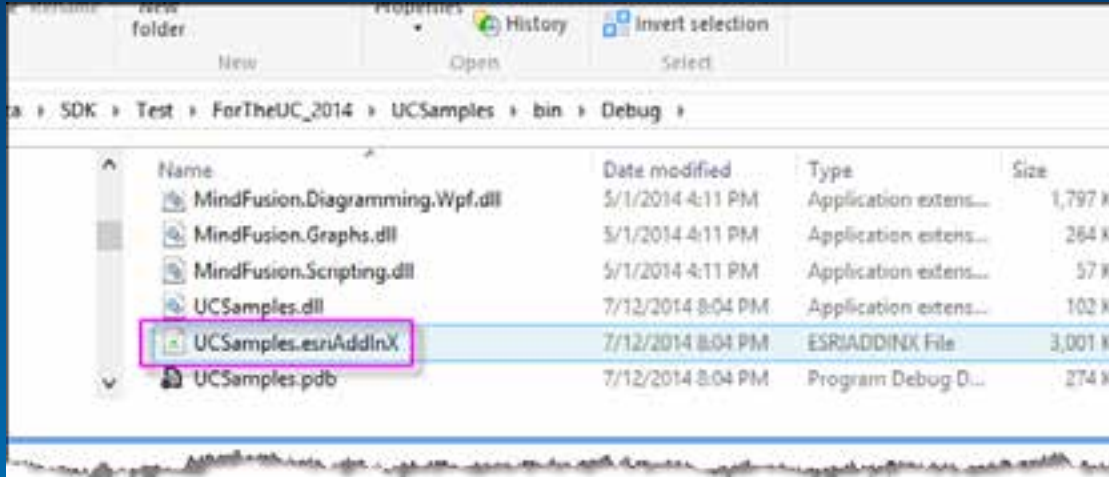
        protected override void OnClick() {
            MapView activeMapView = ForTheUcModule.ActiveMapView;
            Camera camera = activeMapView.Camera;

            if (activeMapView.Is2D) {
                double scale = camera.Scale * 0.75;
                camera.Scale = scale > MinimumScale ? scale : MinimumScale;
            }
            else {
                double z = camera.EyeXYZ.Z * 0.75;
                camera.EyeXYZ.Z = z > MinimumElevation ? z : MinimumElevation;
            }
            activeMapView.Camera = camera;
        }
    }
}
```

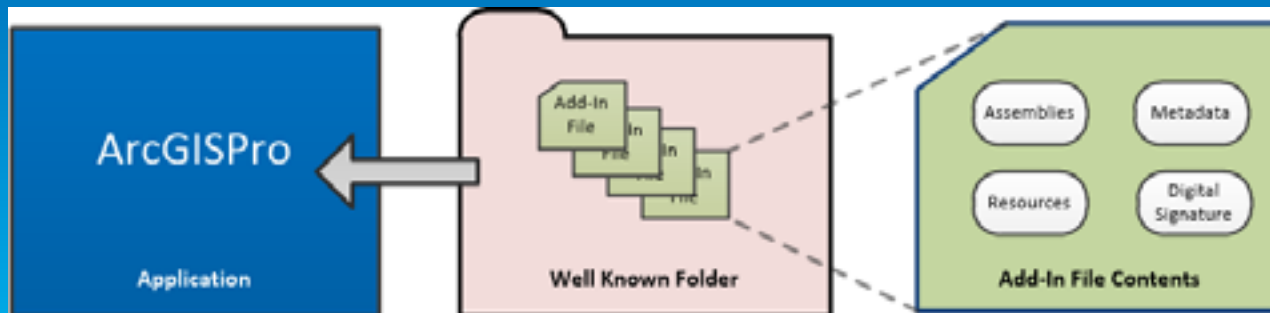


ArcGIS Pro .NET SDK: The Road Ahead - Addin Architecture

- **Deployment.** Daml and code compiled into an Addin package:

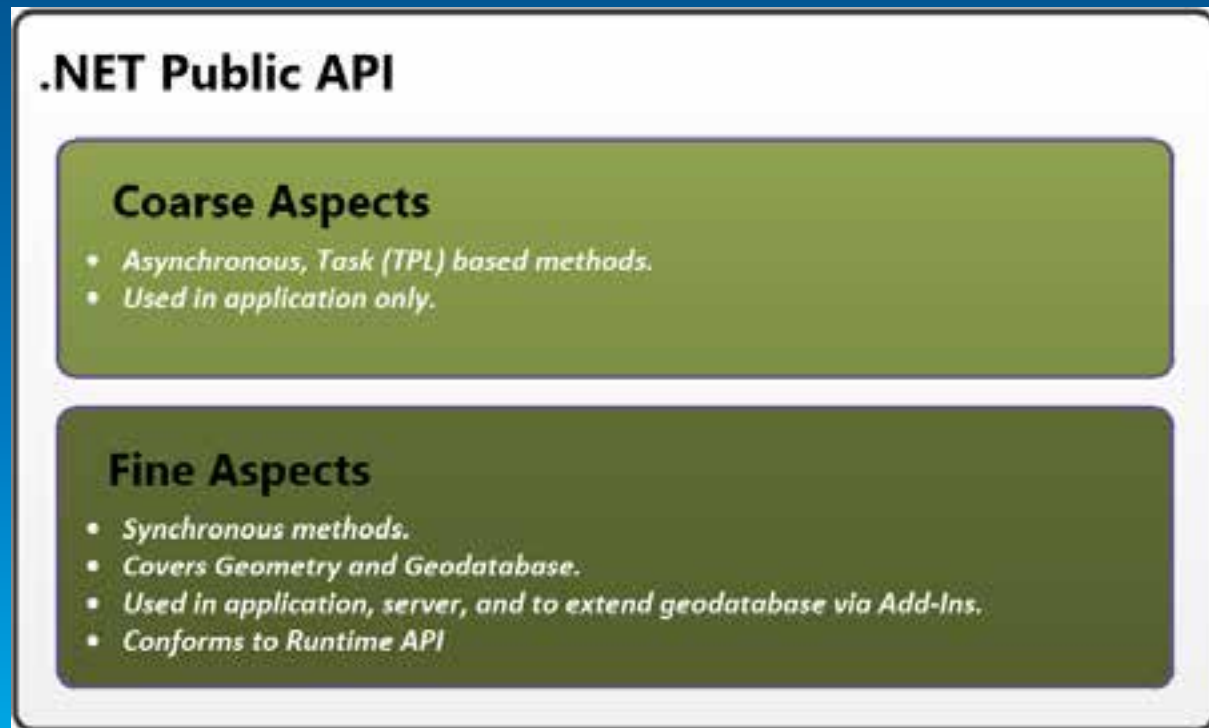


Deployment: The Addin is copied to a well known folder location where it will be loaded by Framework



ArcGIS Pro .NET SDK: The Road Ahead – SDK and API

- **No need for “external” PIAs like with Arcobjects. Simply install ArcGIS Pro.**
- **At Beta4, APIs are available for:**
 - Framework
 - Content management
 - Editing
 - Geometry
 - Geodatabase
 - Geoprocessing
- **At Final, APIs will be available for:**
 - Map Authoring, Visualization, Exploration
 - Symbology, Styles
 - Task
 - Workflow Manager
 - Raster layers, Colorizers



ArcGIS Pro .NET SDK: The Road Ahead – SDK and API

- **At Final:**
 - **No APIs for:**
 - Facility Network
 - Network Analysis
 - Labelling
 - Data Reviewer
 - **APIs still undecided for:**
 - Data Management
 - 3D Analysis (TIN, LAS)
 - Layout

The diagram is a white rounded rectangle with a black border. At the top, it is titled ".NET Public API". Below the title are two stacked, rounded rectangular boxes. The top box is light green and titled "Coarse Aspects", containing two bullet points: "Asynchronous, Task (TPL) based methods." and "Used in application only." The bottom box is a darker green and titled "Fine Aspects", containing four bullet points: "Synchronous methods.", "Covers Geometry and Geodatabase.", "Used in application, server, and to extend geodatabase via Add-Ins.", and "Conforms to Runtime API".

.NET Public API

Coarse Aspects

- *Asynchronous, Task (TPL) based methods.*
- *Used in application only.*

Fine Aspects

- *Synchronous methods.*
- *Covers Geometry and Geodatabase.*
- *Used in application, server, and to extend geodatabase via Add-Ins.*
- *Conforms to Runtime API*

ArcGIS Pro .NET SDK: The Road Ahead – APIs

- **ArcGIS Pro API Workflows at 1.0, 1.1:**
 - 2D and 3D editing
 - 2D and 3D Map interactions (author, query, time, visualize)
 - Content management and sharing (online, portals, web maps)
 - Fine grained APIs for Geometry and Geodatabase
 - Leverage .Net language features and async patterns
 - Etc, etc.
- **What is not supported in the API for Pro?:**
 - Custom data source and feature class extensions
 - Custom extensions (edit, feature class, map)
 - Custom Raster functions
 - Custom GP functions
 - Stand-alone “ArcEngine” applications

ArcGIS Pro Extensibility - Addins for the new DotNet SDK

Patterns

- **MVVM – Pattern for many of the Framework elements**
 - Dockpane, Pane, Custom Control, Property Page
- **TAP – Framework’s managed threading model built on top of TAP**
 - Asynchrony primarily focused on maintaining a responsive UI
 - Framework provides `QueuingTaskFactory` to guarantee that UI actions happen in a sensible order without corruption
 - To implement parallelism, use `System` threads but no Framework support

ArcGIS Pro Extensibility - Addins for the new DotNet SDK

ArcGIS.Desktop.Internal.XXXXX namespaces

- **Reserved for Esri Internal use only**
- **Publicly scoped code (required for Esri modules to share code)**
 - **Shared-Internal code is NOT intended for the API**
 - **Not supported**
 - **API code is developed within Internal namespaces also**
- **UC samples are using the internal namespaces**
 - **Conscious decision to take advantage of functionality not yet released to the “public” namespaces**
 - **Samples delivered with Final will NOT be using Internal namespaces**

ArcGIS Pro .NET SDK: The Road Ahead – **Session Resources**

- **Samples available at <https://github.com/Esri/arcgis-pro-samples-beta4>**

These slides available at:

- **Release schedule:**
 - **EAP now**
 - **Beta in November**
 - **Release 1.1 in March**

Thank you...

ArcGIS Pro .NET SDK: The Road Ahead

- **Please fill out the session survey:**

Online – www.esri.com/ucsessionsurveys

Paper – pick up and put in drop box

Q&A, At the Developer Showcase, Desktop Development Area



Understanding our world.

ArcGIS Pro UI Elements - Button

- Buttons are the least complex of all controls, and are declared in DAML using the *insertButton* element within the *controls* container element.

```
<button id="esri_mapping_showLocateDockPane" className="esri_mapping:ShowFindLocation"
        largeImage="pack://application:,,,/ArcGIS.Desktop.Resources;component/Images/Find32.png"
        smallImage="pack://application:,,,/ArcGIS.Desktop.Resources;component/Images/Find16.png"
        caption="Locate" condition="esri_mapping_mapPane">
  <tooltip heading="Search">
    Find locations by typing in an address, place name or coordinate values.<disabledText></disabledText>
  </tooltip>
</button>
```

The XAML declaration above for the Locate button in ArcGIS Pro includes a few of the most common attributes associated with buttons.



ArcGIS Pro UI Elements – Button Palette

- Button palettes are designed specifically for grouping a set of related buttons together. When a button is selected from the palette of buttons, it becomes the active button and is displayed on the collapsed palette.

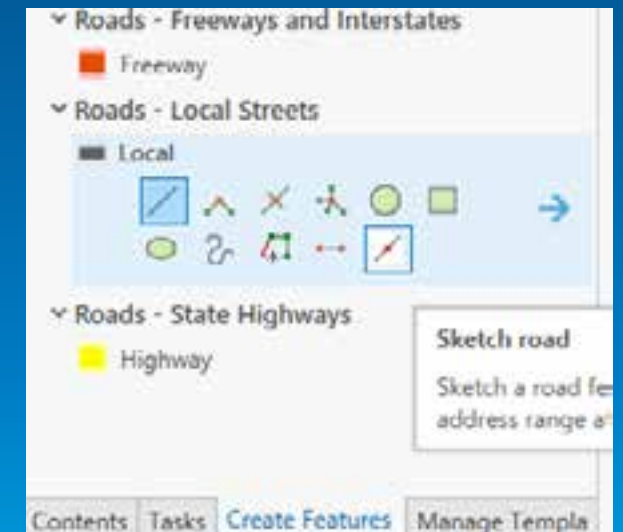
```
<buttonPalette id="esri_mapping_syncViewExtentPalette" caption="Link Views" dropDown="false" menuStyle="false" itemsInRow="1">  
  <button refID="esri_mapping_syncCenterButton" />  
  <button refID="esri_mapping_syncCenterAndScaleButton" />  
</buttonPalette>
```



ArcGIS Pro UI Elements – Construction Tool

- Construction tools are defined in DAML using the tool element. In addition you also need to specify which feature type the construction tool will create. This is achieved by inserting a reference to your construction tool into the categories element updating one of the predefined editing EditTools categories. The example below creates and registers a point construction tool.

```
<categories>
  <updateCategory refID="esri_editing_EditTools_POINT">
    <insertComponent id="ProAppModule5_ConstructionTool1">
      <content guid="1919703d-c64b-427b-be4b-547d79097c89" group="esri_editing_EditTools_POINT_Tools" />
    </insertComponent>
  </updateCategory>
</categories>
<controls>
  <tool id="ProAppModule5_ConstructionTool1" caption="ConstructionTool 1" className="ConstructionTool1"
    loadOnClick="true" smallImage="Images\GenericButtonRed16.png" largeImage="Images\GenericButtonRed32.png">
    <tooltip heading="Tooltip Heading">
      Tooltip text<disabledText />
    </tooltip>
  </tool>
</controls>
```



ArcGIS Pro UI Elements - Dockpane

- **Dock panes are modeless dialogs which can be docked at the top, right, left, or bottom of the view area within the application. Dock panes can also be grouped with other dock panes and docked relative to each other (i.e. below, above, etc), and can also be un-docked and floated. Finally, dock panes can also be pinned/un-pinned so that they slide back into the frame to save space. For each dock pane—each a logical singleton—the framework persists and preserves docking state so that when shown in subsequent sessions, dock panes appear in the same positions and in the same state.**

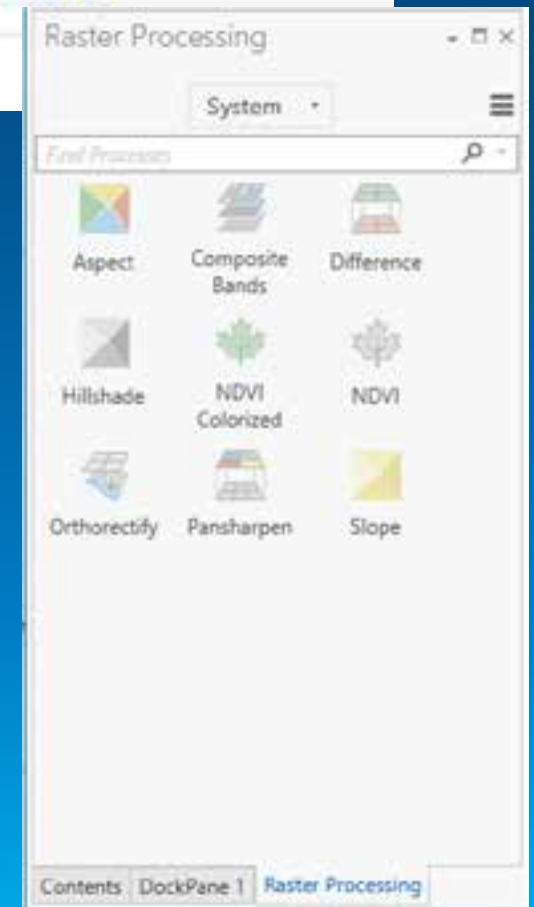
ArcGIS Pro UI Elements – Dockpane (continued)

- **The example below shows the declaration of a dock pane. This dock pane also specifies a content element which only has a className attribute. When the framework creates the dock pane, it will also create its content class and it will set the content's data context to the dock pane. This automatic binding allows you to use the MVVM pattern when developing panes and dock panes. For example, a well-designed dock pane will have as little logic as possible in its XAML code behind; instead, all of the controls in the XAML should be bound to properties in the dock pane base class. The dock pane and pane bases classes should be regarded as view-models.**

ArcGIS Pro UI Elements – Dockpane (continued)

```
<dockPane id="esri_mapping_rasterProcessDockPane" caption="Raster Processing" className="RasterProcess.RasterProcessDockPaneViewModel"
  dock="group" condition="esri_mapping_mapPane" dockWith="esri_core_projectDockPane" initiallyVisible="false"
  showLoadingMessage="true" minWidth="400" disableIfBusy="true">
  <content className="RasterProcess.RasterProcessDockPaneView" />
</dockPane>
```

Dock panes can be declared such that they are positioned relative to other dock panes using the *dockWith* attribute. Dock panes can also be established with a condition so that they only appear when the condition is satisfied.



ArcGIS Pro UI Elements - Gallery

- **The Gallery is a new type of control available for the ribbon that behaves similarly to menus. Galleries typically provide a richer representation of the choices offered, each often representing a preview of the result if chosen. Galleries can be organized to show multiple rows and columns simultaneously and are excellent choices when you don't want to be constrained by the smaller one dimensional area offered by a menu.**
- **Galleries can present a condensed grid within the ribbon itself using the in-line gallery representation. The items presented in this way are often either the most common or most recently used items depending on the implementation. The actual contents of a gallery are normally populated at runtime. The Gallery declaration below is populated entirely at runtime. Relatively static aspects such as the caption, the dropdown image, item size constraint, tooltip, etc. are specified declaratively. The `itemSizeString` is used to specify the maximum width of items displayed in the gallery.**

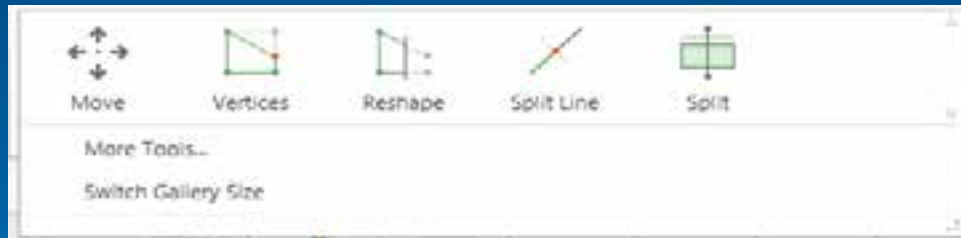
ArcGIS Pro UI Elements – Gallery (Continued)

```
<gallery id="esri_mapping_baseemapGallery" className="Ribbon.BaseemapGalleryViewModel" caption="Basemap" itemsInRow="3" helpContextID=""
  loadingMessage="Loading..." itemWidth="140" itemHeight="115"
  dataTemplateFile="pack://application:,,,/ArcGIS.Desktop.Mapping;component/Map/Ribbon/GalleryTemplates.xaml"
  templateID="BasemapItemTemplate" showItemCaption="true" showItemCaptionBelow="true" resizable="true"
  condition="esri_mapping_BaseemapGalleryCondition"
  largeImage="pack://application:,,,/ArcGIS.Desktop.Resources;component/Images/Baseemap32.png">
  <tooltip heading="">
    Choose a basemap for your map. The basemap is the reference data that displays under the notes and other GIS data you have added to the map.
    <disabledText></disabledText>
  </tooltip>
</gallery>
```



ArcGIS Pro UI Elements – Inline-Gallery

- Inline galleries are very similar to the standard gallery in that they provide a set of choices to the user, however rather than using a drop-down representation the choices are presented in a horizontal condensed grid within the ribbon itself.

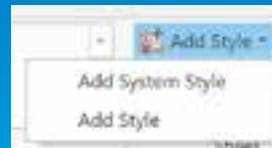


```
<group id="esri_editing_EditGallery" caption="Tools">
  <gallery refID="esri_EditFeaturesGallery" inline="true" size="large"/>
</group>
<gallery id="_esri_EditFeaturesGallery" className="EditFeatures.GalleryCommandListViewModel" caption="Editor tool gallery" itemWidth="72"
  itemsInRow="6" showItemCaption="true" showGroup="false" showItemCaptionBelow="true" resizable="true"
  largeImage="pack://application:,,,/ArcGIS.Desktop.Resources;component/Images/EditingToolBarShow32.png"
  condition="esri_editing_ShowEditFeatures" helpContextID="">
  <tooltip heading="">
    View a gallery of editing tools.<disabledText></disabledText>
  </tooltip>
  <button refID="esri_editing_ShowEditFeaturesFromGallery" />
  <button refID="esri_editing_ToggleEditFeaturesGallerySize" />
</gallery>
```

ArcGIS UI Elements - Menu

- When initially defined, all menu items are simply listed within the menu element in the order they should appear. Legal elements within a menu include buttons (simple menu items), other menus (pull rights), dynamic menus, galleries, and split buttons.

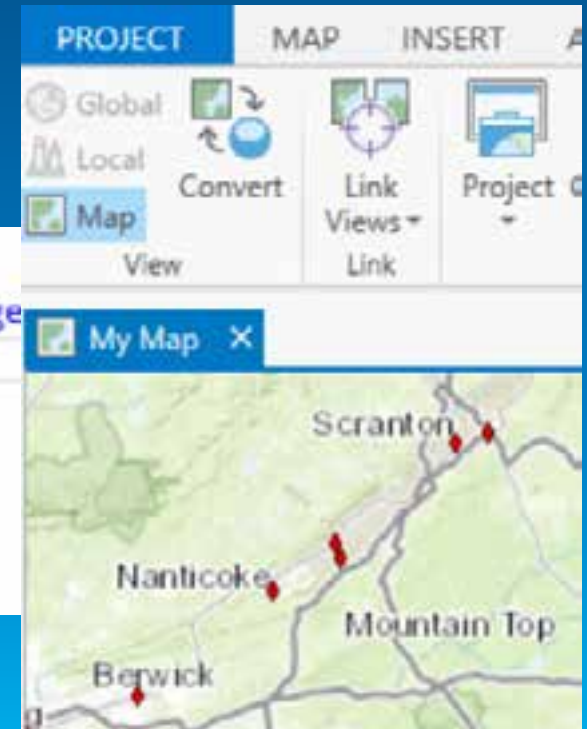
```
<menu id="esri_mapping_addStyleMenu" caption="Add Style"  
  smallImage="pack://application:,,,/ArcGIS.Desktop.Resources;component/Images/StylexFileAdd16.png"  
  largeImage="pack://application:,,,/ArcGIS.Desktop.Resources;component/Images/StylexFileAdd32.png">  
  <button refID="esri_mapping_addCoreStyleButton" />  
  <button refID="esri_mapping_addCustomStyleButton" />  
  <tooltip heading="Add Style">Add a system style or a custom style to the project.</tooltip>  
</menu>
```



ArcGIS Pro UI Elements - Pane

- The framework supports multiple panes, letting users display and interact with multiple subjects. Only one pane can be active at a time. The active pane establishes what is available on the ribbon; switching between different panes may result in changes to what tabs and controls are available. You can open many pane at the same time and these can be grouped, tiled (horizontally or vertically).

```
<pane id="esri_mapping_mapPane" caption="Map" className="MapPaneViewModel"
      smallImage="pack://application:,,,/ArcGIS.Desktop.Resources;component/Image
      isClosable="true"
      defaultTab="esri_mapping_homeTab"
      defaultTool="esri_mapping_navigateTool">
  <content className="MapPaneView" />
</pane>
```



ArcGIS Pro UI Elements – Split Button

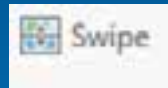
- Split buttons group related buttons together. The associated group is displayed when the arrow portion of the split button is clicked. Once a control is selected from the drop-down list, it becomes the active control in the split button. Split Buttons can contain either simple button controls, or galleries

```
<splitButton id="esri_mapping_layerSymbologySplitButton">  
  <button refID="esri_mapping_showLayerSymbologyDockPane"/>  
  <gallery refID="esri_mapping_layerSymbologyGallery" />  
</splitButton>
```



ArcGIS Pro UI Elements - Tool

- Tools in ArcGIS Pro primarily just put the viewer into a particular mode and are therefore very similar to buttons except they automatically stay checked after being selected and they are closely coupled with panes.



```
<tool id="esri_mapping_layerSwipeTool" className="FeatureLayers.Ribbon.SwipeTool"
  largeImage="pack://application:,,,/ArcGIS.Desktop.Resources;component/Images/EffectsSwipe32.png"
  smallImage="pack://application:,,,/ArcGIS.Desktop.Resources;component/Images/EffectsSwipe16.png" caption="Swipe"
  condition="esri_mapping_singleLayerSelectedCondition">
  <tooltip heading="Swipe">
    Drag to reveal layers beneath the chosen layer. Swipe let you see what is underneath a particular layer
    without having to turn it off in the Contents pane.<disabledText></disabledText>
  </tooltip>
</tool>
```