ArcGIS Pro SDK for .NET: An Introduction

Wolf Kaiser

Uma Harano

GIS INSPIRING WHAT'S NEXT

Session Overview

- Extensibility patterns supported by the Pro SDK
  - ArcGIS Pro Module Add-ins
  - ArcGIS Pro Managed Configurations
- Asynchronous Programming: Introduction to QueuedTask
  - Use of async and await
  - Authoring custom asynchronous functions
- Overview of MVVM (Model View View Model)
  - Dockpane example
- Using existing ArcGIS Pro functionality in your Add-in

What is the ArcGIS Pro SDK for .NET?

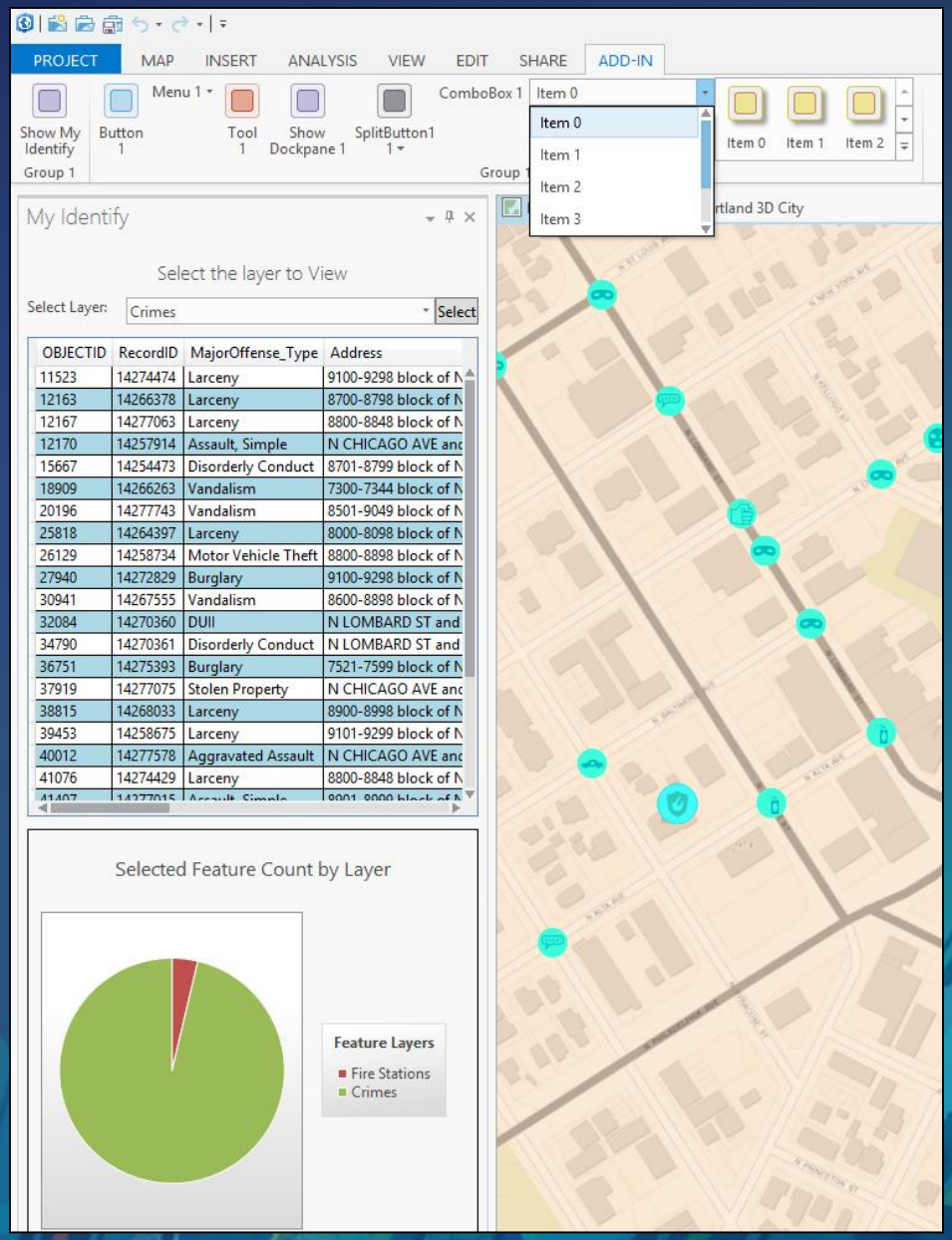
- ArcGIS Pro SDK for .NET extends ArcGIS Pro using .NET
  - Two extensibility patterns
    - ArcGIS Pro Module Add-ins
    - ArcGIS Pro Managed Configurations
  - Includes Visual Studio project and item templates.
  - Installation is integrated with Visual Studio Marketplace
  - ArcGIS Pro API comes with ArcGIS Pro out-of-box
    - File based references (No Global Assembly Cache)

What is the ArcGIS Pro SDK for .NET? (Continued)

- ArcGIS Pro SDK features and patterns
  - 64-bit platform
  - WPF (Windows Presentation Foundation) with .NET 4.6.1 +
  - MVVM pattern (Model View ViewModel)
  - Asynchronous Patterns: Multiple threads

What is an ArcGIS Pro add-in ?

- Extends ArcGIS Pro through:
  - Buttons
  - Tools
  - Dockpanes
  - Embeddable control
  - ..
- Packaged within a single, compressed file with an .esriaddinX file extension
  - C:\Users\<UserName>\Documents\ArcGIS\AddIns\ArcGISPro



What is an ArcGIS Pro add-in? (Continued)

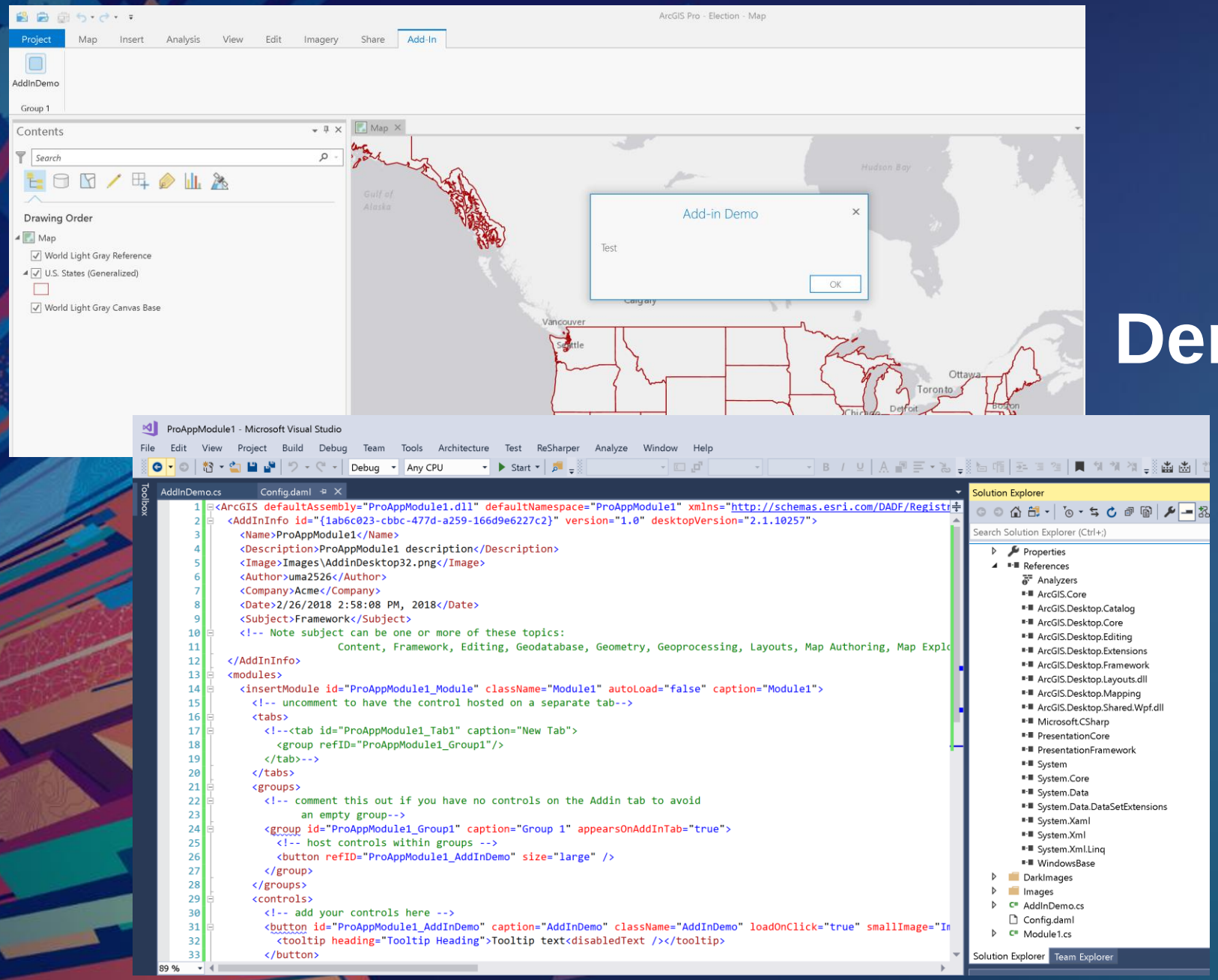
- Uses a declarative-based framework (config.daml)
- The Module class
  - Hub and central access point for a subsystem
  - Similar to the Extension object used in the ArcObjects 10.x framework
  - Singletons instantiated automatically by the Framework

Config.daml

- Declarative add-in definition
- Configuration file that contains framework elements (buttons, dockpane, galleries)
- Can add, modify, delete any default ArcGIS Pro User Interface element
- Uses XML Syntax

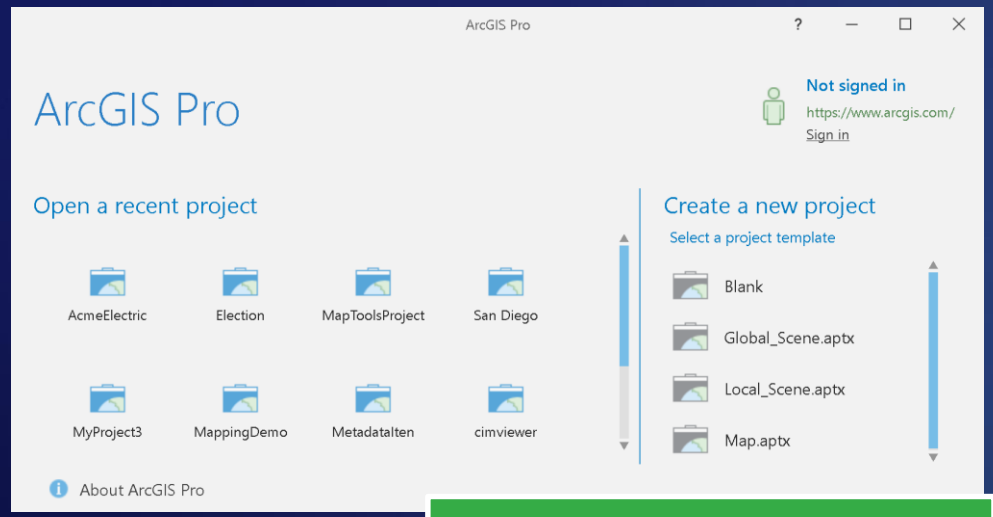
```
<ArcGIS defaultAssembly="WorkingWithDAML.dll" defaultNamespace="WorkingWithDAML"
xmlns="http://schemas.esri.com/DADF/Registry"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://schemas.esri.com/DADF/Registry
file:///C:/Program%20Files/ArcGIS/Pro/bin/ArcGIS.Desktop.Framework.xsd">
  <AddInInfo id="{c6226a24-d69b-46c6-b5e7-9eee4ddad45d}" version="1.0" desktopVersion="1.1.2850">
…
</AddInInfo>
<modules>
    <insertModule id="WorkingWithDAML" className="Module1" autoLoad="false" caption="Module1">
      <tabGroups>
        <!--The new Tab Group is created here-->
        <tabGroup caption="Example State Solution" id="working_with_DAML_ExampleStateTabGroup">
          <color A="255" R="238" G="170" B="90" />
          <borderColor A="0" R="251" G="226" B="195" />
        </tabGroup>
      </tabGroups>
```

Demo: Add-in

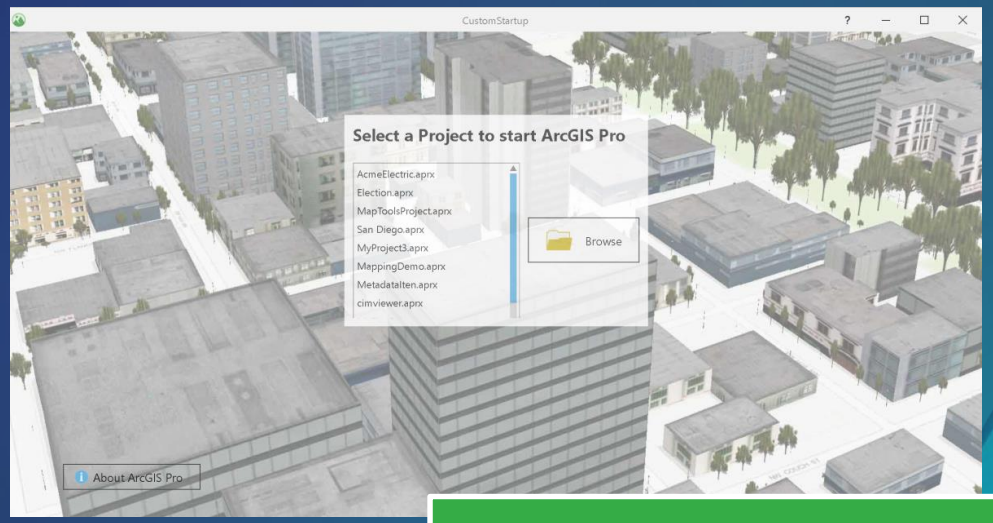

ArcGIS Pro - Election - Map
Project
Map
Insert
Analysis
View
Edit
Imagery
Share
Add-In
AddInDemo
Group 1
Contents
Search
Drawing Order
Map
World Light Gray Reference
U.S. States (Generalized)
World Light Gray Canvas Base
Add-in Demo
Test
OK
Vancouver
Ottawa
Toronto
ProAppModule1 - Microsoft Visual Studio
File
Edit
View
Project
Build
Debug
Team
Tools
Architecture
Test
ReSharper
Analyze
Window
Help
Debug
Any CPU
Start
AddInDemo.cs
Config.daml
<ArcGIS defaultAssembly="ProAppModule1.dll" defaultNamespace="ProAppModule1" xmlns="http://schemas.esri.com/DADF/Registr
<AddInInfo id="{1ab6c023-cbbc-477d-a259-166d9e6227c2}" version="1.0" desktopVersion="2.1.10257">
<Name>ProAppModule1</Name>
<Description>ProAppModule1 description</Description>
<Image>Images\AddinDesktop32.png</Image>
<Author>uma2526</Author>
<Company>Acme</Company>
<Date>2/26/2018 2:58:08 PM, 2018</Date>
<Subject>Framework</Subject>
<!-- Note subject can be one or more of these topics:
Content, Framework, Editing, Geodatabase, Geometry, Geoprocessing, Layouts, Map Authoring, Map Explo
</AddInInfo>
<modules>
<insertModule id="ProAppModule1_Module" className="Module1" autoLoad="false" caption="Module1">
<!-- uncomment to have the control hosted on a separate tab-->
<tabs>
<!--<tab id="ProAppModule1_Tab1" caption="New Tab">
<group refID="ProAppModule1_Group1"/>
</tab>-->
</tabs>
<groups>
<!-- comment this out if you have no controls on the Addin tab to avoid
an empty group-->
<group id="ProAppModule1_Group1" caption="Group 1" appearsOnAddInTab="true">
<!-- host controls within groups -->
<button refID="ProAppModule1_AddInDemo" size="large" />
</group>
</groups>
<controls>
<!-- add your controls here -->
<button id="ProAppModule1_AddInDemo" caption="AddInDemo" className="AddInDemo" loadOnClick="true" smallImage="In
<tooltip heading="Tooltip Heading">Tooltip text<disabledText /></tooltip>
</button>
Solution Explorer
Search Solution Explorer (Ctrl+;)
Properties
References
Analyzers
ArcGIS.Core
ArcGIS.Desktop.Catalog
ArcGIS.Desktop.Core
ArcGIS.Desktop.Editing
ArcGIS.Desktop.Extensions
ArcGIS.Desktop.Framework
ArcGIS.Desktop.Layouts.dll
ArcGIS.Desktop.Mapping
ArcGIS.Desktop.Shared.Wpf.dll
Microsoft.CSharp
PresentationCore
PresentationFramework
System
System.Core
System.Data
System.Data.DataSetExtensions
System.Xaml
System.Xml
System.Xml.Linq
WindowsBase
DarkImages
Images
AddInDemo.cs
Config.daml
Module1.cs
Solution Explorer
Team Explorer

What is an ArcGIS Pro Managed Configuration?

- **Includes all functionality of an add-in plus**
  - Change the application title and icon
  - Change the application splash, start page, and about page
  - Conditional customization of the UI



Default start page



Custom start page

- **Packaged within a single, compressed file with a .ProConfigX file extension**
  - C:\Users\UserName\Documents\ArcGIS\AddIns\ArcGISPro\Configurations

What is an ArcGIS Pro Managed Configuration?

- **Running an ArcGIS Pro Managed Configuration**
  - **Use the ArcGIS Pro "/config" command-line option**

```
C:\ArcGIS\bin\ArcGISPro.exe /config:Acme
```

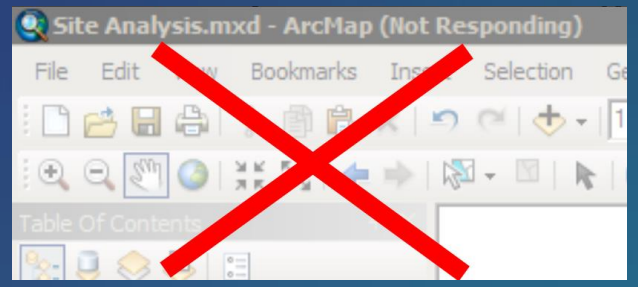
  - **Only one configuration can run per instance of Pro**
- **ConfigurationManager class**
  - **Defined in DAML (generated automatically by the template)**
  - **Provides a set of methods by which a developer can *override* "that" aspect of Pro**

```
public abstract class ConfigurationManager {
  protected internal virtual Window OnShowSplashScreen();
  protected internal virtual FrameworkElement OnShowStartPage();
  protected internal virtual FrameworkElement OnShowAboutPage();
  ...
```

Demo: Configurations

Asynchronous Programming

- ArcGIS Pro is a multi-threaded 64 bit application
- Main motivation to use Asynchronous Programming is to keep the User Interface responsive !



- There are two important asynchronous programming patterns for the SDK:
  - Use of Async / Await functionality in C# or VB .Net
  - Using the ArcGIS Pro Framework's QueuedTask class

ArcGIS Pro Internal Threading Model

- ArcGIS Pro multi-threading incorporates the latest asynchronous language features from Microsoft
- Implements a threading infrastructure tailored to reduce complexity
- ArcGIS Pro SDK developers only need to worry about <u>two</u> threads:
  - The GUI thread (Graphical User Interface thread)
  - A single specialized worker thread called the Main CIM Thread, MCT
    - Internally, ArcGIS Pro uses a large number of threads for:
      - Rasterization, rendering, data loading, Geoprocessing
    - But all this is Isolated from the API and from the developer
- Simplifies coding and ensures the consistency of the ArcGIS Pro state.

Asynchronous Programming

- **To understand Pro multi-threading we have to understand two Categories of Methods in the ArcGIS Pro API**
  - **Coarse-grained asynchronous** methods:
    - Can be called on any thread
    - Once called these methods return to the caller immediately but execute asynchronously in the background
  - **Fine-grained synchronous** methods:
    - Must be called using the QueuedTask class
    - These methods are ‘blocking’ and return to the caller when the method is finished doing its work

Coarse-Grained Methods

- Can be called from any thread, usually called from the GUI thread
- Coarse-grained methods execute in the background on Pro internal threads
- Coarse-grained methods use the ‘Async’ naming convention:
  - method names end with ‘Async’
- Caller uses the async / await semantic

```
//Execute a Geoprocessing Tool
await Geoprocessing.ExecuteToolAsync("SelectLayerByAttribute_management",
     new string[] {"parcels","NEW_SELECTION", "description = 'VACANT LAND'"});
await MapView.Active.ZoomToSelectedAsync(new TimeSpan(0, 0, 3));
```

Demo: Coarse Grained Methods

PlenaryPopup Data

Fine-Grained, Synchronous Methods

Synchronous Methods Must be called using the QueuedTask class

- There is a much greater number of fine grained methods and classes
- No async / await. Runs on the MCT (Main CIM Thread managed by ArcGIS Pro)
- Designed for use in your own coarse-grained async methods
- In other words: this allows you to write your business logic as a ‘background’ task

```
await ArcGIS.Desktop.Framework.Threading.Tasks.QueuedTask.Run(() =>
{
   var layers = MapView.Active.Map.FindLayers("Parcels")
                 .OfType<FeatureLayer>().ToList();
   var parcels = layers[0] as FeatureLayer;
   QueryFilter qf = new QueryFilter()
   {
       WhereClause = "description = 'VACANT LAND'",
       SubFields = "*"
   };
   parcels.Select(qf, SelectionCombinationMethod.New);
});
```

QueuedTask Class

- QueuedTask uses the Pro framework's custom Task scheduler
- Used to run synchronous ArcGIS Pro SDK methods in the background
- Synchronous API methods that require QueuedTask are listed in the API Help and VS IntelliSense using the following text in the description:

  "This method must be called on the MCT. Use QueuedTask.Run"

- Example of synchronous methods in Pro:
  - GetSpatialReference, QueryExtent, Geometry operations
- Usage:

```
Task t = QueuedTask.Run(() =>
{
    // Call synchronous SDK methods here
});
```

Demo: Fine-Grained Methods

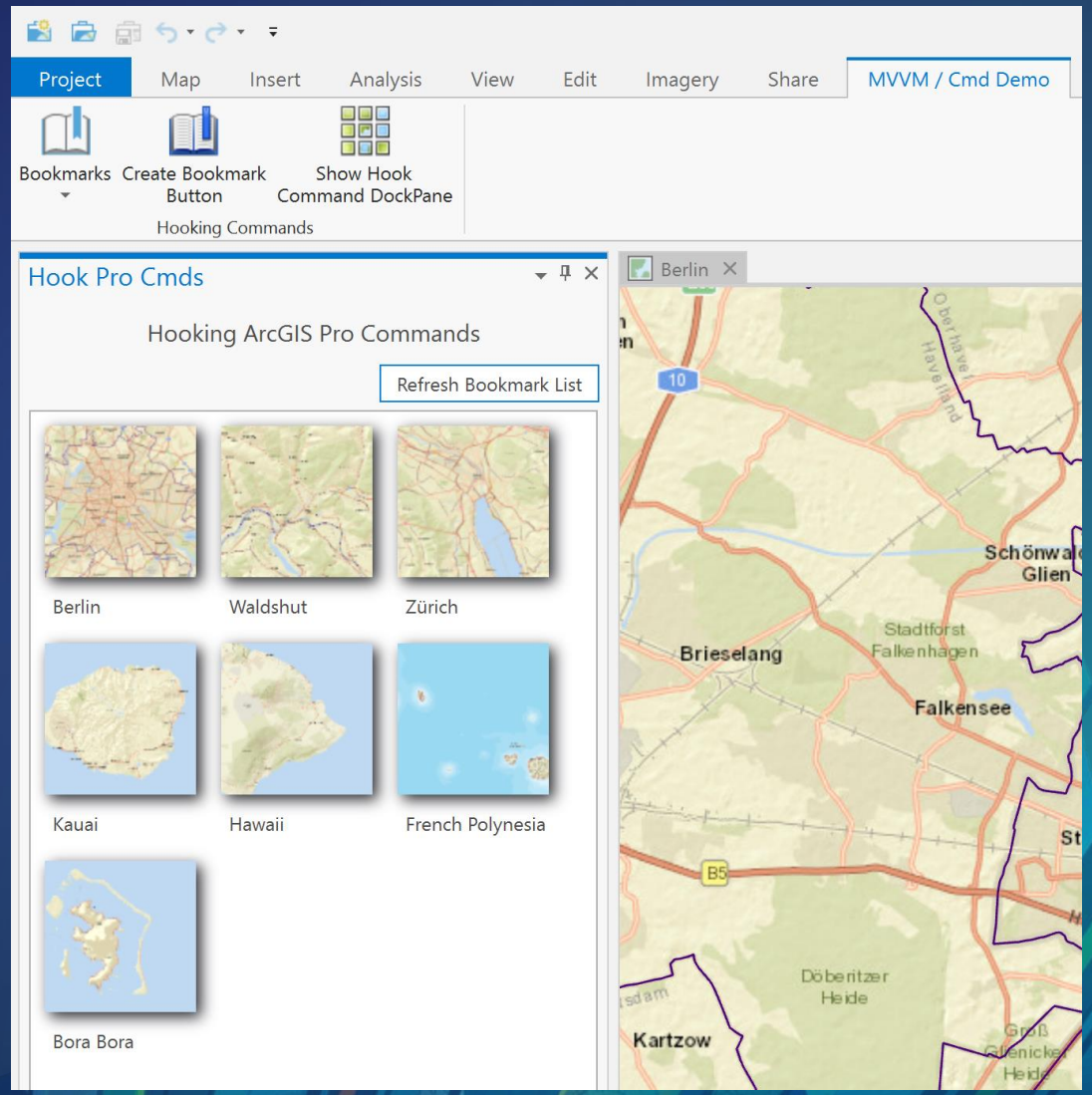
PlenaryPopup Data

MVVM Pattern in Add-ins

- MVVM: Model – View – ViewModel
- MVVM facilitates separation between the GUI development and the business logic or back-end logic coding
- The Basic Pattern is:
  - ViewModel declared in DAML and implemented in code (C# or VB.Net)
  - View referenced in DAML and implemented as WPF UserControl (XAML)
  - Model is optional
- Note: To customize the Pro UI, you must use its MVVM Framework. Substitutes are not allowed.

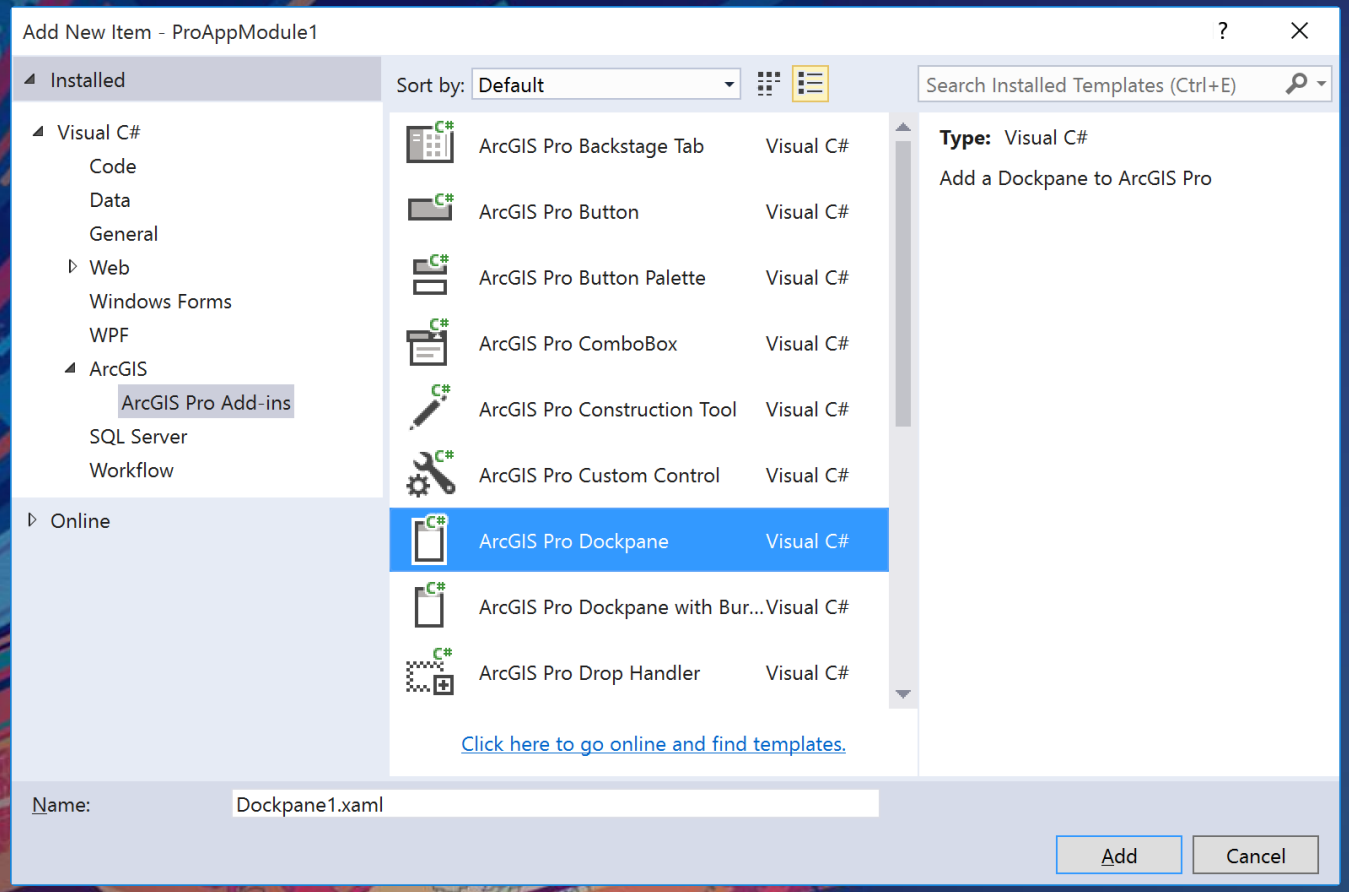
MVVM Pattern in Add-ins

- Model-View-ViewModel (MVVM) Pattern used for many of the Framework elements
  - Dockpane
  - Pane
  - Custom Control
  - Embeddable Control
  - Property Page



MVVM Implementation in Add-ins

- MVVM implementation in Add-ins follows the same Pattern used in other Microsoft WPF / .Net MVVM implementations
- Implement your Add-in UI just as you implement a user control in WPF/.Net
  - You can use many available online WPF MVVM snippets and copy the UI into your Add-in code
- Differences in MVVM for Add-ins versus WPF applications:
  - Multi-threading considerations
  - ArcGIS Pro Styling



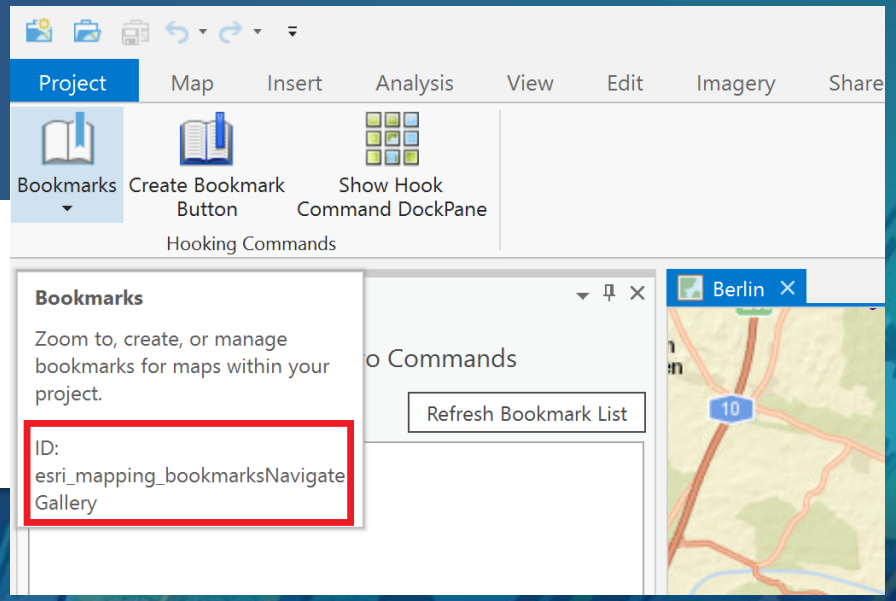
Demo: New Dockpane using MVVM Dockpane template

PlenaryPopup Data

Using Existing ArcGIS Pro Elements in Your Add-in

- Use any ArcGIS Pro framework elements on your Add-in tab including Buttons, Tools, Galleries
- Use the Unique Element Id of any existing ArcGIS Pro Control and add it to your config.daml markup
- Example: Navigate Bookmarks button (see screenshot)
  - DAML button references existing Element ID: esri_mapping_bookmarksNavigateGallery

```
<group id="HookingCommands_Group1" caption="Hooking Commands"
   <button refID="esri_mapping_bookmarksNavigateGallery" />
</group>
```



Use Existing ArcGIS Pro Commands in Your Add-in

- Use the ArcGIS Pro Framework's "GetPlugInWrapper" method with any ArcGIS Pro element's Id to get the IPlugInWrapper interface
- All buttons implement the ICommand interface - with Execute() and CanExecute()

```
// ArcGIS Pro's Create button control DAML ID.
var commandId = DAML.Button.esri_mapping_createBookmark;
// get the ICommand interface from the ArcGIS Pro Button
// using command's plug-in wrapper
// (note ArcGIS.Desktop.Core.ProApp can also be used)
var iCommand = FrameworkApplication.GetPlugInWrapper(commandId) as ICommand;
if (iCommand != null)
{
    // Let ArcGIS Pro do the work for us
    if (iCommand.CanExecute(null))
        iCommand.Execute(null);
}
```

- Using the above pattern you can use any ArcGIS Pro button functionality in your code

ICommand Pattern in MVVM

- Adding a button in MVVM requires the viewmodel to implement the ICommand Interface
  - Use ‘Data Binding’ in the GUI declaration to bind to an ICommand viewmodel property

```
<Button Command="{Binding CloseCmd}" Content="Exit Pro" />
```

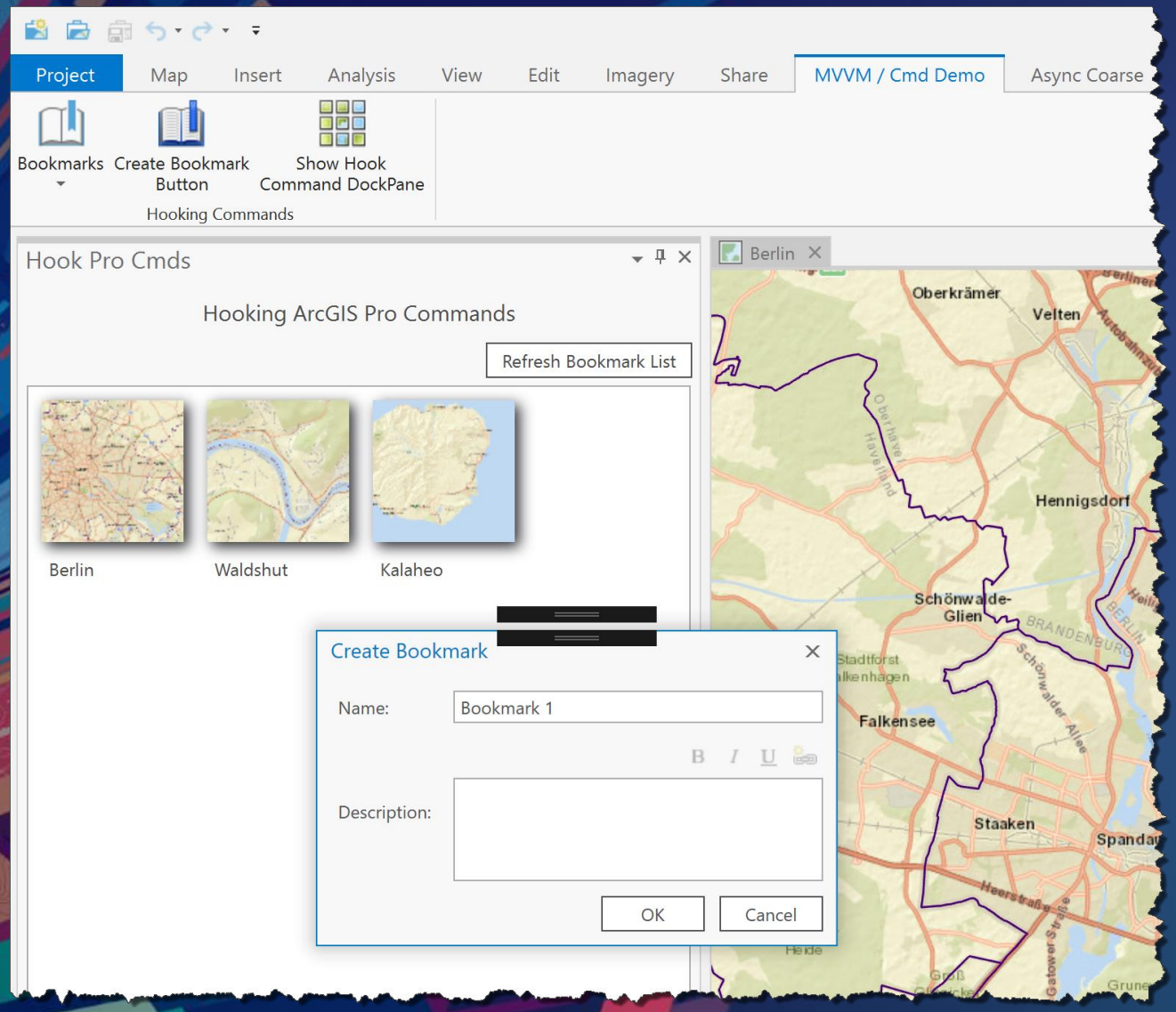
  - Define the corresponding ICommand property in your viewmodel

```
CloseCmd = FrameworkApplication.GetPlugInWrapper(DAML.Button.esri_core_exitApplicationButton)
    as ICommand;

public ICommand CloseCmd { get; set; }
```

- Adding a button to the Dockpane with our ‘custom’ behavior using RelayCommand

```
ZoomInCmd = new RelayCommand(() => MappingModule.ActiveMapView.ZoomInFixedAsync(),
    () => MappingModule.ActiveMapView != null);

public ICommand ZoomInCmd { get; set; }
```

- RelayCommand implements ICommand and allows you to specify your own implementation of Execute and CanExecute



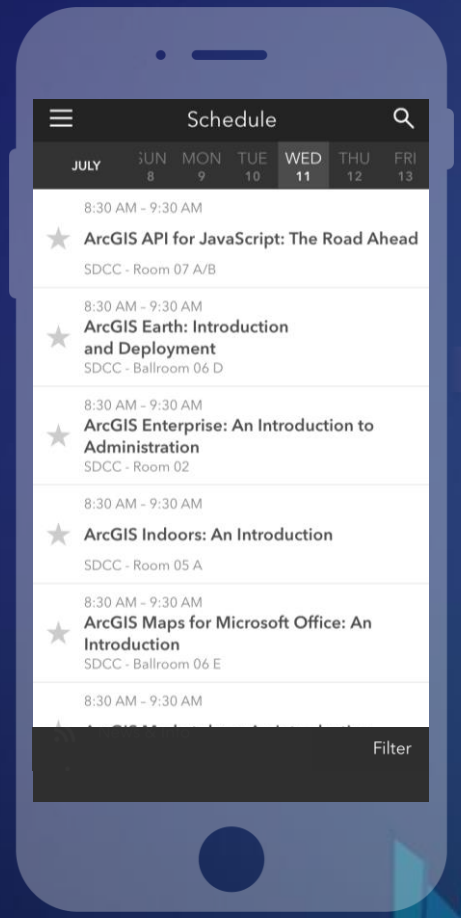
Demo: Hooking Existing

PlenaryPopup Data

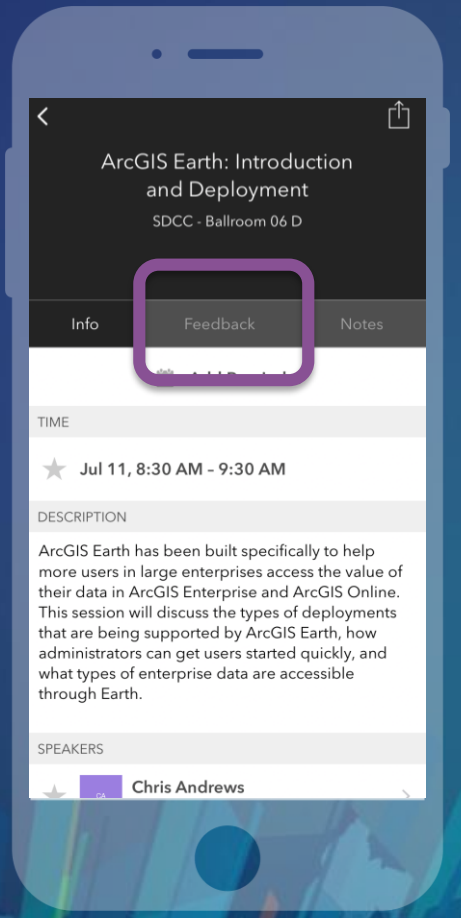
Please Take Our Survey on the App

Download the Esri Events app and find your event

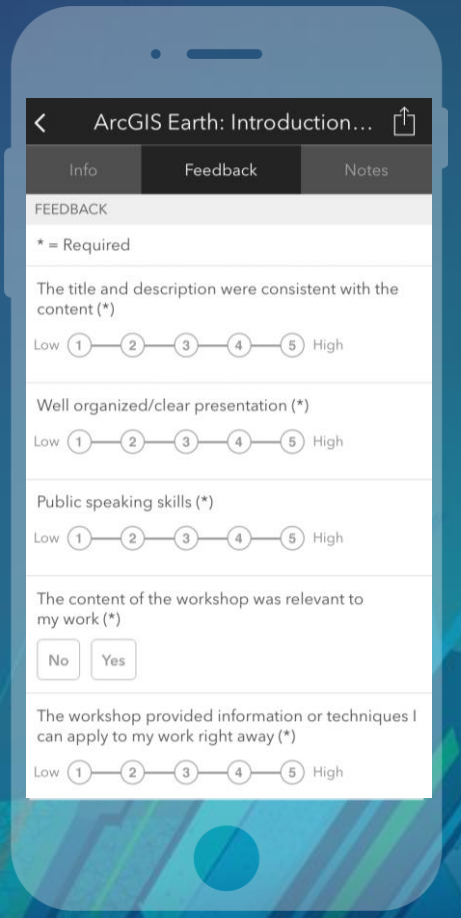
Select the session you attended



Scroll down to find the feedback section



Complete answers and select “Submit”



Pro SDK Sessions

Date	Time	Technical Session	Location
Tue, July 10	10:00 am – 11:00 am	ArcGIS Pro SDK for .NET: An Introduction	Room 14 B, SDCC
	4:00 am – 5:00 pm	ArcGIS Pro SDK for .NET: An Introduction	Room 17 B, SDCC
Wed, July 11	11:15 am - 12:00 pm	ArcGIS Pro SDK for .NET: Introduction to Add-Ins and Configurations	Demo Theater 07 - Expo
	12:15 pm - 1:00 pm	ArcGIS Pro SDK for .NET: Solution Configurations	Demo Theater 08 - Expo
Thu, July 12	8:30 am – 9:30 am	ArcGIS Pro: Q&A with the Development Team	Ballroom 20 D, SDCC
	12:15 pm - 1:00 pm	Beginning ArcGIS Pro SDK Project Development: Tips and Tricks for Troubleshooting	Demo Theater 08 - Expo

Also – visit the ArcGIS Pro Area in the UC Expo –
Pro SDK staff available each day, Tuesday – Thursday!

esri | THE SCIENCE OF WHERE