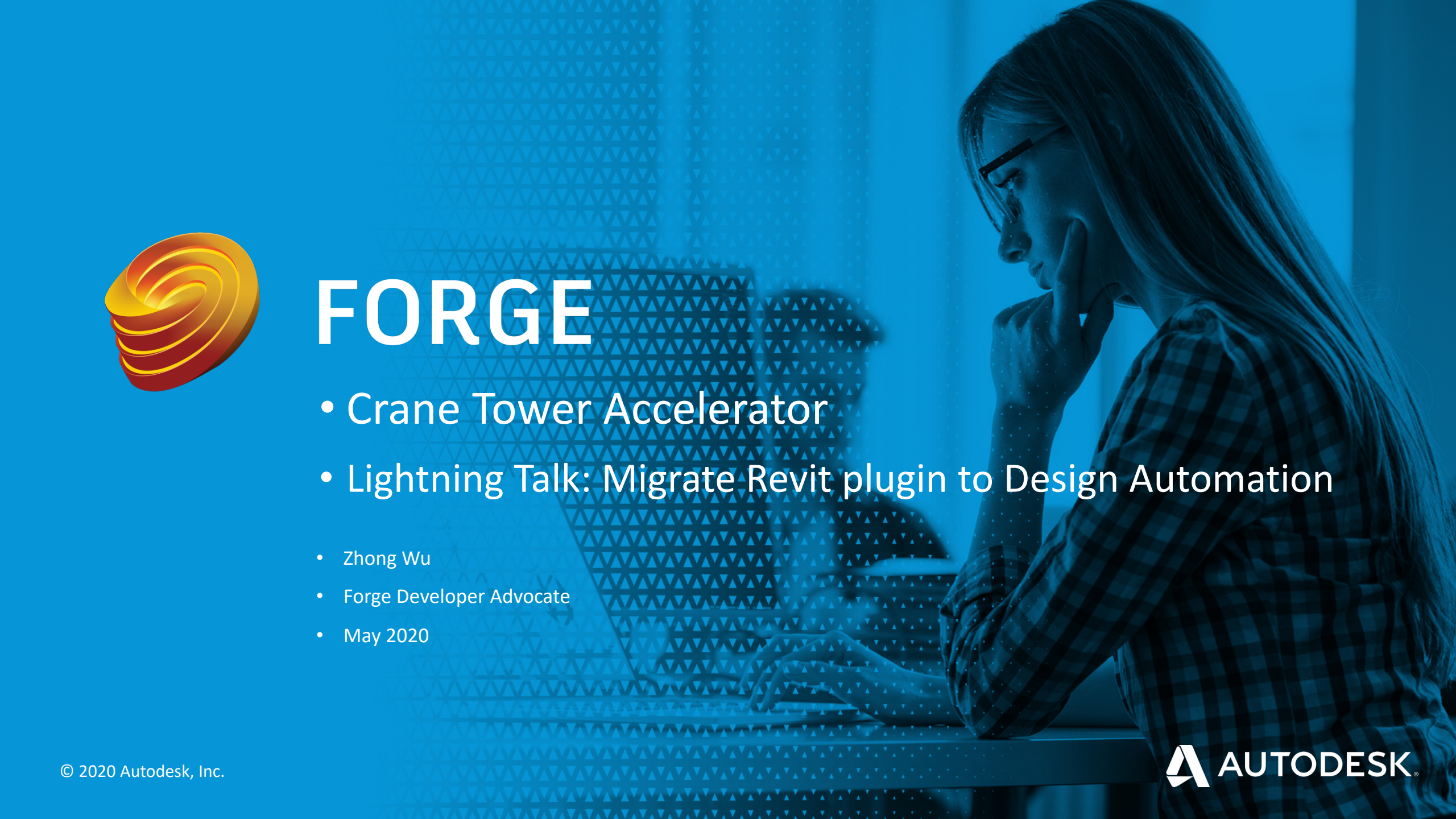


FORGE

- Crane Tower Accelerator
- Lightning Talk: Migrate Revit plugin to Design Automation

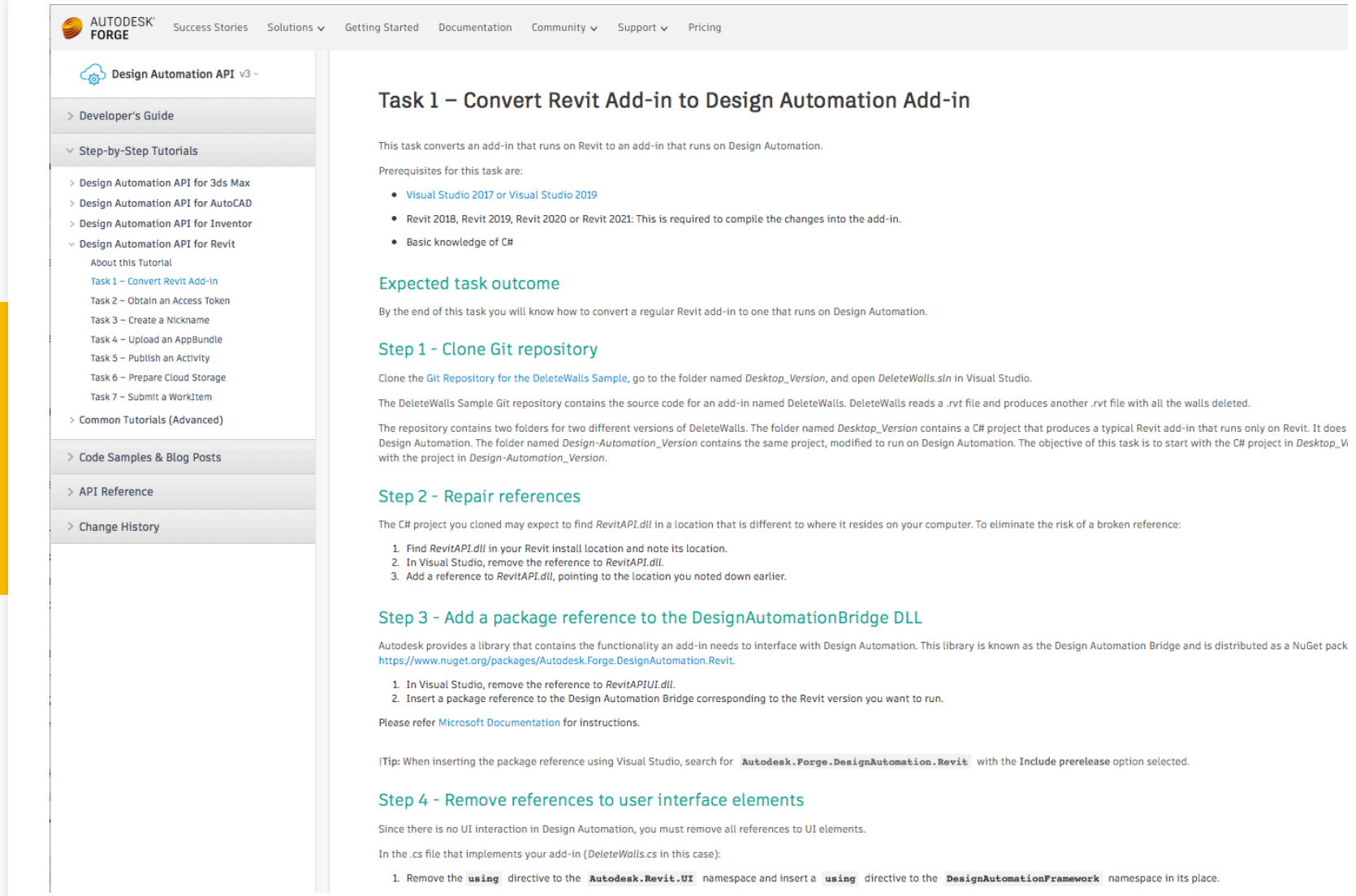
- Zhong Wu
- Forge Developer Advocate

- May 2020

AUTODESK.

Goal

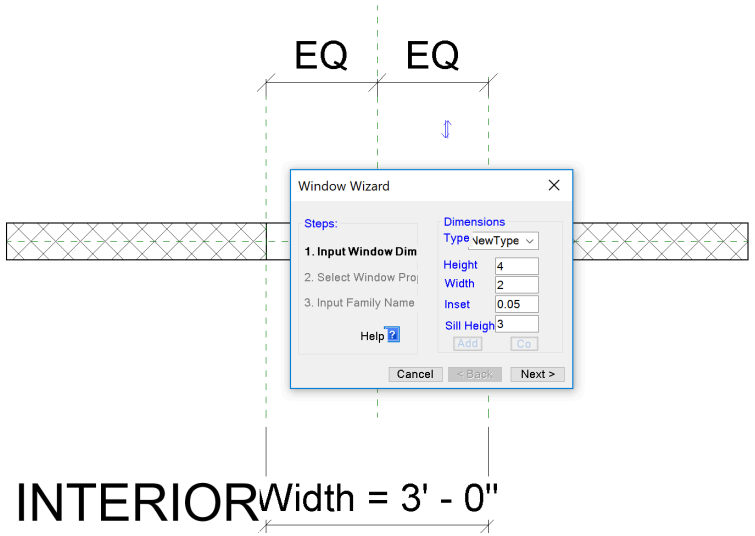
Understand the main tasks to migrate your existing Revit desktop plugin to run on Design Automation engine.

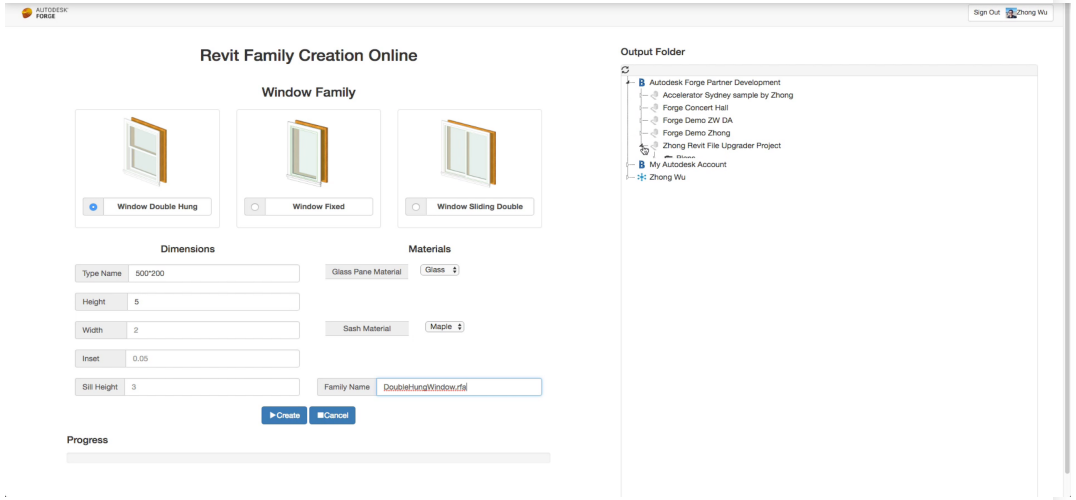


Revit Design Automation Tutorial

\\Revit 2020 SDK\Samples\FamilyCreation\WindowWizard\CS

https://github.com/Autodesk-Forge/forge-createfamily-revit/tree/master/CreateWindow/PlugIn





Plugin Migration Demo

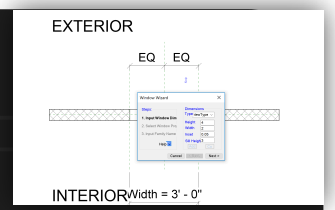
Step 1
Remove UI

- Remove reference to RevitAPIUI.dll, System.Windows.dll
- Remove Autodesk.Revit.UI and System.Windows.Forms
- Remove WizardUI.cs
- UIApplication -> Application
- ExternalCommandData -> CreateWindowData

```
public class CreateWindowData {
    public Application Application { get; set; }
    public Document Document { get; set; }
}
```

Step 2
Handle user input by JSON

- NuGet: Newtonsoft.Json
- Create class TypeDAParams and WindowsDAParams




```
internal class TypeDAParams
{
    public String TypeName { get; set; } = "New Type";
    public Double WindowHeight { get; set; } = 4;
    public Double WindowWidth { get; set; } = 2;
    public Double WindowInset { get; set; } = 0.05;
    public Double WindowSillHeight { get; set; } = 3;

}

internal class WindowsDAParams
{
    public TypeDAParams[] Types { get; set; } = { new TypeDAParams() };

    public String WindowStyle { get; set; } = "DoubleHungWindow";
    public String GlassPaneMaterial { get; set; } = "Default";
    public String SashMaterial { get; set; } = "Default";
    public String WindowFamilyName { get; set; } = "Double Hung.rfa";

    static public WindowsDAParams Parse(string jsonPath)
    {…
    }
}
```

```
"inputJson":{
    "url":"data:application/json,{ 'Types': [{'TypeName': 'My Type', 'WindowHeight': 6, 'Wind
        'WindowSillHeight':3},{'TypeName': 'My New Type', 'WindowHeight': 8, 'WindowWidth': 6
        'WindowSillHeight':6}], 'GlassPaneMaterial': 'Glass', 'SashMaterial': 'Maple', 'Windo
},
```

Step 3

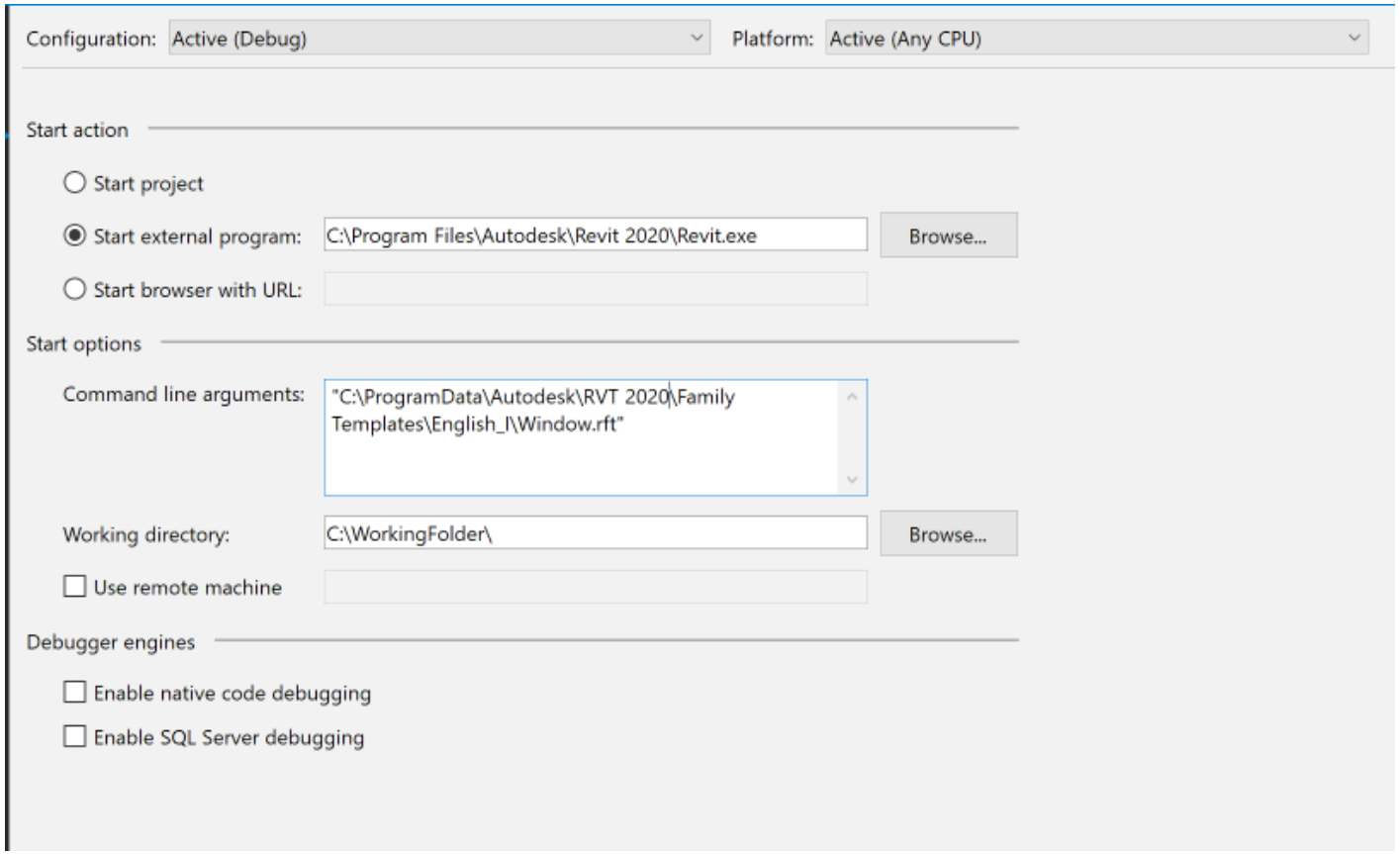
IExternalDBApplication

- Nuget:
  - DesignAutomationBridge.dll,

- IExternalDBApplication

```
[Autodesk.Revit.Attributes.Regeneration(Autodesk.Revit.Attributes.RegenerationOption.Manual)]
[Autodesk.Revit.Attributes.Transaction(Autodesk.Revit.Attributes.TransactionMode.Manual)]
public class CreateWindowApp : IExternalDBApplication
{
    public ExternalDBApplicationResult OnStartup(ControlledApplication application)
    {
        DesignAutomationBridge.DesignAutomationReadyEvent += HandleDesignAutomationReadyEvent;
        return ExternalDBApplicationResult.Succeeded;
    }

    public void HandleDesignAutomationReadyEvent( object sender, DesignAutomationReadyEventArgs e)
    {
        e.Succeeded = true;
        e.Succeeded = CreateWindowFamily(e.DesignAutomationData);
    }

    protected bool CreateWindowFamily( DesignAutomationData data )
    {
        if (data == null)
            return false;

        Application app = data.RevitApp;
        if (app == null)
            return false;

        string modelPath = data.FilePath;
        if (String.IsNullOrWhiteSpace(modelPath))
            return false;
```

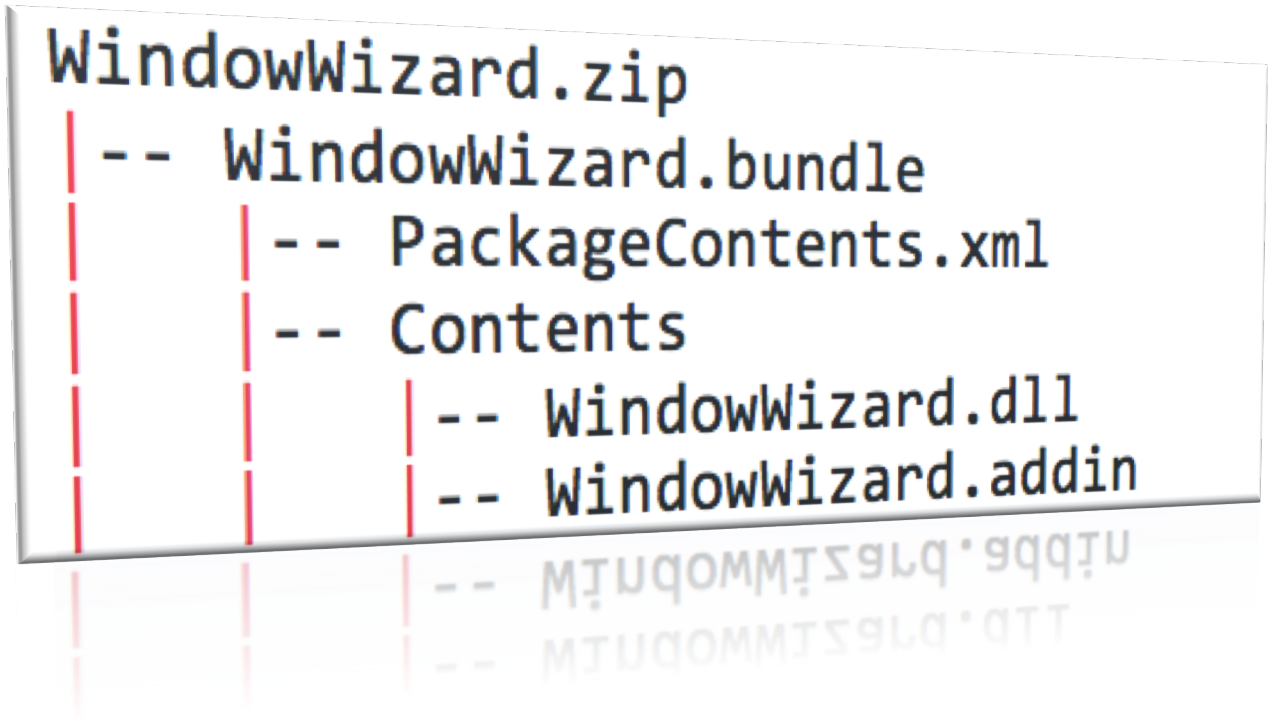
Step 4
Local debug



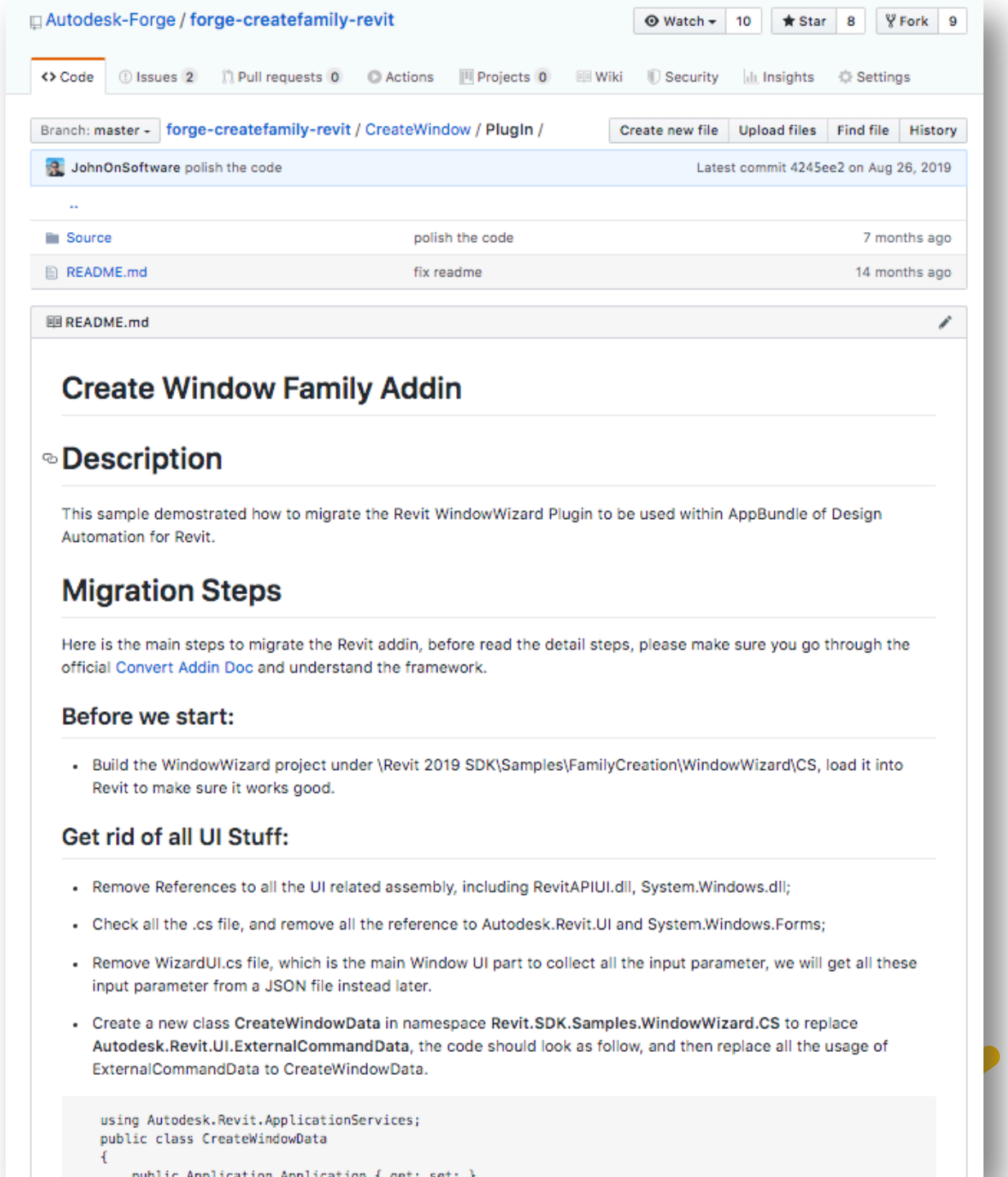
DesignAutomationHandler

https://forge.autodesk.com/blog/design-automation-debug-revit-plugin-locally
https://github.com/Autodesk-Forge/design.automation-csharp-revit.local.debug.tool

Step 5
AppBundle



More details



Autodesk-Forge / forge-createfamily-revit

Watch 10 | Star 8 | Fork 9

Code | Issues 2 | Pull requests 0 | Actions | Projects 0 | Wiki | Security | Insights | Settings

Branch: master ▾ forge-createfamily-revit / CreateWindow / PlugIn / | Create new file | Upload files | Find file | History

JohnOnSoftware polish the code — Latest commit 4245ee2 on Aug 26, 2019

..

Source	polish the code	7 months ago
README.md	fix readme	14 months ago

README.md

Create Window Family Addin

Description

This sample demostrated how to migrate the Revit WindowWizard Plugin to be used within AppBundle of Design Automation for Revit.

Migration Steps

Here is the main steps to migrate the Revit addin, before read the detail steps, please make sure you go through the official Convert Addin Doc and understand the framework.

Before we start:

- Build the WindowWizard project under \Revit 2019 SDK\Samples\FamilyCreation\WindowWizard\CS, load it into Revit to make sure it works good.

Get rid of all UI Stuff:

- Remove References to all the UI related assembly, including RevitAPIUI.dll, System.Windows.dll;
- Check all the .cs file, and remove all the reference to Autodesk.Revit.UI and System.Windows.Forms;
- Remove WizardUI.cs file, which is the main Window UI part to collect all the input parameter, we will get all these input parameter from a JSON file instead later.
- Create a new class **CreateWindowData** in namespace **Revit.SDK.Samples.WindowWizard.CS** to replace **Autodesk.Revit.UI.ExternalCommandData**, the code should look as follow, and then replace all the usage of ExternalCommandData to CreateWindowData.

```
using Autodesk.Revit.ApplicationServices;
public class CreateWindowData
{
    public Application Application { get; set; }
```

AUTODESK®

Make anything™

AUTODESK® FORGE