

# Do You Want to Build an Add-In

John Pierson

Design Technology Specialist  
Parallax Team



# do you want to build an add-in?

**“REVIT ADD-INS FOR BEGINNERS”**

**WWW.PARALLAXTEAM.COM**

# Summary

This hands-on lab aims to provide an introduction to creating your own Revit add-ins using C#. This lab will not be an overly complex look at add-in creation from a computer programmer's point of view.

Instead, it will introduce Visual Studio and C# from a Revit user's perspective. We'll learn how to set up Visual Studio templates to get started quickly, how to observe the Revit database to find out what we can add, and how to build simple functionality to make Revit do what we want. In addition to these, you'll gain insight into a ton of valuable resources to use as you continue your journey into the Revit API.

# Summary

This hands-on lab aims to provide an introduction to creating your own Revit add-ins using C#. This lab will not be an overly complex look at add-in creation from a computer programmer's point of view. Instead, it will introduce Visual Studio and C# from a Revit user's perspective. We'll learn how to set up Visual Studio templates to get started quickly, how to observe the Revit database to find out what we can add, and how to build simple functionality to make Revit do what we want. In addition to these, you'll gain insight into a ton of valuable resources to use as you continue your journey into the Revit API.

# Learning Objectives

- **Learn** about the Revit API.
- **Learn** how to set up Visual Studio for Revit add-in development.
- **Gain** access to valuable resources to build upon.
- **Learn** how to add your own functionality to Revit.

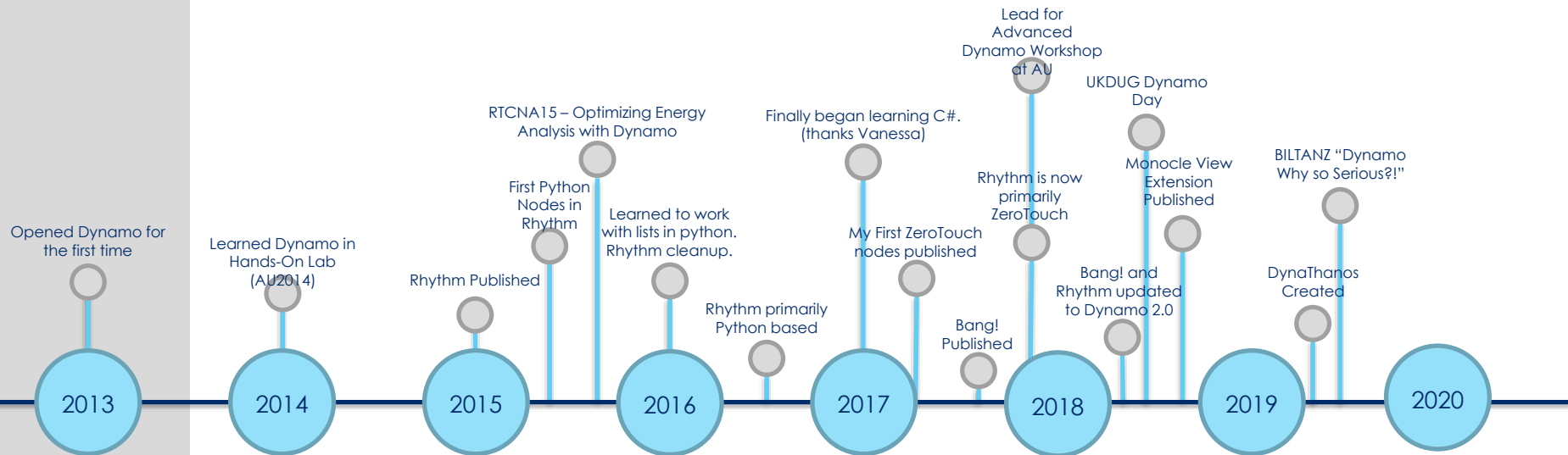
# About the Speaker



## John Pierson

- **Design Technology Specialist** at Parallax Team
- **Reviteer** since 2012
- **Dynamo User** since 2014
- **Dynamo Package Developer** since 2015, (Rhythm, Bang!, DuctTape, Monocle, DynaThanos).
- **Autodesk University and BILTNA** Top Rated Speaker
- **Revit Certified Professional** for Architectural, Structural, MEP

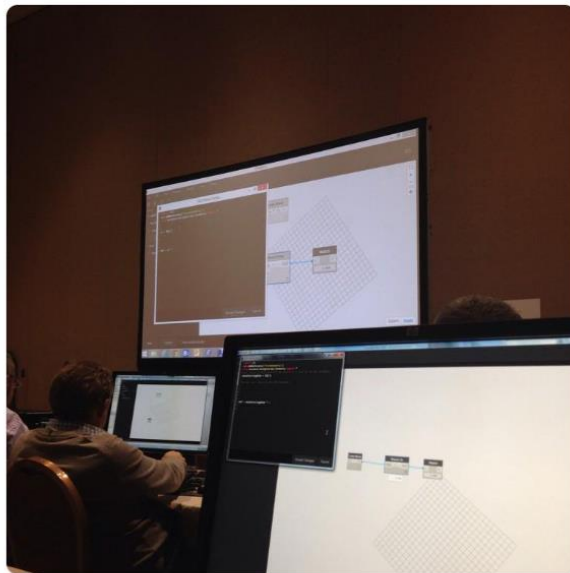
# My Dynamo + Coding Journey



# My Python Journey



Iron python attacked like a cobra! #AU2014

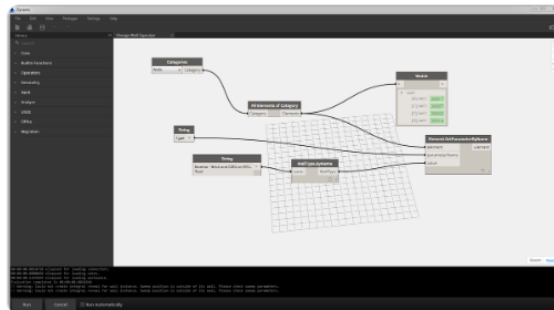


6:13 PM - 1 Dec 2014



Paul F. Aubin @paulfaubin · 7 Dec 2014

Took a bunch of #dynamo courses at #au2014 and actually built my successful routine! (Very silly, but it works!)



john pierson 🤖 🎈  
@60secondrevit

Replying to @paulfaubin

@paulfaubin cool thing is; thanks to those same classes I actually know what that means! A week ago I would have had no idea

11:20 PM - 7 Dec 2014



# Disclaimer

This class aims to demonstrate how I (as a non-formally trained programmer) learned how to utilize create Revit add-ins in a decently efficient manner.

There is a **slight** chance that I:

- Will not get terms perfect
- Will need to reference other Revit API resources for answers
- Will demonstrate workflows that “*real*” programmers would consider “*questionable*”

There is a **100%** chance that we:

- Will have fun
- Will make some awesome stuff
- Leave this class with at least 1 piece of knowledge that we did not have walking in
- Leave this class with resources to be successful in our “Revit API” journeys

# Words of Wisdom

(from some very smart dudes)



**Anthony Hauck**

@anthonyhauck

Following



Replying to @didonenov @60secondrevit @arch\_laboratory

We should be open about these experiences.

@ikeough helped when he said to me,

"Almost every bug fix ends with the coder thinking he or she is an idiot for missing the obvious." It's discouraging for people to think everyone else is a genius creating prize-worthy code.

8:35 AM - 4 Aug 2018

<https://twitter.com/anthonyhauck/status/1025736916791189504>

HYPAR

# Who are you?



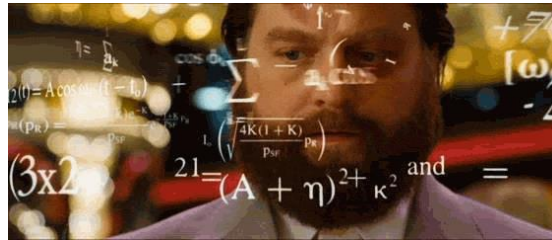
*Image of an actual revit user trying to figure out if they are beginner, intermediate, or advanced.*

# What is an API?

- Technically, API stands for Application Programming Interface (API).
- It's a protocol to communicate with something, in our case the Revit data.
- A set of functions that allow your application to access the features and data of the application.

# What is an API?

- Technically, API stands for Application Programming Interface (API).
- It's a protocol to communicate with something, in our case the Revit data.
- A set of functions that allow your application to access the features and data of the application.



# What is an API?

A common language for us to communicate with the receiving party.



# What is an API?

A common language for us to communicate with the receiving party.



Animals

Zero the Dog

Fur Color : Black

Breed : Labrador

Command: Sit

Command: Stay

Command: Rollover

# What is an API?

A common language for us to communicate with the receiving party.



**Namespace:** Animals is a category with classes.

**Zero :** A class in the animals namespace with it's own methods and properties.

**Properties :** Queries on the object

**Method :** An action on the object.

**Invalid Method:** An action on the object.

*In this case, this API does not support that method, so it returns null.*

Animals

Zero the Dog

Fur Color : Black

Breed : Labrador

Command: Sit

Command: Stay

Command: Rollover



# How to access the Revit API?

- Since Revit has an API, the developers give us a whole set of tools to do cool things.
- We are going to view this through <https://apidocs.co/#>



# What is Visual Studio?

- An Integrated Development Environment (IDE).
- Basically a really cool text editor.
- That allows us to compile code for various applications including Revit!
- Visual Studio is to Code as Revit is to .rvt files.
- For this class we are using Visual Studio Community which is free!

# Anatomy of a Revit add-in

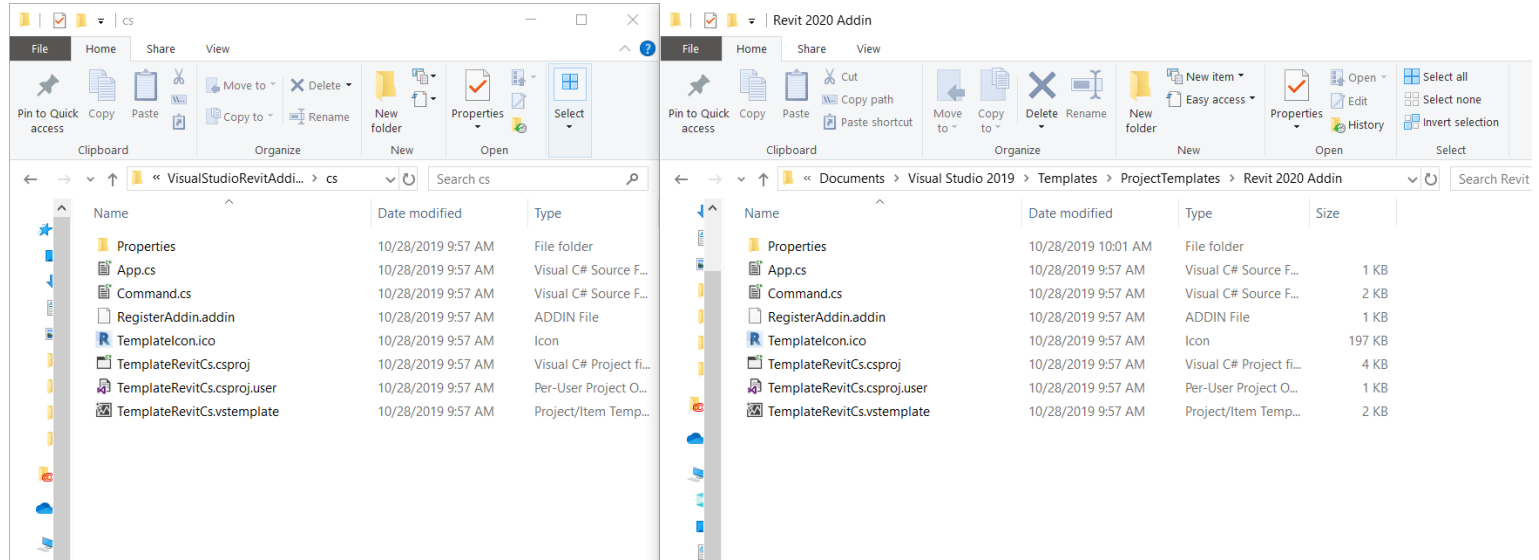
- .dll
- .addin
- Application-level
- Command-level

**Q: How do you start a  
Revit project?**

**Q: How do you start a  
Revit project?**

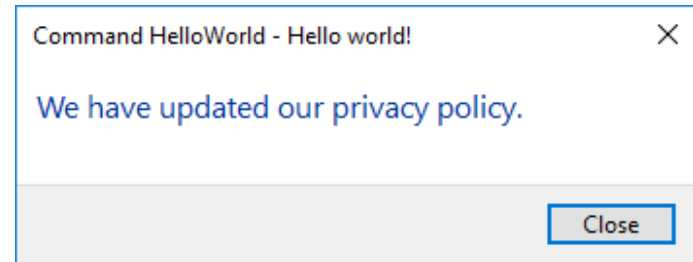
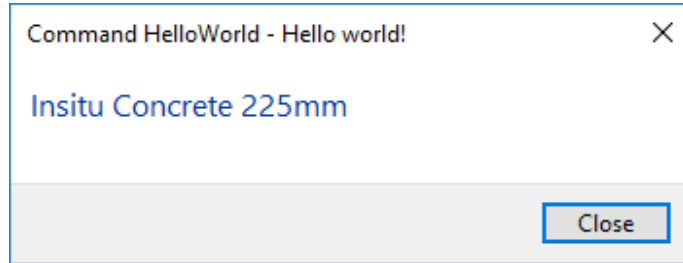
**A: With a template!**

# Running with Revit addin templates



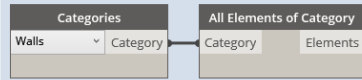
*C:\Users\USERNAME\Documents\Visual Studio  
2019\Templates\ProjectTemplates\Revit 2020 Addin*

# Example 1 | Hello World

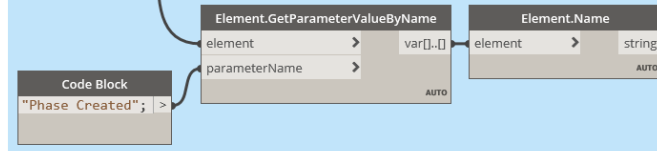


# Example 2 | Phase To Comments

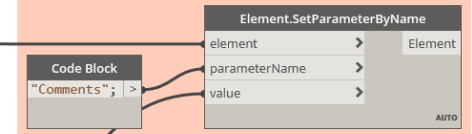
Equivalent of:  
FilteredElementCollector



Equivalent of:  
LookupParameter > As Value String

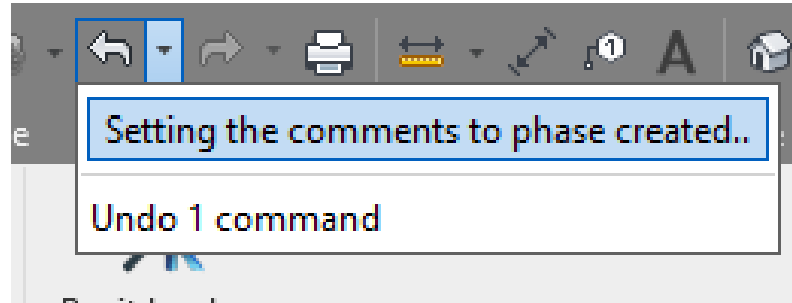


Equivalent of:  
Lookup Parameter > Set Value String

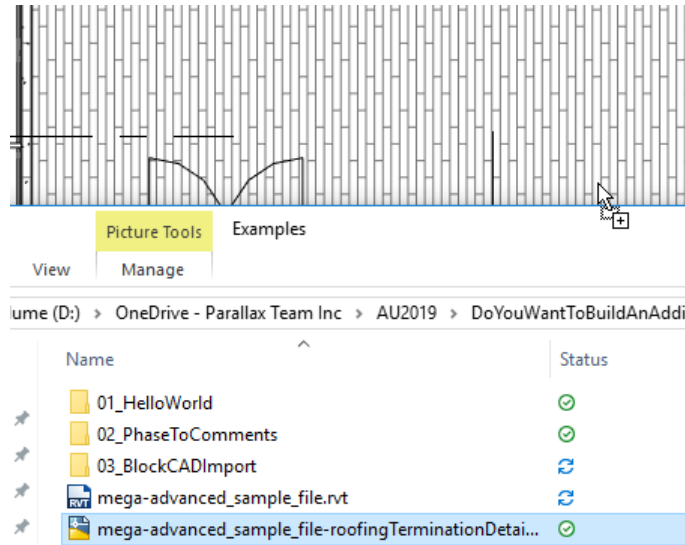




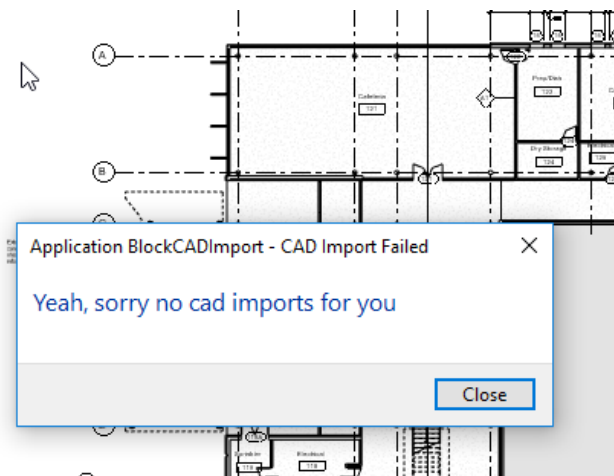
## Example 2 | Phase To Comments



# Example 3 | Block CAD Import Operation

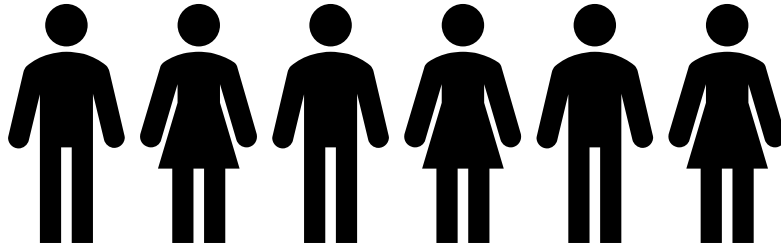


## Example 3 | Block CAD Import Operation



# What is a for each loop?

- Ex: What is your name?



# Transactions?



Charlie knows about transactions!

# Be heard! Provide AU Session Feedback

- Chance to win a pass for AU2020!
- Available on the AU app.
- Give feedback for each session.
- We can see the feedback in real time.





# AUTODESK<sup>®</sup>

## Make anything<sup>™</sup>

Autodesk and the Autodesk logo are registered trademarks or trademarks of Autodesk, Inc., and/or its subsidiaries and/or affiliates in the USA and/or other countries. All other brand names, product names, or trademarks belong to their respective holders. Autodesk reserves the right to alter product and services offerings, and specifications and pricing at any time without notice, and is not responsible for typographical or graphical errors that may appear in this document.

© 2019 Autodesk. All rights reserved.



# Supplemental Slides

Just in case

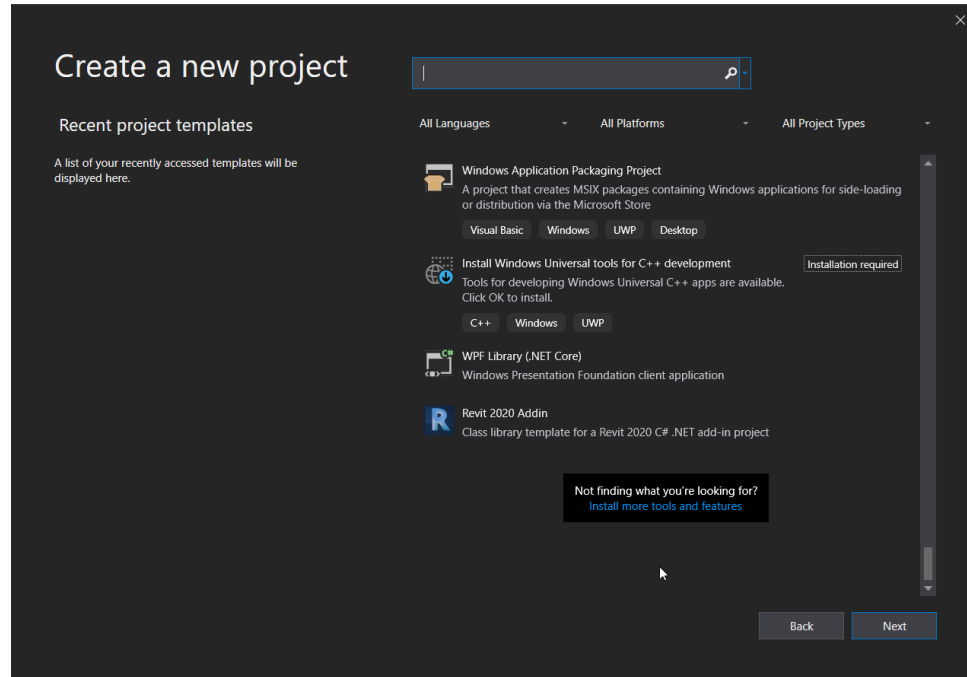


# Running with Revit addin templates

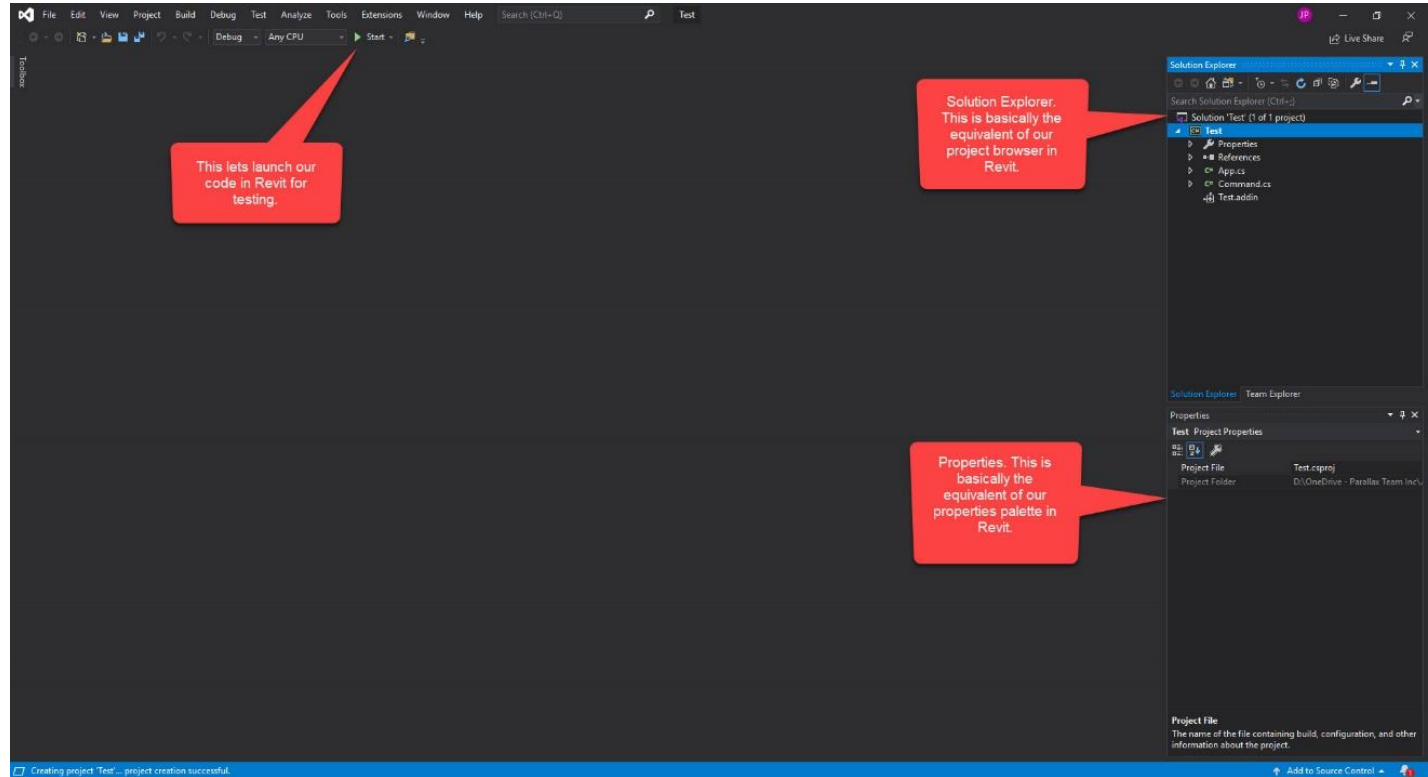
- Download the Revit 2020 Add-In Wizard from [TheBuildingCoder](#).
- Unzip the contents of CS into the visual studio addin directory. (C:\Users\USERNAME\Documents\Visual Studio 2019\Templates\ProjectTemplates\Revit 2020 Addin)
- Restart visual studio.
- Success!

# Using the Template

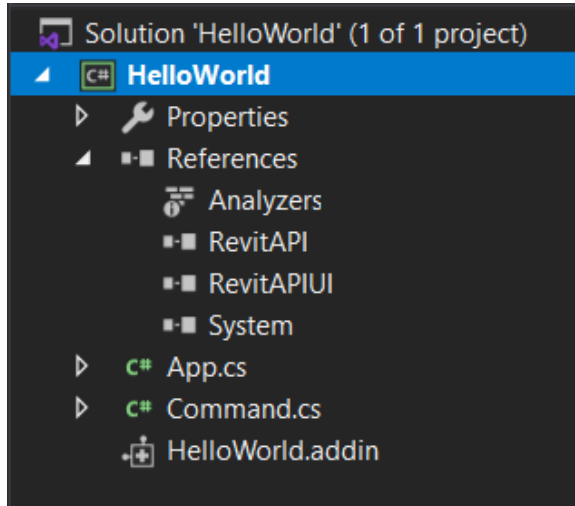
- File > New Project > Revit 2020 Addin > HelloWorld



# Inside Visual Studio



# Inside Visual Studio



**Solution:** Our visual studio solution. This is like the directory that an RVT resides in.

**Project:** This is like our current Revit model when working in Revit

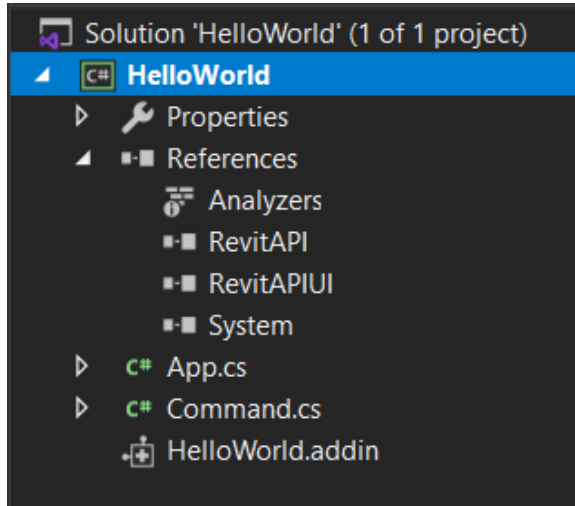
**API References:** Libraries that we can work with. This can be looked at like loading families in a Revit file.

**Application-Level Actions:** Triggers on Revit loading

**Commands:** Triggers on button push

**Addin Manifest:** This tells Revit how to load our addin.

# Example 01| Hello World



**Solution:** Our visual studio solution. This is like the directory that an RVT resides in.

**Project:** This is like our current Revit model when working in Revit

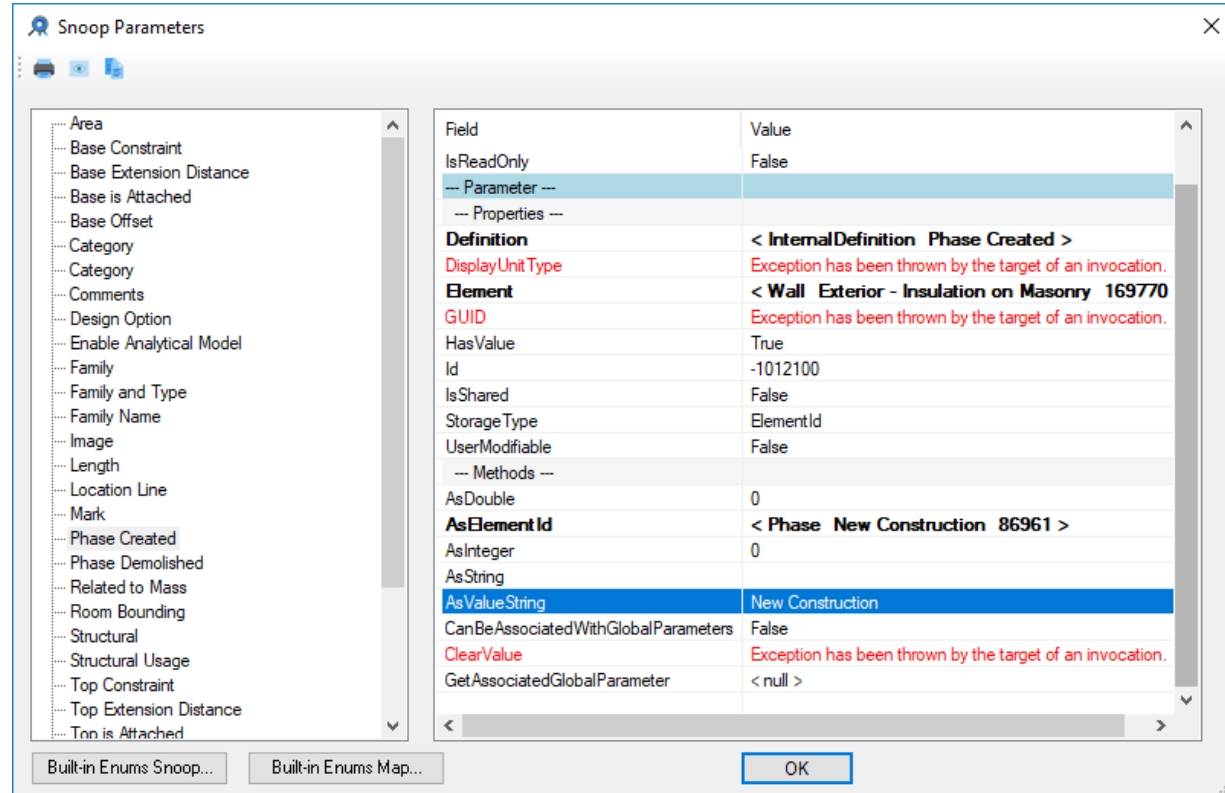
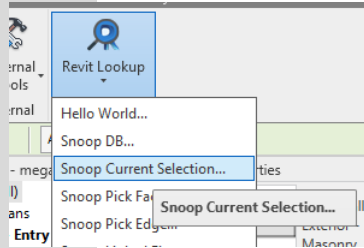
**API References:** Libraries that we can work with. This can be looked at like loading families in a Revit file.

**Application-Level Actions:** Triggers on Revit loading

**Commands:** Triggers on button push

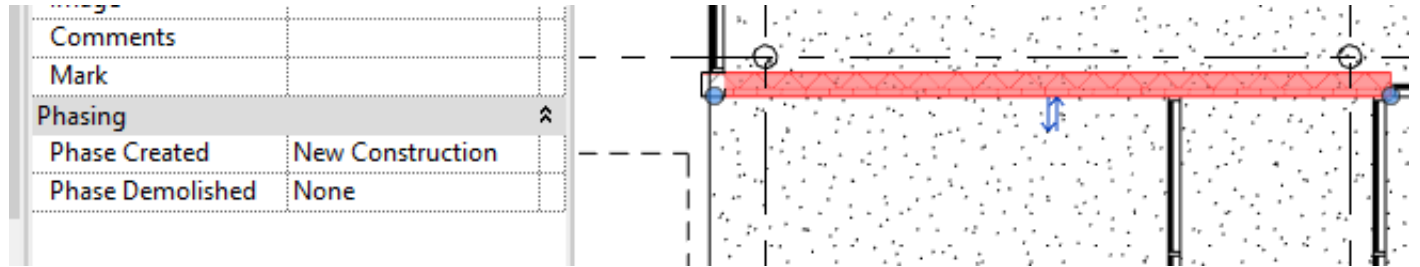
**Addin Manifest:** This tells Revit how to load our addin.

# Example 01| Using Revit Lookup



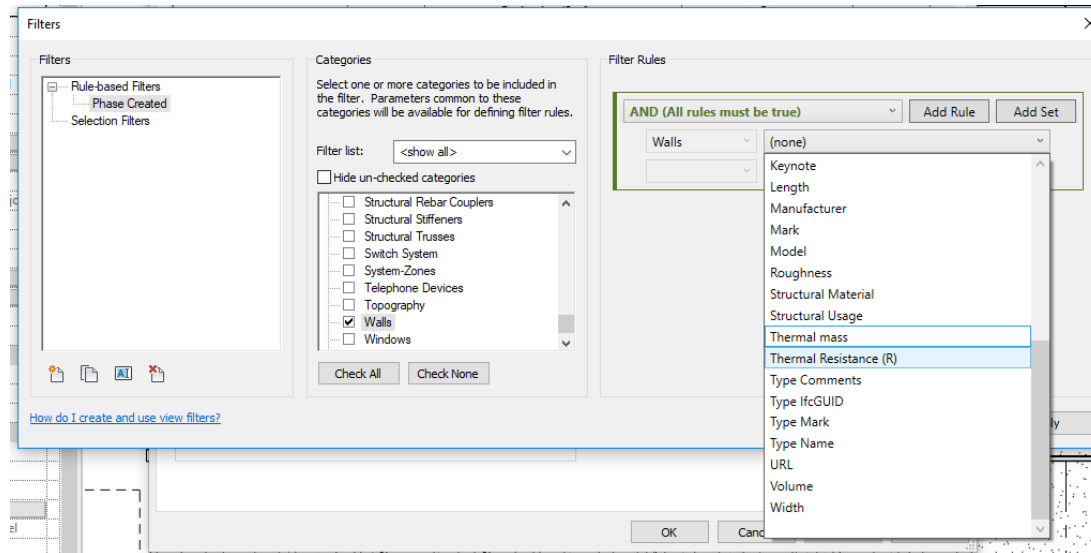
## Example 02| Phase to Comments

- Phase created parameter is pretty darn useful.



# Example 02 | Phase to Comments

- Until..

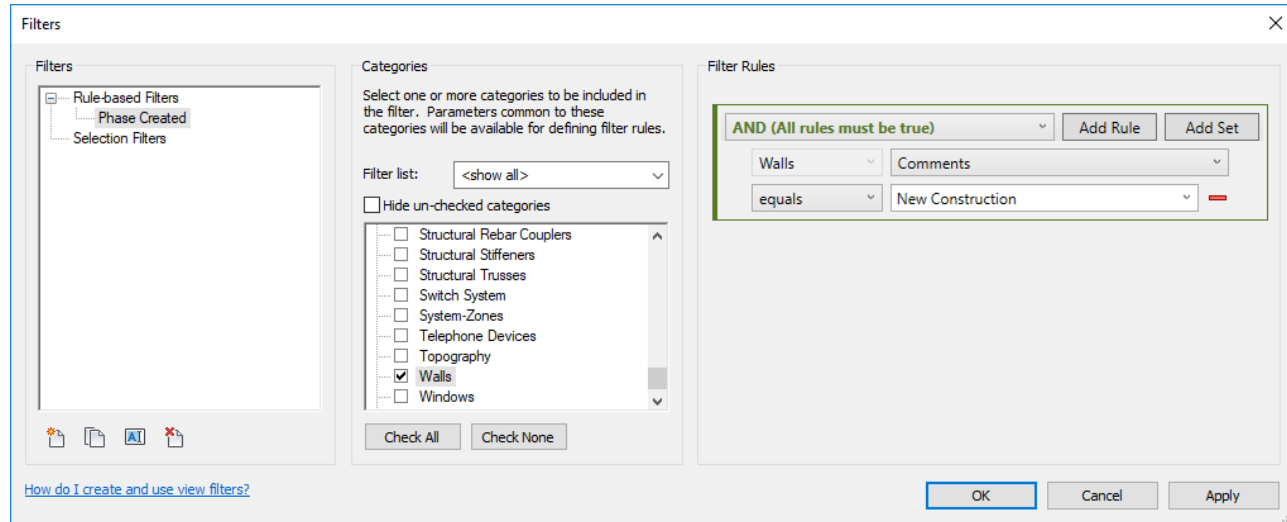


You try to filter against it!



## Example 02 | Phase to Comments

- So, we can just do what we want with another parameter.



## Example 03| Save Revit Users from Themselves

- Please keep CAD out of the Revit file...