



SD21181-L

Coding New Nodes with Dynamo

Adam Sheather
AECOM

Learning Objectives

- Create Basic Zero Touch Node
- Creating Geometry
- Integration with Revit
- Wrapper Setup for 3rd Party Library
- Basic UI development

Description

Learn how to create Dynamo nodes with C#. This lab is designed to show experienced users of software programming the API functions and practical examples of connecting new products, enhancing existing ones (Revit software), and understanding the geometry engine. We'll also take a look at how to create custom user interface (UI) nodes. Users will develop an understanding of where the Dynamo extension stores various aspects of the underlying systems, and what templates you can set up to speed up your software development workflows. Whether you're looking to build simple Zero Touch libraries, complex wrappers to third-party applications, or totally new UI functionality, this class will provide the foundations to help you take significant steps forward. This session features Dynamo Studio and Revit.

Your AU Expert(s)

15+ years in Buildings, Transport & Infrastructure CAD/BIM software user/trainer, developer and project management & delivery. Worked and lived in Australia, Malaysia & Philippines supporting projects located on 6 of 7 Continents for Architects, Engineering Consultants, Main Contractor and a Sub-Contractor. Projects from Residential homes, Luxury Hotels, Retail, Health & Ed, Water, Power, Highways, to 10billion (USD) Rail Developments.

Have developed UI++, DynaWorks & Gorilla Packages for Dynamo.

Current Roles

AECOM – Associate Director BIM Lead Queensland, Australia

BiLT Australia (previously RTC) – Program Manager ANZ

Bad Monkeys – Dynamo Coding Lab Rat

In The Works – El-Capitano Software Product Developer

BrisBIM – Committee Member

Autodesk Expert Elite



Cheat Sheet Dynamo

Please refer to the datasets for all the code and examples/comments in detail!

This section is to list where all the dll's and the relevant commands are to access important parts of Dynamo. This is based on version 1.2.0 and as significant changes have happened in the past be aware your code may need to be updated in the future, however major changes have slowed down that require large rewrites.

Depending on your installed version of Dynamo or Dynamo Studio there are 2 versions. The Dynamo dll's we need to interface with are located here;
C:\Program Files\Dynamo

If you Dynamo Studio only then the DynamoCore folder will contain all your required components, however if you also use the Revit integration additional components required to integrate with Revit with Dynamo will be found DynamoRevit.

Sandbox Mode for Dynamo Revit Users

If you are creating Dynamo specific libraries and don't need the interface with Revit you can use Dynamo for debugging without the overhead of firing up Revit every time.

The Sandbox version is located here

C:\Program Files\Dynamo\Dynamo Revit\1.2\DynamoSandbox.exe

Basic Visual Studio Setup Notes

ZeroTouch and other nodes are simply class libraries there is no need to create other types (ie exe's) to integration with Dynamo.

You can choose to either debug Dynamo Sandbox or Revit then activate Dynamo and the debugger will connect.

You can also setup packages and do simply post event copy/pastes to the DynamoPackage location to test more complex developments like UI.

Examples

Please note as these are examples there are a lot of bad writing programming practices in these files as the idea is to show how the code works not provide production code. Therefore use these to learn the concepts but please with your own project follow proper programming and file/code standards and management.

Base Example

No libraries are required for this example.

This is a very simple example that shows to create very simple zero touch nodes.

This Zero Touch class contains a class with some very basic properties and methods as well as covers some basics of why internal constructors are useful when developing dynamo nodes for UI node management purposes.



DesignScript

Namespace: Autodesk.Designscript.Geometry

Location: C:\Program Files\Dynamo\Dynamo Core\1.2\ProtoGeometry.dll

Purpose

ProtoGeometry is a wrapper for the designscript control of the ASM geometry kernel Autodesk provides for Dynamo Geometry creation.

There are 2 classes that show how to create a cube and a cylinder dynamo geometry objects. The user can load these dll's into Dynamo and use the new node enhancements.

Dynamo Revit Sheet

Namespace: using RevitServices.Persistence;

Location: C:\Program Files\Dynamo\Dynamo Revit\1.2\Revit_ **VERSION** \RevitServices.dll

Purpose

Access to the Document Manager which contains the Revit document instance.

Alias Namespaces:

`using dSheet = Revit.Elements.Views.Sheet;`

`using dView = Revit.Elements.Views.View;`

Location: C:\Program Files\Dynamo\Dynamo Revit\1.2\Revit_ **VERSION** \RevitServices.dll

Purpose

Access to the Dynamo wrapped Revit API objects.

This is a simple exercise that shows some key parts of the DynamoRevit API. How to get the current Revit document, accessing Revit items through Dynamo wrapper and exposing internal data, and finally performing a simple transaction.

BCFier

Namespace: using Autodesk.DesignScript.Runtime;

Location: C:\Program Files\Dynamo\Dynamo Core\1.2\DynamoServices.dll

Purpose

Allows users to hide classes, properties and methods that are not meant to be exposed in the UI when loading in zero touch nodes.

This is a simple demonstration show a basic approach to starting to wrap other libraries into usable Dynamo libraries that Dynamo users can quickly understand and access.