

# ***Readme guide for DynaStandard Extension for use in Dynamo.***

Dynamo Content Library

Selected Graph Information

Extracted data from graph

File Name:

Created By:

Dynamo Version:

Revit Version:

Description:

Required Packages:

Actual Saved Version:

Open in New Graph: ☐ Yes/No

UK DYNAMO  
USER GROUP

Created By: Brendan Cassidy

Rev: P08

## Table of Contents

1.	Introduction.....	1
2.	Reporting issues or providing feedback .....	1
3.	Setup.....	1
3.1.	No DynaStandard menu(FileLoadException).....	2
3.2.	Config Setup .....	3
4.	Running the Extension.....	5
4.1.	Dynamo Standard .....	6
4.2.	Dynamo Library .....	6
4.3.	Dynamo Template .....	6
4.4.	Python Template .....	7
4.4.1.	Python Template – Option 1 .....	7
4.4.2.	Python Template – Option 2 .....	7
4.4.3.	Python Template – Option 3(Default) .....	8
4.5.	Node Re-runner .....	8
4.5.1.	Re-runner by python .....	9
4.6.	Bonus Feature – Add Custom Package Location .....	10
4.7.	Bonus Feature – Force Manual Run .....	10
5.	About Window/Version Info .....	11

Rev	Amendment	Date
P01	First Release	19/02/19
P02	General update. Added temporary Python Template options information to document.	03/05/19
P03	Added Python Template implementation selector.	06/05/19
P04	Added a way to automatically set graphs to be manual for run type. Fixed a issue with finding settings file.	05/08/19
P05	Added explanation and fix for DynaStandard Menu not showing	06/08/19
P06	Added a required packages section to Dynamo Content Library window(Also works with dynamo 2.3 and above package serialisation) Updated About window to include version info.	09/10/19
P07	Updated Readme 3.2 to include important note about & special character	25/01/2020
P08	Updated Readme to match new features added to DynaStandard V1.1.0 Updated images for Python Template images as now a WPF window instead of a form. Fixed a bug where windows usernames with a dot would break paths. Fixed a bug where if you start a custom node and was using manual run setting changer it will crash out dynamo.	03/09/2020

## 1. Introduction

This Dynamo extension is my first c# project and first attempt at creating a dynamo extension, therefore please be productive in any feedback or issue reporting.

I would like to start of by giving my thanks to the members of the dynamo forum/bloggers for the wealth of information/knowledge that has been posted around the place that helped me get to grips with dynamo extension development.

The original idea for my Dynamo extension was to develop a Dynamo content library to help distribute content to all, have easy access to graph templates and any standard documentation related to dynamo. The remit has expanded to include a way to automatically set scripts to manual when opened, set custom package location and a way to re-run dynamo scripts without closing and reopening them.

### Important Note:

- The dynamo extension should be tested prior to any deployment.
- Config file will need to be amended to suit prior to any deployment.

## 2. Reporting issues or providing feedback

### Important Note:

- This is for Dynastandard issues only.

Any errors or feedback about the extension should be reported to the GitHub page below by creating an issue report. Though please follow the provided templates and provide plenty of information related to what you did and what happened plus dynamo versions.

GitHub page: [https://github.com/brencass/DynaStandard\\_public](https://github.com/brencass/DynaStandard_public)

## 3. Setup

To setup the Dynamo extension for use will depend on the version of dynamo you are running. You can either utilise the package manager for dynamo 2.0 and above, though if you are using a dynamo version prior to 2.0 then you will have to utilise the "View extensions" folder method.

### View Extensions install:

You will need to extract the "ViewExtensions" version of DynaStandard it into the following folder on the computer "C:\Program Files\Dynamo\Dynamo Core\<Version>\viewExtensions". Then once this is done you just need to follow the config setup then you are finished for your first run.

### Package Manager:

You will need to either install the extension via package manager or by extracting the files from the package folder on the github page into your package folder which is usually “C:\Users\<username>\AppData\Roaming\Dynamo\Dynamo Core\<version>\packages” though do note that if you want to use this add-on in dynamo core and in revit then you will need to follow the same path but choose the “Dynamo Revit” folder instead of “Dynamo Core” folder.

Once you have installed the extension in either of the above it is time to setup the config file that comes with the extension, this will contain all of the relevant file paths for your content that is to be loaded.

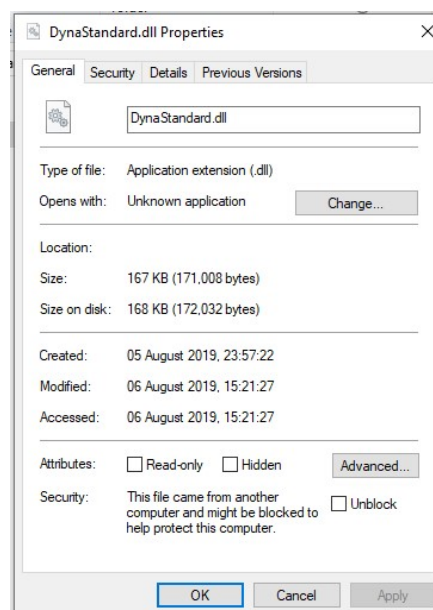
### 3.1.No DynaStandard menu(FileLoadException)

If you notice that you are not getting a new menu called “DynaStandard”, this is probably because the dll is currently blocked and will need to be unblocked. You can also see within the dynamo logs it will show the following:

```
Dynamo will use the package manager server at : https://www.dynamopackages.com
Exception when attempting to load package DynaStandard from C:\Users\ [redacted] \AppData\Roaming\Dynamo\Dynamo Revit\2.0\packages\DynaStand
System.IO.FileLoadException: Could not load file or assembly 'file:///C:\Users\ [redacted] \AppData\Roaming\Dynamo\Dynamo Revit\2.0\packages
DynamoPackageManager (id: FCABC211-056B-4109-AF18-F434DFE48139) extension is added
```

To fix this you need to go to the location the extension is installed to which should be one of the folders from [section 3](#). If you have installed it via the package manager then you will need to be in either “Dynamo Core” or “Dynamo Revit” depending on if you are loading dynamo sandbox or dynamo for revit.

Once you are in the package folder for DynaStandard, go to the bin folder and right click the DynaStandard.dll file and click properties which should bring up a similar window to the below. Then you need to tick the unblock box at the bottom right then click ok. Once this is done you will be ok to relaunch dynamo and it should be working, though if you have launched dynamo via revit you will need to relaunch revit and dynamo for the effect to take place.



## 3.2. Config Setup

To start off with you will need to find the “DynaStandard\_Settings.xml” config file that comes with the extension and open it in notepad or notepad++. This will be in the folder or sub folder for where dynastandard is installed depending on the folder from [section 3](#).

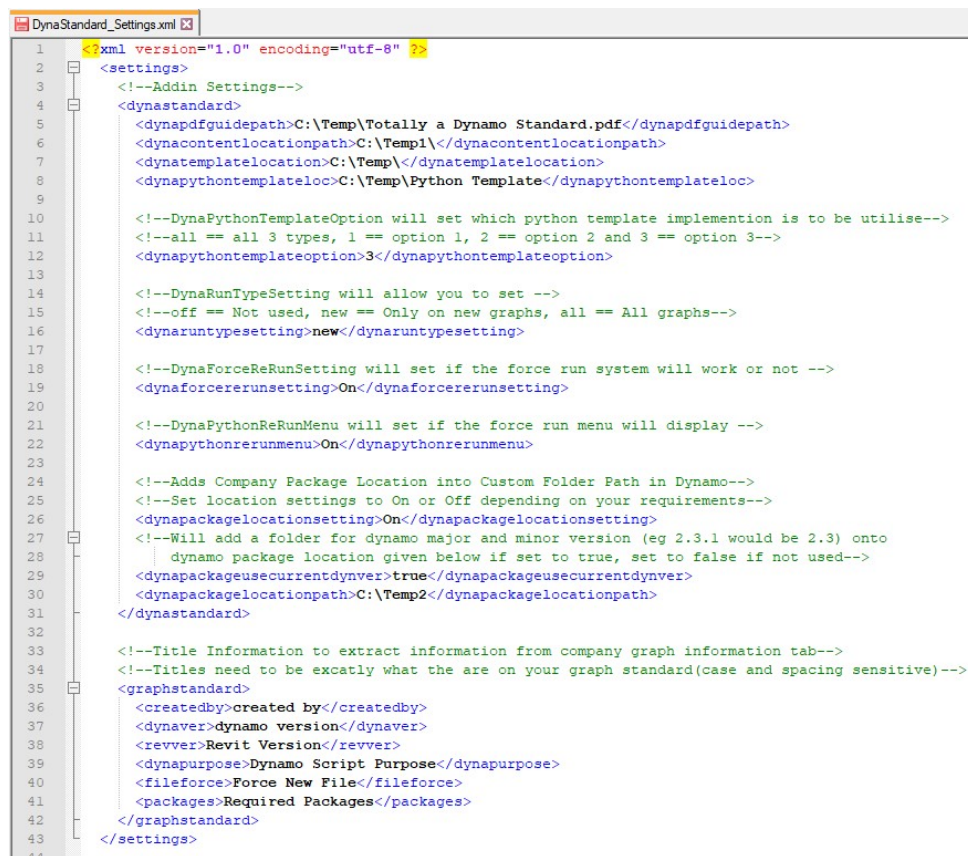
### Important Note:

- If you have a “&” special character within your location paths then you will need to replace this character with “&amp;” within the xml file to eliminate a chance of getting an error on startup.

Example:

```
<DynaTemplateLocation>C:\temp\New &amp; folder</DynaTemplateLocation>
<DynaPythonTemplateLoc>C:\temp\New &amp; folder</DynaPythonTemplateLoc>
```

When you open the file you will see the following(or at least same layout)



```
1  <?xml version="1.0" encoding="utf-8" ?>
2  <settings>
3    <!--Addin Settings-->
4    <dynastandard>
5      <dynapdfguidepath>C:\Temp\Totally a Dynamo Standard.pdf</dynapdfguidepath>
6      <dynacontentlocationpath>C:\Temp1</dynacontentlocationpath>
7      <dynatemplatelocation>C:\Temp</dynatemplatelocation>
8      <dynapythontemplateloc>C:\Temp\Python Template</dynapythontemplateloc>
9
10     <!--DynaPythonTemplateOption will set which python template implementation is to be utilise-->
11     <!--all == all 3 types, 1 == option 1, 2 == option 2 and 3 == option 3-->
12     <dynapythontemplateoption>3</dynapythontemplateoption>
13
14     <!--DynaRunTypeSetting will allow you to set -->
15     <!--off == Not used, new == Only on new graphs, all == All graphs-->
16     <dynaRunTypeSetting>new</dynaRunTypeSetting>
17
18     <!--DynaForceReRunSetting will set if the force run system will work or not -->
19     <dynaforcererunsetting>On</dynaforcererunsetting>
20
21     <!--DynaPythonReRunMenu will set if the force run menu will display -->
22     <dynapythonrerunmenu>On</dynapythonrerunmenu>
23
24     <!--Adds Company Package Location into Custom Folder Path in Dynamo-->
25     <!--Set location settings to On or Off depending on your requirements-->
26     <dynapackageLocationsetting>On</dynapackageLocationsetting>
27     <!--Will add a folder for dynamo major and minor version (eg 2.3.1 would be 2.3) onto
28     | dynamo package location given below if set to true, set to false if not used-->
29     <dynapackageusecurrentdynver>true</dynapackageusecurrentdynver>
30     <dynapackageLocationpath>C:\Temp2</dynapackageLocationpath>
31   </dynastandard>
32
33   <!--Title Information to extract information from company graph information tab-->
34   <!--Titles need to be exactly what they are on your graph standard(case and spacing sensitive)-->
35   <graphstandard>
36     <createdby>created by</createdby>
37     <dynaVer>dynamo version</dynaVer>
38     <revVer>Revit Version</revVer>
39     <dynaPurpose>Dynamo Script Purpose</dynaPurpose>
40     <fileforce>Force New File</fileforce>
41     <packages>Required Packages</packages>
42   </graphstandard>
43 </settings>
44
```

There are two main sections within this text which are “dynastandard” and “graphstandard”. The dynastandard section caters for where the extension should look for content, templates and standard document. The graphstandard section is for what word(s) the extension will look for within the text notes that are within your graph then extract information into relevant sections on the content library window.

Dyna standard section there is 11 sub sections which are as follows:

- DynaPDFGuidePath – Path to your pdf file that has your graph standards.
- DynaContentLocationPath – Directory path to the central location of graphs are kept.
- DynaTemplateLocations – Directory path to the central location of graph template files.
- DynaPythonTemplateLoc – Directory path to central location of Python template files( in python format[.py]).
- DynaPythonTemplateOption – This will set up which python template implementation is to be used by Dynastandard and can be indicated as either 1, 2 or 3 for the relevant python template implementation you want to use.
- DynaRunTypeSetting – This will set up how and if you want graphs to be forcibly set to “Manual” on the run type. If you set the setting to “off” then the extension will not apply a run type of manual. If you want it on then you need to set it to “new” to only apply to new graphs or “all” to apply it to all graphs that get opened.
- Dynaforcererunsetting – This will allow you to utilise the DynaStandard inbuilt force re running of nodes without the python node. If you set it to “off” then the extension will not allow the extension to do the force rerun. If you set it to “on” then the extension will be allowed to do the force rerun.
- DynaPythonReRunMenu – This will create an additional menu item called “Re-Runner in Python” which when clicked on will place a python node with some custom code into your current dynamo graph. This code will handle all the re-running requirements for the script.
- DynaPackageLocationSetting – This will set if you want the custom package location to be set with the location indicated by DynaPackageLocationPath. This will be done by setting it to “On” if you do and “Off” if you do not.
- DynaPackageUserCurrentDynVer – Will add a folder for dynamo major and minor version (eg 2.3.1 would be 2.3) onto the dynamo package location given in DynaPackageLocationPath setting if set to true, set to false if not used(eg you specify specific path without dynamo versioning controls).
- DynaPackageLocationPath – Directory Path to where custom package files are kept.

Important Note:

- For the rerunning to work the node that is required to be rerun has to start with “\*ForceUpdate” otherwise the nodes will not re-run.

Graph Standard sections has 6 sections where you need to fill in the relevant titles from the “Dynamo Script Information” section of your graph/Graph templates with a completed example shown below and how the script information looks within dynamo.

**Important Note:**

- The required package section will read data directly from a script that has been saved in dynamo 2.3 or above instead of data indicated in script information. This is because 2.3 introduced a way to serialise this information with the dynamo file.

**Settings file Example:**

```
<!--Title Information to extract information from company graph information tab-->
<!--Titles need to be exactly what they are on your graph standard(case and spacing sensitive)-->
<graphstandard>
  <createdby>created by</createdby>
  <dynamver>dynamo version</dynamver>
  <revver>Revit Version</revver>
  <dynampurpose>Dynamo Script Purpose</dynampurpose>
  <fileforce>Force New File</fileforce>
  <packages>Required Packages</packages>
</graphstandard>
```

**Graph Example:**

Dynamo Script Information

Dynamo Template V0.3

DYNAMO FILE NAME

XYZ\_Dynamo\_Template.dyn

DYNAMO VERSION

2.5

Note: Indicate the dynamo version that the script can be used with, as the script may have limitations with other dynamo versions because of node/API changes.

CREATED BY

Brendan Cassidy

REVIT VERSION

Revit 2020

Note: Indicate the Revit version that the script was testing with, and note that the script may have limitations with other revit versions because of API limitations or deprecation.

DYNAMO SCRIPT PURPOSE

This script is to try and get a standard dynamo standard so we get a common look for each script. This will help to aid checking of scripts and the usage of these scripts.

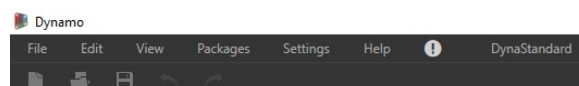
Required Packages

Area to highlight required packages to allow this script to run.

Once all these settings have been changed, just save the file.

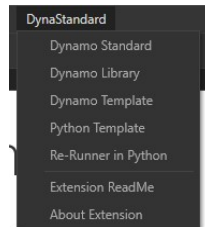
## 4. Running the Extension

As long as you have followed the above, all you will have to do is load dynamo as you would normally for any changes to take effect and for the add-on to load. Once you open dynamo you will be met with the following new menu.



Then when you click on DynaStandard it will have a drop down menu where it contains clickable menu items for you to either see your dynamo standard document, Dynamo content library, python template options or even you're Dynamo Templates.

The last two items being a quick link to this readme and a way to get to the about section for DynaStandard extension.

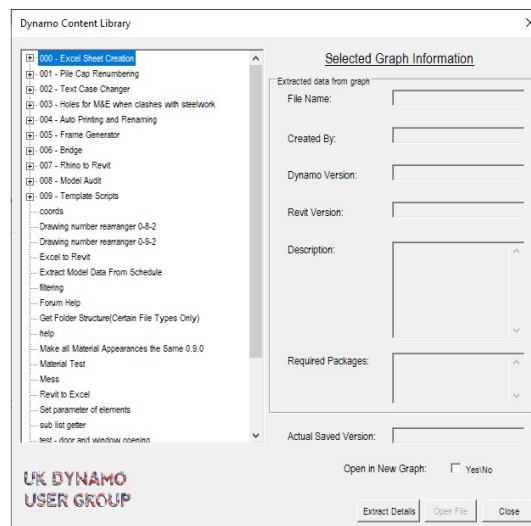


### 4.1.Dynamo Standard

When you click on the Dynamo Standard it will automatically open the Dynamo standard document that you have specified as your standard for dynamo.

### 4.2.Dynamo Library

Dynamo Library will open a new window with your dynamo content loaded within the new window and look similar to the follow(though content will be different).



### 4.3.Dynamo Template

Dynamo Template menu item will have your Dynamo templates files loaded automatically that once clicked will create a brand new dynamo graph with the contents from your template.



## 4.4. Python Template

The 3 No python template menu items will work as the below indicated options

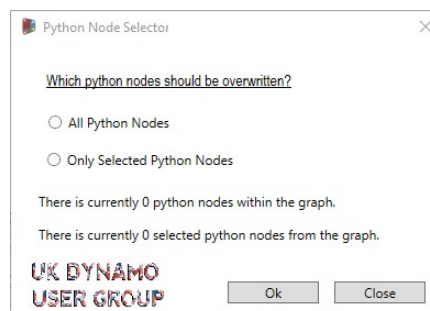
### Important Note:

- The Python Template options are to highlight possible implementations of changing the python nodes script to one of your templates.
- Please put all feedback relating to the Python Template option onto this github post [https://github.com/brencass/DynaStandard\\_public/issues/2](https://github.com/brencass/DynaStandard_public/issues/2)

For the Python Template options to work you will need to have the config file set as needed (default is option 3). Then once you are within a dynamo graph you can select the python node(s) to be change or dynamically select the python node(s) when the window is open. Just make sure you have selected what you need and the correct option of all or selected nodes prior to clicking ok which will make it replace the python code.

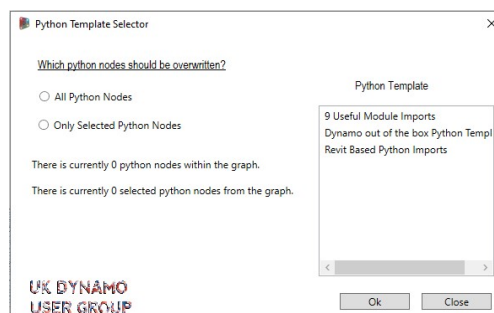
### 4.4.1. Python Template – Option 1

Option 1 works by creating a submenu for each python template file, then it will open the following window. Once it opens you can select if you want all of the python nodes or just the selected python nodes within the currently opened dynamo graph to be changed.



### 4.4.2. Python Template – Option 2

Option 2 works by selecting the python template menu which then opens a window as the below but this is slightly different to the window used in option 1 because it has a selection area for all of the python files it currently found. This then allows you to select the required implementation (all or only selected nodes) with a selection for the required template then once applied it will do the change.



#### 4.4.3. Python Template – Option 3(Default)

This option is a hybrid of option 1 and 2, though it does not include a sub option. This is because when the extension only finds 1 python file within the template directory it will initialise the window that doesn't include a template selection (option 1). If you have a folder that has more than 1 python template then it will switch to the window that has a way to select any of the template files as per option 2.

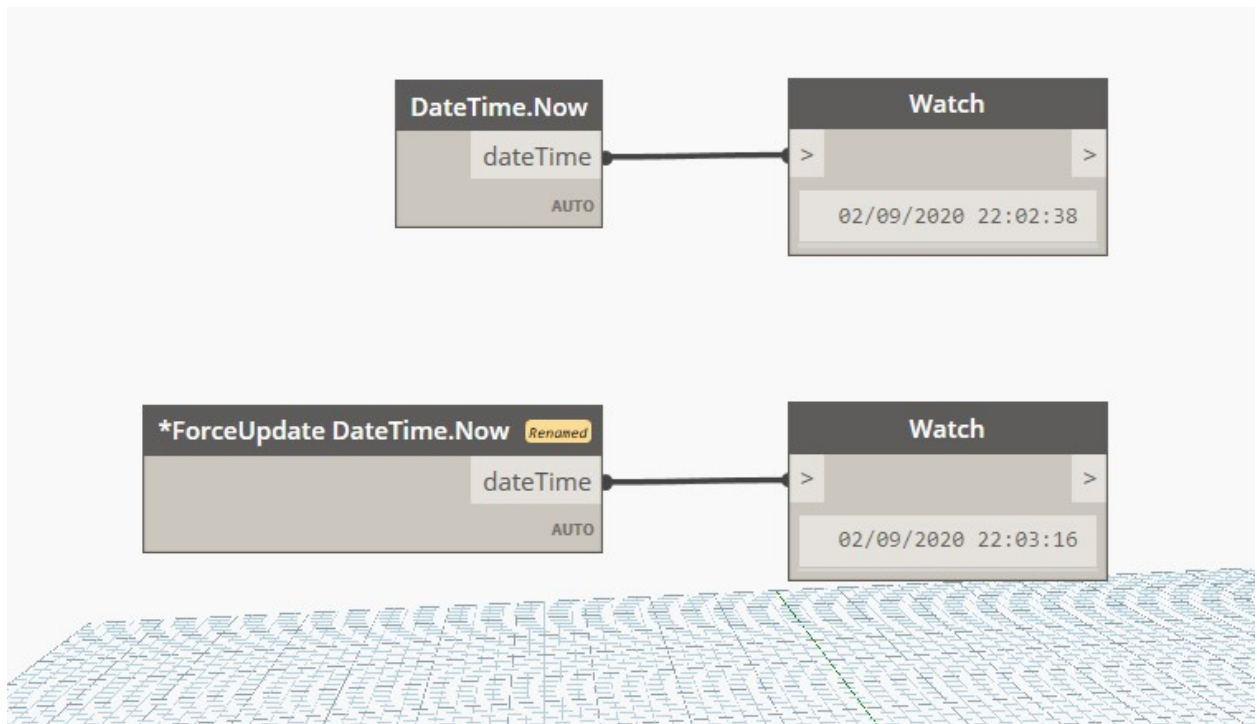
#### 4.5. Node Re-runner

Re-runner within DynaStandard has two parts to it, one is a behind the scenes element where it will run it all for you as long as DynaStandard has the right settings and the nodes have the correct prefix, and the other way is to place a python node from the new menu option onto your graph for it to rerun again after first run though this will require the same prefix and no Dynastandard to be installed.

##### Important Note:

- The original Python code was created by John Pierson in the following post <https://forum.dynamobim.com/t/force-run-python/31709>.

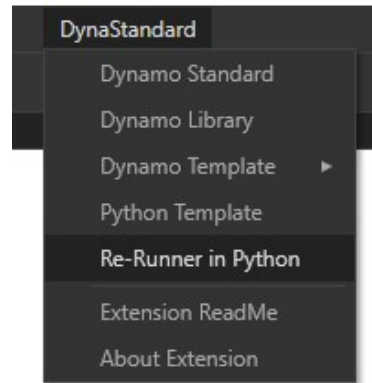
The prefix on nodes is set as “\*ForceUpdate” which you can see from the following image.



Note: Renamed tag is a new addition to dynamo 2.6

#### 4.5.1. Re-runner by python

Within the DynaStandard menu there is a option called “Re-Runner in Python” which is a way to still allow your script to be able to be re-run without the DynaStandard being installed. This is because it will place some existing code onto your graph for you to make everything work when not using DynaStandard.



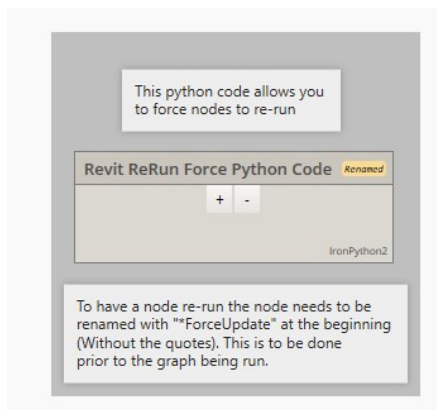
Currently this option will place the required variation of python code depending on which version dynamo is hosted on, this is limited to Civil 3d and Revit.

Dynamo Sandbox plus any other hosted versions of dynamo are not supported by these python rerunner code and will therefore display the following message to stop the python code being placed from the menu.

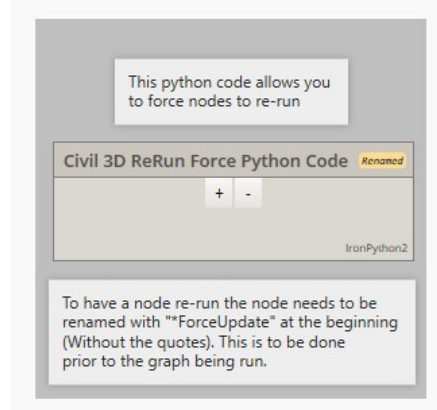
Dynamo SandBox Message:



Dynamo Hosted in Revit:



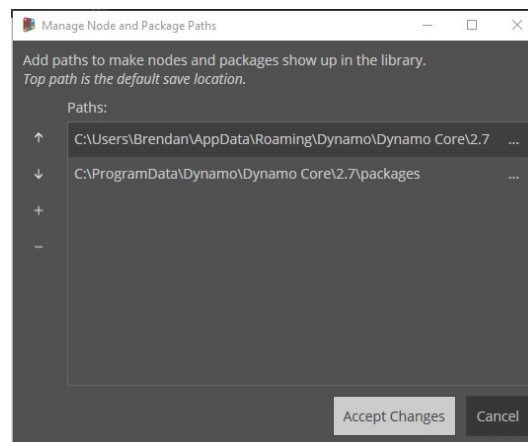
Dynamo Hosted in Civil 3D:



Note: IronPython2 tag is a new addition to dynamo 2.7 because dynamo is moving towards Cpython implementation as a replacement to Ironpython

## 4.6. Bonus Feature – Add Custom Package Location

This extension will automatically add a custom package location that is indicated within the settings file into the manage node and packages path setting of dynamo (if not already added) then tell dynamo to load these packages if not already loaded. This will save the modification of dynamo setting files or even making sure the correct paths are set within the relevant dynamo version setting folder.



### Important Note:

- The original idea for this was posted by Oliver Green for help with on the DynamoBim forum <https://forum.dynamobim.com/t/dynamo-api-set-user-packages-location/52957> though i did submit a solution <https://forum.dynamobim.com/t/dynamo-api-set-user-packages-location/52957/19>

## 4.7. Bonus Feature – Force Manual Run

This extension will cover your back when opening any existing scripts or opening a brand new graph, because it will automatically force dynamo to change to “Manual” run. This should help to stop any infinite loops or opening a script that auto runs just as you remember to click save.



## 5. About Window/Version Info

The about window is a area where you can get the link to the project GitHub page as well as my twitter/GitHub page. The about page includes a way to highlight if you are running a outdated version by checking the latest version on the GitHub project repo website and on the dynamo package website. It will highlight if you are out of date, up to date or even ahead of currently issued version (if you are that lucky).

