AUTODESK UNIVERSITY

IT21116-L

Dynamo for Revit – Exploring the Data Between the Links

Jeff Campbell
Burns & McDonnell

Learning Objectives

- Learn how to identify practical uses for Revit and Dynamo in the manipulation of project data.
- Learn how to diagram a basic workflow for gathering data from linked model files.
- Learn how to develop a small script for copying information between architectural rooms and MEP spaces.
- Learn how to develop a small script for performing information updates on a collection of elements.

Description

We've all heard a lot lately about the Dynamo extension and Revit software. We can't read a blog or go to a conference without someone mentioning this new and exciting tool. We've been told it can help us with geometry. We've been told it can speed up repetitive tasks. Come see some practical uses of the Dynamo extension and how it can expand the data in your model. Take a look at how we can use it to pass information between linked files, talk to Microsoft Excel, and finally automate those day-to-day tasks.

Your AU Expert(s)

Jeff Campbell has been a senior application support specialist for Burns & McDonnell for 11 years. Campbell is responsible for the development and deployment of Autodesk, Inc., applications, and various other engineering applications. Campbell's primary focus is on the AutoCAD Architecture software, AutoCAD MEP software, and Revit software products. Prior to his employment with Burns & McDonnell, Campbell was an assistant professor with the University of Central Missouri, teaching in the College of Applied Sciences and Technology. Campbell lives in Lees Summit, Missouri, with his wife and 2 children.

Introduction

What is Dynamo

Dynamo has been defined as multiple things over the past few years since it appeared. My base definition is that is a graphical programing language that has the ability to create geometry and manipulate data through the use of the Revit API. Dynamo has the unique ability to stand on its own but also to be leveraged as an add-in to Revit for the masses. You no longer have to be a C# .NET programmer to leverage the Revit API to perform task.

Dynamo at its core allows the user to drop graphical "code blocks" and connect them with "wires" in order to provide a logical path of operations. These "code blocks" are Revit API instructions that are easy to understand and provide a quick easy way to access the Revit database.

Dynamo is also free and can be downloaded and learned by Revit users.

What Dynamo is not

Dynamo is not managed code. Users are required to open up the DYN file inside of the editor to run the program. Programmers have no way at this time to securely compile the code or add individual "programs" to the ribbon as icons. Large companies who have workflows in place to create and manage add-ins or scripts that are used in their environment have to use nonstandard conventions in order to control the code.

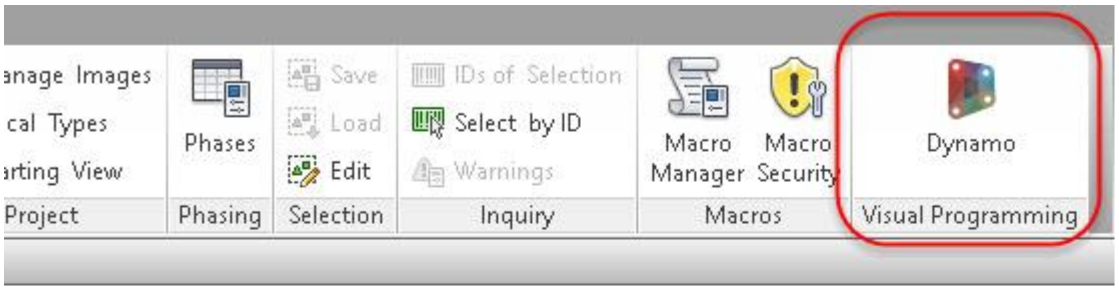
What makes it great

Did I mention Dynamo is FREE? That is only one thing that makes it great. Dynamo is also "open sourced". There is an entire community that has sprung up around Dynamo. They provide the core of the development and features that have been added to the product. They also provide packages of code nodes that be downloaded and utilized. The Dynamo forum is open to everyone for the sharing of ideas and information.

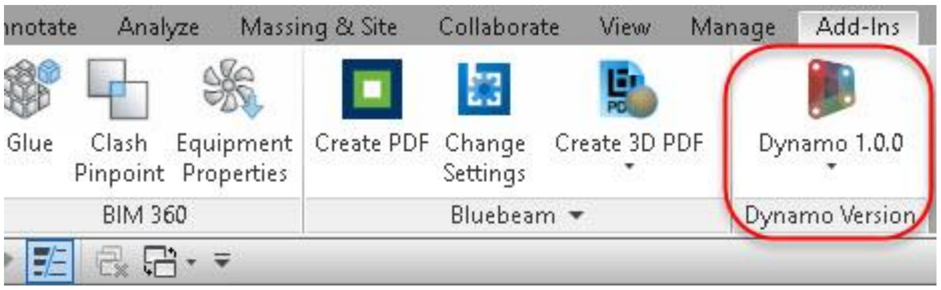
Dynamo Basics

How to get Dynamo

The Dynamo interface for Revit 2017 is no longer added separately by the user. Autodesk has built in the Dynamo add-in and it can be access from the Manage ribbon just like the Macro editor environment.



For those users wanting to gain access to Dynamo on previous version the add-in can be downloaded from http://dynamobim.org/download/. Once downloaded and installed Dynamo can be access from the Add-in ribbon in Revit 2015 and Revit 2016.

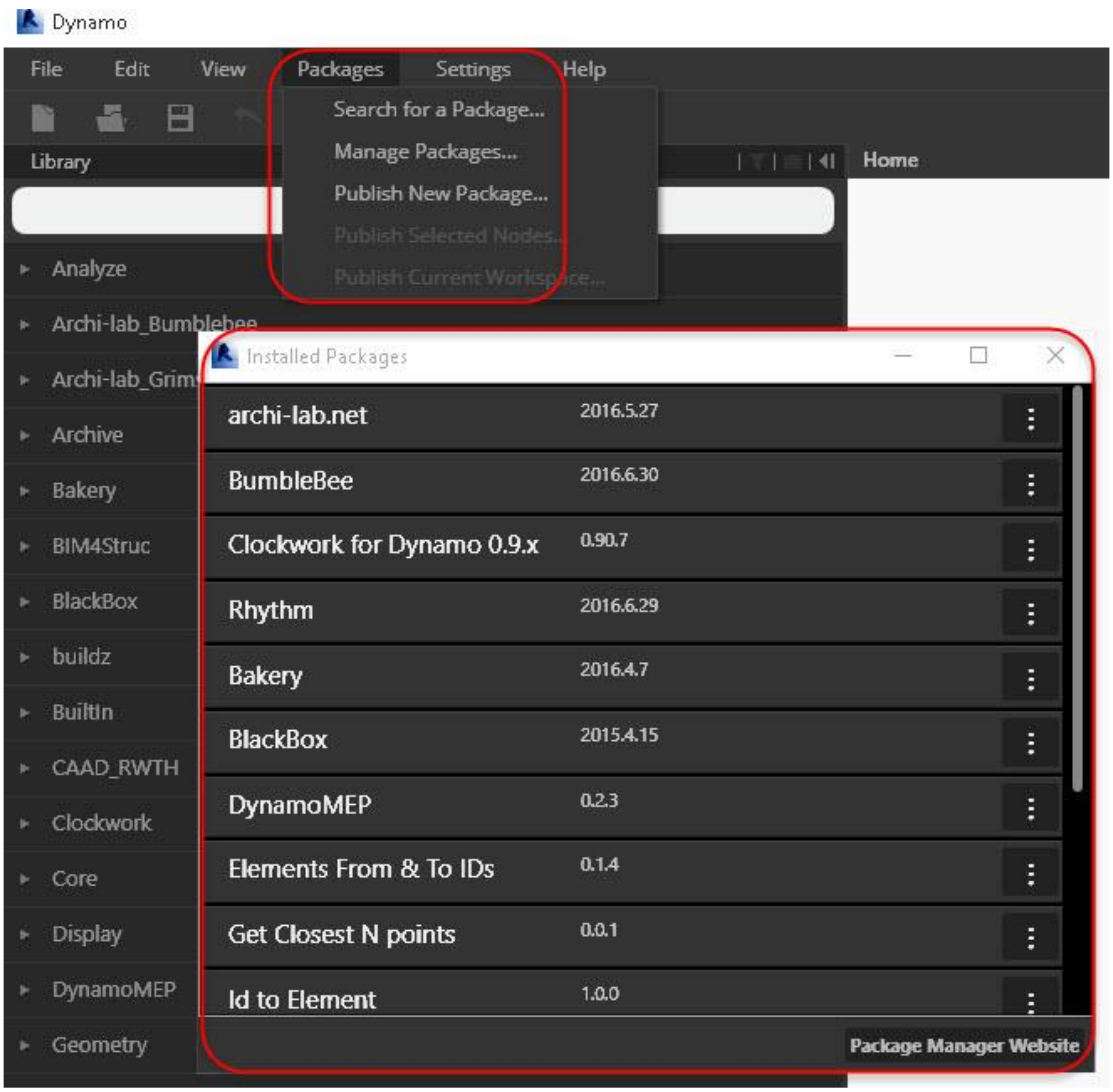


Dynamo is being developed at an outstanding pace with daily releases that are “beta” or un-tested. These releases allow a larger development and testing team that has led to the great growth of Dynamo. Stable builds are released around every 2 months.

Packages what are they?

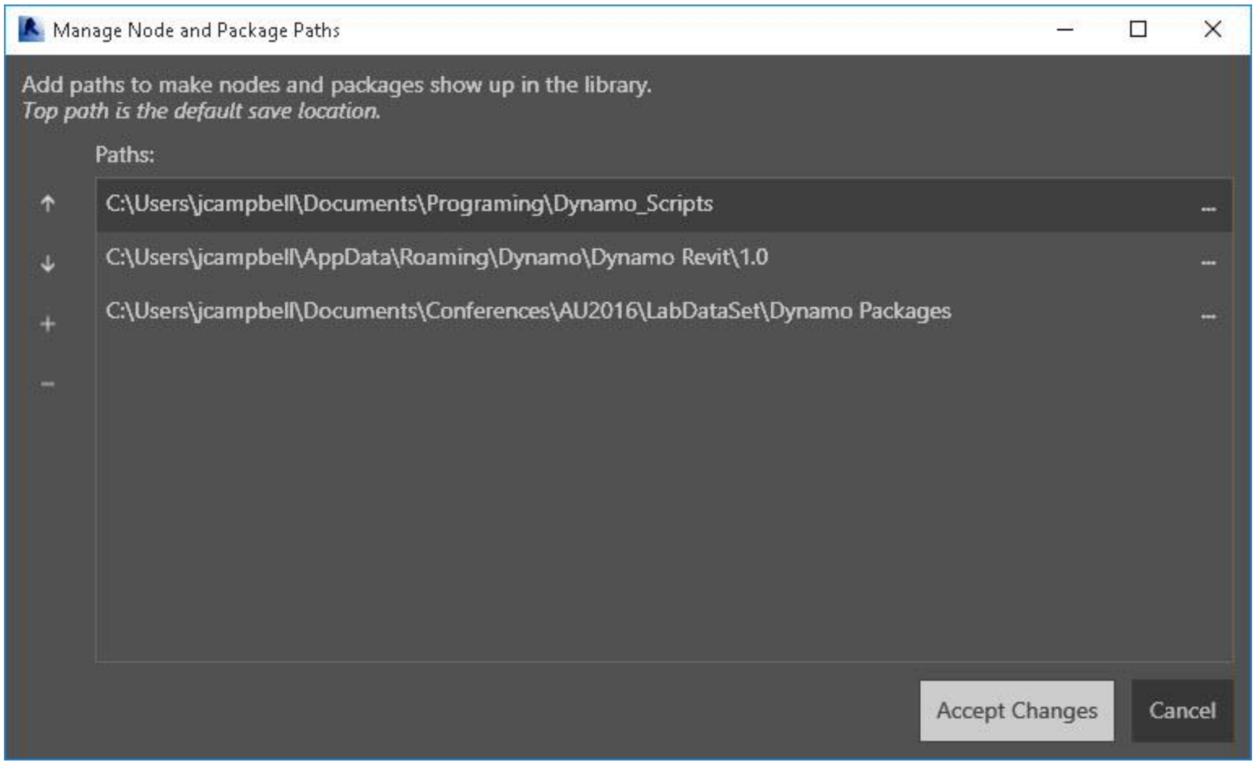
Packages are created and shared by the user community. They may contain a single node or multiple nodes that can be used to create your dynamo scripts. These packages bring additional functionality to Dynamo for those who are not proficient in using design script or the python language. Nodes can also be created and added to a package using traditional programing languages. These can be searched for by using the “Packages>Seach for a Package” pull down.

Packages are managed in the “Dynamo Package manager”. This can be accessed from within the Dynamo interface.



archi-lab.net	2016.5.27
BumbleBee	2016.6.30
Clockwork for Dynamo 0.9.x	0.90.7
Rhythm	2016.6.29
Bakery	2016.4.7
BlackBox	2015.4.15
DynamoMEP	0.2.3
Elements From & To IDs	0.1.4
Get Closest N points	0.0.1
Id to Element	1.0.0

When working with the packages you can also download them and store them in a shared resource location so that versioning can be controlled. This can be accessed from "Settings>Manage Node and Package Paths".



Manage Node and Package Paths

Add paths to make nodes and packages show up in the library.
Top path is the default save location.

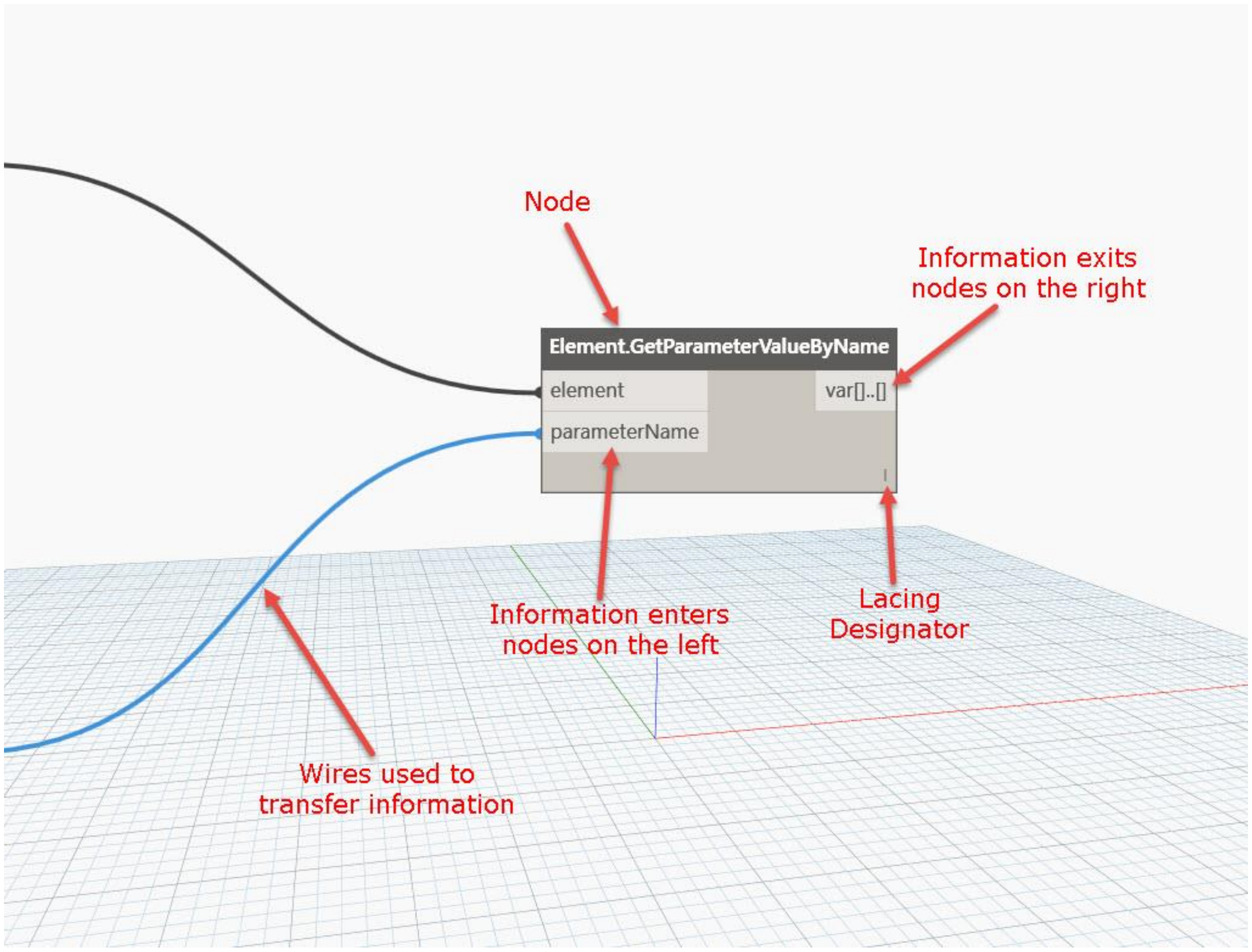
Paths:

C:\Users\jcampbell\Documents\Programing\Dynamo_Scripts
C:\Users\jcampbell\AppData\Roaming\Dynamo\Dynamo Revit\1.0
C:\Users\jcampbell\Documents\Conferences\AU2016\LabDataSet\Dynamo Packages

The creators of these packages will update them based on the Dynamo build. Care should be taken when making use of packages since they are created based on a specific version of Dynamo. When upgrading your base Dynamo the package may not be available causing errors in your definitions.

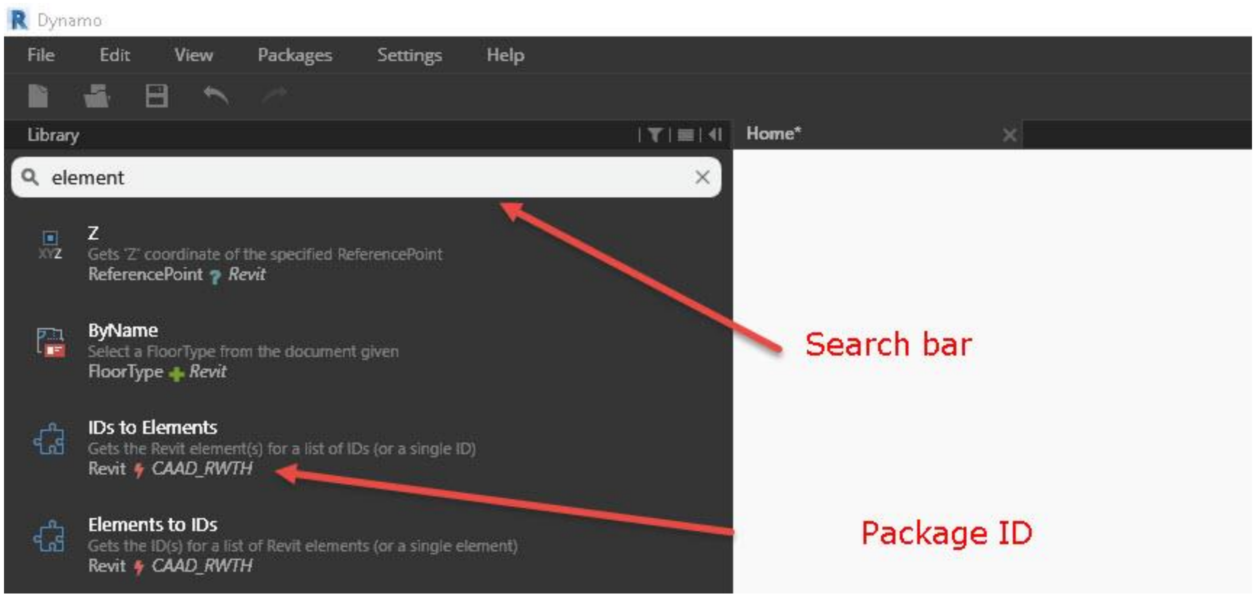
Basic Building block

The node is the basic building block of Dynamo. These nodes represent a function of the Revit API. Wires are then placed between the nodes in order to pass the information. Wires will only flow in one direction.


Node
Information exits nodes on the right
Element.GetParameterValueByName
element
var[]..[]
parameterName
Information enters nodes on the left
Lacing Designator
Wires used to transfer information

Dynamo Interface

The Dynamo canvas is broken up into several areas. Much like the Revit interface you have a drawing area were you place your nodes and a browser tree that contains all of the programing nodes that you have available. The “search” feature located at the top of the browser menu allows you to find nodes based on key words. When using a node you can identify if it is provided by the core Dynamo application or from a package you have added by the identification under the command.



Z
Gets 'Z' coordinate of the specified ReferencePoint
ReferencePoint Revit

ByName
Select a FloorType from the document given
FloorType Revit

IDs to Elements
Gets the Revit element(s) for a list of IDs (or a single ID)
Revit CAAD_RWTH

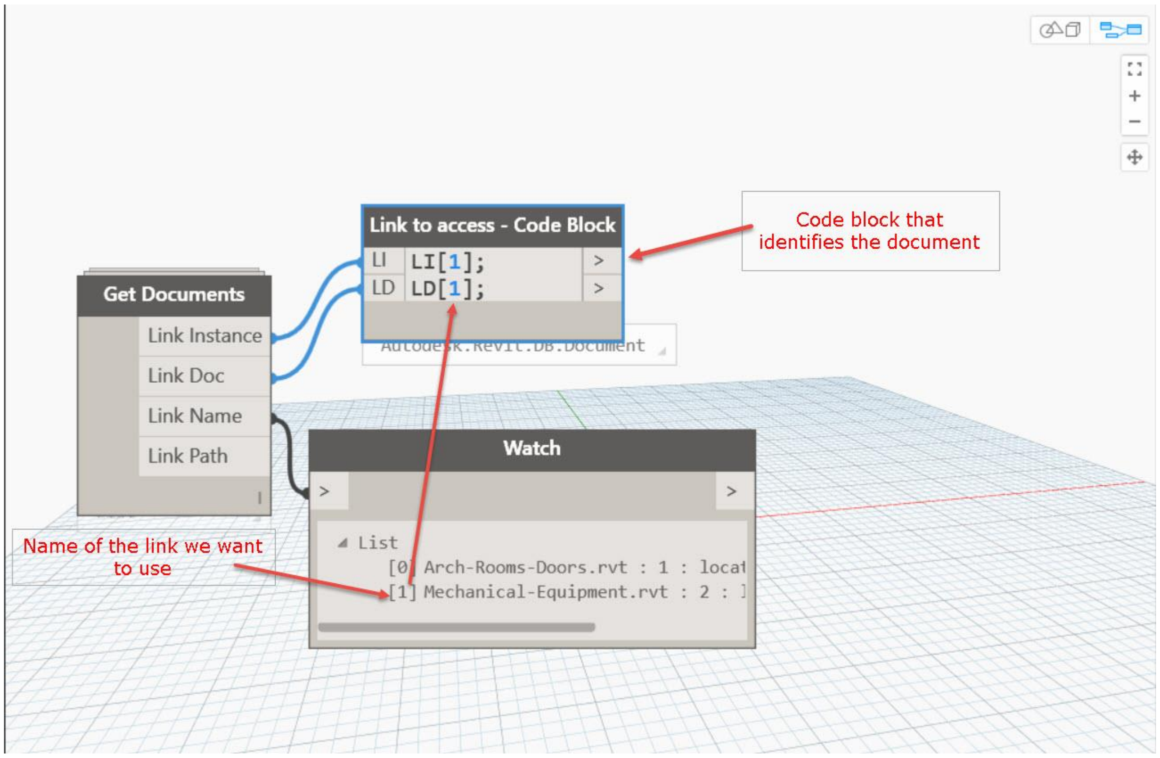
Elements to IDs
Gets the ID(s) for a list of Revit elements (or a single element)
Revit CAAD_RWTH

Accessing links

Dynamo has the ability to not only to crate or modify Revit geometry but also to access and transfer the data that is found in a rich BIM model. One of the things that Dynamo can be used for is to transfer information between linked models. This is a feature that could only be accomplished through the use of the Revit API. With some simple nodes Dynamo has opened up the ability for users to gather and utilize information.

Gain access to a linked model

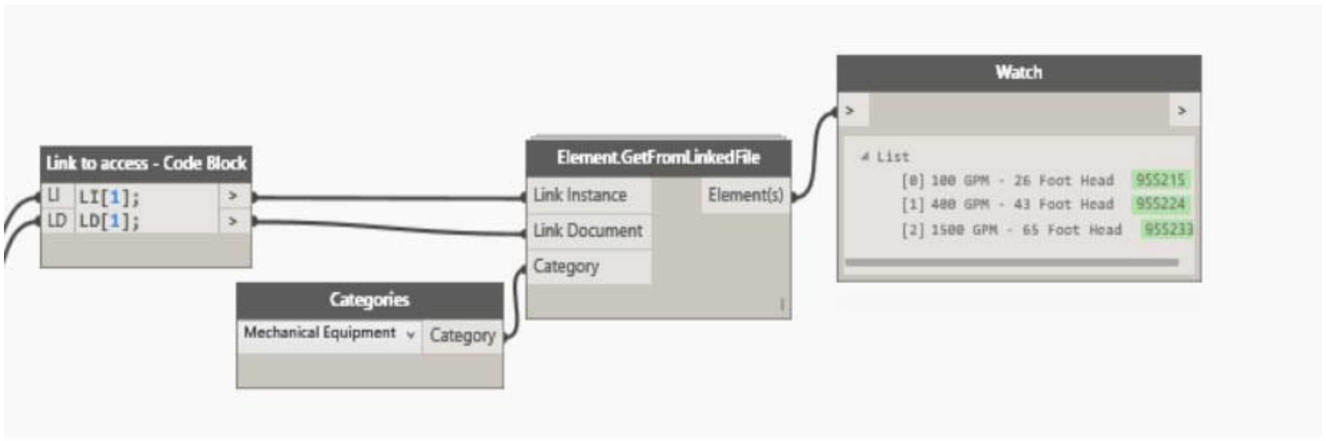
Let's first take a look at how to gain access to information in a linked file. We will start by placing the following nodes. The one used in my example is from the Archi-Labs Package titles "Get Documents". This node allows us to find all of the linked documents in the active model.


Get Documents
Link Instance
Link Doc
Link Name
Link Path
Link to access - Code Block
LI
LD
Autodesk.Revit.DB.Document
Watch
Code block that identifies the document
Name of the link we want to use

```
LI[1];
LD[1];
```

```
List
  [0] Arch-Rooms-Doors.rvt : 1 : locat
  [1] Mechanical-Equipment.rvt : 2 : ]
```

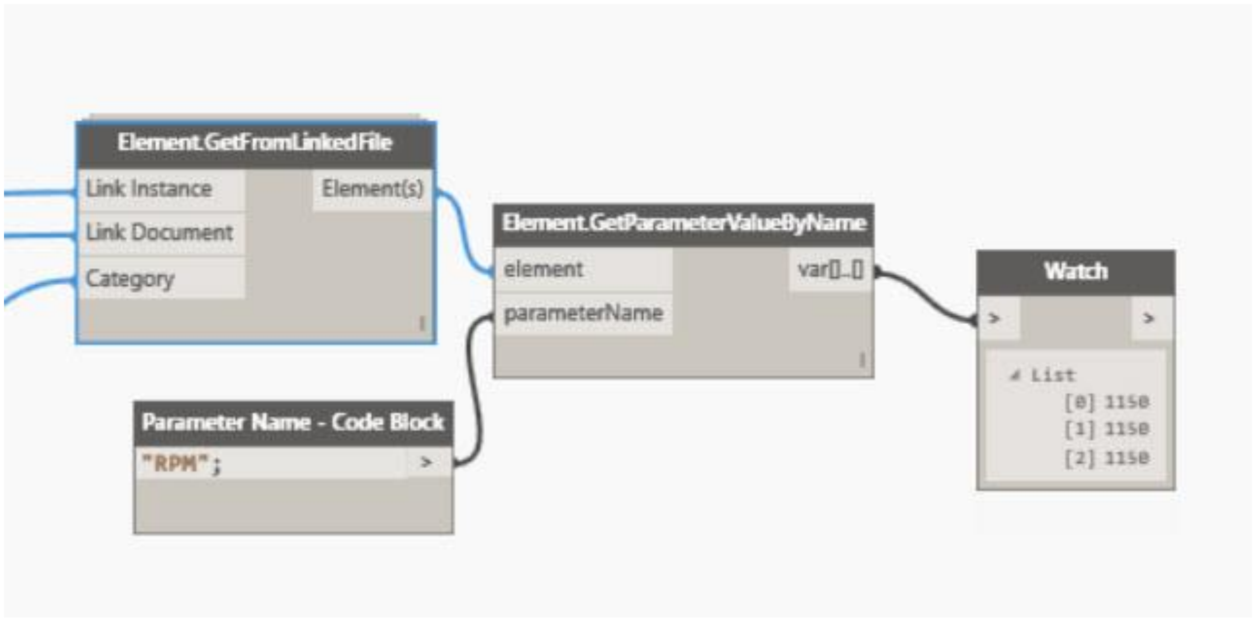
Once we have determined what link we would like to access we can use a code block to specify the Instance and document. Adding the “Element.GetFromLinkedFile” from Steamnodes package then allows us to gain access to a specific category of elements in the linked models.



```
LI[1];
LD[1];
```

```
List
    [0] 100 GPM - 26 Foot Head  955215
    [1] 400 GPM - 43 Foot Head  955224
    [2] 1500 GPM - 65 Foot Head  955233
```

Once the elements have been collected we can use the “Element.GetParatmeterValueByName” node from the core Dynamo product in order to pull information pertaining to instance parameters.


Element.GetFromLinkedFile
Link Instance
Link Document
Category
Element(s)
Element.GetParameterValueByName
element
parameterName
var[]..[]
Parameter Name - Code Block
Watch

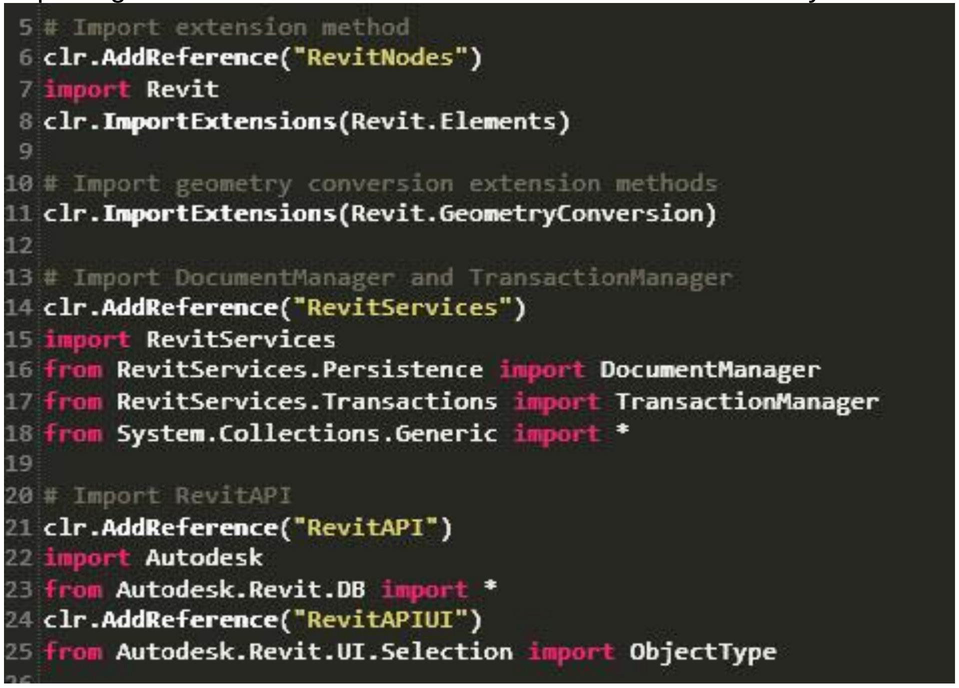
```
"RPM";
```

```
List
    [0] 1150
    [1] 1150
    [2] 1150
```

Selecting Individual link elements

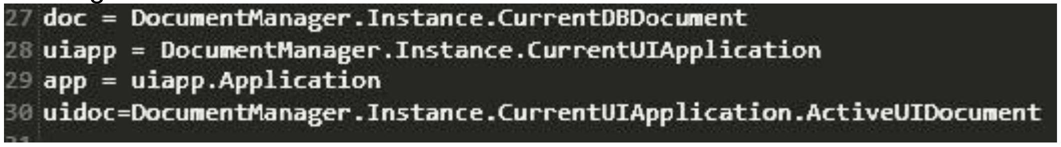
We can also utilize the Python scripting language inside of Dynamo in order to select individual objects inside of our linked Revit models. Inserting a Python code block and examining the individual sections we would need to place inside of it would look similar to the example below.

Importing the items needed to access the Revit API with the Python shell



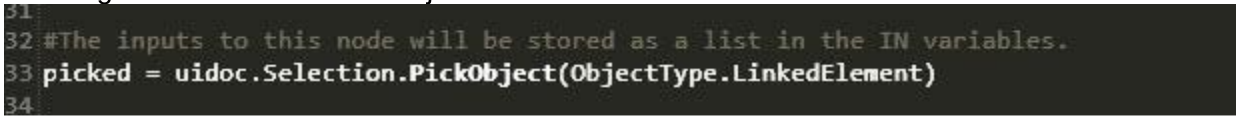
```python
# Import extension method
clr.AddReference("RevitNodes")
import Revit
clr.ImportExtensions(Revit.Elements)

# Import geometry conversion extension methods
clr.ImportExtensions(Revit.GeometryConversion)

# Import DocumentManager and TransactionManager
clr.AddReference("RevitServices")
import RevitServices
from RevitServices.Persistence import DocumentManager
from RevitServices.Transactions import TransactionManager
from System.Collections.Generic import *

# Import RevitAPI
clr.AddReference("RevitAPI")
import Autodesk
from Autodesk.Revit.DB import *
clr.AddReference("RevitAPIUI")
from Autodesk.Revit.UI.Selection import ObjectType
```

Gaining access to the active document database and the Revit user interface



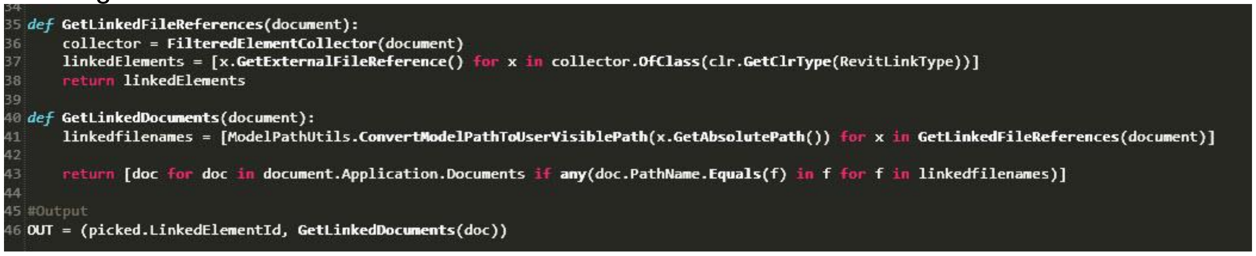
```python
doc = DocumentManager.Instance.CurrentDBDocument
uiapp = DocumentManager.Instance.CurrentUIApplication
app = uiapp.Application
uidoc=DocumentManager.Instance.CurrentUIApplication.ActiveUIDocument
```

Allowing the user to select an object from the linked file



```python

#The inputs to this node will be stored as a list in the IN variables.
picked = uidoc.Selection.PickObject(ObjectType.LinkedElement)

```

Defining functions to find the element ID of the selected element in the linked file



```
def GetLinkedFileReferences(document):
    collector = FilteredElementCollector(document)
    linkedElements = [x.GetExternalFileReference() for x in collector.OfClass(clr.GetClrType(RevitLinkType))]
    return linkedElements

def GetLinkedDocuments(document):
    linkedfilenames = [ModelPathUtils.ConvertModelPathToUserVisiblePath(x.GetAbsolutePath()) for x in GetLinkedFileReferences(document)]

    return [doc for doc in document.Application.Documents if any(doc.PathName.Equals(f) in f for f in linkedfilenames)]

#Output
OUT = (picked.LinkedElementId, GetLinkedDocuments(doc))
```

Resources

Below is a list of helpful resources when working with Dynamo in Revit.

1. Dynamobim.org
2. therevitcomplex.blogspot.com
3. Dynamoprimer.com