



Tips'n'tricks

Perkins + Will

26/07/2017




Manage Revision Clouds With Dynamo

Mauro Sabiu - Perkins+Will



Current Tools

Revision + Cloud Manager



Revision + Cloud Manager
DZ Studio
★★★★★ (3 reviews)

Like 6

Tweet

OS:

Win32 and 64

Language:

English

Description

Smart manage revisions, clouds and sheets.

Revision Cloud Manager:

- Automatically find corresponding sheet and detail number.
- Export to Excel schedule.
- Edit and update clouds in real time.
- Shows who is the last-editor of a revision cloud. (need to install End User Assistant)

USD 10.00

Qty to Purchase

Sub total: USD 10.00

Buy Now

Download Size:

1.7 MB

Release Date:

4/30/2016

Last Updated:

6/29/2017

Version Info:

1.2.3

Website:

<https://zhuliyi.org/>

Cust. Support:

david.liyi.zhu@gmail.com

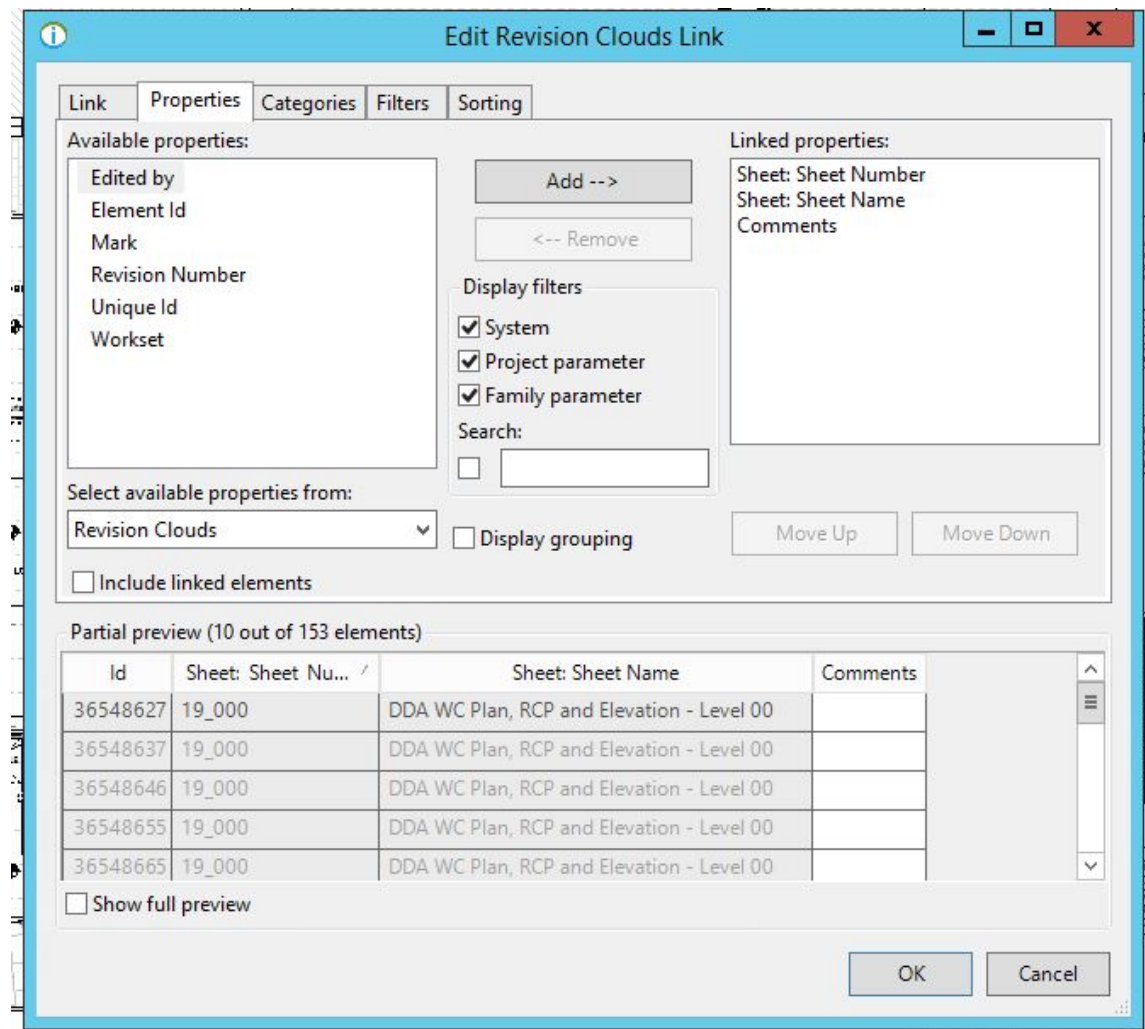
Compatible with:

Autodesk Revit
Version: 2018 , 2017 , 2016 , 2015 , 2014
Autodesk Revit Architecture



Current Tools

Revision + Cloud Manager



Dynamo Workflow

Why?

- Place Clouds Comments on Sheets
- Print Set as input
- Customization



Dynamo Workflow

The screenshot displays a Dynamo workflow titled "ManageRevisionClouds-test3.dyn*" within the Revit interface. The workflow is organized into a vertical column of nodes on the right side of the screen, with a large pink background area labeled "INPUTS" at the top. The nodes include:

- Code Block**: "SET_01";
- Boolean**: True (selected), False
- Text Upper?**
- Boolean**: True (selected), False
- Text Width**
- Integer Slider**: 45
- Write Excel?**
- Boolean**: True (selected), False
- Excel File Path**
- File Path**: Browse... (selected), with a file path "New Foglio di lavoro di Microsoft Excel.xlsx" below it.
- Worksheet Name**
- Code Block**: "Foglio1";

The left side of the screen shows the Revit interface with a section titled "Architect" and "PERKINS+WILL". Below this, there is a "Notes" section with the text: "All information, areas, dimensions and datum levels are subject to detailed survey, detailed design and investigative work. Do not scale from drawing." and a "Revision Log" section. At the bottom, there is a "Key Plan" section with the text: "This drawing is the property of Perkins+Will UK Ltd. No disclosure or copy of it may be made without the written permission of Perkins+Will UK Ltd."

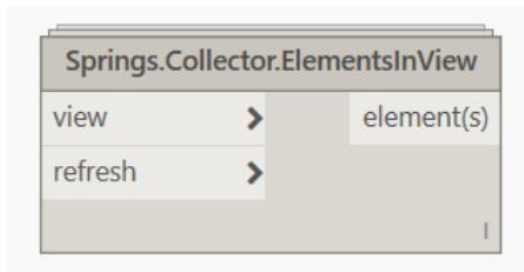


How

Collector.ElementsInView

SpringNodes Package

By Dimitar Venkov



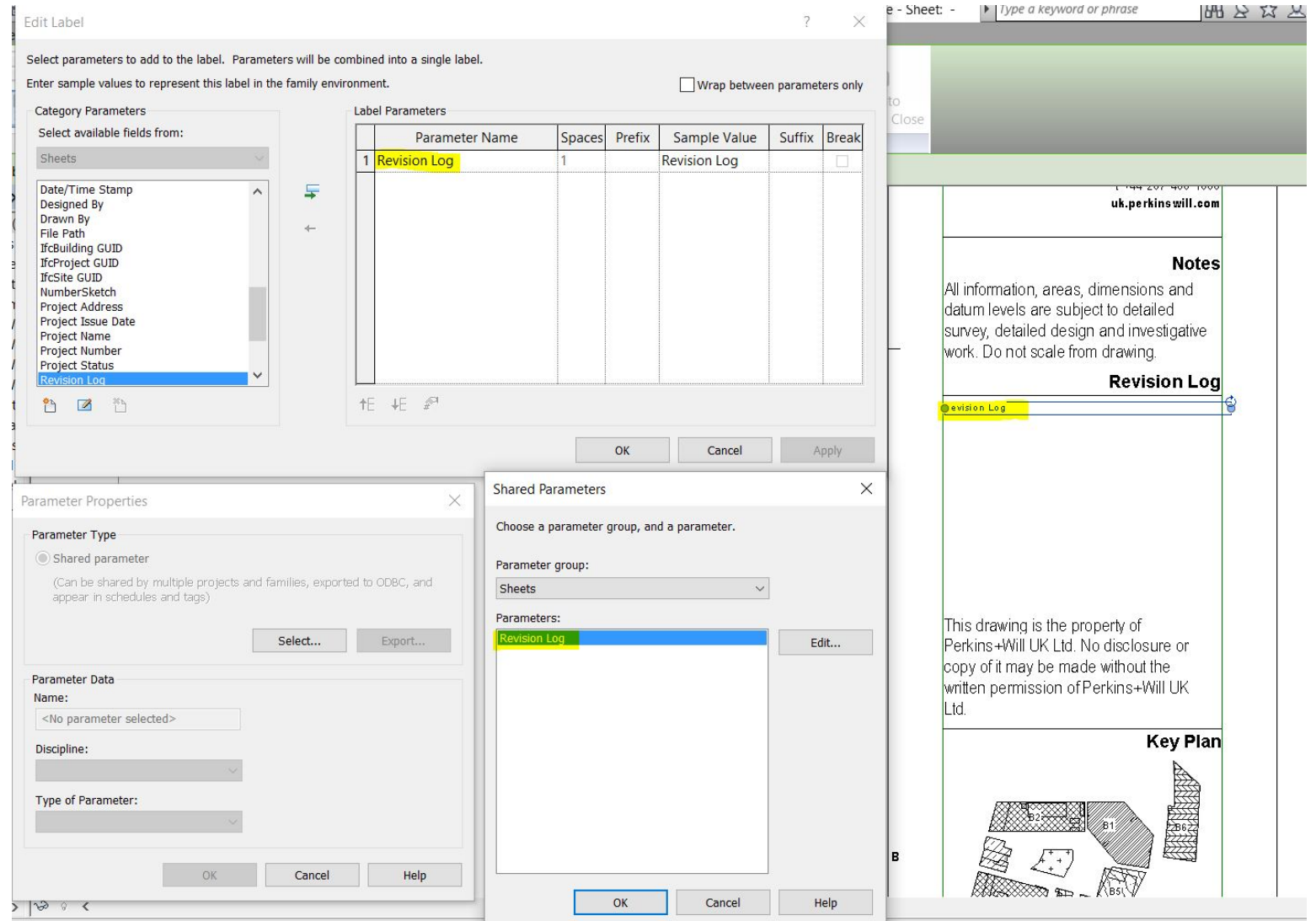
```
Edit Python Script...
1 #Copyright(c) 2016, Dimitar Venkov
2 # @Sdevene, dimitar.venkov@gmail.com
3
4 import clr
5
6 clr.AddReference('RevitServices')
7 import RevitServices
8 from RevitServices.Persistence import DocumentManager
9 doc = DocumentManager.Instance.CurrentDBDocument
10
11 clr.AddReference('RevitAPI')
12 from Autodesk.Revit.DB import *
13
14 clr.AddReference('RevitNodes')
15 import Revit
16 clr.ImportExtensions(Revit.Elements)
17
18 def tolist(obj1):
19     if hasattr(obj1, '__iter__'): return obj1
20     else: return [obj1]
21
22 def output1(l1):
23     if len(l1) == 1: return l1[0]
24     else: return l1
25
26 views = UnwrapElement(tolist(IN[0]))
27 elements = []
28
29 for i in xrange(len(views)):
30     fec = FilteredElementCollector(doc, views[i].Id).OfClass(RevisionCloud).GetElementIterator()
31     fec.Reset()
32     view_el = []
33     while fec.MoveNext():
34         view_el.append(fec.Current.ToDSType(True))
35         elements.append(view_el)
36
37 OUT = output1(elements)
```



How

Set sheets text shared parameter.

And place the label in the titleblocks.



TextWrap

```
import clr
clr.AddReference('ProtoGeometry')
from Autodesk.DesignScript.Geometry import *
import sys
sys.path.append(r'C:\Program Files (x86)\IronPython 2.7\Lib')
import textwrap
```

#The inputs to this node will be stored as a list in the IN variables.

dataEnteringNode = IN

text = IN[0]

len = IN[1]

def wraptext(tx, l):

 b = textwrap.dedent(tx).strip()

 if l<=1:

 l = 1

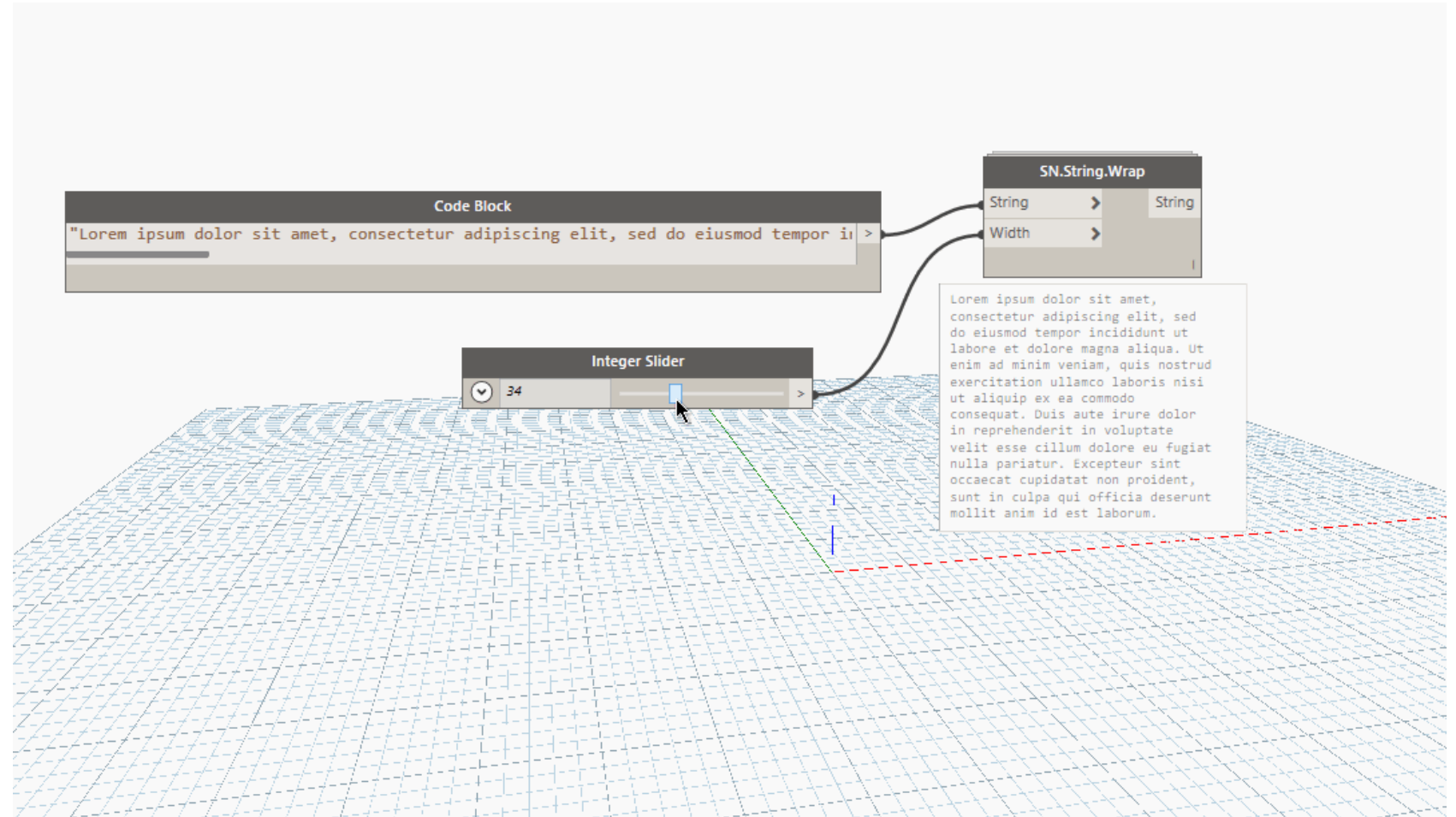
 for width in [l]:

 c = textwrap.fill(b, width=width)

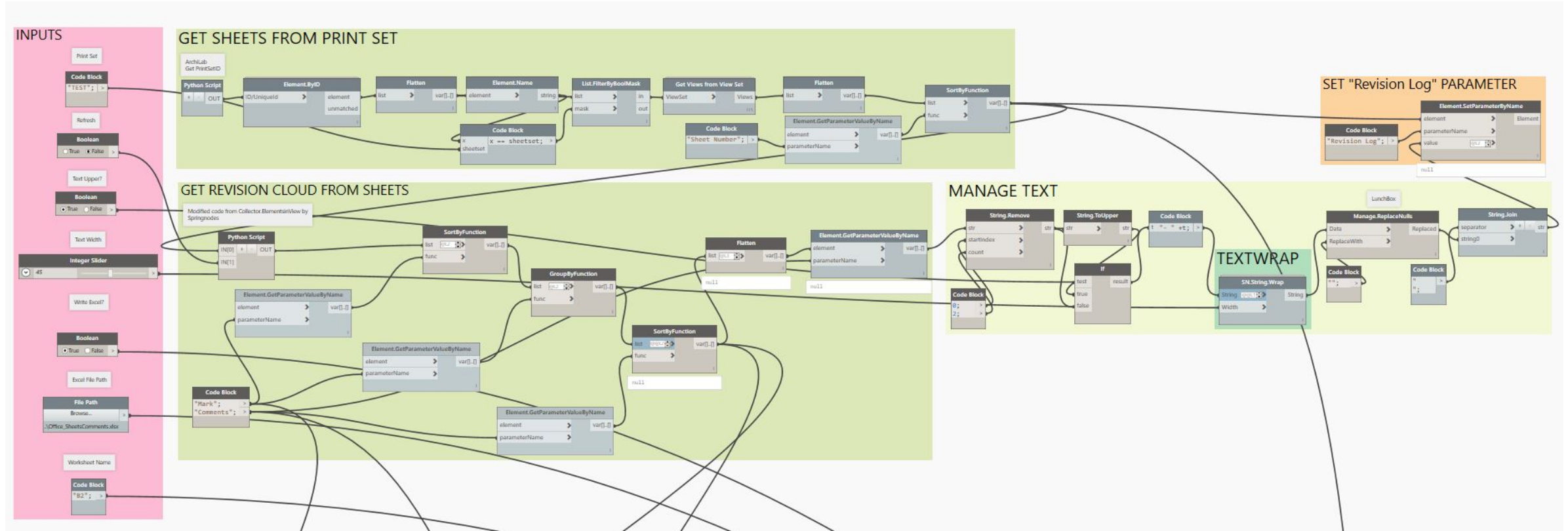
 return c

#Assign your output to the OUT variable.

OUT = wraptext(text, len)



Dyn Definition



Inputs

Thank you

Data-Shapes | Multi Input UI ++

Manage Revision Clouds

SheetSet

Upper Text? ☒ Yes/No

Character Break 40

Write Excel? ☐ Yes/No

Excel File Path

Excel Worksheet Name

**P E R K I N S
+ W I L L**



Design Script - Functions

The easy way to control parameters



Defining a Function within a Code Block

Extract from Dynamo Primer –

“Functions can be created in a code block and recalled elsewhere in a Dynamo definition. This creates another layer of control in a parametric file, and can be viewed as a text-based version of a custom node. In this case, the "parent" code block is readily accessible and can be located **anywhere** on the graph. No wires needed!”

http://dynamoprimer.com/en/07_Code-Block/7-4_functions.html

```
Code Block

def SearchMark()
{
    return = "RAD";
};

def TypeParams()
{
    return = {
        "Manufacturer", //Index 0
        "Model", //Index 1
        "Thermal Output", //Index 2
        "Water Flow Rate", //Index 3
        "Pipe Connections Size", //Index 4
        "Type Comments", //Index 5
        "Type Mark";
    //Index 6;
};

def InstanceParams()
{
    return = {
        "Mark", //Index 0
        "Level", //Index 1
        "Schedule Level", //Index 2
        "Type Id", //Index 3
        "Height", //Index 4
        "Length";
    //Index 5;
};

def ExcelHeaders()
{
    return = {"TypeID", "UniqueID", "Ref No.", "Location", "Manufacturer",
        "Model", "Output (W)", "Flow Rate (kg/s)", "Size (WxH)", "Connections",
        "Notes"};
};
```



Defining a Function within a Code Block

Within the Function you can define lists of Strings, Numbers or Boolean values. Anything you can have in a list, you can have in a list within a Function.

These can then be recalled anywhere on the graph without the need for lengthy wires, or node duplication.

If you need to change the parameter value simply change it in the Function no need to search the graph.

```
Code Block
def SearchMark()
{
    return = "RAD";
};

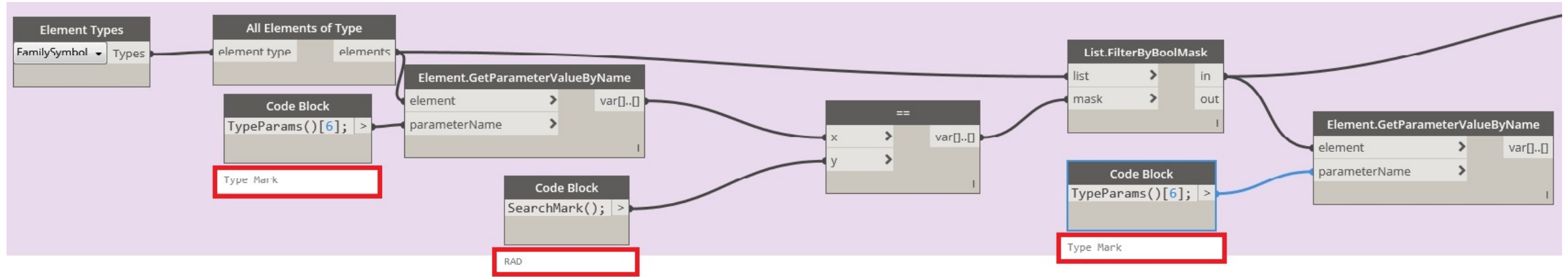
def TypeParams()
{
    return = {
        "Manufacturer", //Index 0
        "Model", //Index 1
        "Thermal Output", //Index 2
        "Water Flow Rate", //Index 3
        "Pipe Connections Size", //Index 4
        "Type Comments", //Index 5
        "Type Mark"};
    //Index 6;
};

def InstanceParams()
{
    return = {
        "Mark", //Index 0
        "Level", //Index 1
        "Schedule Level", //Index 2
        "Type Id", //Index 3
        "Height", //Index 4
        "Length"};
    //Index 5;
};

def ExcelHeaders()
{
    return = {"TypeID", "UniqueID", "Ref No.", "Location", "Manufacturer",
        "Model", "Output (W)", "Flow Rate (kg/s)", "Size (WxH)", "Connections",
        "Notes"};
};
```



Using the **defined** Function in a graph



As with normal lists you recall the whole list –
FunctionName() ;

Or an index of that list -
FunctionName() [0];



Dynamo Coordinate System

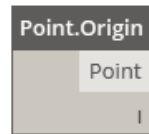
The easy way to control parameters



Dynamo Coordinate System

Assuming that:

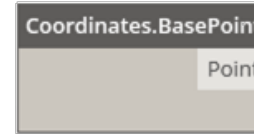
Dynamo



Point(X = 0.000, Y = 0.000, Z = 0.000)

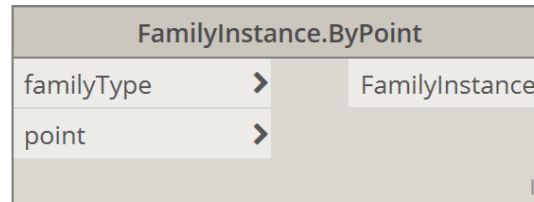
=

Revit

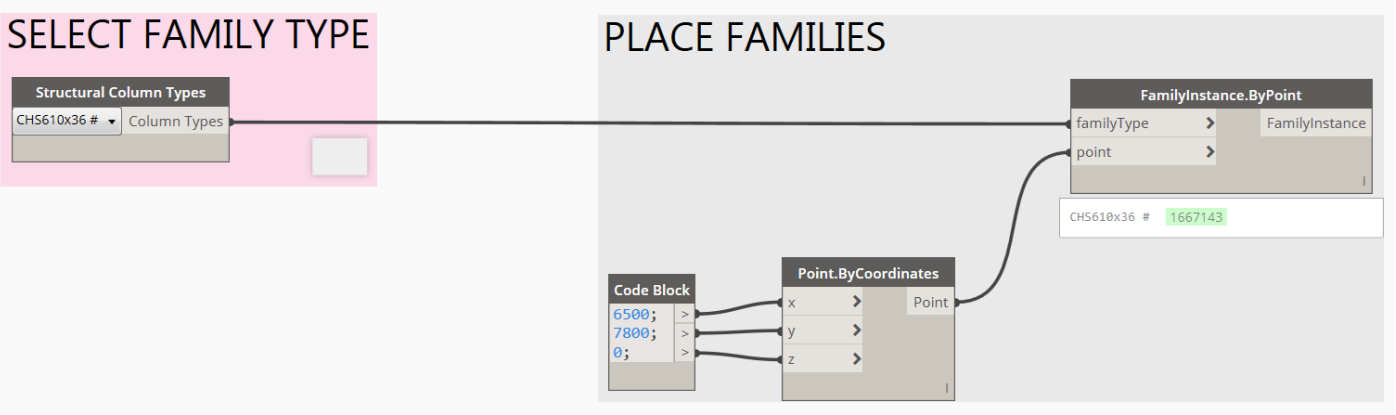
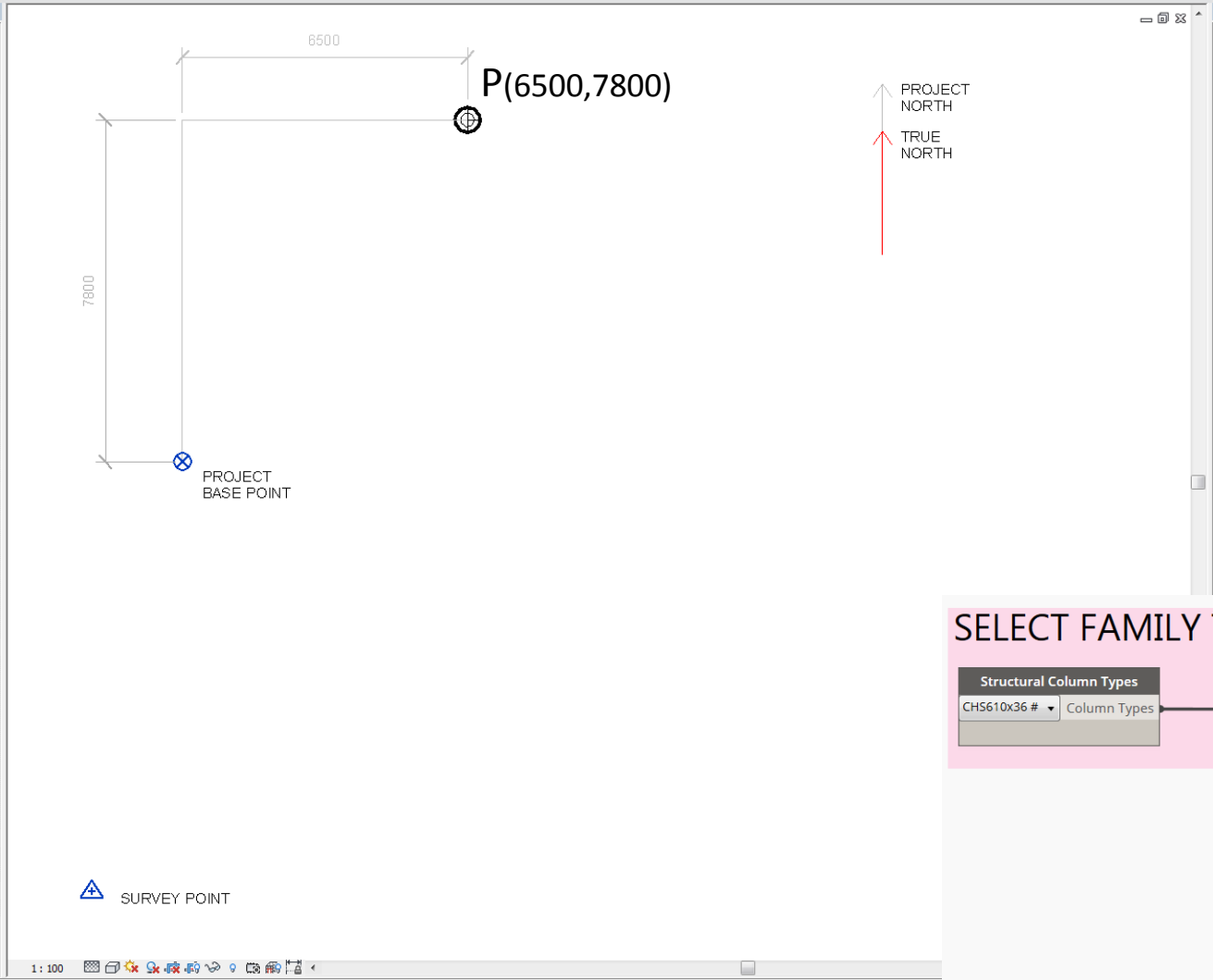


Point(X = 0.000, Y = 0.000, Z = 0.000)

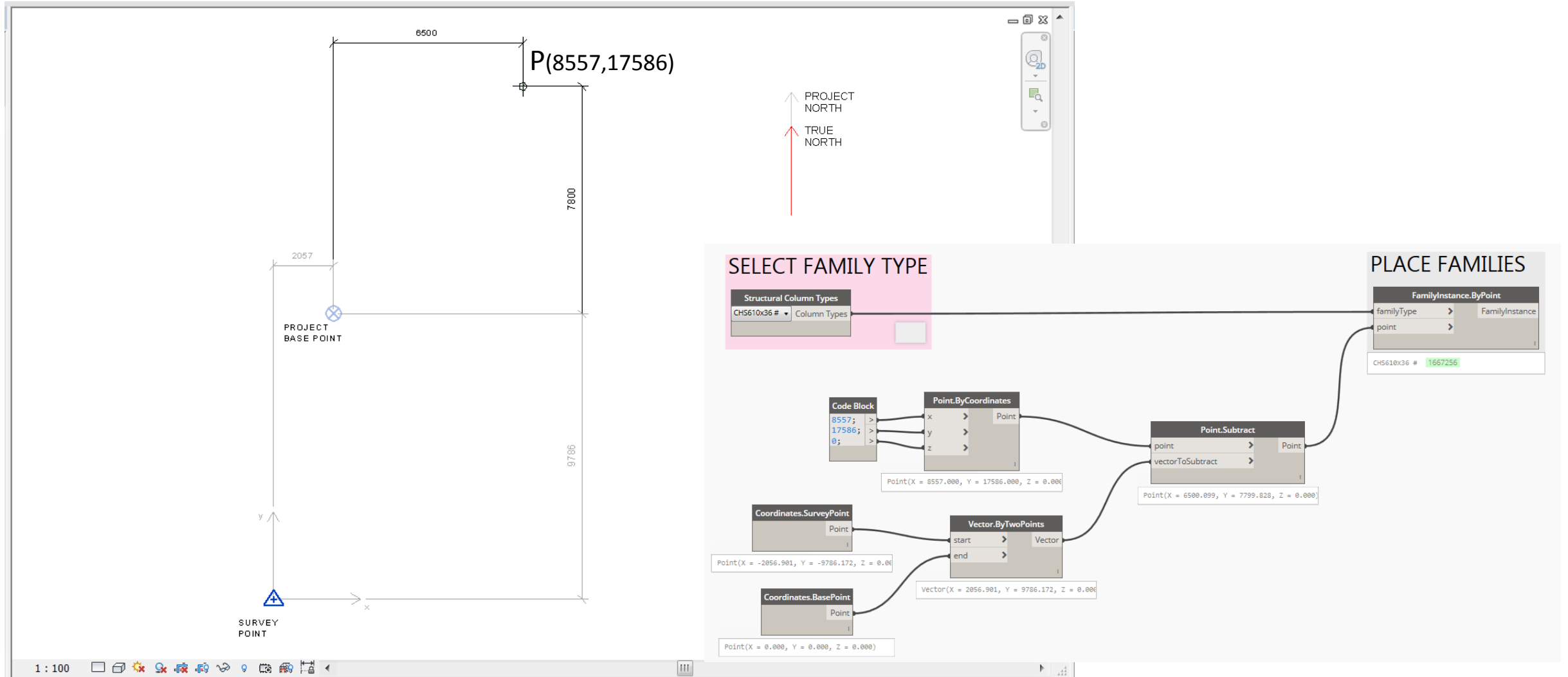
How do I place a family at a datum point P in Revit?



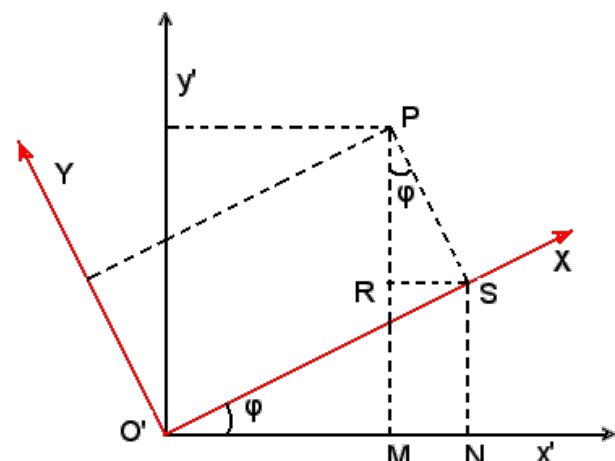
Scenario #1: P_{coord} from PBP, Project North = True North



Scenario #2: P_{coord} from SP, Project North = True North

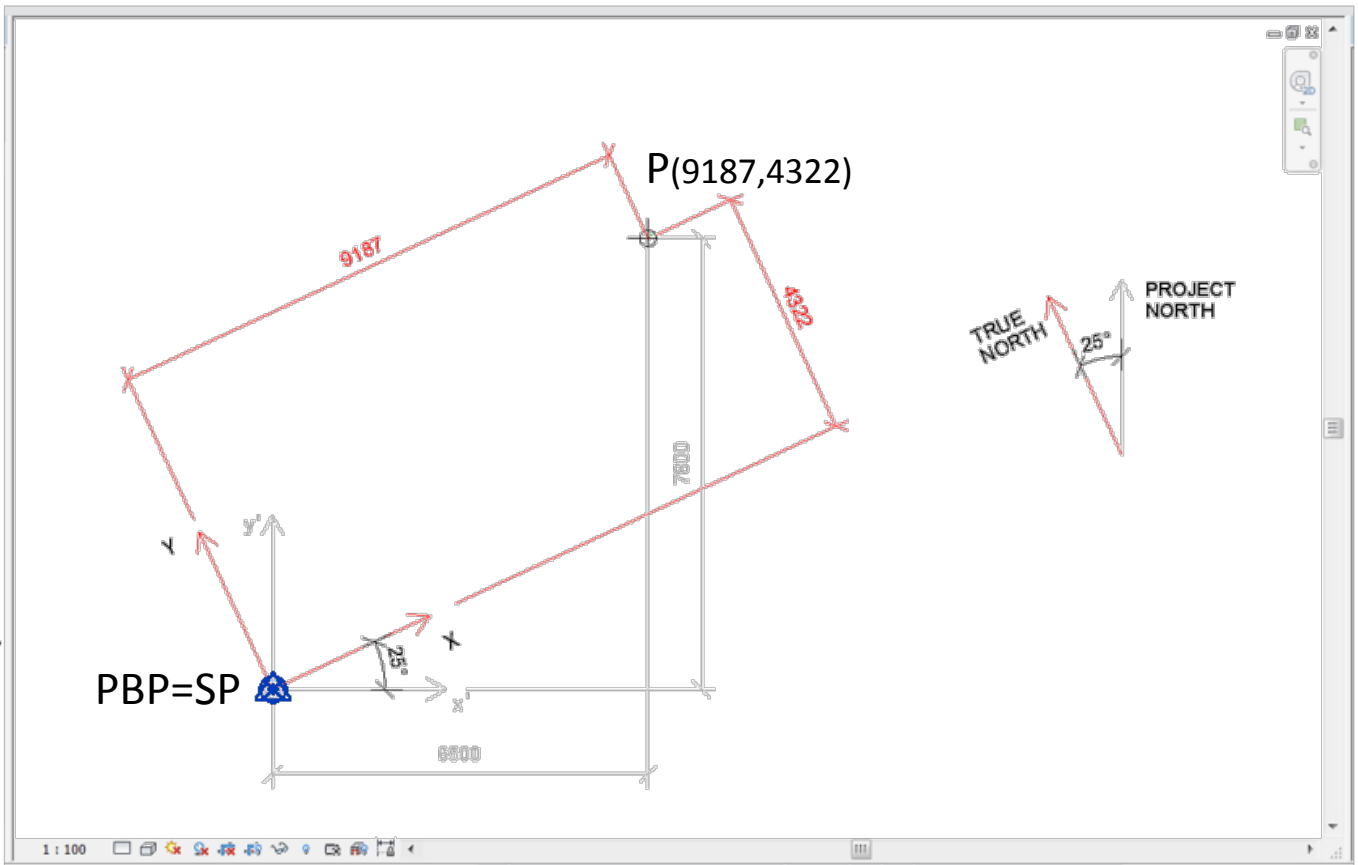


Scenario #3: P_{coord} from SP, Project North ≠ True North



$X = x' \cos\phi + y' \sin\phi$
 $Y = -x' \sin\phi + y' \cos\phi$

$x' = X \cos\phi - Y \sin\phi$
 $y' = X \sin\phi + Y \cos\phi$



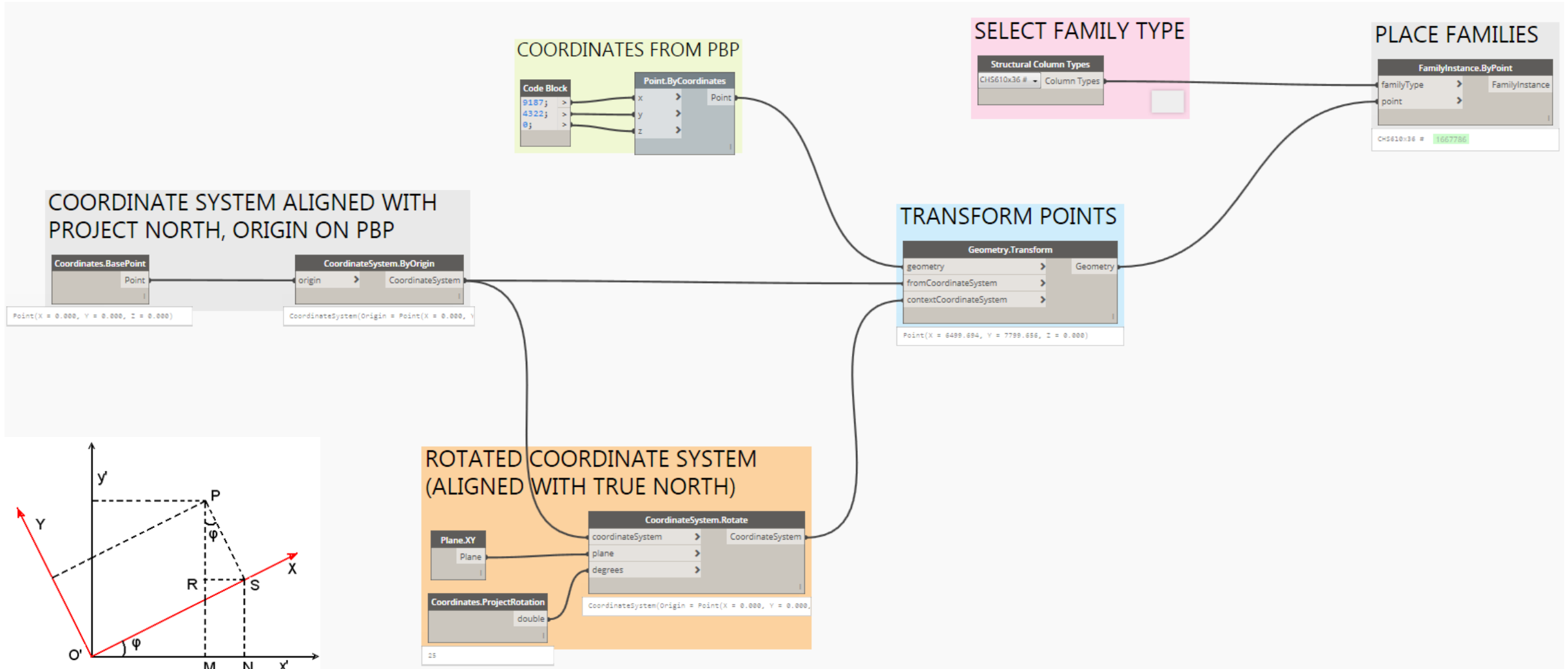
	degree	radians
ANGLE	25	0.436
sin	0.423	
cos	0.906	

POINT COORDINATES	
X	Y
9187	4322

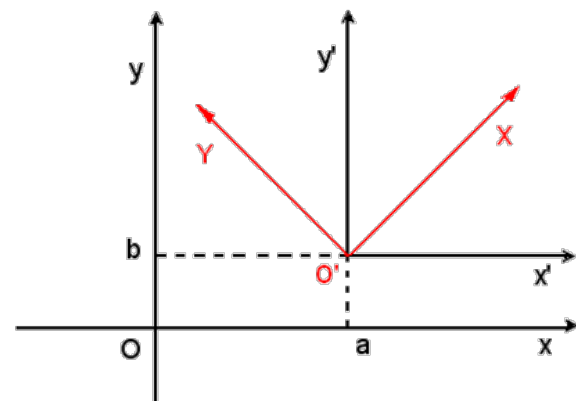
TRANSFORMED COORDINATES	
x'	y'
6500	7800



Scenario #3: P_{coord} from SP, Project North \neq True North



Scenario #4: P_{coord} from SP, Project North ≠ True North

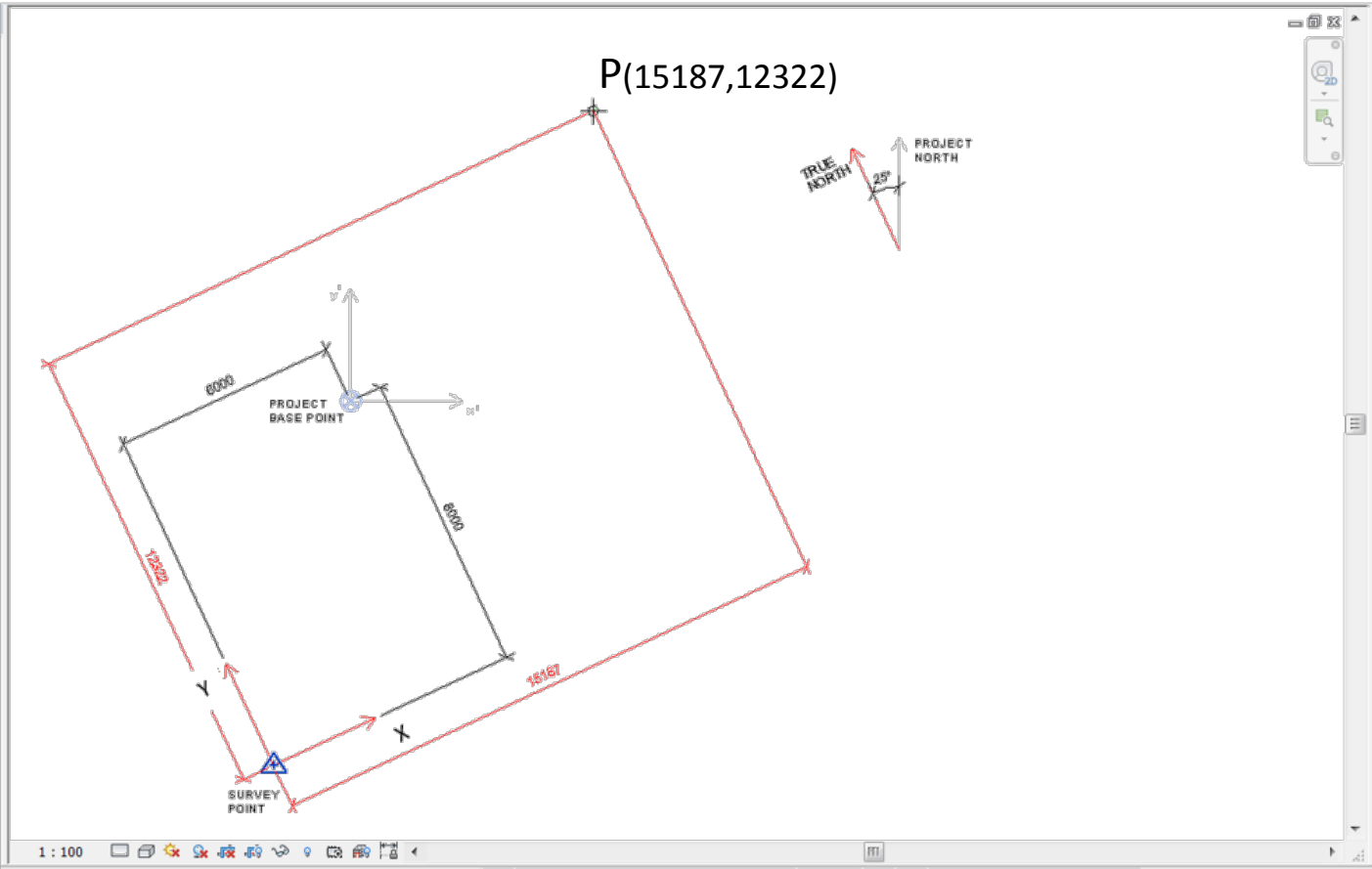


$$X = (x - a) \cos \phi + (y - b) \sin \phi$$

$$Y = -(x - a) \sin \phi + (y - b) \cos \phi$$

$$x' = (x - a) \cos \phi - (y - b) \sin \phi$$

$$y' = (x - a) \sin \phi + (y - b) \cos \phi$$



	degree	radians
ANGLE	25	0.436
sin	0.423	
cos	0.906	

POINT COORDINATES	
X	Y
15187	12322

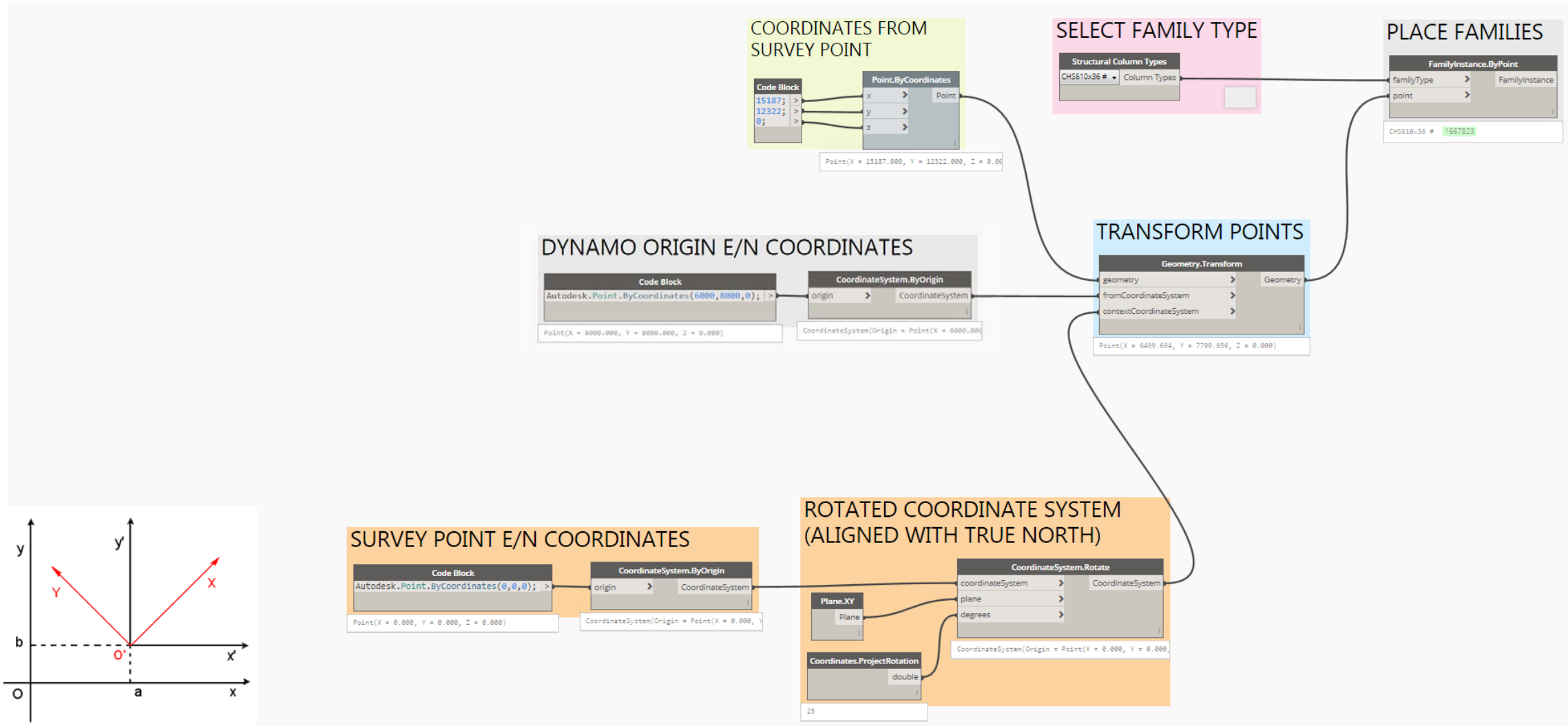
PBP COORDINATES	
a	b
6000	8000

X-a	Y-b
9187	4322

TRANSFORMED COORDINATES	
x'	y'
6500	7800



Scenario #4: P_{coord} from SP, Project North \neq True North



Think in terms of Eastern and Northings

