

# PropertyWizard Help

Release Notes V1-0-5

PropertyWizard is an add-in for Autodesk® Revit® that allows you to add formulas to project parameters, just as you can add formulas to family parameters in the family editor.

For example, with PropertyWizard you can:

- Set the 'Sheet Number' of all the sheets from their other parameters.
- Set the Door Numbers of all the doors, using data from the door's Level and Type.
- Set visible properties for X and Y location on all pile foundations so you can schedule them.

And best of all, because PropertyWizard is linked into Revit's parametric change engine, the parameter values are kept up-to-date all the time. No need to repeatedly run Dynamo scripts to update the parameters. #ReviseInstantly

Each PropertyWizard formula has 3 parts:

1. The Category of element that you want to apply it to.
2. The Target Property you want to set.
3. The Formula you want to use.

The Category can be any Revit Category. The formula will apply to all elements in the Category (you can use filtering in the formula to affect just a subset). Target Property has to be an Instance Property in this version, and can be any of these types: Number, Integer, Length, Area, Text, Yes/No or Angle.

The Formula language is an extension of the normal Revit family formula language, so if you can write Revit family formulas, you can write PropertyWizard formulas. The extensions include string manipulation and concatenation (which you don't have in family formulas), access to hidden properties from the Revit API, properties from linked elements, and Project Information properties.

PropertyWizard lets you place your Revit data just where you need it to be, so you can generate your data and drawing outputs efficiently, without having to run Dynamo scripts or manually copy data.

## Support

I want you to get the most out of PropertyWizard, so please, if you need any help or advice in using it, get in touch with me at [support@davidwooddesign.co.uk](mailto:support@davidwooddesign.co.uk).

## Contents

PropertyWizard Help .....	1
Support .....	1
Installation .....	2
Uninstall the Beta version.....	2
Getting Started.....	3
Reference .....	6
Running PropertyWizard.....	6
Activating PropertyWizard in a project.....	6
Creating a formula .....	7
Target Properties .....	7
Formulas .....	7
Source Properties.....	7
Operators and Functions: .....	8
Release Notes (What's New) .....	10
Compatibility.....	10
Beta versions.....	10

## Installation

### Uninstall the Beta version

If you have a Beta version of PropertyWizard, you have to uninstall it to enable the new version to work. Download the uninstaller from:

<http://www.davidwooddesign.co.uk/uninstall-propertywizard-beta/>

To manually uninstall, delete the 'PropertyWizard.addin' files and the 'DWD' folders from each of the version folders 2016 to 2019 in C:\ProgramData\Autodesk\Revit\Addins.

### Autodesk App Store downloads

If you downloaded PropertyWizard from the Autodesk App Store, follow the instructions that came with the package.

### DavidWoodDesign downloads

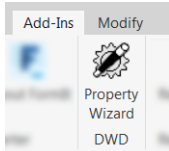
If you downloaded PropertyWizard from [www.davidwooddesign.co.uk](http://www.davidwooddesign.co.uk):

1. Close any Revit sessions before you start.
2. If you received a zip file, unzip it to reveal the installation exe file.
3. Run the InstallPropertyWizard\_V1-0-5.exe file. The installer will go through a couple of dialogs and then unpack the PropertyWizard components. It may need admin permission.

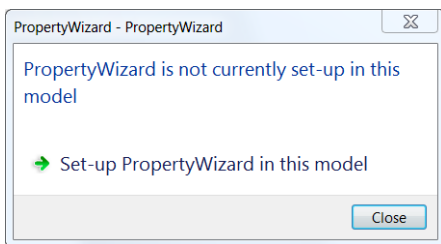
## Getting Started

This is an introductory tutorial for PropertyWizard.

1. Open a new Revit model.
2. Draw some walls.
3. Open PropertyWizard from the AddIns tab.

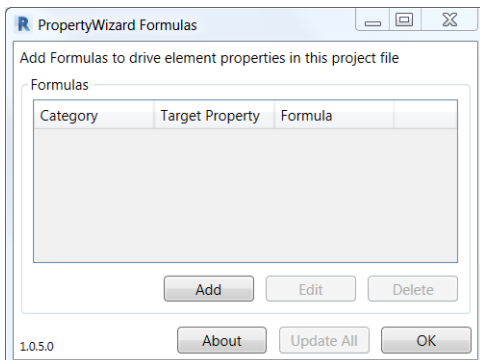


4. Since this is the first time you have run it in this model, PropertyWizard will show this dialog:



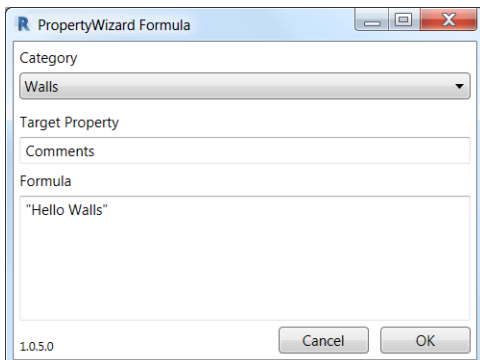
Select 'Set-up PropertyWizard in this model'.

5. This is the main PropertyWizard window:



6. Click 'Add' to add a new Formula.

7. This is the Formula window:



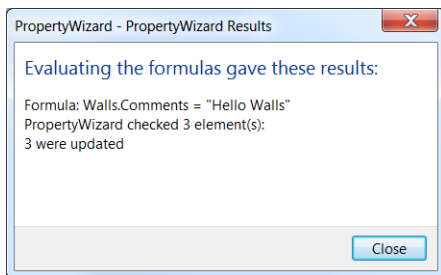
8. Enter these values to create your first Formula:

*Category:* Walls

*Target Property:* Comments

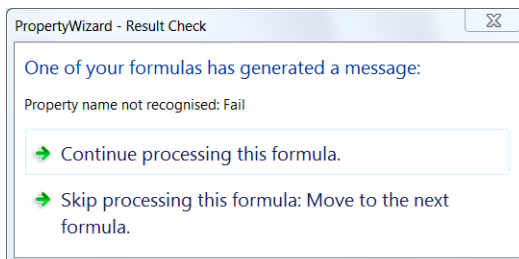
*Formula:* "Hello Walls"

9. Click 'OK' and then click 'UpdateAll' in the main PropertyWizard Window. This will update your walls' Comments property using your formula. PropertyWizard will show a results dialog like this:



The first line of the result shows the formula. The second line shows how many elements (walls) were checked, and the remaining lines will show how many elements were up-to-date, updated, or had errors.

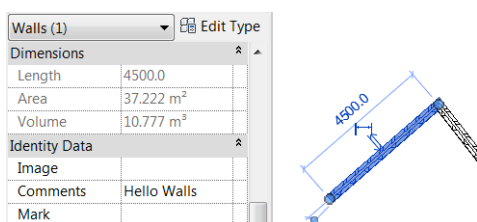
If PropertyWizard finds an error while processing your formula, it will show a dialog like this:



You can choose to continue processing this formula for the remaining elements, or to stop processing this formula and skip the remaining elements.

If you get an error, check your typing and try again. Be sure to target *Comments* (not *comments* or *Comment*) and include straight (not curly) quotes around "Hello Walls".

10. Click 'OK' to close the main PropertyWizard Window, and check the properties of your walls:



This is static text, and will not change, but it has been set for every wall in your project.

Now try getting a value from another property.

11. Get the value of the walls' Name property:

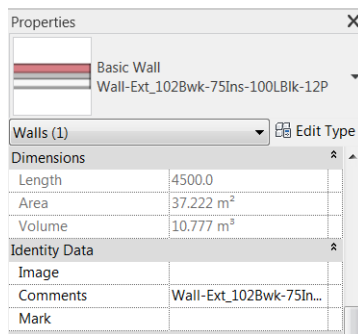
*Category:* Walls

*Target Property:* Comments

*Formula:* Name

The Comments property of each wall now contains the 'Name' of the wall, which is generally the wall's Type name.

Change the Type name, and the Comments property will change to match.



The Comments property is a text, or 'string' property. Now try a numerical property.

12. Use Project Parameters to create a new wall parameter called 'TestNumber' of Type 'Number'. Number parameters can store a floating-point number.

*Category:* Walls

*Target Property:* TestNumber

*Formula:* 42

13. Then try an arithmetic formula:

*Category:* Walls

*Target Property:* TestNumber

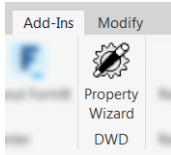
*Formula:* 6 \* 7

## Reference

This section provides reference material.

### Running PropertyWizard

PropertyWizard adds one button to a panel on the Add-Ins toolbar:

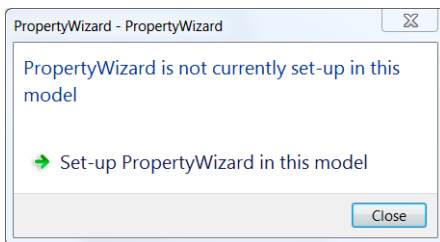


The button allows you to activate PropertyWizard and create formulas.

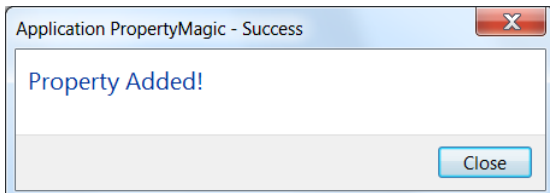
### Activating PropertyWizard in a project

PropertyWizard will do nothing until you activate it.

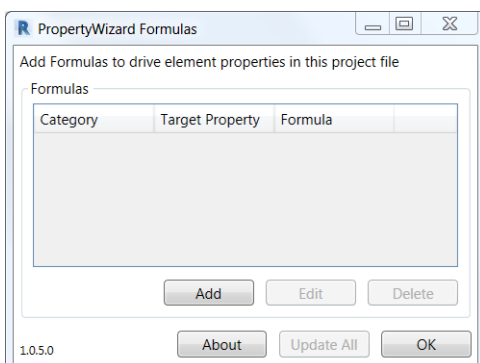
You activate it for each Revit .rvt file individually: just click the PropertyWizard button. In a new project, you will get this dialog:



Click 'Set-up PropertyWizard in this model' to set up PropertyWizard in the project file. This will add PropertyWizard's shared parameter to the model.

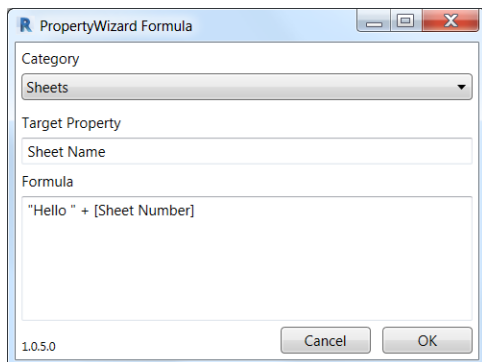


The button will also launch the PropertyWizard main window, which lists the formulas you have in the project:



From here you can add new formulas, and edit and delete your existing ones.

Both Add and Edit buttons open the Formula Edit window:



### Creating a formula

Each formula has three parts:

1. Select the Category – the formula will apply to all elements of this category.
2. Type in the name of the Target Property – the parameter you want to set.
3. Type in the Formula.

### Target Properties

This should be an Instance property, not a Type property, in this version.

In this version, Target Property can be any of these data types:

- Number and Integer
- Length
- Area
- Text
- Yes/No
- Angle

### Formulas

In the formula, you can type in:

- Operators and Functions, like +, sin() and if().
- Numbers and Integers. Numbers have a decimal point, Integers are whole numbers.
- Lengths (using the suffixes 'mm', 'm' or 'ft').
- Areas (using the suffix 'sqm').
- Text strings, delimited by double-quotes.
- The literal values 'true' and 'false' (which should be typed without the quotes).
- Names of source properties. The formula will use the value of the property.

### Source Properties

You can refer to:

- Properties of the target element.
- Nested properties of the target element (e.g. properties of a Door's Level or a Wall's Type).
- API properties of the target element or nested elements (i.e. properties from the Revit API).
- Project Information properties.

Properties of the target element are just referred to by their name as seen in the Revit Properties window, for example, 'Level' (which should be typed without the quotes). If the name includes spaces or special characters, then it should be enclosed with square brackets like this: '[Sheet Name]' (which again should be typed without the quotes).

Nested properties are referenced by 'dot notation' as used in Dynamo. So the name of a Door's Level can be referred to as 'Level.Name'. There is no limit to the depth of nesting, but all the properties except the last must be Elements or ElementIDs in this version.

API properties are the 'behind the scenes' properties exposed in the Revit API. For example, the 'Location' property of a Structural Pile. To find the correct names, I use a combination of the RevitLookup add-in (compiled versions available from Troy Gates' <http://revitcoaster.blogspot.com/> and Harry Mattison's <https://boostyourbim.wordpress.com/>), and the official Revit API Help (downloadable from Autodesk, or online at Gui Talarico's <http://www.revitapidocs.com/>). For more details on installing and using RevitLookup:

<http://www.davidwooddesign.co.uk/2018/08/19/how-to-install-revitlookup/>

<http://www.davidwooddesign.co.uk/2018/08/27/what-properties-can-you-use-in-propertywizard-formulas/>

Project Information properties can be referenced using dot notation starting with 'ProjectInfo', for example. 'ProjectInfo.[Project Name]'.

### Operators and Functions:

	Format	Notes
<b>Number operators</b>		
Multiply	a * b	
Divide	a / b	
Add	a + b	+ also provides string concatenation.
Subtract	a - b	
<a href="#">Exponentiation</a>	a ^ b	a to the power b.
<b>Number functions</b>		
Square Root	sqrt(a)	sqrt(4) == 2
<a href="#">Logarithm</a> base 10	log(a)	
<a href="#">Exponentiation</a> base 10	exp(a)	
Absolute value	abs(a)	
Sine	sin(a)	
Cosine	cos(a)	
Tangent	tan(a)	
Arcsine	asin(a)	
Arccosine	acos(a)	
Arctangent	atan(a)	
PI	pi()	
Round	round(a)	
Roundup	roundup(a)	
Rounddown	rounddown(a)	



### Logical functions

And	and(a, b...)	True only if both a and b are true.
Or	or(a, b...)	True if either a or b is true.
If	if(a, b, c)	If a is true, then b, else c.

### Flow control functions

Try	try(a, b)	Tries a. If a throws an error, returns b
-----	-----------	--

### Logical operators

And	a && b	True only if both a and b are true.
Or	a    b	True if either a or b is true.
Not	!a	True if a is false.
Equal	a == b	True if a is equal to b.
Not equal	a != b	True if a is not equal to b.
Greater than	a > b	True if a is greater than b.
Less than	a < b	True if a is less than b.
Greater than or equal to	a >= b	True if a is greater than, or equal to, b.
Less than or equal to	a <= b	True if a is less than, or equal to, b.

### String Functions

String concatenation	strcat(a, b,...)	strcat("bow", "tie") == "bowtie"
String length	strlen(a)	
Sub-string	substr(a, b, c)	substr("tree", 1, 2) == "re"
String find	instr(a, b)	instr("tree", "re") == 1 returns -1 if not found
String format	format(a, b...)	Implements .Net String.Format

### Units

Millimetres	a mm
Metres	a m
Feet	a ft
Square metres	a sqm

### Coordinates

Internal -> Shared	internaltoshared(a)	Converts a Revit database XYZ value into shared coordinates
Shared -> Internal	sharedtointernal(a)	

## Release Notes (What's New)

### Compatibility

Each new version of PropertyWizard is designed to work properly with Revit files that contain formulas from previous versions.

In addition, the current version is file-compatible with previous versions of PropertyWizard, meaning the PropertyWizard data stored in the Revit model file has not changed. Nevertheless, if you are sharing a file between users, I recommend that all users who are accessing the same file use the same version of PropertyWizard.

#### V1-0-5 New Features

- Bugfixes and stability improvements.

#### V1-0-4 New Features

- Feedback window clarified.
- About window updated.
- Bugfixes and stability improvements.

#### V1-0-3 New Features

- Improved handling of incompatible formulas.
- Improved UX for licensing.
- Improved UX for logging.
- Bugfixes and stability improvements.

#### V1-0-2 New Features:

- Improved reporting of formula errors.
- Bugfixes and stability improvements.

#### V1-0-1 New Features:

- Bugfixes and stability improvements.
- Preview version for Autodesk App Store.

#### V1-0-0 New Features:

- Preview version for Autodesk App Store.

### Beta versions

#### V0-20-0 New Features:

- Bugfixes and stability improvements.

#### V0-19-0 New Features:

- Improved reporting of formula errors.
- Improved handling of incompatible formulas.
- Improved UX for licensing.
- Improved UX for logging.
- Bugfixes and stability improvements.

V0-18-0 New Features:

- Bugfixes and stability improvements.

V0-17-0 New Features:

- Bugfixes and stability improvements.

V0-16-0 New Features:

- Views added to Category selector.
- Bugfixes and stability improvements.

V0-15-0 New Features:

- General Public Beta Release.
- Change to Azure from Sentry logging.
- Bugfixes and stability improvements.
- Fit and finish improvements.

V0-14-0 New Features:

- Improved error handling.
- Feedback form added.
- Currency data type added.
- Bugfixes and stability improvements.
- Fit and finish improvements.

V0-13-0 New Features:

- 'try' function added.
- 'Continue processing this formula, and ignore errors.' option added to result dialog.
- Bugfixes and stability improvements.

V0-12-0 New Features:

- First Public Beta release.
- Bugfixes and stability improvements.
- Fit and finish improvements.

V0-11-0 New Features:

- Bugfixes and stability improvements.
- Fit and finish improvements.

V0-10-0 New Features:

- Renamed 'PropertyWizard'.
- Preparation for public Beta.
- Activity and Errors are logged to Sentry.io for analysis and bugfixing.
- Bugfixes and stability improvements.
- Fit and finish improvements.

*Autodesk and Revit are registered trademarks or trademarks of Autodesk, Inc., and/or its subsidiaries and/or affiliates in the USA and/or other countries.*