



**Native**soft 

**Native**Geometry

22



## **1 Getting started with NativeGeometry**

1.1 Welcome to NativeGeometry

4

1.2 Make your application using NativeGeometry

6

## **2 Using NativeGeometry**

2.1 Set the geometry of an object

8



## Welcome to NativeGeometry

NativeGeometry is a system which assists application developers working with Runtime Revolution who wish to design cross-platform and localizable graphical user interfaces (GUI) at the speed of the thought.

Using NativeGeometry, you will be able to design without any effort complex GUI, NativeGeometry has been designed to be the less intrusive and to requires only few lines of code inside your application to be fully operational.

NativeGeometry is in version 2.x because the first version was included inside NativeSpeak Create 1.x, NativeGeometry 2.x is a complete rewrite and is using all the latests improvement offered by Runtime Revolution. If you had already used NativeSpeak 1.x, forget everything about it and check it out this manual to learn how to use efficiently NativeGeometry!

### Setting up NativeGeometry

License Manager - NativeGeometry

NativeSoft NativeGeometry - License Manager

Enter your serial number

To fully use NativeGeometry, you must enter a valid license Key.

If you do not have a valid license key, click here to purchase one : [NativeSoft Store](#).

To try freely NativeGeometry, click "Cancel" to use the "Starter Kit" edition, that is limited to 3 objects by stack.

User name:

Company:

License Key:

Cancel Previous Register

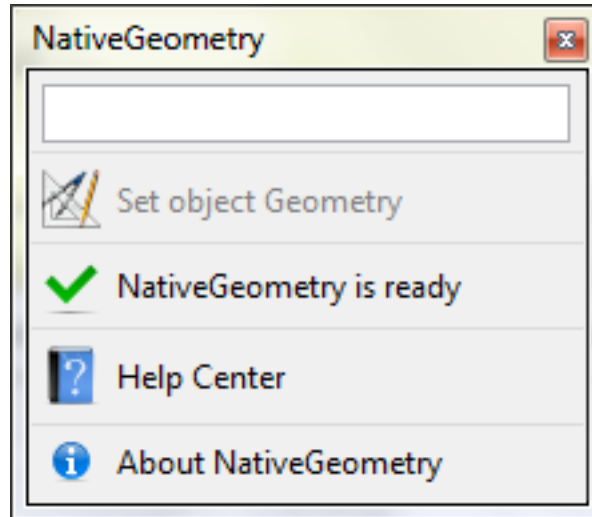
Once the installation of NativeGeometry is finished, simply re-start Revolution.

The NativeGeometry License Manager will open: accept the license agreement, and enter your user name(1), your company(2) and your license key (3). If you do not have a license key, you can try

NativeGeometry by clicking "Cancel" (5) with the starter kit edition that is limited to 5 objects, or buy a license on the [NativeSoft Store](#).

Once finished, click "Register" (4).

## First contact with NativeGeometry

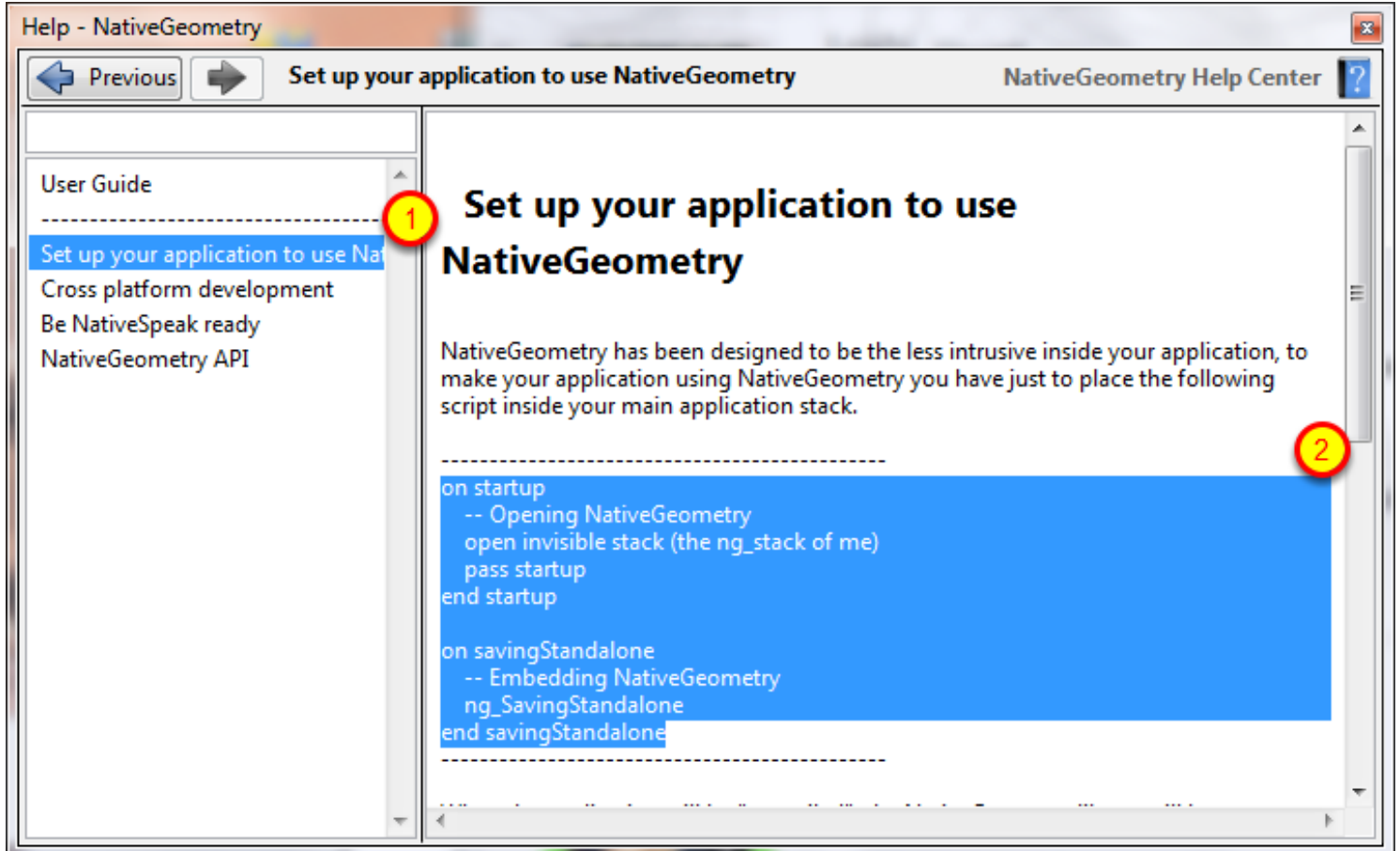


Once Revolution is re-started after the installation, you will see this palette. It is the main NativeGeometry palette.

## Make your application using NativeGeometry

NativeGeometry has been designed to be effort-less for the developer, to make an application using NativeGeometry, simply follow those steps.

**Place inside your main application stack the following code**



Open the help NativeGeometry help center, and click (1) "Set up your application to use NativeGeometry", then copy the script in (2) inside your main application stack.

And that's all! When you will build your application, NativeGeometry will be embeded inside your application and at the startup will be loaded.

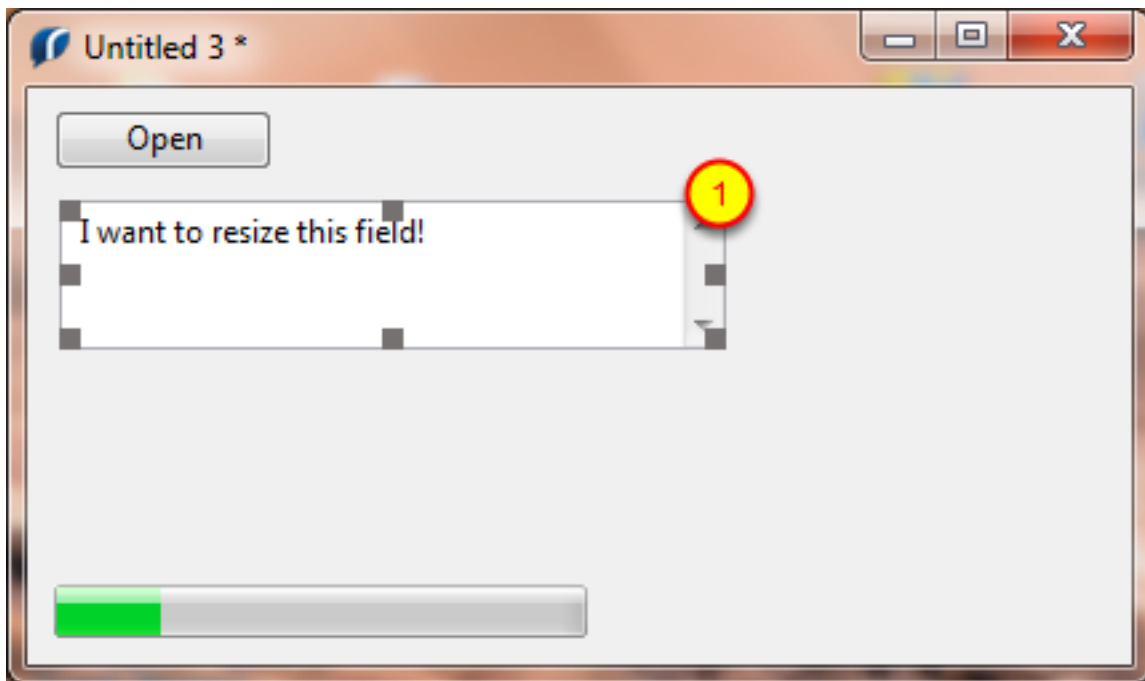
There is just one main thing you have to ever do in order to have NativeGeometry working:

**EVER PASS "resizestack" AND "preOpenCard" HANDLERS!**



## Set the geometry of an object

### A sample stack

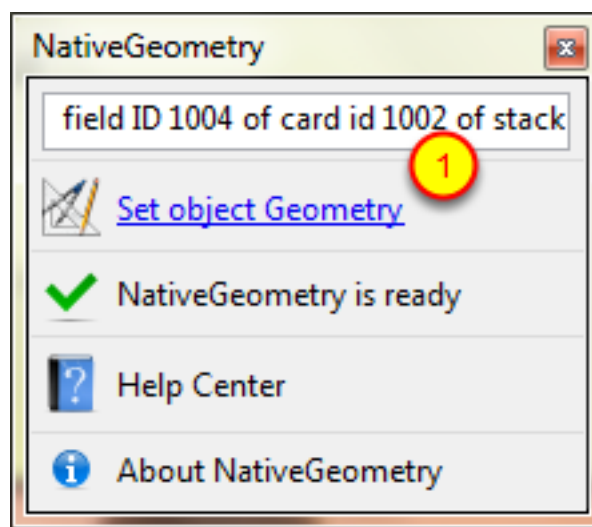


To illustrate this example, we have created a simple stack with a button, a field and a scrollbar.

We want to have the field to be placed between the button and the scrollbar, and to have the field to fill the width of the card.

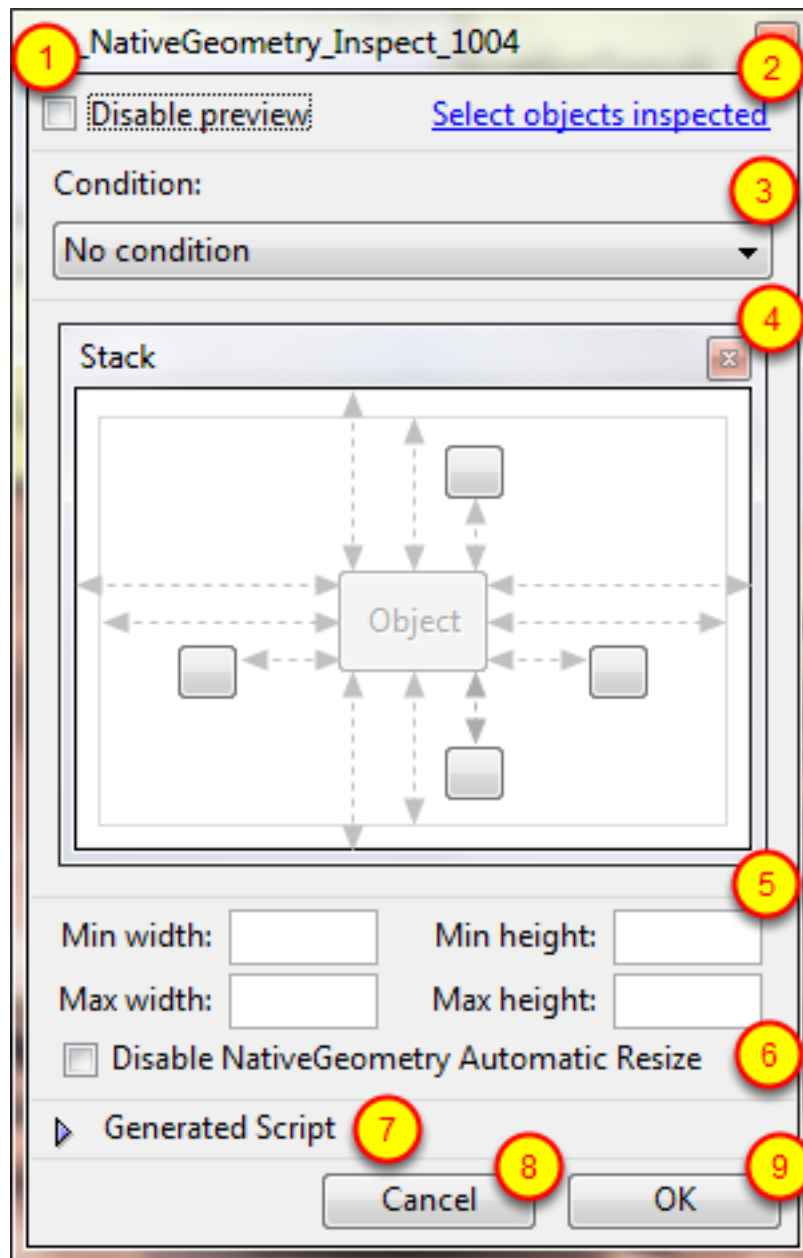
The first step is to select the object that you want to set the geometry of.

### Open the geometry inspector palette



Click "Set object Geometry" to open the geometry inspector palette.

## The geometry inspector palette



The geometry inspector is empty, it means that the object does not have any geometry relations set.

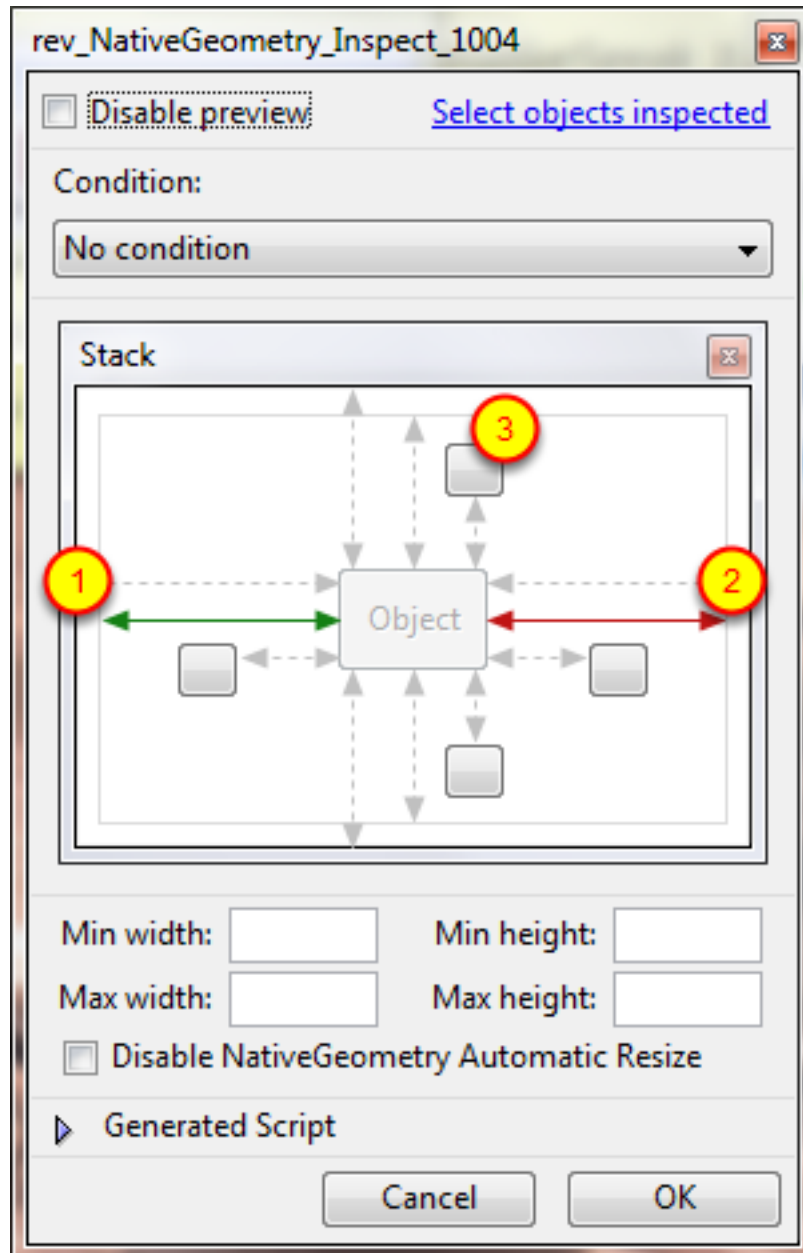
Some explanation about this palette:

- 1 "Disable preview", disable the preview of geometry modifications. If not activated the object geometry will be directly applied. (If you press "Cancel" then the object revert back to its original position)
- 2 "Select objects inspected", selects the objects edited by this geometry inspector.
- 3 "Condition", NativeGeometry handle single level geometry condition, see the associated chapter to learn how to use them.
- 4 "The geometry operation", define here what you want the object to do.
- 5 "Min/Max width/height", set the minimal or the maximal width/height that the object is allowed to

have.

- 6 "Disable NativeGeometry Automatic Resize", activate this button to disable the NativeGeometry feature that will make the object to fit to its content.
- 7 "Generated Script", click here to see the script that the NativeGeometry palette has generated.
- 8 "Cancel", cancel any updates and close the inspector.
- 9 "OK", apply any updates and close the inspector.

### Set the left, the right and the top of the object



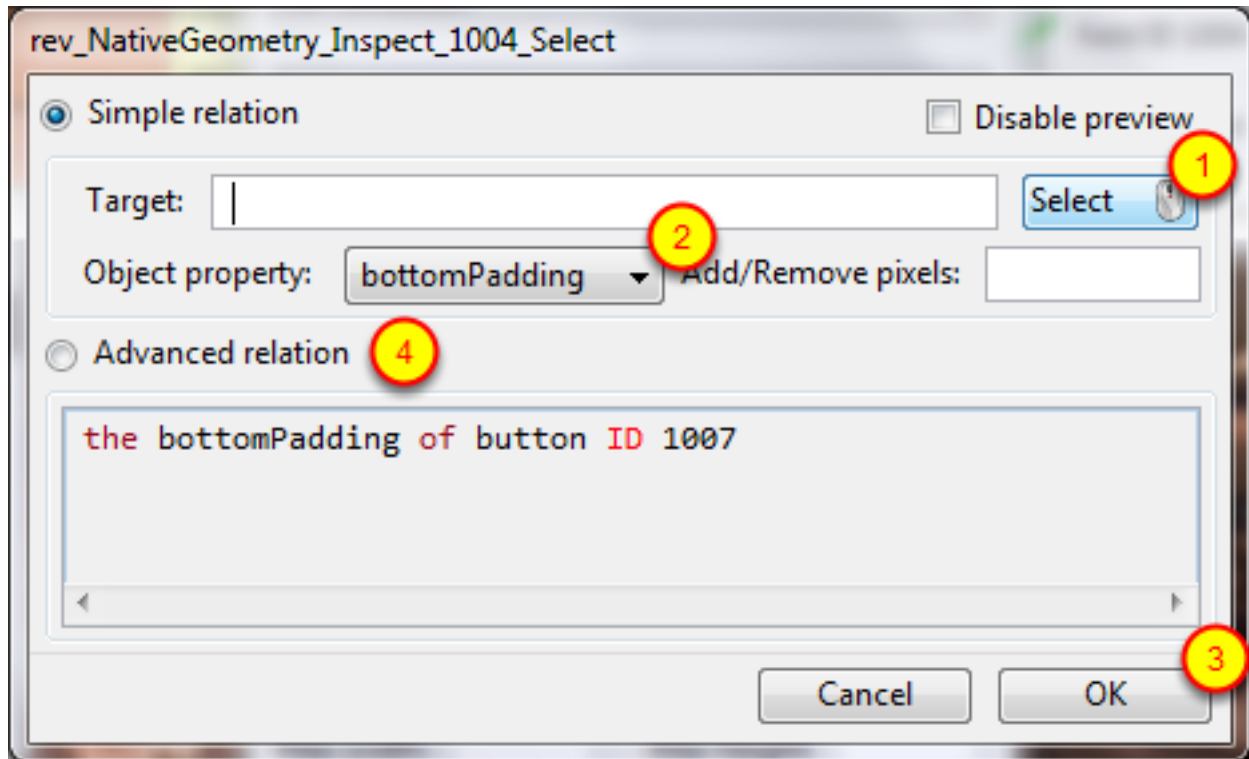
Information: The padding is a virtual property that is platform dependant and means the "left/top/right/bottom" of an object + a space defined by the platform.

To set the object to the left padding of the card, click "1".

To resize the object to the right padding of the card, double click "2".

To set the object to the bottom padding of another object, click "3".

## The object dependency and custom relation palette

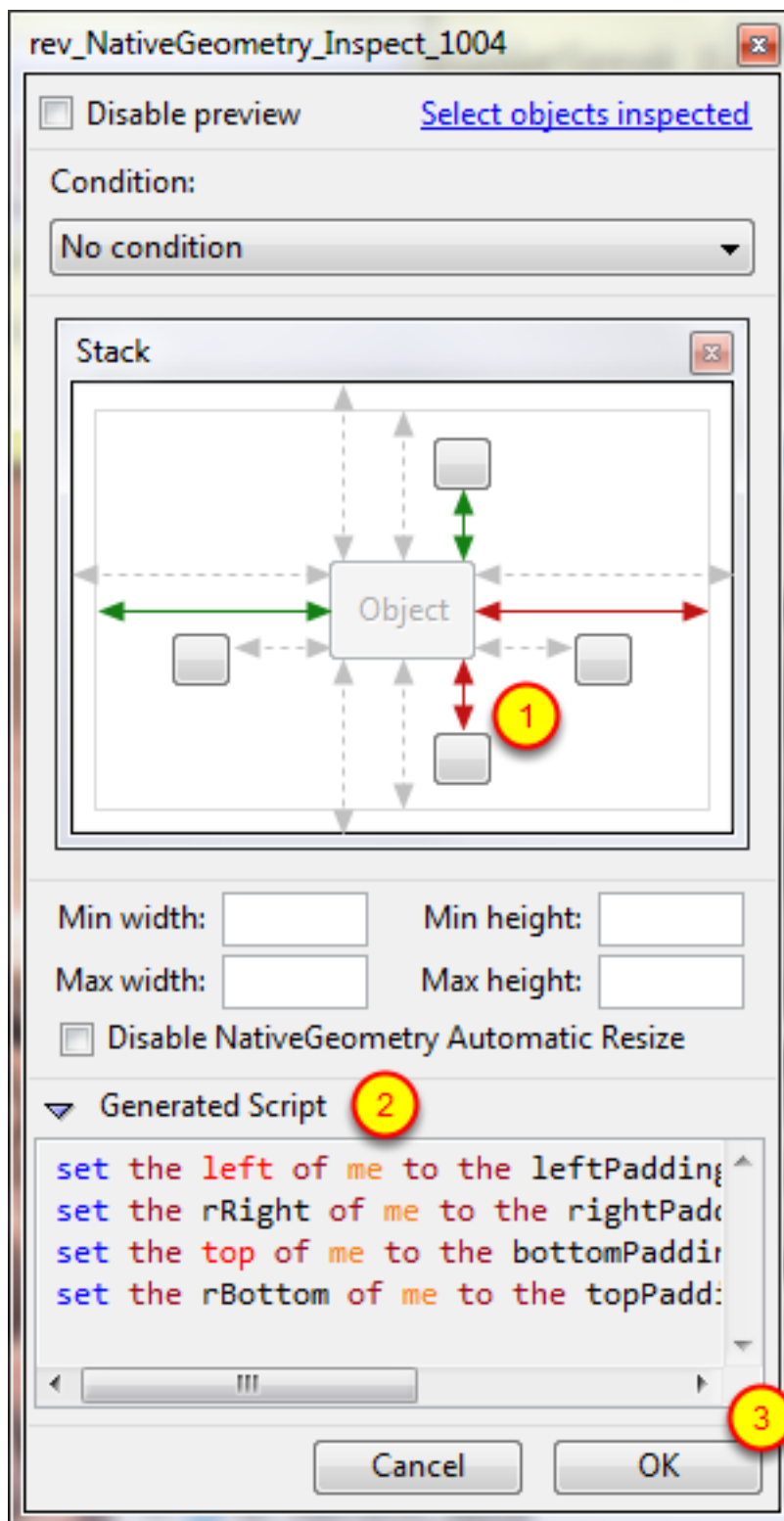


To select the object that you want to depend on, click (1) "Select", then click the object that you want to depend on.

Select the object property that you want to use (2) and click (3) "OK" to apply.

Note: You can also set an advanced relation by clicking (4) "Advanced relation"

## Finalizing the object geometry



Repeat the previous step in order to resize the bottom of the field to the top padding of the scrollbar (1).

Click (2) "Generated Script" to see the script generated by the inspector.

Click (3) "OK" to apply and close the geometry inspector.

The geometry of the object has been set, save your stack in order to save its geometry.  
(NativeGeometry geometry properties are located inside custom properties of the stack/card)