

- Developer
- / Documentation
- / Unreal Engine ▾
- / Unreal Engine 5.4 Documentation
- / Creating User Interfaces
- / Basics
- / Building Your UI
- / UMG Slots

UMG Slots

An overview of how to use UMG slots.

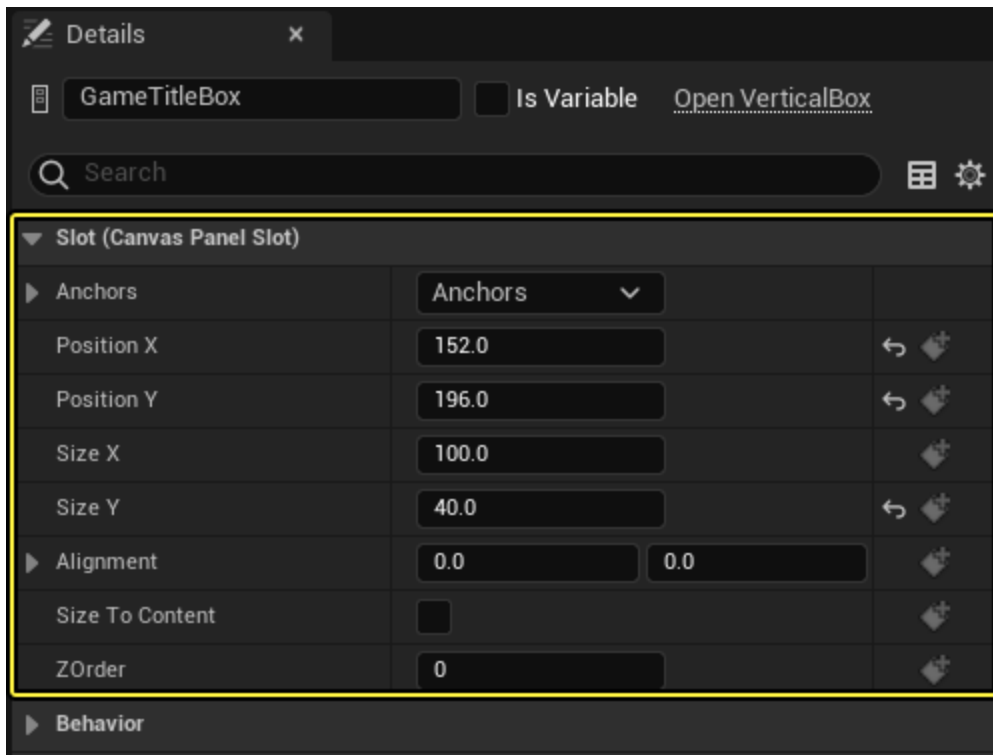


Slots are the invisible glue that bind Widgets together. In Slate they are a lot more explicit in that you must first create a Slot then choose what control to place inside of it. In UMG, however, you have Panel Widgets that automatically use the right kind of Slot when Child Widgets are added to them.

Moreover, each Slot is different. For example, if you were to place a control on a Grid, you would expect to be able to set things like Row and Column. But these properties have no business being on a Widget that was placed on a Canvas. That is where Slots come in. A **Canvas Slot** understands how to layout content absolutely and through Anchors, while a **Grid Slot** only understands Rows and Columns.

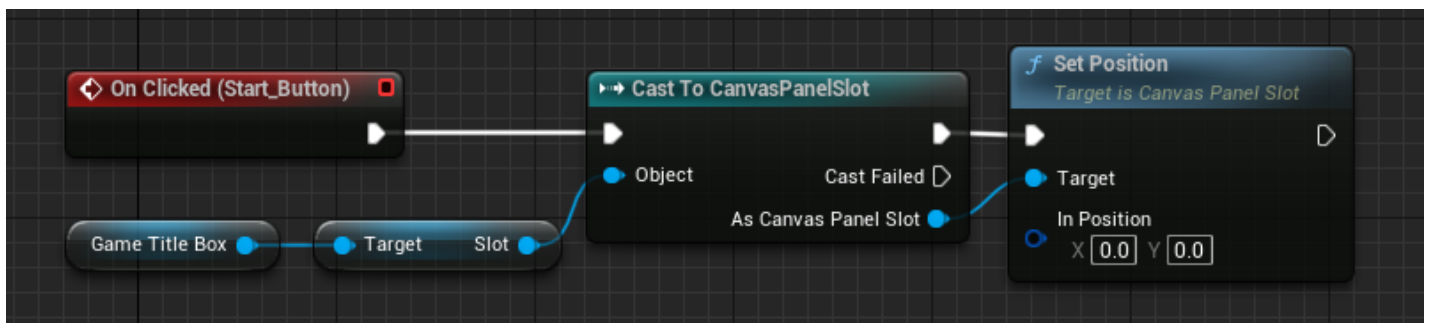
Accessing Slots

By convention, all Slot-related properties appear under the **Slot** category in the **Details** panel. You will also notice that the type of Slot being used by your Widget is displayed in parentheses.



Setting Layout Properties

At runtime, to modify properties under Layout, you can access the Slot member of the Widget in Blueprint or C++ and then **Cast** it to the correct **Slot Type**. Once doing so, you will then be able to modify the properties, an example of which is indicted below.



Above, a **Vertical Box** entitled **GameTitleBox** has been placed on a **CanvasPanel**. By getting the Slot associated with the Vertical Box and **Casting** to the **CanvasPanelSlot** type, we are then able to set the position of the box when our "StartButton" is clicked.



Currently in Blueprints, only SETTER nodes are exposed. If you need to GET properties from the Layout, you may want to create a Variable to store your property and upon **Event Construct**, and **SET** your Layout property via your Variable so that you have reference to and can access it later.