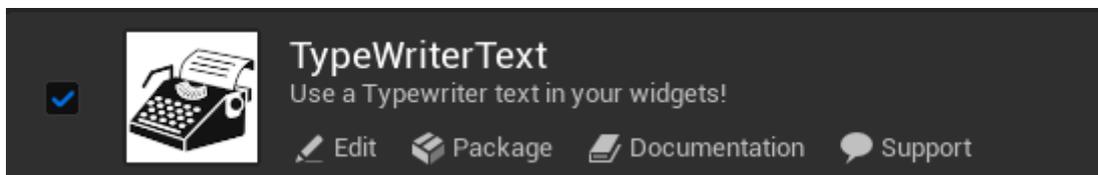


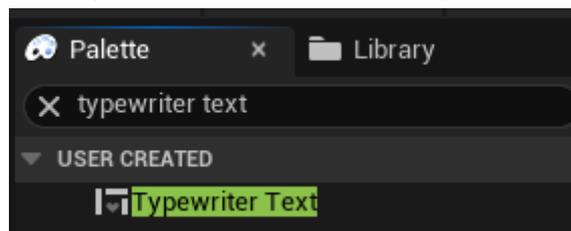
Documentation

- **Widget**..... 1
 - ❖ TypewriterWidget..... 1
 - ❖ TypewriterText..... 1
- **Functions**..... 2
 - ❖ AnimateText..... 2
 - ❖ SetText..... 2
- **Customization**..... 2
 - ❖ TextStyle..... 2
 - ❖ Duration..... 2
 - ❖ UserText..... 3
- **Delegates**..... 3
 - ❖ OnCharacter..... 3
 - ❖ OnCompleted..... 3
- **How to Use**
 - ❖ Example

To enable the Textwriter Text plugin first you have to go to the Epic Games launcher, UE5, then download the plugin. After that open your project, go to plugins, find “Typewriter Text”, enable it then restart. Now your plugin should be enabled and ready to use.

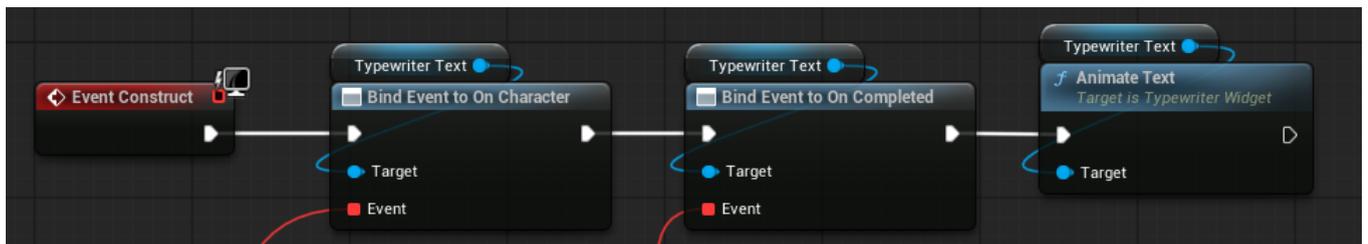


To use it go to your widget where you want to have the text, search for “Typewriter Text” NOT the “Typewriter Widget” then place it wherever you want.



Mark it as a variable so you can access its methods. I also recommend binding the Delegates OnBeginPlay so whenever you use the animation method they’re correctly set.

Now that the Typewriter Text is added to your widget, use it however you want and call the “AnimateText” to animate your text and display it!



- Widget
 - ❖ **TypewriterWidget**

The `TypewriterWidget` class is the base class, implemented in C++, for the custom Typewriter Text system. It inherits from `UUserWidget`.

```
UTypewriterWidget : public UUserWidget
```

- ❖ TypewriterText

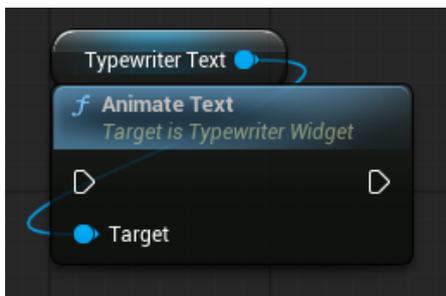
The `TypewriterText` class is a Blueprint-derived widget based on `TypewriterWidget`. It allows for deeper customization of the text behavior and appearance.

This is the class you should add to your widgets to use the typewriter effect—avoid adding the `TypewriterWidget` base class directly.



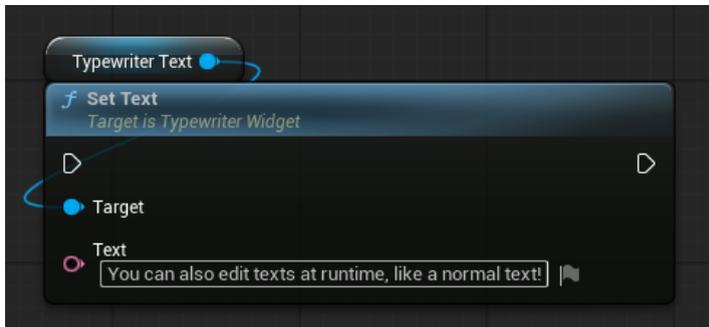
- Functions
 - ❖ AnimateText

Displays each character of the assigned `FText` one by one, until the entire text is shown. Works with both the widget's default text or a value set via `SetText()`.



❖ SetText

Changes the widget's `FText` at runtime. Can also be used to set formatted text.



● Customization

❖ TextStyle

Modifies the text style without the need to open and edit the `TypewriterText` widget in the editor.

```
FTextBlockStyle TextStyle;
```

❖ Duration

Specifies the total duration of the text animation. The per-character display time is calculated based on this total duration.

```
float Duration = 1.0f;
```

❖ UserText

The default text the `TypewriterText` widget will display.

```
FText UserText;
```

● Delegates

❖ OnCharacter

Triggered each time a character is displayed.

```
OnCharacter.Broadcast();
```

❖ OnCompleted

Triggered when the animation finishes.

```
OnCompleted.Broadcast();
```