

InteractiveChartsToolkit 2

1. Introduction

1.1 Overview

Interactive Charts Toolkit 2 is an Unreal Engine 5 plugin for creating dynamic charts and graphs using UMG widgets.

The plugin provides a collection of configurable chart widgets that can be used directly in UMG and controlled through Blueprint.

The currently supported chart types are:

- Bar Chart
- Stacked Bar Chart
- Gantt Chart
- Gauge Chart
- Barometer Chart
- Step Line Chart
- Treemap Chart
- Circle Relationship Chart

Each chart is designed to be configurable through its widget properties. Data, appearance, layout, animation, and other chart-specific settings can be configured through the widget itself.

1.2 Design Philosophy

The chart system is designed around a property-driven workflow.

Most chart configuration is performed directly through the chart widget's exposed properties. The chart does not require a series of function calls to configure every individual element.

For example, a chart can generally be configured by:

1. Adding the chart widget to a UMG Widget Blueprint.
2. Setting its data.
3. Configuring its visual and layout properties.
4. Displaying the widget.

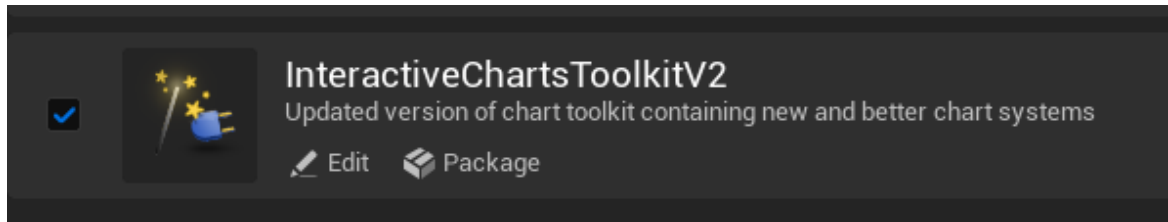
When data or configuration changes at runtime, the appropriate refresh/update function can be called to regenerate the chart.

This allows the same chart widget to be used with both static and dynamic data.

2. How to install

Steps:

1. Install the plugin from the Fab Marketplace.
2. Open your Unreal Engine project.
3. Enable the plugin if prompted.



4. Restart Unreal Engine.

To access the included examples, enable **Show Plugin Content** in the Content Browser and open the demo maps.

3. Bar Chart

3.1 Overview

The **Bar Chart** is used to compare numerical values across multiple categories. Each category is represented by a bar, where the size of the bar corresponds to its associated value.

The chart supports both **vertical and horizontal orientations** and provides options for controlling the appearance, layout, labels, grid, animation, and interaction of the chart.

The Bar Chart is suitable for displaying data where individual values need to be compared against one another.

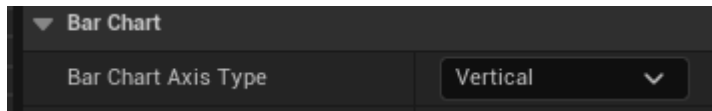
3.2 Chart Properties

The Bar Chart is configured primarily through the properties exposed by the widget. These properties control the chart data, orientation, appearance, layout, labels, grid, animation, and interaction.

- **Bar Chart Axis Type**

Controls the orientation of the Bar Chart.

- **Vertical:** Bars are displayed vertically.
- **Horizontal:** Bars are displayed horizontally.



- **Bar Chart Entries**

Contains all the data used to generate the bars and groups.

Each **parent entry represents a group**, and the **Value Entries inside it represent the bars**.

- **Single Bar**

Add one Value Entry to each group.

Group A → Entry A

Group B → Entry B

Group C → Entry C

This creates one bar for each group.

- **Grouped Bars**

Add multiple Value Entries to a group.

Group A → Entry A1, Entry A2, Entry A3

Group B → Entry B1, Entry B2, Entry B3

This creates multiple bars within each group.

Each group and entry can also have a **name and description**.

Each Value Entry contains:

- **Value:** Represents the numerical value of the bar. The chart uses this value to determine the bar's visual size relative to the other bars.
- **Colors:** Defines the bar color. Multiple colors can be added to create a gradient.

Each bar can have its own color. To use the same color for all bars in a group, assign the same color to each entry.

Note: Negative values are not supported. Do not use negative values in the **Value** property.



- **Appearance**

These properties control the visual appearance of the bars in the chart.

○ Corner Radius

- Controls the corner radius of the bars, allowing the bars to have rounded or sharp corners.

Corner Radius	0.0
---------------	-----

○ Bar Outline

Controls the outline displayed around the bars.

- **Show Outline:** Enables or disables the bar outline.
- **Outline Thickness:** Controls the thickness of the outline.
- **Outline Color:** Controls the color of the outline.

Show Bar Outline	☒
Outline Thickness	3.0
▶ Outline Color	

○ Bar Width

- Controls the width of each bar.

Bar Width	60.0
-----------	------

○ Bar Spacing

- Controls the spacing between bars within the same group.

Bar Spacing	20.0
-------------	------

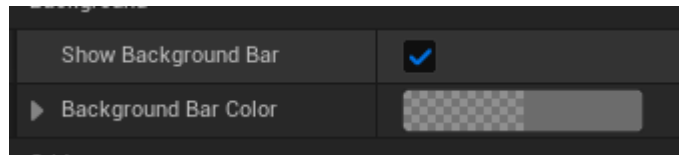
○ Group Spacing

- Controls the spacing between separate groups.

Group Spacing	100.0
---------------	-------

○ Show Background Bar and Background Bar Color

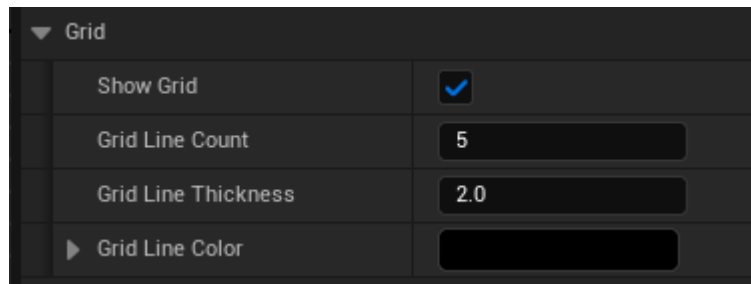
- Controls whether a background bar is displayed behind each bar and sets its color when enabled.



• Grid Line Properties

Controls the generation and appearance of grid lines in the Bar Chart, including:

- Whether grid lines are displayed.
- Number of grid lines.
- Grid line thickness.
- Grid line color.

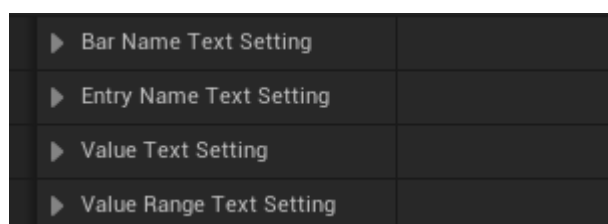


• Text Properties

These properties control the text displayed on the Bar Chart, including **value range**, **bar name**, **entry name**, and **value labels**.

Each text setting provides options to control:

- **Visibility:** Shows or hides the text.
- **Font Settings:** Controls the font and its appearance.
- **Text Color:** Controls the color of the text.
- **Padding:** Controls the spacing around the text.
- **Text Transform:** Controls the text position and transformation.



▼ Bar Name Text Setting	
Visible	☒
Text	
▼ Font	
Font Family	F_GoogleSansFlex ↶ 📁
Typeface	Default ▼
Size	12
Letter Spacing	0
Skew Amount	0.0
Font Material	None None ↶ 📁
▶ Outline Settings	
▶ Color	
▶ Transform	

- **Value Range Text Fractions**

Controls the number of decimal places displayed for the **value range labels**.

It provides separate **minimum and maximum fraction limits**, allowing the displayed value range numbers to show the desired number of decimal places when needed.

Min Value Range Text Fractions	0
Max Value Range Text Fractions	2

- **Animation Duration**

Controls the duration of the bar animation when the chart is generated or refreshed. The animation is played only when **Play Animation** is enabled.

Bar Animation Duration	0.75
------------------------	---------------------------

- **Show Tooltip**

Controls whether a tooltip is displayed when hovering over a bar.

Show Tooltip	☒
--------------	-------------------------------------

3.3 Chart Functions

The Bar Chart provides two functions for generating and updating the chart: **Refresh Chart** and **Update Chart**.

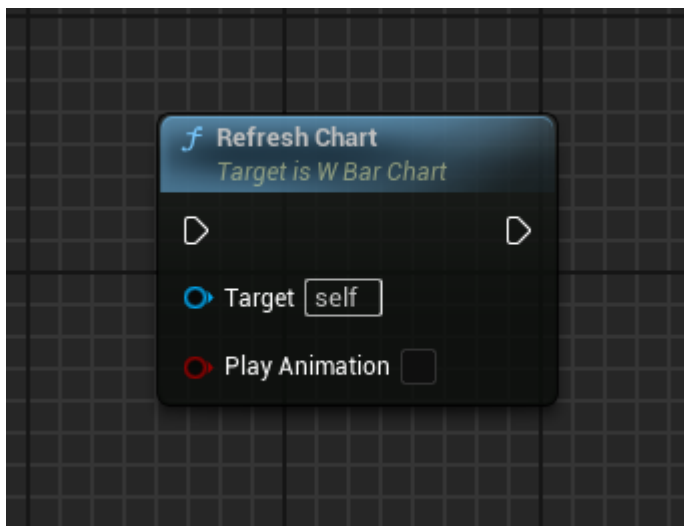
- **Refresh Chart**

Refreshes the chart using its current data and chart properties.

Use this function when the chart properties or existing data have been changed and the chart needs to be regenerated.

It includes a **Play Animation** parameter:

- **True:** Plays the bar animation when generating the chart.
- **False:** Generates the chart without animation.



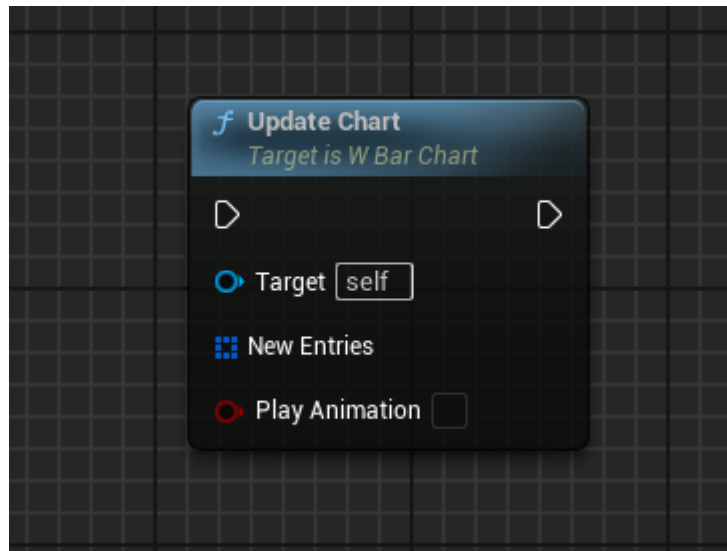
Note: Changes made to the chart properties are applied when the chart is refreshed.

- **Update Chart**

Updates the chart with new bar entry data.

Pass the new **Bar Chart Entries** to generate the chart using the updated data. It also includes a **Play Animation** parameter to control whether the bar animation should play.

- **True:** Plays the bar animation.
- **False:** Generates the chart without animation.



- **Runtime Usage**

The chart is generated at runtime only when **Refresh Chart** or **Update Chart** is called.

For example, to generate the chart when using it in a Widget Blueprint, call one of these functions from the **WBP_BarChart Construct** event.

Note: Simply changing the widget properties does not automatically regenerate the chart. Call **Refresh Chart** after changing properties, or **Update Chart** when providing new chart data.

4. Stacked Bar Chart

4.1 Overview

The **Stacked Bar Chart** is used to compare the total values of multiple categories while also showing how each category is divided into different entries.

Each group is represented by a single bar, and the entries inside the group are **stacked together** within that bar. The size of each section is based on its value, while the total size of the bar is based on the combined value of all entries in the group.

The chart supports both **vertical and horizontal orientations** and provides options for controlling the appearance, layout, labels, grid, animation, and interaction.

The Stacked Bar Chart is suitable for displaying **total values with their individual components**, making it easy to compare both the overall value and the contribution of each entry.

Note: Unlike the regular Bar Chart, the Stacked Bar Chart does not support gradient colors. Each stacked entry uses its assigned color.

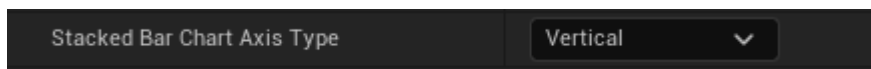
4.2 Chart Properties

The Stacked Bar Chart is mainly configured through the properties exposed by the widget. These properties control the chart data, orientation, appearance, layout, labels, grid, animation, and interaction.

- **Stacked Bar Chart Axis Type**

Controls the orientation of the Stacked Bar Chart.

- **Vertical:** Displays the stacked bars vertically.
- **Horizontal:** Displays the stacked bars horizontally.



- **Stacked Bar Chart Entries**

Contains all the data required to generate the Stacked Bar Chart, including **groups, entries, values, and colors**.

Each **group represents one stacked bar** and contains multiple entries that are stacked within that bar.

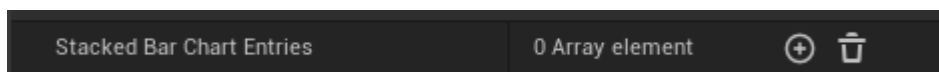
Each entry contains:

- **Value:** Represents the value of that section of the stacked bar. The size of the section is based on its value.
- **Color:** Defines the color of the section.

The **total size of the stacked bar** is determined by the sum of all entry values within the group.

A group can contain any number of entries, including a single entry.

For a consistent chart, groups should generally contain the required number of entries according to the data being represented.



- **Bar Width and spacing**

Controls the **width of each stacked bar** and the **spacing between the stacked bars**.

▼ Appearance	
Bar Width	40.0
Bar Spacing	20.0

- **Grid Line Properties**

Controls the generation and appearance of grid lines in the Bar Chart, including:

- Whether grid lines are displayed.
- Number of grid lines.
- Grid line thickness.
- Grid line color.

▼ Grid	
Show Grid	☒
Grid Line Count	5
Grid Line Thickness	2.0
▶ Grid Line Color	

- **Text Properties**

These properties control the text displayed on the Stacked Bar Chart, including **value range, bar name and value labels**.

Each text setting provides options to control:

- **Visibility:** Shows or hides the text.
- **Font Settings:** Controls the font and its appearance.
- **Text Color:** Controls the color of the text.
- **Text Transform:** Controls the text position and transformation.

▶ Bar Name Text Setting	
▶ Value Text Setting	
▶ Value Range Text Setting	

▼ Bar Name Text Setting

Visible

☒

Text

▼ Font

Font Family

F_GoogleSansFlex

↶ ↷

Typeface

Default

▼

Size

12

Letter Spacing

0

Skew Amount

0.0

Font Material

None

None

↶ ↷

▶ Outline Settings

▶ Color

▶ Transform

- **Value Range Text Fractions**

Controls the number of decimal places displayed for the **value range labels**.

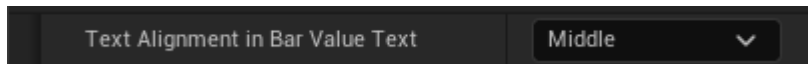
It provides separate **minimum and maximum fraction limits**, allowing the displayed value range numbers to show the desired number of decimal places when needed.

Min Value Range Text Fractions	0
Max Value Range Text Fractions	2

- **Text Alignment in Bar Value Text**

Controls the vertical alignment of the value text within each stacked bar segment.

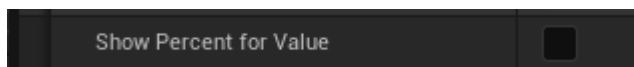
- **Top:** Aligns the value text toward the top of the segment.
- **Middle:** Aligns the value text to the center of the segment.
- **Bottom:** Aligns the value text toward the bottom of the segment.



- **Show Value as Percent**

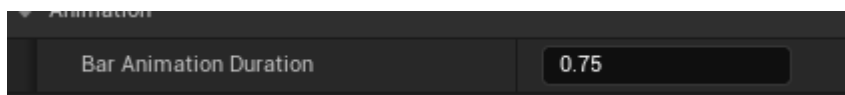
When enabled, the **Value Text** for each entry displays its percentage of the total value of the group instead of its actual value.

The percentage is calculated based on how much the entry contributes to the total stacked bar.



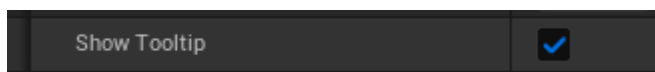
- **Animation Duration**

Controls the duration of the bar animation when the chart is generated or refreshed. The animation is played only when **Play Animation** is enabled.



- **Show Tooltip**

Controls whether a tooltip is displayed when hovering over a bar.



4.3 Chart Functions

The Stacked Bar Chart provides two functions for generating and updating the chart: **Refresh Chart** and **Update Chart**.

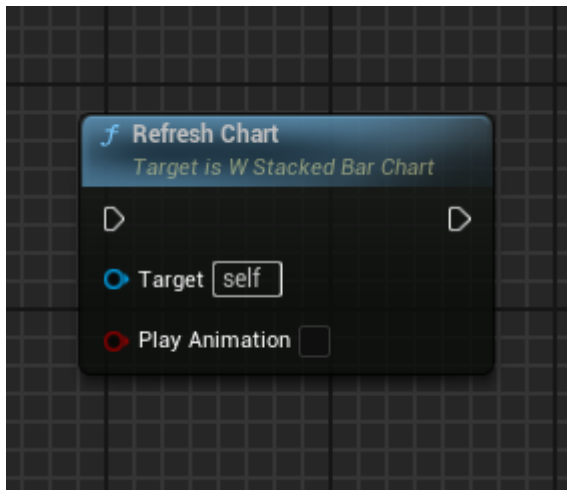
- **Refresh Chart**

Refreshes the chart using its current data and chart properties.

Use this function when the chart properties or existing data have been changed and the chart needs to be regenerated.

It includes a **Play Animation** parameter:

- **True:** Plays the bar animation when generating the chart.
- **False:** Generates the chart without animation.



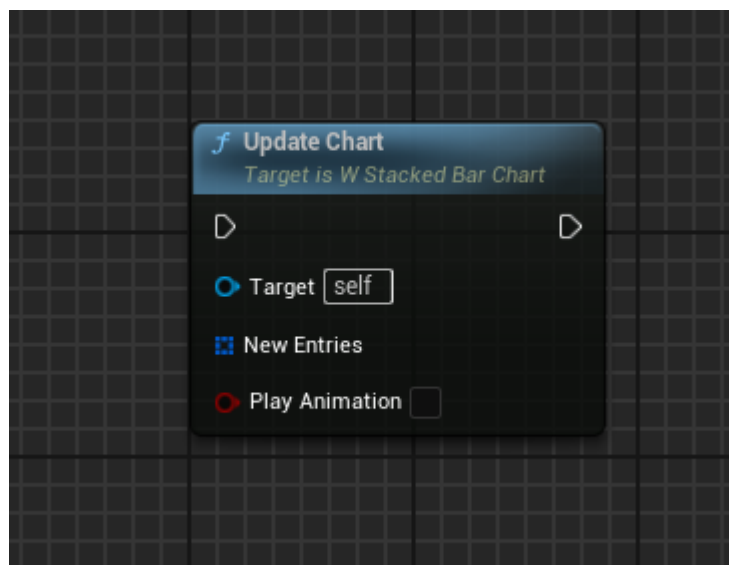
Note: Changes made to the chart properties are applied when the chart is refreshed.

- **Update Chart**

Updates the chart with new bar entry data.

Pass the new **Stacked Bar Chart Entries** to generate the chart using the updated data. It also includes a **Play Animation** parameter to control whether the bar animation should play.

- **True:** Plays the bar animation.
- **False:** Generates the chart without animation.



4.4 Runtime Usage

The chart is generated at runtime only when **Refresh Chart** or **Update Chart** is called.

For example, to generate the chart when using it in a Widget Blueprint, call one of these functions from the **WBP_StackedBarChart Construct** event.

Note: Simply changing the widget properties does not automatically regenerate the chart. Call **Refresh Chart** after changing properties or **Update Chart** when providing new chart data.

5. Gauge Chart

5.1 Overview

The **Gauge Chart** is used to display a value within a defined range. It provides a visual representation of how a value compares to its minimum and maximum limits.

The Gauge Chart supports two types:

- **Single Gauge:** Displays a single gauge value.
- **Multiple Gauge:** Displays multiple gauge values within the same chart.

Both types use similar basic concepts, such as minimum value, maximum value, radius, and thickness, but they have **different data structures, properties, and generation behaviour**.

5.2 Single Gauge

The **Single Gauge** displays one value on a gauge. It is suitable for showing a single metric such as progress, performance, temperature, or completion percentage.

- **Chart Properties**

- **Gauge Chart Type**

Controls the shape or portion of the gauge that is displayed.

- **Half:** Displays a half-circle gauge.
 - **Partial:** Displays a partial-circle gauge.
 - **Three Quarter:** Displays a three-quarter-circle gauge.
 - **Full:** Displays a complete circular gauge.

Type

Chart Type

Partial

Value

Represents the current value displayed by the gauge.

The gauge uses this value together with the minimum and maximum values to determine how much of the gauge should be filled.

Value

33.0

Minimum Value

Defines the minimum value of the gauge range.

Min Value

0.0

Maximum Value

Defines the maximum value of the gauge range.

Max Value

100.0

Radius

Controls the overall size of the gauge by defining its radius.

Radius

100.0

Thickness

Controls the thickness of the gauge arc.

Thickness

35.0

Create Rounded Corners

Controls whether the ends of the gauge arc have rounded corners.

Create Rounded Corners

☒

Border Properties

Controls the border displayed around the gauges.

- **Create Border Around Progress:** Controls whether a border is displayed around the gauge progress.
- **Border Thickness:** Controls the thickness of the border.

Create Borders Around Progress	☐
Border Thickness	1.0

○ Color Properties

Controls the colors used in different parts of the gauge.

- **Background Color:** Controls the background color of the gauge.
- **Progress Colors:** Defines the colors used for the gauge progress.
Multiple colors can be added to create a gradient from the start to the end of the gauge.
- **Border Colors:** Defines the color of the border when the border is enabled.

▶ Background Color	<input type="color"/>
▶ Progress Colors	3 Array elements + ✕
▶ Border Color	<input type="color"/>

○ Text Properties

Controls the text displayed on the gauge, including the **Gauge Name** and **Value Label**.

Both text settings provide options to customize the text appearance and position, including:

- **Text:** Defines the text to display. For the Gauge Name, this is entered manually. The Value Label uses the gauge's **Value**.
- **Font Settings:** Controls the font and font appearance.

- **Text Color:** Controls the color of the text.
- **Text Transform:** Controls the position and transformation of the text.

○ Name Text

Controls the appearance of the gauge name. The **Text** property defines the name displayed on the gauge.

▼ Name Text Setting	
Visible	☒
Text	Gauge Chart Single 
▶ Font	
▶ Color	
▶ Transform	

○ Value Text

Controls the appearance of the gauge value label. The displayed value is taken from the gauge's **Value** property.

▼ Value Text Setting	
Visible	☒
Text	
▶ Font	
▶ Color	
▶ Transform	

○ Value Range Text Fractions

Controls the number of decimal places displayed for the **value range labels**.

It provides separate **minimum and maximum fraction limits**, allowing the displayed value range numbers to show the desired number of decimal places when needed.

Min Value Range Text Fractions	0
Max Value Range Text Fractions	2

Value Text Prefix and Postfix

Controls the text displayed before or after the gauge value.

Prefix: Adds custom text before the value.
Example: \$75

Postfix: Adds custom text after the value.
Example: 75% or 75°C

Value Text Prefix		
Value Text Postfix		

Chart Functions

The Gauge Chart provides one optional function that can be overridden to customize the value text displayed on the gauge.

Get Gauge Value Label Text

This function can be overridden to customize or format the **Gauge Value Label**.

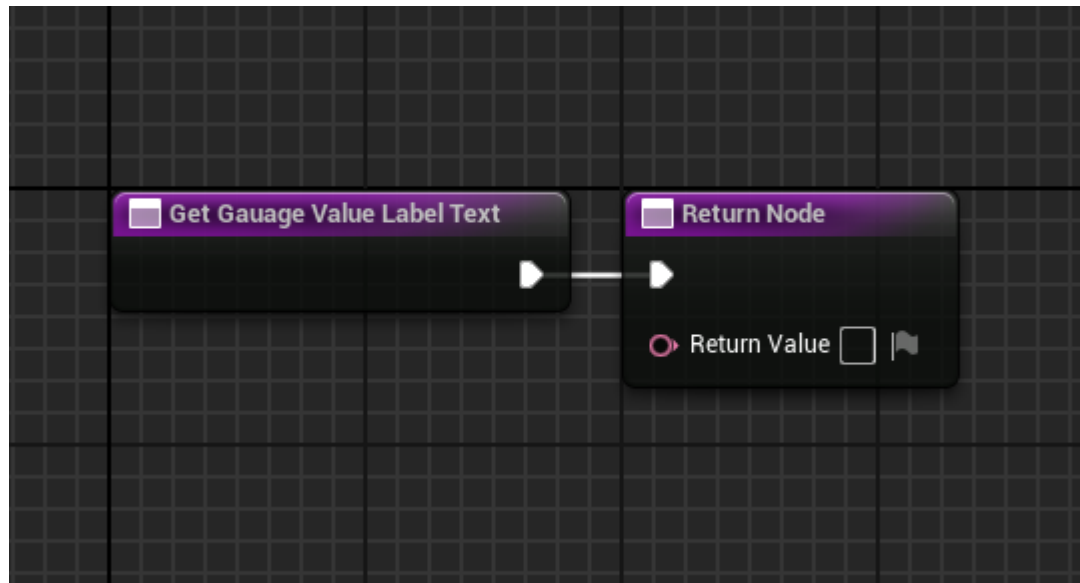
The function's return value is used directly as the text displayed on the gauge.

For example, it can be used to:

- Add custom formatting.
- Change the value representation.
- Add additional information to the value text.

```

▼ Gauge Chart
  ▼ Text
    ↻ GetGaugeValueLabelText
  
```



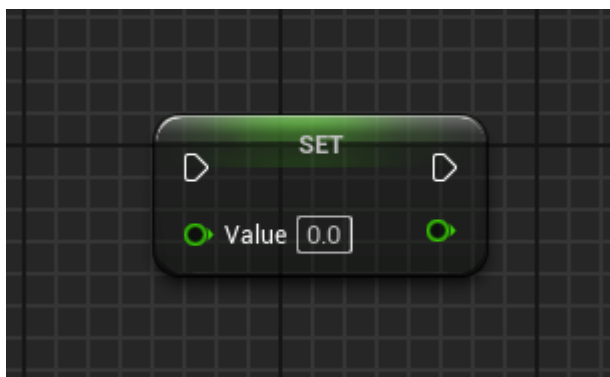
Note: Anything returned by this function will be displayed as the Gauge Value Text.

• Runtime Usage

The Gauge Chart does not require any function call to generate the chart. It can be configured and displayed directly in the **Widget Blueprint Editor**.

The gauge can also be updated at runtime by changing its **Value** property as required.

For smooth progress animation, gradually interpolate the value over time instead of changing it directly, for example by using a **Lerp** between the current value and the target value.



5.3 Multiple Gauge

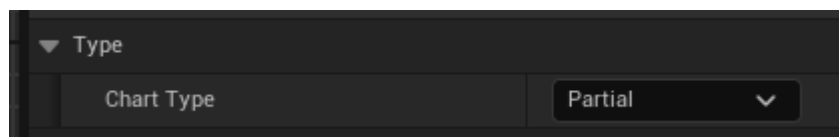
The **Multiple Gauge** displays multiple gauge values simultaneously, with each gauge arranged concentrically around the center. It is suitable for comparing multiple related metrics within a single chart, such as performance, progress, temperature, or completion percentages.

• Chart Properties

○ Gauge Chart Type

Controls the shape or portion of the gauge that is displayed.

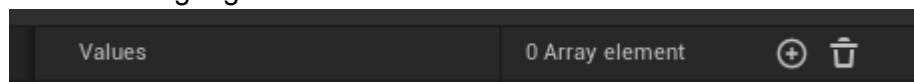
- **Half:** Displays a half-circle gauge.
- **Partial:** Displays a partial-circle gauge.
- **Three Quarter:** Displays a three-quarter-circle gauge.
- **Full:** Displays a complete circular gauge.



○ Value

Defines the current value for each gauge.

Each element in the **Value** array corresponds to one gauge. The value, together with the configured minimum and maximum values, determines how much of that gauge is filled.



○ Minimum Value

Defines the minimum value of the gauge range.



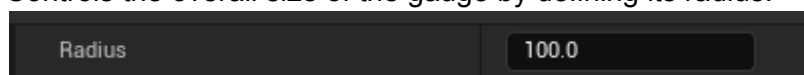
○ Maximum Value

Defines the maximum value of the gauge range.



○ Radius

Controls the overall size of the gauge by defining its radius.



○ Thickness

Controls the thickness of the gauge arc.

Thickness	35.0
-----------	------

○ Create Rounded Corners

Controls whether the ends of the gauge arc have rounded corners.

Create Rounded Corners	☒
------------------------	-------------------------------------

○ Border Properties

Controls the border displayed around the gauges.

- **Create Border Around Progress:** Controls whether a border is displayed around the gauge progress.
- **Border Thickness:** Controls the thickness of the border.

Create Borders Around Progress	☐
Border Thickness	1.0

○ Gap Between Gauges

Controls the gap between each gauges

Gap Between Gauges	2.5
--------------------	-----

○ Text Properties

Controls the text displayed on the gauge.

Text setting provides options to customize the text appearance and position, including:

- **Text:** Defines the text to display. For the Gauge Name, this is entered manually. The Value Label uses the gauge's **Value**.
- **Font Settings:** Controls the font and font appearance.
- **Text Color:** Controls the color of the text.
- **Text Transform:** Controls the position and transformation of the text.

○ Name Text

Controls the appearance of the gauge name. The **Text** property defines the

name displayed on the gauge.

▼ Name Text Setting	
Visible	☒
Text	Gauge Chart Multi
▶ Font	
▶ Color	
▶ Transform	

○ Multi Gauge Data

Contains the color and appearance settings for each gauge in the Multiple Gauge.

Each entry in the array represents **one gauge** and contains:

- **Background Color:** Defines the background color of the gauge.
- **Progress Color:** Defines the color of the gauge progress.
- **Border Color:** Defines the color of the gauge border, when enabled.

The number of entries in the **Multi Gauge Data** array determines the number of gauges displayed. Each gauge can have its own individual color settings.

▼ Data	
▼ Multi Gauge Data	3 Array elements + -
▼ Index [0]	3 members ▼
▶ Background Color	
▶ Progress Colors	2 Array elements + -
▶ Border Color	
▼ Index [1]	3 members ▼
▶ Background Color	
▶ Progress Colors	2 Array elements + -
▶ Border Color	
▼ Index [2]	3 members ▼
▶ Background Color	
▶ Progress Colors	2 Array elements + -
▶ Border Color	

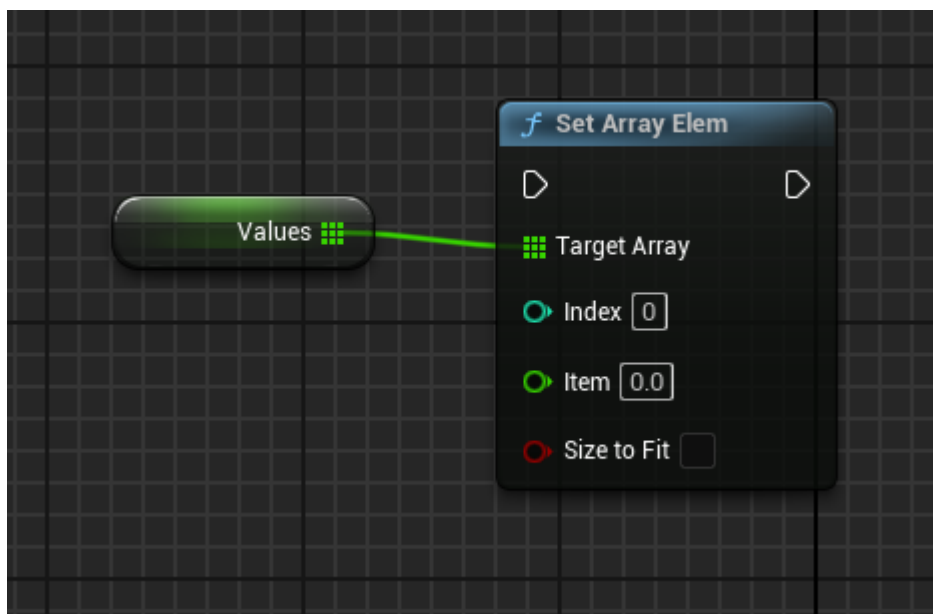
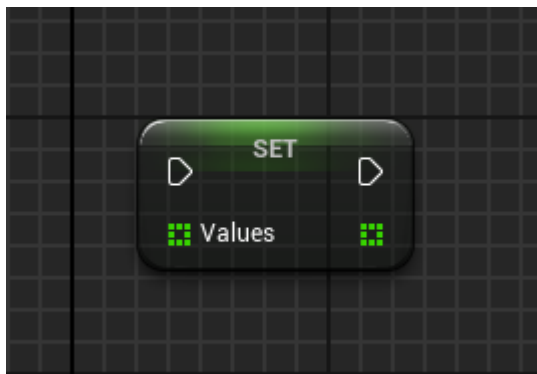
- **Runtime Usage**

The Gauge Chart does not require any function call to generate the chart. It can be configured and displayed directly in the **Widget Blueprint Editor**.

The gauge can be updated at runtime by changing its **Value** property as required. The gauge progress will update according to the new value.

For the **Multiple Gauge**, update the required element in the **Value array** to change the corresponding gauge. Replacing the entire array will update all gauges.

For smooth progress animation, gradually interpolate the value over time, for example by using a **Lerp** between the current value and the target value.



6. Barometer Chart

6.1 Overview

The **Barometer Chart** is used to visually represent a single numerical value within a defined range. It provides a gauge-like display designed to show the current level of a measurement in an easy-to-read format.

The Barometer Chart supports **one value at a time** and provides various properties to control its value, range, appearance, colors, text, and other visual settings.

The chart can be configured directly through the widget properties and updated at runtime by changing its value.

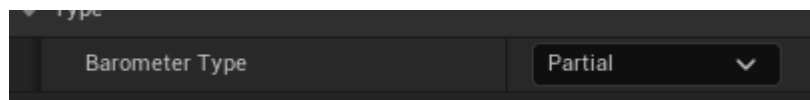
6.2 Chart Properties

- **Barometer Type**

Controls the shape or portion of the Barometer Chart that is displayed.

The Barometer supports three types:

- **Half:** Displays a half-circle barometer.
- **Partial:** Displays a partial-circle barometer.
- **Three Quarter:** Displays a three-quarter-circle barometer.



- **Value**

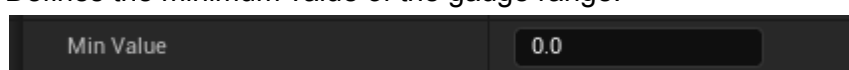
Represents the current value displayed by the barometer.

The barometer uses this value together with the minimum and maximum values to determine how much of the gauge should be filled.



- **Minimum Value**

Defines the minimum value of the gauge range.



- **Maximum Value**

Defines the maximum value of the gauge range.

Max Value	100.0
-----------	-------

- **Appearance Properties**

Controls the overall appearance and shape of the Barometer Chart.

- **Draw Progress**

Controls whether the progress section of the barometer is displayed.

Draw Progress	☒
---------------	-------------------------------------

- **Radius**

Controls the overall size of the barometer.

Radius	125.0
--------	-------

- **Thickness**

Controls the thickness of the barometer arc.

Thickness	20.0
-----------	------

- **Rounded Corners**

Controls whether the ends of the barometer arc have rounded corners.

Create Rounded Corners	☐
------------------------	--------------------------

- **Border Settings**

Controls the border displayed around the Barometer progress.

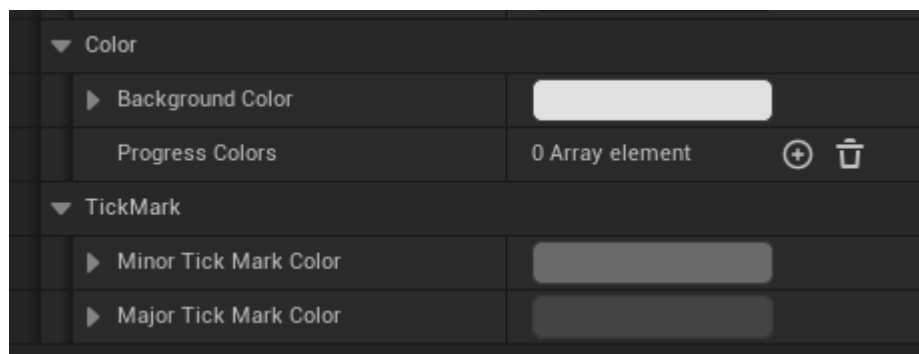
- **Create Border Around Progress:** Enables or disables the border around the progress section.
 - **Border Thickness:** Controls the thickness of the progress border.
 - **Border Color:** Controls the color of the progress border.

▼ Border	
Create Borders Around Progress	☒
Border Thickness	2.0
▶ Border Color	

- **Color Properties**

Controls the colors used for different elements of the Barometer Chart.

- **Background Color:** Controls the background color of the barometer.
- **Progress Colors:** Defines the colors used for the progress. Multiple colors can be added to create a gradient.
- **Major Tick Mark Color:** Controls the color of the major tick marks.
- **Minor Tick Mark Color:** Controls the color of the minor tick marks.

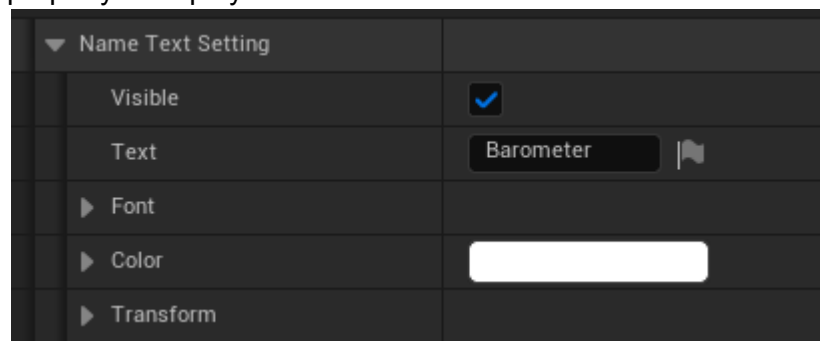


- **Text Properties**

Controls the text displayed on the Barometer Chart.

- **Name Text Settings**

Controls the appearance of the barometer name. The value entered in the **Text** property is displayed as the barometer name.



- **Value Text Settings**

Controls the appearance of the current value displayed on the barometer. The displayed value is taken from the barometer's **Value** property.

▼ Value Text Setting	
Visible	☒
Text	
▶ Font	
▶ Color	
▶ Transform	

- **Tick Text Settings**

Controls the appearance of the tick labels. The tick text is displayed above the **major tick marks**.

▼ Tick Text Setting	
Visible	☒
Text	
▶ Font	
▶ Color	
▶ Transform	

- **Tick Mark Settings**

Controls the visibility, placement, spacing, and size of the tick marks and their labels on the Barometer.

- **Show Tick Marks:** Controls whether tick marks are displayed.
- **Tick Mark Placement Type:** Controls where the tick marks are placed relative to the barometer: **Outside**, **Center**, or **Inside**.
- **Tick Mark Offset:** Controls the offset of the tick marks from their default position.
- **Tick Mark Label Offset:** Controls the offset of the tick mark labels.
- **Start/End Tick Mark Offset:** Adds an offset to the start and end of the tick mark range, allowing the tick marks to be aligned with the barometer borders or edges.
- **Show Major Tick Marks:** Controls whether major tick marks are displayed.
- **Show Minor Tick Marks:** Controls whether minor tick marks are displayed.
- **Show Tick Mark Labels:** Controls whether labels for the major tick marks are displayed.
- **Major Tick Mark Count:** Defines the number of major tick marks.
- **Minor Tick Marks Per Major Tick Mark:** Defines the number of minor tick marks generated between each pair of major tick marks.
- **Major Tick Mark Length:** Controls the length of the major tick marks.

- **Minor Tick Mark Length:** Controls the length of the minor tick marks.

TickMark	
Show Tick Marks	☒
Tick Mark Placement Type	Outside ▾
Tick Mark Offset	2.0
Tick Mark Label Offset	15.0
Start End Tick Mark Offset	0.5
Show Major Tick Marks	☒
Show Minor Tick Marks	☒
Show Tick Mark Labels	☒
Major Tick Mark Count	6
Minor Tick Marks Per Major	5
Major Tick Mark Length	12.0
Minor Tick Mark Length	6.0
Major Tick Mark Thickness	4.0
Minor Tick Mark Thickness	2.5

- **Needle Settings**

Controls the appearance, size, position, and rotation of the needle on the Barometer Chart.

- **Show Needle:** Controls whether the needle is displayed.
- **Needle Brush:** Defines the image, size and color used for the needle.
- **Needle Pivot:** Defines the pivot point used for rotating the needle around the barometer. The effective default value is approximately **(0.5, 0.925)**. This value may need to be adjusted depending on the needle image for proper rotation.
- **Needle Offset:** Controls the position of the needle relative to the center of the barometer.
- **Needle Rotation Offset:** Applies an additional rotation to the needle image to align it with the desired angle.

Needle	
Show Needle	☒
▶ Needle Brush	
▶ Needle Pivot	0.5 0.925
▶ Needle Offset	0.0 0.0
Needle Rotation Offset	90.0

• Needle Center Settings

Controls the appearance of the center point of the needle. The needle center is displayed at the starting point of the needle and overlaps it to provide a clean center appearance.

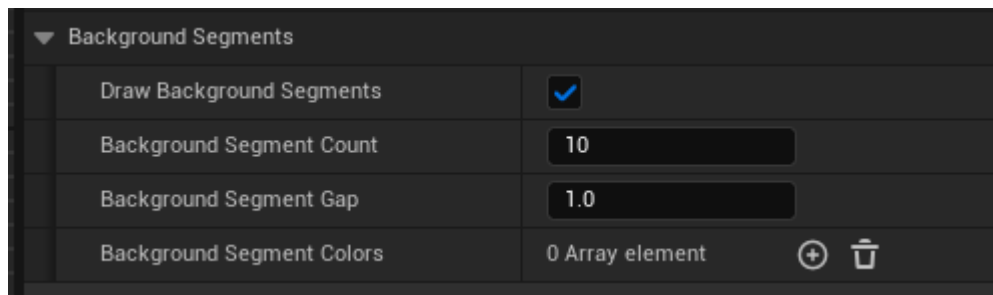
- **Show Needle Center:** Controls whether the needle center is displayed.
- **Needle Center Brush:** Defines the image and color used for the needle center.
- **Needle Center Offset:** Controls the position of the needle center relative to the barometer center.

Show Needle Center	☒
▶ Needle Center Brush	
▶ Needle Center Offset	0.0 0.0

• Background Segment Settings

Controls the coloured segments displayed in the background of the Barometer instead of using a single background color.

- **Draw Background Segments:** Controls whether background segments are displayed.
- **Background Segment Count:** Defines the number of background segments to generate.
- **Background Segment Gap:** Controls the spacing between each background segment.
- **Background Segment Colors:** Defines the colors used for the background segments. Add the required number of colors to create the desired segmented appearance.



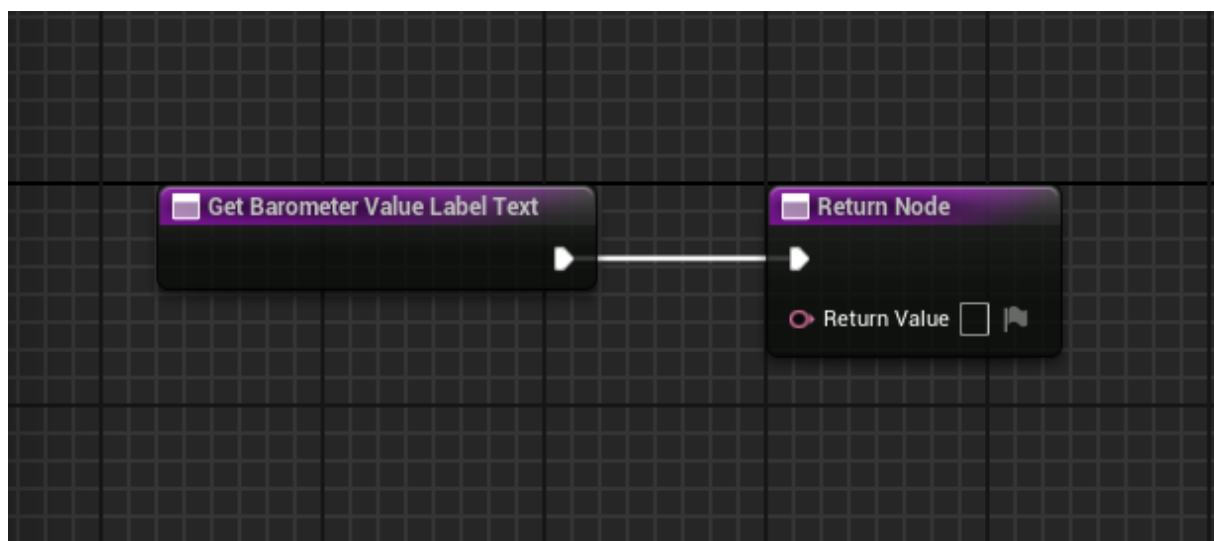
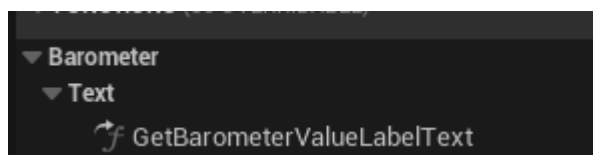
6.3 Chart Functions

The Barometer Chart provides one optional function that can be overridden to customize the value text displayed on the chart.

- **Get Barometer Value Label Text**

This function can be overridden to customize or format the **Value Label** displayed on the Barometer.

The function's return value is used directly as the value text shown on the chart. This can be used to change the formatting, add prefixes or postfixes, or display custom text based on the current value.



6.4 Runtime Usage

The Barometer Chart does not require any function call to generate the chart. It can be configured and displayed directly in the **Widget Blueprint Editor**.

The barometer can be updated at runtime by changing its **Value** property. The barometer progress and needle will update according to the new value.

For smooth value animation, gradually interpolate the value over time, for example by using a **Lerp** between the current value and the target value.



7. Step Line Chart

7.1 Overview

The **Step Line Chart** is used to display changes in numerical values across an ordered set of data points. Instead of connecting points with a diagonal line, the chart uses **horizontal and vertical segments**, creating a step-like appearance.

It is useful for displaying data where values change at specific points, such as status changes, measurements, levels, or other sequential data.

The Step Line Chart supports customization of the **data, line appearance, padding, grid, text, and animation** through the widget properties.

7.2 Chart Properties

- **Step Line Entry Data**

Contains all the data required to generate the Step Line Chart.

- **Group Name:** Defines the name of the Step Line Chart data group.
- **Description:** Provides additional information about the data.
- **Entry Container:** Contains the individual data points of the step line. Each entry includes:
 - **Name:** Defines the name of the individual data point.

- **Value:** Defines the numerical value of the data point.
- **Value Text Translation:** Controls the position of the value text for the individual point, allowing the label to be adjusted for better representation.

Step Line Entry Data	
Group Name	
Group Description	
Chart Entries	0 Array element + -

• Step Line Point Position

Controls the position of the data points along the step line.

- **Start:** Places the point at the start of the step segment.
- **Middle:** Places the point in the middle of the step segment.
- **End:** Places the point at the end of the step segment.

Step Line Point Position	Middle ▼
--------------------------	-----------------------

• Line Settings

Controls the appearance and positioning of the step line within the chart.

- **Line Color:** Controls the color of the step line.
- **Line Thickness:** Controls the thickness of the step line.
- **Step Line Padding:** Adds padding to the sides of the chart, creating space between the chart edges and the start/end points of the step line.

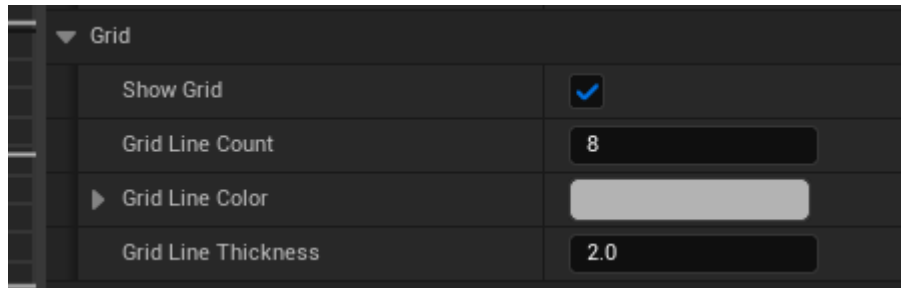
Line	
▶ Line Color	<input type="color" value="#0000FF"/>
Line Thickness	3.0
▶ Step Line Padding	20.0, 0.0

• Grid Settings

Controls the grid lines displayed in the Step Line Chart.

- **Show Grid:** Controls whether grid lines are displayed.

- **Grid Line Count:** Defines the number of grid lines.
- **Grid Line Color:** Controls the color of the grid lines.

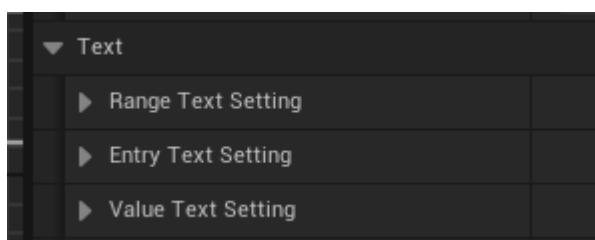


• Text Settings

Controls the appearance of the different text elements displayed in the Step Line Chart.

- **Range Text Settings:** Controls the appearance of the value range labels displayed along the chart.
- **Entry Text Settings:** Controls the appearance of the entry names displayed for each data point.
- **Value Text Settings:** Controls the appearance of the value labels displayed for each data point.

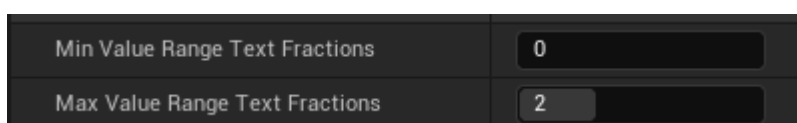
Each text setting provides options such as **font, color, padding, visibility, and text transform** to customize the text appearance and position.



• Value Range Text Fractions

Controls the number of decimal places displayed for the **value range labels**.

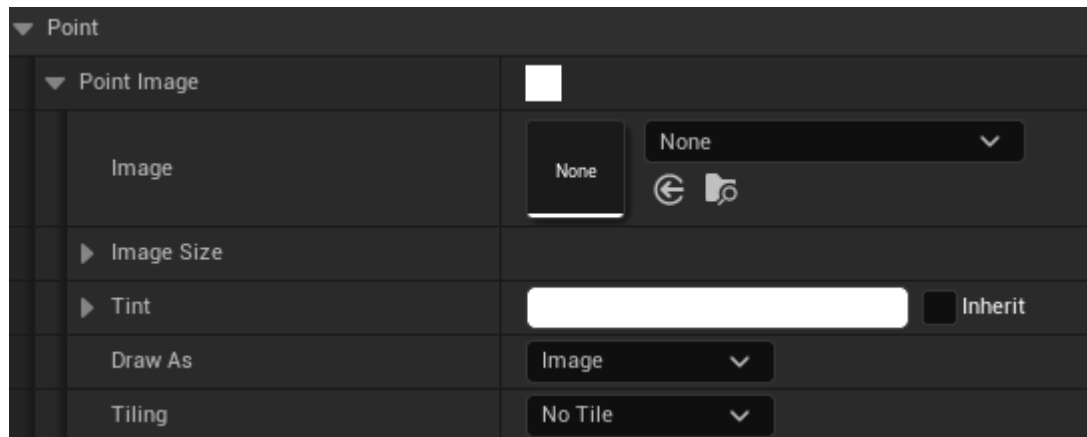
It provides separate **minimum and maximum fraction limits**, allowing the displayed value range numbers to show the desired number of decimal places when needed.



- **Point Image**

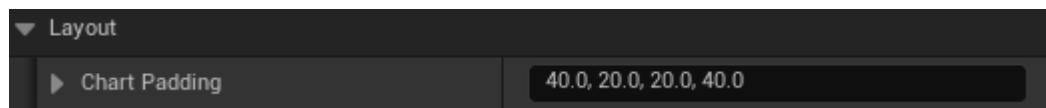
Controls the appearance of the individual data points displayed on the Step Line Chart.

- **Point Image Brush:** Defines the image, size, and color used for the points on the step line.



- **Chart Padding**

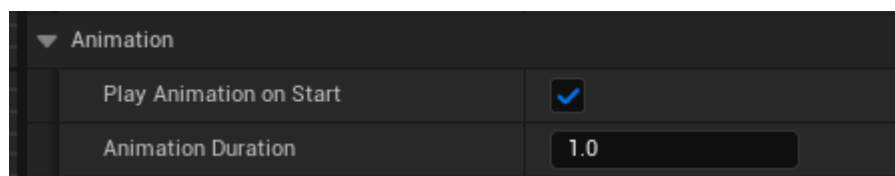
Adds padding around the generated chart within the widget bounds, creating space between the chart content and the edges of the widget.



- **Animation Settings**

Controls the animation of the Step Line Chart when it is generated.

- **Play Animation on Start:** Controls whether the step line animation plays when the chart is generated.
- **Animation Duration:** Defines how long the chart animation takes to complete.



7.3 Chart Functions

The Step Line Chart provides two functions for generating and updating the chart: **Refresh Chart** and **Update Chart**.

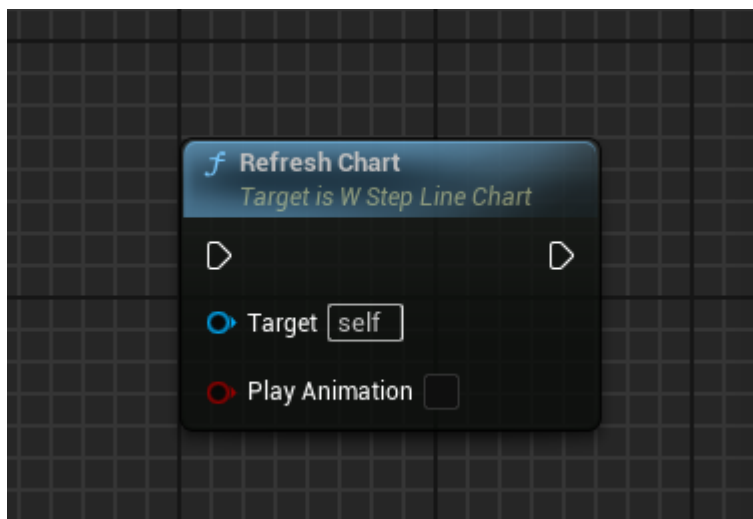
- **Refresh Chart**

Refreshes the chart using its current data and chart properties.

Use this function when the chart properties or existing data have been changed and the chart needs to be regenerated.

It includes a **Play Animation** parameter:

- **True:** Plays the bar animation when generating the chart.
- **False:** Generates the chart without animation.



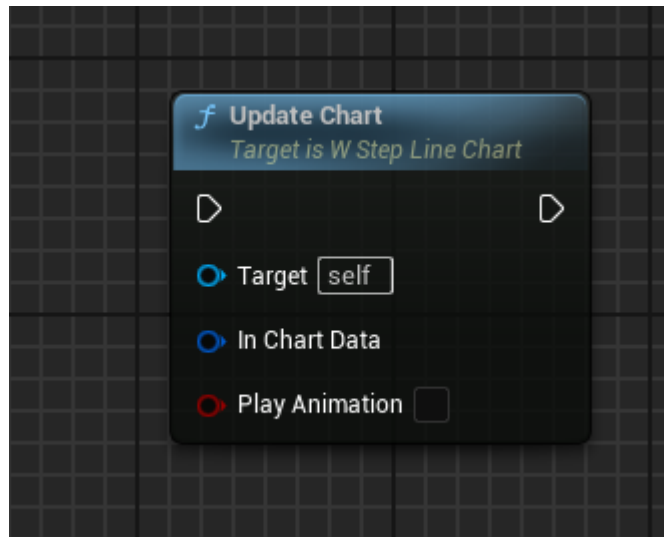
Note: Changes made to the chart properties are applied when the chart is refreshed.

- **Update Chart**

Updates the chart with new bar entry data.

Pass the new **Step Line Chart Data** to generate the chart using the updated data. It also includes a **Play Animation** parameter to control whether the bar animation should play.

- **True:** Plays the bar animation.
- **False:** Generates the chart without animation.



7.4 Runtime Usage

The Step Line Chart is generated directly in the **Widget Blueprint Editor**, so no function call is required to generate the chart at runtime.

The chart can be updated at runtime using **Refresh Chart** or **Update Chart** when the data or chart properties need to be changed.

If **Play Animation on Start** is enabled, the chart animation will play automatically when the project starts.

For runtime changes, the **Play Animation** parameter in the Refresh/Update functions can be used to control whether the animation should play.

8. TreeMap Chart

8.1. Overview

The **TreeMap Chart** is used to represent hierarchical data using rectangular areas. Each entry is displayed as a rectangle, where the **size of the rectangle is determined by its value**.

The chart supports **groups and entries**, allowing related data to be organized together. Each group can contain multiple entries, and the layout automatically arranges the entries based on their values.

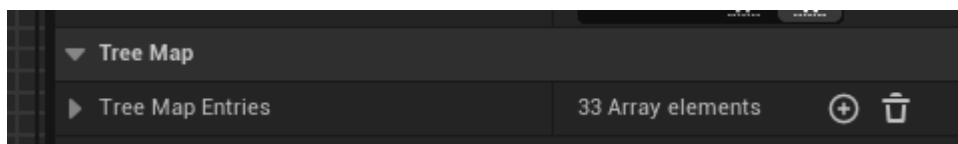
The TreeMap Chart provides properties to control the **data, colors, layout, text, padding, animation, and interaction**.

8.2 Chart Properties

- **Treemap Entries**

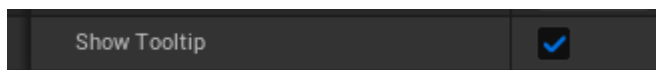
Contains all the data required to generate the TreeMap Chart, including groups and their entries.

- **Group:** Defines a group of related entries. Each group contains:
 - **Group Name:** Defines the name of the group.
 - **Group Description:** Provides additional information about the group.
 - **Group Color:** Defines the color associated with the group.
 - **Value Entries:** Contains the individual entries within the group.
- **Value Entry:** Defines an individual TreeMap item and contains:
 - **Name:** Defines the name of the entry.
 - **Description:** Provides additional information about the entry.
 - **Value:** Determines the size of the entry's rectangle.
 - **Color:** Defines the color of the entry.



- **Show Tooltip**

Controls whether a tooltip is displayed when hovering over a bar.

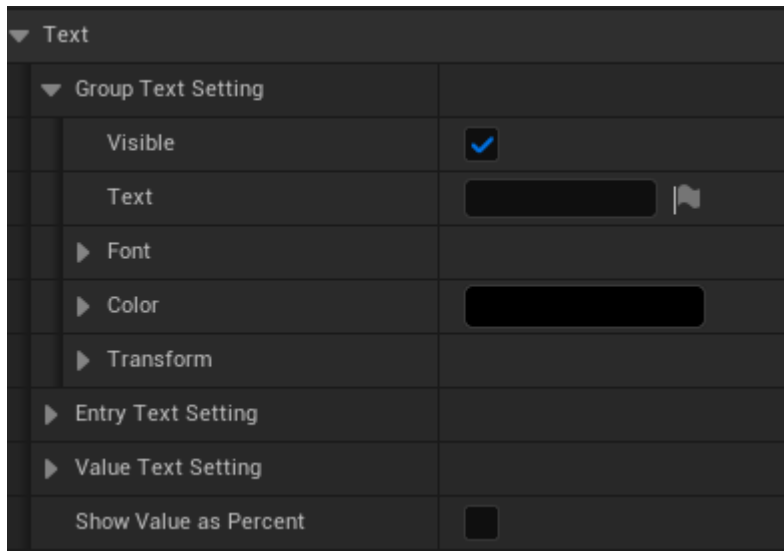


- **Text Settings**

Controls the text displayed for groups and entries in the TreeMap Chart.

- **Group Name Text Settings:** Controls the appearance of the group names.
- **Entry Name Text Settings:** Controls the appearance of the individual entry names.

- **Value Text Settings:** Controls the appearance of the value displayed for each entry.
- **Show Value as Percentage:** When enabled, displays each entry's value as a percentage of the total value of its group instead of displaying the raw value.



● Layout Settings

Controls the spacing, padding, and sizing of groups and entries in the TreeMap Chart.

- **Group Padding:** Controls the padding between the contents of a group and its boundary.
- **Group Spacing:** Controls the spacing between separate TreeMap groups.
- **Entry Padding:** Controls the padding between an entry and its group or row boundary.
- **Entry Spacing:** Controls the spacing between adjacent entries.
- **Group Header Height:** Defines the height reserved for displaying the group header.

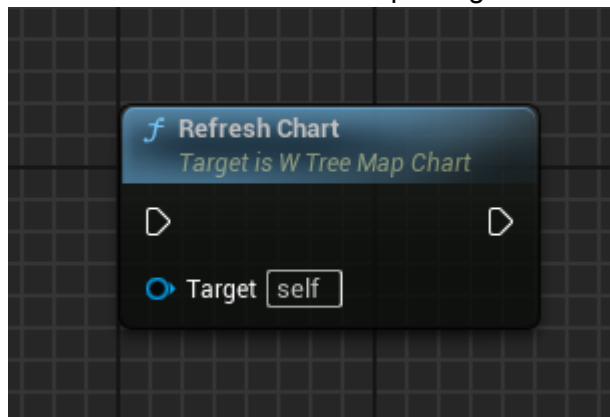


8.3 Chart Functions

The TreeMap Chart provides two functions for generating and updating the chart.

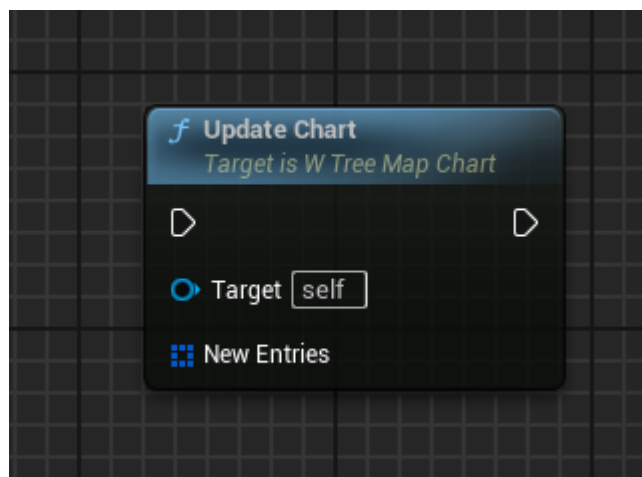
- **Refresh Chart**

Refreshes the TreeMap using its current **TreeMap Entries** and chart properties.



- **Update Chart**

Updates the TreeMap with new **TreeMap Entries** and regenerates the chart using the provided data.



8.4 Runtime Usage

The chart is generated at runtime only when **Refresh Chart** or **Update Chart** is called.

For example, to generate the chart when using it in a Widget Blueprint, call one of these functions from the **WBP_TreeMapchart Construct** event.

Note: Simply changing the widget properties does not automatically regenerate the chart. Call **Refresh Chart** after changing properties or **Update Chart** when providing new chart data.

9. Circle Relationship Chart

9.1. Overview

The **Circle Relationship Chart** is used to visualize relationships between multiple nodes in a circular layout. Each node is displayed around a circle, while curved connection lines represent the relationships between nodes.

It is useful for displaying **connections, dependencies, associations, or relationships** between different items in an easy-to-understand visual format.

The chart supports customization of the **nodes, relationships, node appearance, text, connection lines, and layout** through the widget properties.

9.2 Chart Properties

- **Circle Relationship Chart Data**

The Circle Relationship Chart uses two main data properties: **Nodes** and **Relationships**.

- **Nodes**

Contains all the nodes displayed in the relationship graph. Each node represents an individual item in the chart.

Each node contains:

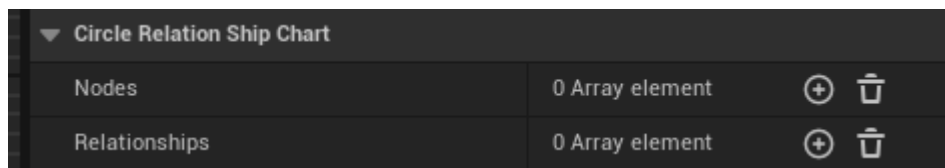
- **Node Name:** A unique name used to identify the node and reference it when creating relationships.
- **Text:** The text displayed for the node.
- **Node Color:** Defines the color of the node.
- **Radius:** Controls the size of the node.

- **Relationships**

Defines the connections between the nodes in the graph.

Each relationship contains:

- **Source Node:** The name of the node from which the relationship starts. It must match a **Node Name** from the Nodes array.
- **Target Nodes:** Contains the names of the nodes connected to the source node. Multiple target nodes can be added to create multiple connections from the same source node.

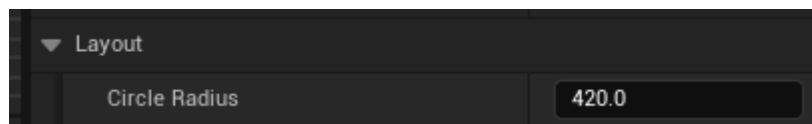


Note: Relationships are created using **node names**, so the names used in Source Node and Target Nodes should match the corresponding **Node Name** values in the Nodes array

• Circle Radius

Defines the radius of the circle on which the nodes are positioned.

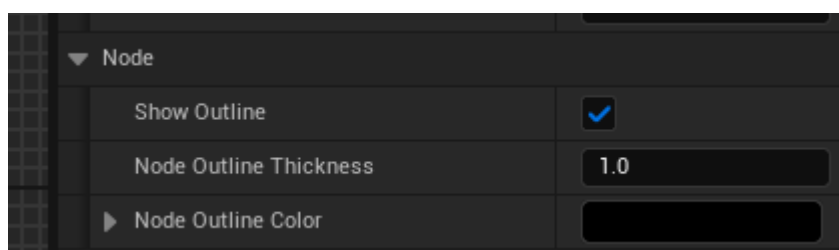
A larger value places the nodes farther from the center, while a smaller value brings them closer together.



• Node Outline Settings

Controls the outline displayed around each node.

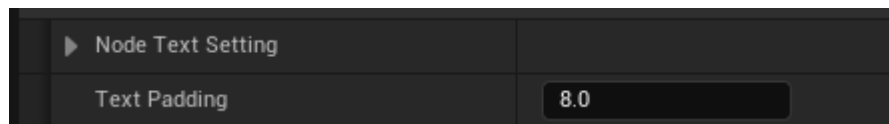
- **Show Outline:** Controls whether an outline is drawn around the nodes.
- **Node Outline Thickness:** Controls the thickness of the node outline when enabled.
- **Node Outline Color:** Controls the color of the node outline when enabled.



- **Text Settings**

Controls the appearance and positioning of the text displayed for each node.

- **Node Text Settings:** Controls the visibility and appearance of the node text, including font, color, padding, and text transform.
- **Text Padding:** Defines the spacing between the node text and the node boundary.



- **Relationship Settings**

Controls the appearance and shape of the relationship lines connecting the nodes.

- **Relationship Thickness:** Controls the thickness of the lines connecting the nodes.
- **Relationship Curve Strength:** Controls how strongly the relationship lines curve toward the center of the chart. Higher values create a stronger curve.
- **Curve Segments:** Defines the number of segments used to generate the curved relationship lines. Higher values produce smoother curves.

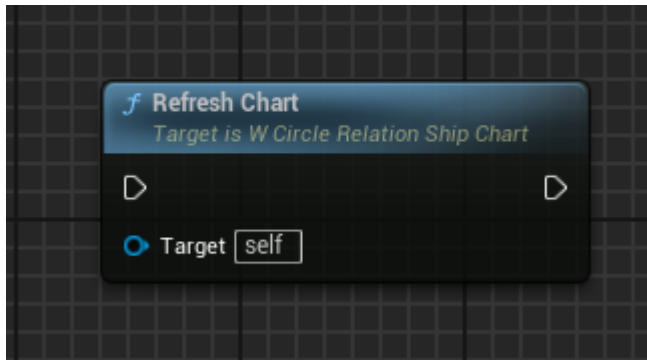


9.2 Chart Functions

The Circle Relationship Chart provides two functions to generate and update the relationship graph.

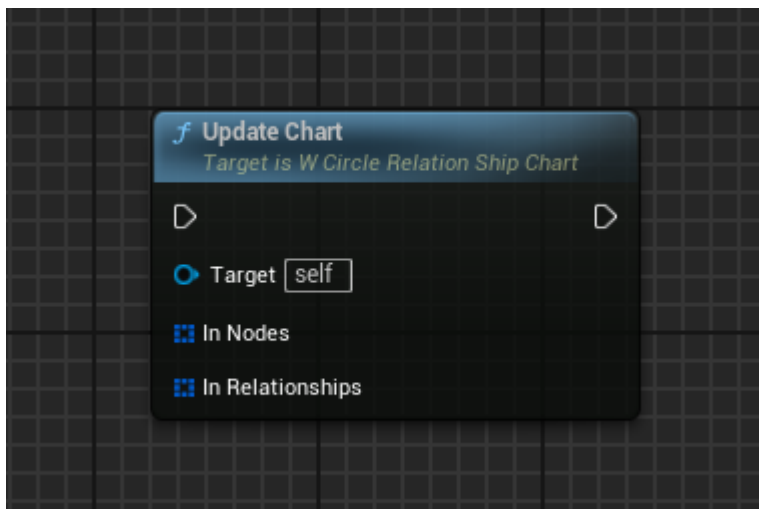
- **Refresh Chart**

Refreshes the chart using the current **Nodes** and **Relationships** data.



- **Update Chart**

Updates the chart with new **Nodes** and **Relationships** data and regenerates the relationship graph.



9.4 Runtime Usage

The Circle Relationship Chart is generated directly in the **Widget Blueprint Editor**, so no function call is required to generate the chart at runtime.

The chart can be updated at runtime using **Refresh Chart** or **Update Chart** when the data or chart properties need to be changed.

10. Gantt Chart

10.1 Overview

The **Gantt Chart** is used to visualize tasks and their durations across a timeline. Each task is represented as a bar, where its position and length indicate its **start time and end time**.

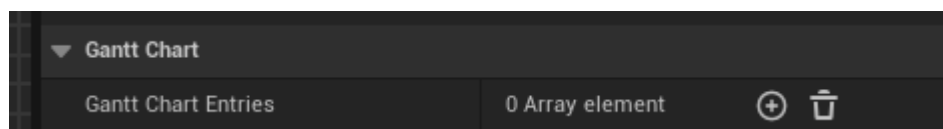
The chart supports **multiple timelines**, allowing the chart to display time in different units such as **hours, days, weeks, or months**.

It is suitable for displaying **project schedules, task durations, planning, milestones, and other time-based data**.

10.2 Chart Properties

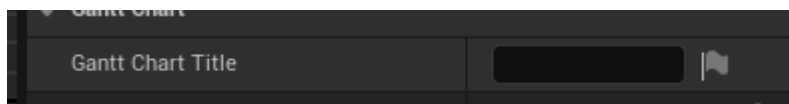
- **Gantt Chart Entries**

- Contains all the data used to generate the Gantt Chart. Each element in this array represents a **group** displayed as a separate row in the chart.
- Each group can contain **multiple timeline entries**, allowing several tasks or time periods to be displayed within the same group. The start and end times of each entry determine its position and duration on the timeline.



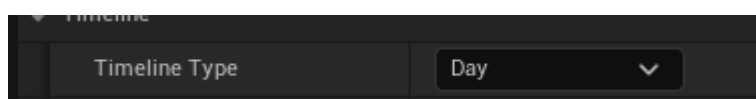
- **Gantt Chart Title**

- Defines the title text displayed on the **left side of the Gantt Chart**. It can be used to label or identify the chart.



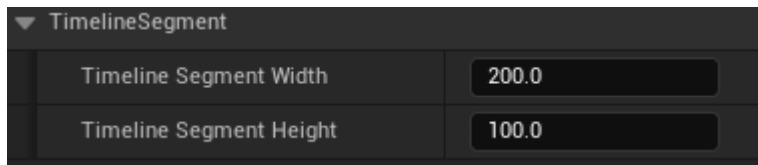
- **Timeline Type**

- Defines the time interval used to divide and display the Gantt Chart timeline, such as **Hour, Day, Week, or Month**. This determines how the timeline is organized and how its time divisions are displayed.



- **Timeline Segment Settings**

- **Timeline Segment Width:** Defines the width of each time segment displayed in the timeline.
- **Timeline Segment Height:** Defines the height of each time segment displayed in the timeline.

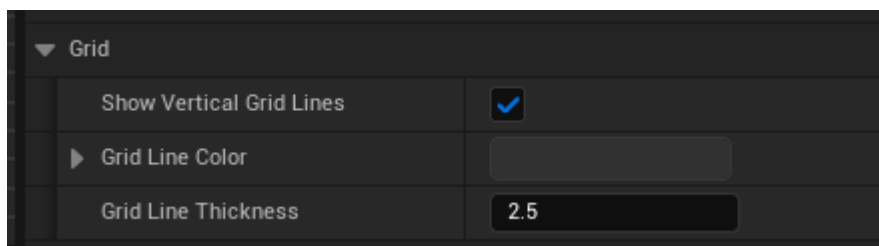


TimelineSegment	
Timeline Segment Width	200.0
Timeline Segment Height	100.0

- **Grid Line Settings**

Controls the appearance of the vertical grid lines displayed across the Gantt Chart.

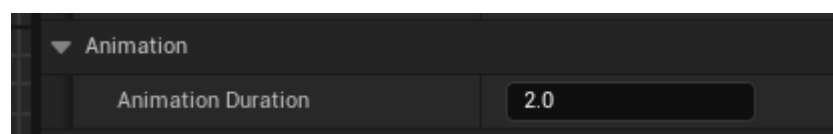
- **Show Vertical Grid Lines:** Controls whether vertical grid lines are displayed in the chart.
- **Grid Line Color:** Defines the color of the vertical grid lines.
- **Grid Line Thickness:** Defines the thickness of the vertical grid lines.



Grid	
Show Vertical Grid Lines	☒
Grid Line Color	
Grid Line Thickness	2.5

- **Animation Settings**

- **Animation Duration:** Defines how long the Gantt chart entry bars take to complete their animation, in seconds. A higher value results in a slower animation, while a lower value makes the animation faster.

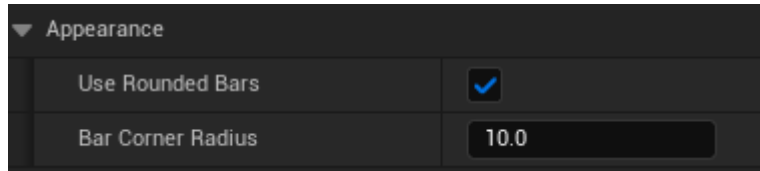


Animation	
Animation Duration	2.0

- **Bar Appearance Settings**

Controls the appearance of the bars used to represent entries in the Gantt Chart.

- **Use Rounded Bars:** Controls whether rounded corners are applied to the Gantt Chart bars.
- **Bar Corner Radius:** Defines how rounded the corners of the bars are when rounded bars are enabled.



- **Show Tooltip**

Controls whether a tooltip is displayed when hovering over a Gantt Chart entry.



○

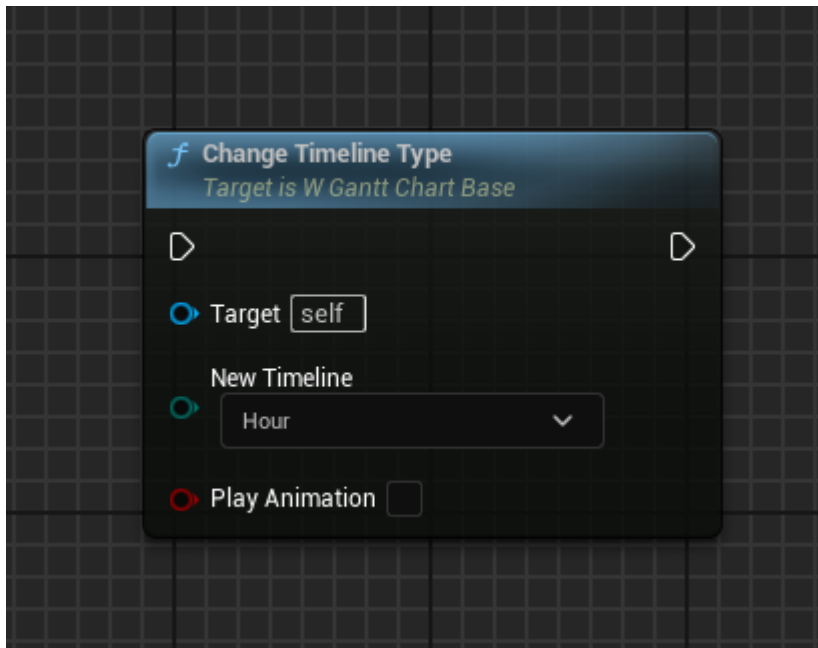
10.3 Chart Functions

The Gantt Chart provides three functions for changing the timeline, updating the chart data, and regenerating the chart.

- **Change Timeline Type**

Changes the timeline interval used by the Gantt Chart.

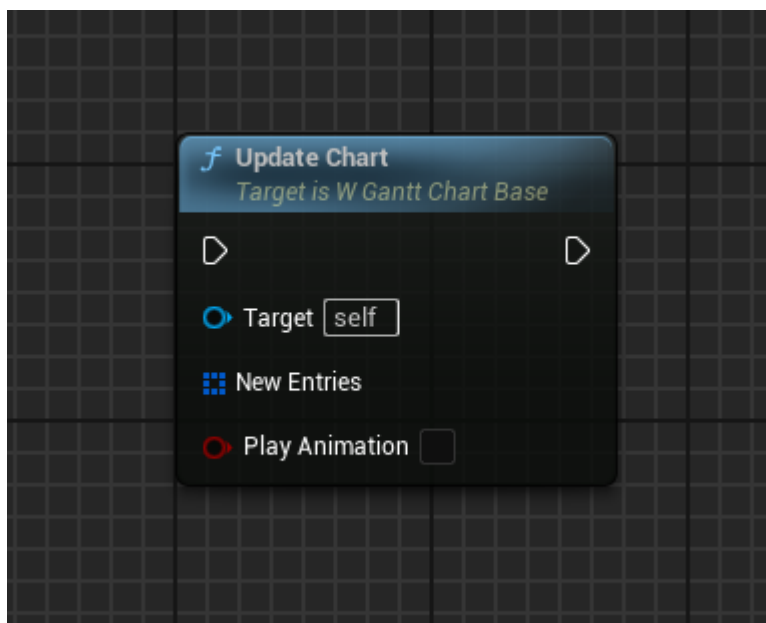
- **New Timeline:** Defines the new timeline type to use, such as Hour, Day, Week, or Month.
- **Play Animation:** Controls whether the chart generation animation plays after changing the timeline type.



- **Update Chart**

Updates the Gantt Chart with new entry data and regenerates the chart.

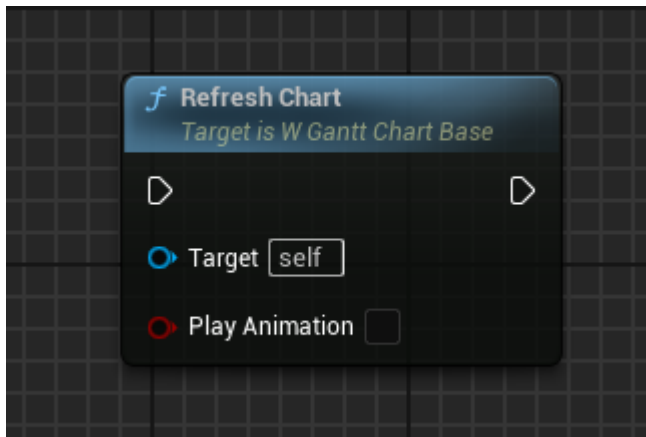
- **New Entries:** The new groups and timeline entries used to generate the chart.
- **Play Animation:** Controls whether the chart generation animation plays when the chart is updated.



- **Refresh Chart**

Regenerates the Gantt Chart using the currently configured **Gantt Chart Entries** and chart properties.

- **Play Animation:** Controls whether the chart generation animation plays when the chart is refreshed.



10.4 Runtime Usage

The Gantt Chart is **not generated or displayed in the Widget Blueprint Editor**. It is generated only at runtime.

To generate the chart, call any one of the following functions from the required runtime logic:

- **Refresh Chart:** Generates the chart using the currently configured Gantt Chart Entries and properties.
- **Update Chart:** Generates the chart using the provided Gantt Chart Entries.
- **Change Timeline Type:** Changes the timeline type and regenerates the chart using the new timeline setting.

The **Play Animation** parameter can be enabled when calling these functions to play the Gantt bar generation animation.

Widget and Demo Reference

This section provides a reference for all chart widgets included in the plugin and their corresponding demo widgets. It includes the **widget class**, **demo Widget Blueprint**, and their **Content Browser paths**.

The chart widgets are the reusable components intended to be added to your own Widget Blueprints. The demo widgets provide working examples with preconfigured data and properties, allowing you to understand how each chart is configured and how data is supplied.

- **Chart Widgets and Demo Widgets**

- **Bar Chart**

Chart Widget: WBP_BarChart

Demo Widget: WBP_BarChartSample

Chart Widget Path: /InteractiveChartsToolkitV2/Charts/Bar/WBP_BarChart

Demo Widget Path:

/InteractiveChartsToolkitV2/Demo/Widget/Charts/WBP_BarChartSample

- **Stacked Bar Chart**

Chart Widget: WBP_StackedBarChart

Demo Widget: WBP_StackedBarChartSample

Chart Widget Path:

/InteractiveChartsToolkitV2/Charts/Bar/StackedBarChart/WBP_StackedBarChar

Demo Widget Path:

/InteractiveChartsToolkitV2/Demo/Widget/Charts/WBP_StackedBarChartSample

- **Gauge Chart Single**

Chart Widget: WBP_GaugeChartSingle

Demo Widget: WBP_GaugeChartSample

Chart Widget Path:

/InteractiveChartsToolkitV2/Charts/Guage/WBP_GaugeChartSingle

Demo Widget Path:

/InteractiveChartsToolkitV2/Demo/Widget/Charts/WBP_GaugeChartSample

- **Multiple Gauge**

Chart Widget: WBP_GaugeChartMultiple

Demo Widget: WBP_GaugeChartSample

Chart Widget Path:

/InteractiveChartsToolkitV2/Charts/Guage/WBP_GaugeChartMultiple

Demo Widget Path:

/InteractiveChartsToolkitV2/Demo/Widget/Charts/WBP_GaugeChartSample

- **Barometer**

Chart Widget: WBP_Barometer

Demo Widget: WBP_BarometerSample

Chart Widget Path: /InteractiveChartsToolkitV2/Charts/Barometer/WBP_Barometer

Demo Widget Path:

/InteractiveChartsToolkitV2/Demo/Widget/Charts/WBP_BarometerSample

- **Step Line Chart**

Chart Widget: WBP_StepLineChart

Demo Widget: WBP_StepLineChartSample

Chart Widget Path:

/InteractiveChartsToolkitV2/Charts/StepLine/WBP_StepLineChart

Demo Widget Path:

/InteractiveChartsToolkitV2/Demo/Widget/Charts/WBP_StepLineChartSample

○ **TreeMap Chart**

Chart Widget: WBP_TreeMapChart

Demo Widget: WBP_TreeMapChartSample

Chart Widget Path:

/InteractiveChartsToolkitV2/Charts/TreeMap/WBP_TreeMapChart

Demo Widget Path:

/InteractiveChartsToolkitV2/Demo/Widget/Charts/WBP_TreeMapChartSample

○ **Circle Relationship Chart**

Chart Widget: WBP_CircleRelationshipChart

Demo Widget: WBP_CircleRelationSample

Chart Widget Path:

/InteractiveChartsToolkitV2/Charts/CircleRelationship/WBP_CircleRelationshipChart

Demo Widget Path:

/InteractiveChartsToolkitV2/Demo/Widget/Charts/WBP_CircleRelationSample

○ **Gantt Chart**

Chart Widget: WBP_GanttChart

Demo Widget: WBP_GanttChartSample

Chart Widget Path: /InteractiveChartsToolkitV2/Charts/Gantt/WBP_GanttChart

Demo Widget Path:

/InteractiveChartsToolkitV2/Demo/Widget/Charts/WBP_GanttChartSample

● **Chart Widgets**

The **chart widgets** are the main reusable components of the plugin. They can be added to your own Widget Blueprints and configured using their exposed properties.

● **Demo Widgets**

The **demo widgets** are example Widget Blueprints containing a configured instance of their corresponding chart widget.

Each demo contains example data so you can see how the chart is expected to be configured and how its data structures are populated.

The demo widgets can be used to:

- Understand how to add the chart widget.
- See how the chart's data is structured.
- Inspect example groups and entries.
- Understand how values, colors, names, and descriptions are assigned.
- Test the chart's available properties.
- See how runtime functions are used where applicable.
- Use the existing data as a starting point for your own chart.

- **Adding Your Own Data**

To create your own chart, add the corresponding **chart widget** to your Widget Blueprint and configure its exposed data properties.

For charts with structured data, such as **TreeMap**, **Stacked Bar**, **Circle Relationship**, and **Gantt**, the demo provides an example of how the nested data structures should be populated.

You can open the corresponding demo Widget Blueprint, inspect the configured chart data, and replace the example values with your own data.

For charts that are generated only at runtime, the required **Refresh Chart** or **Update Chart** function must be called after the data has been configured. Charts that support editor preview can be configured and viewed directly in the Widget Blueprint Editor.

Support & Customization:

This pack is designed to be a “living” asset and will continue to evolve over time. If you encounter any issues or have questions regarding implementation, please feel free to reach out through our [support link](#) or [Discord server](#).

For faster support, we recommend joining our [Discord server](#). Our team is ready to help and answer any questions you may have.

Level up your game's feel and performance today!

★★★★★ Your Review means the world to us!
Great feedback comes back as even better features.

If there are any suggestions or ideas for improvement, feel free to contact us. If possible, we will include your request in future updates.

 Follow us for updates and news!

[Youtube](#) | [Facebook](#) | [Email](#) | [Linkedin](#)

Website: 300mind.studio

*Thank you for choosing the **InteractiveChartsToolkit 2**. We hope this documentation helps you. Happy Gaming!*

Enjoy building your game! 

THANK YOU