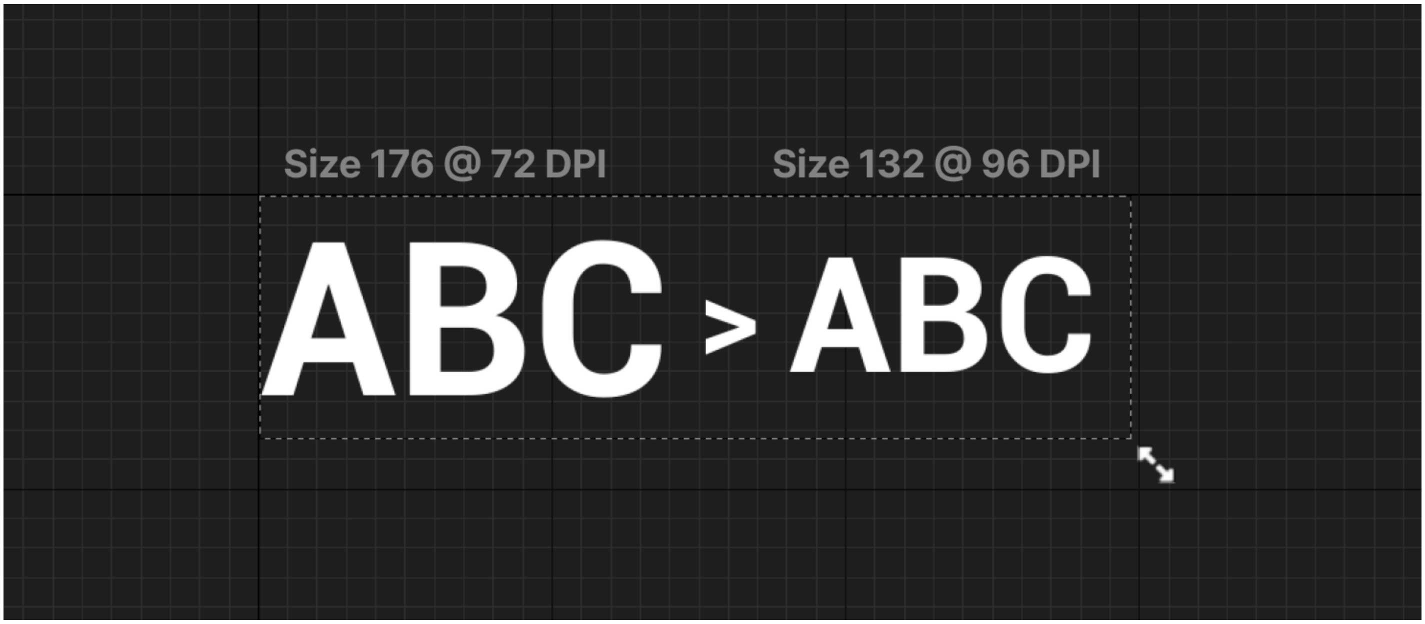
Developer
/ Documentation
/ Unreal Engine
/ Unreal Engine 5.4 Documentation
/ Creating User Interfaces
/ Text Formatting, Localization, and Fonts
/ Fonts
/ Font DPI Scaling

Font DPI Scaling

Configure the way text size is calculated in your project.



UMG calculates the size of characters in text using a global **Font Resolution** value, represented as DPI (dots per inch):

```
Pixel Size = ( Point Size of Font * Screen Resolution ) / (Font Resolution in
```

Copy full snippet

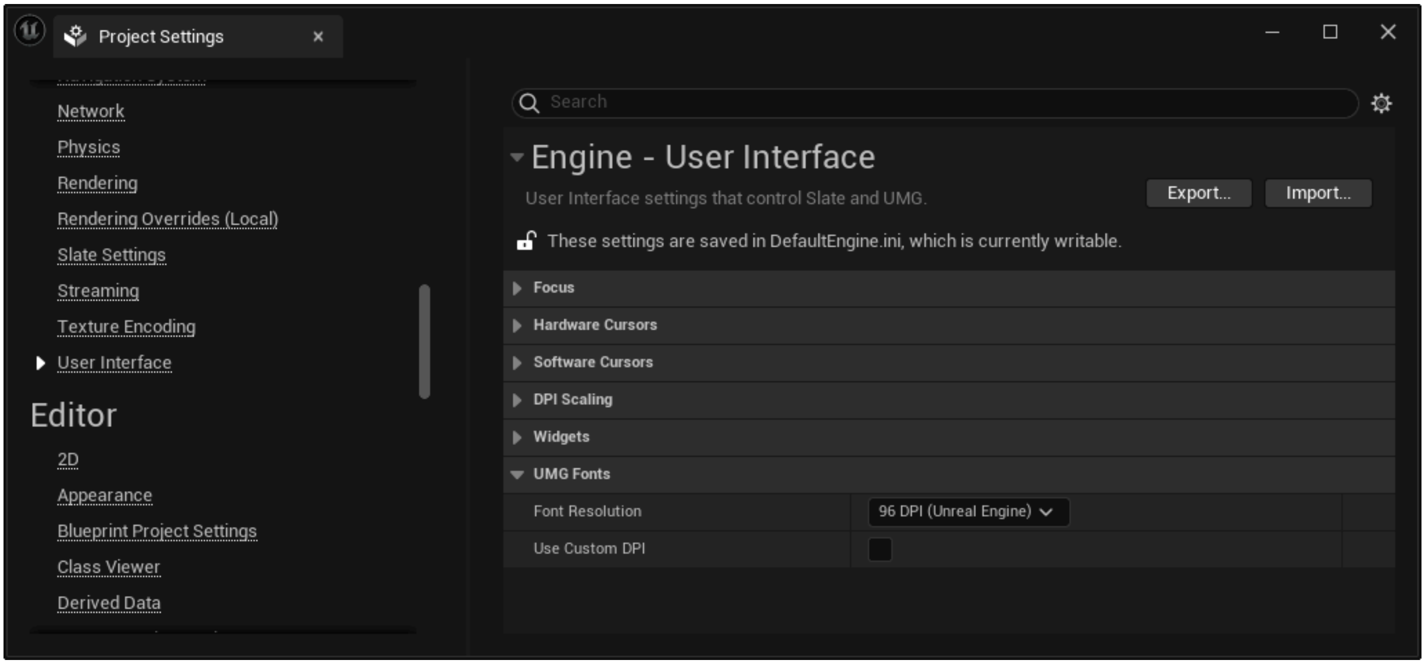
For example, a 12-point font displayed at 72 DPI would be 12/72 of an inch. In **Unreal Engine (UE)**, you can set the Font Resolution to customize how font sizes are interpreted. This makes it possible to develop user interfaces with the standards your team is most comfortable with, or the standards that are most often used in your external applications. This page provides:

- Instructions on how to configure your project's DPI.
- Technical information about how font scaling is handled.
- Best practices for managing font sizes.

> The above computation is simplified for the purpose of explaining the relationship between font point size and DPI. The actual pixel size is computed based on more factors, such as scaling. While Font Resolution changes the way Font Sizes are interpreted for the benefit of your team, the final calculation for on-screen pixel size includes a conversion to Unreal Engine's native 96 DPI. Refer to Font Resolution and Font Size for more information.

Configure Global Font Resolution (DPI)

You can globally change the DPI of your fonts in Unreal Engine in **Project Settings** > **Engine** > **User Interface**. The settings for font resolution are located in the **UMG Fonts** section.



Engine - User Interface

User Interface settings that control Slate and UMG.

These settings are saved in DefaultEngine.ini, which is currently writable.

UMG Fonts

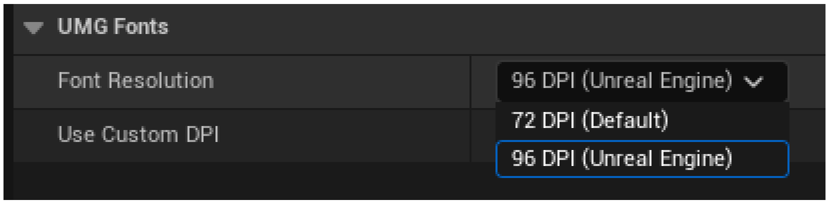
Font Resolution: 96 DPI (Unreal Engine)

Use Custom DPI: □

The **Font Resolution** setting changes the DPI used to calculate the size of all fonts in your project. The **Use Custom DPI** setting makes it possible to set a value other than the default values.

Configure Standard DPIs

When **Use Custom DPI** is unchecked, use the **Font Resolution** dropdown to choose either 96 DPI (Unreal Engine) or 72 DPI (Default).



UMG Fonts

Font Resolution: 96 DPI (Unreal Engine)

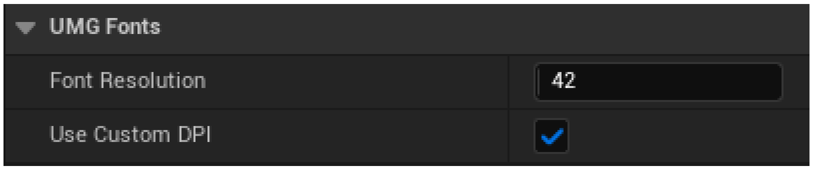
Use Custom DPI

96 DPI (Unreal Engine) is the legacy setting, as past versions of UE used 96 DPI to calculate font sizes. In the current version of UE, this is used as the default setting to facilitate older projects.

72 DPI (Default) is the web design industry standard for calculating font sizes for user interfaces. We recommend this setting for new projects, as it is used by popular UI design software such as Figma. This will be the default value in future versions of Unreal Engine. This will be the default DPI in future versions of UE.

Configure Custom DPIs

When **Use Custom DPI** is checked, the Font Resolution dropdown is replaced with an integer value slider. Use this to set your Font Resolution to any integer DPI between 1 and 1000.



UMG Fonts

Font Resolution: 42

Use Custom DPI: ☑

Using non-standard font resolutions risks creating rounding errors when calculating the size of your text. See Font Resolution and Font Size below for more information.

Font Resolution and Font Size

The Font Resolution setting only changes the way UMG interprets Font Sizes, not the actual internal DPI that is used to calculate them. No matter what your Font Resolution is set to, Unreal Engine converts font sizes to a value appropriate for 96 DPI, then internally uses 96

DPI to calculate the final display of characters. Therefore, the actual formula for calculating onscreen size is as follows:

```
Pixel Size = ( Point Size of Font * Screen Resolution ) / (Font Resolution in
```

Copy full snippet

For example, if you set Font Resolution to 72 DPI and set a text element to 12 point font, UE converts that text's font size value to 16, and then displays the text at 96 DPI. The result is the expected text size for a 12-point font in a 72 DPI application.

This conversion makes it possible to maintain the onscreen size of text elements throughout your project even when you change the global Font Resolution, and ensures backwards compatibility with older projects that have not yet adopted the 72 DPI standard. This also facilitates use of fractional font sizes, such as 12.5 or 12.2-point fonts.

This approach can create rounding errors which result in unexpected font sizes. We recommend that you determine your team's preferred DPI early.

Font Size Best Practices

When managing text in your projects, you should use the smallest number of different font sizes possible.

To optimize displaying text, UE creates font atlas assets that contain only the glyphs your project uses, rasterizing them and adding them to the font atlas on an as-needed basis. For example, if the only string in your project is the word "Scaling" in Roboto 12 font, only the exact 7 letters in that string will take space in the font atlas.

However, UE creates an atlas of the glyphs for every different size of every different font your project uses. In other words, if you display "Scaling" in Roboto 12, Roboto 18, Roboto 72, Arial 18, Arial 24, and Arial 36, UE creates six font atlases that each contain the needed glyphs for the word "Scaling" rather than re-using the same glyphs for each size. Furthermore, if you apply any scaling to widgets that contain text, UE's text engine rasterizes them at the closest size after applying the transform. In other words, if you scale the whole screen down by a factor of 50%, a string originally set to use Roboto 12 actually rasterizes with a size of 6, adding another atlas for UE to manage. Any of these atlases may be loaded in memory at any given time depending on how your UI is constructed and displayed.

Therefore, minimizing the number of different sizes you use across your project ensures that you don't unnecessarily bloat your project's font assets, and also helps enforce standardization of fonts across your project. The more utility you get out of a single size when designing your UI, the more easily glyphs can be reused, and the fewer atlases that need to be loaded.