

AF Content Driven UI Sizing

User Guide | Unity uGUI

Overview

AF Content Driven UI Sizing is a Unity uGUI component that determines the dimensions required by an existing UI element based on supported content. It can adjust an undriven RectTransform directly or provide preferred dimensions to Unity's layout system.

It is not a complete responsive-screen framework. Its purpose is content-driven sizing of the specific element to which the component is attached.

Requirements and installation

Unity uGUI (Canvas-based UI) and TextMesh Pro are required. TextMesh Pro is a script dependency even when an individual example contains only images.

Import the package into your Unity project. Open the included demonstration scene to examine text-only, mixed-content, nested-layout, RawImage and Input Field examples.

The runtime component is `AFContentResponsive` in the `ArcanaFutura.UI.ContentResponsive` namespace. Add it to a UI GameObject with a RectTransform.

Quick start

1. Select an existing content-bearing UI element (for example, a button or panel).
2. Add the AF Content Responsive component to that element, not indiscriminately to the entire UI hierarchy.
3. Enable Responsive Width, Responsive Height, or both. Both are disabled by default to preserve the existing authored dimensions until explicitly enabled.
4. Configure child Layout Groups, padding, spacing, and Layout Elements as appropriate.
5. Optionally enable minimum or maximum constraints for either axis.
6. Change the text or other supported content and inspect the requested and actual dimensions in the Inspector.

Responsive dimensions and constraints

Responsive Width: allows content measurement to determine the requested width.

Responsive Height: allows content measurement to determine the requested height.

When an axis is not responsive, its authored size is preserved rather than calculated from content.

Optional minimum and maximum width/height constraints limit the requested result on each axis. Ensure a chosen minimum does not exceed the corresponding maximum.

When Unity's layout system drives the RectTransform, the component provides preferred dimensions rather than forcibly overriding the parent layout.

Supported content

TextMesh Pro text: measures supported TMP text content.

Unity UI Image and RawImage: measures the configured displayed UI size, not the underlying sprite or texture's native pixel resolution.

Mixed content: text and images may participate in the same content-bearing element.

Horizontal, Vertical, and Grid Layout Groups: content measurements account for supported layout arrangements and settings, including applicable padding and spacing.

Nested UI: components can be attached to individual content-bearing elements at different hierarchy levels.

Standard Unity UI InputField and TMP_InputField: special handling is provided for text and placeholder sizing.

Images and Layout Elements

CRUIE measures images; it does not resize or scale sprites, textures, or text to fit a target size.

When an Image or RawImage is managed by a Layout Group, provide appropriate preferred dimensions, typically with a Layout Element. Configure both preferred width and preferred height when preserving a particular displayed proportion is important.

An element with Layout Element → Ignore Layout is excluded from the corresponding layout measurement. The custom Inspector can show such objects as ignored to make this behavior explicit.

Input Fields

Attach CRUIE to the Input Field's content-bearing root element, not separately to its Text Area or placeholder.

Starting Size Source offers Placeholder (default) or Text for the initial measurement.

The component does not resize the field on every keystroke. After editing finishes, it measures the committed text. When cleared, placeholder-based sizing can resume.

Keep the Input Field's authored child offsets and margins intact. CRUIE does not need to rewrite those values. If a result seems incorrect, verify the Text Area offsets and the text component's margins before changing code.

Inspector diagnostics

Measurements (Read Only) displays Original Width/Height, Requested Width/Height, and Actual Width/Height. Original is the captured authored baseline; requested is the calculated target; actual is the RectTransform's resulting size.

Detected Content (Read Only) lists recognized content in hierarchy order, including object references and available current/cached dimensions.

Items excluded by Ignore Layout can appear grayed out with an "Ignored by Layout" label. They are visible for diagnosis but excluded from sizing.

Diagnostics help explain what is measured; they are not additional sizing controls.

Scope and integration

Works with Unity's Horizontal, Vertical, and Grid Layout Groups, accounting for their configured layout settings.

Respects dimensions controlled by other Unity layout components, integrating with the existing layout system rather than overriding it.

Respects Ignore Layout settings and excludes those elements from measurement.

Does not automatically adapt layouts to screen resolutions, device orientations, or breakpoints.

Does not automatically scale text, sprites, or textures to fit available space.

For UI controls with multiple independently sized elements, attach the component to the specific content-bearing elements that require adaptive sizing.

AF Wrap Layout Group can complement this component but is a separate, optional asset.

Troubleshooting

Element does not resize: check that the relevant Responsive Width/Height option is enabled, and whether a parent Layout Group is driving the dimension.

Image appears collapsed or distorted: verify its Layout Element preferred width and height and the parent Layout Group configuration.

Unexpected size or padding: inspect Layout Group padding/spacing, child RectTransform offsets, text margins, and min/max constraints.

Missing item in Detected Content: check whether it is inactive, ignored by layout, or not a supported measured content type. An ignored element may still be listed for diagnosis.

Input Field sizing seems incorrect: check Starting Size Source, complete an edit before judging the final size, and inspect Text Area offsets and margins.

Competing layout components can cause repeated sizing requests. Review conflicting settings and consider appropriate size constraints. Do not rely on the component to resolve arbitrary circular layout dependencies.

Demo scene

The included demo illustrates plain TMP text, text with padding, horizontal and vertical image/text layouts, a RawImage with text, nested layouts, and an Input Field.

The demo uses ordinary Unity UI elements. It is intended to explain sizing behavior, not to provide a complete visual UI kit.

Technical reference

Asset: AF Content Driven UI Sizing

Component: AFContentResponsive

Namespace: ArcanaFutura.UI.ContentResponsive

UI framework: Unity uGUI

Text support: TextMesh Pro and standard UI Input Field text

Layout interface: ILayoutElement

Rendering: No render-pipeline-specific behavior

Exact minimum supported Unity editor version: confirm against the final package before publishing.

Contact

Have a question, need help, or found a bug? Please use our online contact form and we'll get back to you as soon as possible

.

Website: <https://arcanafutura.eu/>

Support: <https://arcanafutura.eu/asset-support>

License and Disclaimer

AF Content Driven UI Sizing is provided under the applicable Unity Asset Store license. It may be used and modified as part of your own commercial or non-commercial projects, subject to the terms of that license.

You may modify the supplied source code for use in your projects. Modification does not grant the right to redistribute or resell AF Content Driven UI Sizing itself. The asset, its source code, or modified versions of the asset may not be redistributed, resold, sublicensed or otherwise made available as a standalone asset or development tool.

AF Content Driven UI Sizing is provided "as is", without warranties of any kind. Arcana Futura is not liable for losses or damages resulting from its use.

AF Content Driven UI Sizing is provided under the Unity Asset Store End User License Agreement (EULA). <https://unity.com/legal/as-terms>

Arcana Futura