image/svg+xml Manual Scripting API Changelog License

Search

docs.unity3d.com

Enter here to filter...

Scripting API / UnityEngine.UI / Text

Class Text

+ Inheritance

□ Object

□ ...

Graphic

MaskableGraphic

Text

...

+ Inherited Members

MaskableGraphic.m_ShouldRecalculateStencil

MaskableGraphic.m_MaskMaterial

MaskableGraphic.m_IncludeForMasking

MaskableGraphic.onCullStateChanged

MaskableGraphic.maskable

...

Namespace: UnityEngine.UI

Syntax

```
[RequireComponent(typeof(CanvasRenderer))]
[AddComponentMenu("UI/Text", 10)]
public class Text : MaskableGraphic, ICanvasElement, IClippable, IMaskable, IMaterialModifier, ILayoutElement
```

Constructors

Text()

Declaration

```
protected Text()
```

Fields

m_DisableFontTextureRebuiltCallback

Declaration

```
[NonSerialized]
protected bool m_DisableFontTextureRebuiltCallback
```

Field Value

Type	Description
Boolean	

m_Text

Declaration

```
[TextArea(3, 10)]
[SerializeField]
protected string m_Text
```

Field Value

Type	Description
String	

s_DefaultText

Declaration

```
protected static Material s_DefaultText
```

Field Value

Type	Description
Material	

Properties

alignByGeometry

Use the extents of glyph geometry to perform horizontal alignment rather than glyph metrics.

Declaration

```
public bool alignByGeometry { get; set; }
```

Property Value

Type	Description
Boolean	

Remarks

This can result in better fitting left and right alignment, but may result in incorrect positioning when attempting to overlay multiple fonts (such as a specialized outline font) on top of each other.

alignment

The positioning of the text reliative to its [[RectTransform]].

Declaration

```
public TextAnchor alignment { get; set; }
```

Property Value

Type	Description
TextAnchor	

Remarks

This is the positioning of the Text relative to its RectTransform. You can alter this via script or in the Inspector of a Text component using the buttons in the Alignment section.

Examples

```
//Create a Text GameObject by going to __Create__>__UI__>__Text__. Attach this script to the GameObject to see it working.

using UnityEngine;
using UnityEngine.UI;

public class UITextAlignment : MonoBehaviour
{
    Text m_Text;

    void Start()
    {
        //Fetch the Text Component
        m_Text = GetComponent<Text>();
        //Switch the Text alignment to the middle
        m_Text.alignment = TextAnchor.MiddleCenter;
    }

//This is a legacy function used for an instant demonstration. See the <a href="https://unity3d.com/learn/tutorials/s/user-interface-ui">UI Tutorials pages </a> and [[wiki:UISystem|UI Section]] of the manual for more information on creating your own buttons etc.
    void OnGUI()
    {
        //Press this Button to change the Text alignment to the lower right
        if (GUI.Button(new Rect(0, 0, 100, 40), "Lower Right"))
        {
            m_Text.alignment = TextAnchor.LowerRight;
        }

        //Press this Button to change the Text alignment to the upper left
        if (GUI.Button(new Rect(150, 0, 100, 40), "Upper Left"))
        {
            m_Text.alignment = TextAnchor.UpperLeft;
        }
    }
}
```

cachedTextGenerator

The cached TextGenerator used when generating visible Text.

Declaration

```
public TextGenerator cachedTextGenerator { get; }
```

Property Value

Type	Description
TextGenerator	

cachedTextGeneratorForLayout

The cached TextGenerator used when determine Layout

Declaration

```
public TextGenerator cachedTextGeneratorForLayout { get;
 }
```

Property Value

Type	Description
TextGenerator	

flexibleHeight

Declaration

```
public virtual float flexibleHeight { get; }
```

Property Value

Type	Description
Single	

Implements

ILayoutElement.flexibleHeight

flexibleWidth

Declaration

```
public virtual float flexibleWidth { get; }
```

Property Value

Type	Description
Single	

Implements

ILayoutElement.flexibleWidth

font

The Font used by the text.

Declaration

```
public Font font { get; set; }
```

Property Value

Type	Description
Font	

Remarks

This is the font used by the Text component. Use it to alter or return the font from the Text. There are many free fonts available online.

Examples

```
//Create a new Text GameObject by going to Create>UI>Text in the Editor. Attach this script to the Text GameObject. Then, choose or click and drag your own font into the Font section in the Inspector window.

using UnityEngine;
using UnityEngine.UI;

public class TextFontExample : MonoBehaviour
{
    Text m_Text;
    //Attach your own Font in the Inspector
    public Font m_Font;

    void Start()
    {
        //Fetch the Text component from the GameObject
        m_Text = GetComponent<Text>();
    }

    void Update()
    {
        if (Input.GetKey(KeyCode.Space))
        {
            //Change the Text Font to the Font attached in the Inspector
            m_Text.font = m_Font;
            //Change the Text to the message below
            m_Text.text = "My Font Changed!";
        }
    }
}
```

fontSize

The size that the Font should render at. Unit of measure is Points.

Declaration

```
public int fontSize { get; set; }
```

Property Value

Type	Description

Int32	

Remarks

This is the size of the Font of the Text. Use this to fetch or change the size of the Font. When changing the Font size, remember to take into account the RectTransform of the Text. Larger Font sizes or messages may not fit in certain rectangle sizes and do not show in the Scene. Note: Point size is not consistent from one font to another.

Examples

```
//For this script to work, create a new Text GameObject by going to Create>U>Text. Attach the script to the Text GameObject. Make sure the GameObject has a RectTransform component.

using UnityEngine;
using UnityEngine.UI;

public class Example : MonoBehaviour
{
    Text m_Text;
    RectTransform m_RectTransform;

    void Start()
    {
        //Fetch the Text and RectTransform components from the GameObject
        m_Text = GetComponent<Text>();
        m_RectTransform = GetComponent<RectTransform>();
    }

    void Update()
    {
        //Press the space key to change the Font size
        if (Input.GetKey(KeyCode.Space))
        {
            changeFontSize();
        }
    }

    void changeFontSize()
    {
        //Change the Font Size to 16
        m_Text.fontSize = 30;

        //Change the RectTransform size to allow larger fonts and sentences
        m_RectTransform.sizeDelta = new Vector2(m_Text.fontSize * 10, 100);

        //Change the m_Text text to the message below
        m_Text.text = "I changed my Font size!";
    }
}
```

fontStyle

Font style used by the Text's text.

Declaration

```
public FontStyle fontStyle { get; set; }
```

Property Value

Type	Description
FontStyle	

horizontalOverflow

Horizontal overflow mode.

Declaration

```
public HorizontalWrapMode horizontalOverflow { get; set;
 }
```

Property Value

Type	Description
HorizontalWrapMode	

Remarks

When set to HorizontalWrapMode.Overflow, text can exceed the horizontal boundaries of the Text graphic. When set to HorizontalWrapMode.Wrap, text will be word-wrapped to fit within the boundaries.

layoutPriority

Declaration

```
public virtual int layoutPriority { get; }
```

Property Value

Type	Description
Int32	

Implements

ILayoutElement.layoutPriority

lineSpacing

Line spacing, specified as a factor of font line height. A value of 1 will produce normal line spacing.

Declaration

```
public float lineSpacing { get; set; }
```

Property Value

Type	Description
Single	

mainTexture

Text's texture comes from the font.

Declaration

```
public override Texture mainTexture { get; }
```

Property Value

Type	Description
Texture	

Overrides

Graphic.mainTexture

minHeight

Declaration

```
public virtual float minHeight { get; }
```

Property Value

Type	Description
Single	

Implements

ILayoutElement.minHeight

minWidth

Declaration

```
public virtual float minWidth { get; }
```

Property Value

Type	Description
Single	

Implements

ILayoutElement.minWidth

pixelsPerUnit

Provides information about how fonts are scale to the screen.

Declaration

```
public float pixelsPerUnit { get; }
```

Property Value

Type	Description
Single	

Remarks

For dynamic fonts, the value is equivalent to the scale factor of the canvas. For non-dynamic fonts, the value is calculated from the requested text size and the size from the font.

preferredHeight

Declaration

```
public virtual float preferredHeight { get; }
```

Property Value

Type	Description
Single	

Implements

ILayoutElement.preferredHeight

preferredWidth

Declaration

```
public virtual float preferredWidth { get; }
```

Property Value

Type	Description
Single	

Implements

ILayoutElement.preferredWidth

resizeTextForBestFit

Should the text be allowed to auto resized.

Declaration

```
public bool resizeTextForBestFit { get; set; }
```

Property Value

Type	Description
Boolean	

resizeTextMaxSize

The maximum size the text is allowed to be. 1 = infinitely large.

Declaration

```
public int resizeTextMaxSize { get; set; }
```

Property Value

Type	Description
Int32	

resizeTextMinSize

The minimum size the text is allowed to be.

Declaration

```
public int resizeTextMinSize { get; set; }
```

Property Value

Type	Description
Int32	

supportRichText

Whether this Text will support rich text.

Declaration

```
public bool supportRichText { get; set; }
```

Property Value

Type	Description
Boolean	

text

Text that's being displayed by the Text.

Declaration

```
public virtual string text { get; set; }
```

Property Value

Type	Description
String	

Remarks

This is the string value of a Text component. Use this to read or edit the message displayed in Text.

Examples

```
using UnityEngine;
using UnityEngine.UI;

public class Example : MonoBehaviour
{
    public Text m_MyText;

    void Start()
    {
        //Text sets your text to say this message
        m_MyText.text = "This is my text";
    }

    void Update()
    {
        //Press the space key to change the Text message
        if (Input.GetKey(KeyCode.Space))
        {
            m_MyText.text = "My text has now changed.";
        }
    }
}
```

verticalOverflow

Vertical overflow mode.

Declaration

```
public VerticalWrapMode verticalOverflow { get; set; }
```

Property Value

Type	Description
VerticalWrapMode	

Methods

CalculateLayoutInputHorizontal()

Declaration

```
public virtual void CalculateLayoutInputHorizontal()
```

Implements

ILayoutElement.CalculateLayoutInputHorizontal()

CalculateLayoutInputVertical()

Declaration

```
public virtual void CalculateLayoutInputVertical()
```

Implements

ILayoutElement.CalculateLayoutInputVertical()

FontTextureChanged()

Called by the FontUpdateTracker when the texture associated with a font is modified.

Declaration

```
public void FontTextureChanged()
```

GetGenerationSettings(Vector2)

Convenience function to populate the generation setting for the text.

Declaration

```
public TextGenerationSettings GetGenerationSettings(Vect
or2 extents)
```

Parameters

Type	Name	Description
Vector2	*extents*	The extents the text can draw in.

Returns

Type	Description
TextGenerationSettings	Generated settings.

GetTextAnchorPivot(TextAnchor)

Convenience function to determine the vector offset of the anchor.

Declaration

```
public static Vector2 GetTextAnchorPivot(TextAnchor anch
or)
```

Parameters

Type	Name	Description
TextAnchor	*anchor*	

Returns

Type	Description
Vector2	

OnDisable()

Declaration

```
protected override void OnDisable()
```

Overrides

MaskableGraphic.OnDisable()

OnEnable()

Declaration

```
protected override void OnEnable()
```

Overrides

MaskableGraphic.OnEnable()

OnPopulateMesh(VertexHelper)

Declaration

```
protected override void OnPopulateMesh(VertexHelper toFi
ll)
```

Parameters

Type	Name	Description
VertexHelper	*toFill*	

Overrides

Graphic.OnPopulateMesh(VertexHelper)

UpdateGeometry()

Declaration

```
protected override void UpdateGeometry()
```

Overrides

Graphic.UpdateGeometry()

Extension Methods

UIBehaviourExtensions.InvokeOnEnable(UIBehaviour)
UIBehaviourExtensions.InvokeOnDisable(UIBehaviour)
UIBehaviourExtensions.InvokeAwake(UIBehaviour)
UIBehaviourExtensions.InvokeRebuild(UIBehaviour, CanvasUpdate)
UIBehaviourExtensions.InvokeLateUpdate(UIBehaviour)
UIBehaviourExtensions.InvokeUpdate(UIBehaviour)
UIBehaviourExtensions.InvokeOnRectTransformDimensionsChange(UIBehaviour)
UIBehaviourExtensions.InvokeOnCanvasGroupChanged(UIBehaviour)
UIBehaviourExtensions.InvokeOnDidApplyAnimationProperties(UIBehaviour)
GraphicExtension.InvokeOnPopulateMesh(Graphic, VertexHelper)