

TEXT SCRIPT INTERFACE

View online: <https://www.construct.net/en/make-games/manuals/construct-3/scripting/scripting-reference/plugin-interfaces/text>

The `ITextInstance` interface derives from `IWorldInstance` to add APIs specific to the `Text` plugin.

Text APIs

text

The string currently displayed by the Text object.

typewriterText(str, duration)

Set the text over time by starting with an empty string and gradually adding characters until the full text of `str` is written out, over a `duration` specified in seconds. Note modifying the `text` property while text is being written out will cancel the effect.

typewriterFinish()

If text is being written out with the `typewriterText()` method, force it to finish immediately.

fontColor

An array with 3 elements specifying the red, green and blue color of the text, with color values as floats in the 0-1 range.

Try not to confuse this with the `IWorldInstance` property `colorRgb`, which applies a color tint to the overall appearance of the object.

fontFace

A string specifying the font used to display the text, e.g. "Arial".

isBold

isItalic

Booleans indicating whether the font face is displayed with bold or italic styles.

sizePt

The size of the displayed text, in points (pt).

lineHeight

The extra space in pixels to add vertically between lines. 0 is the default size, negative values make lines closer together, and positive values space lines out further apart.

horizontalAlign

A string specifying the horizontal alignment of the text within the object bounding box, which must be one of `"left"`, `"center"` or `"right"`.

verticalAlign

A string specifying the vertical alignment of the text within the object bounding box, which must be one of `"top"`, `"center"` or `"bottom"`.

readAloud

A boolean indicating whether the contents of the text object will be read aloud by screen reader software. See the *Read aloud* property of the Text object for more details.

textDirection

Set or get a string of either `"ltr"` (left-to-right) or `"rtl"` (right-to-left) specifying the direction of the text flow. See the *Text direction* property of the Text object for more details.

wordWrapMode

A string specifying the way to wrap text when it reaches the end of a line. This can be either `"word"` to wrap entire space-separated words, `"character"` to wrap at any character, or `"cjk"` to wrap at any character but with special handling for Chinese, Japanese and Korean punctuation.

setFixedResolutionMode(scale)

setAutoResolutionMode()

By default Text objects use *Automatic* resolution mode, which means the resolution of the text adjusts according to the 2D display scale. This produces the best quality display, but only works with 2D display, and can also cause the text to constantly re-render when being smoothly scaled, which can sometimes have a significant performance overhead. *Fixed* resolution mode causes the text to render at a fixed resolution according to a provided scale factor, and ignore the display scale. This usually results in a reduced display quality but better performance.

textWidth**textHeight****getTextSize()**

Read-only values indicating the size of the actual text content within the text object's rectangle. The method allows getting both values at the same time.

hasTagAtPosition(tag, x, y)

Return a boolean indicating if there is text with a specific tag at the given position (case insensitive). For example if the text has the BBcode `Hello [tag=mytag]world[/tag]`, then testing if the tag "mytag" is at a given position return `true` if that position is over just the part of the text that says "world", else `false`.

getTagAtPosition(x, y)

Look up the tag for a part of the text at a given position and return the tag if any, else return an empty string if no tag is specified. For example if the text has the BBcode `Hello [tag=mytag]world[/tag]`, then the tag at a position over the word "world" is "mytag", and the tag at a position over the word "Hello" is "".

getTagCount(tag)**getTagPositionAndSize(tag, index)**

Get the number of fragments, and the size and position of each fragment by its zero-based index, for a given tag. Note that a single tag may be broken in to multiple fragments - see the section *Tagged range fragmentation* in the [Text object manual entry](#) for more details. The `getTagPositionAndSize()` method returns the position and size as an object with the properties `{x, y, width, height}`.

changeIconSet(objectClass)

Changes the *Icon set* property, replacing the Sprite used for BBcode icons to the one specified by the given `IObjectClass` (which must be from a Sprite object). This can be used to change the set of icons displayed by the Text object. Note if the new Sprite object does not have the same number of animation frames, or the same animation frame tags, then some icons may disappear.

async getAsString()

Converts the contents of the Text object, including any icons, in to a string of HTML. This process is asynchronous so the method must be awaited. It resolves with a string of HTML code. The result is cached (and updates the `AsHTML` expression) so repeat calls will resolve with the same HTML string, until the Text object contents is modified, after which the next call will regenerate the HTML string again.