

## Raymon RTL 5.0

RTL plugin is a very powerful and fast text converter for generating **Arabic, Hebrew, Farsi, Urdu, Yiddish, Kurdish and Pashto**<sup>1</sup> right to left text. The result from this plugin can be used in Unity3D, Adobe Photoshop, Flash, 3Ds Max, Maya and other designing software or game engines. However the main focus of the plugin is to produce well-format right to left texts for Unity3D game engine. (If you would use this plugin in your web services, web application back-end or mobile applications please contact us [raymongame@gmail.com](mailto:raymongame@gmail.com))



## Features

RTL is awesome because of:

- 1- Supporting most right to left languages each separately or mixed with English or any left to right languages.
- 2- Supporting different number formats (English, Arabic and Farsi) like ٤٥٦ and ٤٥٦.
- 3- Supporting simple math equation conversion. ٤+٥=٩ not a problem anymore.
- 4- Supporting combinations of number, punctuation and texts<sup>2</sup> in one call.

<sup>1</sup> RTL doesn't support yet these characters in Pashto language: څ، ځ، ښ، ړ، د، ت، گ، ن، ر.

<sup>2</sup> Not all complicated combinations of punctuation, parentheses, RTL and LTR altogether are supported yet.

### Mixed languages support

Unity: Hello world (السلام دالن)

RTL: سلام دنيا Hello world

### Mathematic

Unity:  $x+x^2=x-14$   
 $x+x^2-x=14$   
 $x^2=14$   
 $x=7$

RTL: مساله  
 $x+x^2=x-14$   
 $x+x^2-x=14$   
 $x^2=14$   
 $x=7$

### Different number format conversion

Unity: Eng= 0123456789  
Eng= ٠١٢٣٤٥٦٧٨٩  
عربي = ٠١٢٣٤٥٦٧٨٩  
فارسي = ٠١٢٣٤٥٦٧٨٩

RTL: Eng= 0123456789  
Eng= ٠١٢٣٤٥٦٧٨٩  
عربي = ٠١٢٣٤٥٦٧٨٩  
فارسي = ٠١٢٣٤٥٦٧٨٩

### Mix of punctuation, text and numbers

Unity: Answer=۲\*(۵)  
جواب: 2+5=7  
Sample [2+5=7], ۲\*(۵+۱)=۱۲

RTL: Answer=۲\*(۵)  
جواب: 2+5=7  
Sample [2+5=7], ۲\*(۵+۱)=۱۲

## How to use

RTL Converter is a class library (.dll) you can add to your game projects. This class library contains two main public static methods `Convert()` and `ConvertWordWrap()` returning the converted RTL text. In addition, there are a few helper methods available such as `IsRTL()` and `IsRTLNumber()`. Each method has several parameters to let developers adjust the conversion. You can setup number formats and word-wrap parameters when you call RTL methods.

### Main Methods:

#### - `Convert(...)`:

This method converts and returns the well-format right to left text.

#### - `ConvertWordWrap(...)`:

This method converts the input text to proper word wrapped right to left text to be used in Unity3D.

Note: There are some specific structures used by RTL like `UnityEngine.GUIStyle` which is only available in Unity3D game engine.

### Parameters:

Here is a list of parameters used in mentioned methods:

#### - `originalText`:

The original text you would like to convert.

#### - `numberFormat`:

This parameter helps to adjust the output number format:

**Context (default):** Number format will be the same as it is in the original text.

**Arabic:** Force the numbers appear in Arabic format (٠١٢٣٤٥٦٧٨٩).

**Farsi:** Force the numbers appear in Farsi format (٠١٢٣٤٥٦٧٨٩).

**English:** Force the numbers appear in English (0123456789) format.

#### - `isLTRContext`:

You can set this parameter to true if the original context is in English (or any other left to right language) and it contains only a few right to left phrases. In other words if you need to fix the format for a few words in an English context you need to set this parameter to true and pass the whole text as input to RTL methods.

Note: If the input context is in a right to left and it has only a few English words or there's no left to right words at all in the text, this parameter should be set to false.

(Default value is `false`)

#### - `rtlWidth`:

Width of GUI element which will be used to show the converted text in game. This is useful because Unity game engine doesn't support word wrap for right to left languages. This parameter lets RTL override Unity's calculations and correct word wrapping.

#### - `rtlStyle`:

This parameter defines the GUI style which will be used to render the converted text. This is required when you would have converted text word wrapped.

The default value for this parameter is set to Unity's default `GUIStyle`.

## Example usage in general applications

```
string originalText = "السلام و عليك";  
string convertedText = RTL.Convert(originalText);
```

By calling this simple function, a well-formatted right to left Arabic text will be returned. It's now ready to be rendered in Unity game engine or in any other applications.

## Example usage in Unity3D

```
GUILayout.TextField(RTL.Convert(originalText),GUILayout.Height(150),GUILayout.Width(250));
```

This example uses Unity's `GUILayout` to show converted right to left text.

It's always a good idea not to use the RTL plugin or any computational function directly in `OnGUI()`. Generally, `Update()` method is always a better choice. You can store the result of the function in a variable, then use this variable in `OnGUI()` function.

## More examples:

- 1- `//Get RTL text with more options and adjustments (no word-wrapping - Arabic text)`  
`convertedText = RTL.Convert(InputText, RTL.NumberFormat.Context, false);`
  
- 2- `//Get RTL text (with word-wrapping - default GUI style)`  
`convertedText = RTL.ConvertWordWrap(InputText, 200, null);`
  
- 3- `//Get RTL text (word-wrapping enabled based on element width 300px and rtlStyle)`  
`convertedText = RTL.ConvertWordWrap(InputText, 300, skin.customStyles[0],`  
`RTL.NumberFormat.Arabic, false);`

If you have any questions please contact us: [[raymongame@gmail.com](mailto:raymongame@gmail.com)]

Good Luck,

RaymonGame