



LOCALIZATION (MULTI-LANGUAGE)



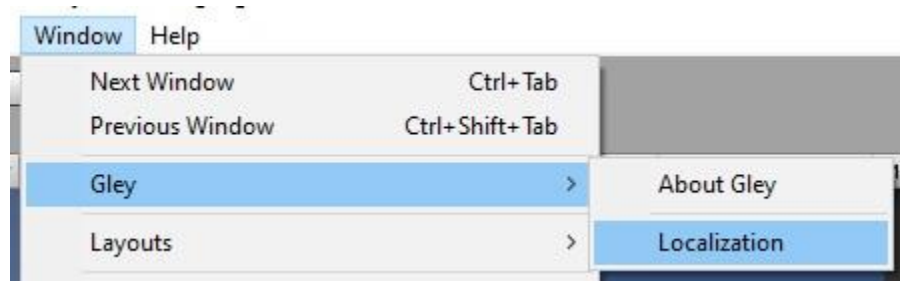
1. WHY DO YOU NEED TO USE THIS PLUGIN

- Make your app international by translating it into multiple languages fast and easy.
- Supports all languages supported by Unity. [Complete list available here](#)
- All settings are done inside a custom Settings Window.
- Automatically detects the device language and if no translation is available in that language the default language will be used.
- Saves and Loads the last language used.
- Auto Translation implemented inside Settings Window. Automatically translations can be done by a single click inside asset.
- Import/Export translations to .CSV files
- Works with Unity UI, Text Mesh Pro and nGUI
- Translations can be done automatically by adding a component to your text or by calling a single line of code
- Full example scenes included
- All code is available and fully commented
- Works with Unity 2017.1 and above with Free, Plus or Pro license.
- Works on all supported Unity platforms.



2. SETUP GUIDE

- Import **Gley Localization Plugin** into Unity.
- Go to **Window->Gley->Localization** to open the Settings Window.



- Settings Window will open





Enable external tool support



If you are using Text Mesh Pro or nGUI in your game enable the corresponding tool by selecting it and hit **Save**



Add supported languages for app

Active Languages:

Hide Languages

English	Remove
French	Remove
Spanish	Remove

Default Language: English

New Language: Afrikaans

Add

- Select new language you want to add from dropdown and click **Add**
- To remove a language click **Remove**
- **Default Language** - the language used if app has no support for user preferred language
 - Ex: if your user has his device in German and your app is not translated in german, the default language will be used



Add app texts

Game Texts: Default language English

▶ 0. ExitID	Exit	Remove
▶ 1. NextID	Next	Remove
▶ 2. PlayID	Play	Remove
▶ 3. PrevID	Previous	Remove
▼ 4. SaveID		
ID 4	SaveID	Remove
English	Save Selected Language	Translate
French	Enregistrer la langue sélectionnée	Translate
Spanish		Translate

Add Word

- **Add Word** -> adds another text to translate. It can be a phrase or a single word.
 - **ID** -> a unique id for each word. It must start with a letter and it cannot contain special characters. It will be converted to an enum for easy access inside your app.
 - **Language** -> translation for each language.
- **Translate button** -> Automatically translates the text from the default language to the current language.
- **Remove button** -> removes the current word



Add app texts

- **New line**

- To add new line inside settings window just press **Enter**
- To add new line inside your .csv(excel) add **{newLine}**

Ex:

Hello

My name is ...

Should be written like this:

Hello{newLine} My name is ...

- **Comma**

- To add comma inside you .csv(excel) add **/,** or **{comma}**

Ex:

Hello, my name is ...

Should be written like this:

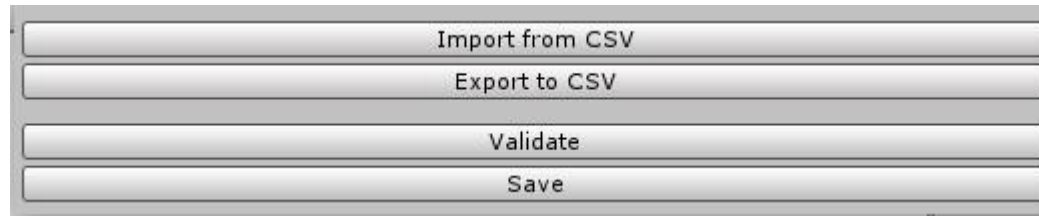
Hello/, my name is ...

Or

Hello{comma} my name is ...



Import/Export .csv



- **Import from CSV** -> Import texts from external csv file
- **Export to CSV** -> Export current added words to a csv file

ID	EnumID	English	French	Spanish	
ExitID	3	Exit	Sortie	Salida	
NextID	1	Next	Prochain	Próximo	
PlayID	0	Play	Jouer	Jugar	
PrevID	2	Previous	précédent	Anterior	
SaveID	4	Save Selected Language	Enregistrer la langue sélectionnée	Guardar idioma seleccionado	

The .csv must have the following columns:

ID - a unique id of your translation

EnumID - generated automatically by settings window

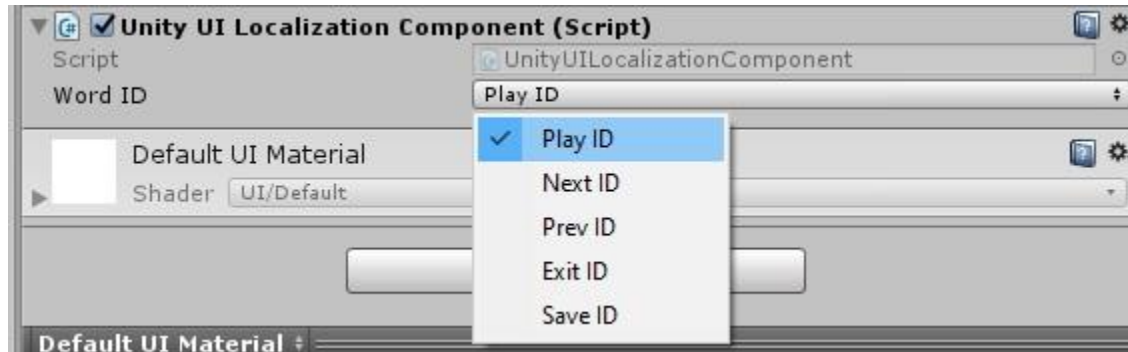
Languages - each language you want to localize your app has its own column

- **Validate** -> checks all your translations for empty words
- **Save** -> after all settings are made, press save to store them



3. USER GUIDE - Text Component

- The easiest way to translate a text is to add a Localization component on it.
- Supported localization components:
 - **UnityUILocalizationComponent** - Works with Unity UI text
 - **TMPProLocalizationComponent** - Works with Text Mesh Pro text
 - **TMPProUGUILocalizationComponent** - Works with Text Mesh Pro UI text
 - **NGUILocalizationComponent** - Works with nGUI Label



- From the WordID enum select the id you want to display.
- When this component is active it will display the text associated with the current language of the game.
- If the language changes the component will automatically update the text for the new language



3. USER GUIDE - Scripting

Get current selected language:

- **GleyLocalization.Manager.GetCurrentLanguage();**

Get translated text in current language:

- **GleyLocalization.Manager.GetText(WordIDs.NextID);**
- **GleyLocalization.Manager.GetText("NextID");**

Change supported language:

- **GleyLocalization.Manager.PreviousLanguage();**
- **GleyLocalization.Manager.NextLanguage();**

Save new language as current:

- **GleyLocalization.Manager.SetCurrentLanguage(SupportedLanguages.NewLanguage);**



4. PLAYMAKER SUPPORT

- **Supported Playmaker Actions:**
 - **Get Current Language**
 - **Get Localized Text**
 - **Next Language**
 - **Previous Language**
 - **Set Current Language**

The above Playmaker actions behavior is equivalent with corresponding methods from Section 3 - User Guide.

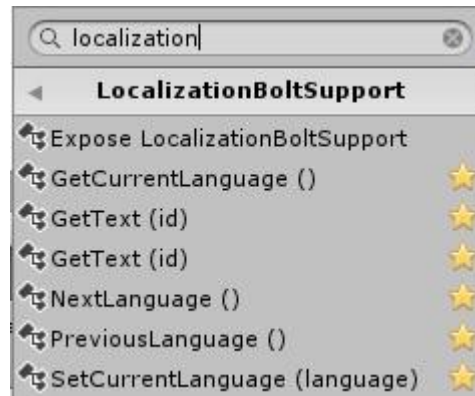


5. BOLT SUPPORT

- **Supported Bolt Actions:**
 - **Get Current Language**
 - **Get Localized Text**
 - **Next Language**
 - **Previous Language**
 - **Set Current Language**

The above Bolt actions behavior is equivalent with corresponding methods from Section 3 - User Guide.

The above Bolt actions are located inside **LocalizationBoltSupport** script as shown below:





6. EXAMPLE

You can find the example test scene here:

Assets/GleyPlugins/Localization/Example/SetLanguageExample.unity

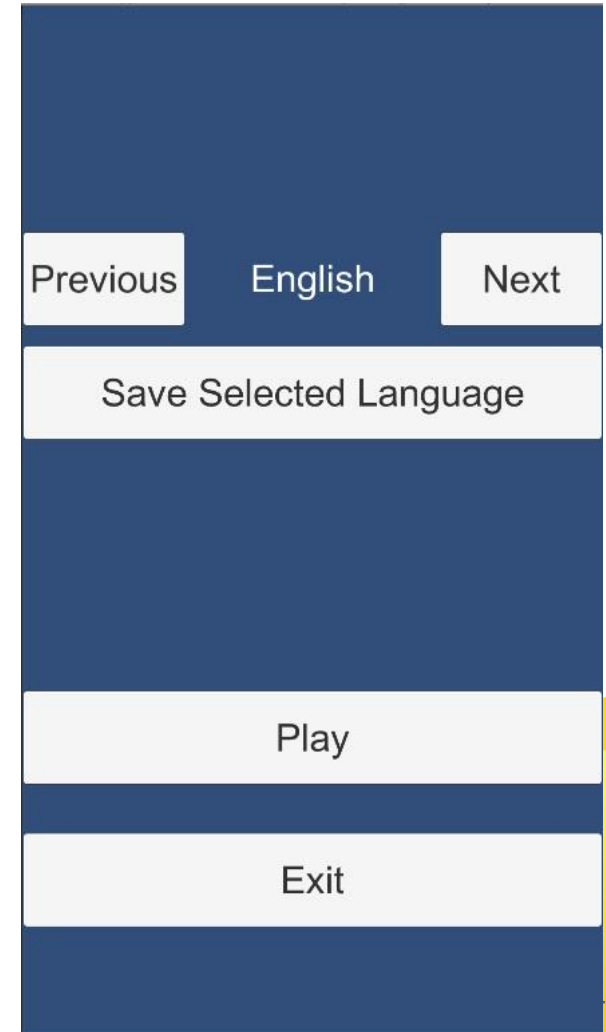
For the scene to look like in the picture you have to copy the translation file we used

From: Assets/GleyPlugins/Localization/Example/LocalizationFile

To: Assets/GleyPlugins/Localization/Resources/LocalizationFile

You can change the current language and see the text changing automatically.

To see the code we used open **SetLanguageExample.cs**





Version 1.2.0 / 2020