

# Language Localization

*for Unity Engine*

by **Matej Vanco**

*last update: 15/10/2022 [dd/mm/yyyy]*

# Content

— — —

[Localization Manager](#)

[Language Setup](#)

[API](#)

[FAQ](#)

[Support](#)

# Localization Manager

— — —

**Language Localization** is a simple tool to localize any application in text. The plugin contains user-friendly **Editor Window** that allows you to set up keys which will correspond to the localized text in any language.

To access the Localization Manager window, go to **Window/Localization Manager**.

First of all, you will need to create **Localization Manager file** that will store created keys.

If the **Localization Manager file exists**, you are able to start creating keys.

You are also free to create custom categories for much clear organisation.

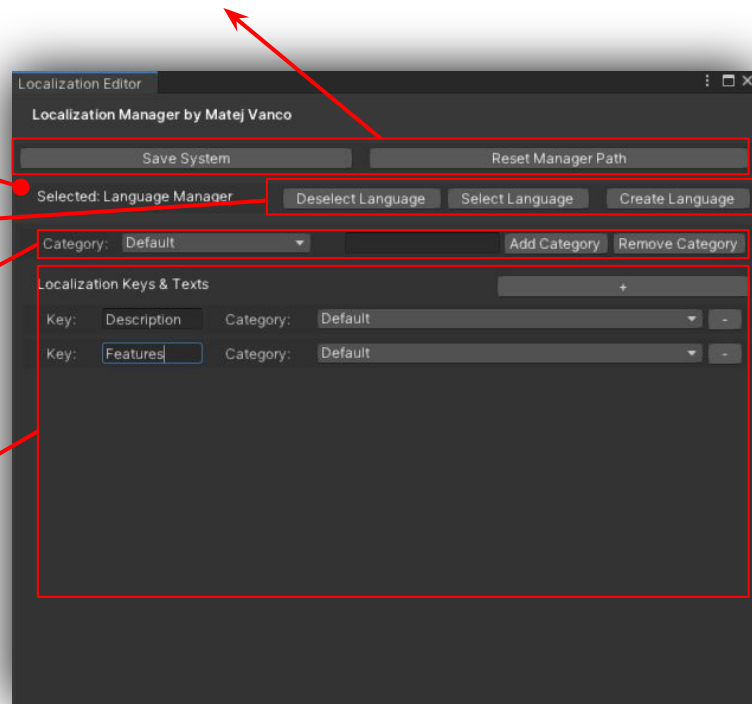
Save the whole system & reset Localization Manager path

Currently selected file

Select/ Deselect or Create language file

Create, Choose & Remove category

Created keys array;  
Key Name, Key Category



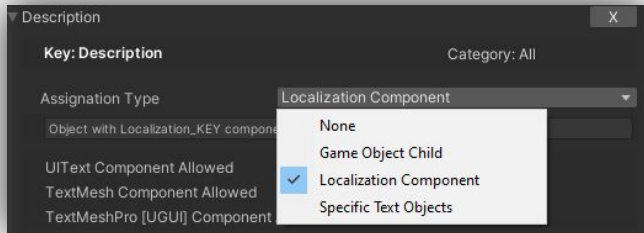
# Language Setup

If the Localization Manager file exists, you are able to create a language file.

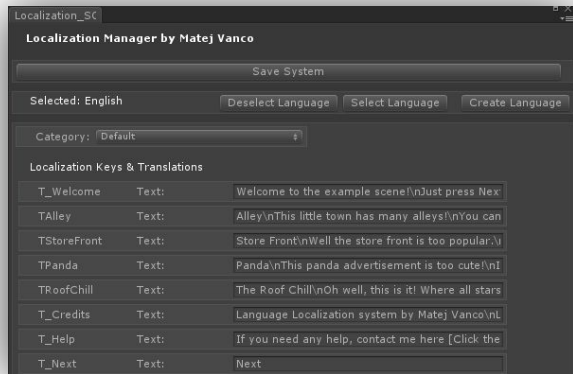
1. Create couple of keys that will represent text you want to translate.
2. Click **Create** or **Choose** language (if you have one)
3. Add text for each key

To check the results, go to your **Scene** and choose any object that will represent **Localization Source**. Add **Localization Source** component.

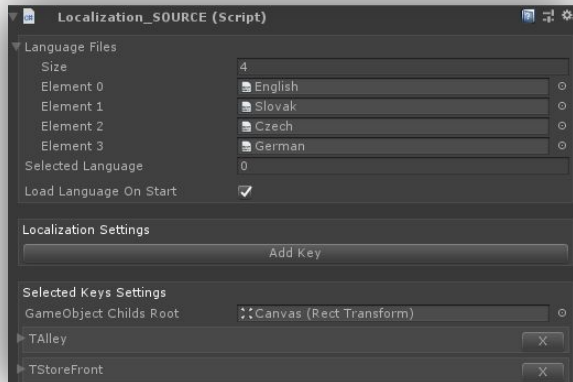
Add **language files** and you can easily choose any key.  
The key can be assigned to **Text-Mesh**, **Text-Mesh-Pro** or **UI-Text**.  
You can also **automate** the process of the text object assignment by choosing between 4 **Assignment Types**.



Language file selected in Localization Editor



Language Localization on object



# API

The Language Localization contains its own API. The API can be found in the **Localization\_SOURCE.cs**

**public void Lang\_RefreshKeyAssignations()**

- Refresh all resource objects by the selected options *(this is optional, otherwise it's called OnAwake automatically)*

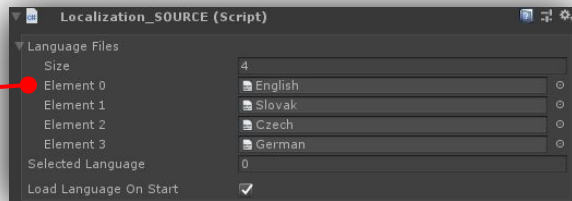
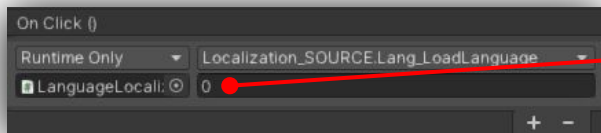
**public void Lang\_LoadLanguage(int languageIndex)**

- Load language database by the selected language index

**public void Lang\_ReturnText(string keyInput)**

- Return existing text by the specific key input in the currently selected language

You can call these methods via UI Buttons or other events.



You can also declare your own method and invoke from **Action-type Action\_LanguageLoaded()**.

# FAQ

---

1. **Is Language Localization compatible with mobile devices?**

- Yes, it is compatible with all operating systems & devices.

2. **How many languages am I able to create?**

- As many as you wish!

3. **Am I able to use diacritics?**

- Yes, you are able to use diacritics as well as interpunction.

4. **Am I able to edit translations in custom text editor?**

- Yes, you can edit translations wherever you wish. Fortunately the **LL** plugin does contain custom Language Editor Window which helps you to edit translations in very quick & effective way. Check the [Localization Manager](#) slide for more.

5. **Am I able to make new lines in translations?**

- Yes, you are able to make new lines, tabs, breaks and all these stuff without any issue.

6. **Am I able to assign localized text to my internal variable?**

- No, **LL** plugin doesn't contain such a feature, so it's necessary to do it by yourself.

Example: Create a key with translation that contains your-defined macro (eg. Kill count is #N#). In your code, call `Lang_ReturnText(YourKeyWithMacro)` and replace the #N# macro with any value you wish.

7. **Does the plugin contains any example scene?**

- Yes, **LL** plugin does contain a quick example scene with example language database.

8. **Does the plugin support TMP?**

- Yes, **LL** plugin does support TMP (Text Mesh Pro). UGUI version only!

— — —

# Thank you!

**Join to my official discord server to stay in touch!**

*[Just click the image below]*

Or contact me [here](#) (which may take a while)

