Mce Inc Documentation :

ISI Localization plugin

Mamadou Cisse

What is it ?

A generic solution to handle localization of any text, audio and image in any number of languages.

What's in the Package?

ISILocalization.cs: This is main class of this localization engine.

It is the class that provides access to all localized contents (string, sprites, audio) thanks to it static functions. This class is a singleton class, so you don't have to initialize it by script.

ISILocalizedLanguage.cs: This class represents a language with all localized contents in the language.

Language.cs: An enum that contains all supported languages. If you want to add a new language, you just need to create new value in this enumeration.

Flag.cs: Component to attach to a button in unity inspector. It allows to choose a language.

LocalizedString.cs: Component to attach to a GameObject in unity inspector. It allows to change dynamically the text of a label at runtime.

(the GameObject must have a UI.Text or TextMesh component).

LocalizedAudio.cs: Component to attach to a GameObject in unity inspector. It allows to play localized audio at runtime (the GameObject don't need another component like audio source, MceIncEngine

instantiate automatically an GameObject with AudioSource component at runtime).

LocalizedSprite.cs: Component to attach to a Game Object in unity inspector. It allows to use a localized image at runtime (the Game Object must have a SpriteRenderer or Image component).

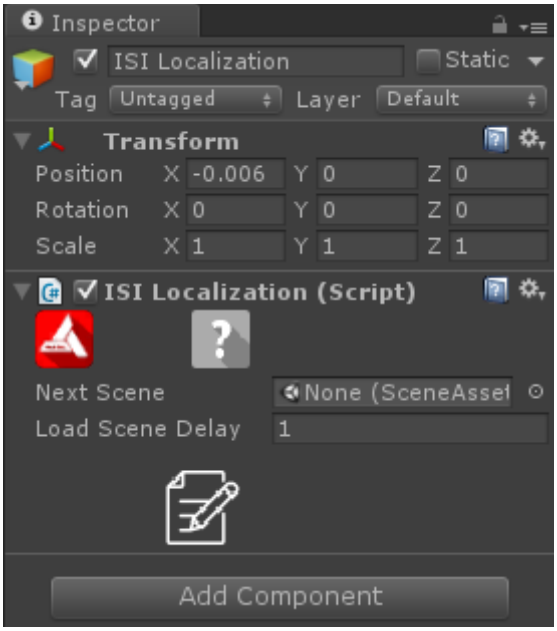
EnumAttribute.cs: Custom attribute that allows to display a string array in inspector like popup.

MceIncEngine.dll: A Dll file placed at "Mce Inc/Dll". This dll provides access to some tools of Mce Inc Engine like the sound manager.

Getting Stated

1 - Create your data.

First of all, to use this engine, you must add the component ISILocalization.cs to a Game Object on unity inspector.



Open the editor by clicking on the editor button.

The editor window is separated in two area:

- **The left area** which allows you to create the keys that you use to access the localized contents (string, audio and sprites).

- **The right area** which allows you to translate the keys in the case of localized strings, and associate a key to a content in the case of localized audio and localized sprite.

If you want to format a text, you can use the syntax "{i}" where "i" is an integer replaced at runtime thanks to

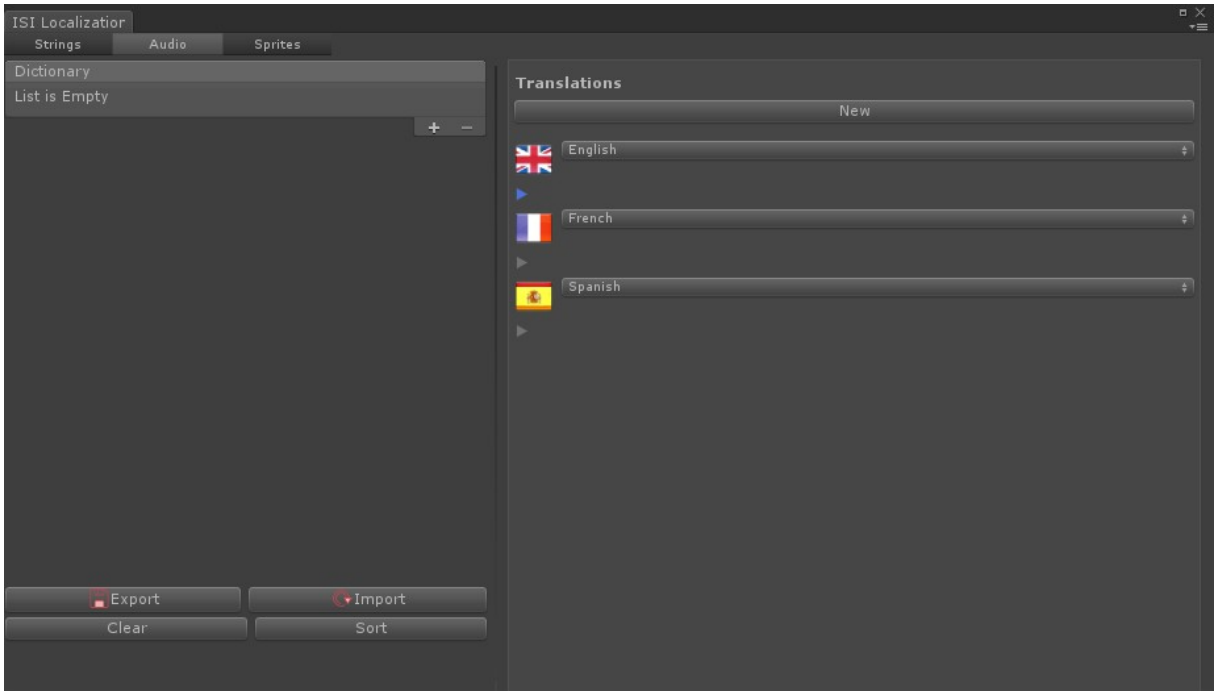
string. Format (string value, params string [] args data); method.

Example:

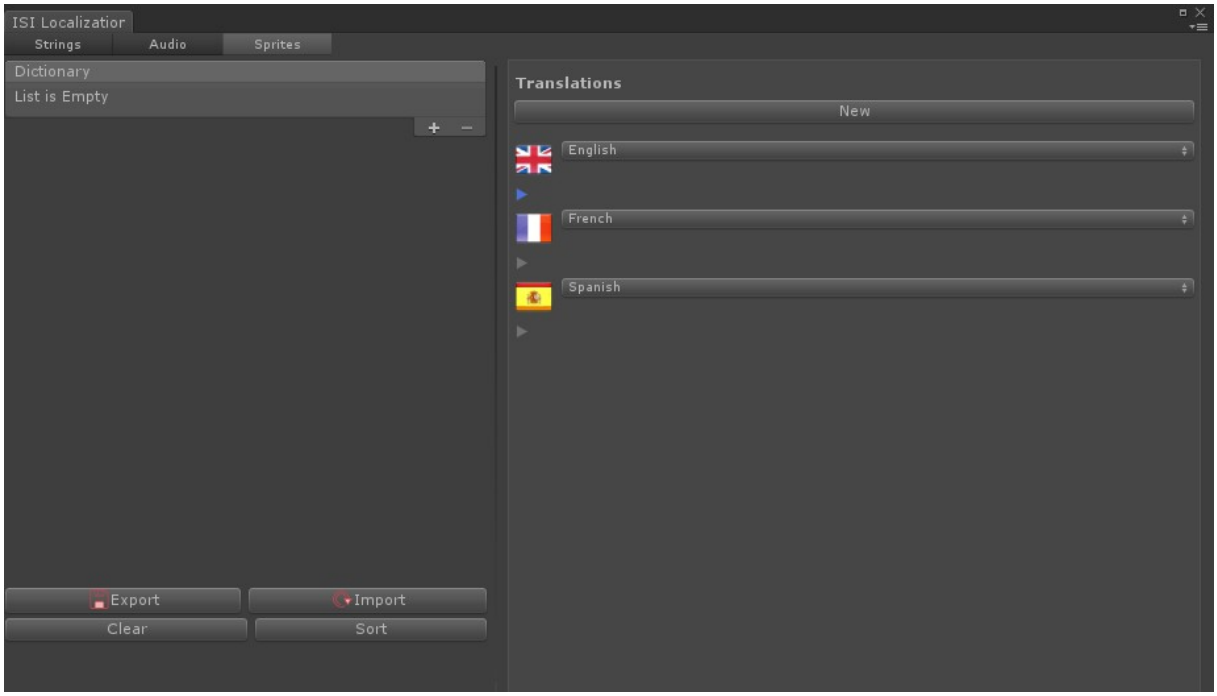
In the case of the key "FORMATED_TEXT" that has "This is a formatted text {0}" like translation in the following image, "{0}" is replaced by another data when the game starts.

Localized string .

Localized audio .



Localized sprite .



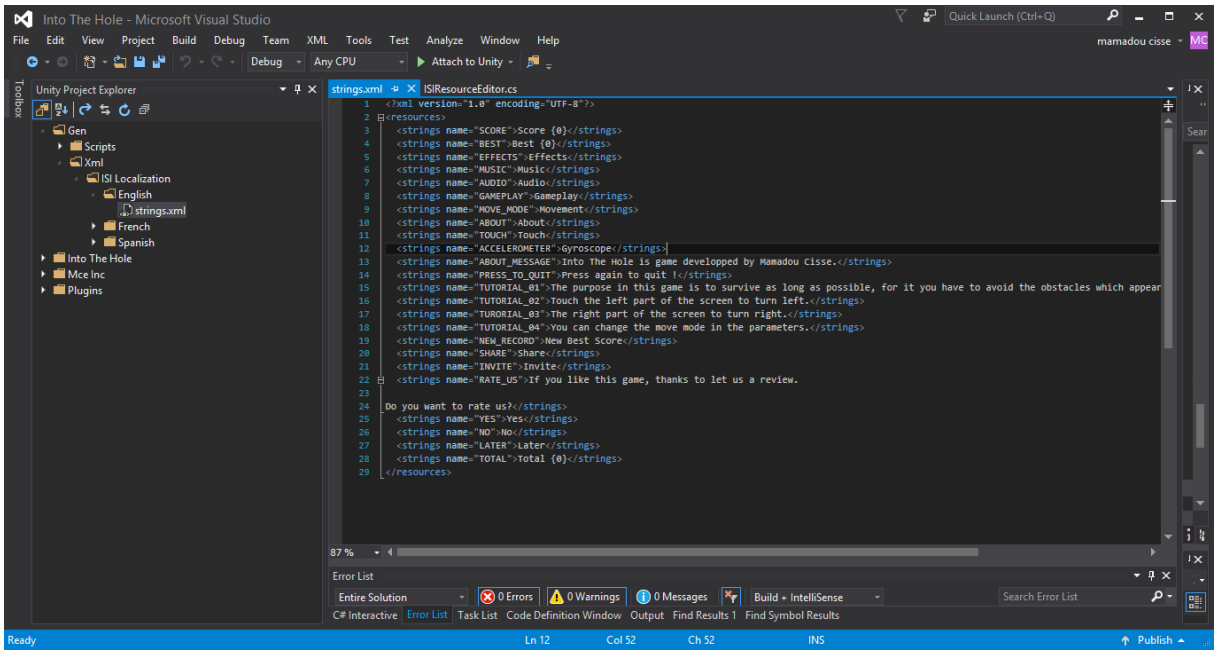
2 - Save and export your data to xml.

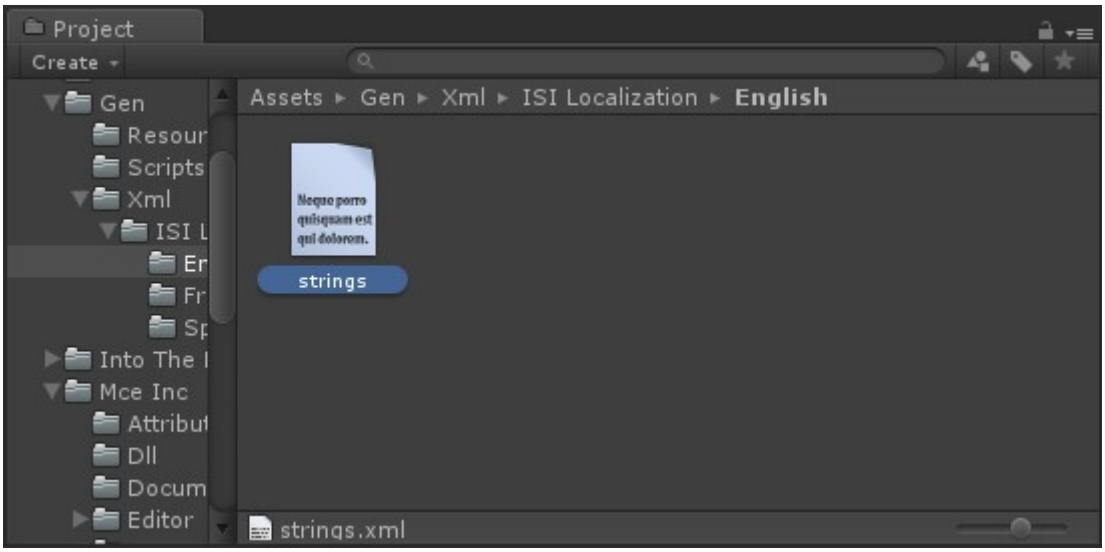
When you finish to create your data, click on "Export" button to generate:

- **The xml files** of your data and generate at

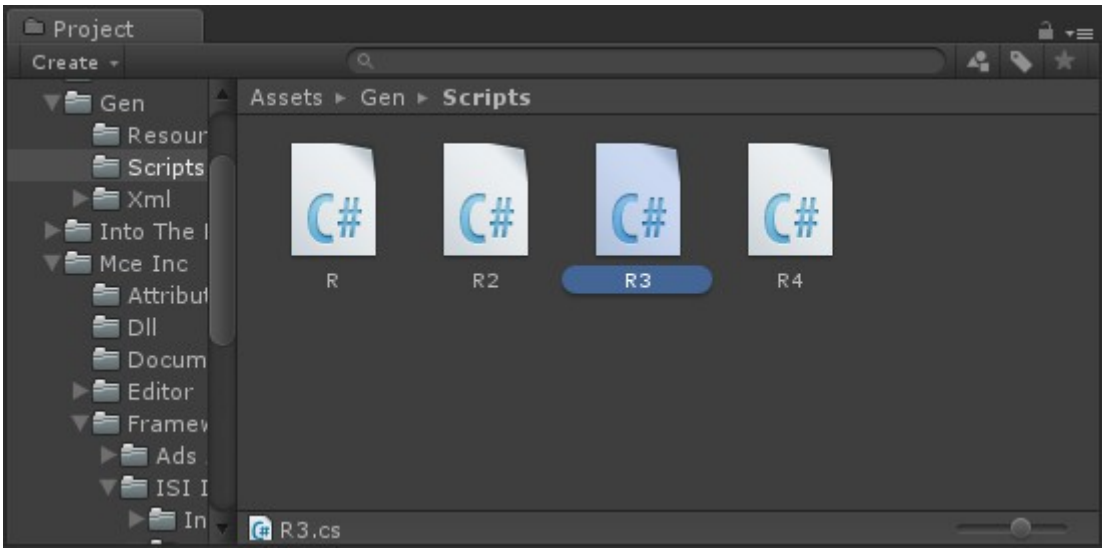
  "Assets/Gen/Xml/ISI Localization/"+ NAME_OF_THE_LANGUAGE+"/string.xml".

  You can use the generated xml files for other projects, you must put them in the right folders and click the button "Import".





- **The class "R3"** at "Assets/Gen/Scripts".

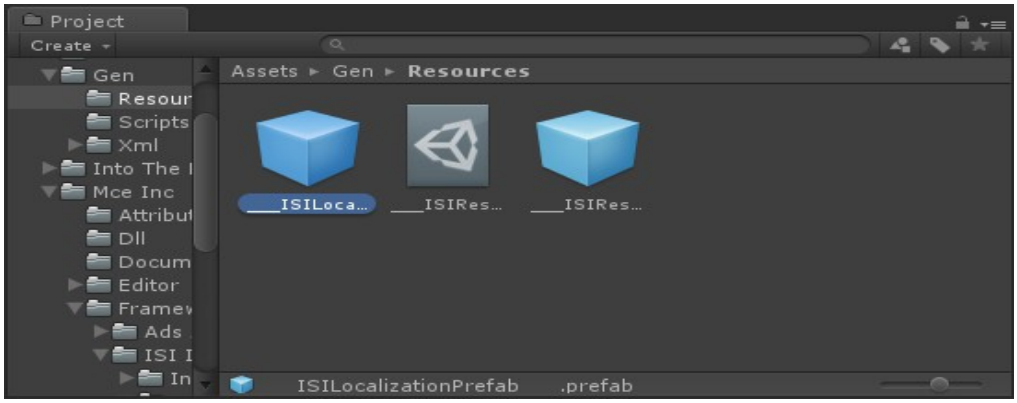


This class provides you access to your keys in script.

- **The prefab "___ISILocalizationPrefab___"** at "Assets/Gen/Resources".

This prefab is instantiated automatically at runtime when you call any function of ISILocalization class. That means the plugin works in any scene of your project even you don't put the GameObject in the scene. It is instantiated automatically.

When the prefab is instantiated, the default language "English" is used like current language. You can modify the default language in the menu "Tools/Mce Inc./Localization/ISI Localization Default Language",



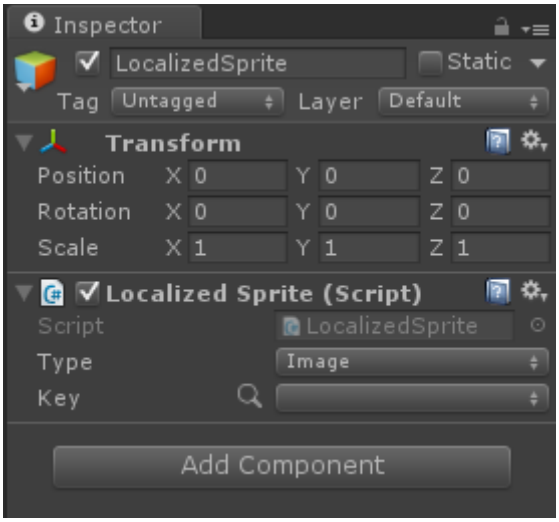
3 - Use your data to translate your product.

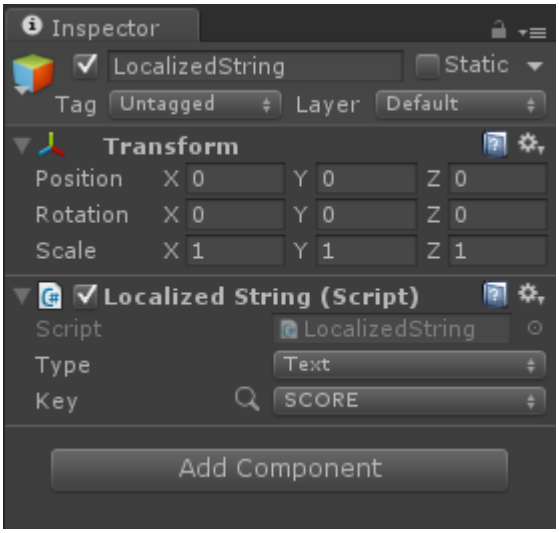
There are two ways to use the data depending of the context.

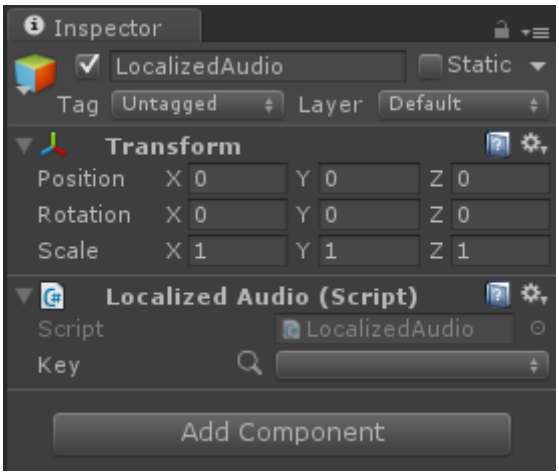
If you don't need do use the data in script use the pre-configured components (LocalizedString.cs, LocalizedAudio.cs, LocalizedSprite.cs).

These component works in the same way.

- Add them to a GameObject on unity inspector.
- Choose the right key that is linked to your content on the dropdown list that appears.







If you want to use the data by script, call the static functions of ISILocalization class or access directly the data thanks to R3 class.

Example 1 : Simple Localized String

For a string key '**HelloWorld**' with the value '**Bonjour tout le monde**' in french , you can access the value by script in two ways:

- `string text = ISILocalization.GetValueOf("HelloWorld");`
- `string text = R3.strings.HelloWorld ;`

Example 2 : Formated Localized String

Let say you want to translate a sentence which can change at runtime. To do that, you have to format your sentence with a brackets '**{}**'. In this example we use the sentence 'You have clicked {0} times' which have '**MyKey**' as key.

To get the value of this localized string, you can do it in two ways :

- `string text = ISILocalization.GetFormatedValueOF("MyKey", 10);`
- `string text = R3.strings.MyKey.Format(10);`

In this case, '**{0}** 'is replaced by 10.

Localized audio and localized sprite works in the same way.

Use the component Flag.cs to select a language on unity editor.

4 - Complement.

- **Language Change Event:** You can use the delegate "ISILocalization.OnLanguageChanged" to do an action when the application language change.
- **See All ISILocalization Components In Scene:** Go to the menu 'Tools/Mce Inc./Localization/ISI Localization/Find Dependencies In Scene' to see all Game Objects in current scene which have component linked to ISI Localization plugin.

- **Good Practice:** You must **NEVER** directly modify the prefab of the ISILocalization in the assets folder. If you want to modify the prefab, place it on the scene modify it and when you click on the button export, the prefab will be automatically updated. (do not forget to delete the prefab which you placed in the scene.)

- **Mce Inc Engine Doc :** Go to the menu 'Tools/Mce Inc./Welcome' and click on the button 'SHOW DOCUMENTATION' to show MceIncEngine.dll documentation.

  You can also click on this link if you want to consult the documentation now.

Thanks for your purchase, please rate this asset if you like it and consult my asset store page to see more assets from Mce Inc.

If you have any question, feel free contact me at:

- Mail => mciissee@gmail.com
- Twitter => http://www.twitter.com/IncMce
- Facebook => http://www.facebook.com/mceinc
- Web-Site => http://mciissee.wixsite.com/mceinc

.

Good luck for your projects.