Unity-Text-Localisation

String localisation system for Unity.

How To Use

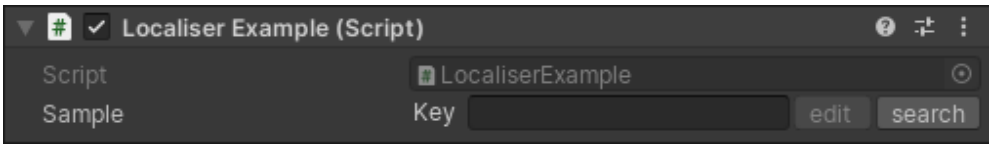
Use Localisation namespace wherever you want to use Localisation.

```
using Localisation;
```

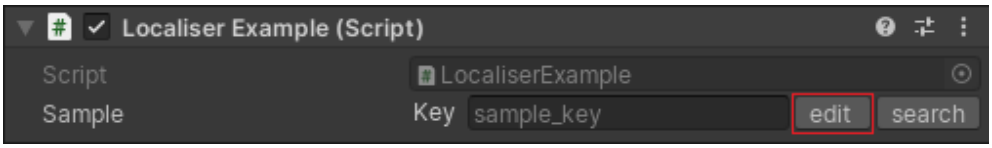
Declare a public LocalisedString value inside any attached script.

```
public LocalisedString sample;
```

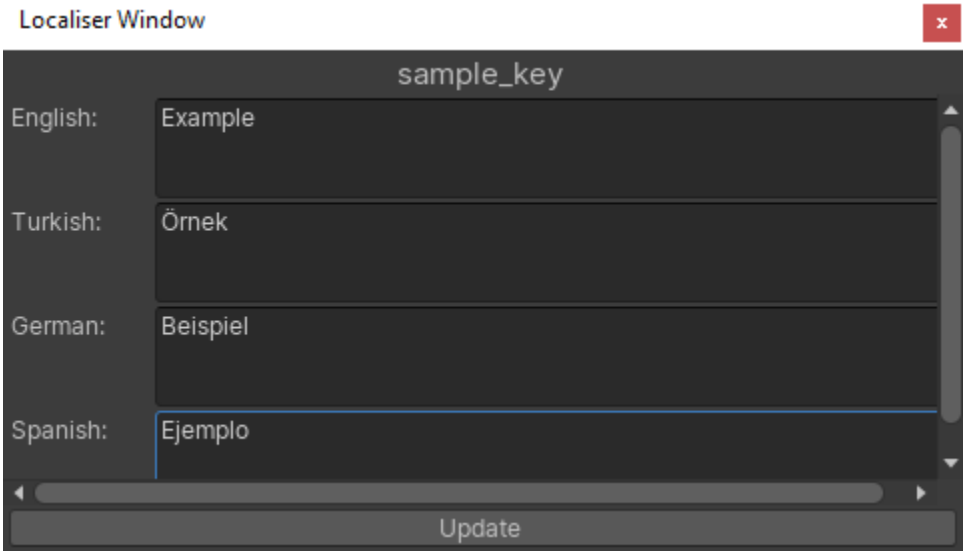
From the inspector, you can edit that Localised String on your attached script.



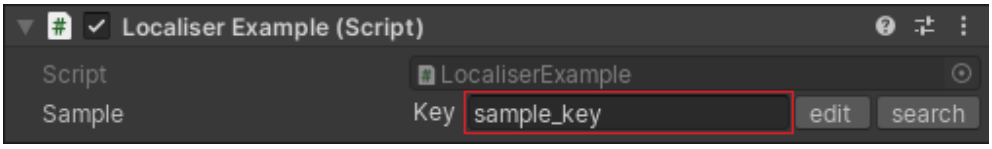
Type a key and click edit button to localise a new value.



Enter localised values for the key and click update button on the bottom of the prompted window.

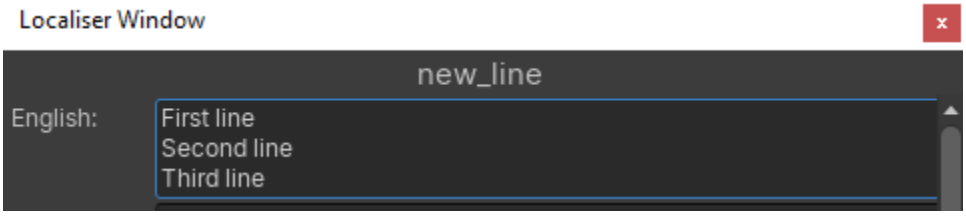


Note that the key now is in vivid color. This means that the key is valid.

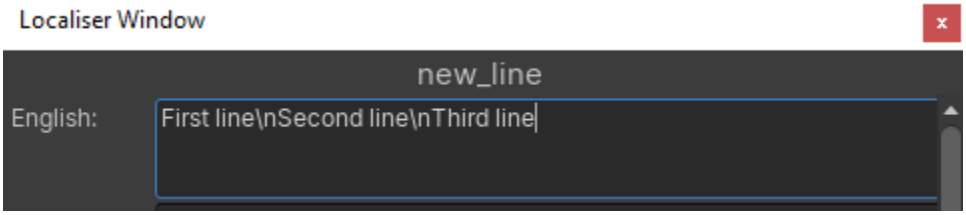


Inserting New Line

Insert line breaks by hitting return key as usual.



Alternatively, it is possible to insert new line by "\n". It will be converted to new line.



Add New Languages

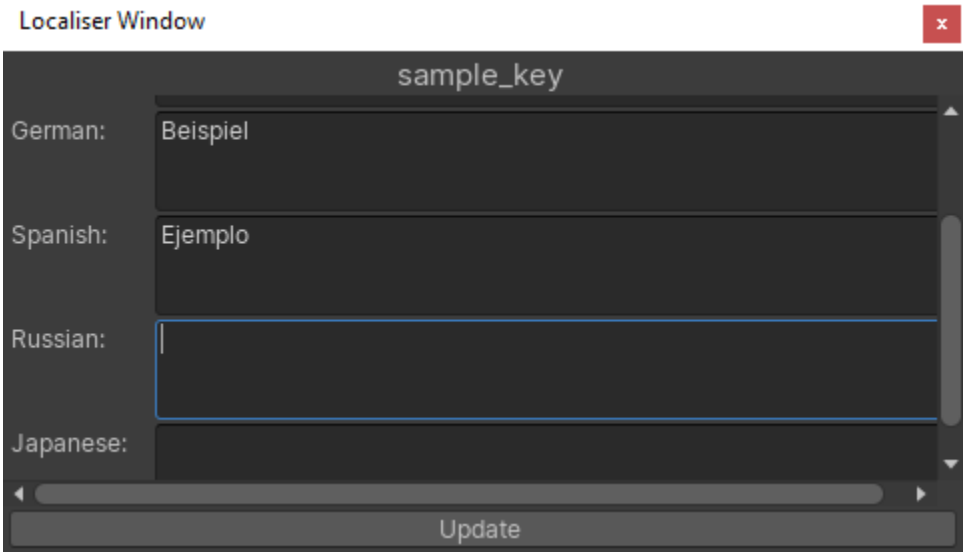
Open Assets/Localisation/Scripts/Languages.cs file. By default there are four languages defined.

```
public enum Language
{
   English,
   Turkish,
   German,
   Spanish
}
```

Add desired language(s).

```
public enum Language
{
   English,
   Turkish,
   German,
   Spanish,
   Russian,
   Japanese
}
```

New languages will appear automatically in the edit window.



Delete Existing Languages

Removing a language from the middle of the list will cause some conflicts. After removing language from enum, it is recommended to delete column of that language from Assets/Localisation/Resources/Localisation.csv via some software such as Excel. Also all the keys can be manually updated from the inspector as explained.

Being additive when building localisation system is strongly recommended.

Runtime Localisation

In runtime, you can get localised values from any script.

```
string localisedString = LocalisationManager.GetLocalisedValue(sample.key);
```

Since English is the default language, former code should return the English value. You can change the language beforehand.

```
LocalisationManager.ChangeLanguage(Language.German);
string germanSample = LocalisationManager.GetLocalisedValue(sample.key);
```

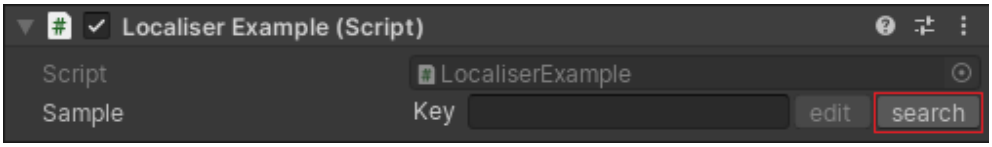
Subscribe your methods to OnLanguageChange event, and they will be called when language changes.

```
public LocalisedString sample;
private string localisedString;

void OnEnable()
{
    LocalisationManager.OnLanguageChange += Localise;
}
```

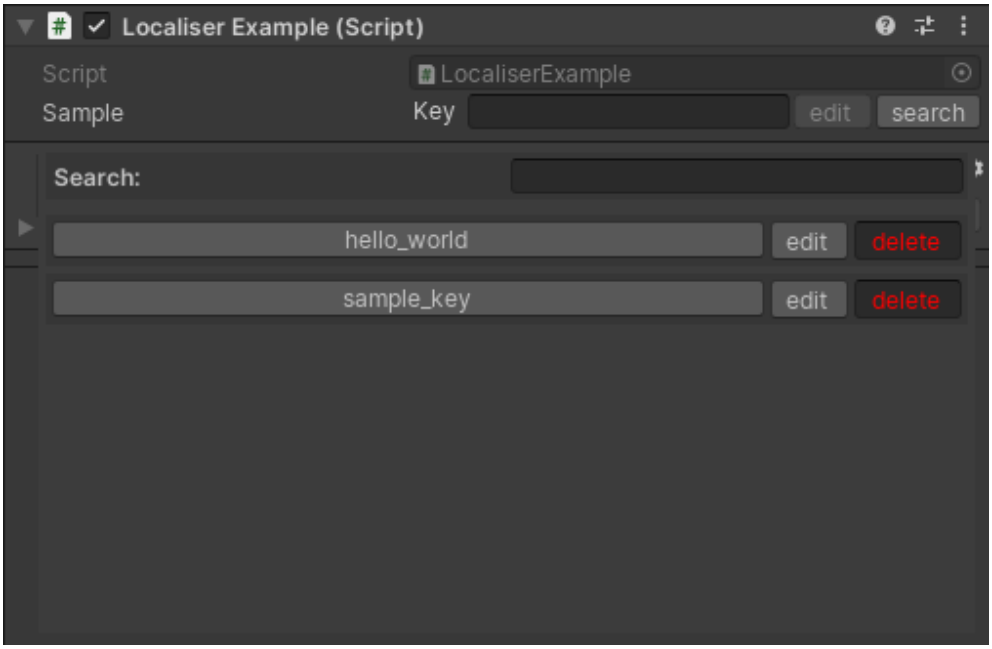
```
void OnDisable()
{
    LocalisationManager.OnLanguageChange -= Localise;
}

void Localise()
{
    localisedString = LocalisationManager.GetLocalisedValue(sample.key);
}
```

Easy Management

Search among existing keys via search button.



Choose, edit or delete existing keys easily.



See the sample scene in Assets/Localisation/Sample folder.

https://github.com/demircialiihsan