

I18N Unity Plugin

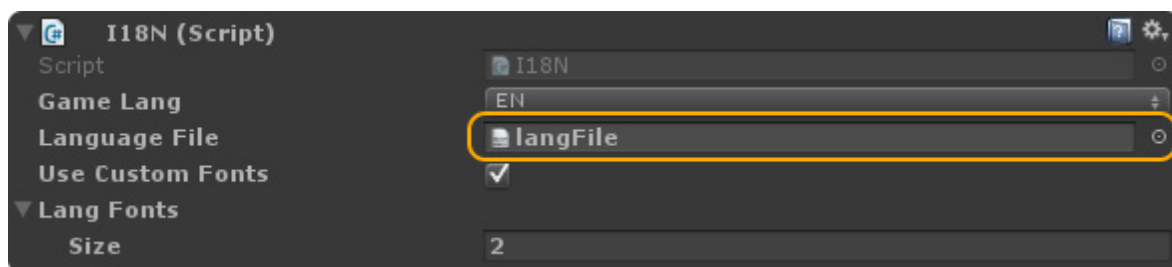
Features

- Text translation based on application language.
- Supports texts with parameters and special characters.
- Use different fonts for different languages.
- Easy to integrate and use.
- Works with unity 4.6+

How to use

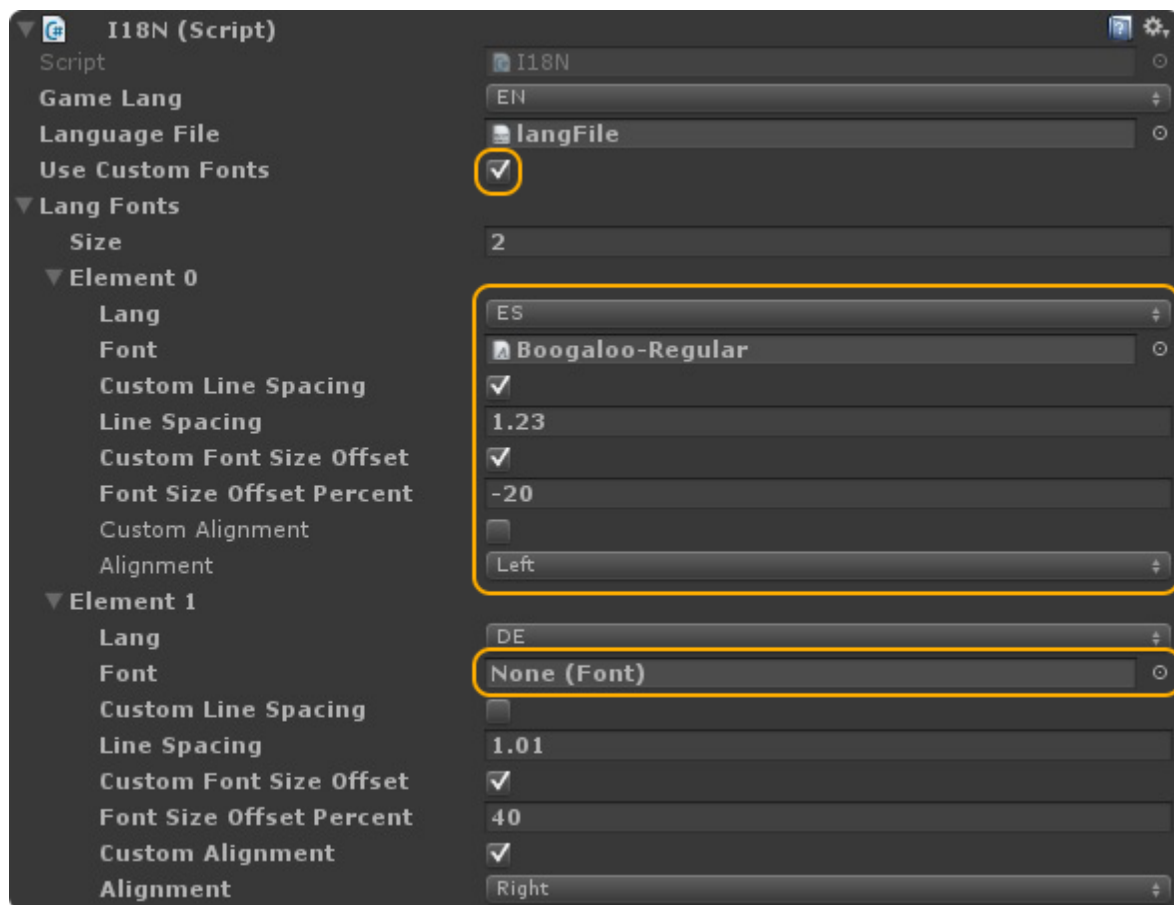
Import

1. Import plugin to your project,
2. add I18N prefab to the scene,
3. attach language file to I18N prefab.
4. You can use translations now!



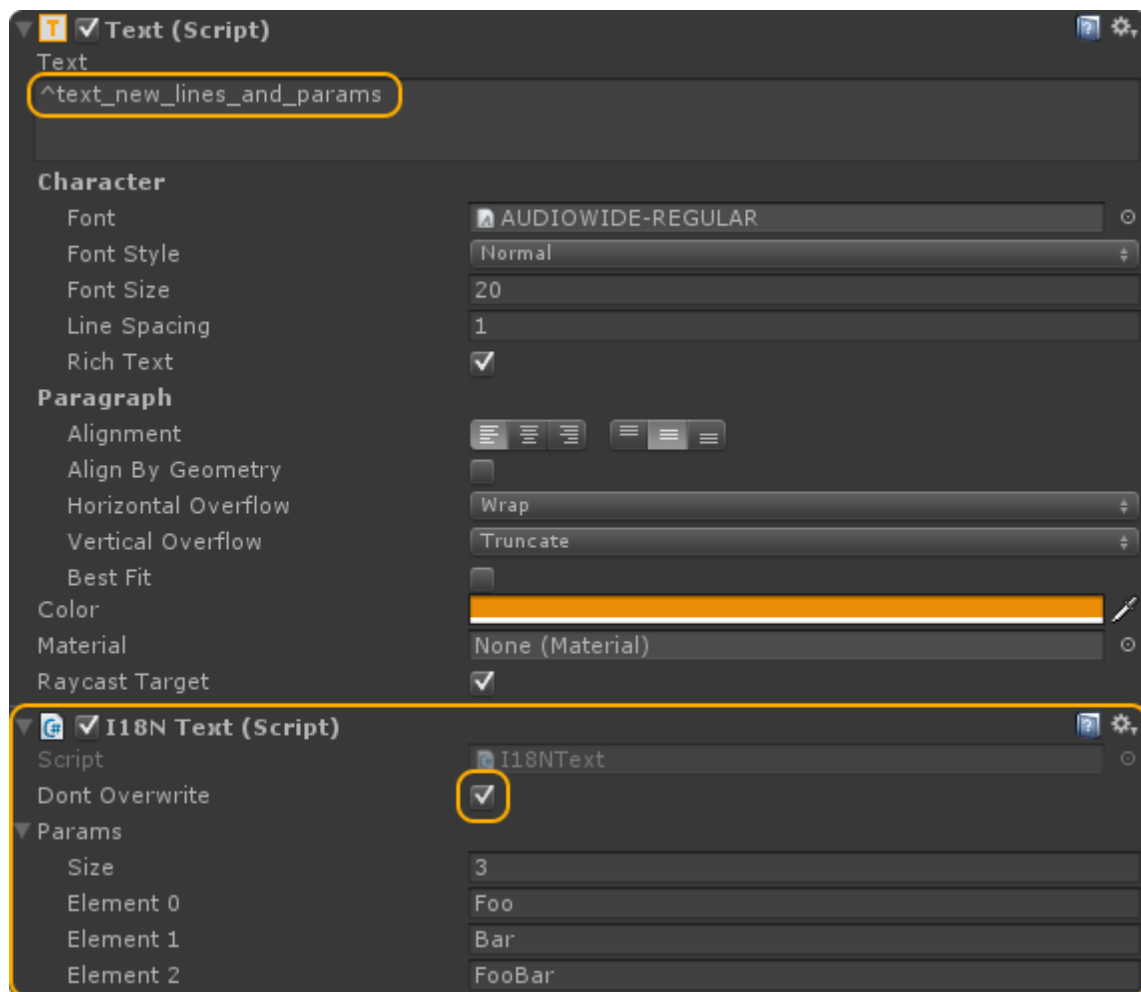
Custom font for each language

1. Find your I18N object in the scene,
2. change size of Lang Fonts property to 1 and check “Use Custom Fonts”,
3. select language for which you need to add custom font,
4. select custom font file (either drag and drop or select from project).
5. You can set custom Line Spacing and Font Size Offset Percent for that language. Remember to click the checkbox!
6. If you don't want to change font for whole language (you need to change only its parameters) simply leave Font field empty.
7. If you need to add more custom fonts for different languages, Change size of Lang Fonts property again.



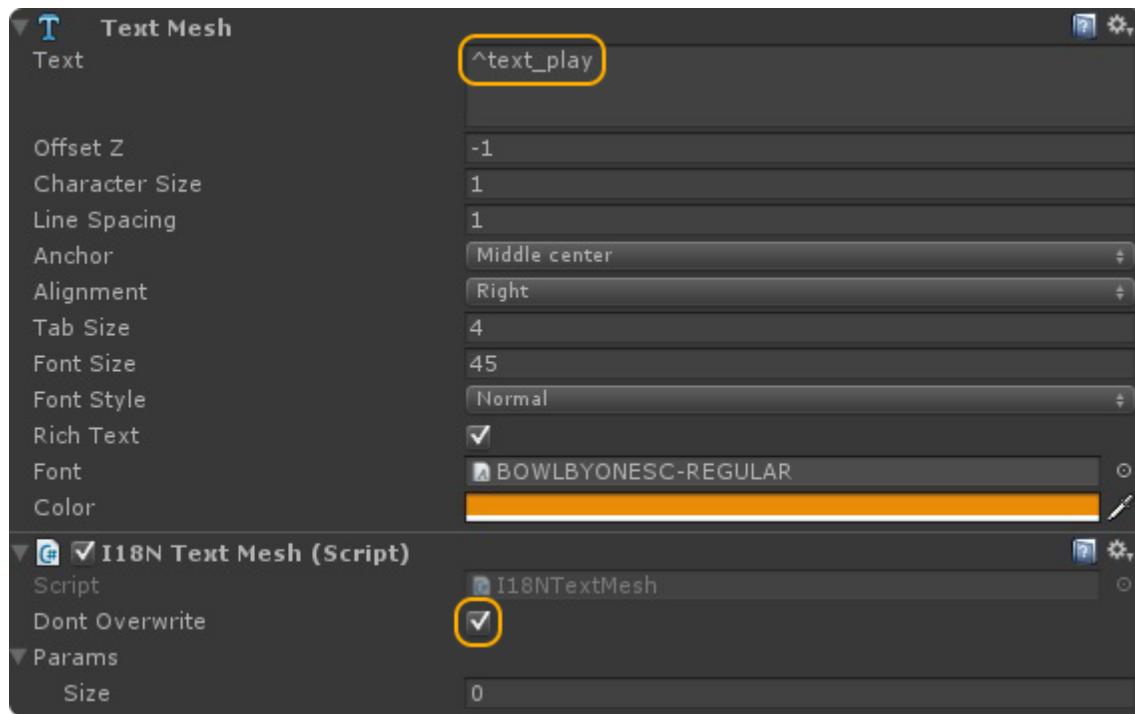
Text component

1. Find your UI Text component in hierarchy,
2. change its text property to any language key from your language file,
3. add I18NText component to your UI Text object.
4. If your text contains parameters, you can add them by changing the size of Params property and adding new params values.
5. If you don't want to override Text component properties with custom font, simply check "Dont Overwrite" option.



Text Mesh component

1. Find your Text Mesh component in hierarchy,
2. change its text property to any language key from your language file,
3. add I18NTextMesh component to your TextMesh object.
4. If your text contains parameters, you can add them by changing the size of Params property and adding new params values.
5. If you don't want to override Text component properties with custom font, simply check "Dont Overwrite" option.



Audio component

1. Find your AudioSource component in hierarchy,
2. set its AudioClip property to your default clip,
3. add I18NAudioSource component to your AudioSource object.
4. Change the size of Clips property and add more sounds for each language.
5. If you dont add Default Clip to I18NAudioSource it will be set from AudioSource component

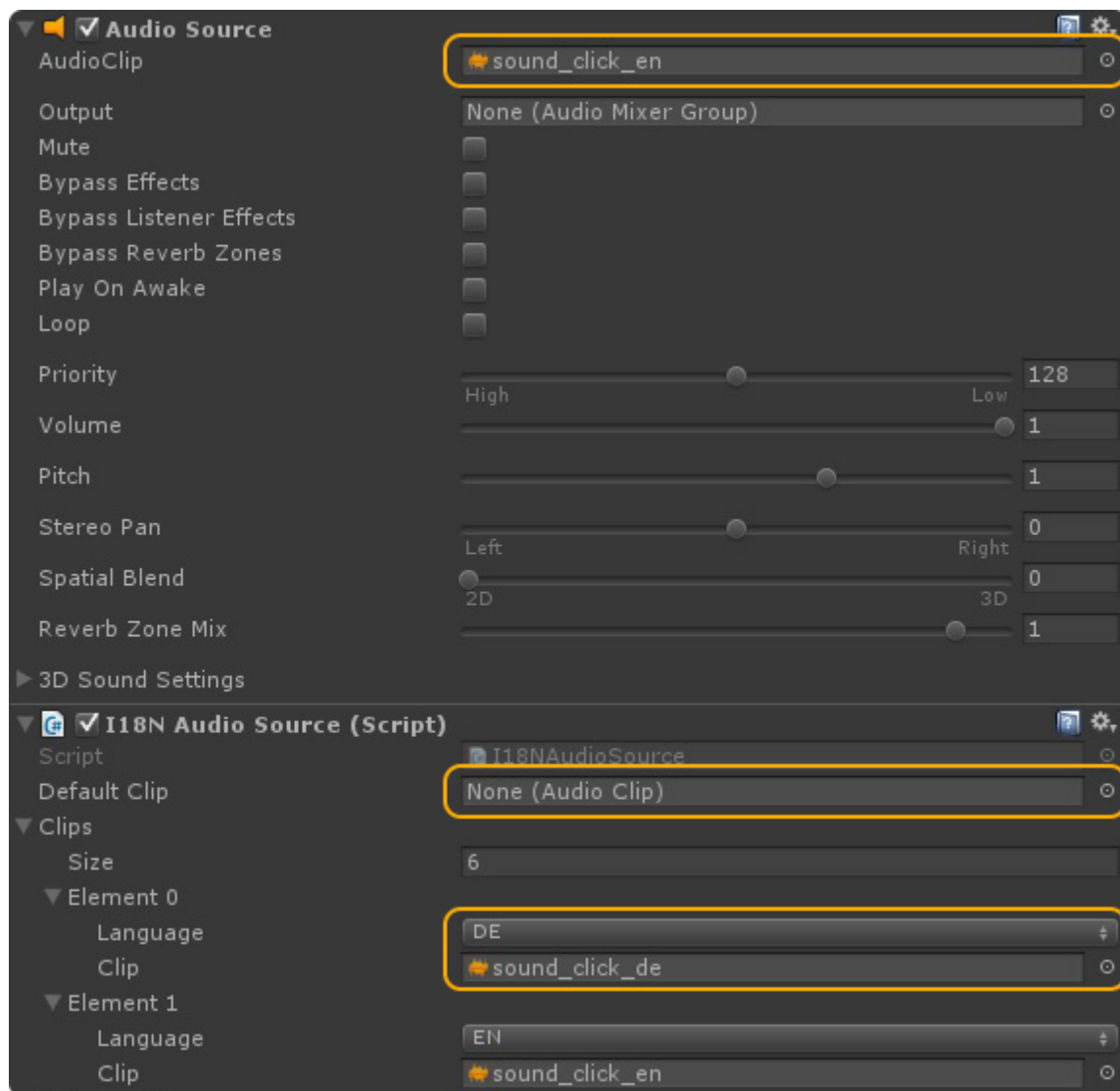
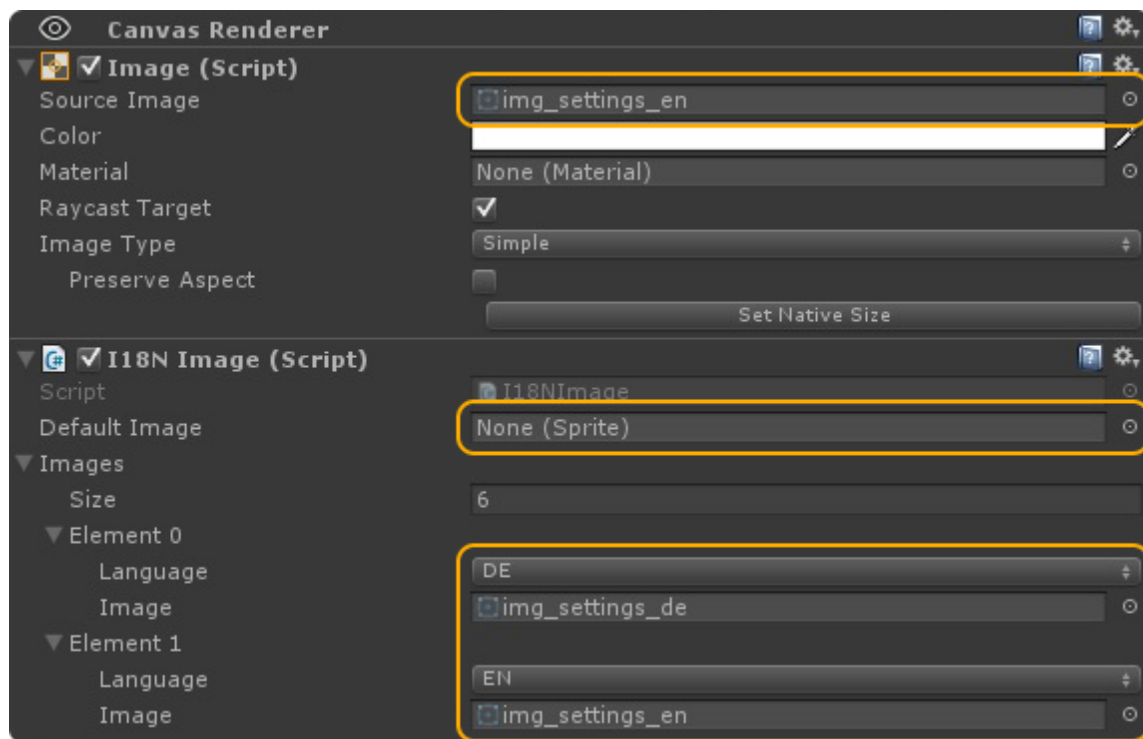


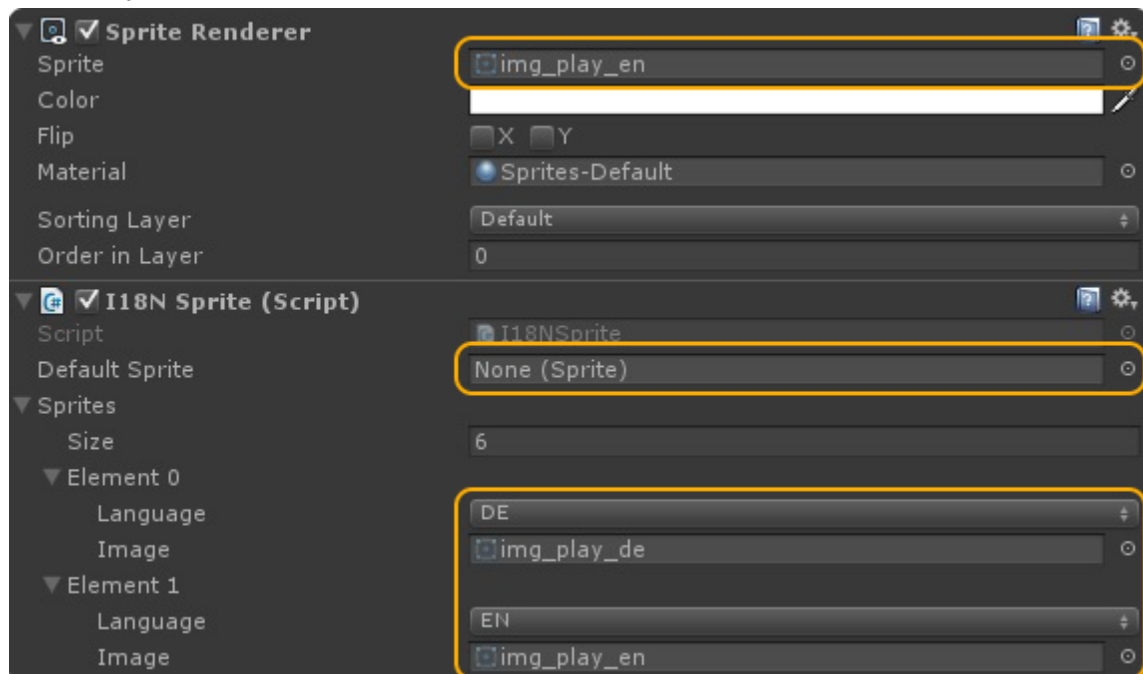
Image component

1. Find your Image component in hierarchy,
2. set its Source Image property to your default image,
3. add I18NImage component to your Image object.
4. Change the size of Images property and add more images for each language.
5. If you dont add Default Image to I18NImage it will be set from Image component



Sprite component

1. Find your Sprite component in hierarchy,
2. set its Sprite property to your default sprite,
3. add I18NSprite component to your Sprite object.
4. Change the size of Sprites property and add more images for each language.
5. If you don't add Default Sprite to I18NSprite it will be set from Sprite component



Language File

File Structure

- File should be saved as *.csv (Unity can import it as Text Asset), but each column must be separated with tabs ('\t')
- We recommend to use Google Docs to create and edit that file and then use File ->Download as->Tab-separated values (*.tsv), and change its extension to *.csv

Example translation table

LangCode	EN	PL	...
^text_foo	Translation_EN	Translation_PL	...
^text_bar	Another_EN	Another_PL	...
...

New lines

- Add '\n' into your translation value to convert it into new line
- Example:
"Example Foo\nText" →
Example Foo
Text

Parameters

- You can add parameters to translation values.
- Example:
"Example {0} Text {1}" → string[] {"Foo", "Bar"}
Example Foo Text Bar

From Code

void I18N.instance.setLanguage(LanguageCode langCode)

Change application language using enum.

LanguageCode langCode - new language from enum.

void I18N.instance.setLanguage(string langCode)

Change application language using string (used from Editor)

string langCode - new language string (if can not parse it to enum, default language will be used).

string I18N.instance.getValue(string key)

Get translation value in current application language.

string key - language key. Should start with “^” character.

Returns translation value string.

string I18N.instance.getValue(string key, string[] parameters)

Get translation value in current application language. Can pass parameters for translation text.

string key - language key. Should start with “^” character.

string[] parameters - optional translation text parameters.

Returns translation value string.