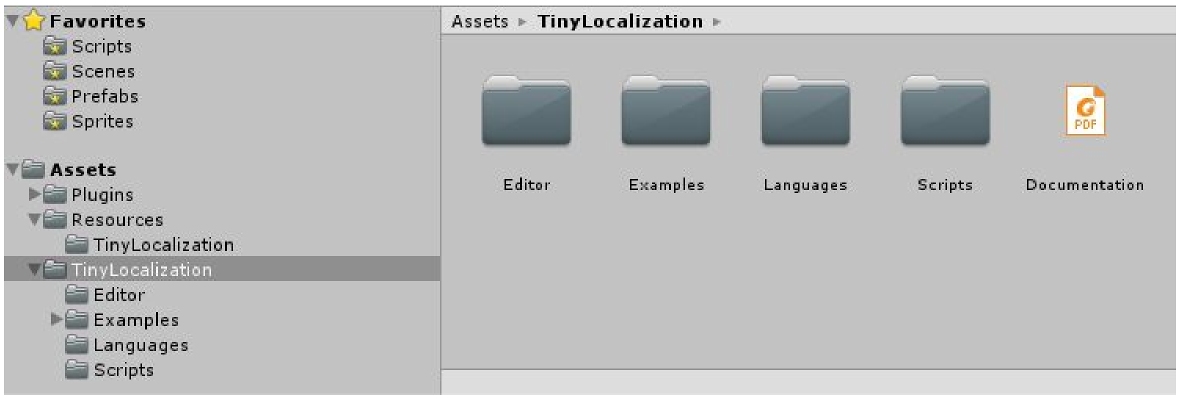
TinyLocalization Documentation

TinyLocalization - tiny and easy text localization extension for Unity.

Get started

To start with TinyLocalization you need first to import package;

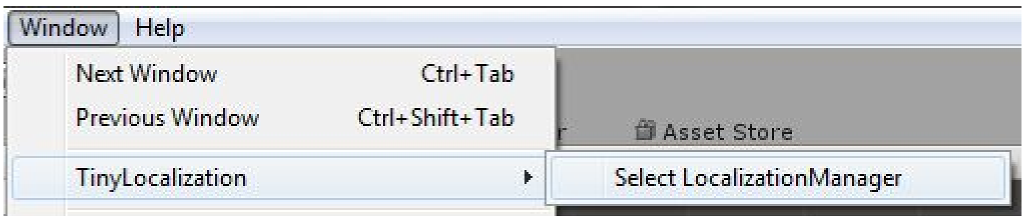
You will see the work folder:



- **Editor:** *the scripts for editor;*
- **Examples:** *here you can find examples of TinyLocalization use;*
- **Languages:** *folder with languages;*
- **Scripts:** *all scripts for TinyLocalization work.*

LocalizationManager

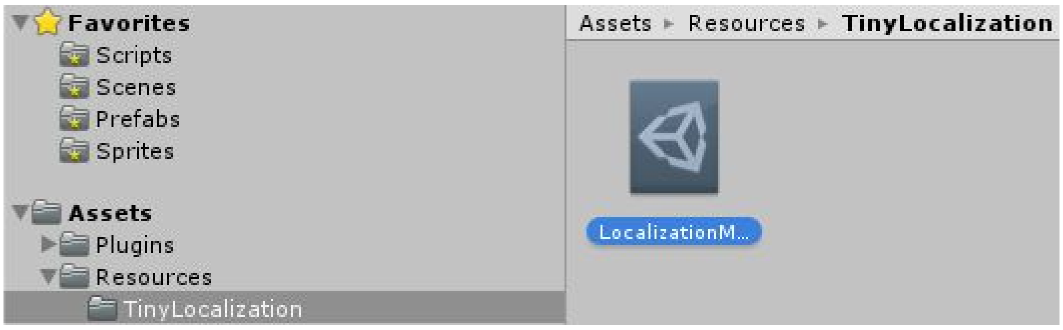
Open it by going to top menu
“Window -> TinyLocalization -> Select LocalizationManager”:



If you doing this first time it creates the instance of ***LocalizationManager***.

-

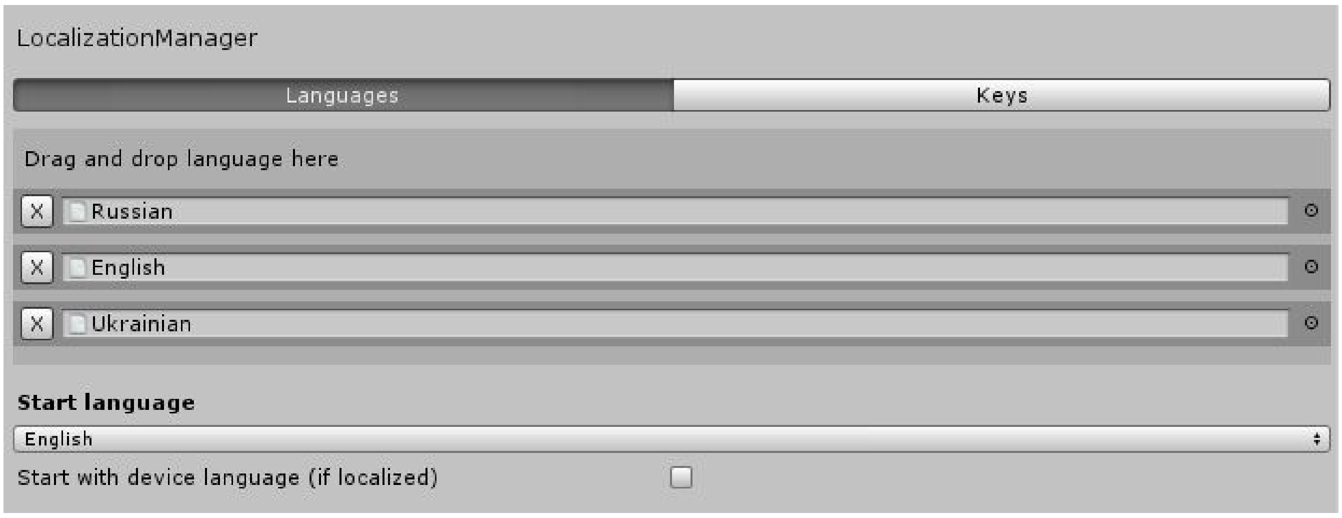
Or go ***“Resources/TinyLocalization”*** folder and click on ***LocalizationManager*** to open manager window if it already created:



-

If ***TinyLocalization*** folder or ***LocalizationManager*** not exist do the first step or open some example. It will create the instance of ***LocalizationManager***.

LocalizationManager
Languages tab


LocalizationManager

Languages | Keys

Drag and drop language here

X Russian

X English

X Ukrainian

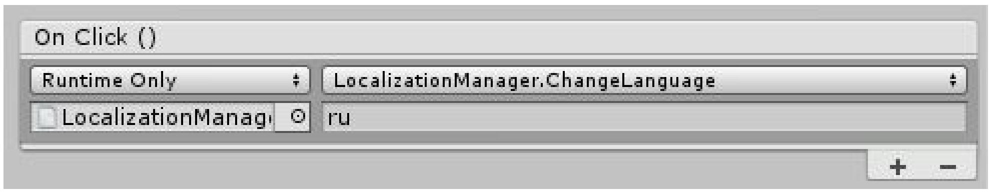
Start language

English

Start with device language (if localized) ☐

- Here you can “Delete” or “Add” language by dragging and dropping from “Languages” folder;
- Choose “Start language”:
  Language which will be loaded at a start of your game. If you set language by “ChangeLanguage()” function, LocalizationManager will be using the set language not the “Start Language”;
- “Start with device language” - if this option is checked and device language is localized, “CurrentLanguage” will return device language localization. If you set language by “ChangeLanguage()” function, *LocalizationManager* will use changed language(not the device language).

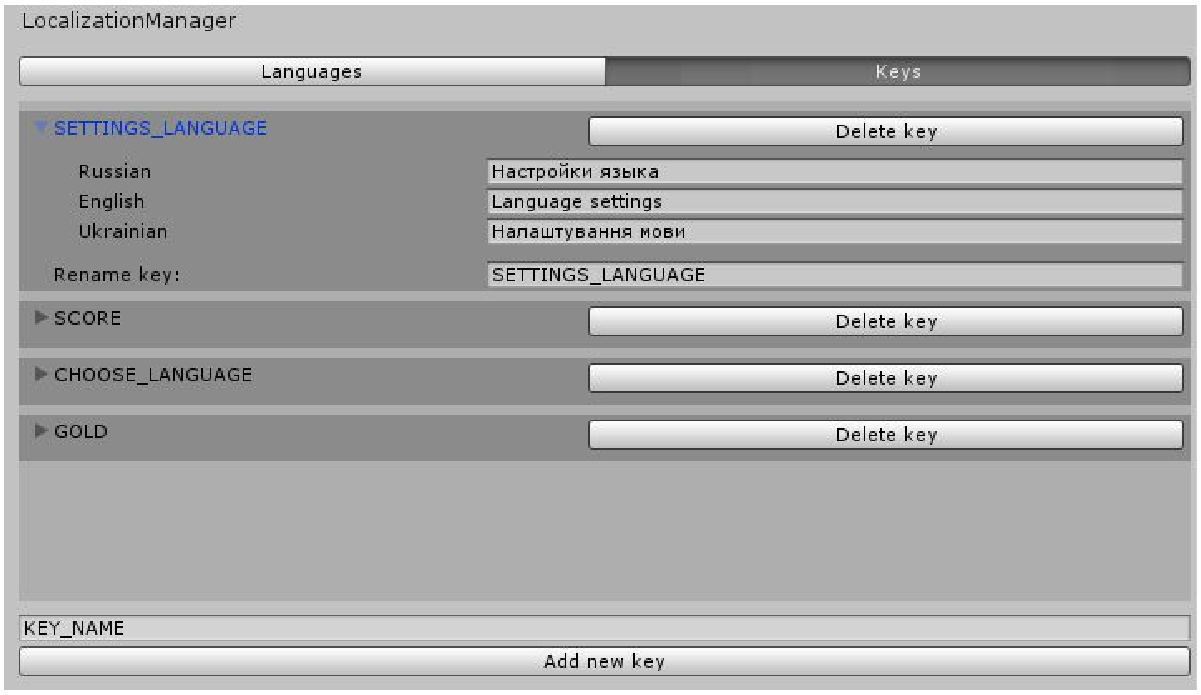
CurrentLanguage is not displayed but you can set it from the code. Or from “Button” *On Click* action by dragging *LocalizationManager* on it:


On Click ()

Runtime Only | LocalizationManager.ChangeLanguage

LocalizationManag | ru

LocalizationManager
Keys tab



LocalizationManager

Languages | Keys

▼ SETTINGS_LANGUAGE — Delete key

Russian	Настройки языка
English	Language settings
Ukrainian	Налаштування мови
Rename key:	SETTINGS_LANGUAGE

▶ SCORE — Delete key

▶ CHOOSE_LANGUAGE — Delete key

▶ GOLD — Delete key

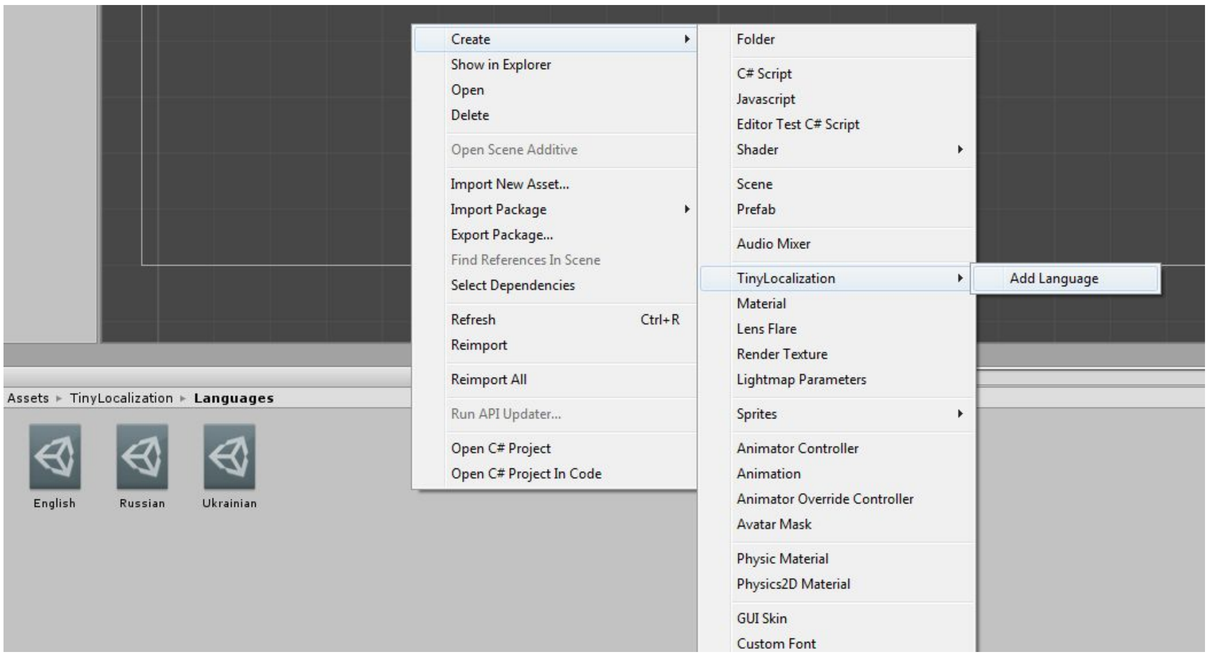
KEY_NAME

Add new key

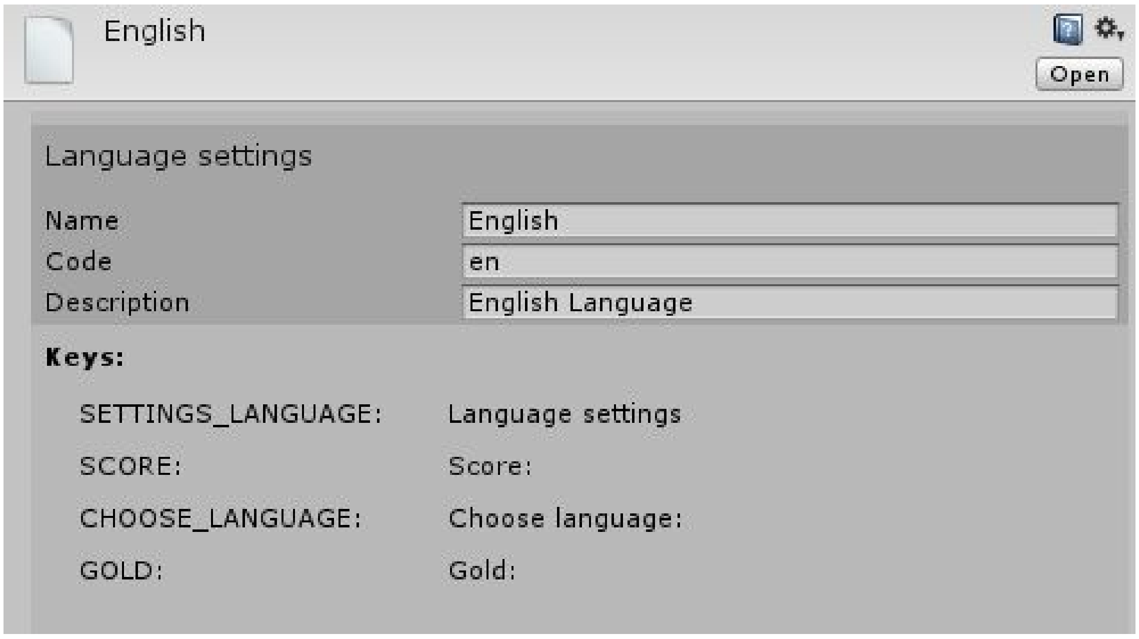
- Set translation to each key;
- Rename keys;
- Delete keys;
- Add new keys (will be added to all languages).

Add new language

- Go to “Languages” folder;
- Right mouse click -> Create -> TinyLocalization -> Add Language;
- Then go to “LocalizationManager” and drag and drop language to languages window.



Language settings



Language settings

Name	English
Code	en
Description	English Language

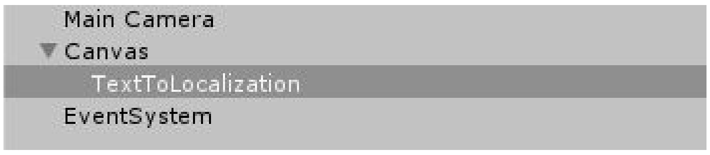
Keys:

SETTINGS_LANGUAGE:	Language settings
SCORE:	Score:
CHOOSE_LANGUAGE:	Choose language:
GOLD:	Gold:

- **Name:** *displaying name;*
- **Code:** *language code. Used for language identification;*
- **Description:** *description for you.*

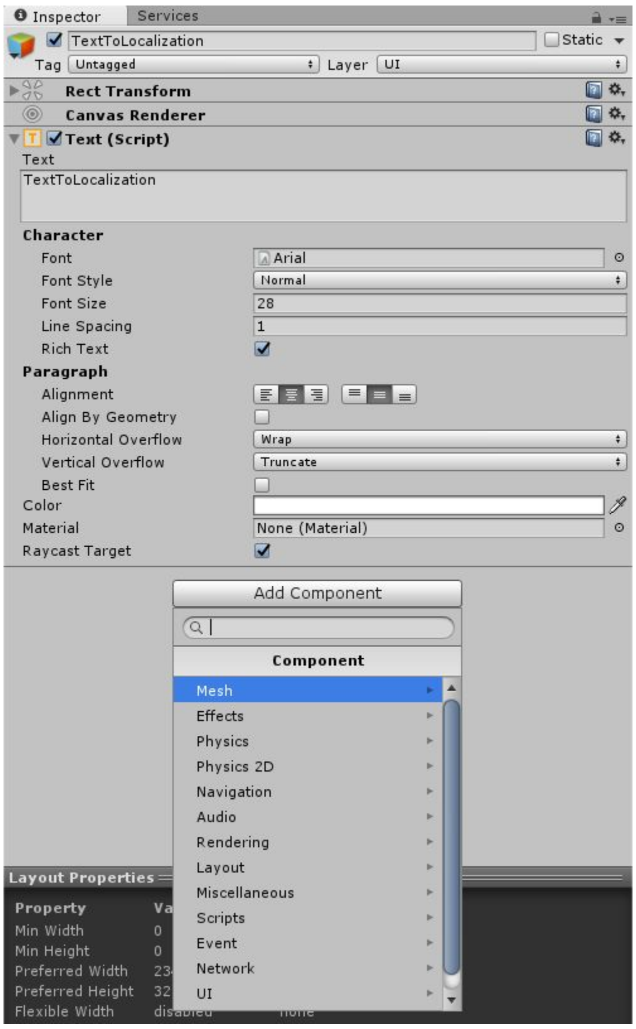
Localization

- Non code:
  - Go to GameObject with “Text” component you want to localize:



```
Main Camera
▼ Canvas
    TextToLocalization
EventSystem
```

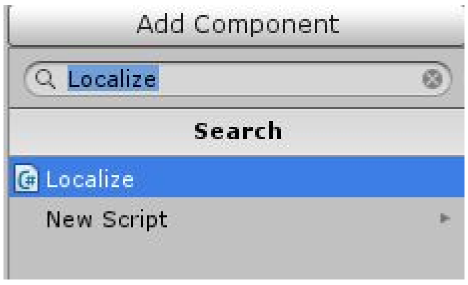
  - Go to “Inspector” view and press “Add Component”:



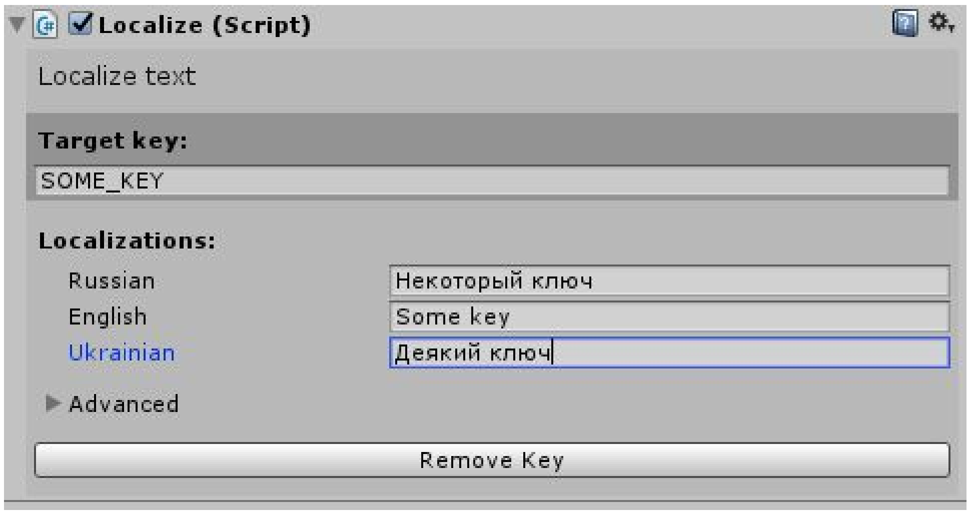
```
Inspector   Services
☑ TextToLocalization                        ☐ Static
Tag Untagged          Layer UI
▶ Rect Transform
  Canvas Renderer
▼ ☑ Text (Script)
Text
TextToLocalization
Character
  Font                  Arial
  Font Style            Normal
  Font Size             28
  Line Spacing          1
  Rich Text             ☑
Paragraph
  Alignment
  Align By Geometry     ☐
  Horizontal Overflow   Wrap
  Vertical Overflow     Truncate
  Best Fit              ☐
Color
Material                None (Material)
Raycast Target          ☑

Layout Properties
Property          Va
Min Width         0
Min Height        0
Preferred Width   23
Preferred Height  32
Flexible Width    disabled    none
```

- Type “Localize” and press Enter:



- Type key name and translation values:



Localize (Script)

Localize text

Target key:

SOME_KEY

Localizations:

Russian	Некоторый ключ
English	Some key
Ukrainian	Деякий ключ

- You’ve done it!

- Code variant:
  You can find it in “Examples” folder

Script references

*Only **public***

- LocalizationManager
  - Events:
    - ***ChangeLanguageAction* OnChangeLanguage**
      Call on language change
  - Private Const Variables:
    - ***string* ASSET_PATH**
      Path to LocalizationManager asset
    - ***string* PREFS_LANGUAGE_CURRENT**
      Key to save CurrentLanguage to PlayerPrefs
  - Public Variables:
    - ***bool* StartWithDeviceLanguage**
      Scene start with device language if CurrentLanguage not set
  - Static Properties:
    - ***LocalizationManager* Instance**
      Singleton
  - Public Properties:
    - ***List<Language>* Languages**
      Copy of list of language
    - ***string* StartLanguage**
      Scene start with this language if CurrentLanguage not set
    - ***Language* CurrentLanguage**
      Current language. Saved in PlayerPrefs

- <u>Public Functions:</u>
  - ***void*** **CleanCurrentLanguage()**
    Delete CurrentLanguage save in PlayerPrefs
  - ***void*** **AddLanguage(*****Language*** **language)**
  - ***void*** **RemoveLanguage(*****Language*** **language)**
  - ***Language*** **GetLanguage(*****string*** **code)**
    Get language by two-letter code
  - ***void*** **SynsLanguages()**
    Synhronize all keys in all languages. If key not exist create and set empty value
  - ***void*** **ChangeLanguage(*****string*** **code)**
    Change current language
  - ***void*** **SetKey(*****string*** **key,** ***string*** **textValue)**
  - ***void*** **RemoveKey(*****string*** **key)**
  - ***bool*** **ContainsKey(*****string*** **key)**
  - ***void*** **RenameKey(*****string*** **key,** ***string*** **newKey)**
  - ***string*** **GetLocalizedText(*****string*** **key)**
  - ***string*** **LanguageNameToCode(*****string*** **systemName)**
    Convert Unity system languages names to two-letter codes
  - ***string*** **LanguageNameToCode(*****string*** **systemName)**
    Convert Unity system languages names to two-letter codes

- Language
  - Events:
    - ***SetKeyAction* OnKeySet**
      Event calls when key set. Only in editor mode
  - Public Variables:
    - ***string* languageName**
    - ***string* code**
      ISO 639-1 two-letter code
    - ***string* description**
  - Public Properties:
    - ***List<LocalizationKey>* Keys**
      Copy of keys
  - Public Functions:
    - ***void* SetKey(*string* key, *string* textValue)**
    - ***void* RemoveKey(*string* key)**
    - ***string* GetText(*string* key)**
    - ***LocalizationKey* GetLocalizationKey(*string key)***
    - ***bool* ContainsKey(*string* key)**
    - ***void* Union(*Language* language)**
      Union keys

- Localize
  - Public Properties:
    - ***string* TargetKey**
      Localize target key
    - ***string* Postfix**
      Additional string after main translation value