

# Transfluent Unity Plugin documentation

This page gathers documentation about the use and abuse of Transfluent Unity Plugin.

## How to start using it

If you're just starting to localize your game, no data conversion is needed as you can use the TransfluentData object's data to fashion your own locale system for runtime in any way you want.

If you have already translated your game into other languages, but want to move to using Transfluent Unity Plugin for all future translations you will probably have to create a small utility to tie in with the Transfluent Unity Plugin and sync your own data with the TransfluentData format. All you really need is all the text entries in the source language in TransfluentData format to start ordering new translations.

If you want to later add texts to existing translations using Transfluent Utility Plugin, you will need to convert all text data to the TransfluentData format and use the utility itself to manage the localizations.

## Scripting reference

### TransfluentUtility.cs

The main class for the Transfluent Unity Plugin. It adds a "Tools" entry into your toolbar to open the utility window.

class TransfluentUtility exposes these public functions:

```
TransfluentUtility Open();  
    Opens or focuses an instance of the TransfluentUtility window and returns  
    the reference to it. NOTE: Remember that the utility needs a full update  
    cycle to initialize everything.
```

```
void Close();  
    Closes the current instance of the utility.
```

```
TransfluentUtility GetInstance();  
    Returns a reference to the TransfluentUtility window,  
    null if it isn't open yet.
```

```
void ResetData();  
    Clears all text entries from the utility
```

```
TransfluentLanguage[] GetTransfluentLanguages();  
    Returns all Transfluent languages stored in the utility's data.
```

```
TransfluentLanguage[] GetLanguagesInUse();  
    Gets all Transfluent languages added to the text items in the utility.
```

```
Dictionary<string, TransfluentText> GetTexts();  
    Gets a collection of all texts in the utility.
```

```
TransfluentText GetText(string textId);  
    Gets a text entry of a certain id. Null if not found.
```

```
void SetText(TransfluentText text)  
    Sets or adds the supplied text entry to the utility.
```

```
void SetTextItem(string textId, int languageId, string text)  
    Sets or adds the supplied text in the provided language to an entry with  
    the specified id. Will fail if language is not found.
```

```
Dictionary<string, string> GetTextsOfLanguage(int languageId);
```

Gets a collection of texts in a certain language.  
Will fail if language is not found.

void AddText(TransfluentText text);  
Adds the provided text entry to the utility's own collection.  
NOTE: See TransfluentText documentation on how to properly populate the object.

void AddMessage(string text)  
Adds a message to the utility log and Unity console log.

void AddWarning(string text)  
Adds a warning to the utility log and Unity console log.

void AddError(string text)  
Adds an error to the utility log and Unity console log.

## TransfluentData.cs

The class TransfluentData is the data container for the texts and languages as well as ordered translations. The utility can load and save these files, but won't automatically update the data container when you alter data within the utility. The script file also contains all the different classes the Transfluent Unity Plugin uses to handle the text entries.

class TransfluentData exposes these public members:

int m\_sourceLanguageId;  
The language id of the development language (source language) as specified by Transfluent Backend API. Create all new texts in this language and send to Transfluent for translation.

List<TransfluentLanguage> m\_languages;  
A list of all known TransfluentLanguages.

TransfluentTextGroup m\_textGroup;  
Container for all text entries of the current project;

List<TransfluentTranslationEntry> m\_translations;  
A list of all ordered and yet unfinished translation orders.

class TransfluentLanguage exposes these public members and functions:

string m\_code;  
The language code in the format: "en-us" (English US), "en-gb" (English UK) or "fr-fr" (French) and so on.

string m\_name;  
The common name for the language in English

int m\_id;  
The language id as specified by Transfluent Backend API

TransfluentLanguage Clone(TransfluentLanguage original)  
A static function that creates a copy of the original language instance.

class TransfluentTextGroup exposes these public members:

string m\_id;  
The id of the group. Must be the same as the product name in PlayerSettings in the Editor.

```
List<TransfluentText> m_texts;  
    All the text entries in one big list.
```

class TransfluentText exposes these public members and functions:

```
string m_id;  
    The id of the text entry, must be unique.  
  
string m_groupId;  
    Not in use, grouping feature might be added later.  
  
List<TransfluentTextItem> m_texts;  
    All the text items added to the text entry.  
  
TransfluentText Clone(TransfluentText original)  
    A static function that creates a copy of the original language instance.  
  
bool TryGetText(string languageCode, out TransfluentTextItem text)  
    Returns true if it finds a text item of the specified language in the text entry  
    which will be set to the out parameter. False if it wasn't found.
```

class TransfluentTextItem exposes these public members:

```
string m_languageCode;  
    The language code of this item in the format: "en-us" (English US),  
    "en-gb" (English UK) or "fr-fr" (French) and so on.  
  
string m_text;  
    The actual text of this item.
```

class TransfluentTranslationEntry exposes these public members and functions:

```
string m_textId;  
    The id of the text entry for which translation was ordered.  
  
string m_groupId;  
    The group id of the text entry for which translation was ordered.  
    Should always be the same as the product name in PlayerSettings in the Editor.  
  
int m_sourceLanguageId;  
    The language id of the source language from which the translations will be made,  
    as specified by Transfluent Backend API.  
  
int m_targetLanguageId;  
    The language id of the target language to which the translations will be made,  
    as specified by Transfluent Backend API.  
  
TransfluentTranslationDateTime m_ordered;  
    When the order was placed.  
  
TransfluentTranslationDateTime m_lastChecked;  
    When the order status was last checked.  
  
DateTime ordered;  
    Gets or sets when the orders was placed, in System.DateTime format of  
    universal time.  
  
DateTime lastChecked;  
    Gets or sets when the orders status was last checked,  
    in System.DateTime format of universal time.
```

TransfluentTranslationEntry Clone(TransfluentTranslationEntry original)  
A static function that creates a copy of the original language instance.

class TransfluentTranslationDateTime exposes these public members

int m\_year;  
int m\_month;  
int m\_day;  
int m\_hour;  
int m\_minute;  
int m\_second;  
The time in integer components for easy and automatic serialization within Unity.

DateTime dateTime  
Gets or sets the time, in System.DateTime format of universal time