OBJECTIS

oStudio 3.0 - Dictionary

User Guide

Revision 1.1

November 26, 2015

1. Document history

Version	Date	Changes
1.0	10.11.15	Initial version
1.1	26.11.15	Updated version

2. Table of contents

1. DOCUMENT HISTORY 3

2. TABLE OF CONTENTS 5

3. GETTING STARTED 7

3.1. Download & installation 7
3.2. My first dictionary project 9

4. TRANSLATING MY APPLICATION 10

4.1. Overview 10
4.2. Common features 10
Editing translations 10
Translation search 10
Translation selection and copy 11
Hiding columns 11
4.3. Changing side 12
4.4. Translator side 13
Dealing with languages 14
4.5. Developer side 16
Visual Studio projects import 17
Translations management 19
Update or create translation files 22
Update Dictionary from Visual Studio projects 23
Linked Visual Studio projects management 25
Options 27

5. IMPORT & EXPORT 28

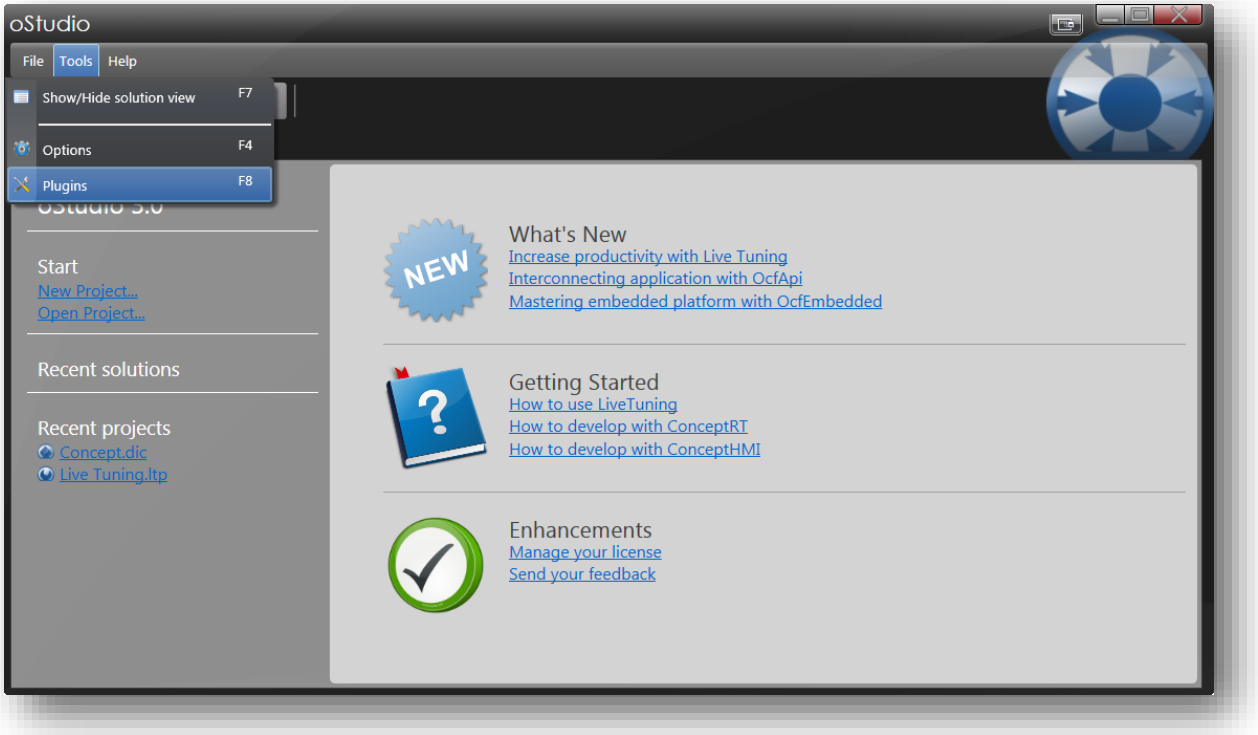
6. ANNEX : *CONCEPT.INTL* BEST PRACTICES 29

6.1. Key naming convention 29
6.2. Intl files classification 29
Plugin translation management 31
Concept libraries translation management 32

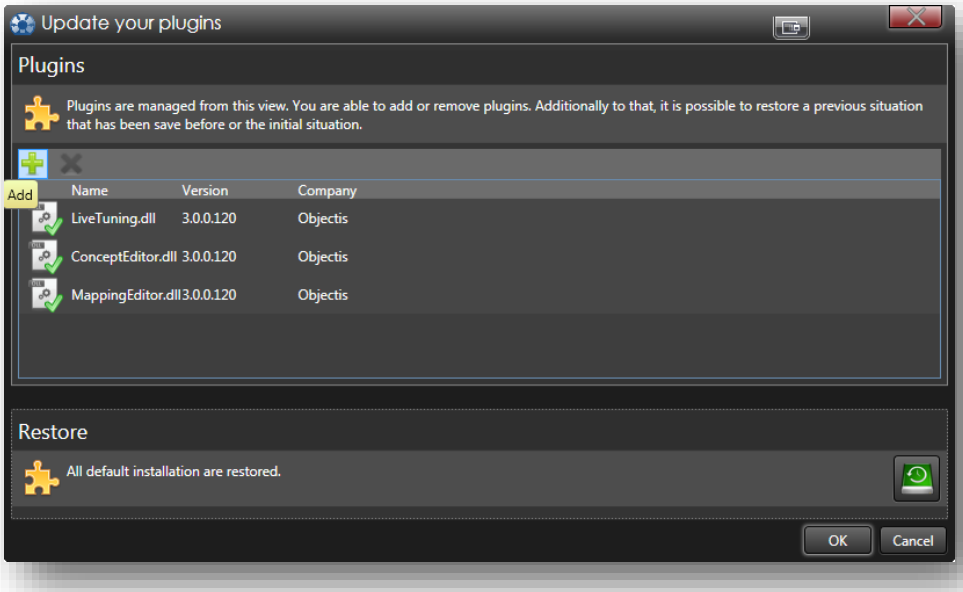
3. Getting started

3.1. Download & installation

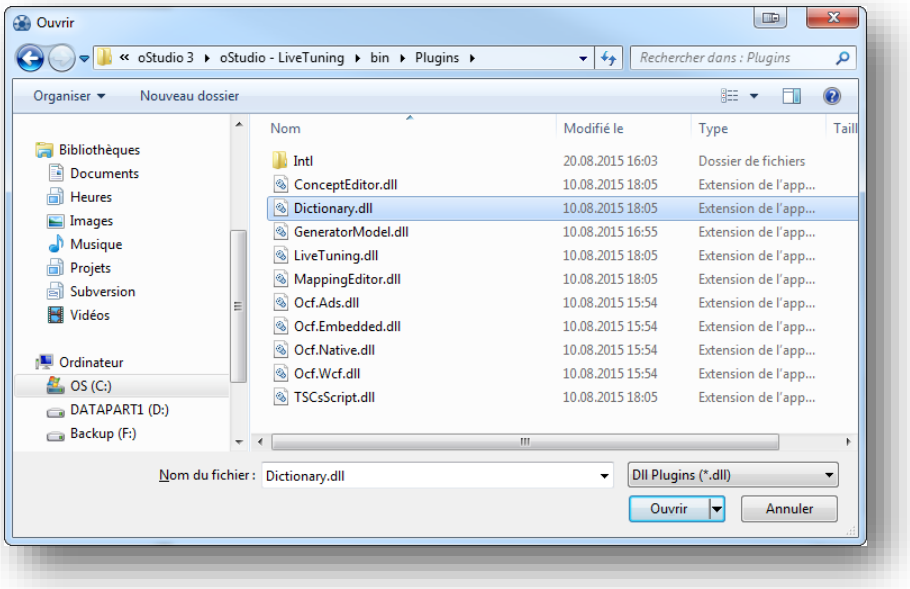
If not already installed, oStudio can be found here (www.objectis.ch). Once installed, go to the plugins manager, *Tools -> Plugins (F8)*



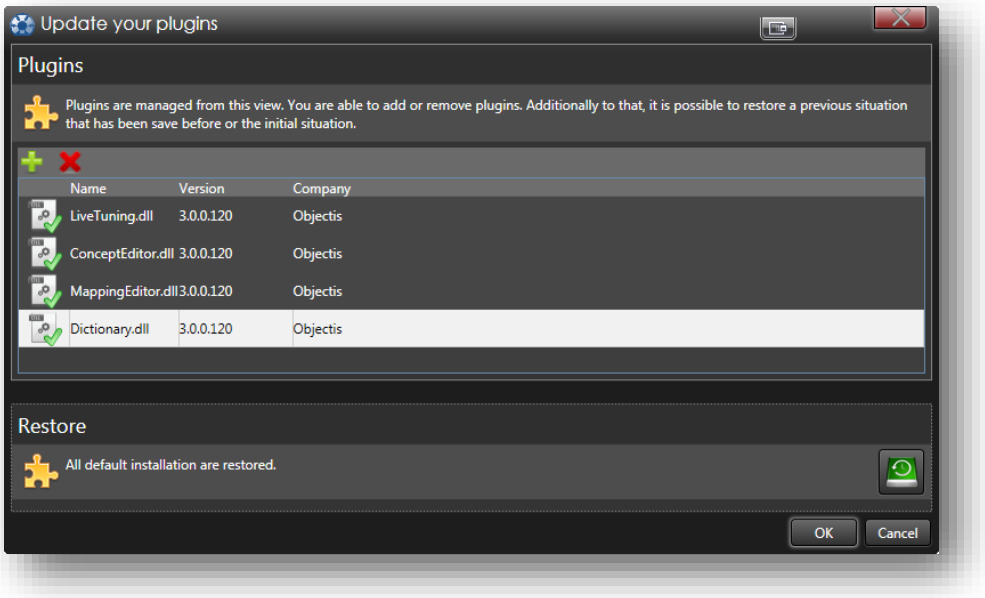
If the Dictionary plugin is not in the list of plugins, click on *Add*.



Select the file "Dictionary.dll" from the installation path (Program Files\Objectis\oStudio 3\oStudio - LiveTuning\bin\Plugins) in the "Plugins" folder. Then click open.

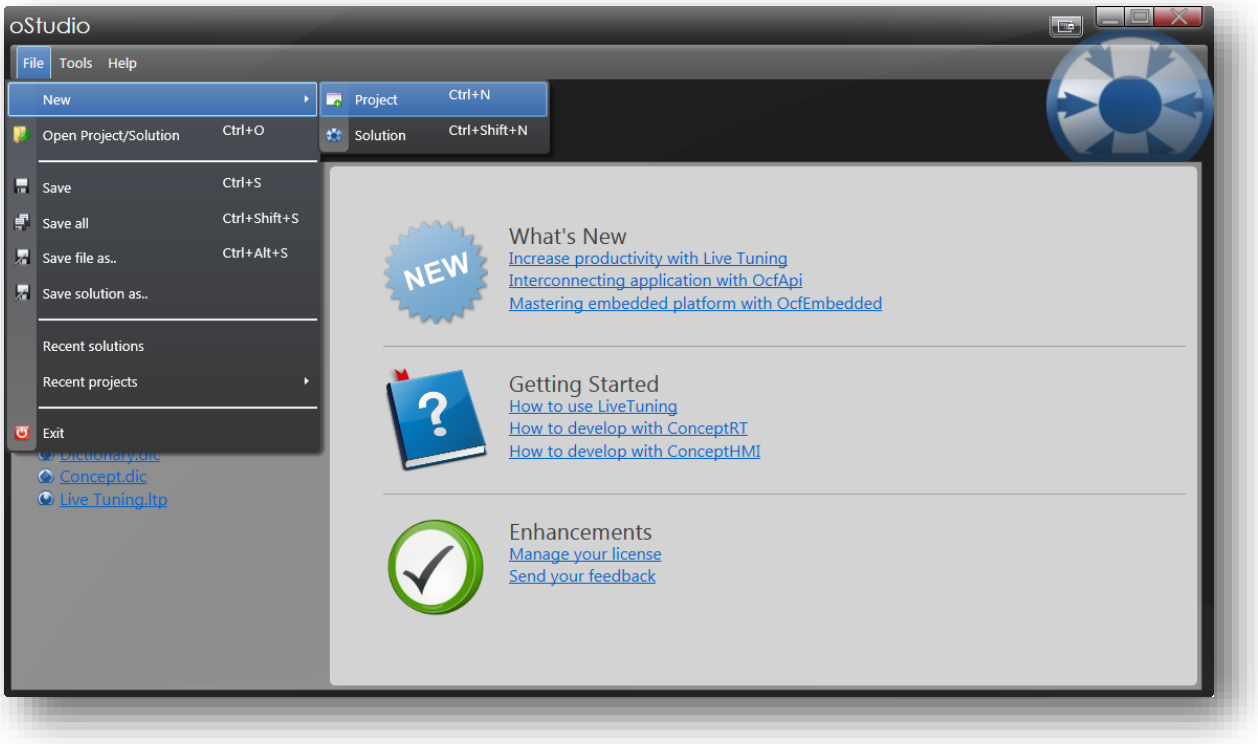


The plugin should now appear in the list.

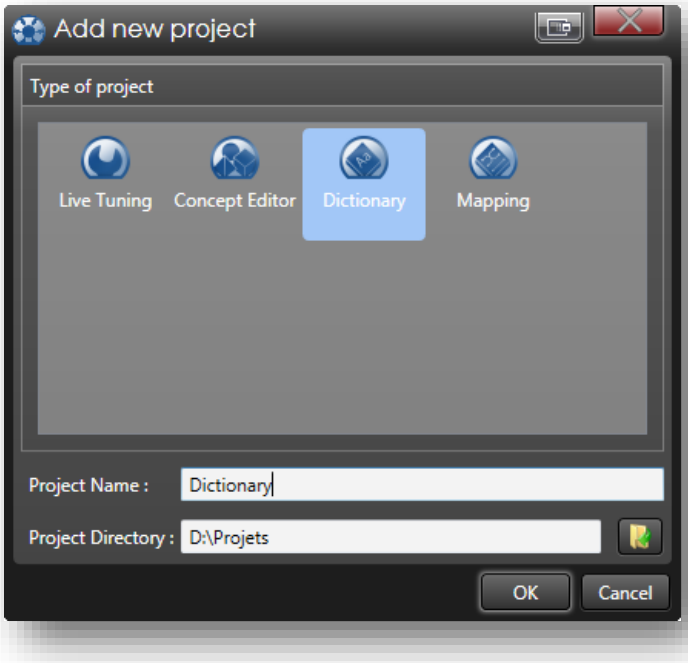


3.2. My first dictionary project

Create a new project by selecting *File -> New -> Project (Ctrl+N)*



Select a Dictionary project



4. Translating my application

4.1. Overview

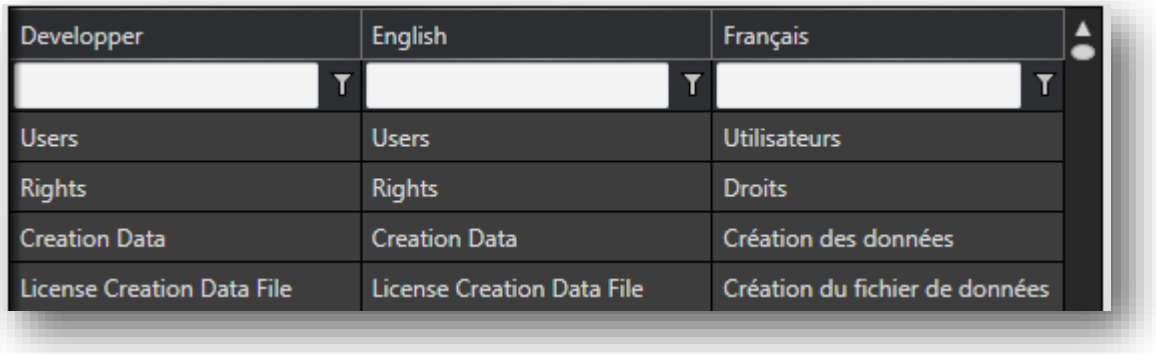
When translating an application, there are two sides involved in the process. The developer, who defined the keys and texts in his language and the translator, who will create new languages and translations for each of those texts.

4.2. Common features

Here are the common features to a translator and a developer.

Editing translations

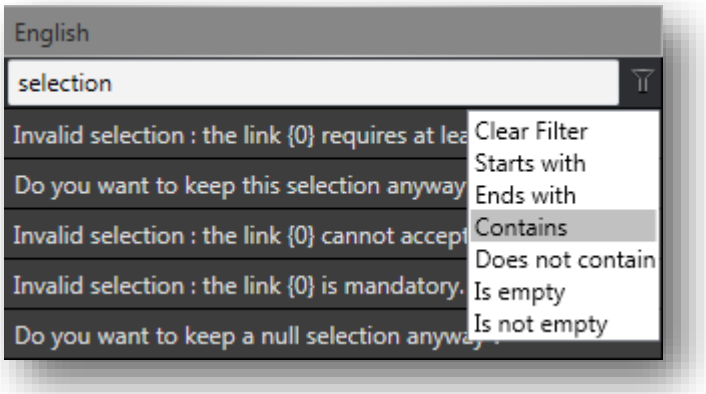
A translation is always composed of the developer's text and the equivalent for each language defined in this project, which you can edit.



Translation search

Several filters can be applied to simplify the edition.

Search by word



This filter enables you to search for specific translations by typing characters contained, at the start or at the end or not contained in the translation. It is also possible to look only for the empty translations, which can be useful when updating a translation file.

Sort alphabetically

Translations can be sorted by column alphabetically ascending or descending. To access these different sorting modes click one or more times on the header of the column.

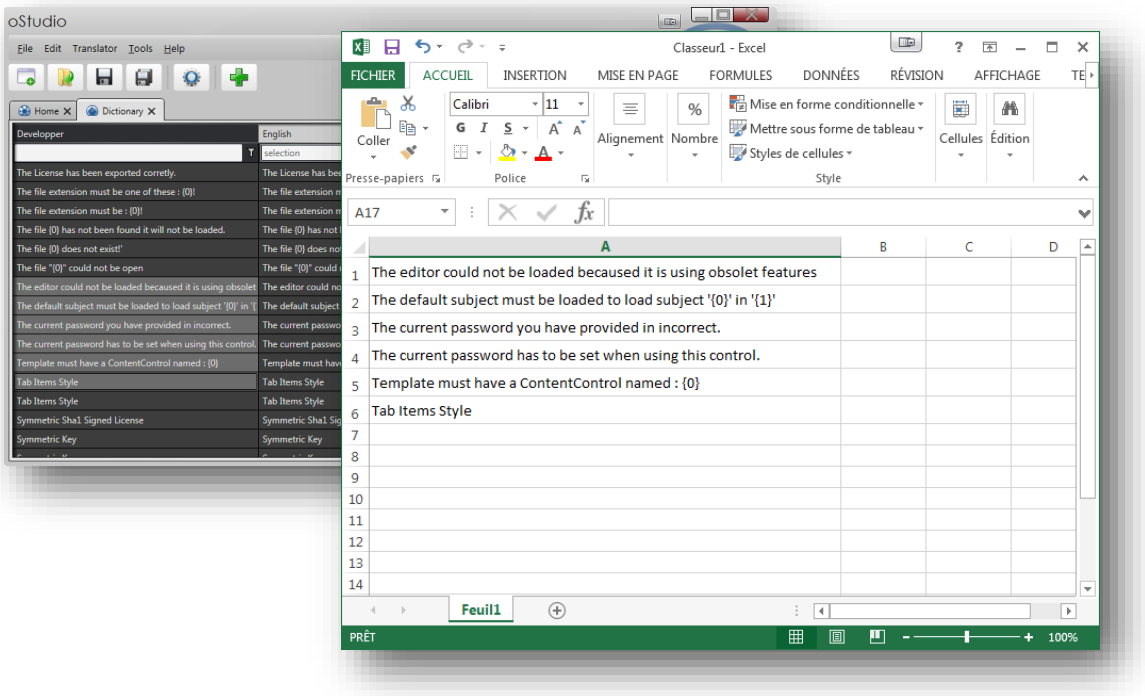
No sort

Ascending

Descending

Translation selection and copy

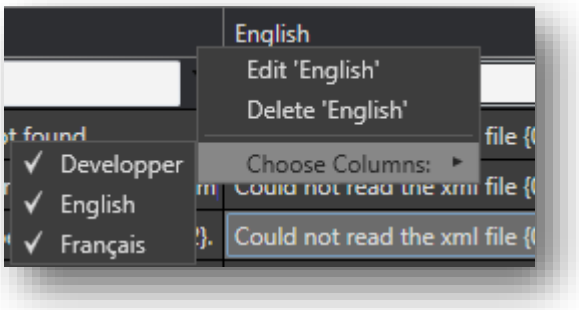
To select multiple translations you can either press Ctrl and select translations separately or press Maj and select the first and last translation of a single column to add all of the items between them to your selection. To select the whole table you can press Ctrl+A. The resulting selection can be copied (Ctrl+C) and pasted (Ctrl+V) in an Excel document.



It is also possible to copy a multi selection, where the selected items follow each others, in another place in the table. For example you can copy items from the developer column to the english column by selecting the cell where you would like to paste the first selected item and pasting the selection (Ctrl+V).

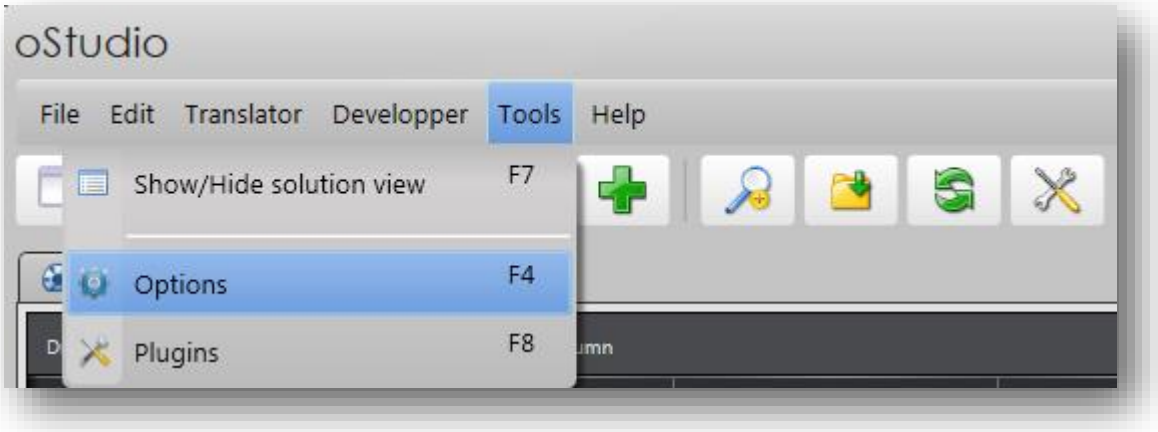
Hiding columns

By doing a right click on the columns header, you can choose which column to show or hide.

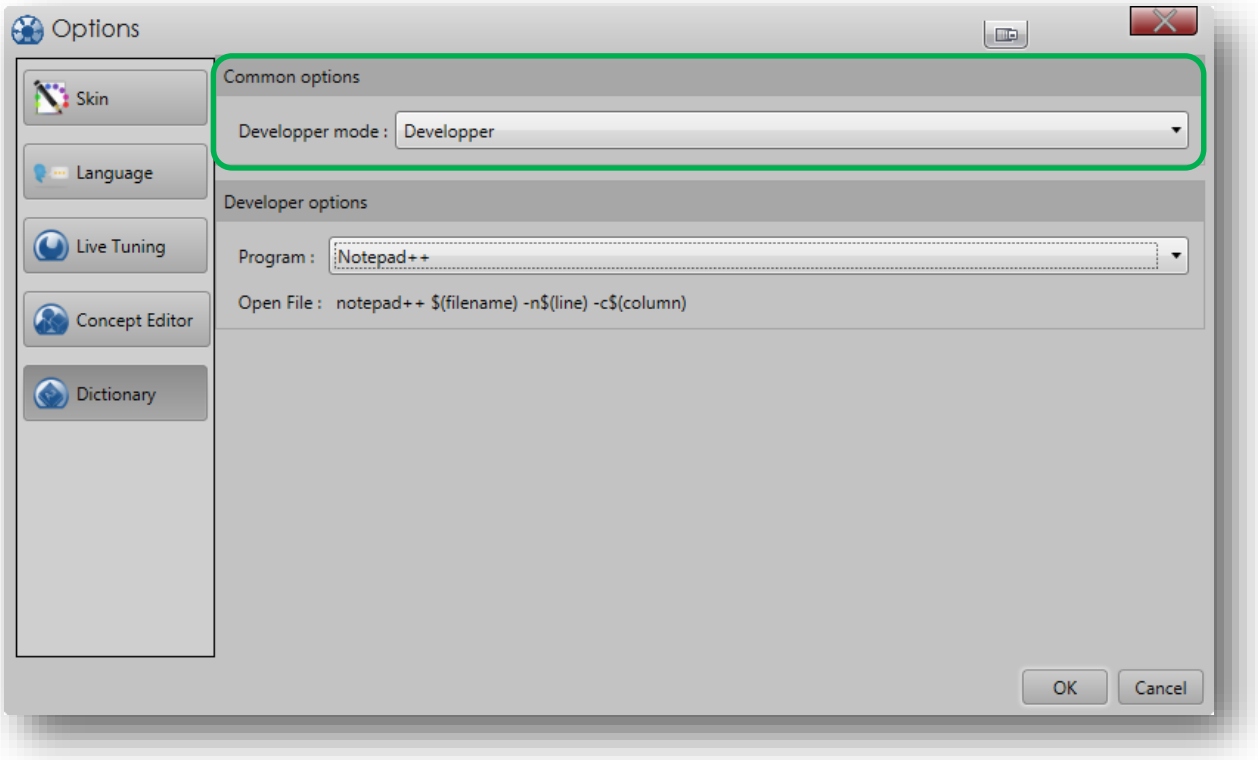


4.3. Changing side

When opening an existing Dictionary project, you might find yourself in the wrong mode. To change it, go to the Options menu (*Tools* -> *Options*) and select the *Dictionary* tab.

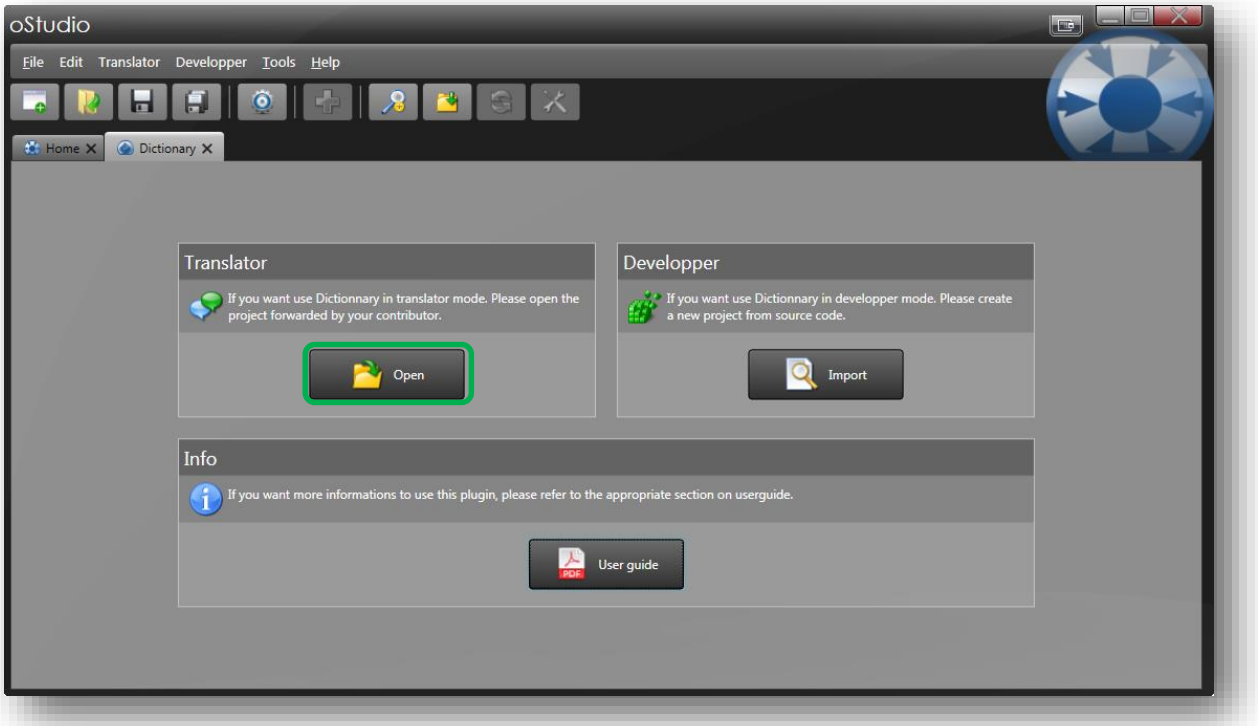


In this tab, under Common options, you can choose between Developer and Translator mode. Click OK to apply your changes.

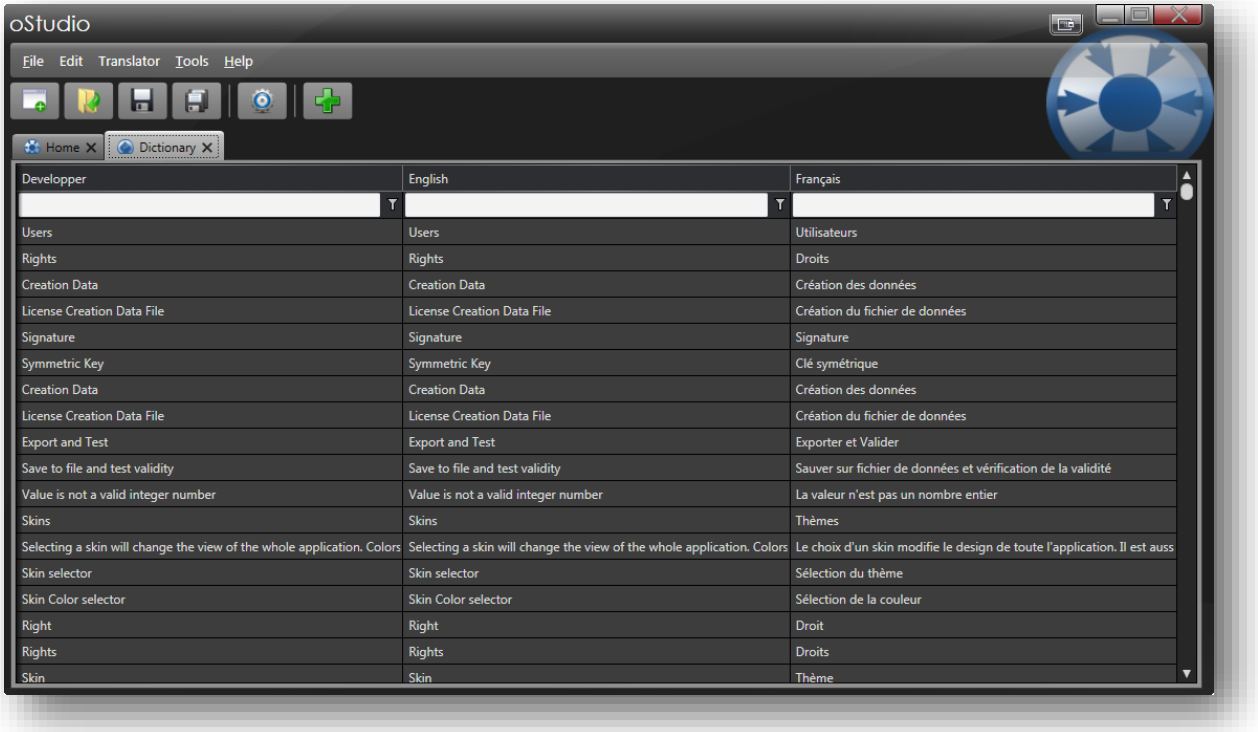


4.4. Translator side

If you work as a translator, you should have a Dictionary project file (.dic) in your possession.



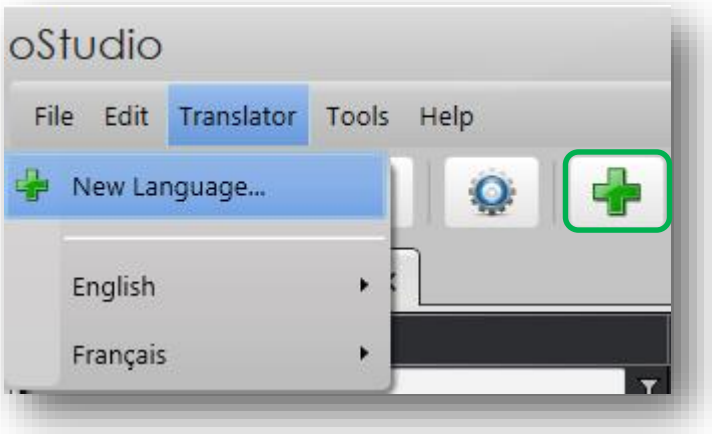
Click on *Open* to choose this file and start editing it.



Dealing with languages

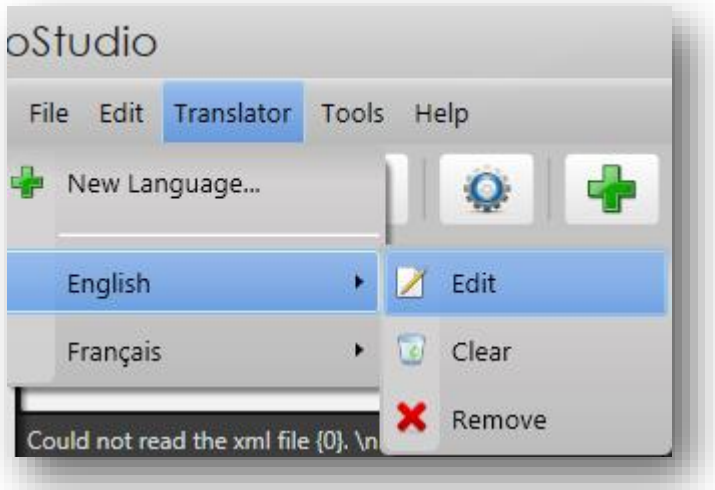
Add a language

To add a new language you can either click on *Add* or access the *Translator* menu and press on *New Language*.

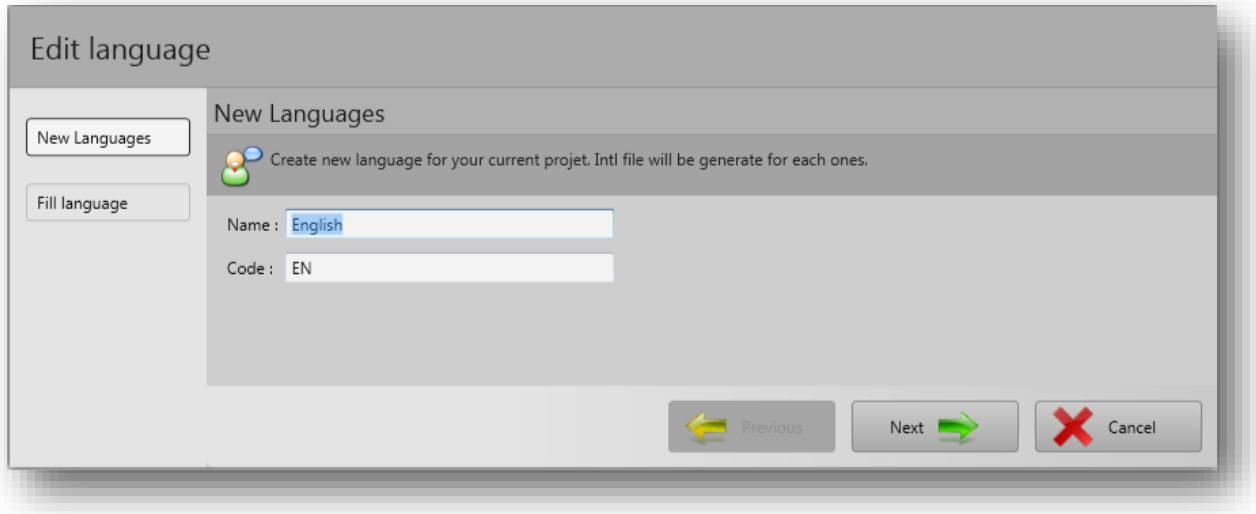


Edit a language

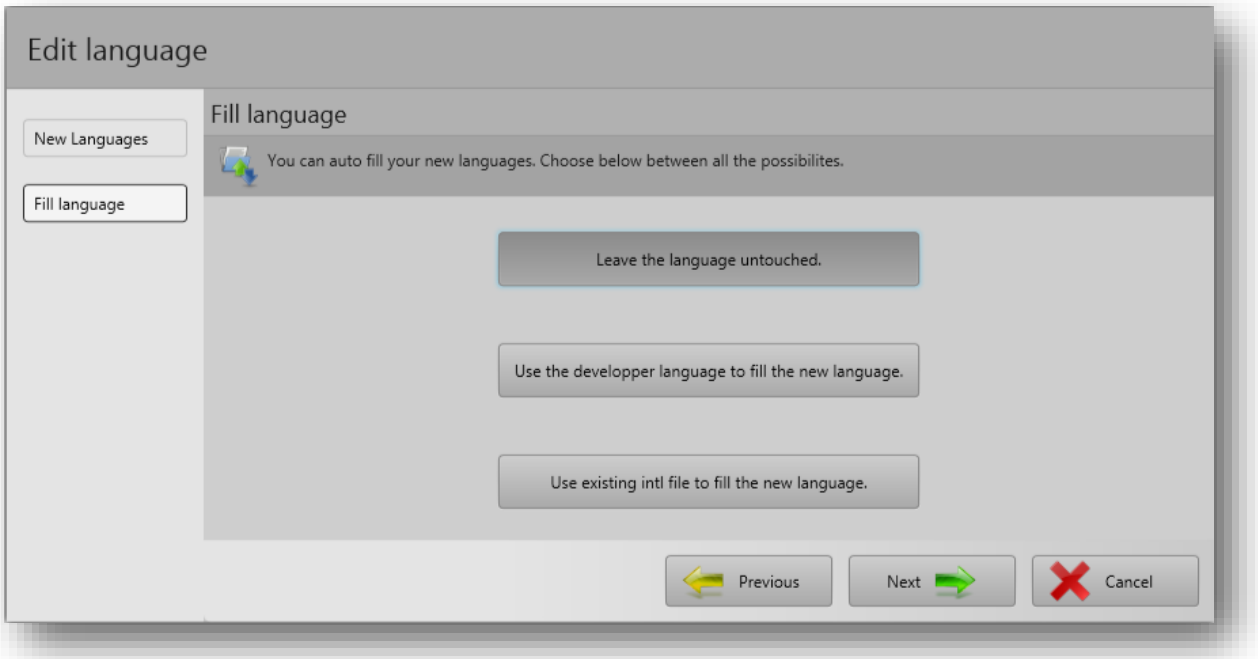
If you would like to edit an existing language you can do so by accessing the following menu Translator -> Language to edit -> Edit.



In the next screen you can change the name and code of the language.



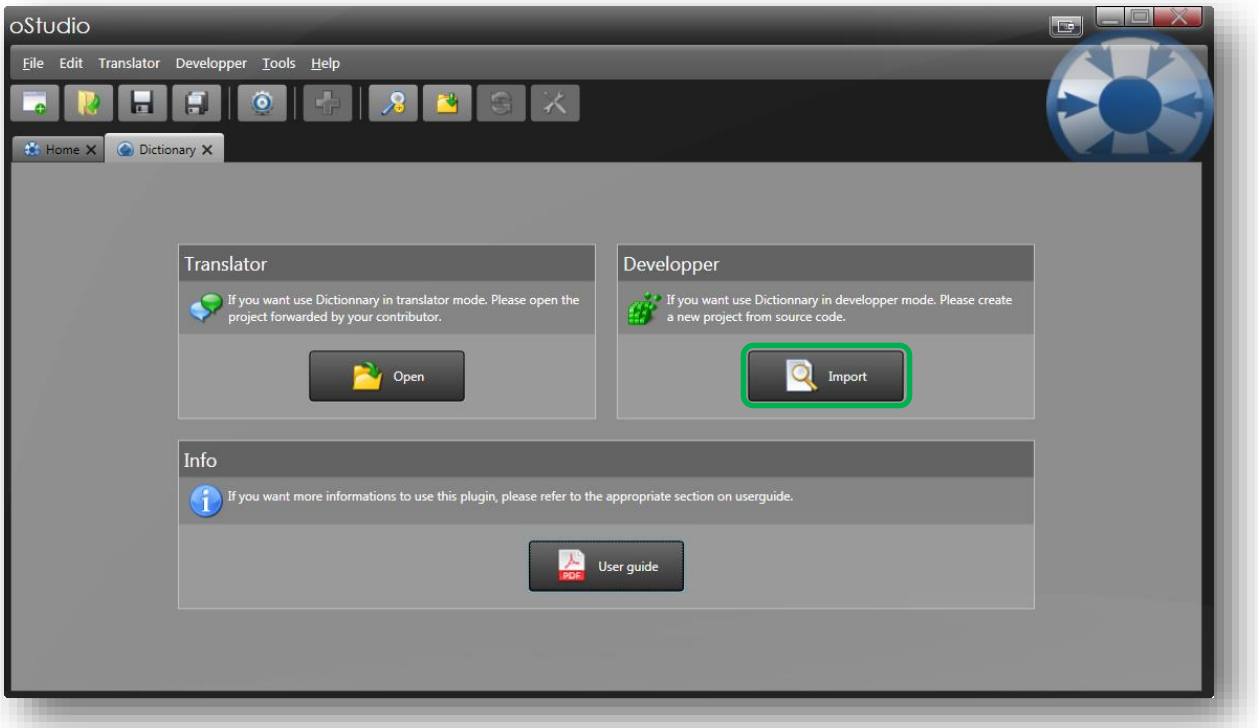
Then you can either leave the language untouched, fill the edited language with the developer language or choose an existing translation file to fill the edited language.



4.5. Developer side

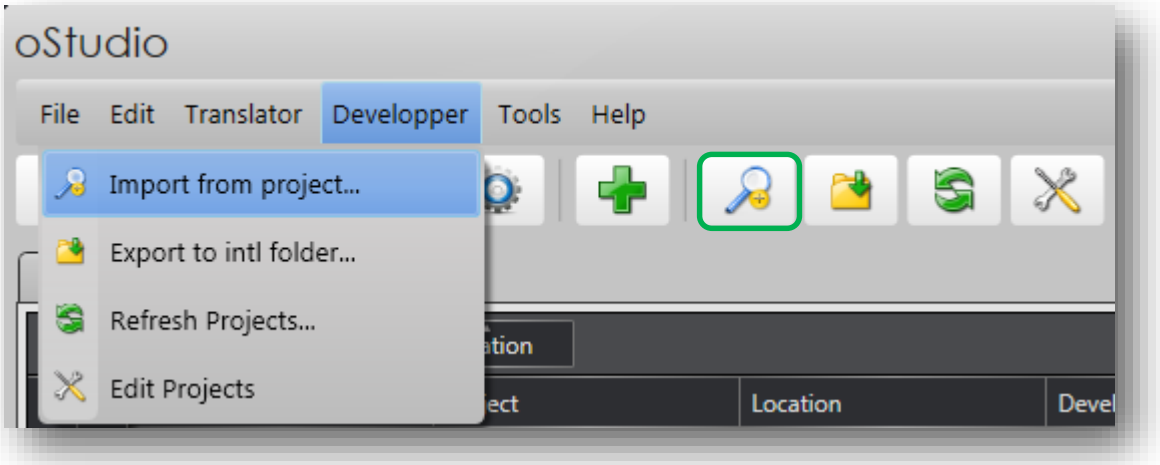
The developer side is here to simplify the creation of translations files based on the source code of the application.

If you work as a developer and would like to set up the internationalization in your application click on *Import*.

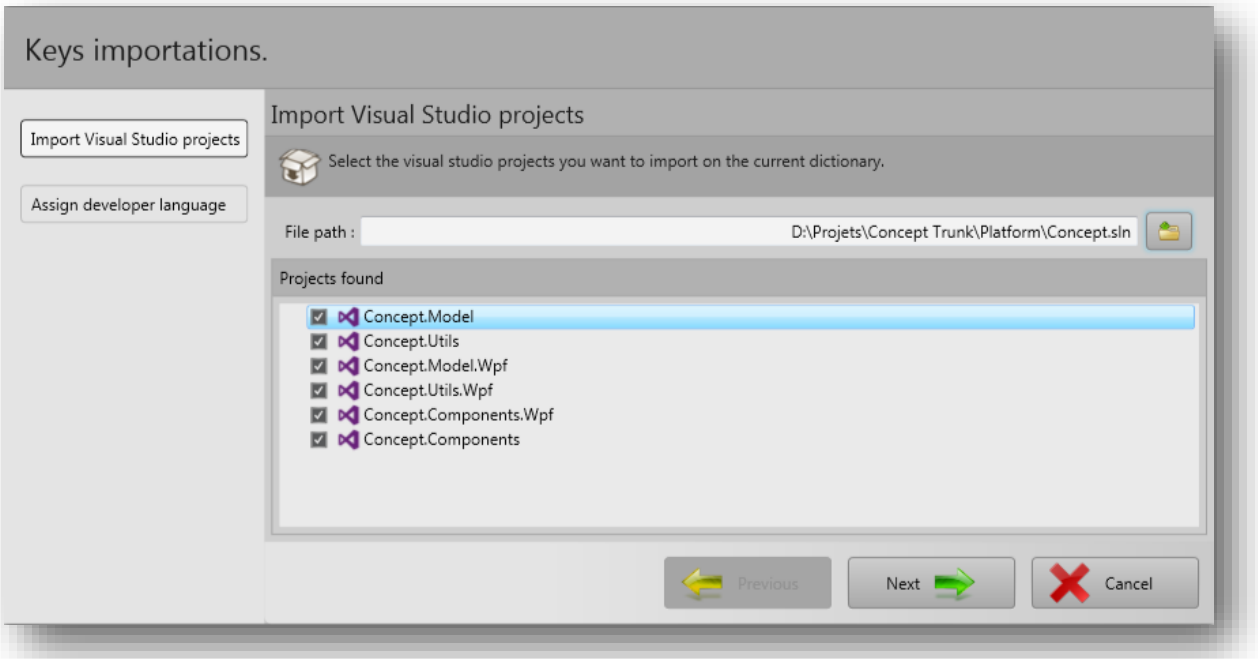


Visual Studio projects import

If you would like to add Visual Studio projects to an existing Dictionary project click on the following button or go to the following menu *Developer -> Import from project*.

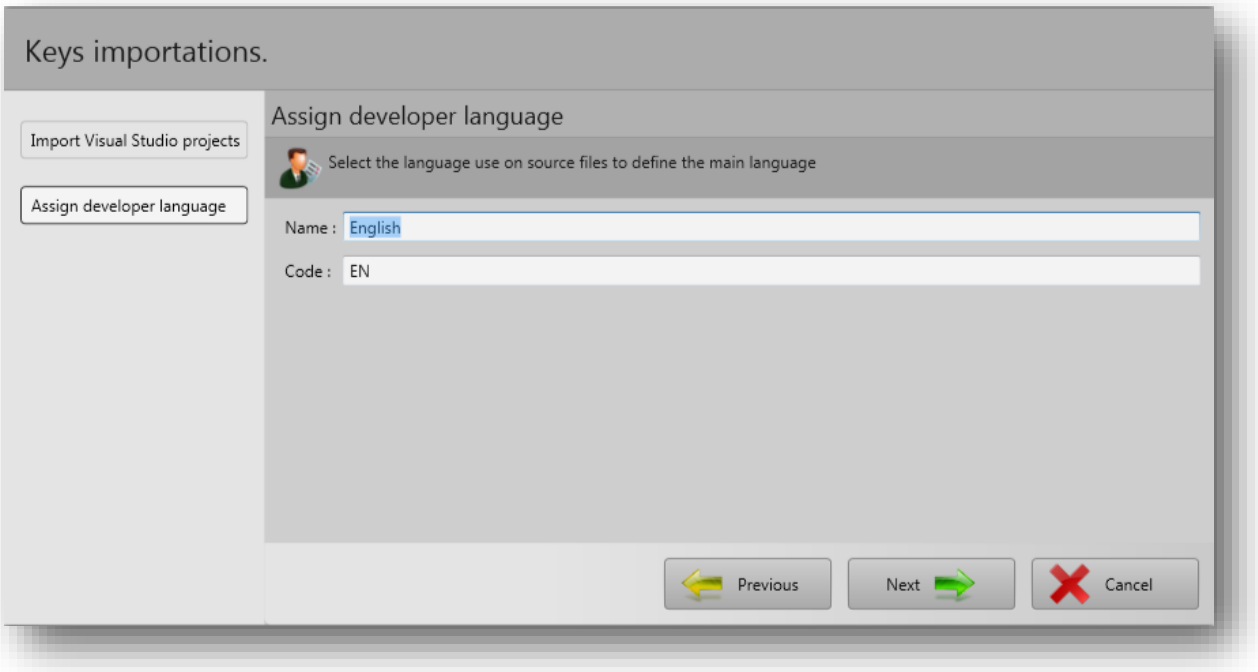


To import Visual Studio projects to the Dictionary, you can either choose a project or a solution, the projects contained in it will be detected automatically.



Then you can select the projects you which to include in this Dictionary. For example it is better to separate the main projects of the application from the plugins to keep modularity.

Click *Next* to confirm your choice. Then define the developer language, by specifying a name and a code. This language will be automatically created.

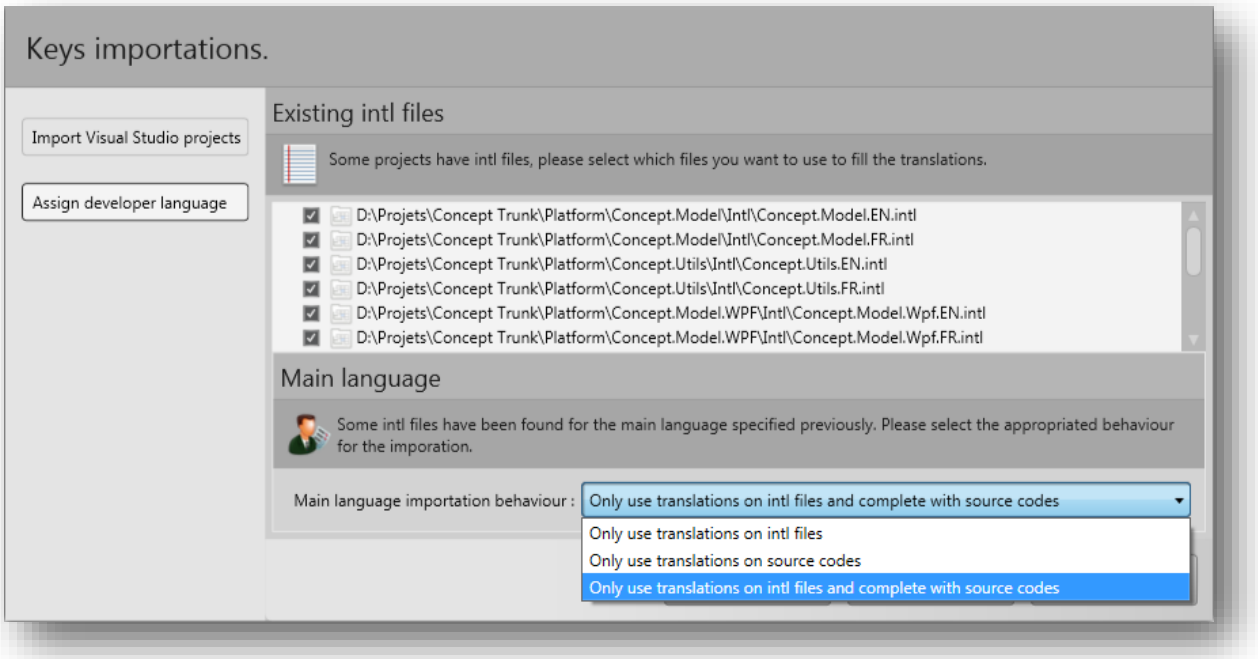


The next screen appears if existing translation files have been found throughout the projects.

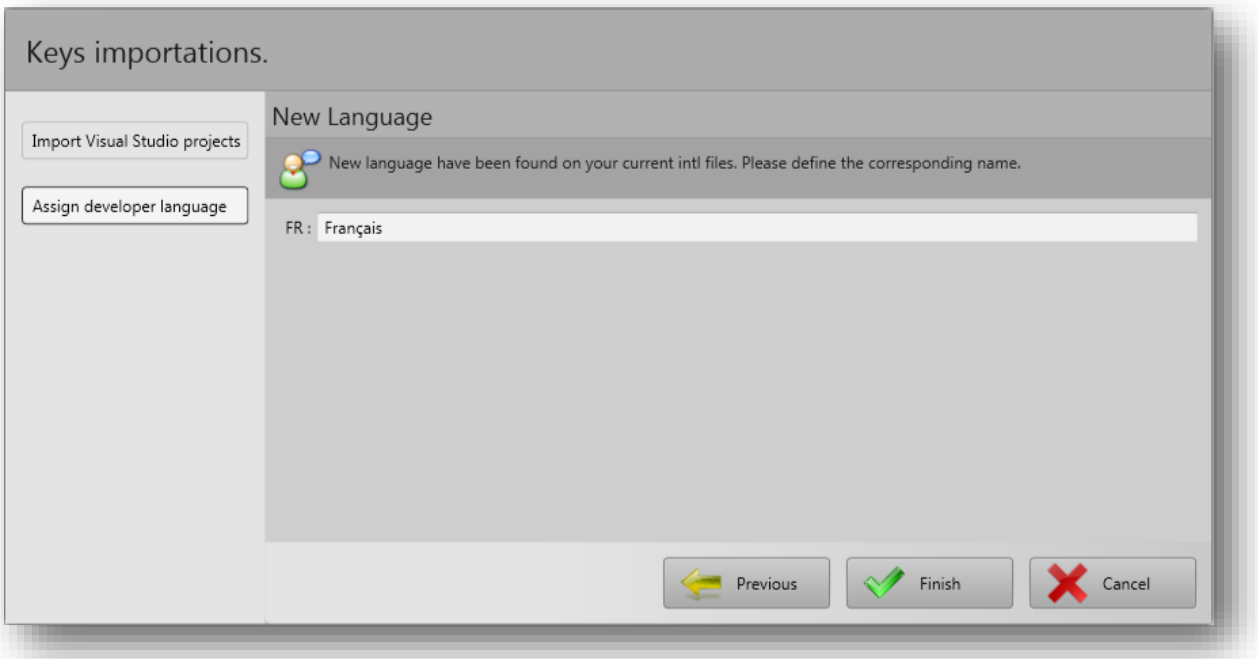
You can choose which translation files you would like to use in the current Dictionary project.

If files containing translations for the developer language have been found, you can choose between the following importation behaviours:

- ✓ Only use translations on intl files: the exact content of the existing translation files will be copied in the current Dictionary.
- ✓ Only use translations on source codes: the content of the existing translation files will be ignored, the translations defined in source codes will be used to fill the current Dictionary.
- ✓ Only use translations on intl files and complete with source codes: the content of the existing translation files will be used to fill the current Dictionary, missing translations will be filled with the source codes translations.



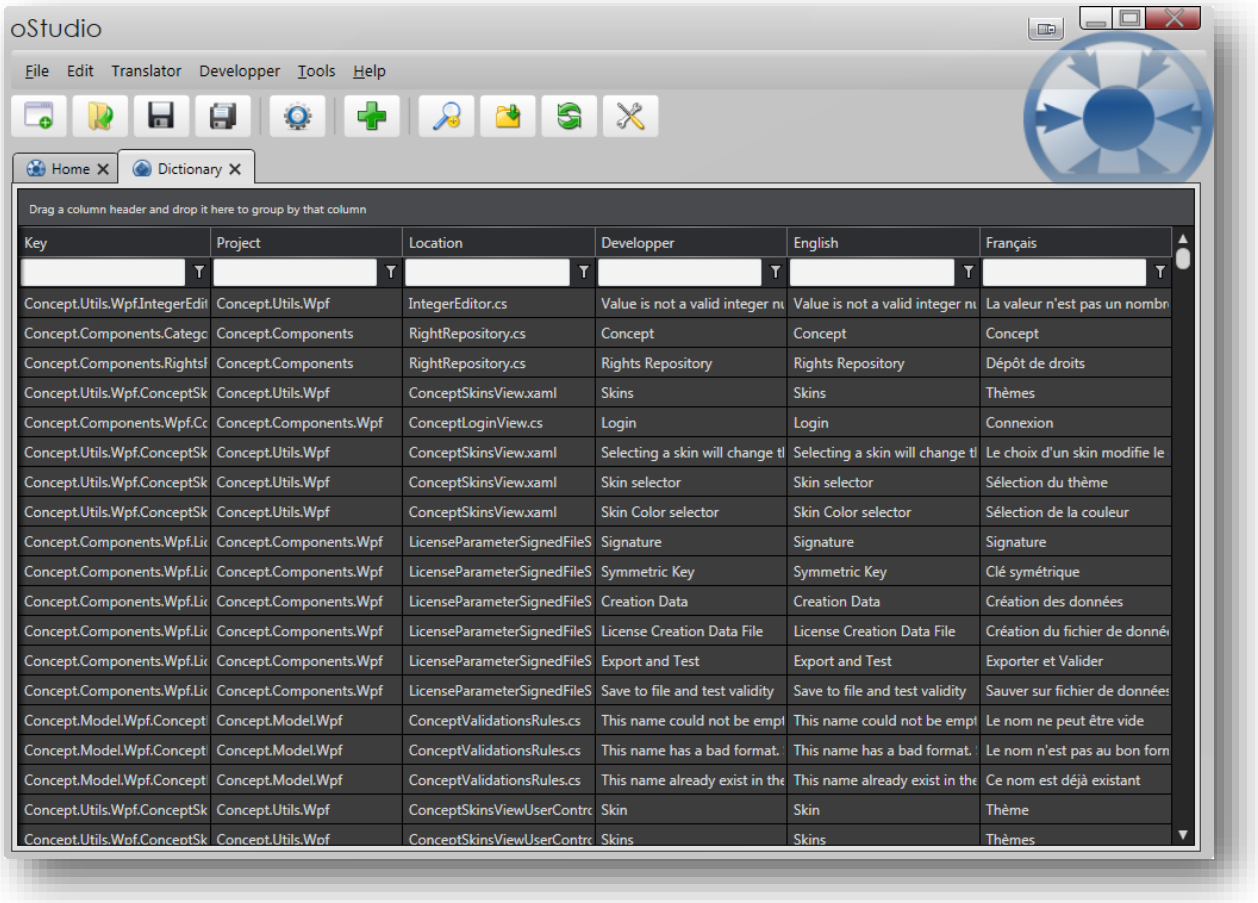
If the existing translation files contained a language not yet defined in the current Dictionary, you will be asked to define the name corresponding to the existing language code.



You can then press *Finish* to validate the import.

Translations management

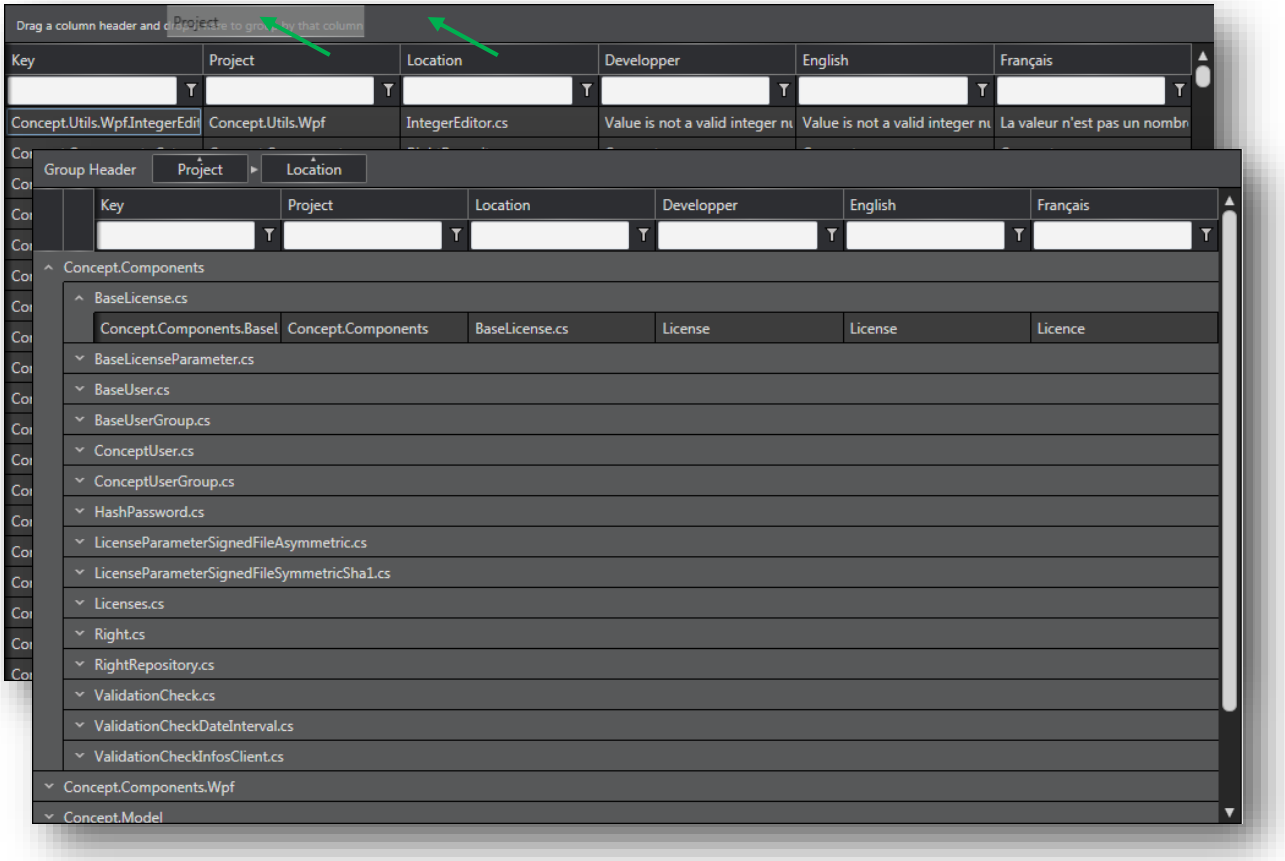
Once you have imported your Visual Studio projects, the following screen will appear.



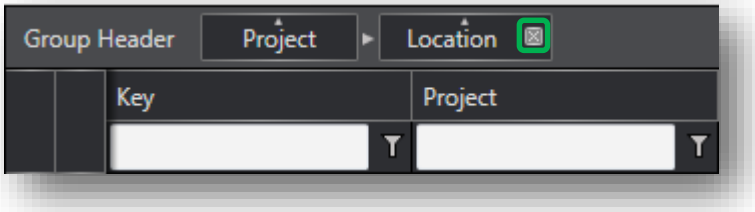
For each translation the associated key project and location is displayed. For further information about the sorting, searching and selection possibilities go to the chapters 5.2-5.4.

Advanced possibilities

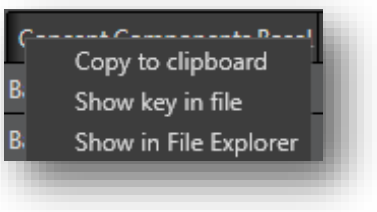
To have a better overview of the translations, the columns headers *Key*, *Project* and *Location* can be dragged in the table header to obtain a hierarchical display of the data as shown below.



To come back to the flat display, click on the cross of each group in the group header.



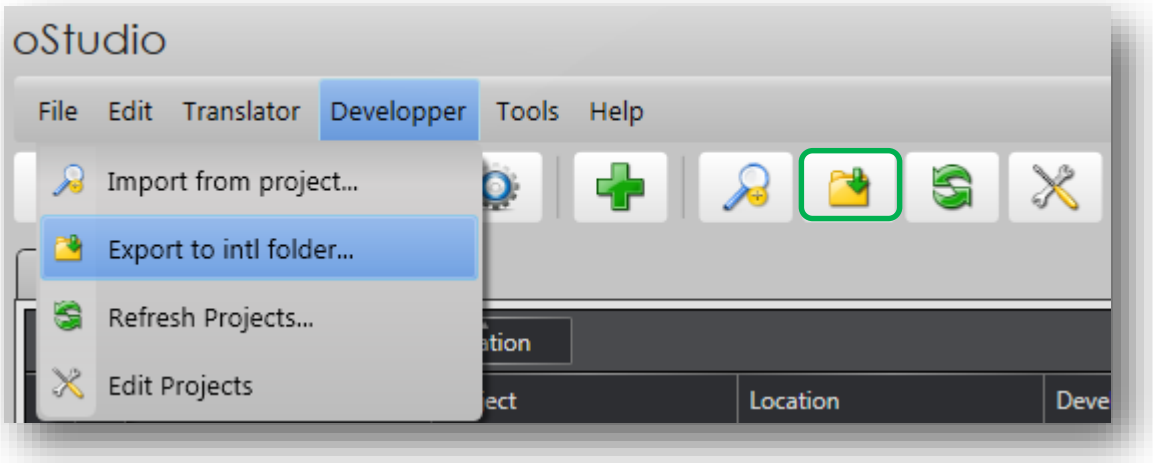
On a left click on a translation row you have access to the following functionalities.



- Copy to clipboard: Copy the key to the clipboard
- Show key in file: Open the source file containing the key definition
- Show in File Explorer: Open the folder where the source file containing the key definition is located.

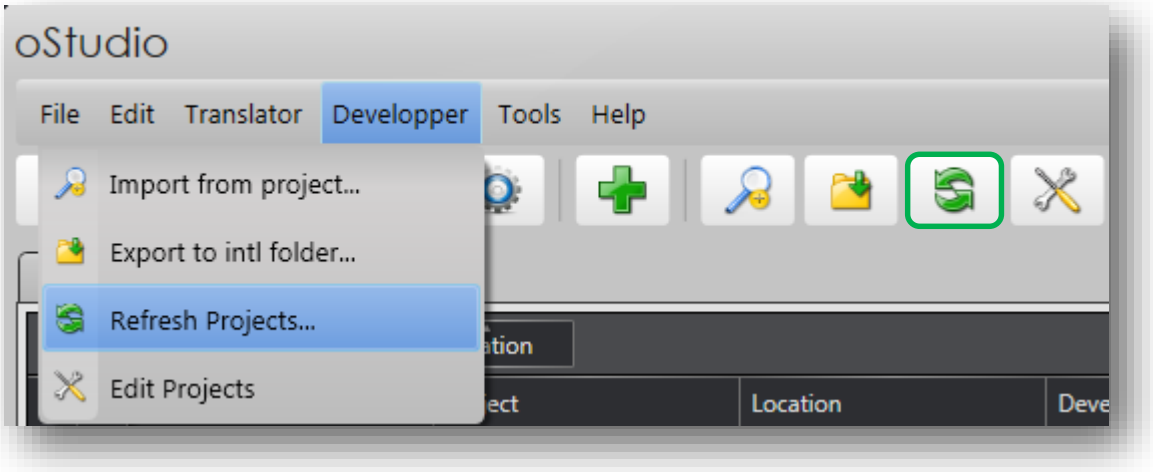
Update or create translation files

Once the modifications on the translations are done, click on the following button or go to the following menu *Developer -> Export to intl folder* to update or create the translation files of each imported Visual Studio projects.

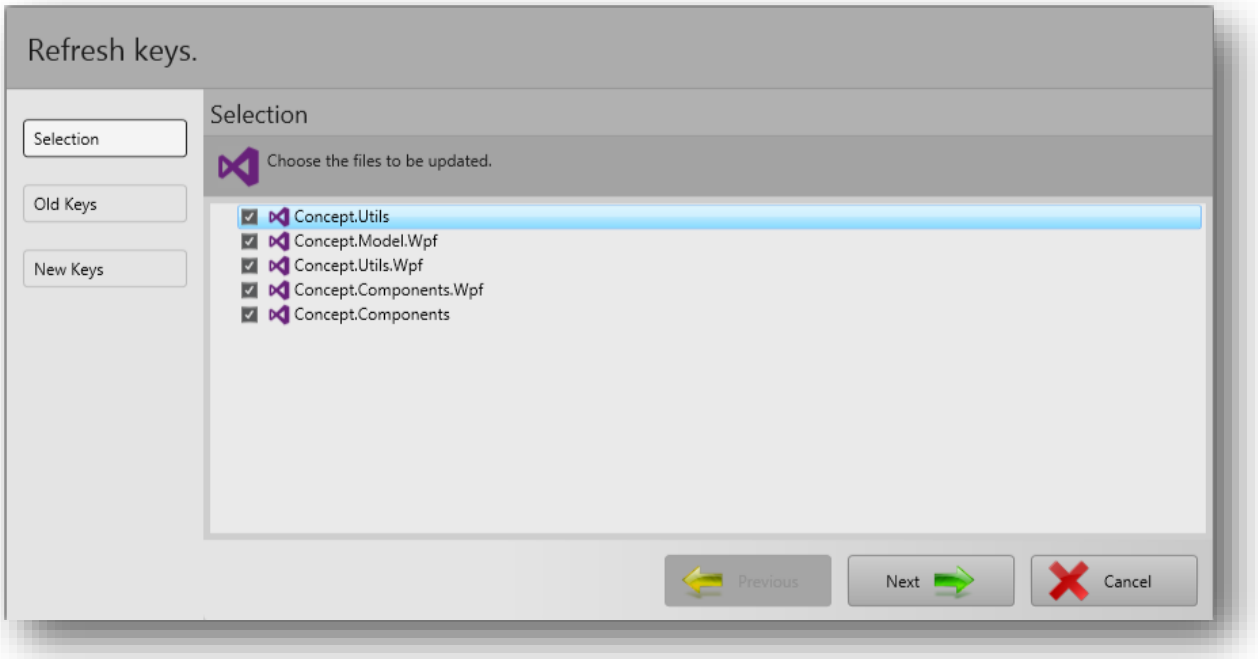


Update Dictionary from Visual Studio projects

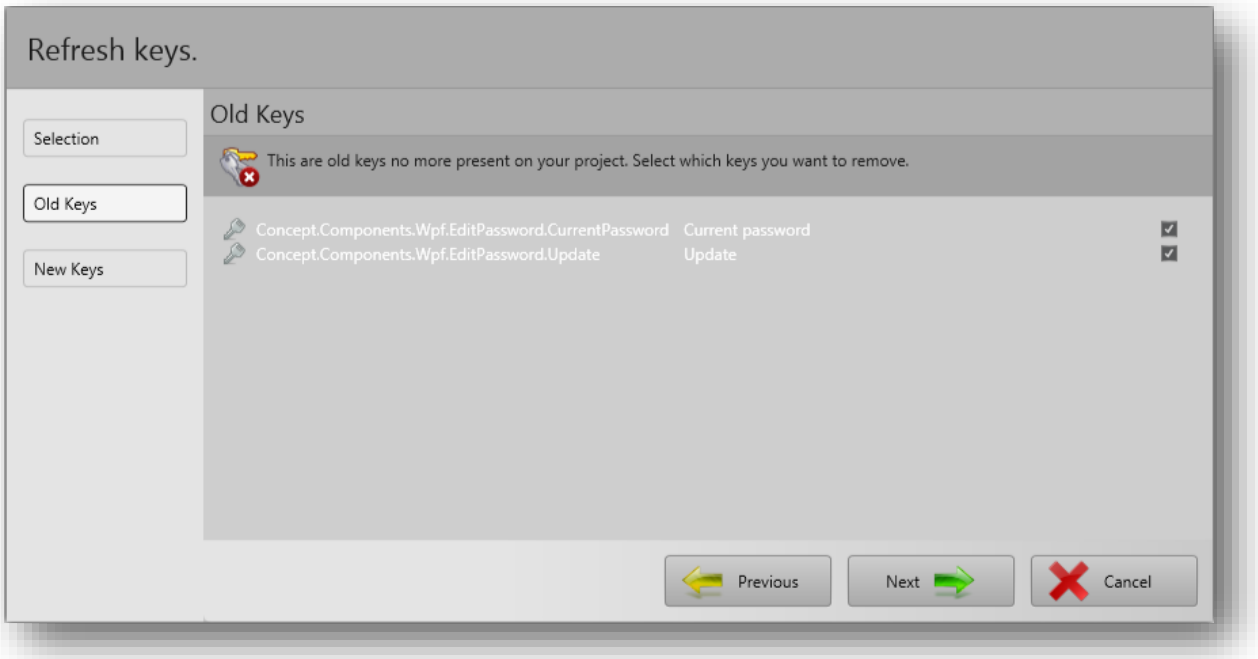
If your Visual Studio projects have changed since the last time you updated your Dictionary, click on the following button or go to the following menu *Developer -> Refresh Projects* to update translations based on source codes.



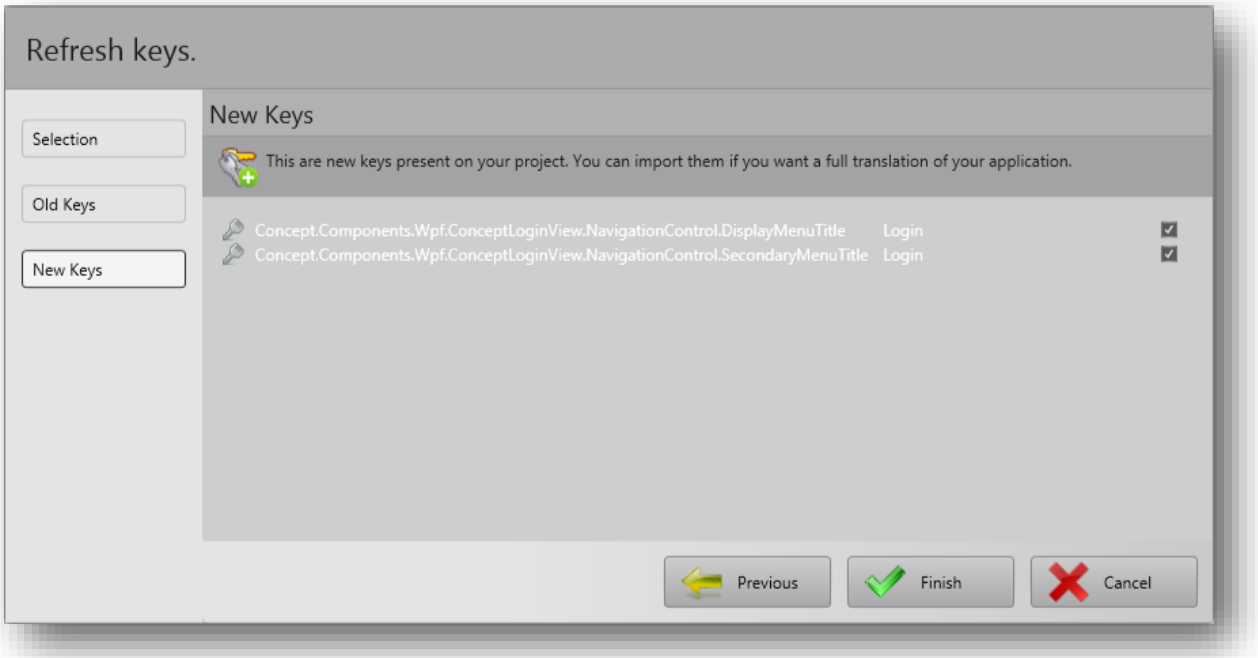
On the next screen, select the project(s) you would like to update from, then press *Next*.



The following screen shows keys no longer used in the Visual Studio projects, you can select the ones you would like to remove from the current Dictionary.

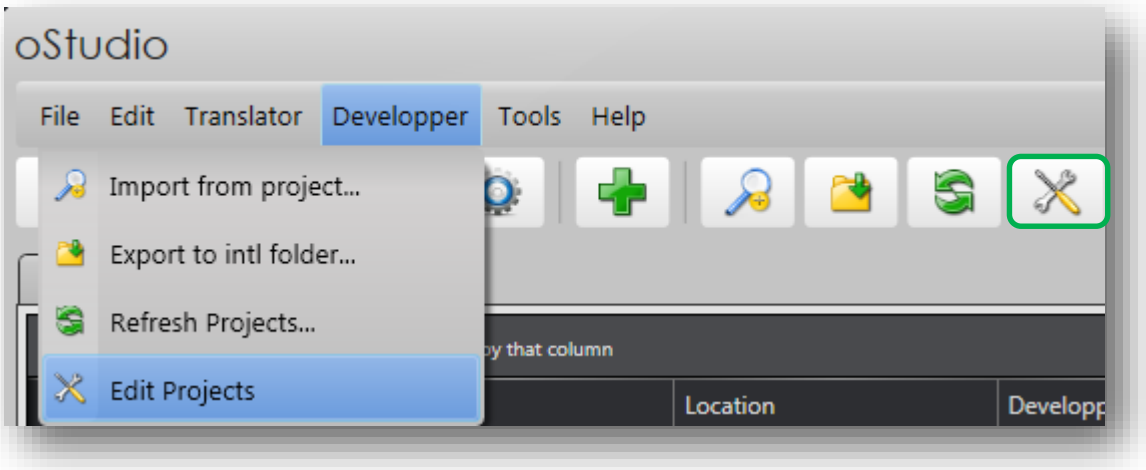


This screen shows keys contained in the Visual Studio projects but not in this Dictionary. Select the keys you would like to add to the current Dictionary. Press *Finish* to end the procedure.

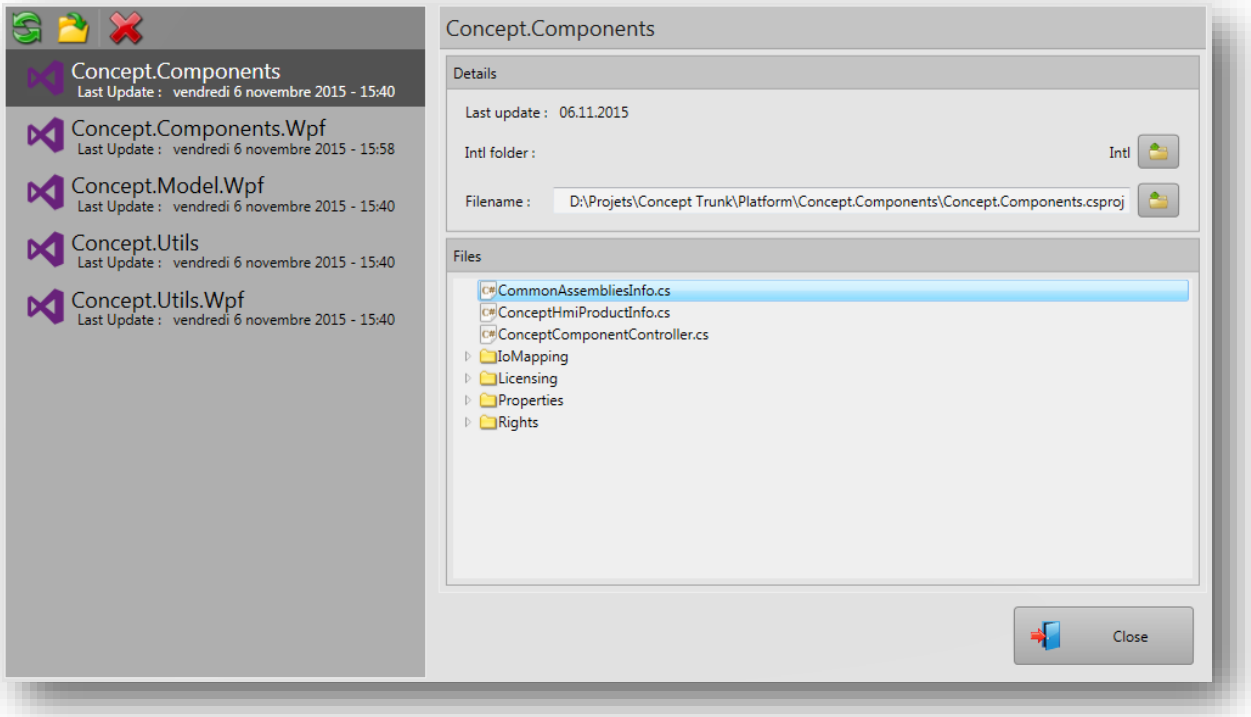


Linked Visual Studio projects management

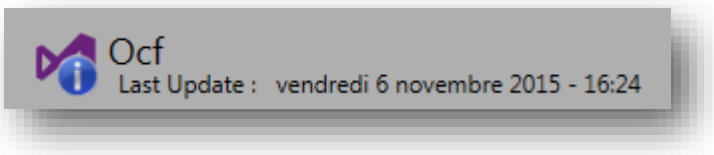
Information on the Visual Studio projects imported previously is kept in the Dictionary project. In case you have been sent a Dictionary project from somebody else, the information about the imported Visual Studio projects might be wrong. To edit linked Visual Studio projects click on the following button or go to the following menu *Developer -> Edit Projects*.



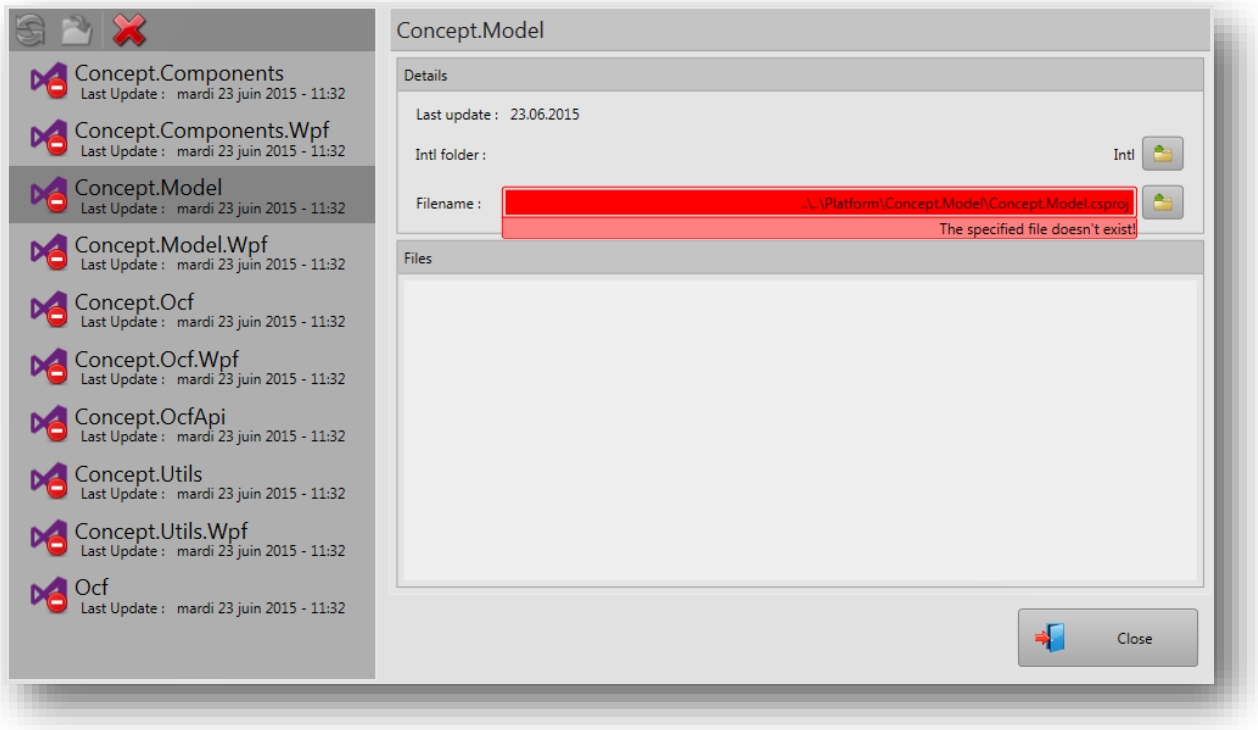
On the left of the following screen you can see the linked Visual Studio project. From this list you can either remove, refresh or open the selected project in the file explorer. The right part of the screen tells you when the last update of this project was done and where the project and linked translation file are located.



If the project does not have a linked translation file yet the following icon will appear next to the concerned project.



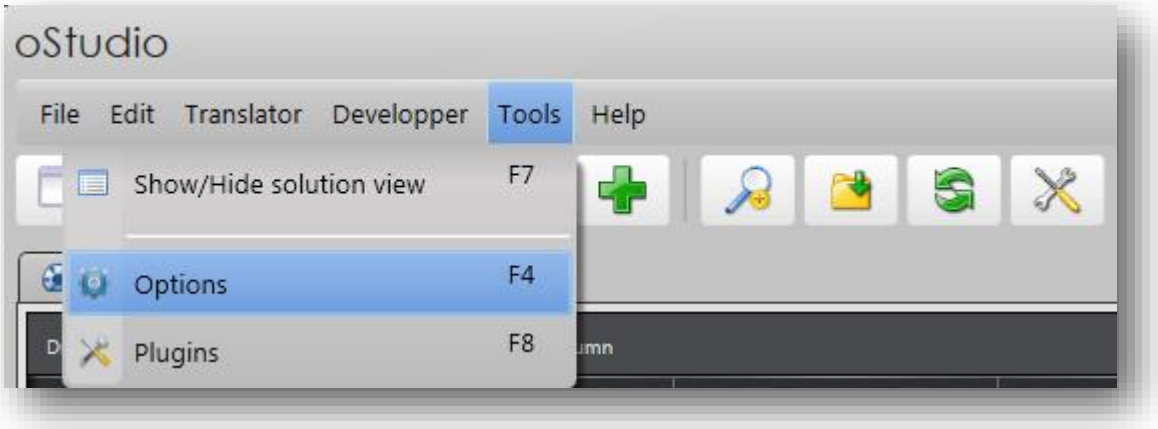
The following screen shows what can happen if information about the linked Visual Studio projects is wrong.



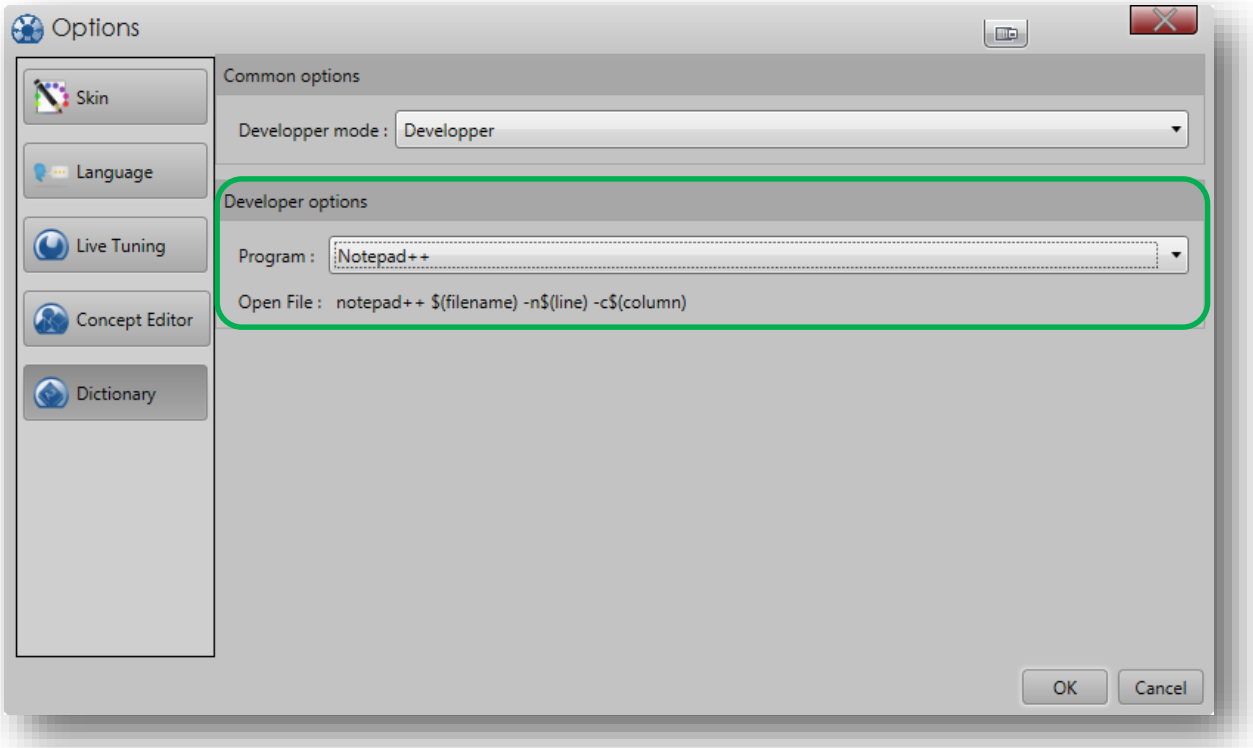
In this case, change the file path to the location of your projects.

Options

To access to developer options for Dictionary projects go to the following menu *Tools->Options* and select *Dictionary*.



Under « developer options » you can choose which program to use when requesting to show a key in its file (Chapter 6.2 Advanced possibilities).



5. Import & Export

When deploying an application it can be useful to regroup the translations in a single file. To do so, use the *Import / Export* functions available in the following menus *Edit -> Import from* & *Edit -> Export to* you can choose between .csv and .intl formats.

Be aware that regrouping translations into one single file may not be appropriate if the application can be extended with plugins. In this case, you may use the default Intl files classification explained further in the document (6.2).

6. Annex : *Concept.Intl* best practices

When developing an application with Concept, consider using translatable texts (Intl) from the very beginning.

Here are a few other tips on how to manage internationalization efficiently in your C#/WPF projects using Concept.

6.1. Key naming convention

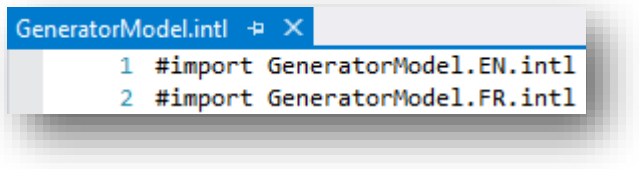
When declaring a new translation in C#, it is important to define a unique key using a fixed naming convention, below is the one we recommend.

Namespace.Class.TextDefinition

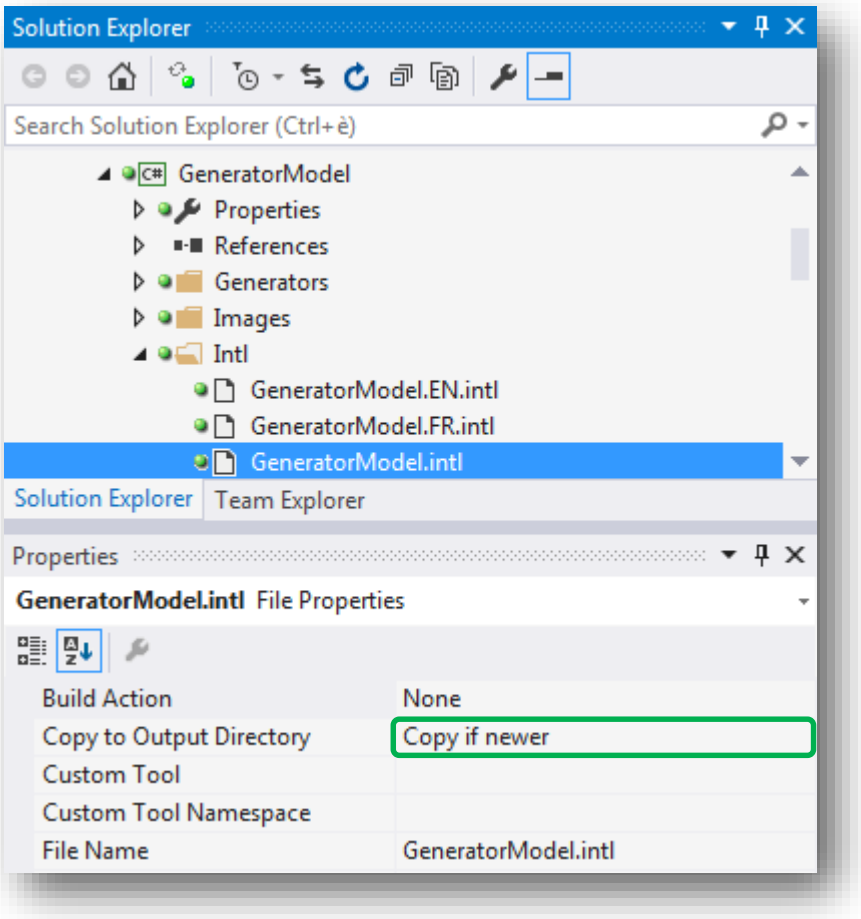
For common translations such as validation texts like "Ok" and "Cancel", we recommend creating these once in a static class.

6.2. Intl files classification

The oStudio plugin Dictionary generates one translation file for each language and one additional file referencing them. Below is an example of such a file.

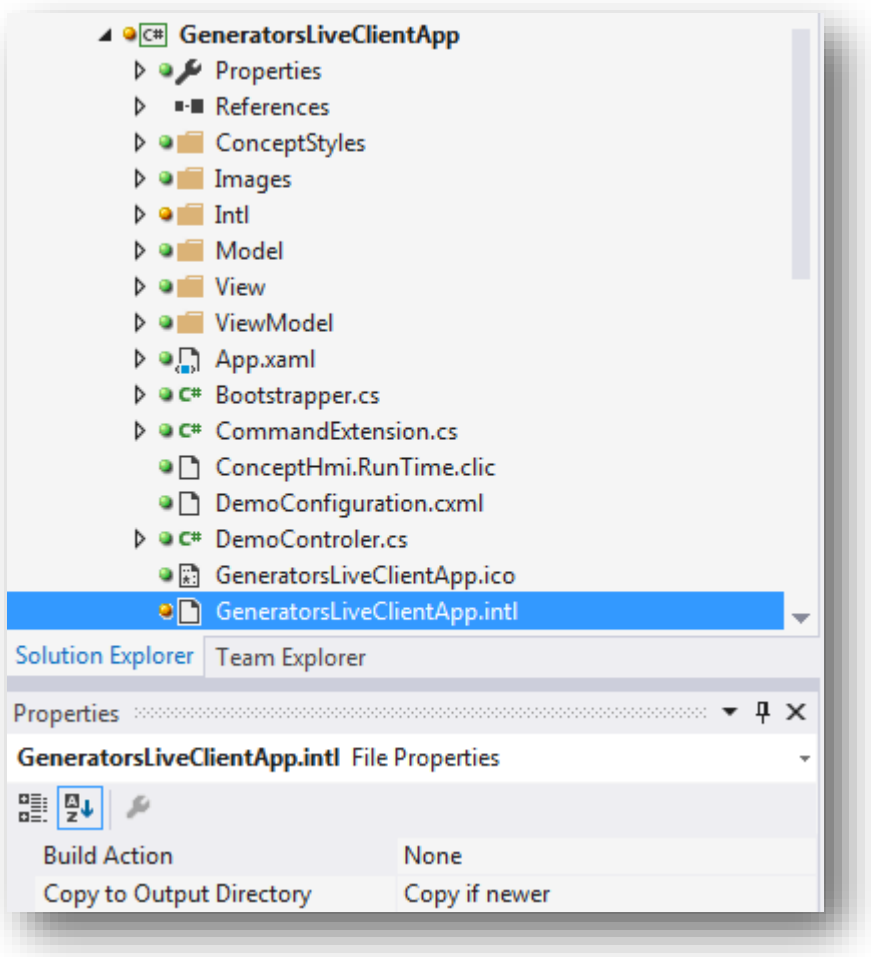


In a Visual Studio solution, each project has an Intl folder containing the « .intl » files generated by the oStudio plugin Dictionary. In the *Properties* tab of Visual Studio set *Copy to Output Directory* to *Copy if newer* for each of these files.

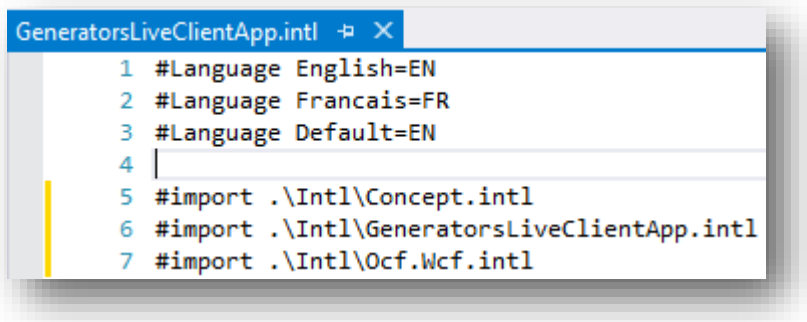


This way each project of the solution will have their translation files copied in a folder "Intl" in the application's output directory.

The startup project of your solution must contain an .intl file containing references to every translation files necessary for the application. This file also has to be configured as Copy if newer in its properties, as shown below.



It is important that this file be copied next to the executable of the application. Below is an example of the content of this file.



The first three lines gives the language name corresponding to each language code and the default language of the application. The next lines describe the path to the main translation file (the file referencing every languages files) of each project contained in this application.

Plugin translation management

A plugin project has to have the same translation structure than a normal project that is a folder named "Intl" containing the generated translation files with their properties set to *Copy if newer*.

Concept libraries translation management

Concept Hmi libraries come with an Intl folder. The content of this folder has to be added with the other translation files of the application. To do so, you have two possibilities.

The first one is to add these files in the folder Intl of the startup project and copy them to the binaries folder (Copy if newer setting).

The second one is to add a post-build event command in the startup project properties to directly copy the content of the Intl folder located next to the Concept Hmi libraries. Here's an example of such a command.

```
xcopy /y /d  "$(ProjectDir)Libraries\Intl" "$(ProjectDir)$(OutDir)\Intl"
```