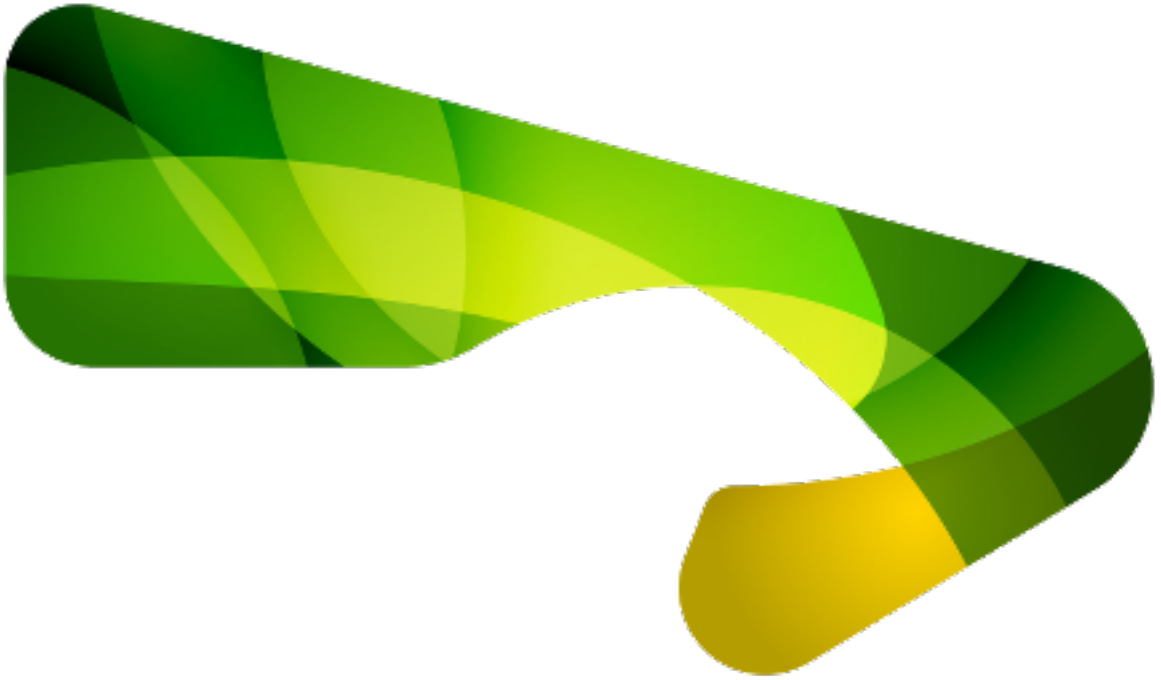


# Fluenta



*Copyright © 2015-2025 Maxprograms*

# Table of Contents

<b>Introduction .....</b>	<b>1</b>
<b>DITA Projects .....</b>	<b>2</b>
Add Project .....	2
Edit Project .....	4
Remove Project .....	5
Generate XLIFF .....	5
Import XLIFF .....	7
Project Status .....	9
<b>Translation Memories .....</b>	<b>11</b>
Add Memory .....	11
Edit Memory .....	12
Import TMX File .....	13
Export Memory as TMX File .....	13
<b>Advanced Configuration .....</b>	<b>14</b>
General Options .....	14
XML Options .....	15
Configuration Files .....	16
Add Configuration File .....	16
Edit Configuration File .....	18
Remove Configuration File .....	18
XML Catalog .....	19
<b>Subscriptions .....</b>	<b>20</b>
First Registration - Evaluation Request .....	20
Register a Subscription Key .....	21
<b>Command Line Interface .....</b>	<b>22</b>
Create Project .....	22
Remove Project .....	23
Retrieve Project List .....	23
Generate XLIFF Files .....	25
Import XLIFF File .....	26
Create Memory .....	27
Remove Memory .....	28
Retrieve Memory List .....	28
Import TMX File .....	29
Export TMX File .....	29
Subscription Management .....	29
Retrieve Version Information .....	29
<b>Licenses .....</b>	<b>31</b>
Glossary .....	i

# Introduction

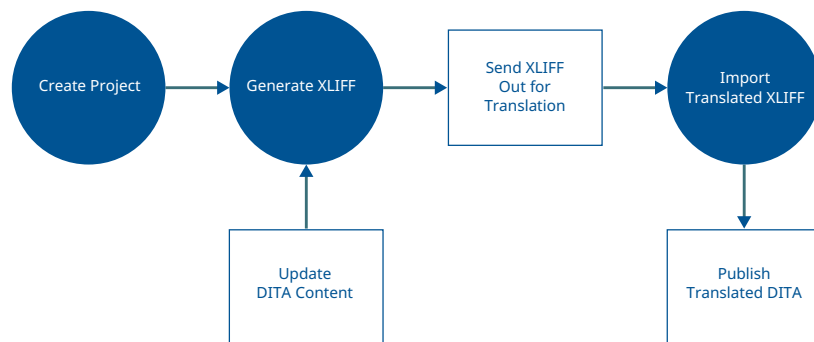
Fluenta is a tool designed to simplify the translation of DITA projects. It can parse a DITA map, resolving the references to all topics and subtopics, preparing a unified [XLIFF](#) file that you can send to your [Language Service Provider \(LSP\)](#).

Fluenta implements the procedure for translating DITA projects recommended by the [OASIS DITA Adoption TC](#).

## How it works

1. Start by [creating a project](#). All you have to do is provide the location of your DITA map and select the languages that you want to translate into.
2. When you are ready to translate your project, [generate an XLIFF file](#) from it.
3. Send the XLIFF file to your [Language Service Provider \(LSP\)](#) and wait for a translated XLIFF.
4. [Import the translated XLIFF](#) and select a folder where to store the translated version of your map and topics.

The four steps described above are all you need to get a translated version of your DITA project. The following diagram shows the processes involved:



After updating your DITA content, you may want to update the translations of your project. All you have to do at this moment is:

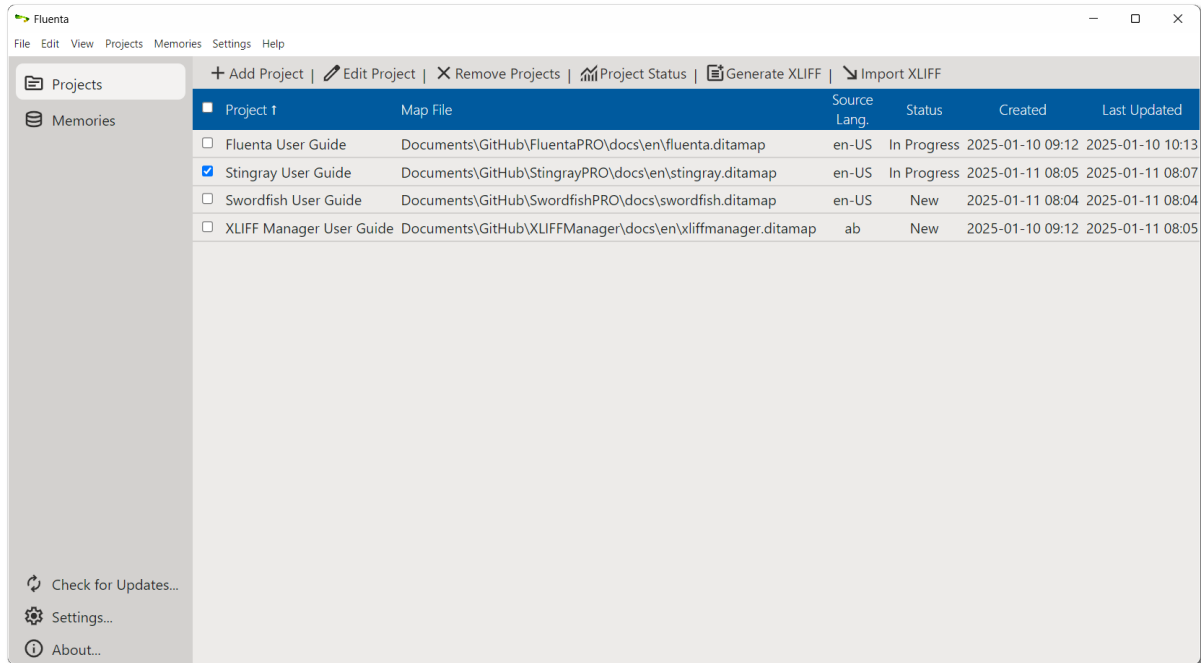
1. Generate a new XLIFF file.
2. Send the new XLIFF file to your Language Service Provider and wait for a translated XLIFF.
3. Import the translated XLIFF and select the folder where to store the updated translated version of your map and topics.

Fluenta automatically recovers In-Context Exact (ICE) matches from the translation that was stored in the previous cycle. This means that you don't need to pay again for the translation of content that didn't change.

Thanks to the [Translation Memory](#) technology included in Fluenta, you can also recover translations of parts that were slightly changed.

## DITA Projects

Fluenta organizes the translation of your DITA maps into projects. The **Projects view** in Fluenta contains a list of your existing projects, and the buttons in its toolbar allow you to manage all project-related tasks.



## Add Project

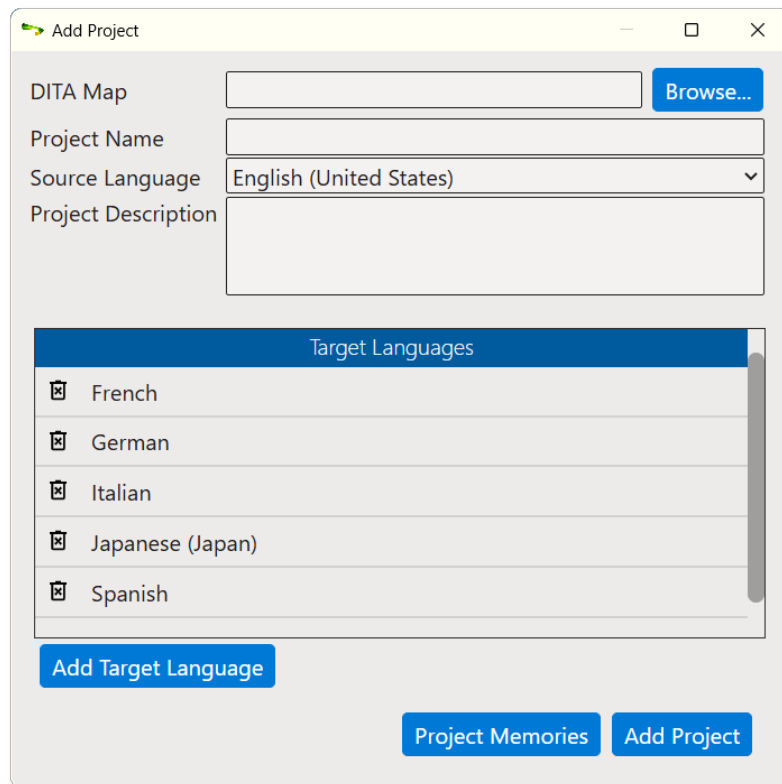
### About this task

Follow these steps to create a translation project from your DITA map.

### Procedure

1. In main menu, select **Projects** → **Add Project** or click the **+ Add Project** button in **Projects** view toolbar.

The **Add Project** dialog appears:

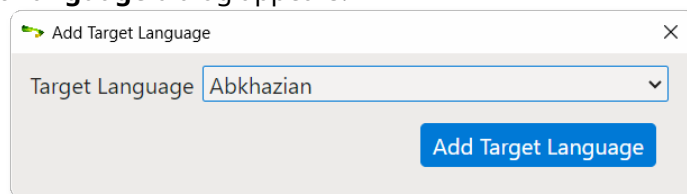


The 'Add Project' dialog box is shown. It has a title bar with a green icon and the text 'Add Project'. The main area contains the following fields and controls:

- DITA Map:** A text box with a 'Browse...' button to its right.
- Project Name:** A text box.
- Source Language:** A drop-down menu currently showing 'English (United States)'.
- Project Description:** A large text area.
- Target Languages:** A list with a blue header. It contains five items, each with a checkbox and a language name:
  - ☒ French
  - ☒ German
  - ☒ Italian
  - ☒ Japanese (Japan)
  - ☒ Spanish
- Buttons:** At the bottom, there are three buttons: 'Add Target Language' (blue), 'Project Memories' (blue), and 'Add Project' (blue).

2. Type the name of the DITA map in the **DITA Map** text box or use the **Browse...** button to select a DITA map from the file system.
3. Type a project name in the **Project Name** text box.
4. Select the source language of your project using the **Source Language** drop-down.
5. Optionally, enter a description for the project in the **Project Description** text box.
6. Add additional target languages to the default set if required.
  - a. Click **Add Target Language** button.

The **Add Target Language** dialog appears:



The 'Add Target Language' dialog box is shown. It has a title bar with a green icon and the text 'Add Target Language'. The main area contains the following controls:

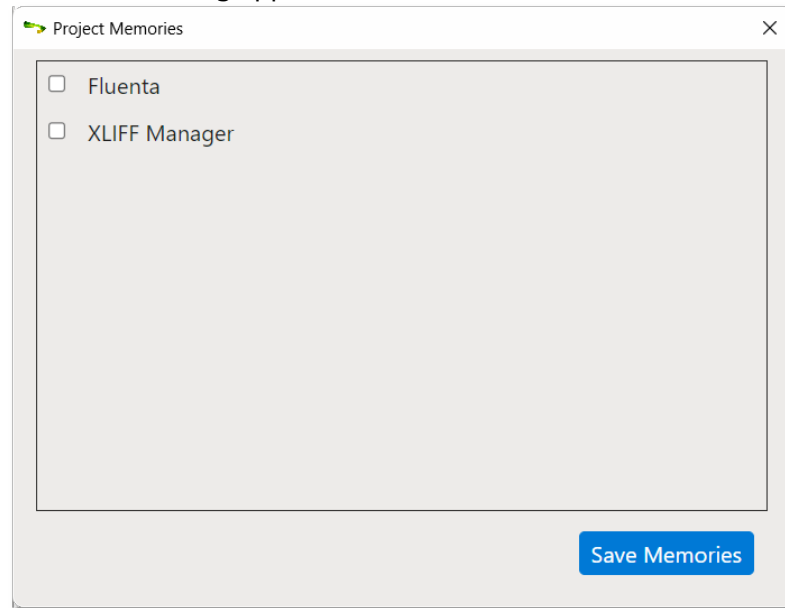
- Target Language:** A drop-down menu currently showing 'Abkhazian'.
- Button:** A blue button labeled 'Add Target Language'.

- b. Select a language from the **Target Language** drop-down list.
      - c. Click the **Add Target Language** button.
- Selected language is added to the project and the dialog is closed.
7. If you want to remove target languages from the list, click the corresponding **Remove Language** buttons (☒) on the leftmost column.
  8. Associate existing additional memories with the new project if required.

You can reuse Translation Memory data from related Fluenta projects to save on translation costs.

- a. Click on the **Project Memories** button.

The **Project Memories** dialog appears:



- b. Select the check boxes next to the existing memories that you want to associate with the new project.
- c. Click the **Save Memories** button.

Selected memories are associated with the new project and the dialog closes.

9. Click the **Add Project** button.

## Results


A new project is created and the list of projects in the **Projects** view is updated to reflect the changes.

## Edit Project

### About this task

Follow these steps to edit a translation project.

### Procedure

1. In **Projects** view, select the project that you want to edit by clicking on it.
2. In main menu, select **Projects** → **Edit Project** or click the  **Edit Project** button in **Projects** view toolbar.

The **Edit Project** dialog appears:

**Edit Project**

DITA Map:  [Browse...](#)

Project Name:

Source Language:

Project Description:

**Target Languages**

- ☒ French
- ☒ German
- ☒ Italian
- ☒ Japanese (Japan)
- ☒ Spanish

[Add Target Language](#) [Project Memories](#) [Update Project](#)

3. Edit all fields as required.
4. Click the **Update Project** button.

## Results

Selected project is updated and the list of projects in the **Projects** view is updated to reflect the changes.

## Remove Project

Follow these steps to permanently remove a Project from Fluenta.

### Procedure

1. In **Projects** view, select all projects that you want to remove by clicking on them.
2. Select **Project** → **Remove Projects** from the main menu or click the ☒ **Remove Projects** button in **Projects** view toolbar.
3. Confirm projects removal in the dialog that is displayed.

## Generate XLIFF

### About this task

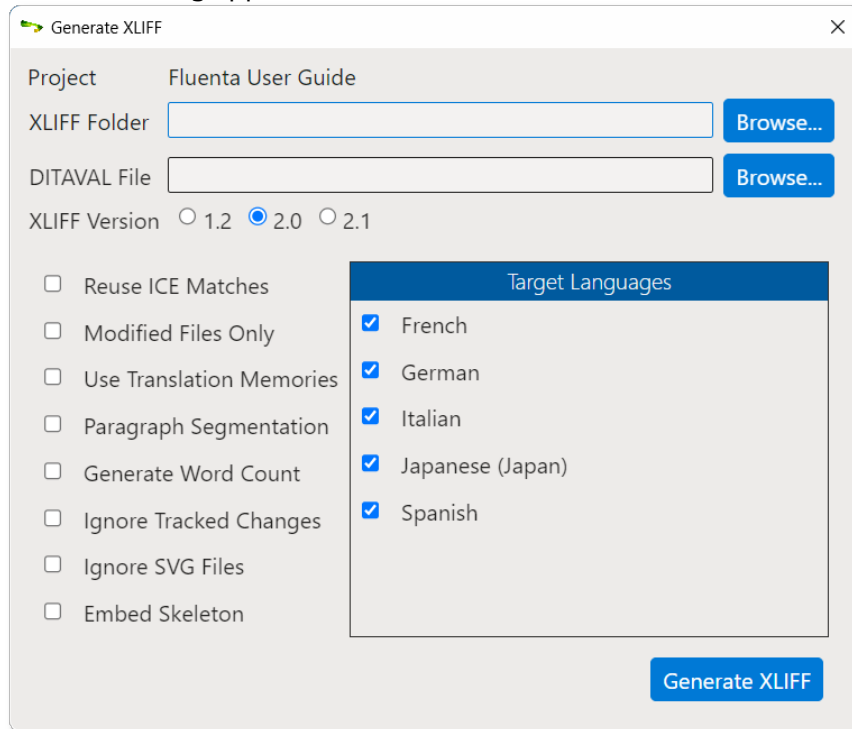
Follow these steps to generate [XLIFF](#) files that you can send to your [Language Service Provider \(LSP\)](#) for translating your DITA projects.

### Procedure

1. In **Projects** view, select the project that will be translated.

- In main menu, select **Projects** → **Generate XLIFF** or click the  **Generate XLIFF** button in the **Projects** view toolbar.

The **Generate XLIFF** dialog appears:



Project: Fluenta User Guide

XLIFF Folder:  Browse...

DITaval File:  Browse...

XLIFF Version: ☐ 1.2 ☒ 2.0 ☐ 2.1

☐ Reuse ICE Matches  
☐ Modified Files Only  
☐ Use Translation Memories  
☐ Paragraph Segmentation  
☐ Generate Word Count  
☐ Ignore Tracked Changes  
☐ Ignore SVG Files  
☐ Embed Skeleton

Target Languages

- ☒ French
- ☒ German
- ☒ Italian
- ☒ Japanese (Japan)
- ☒ Spanish

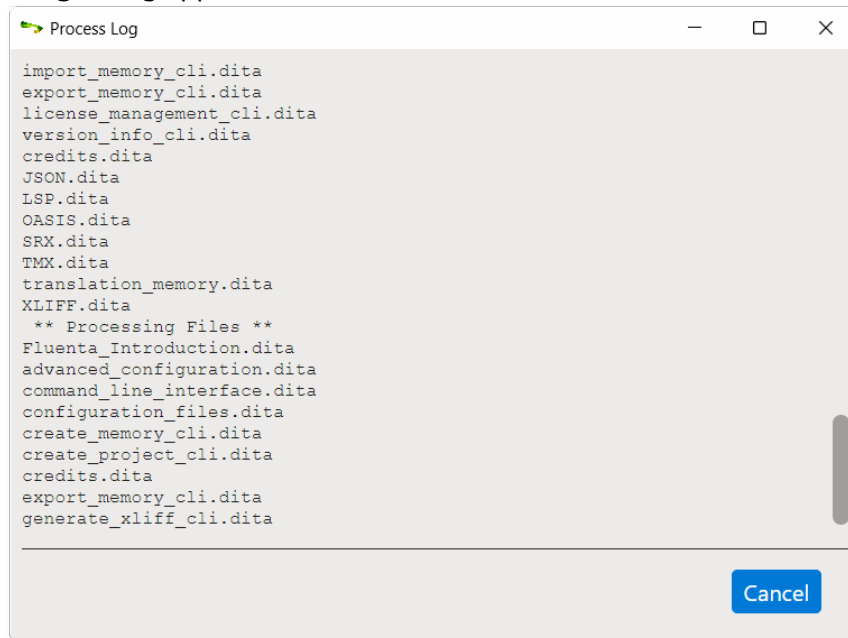
Generate XLIFF

- Type the name of the folder where the XLIFF files should be placed in the **XLIFF Folder** text box or use the **Browse...** button to select a folder in the file system.
- If you need to exclude some topics, enter the name of a DITaval file that you want to use for conditional processing in the **DITaval File** text box or use the **Browse...** button next to it to select a file from the file system.
- Use the **XLIFF Version** radio buttons to select the version of XLIFF to generate, current supported options are 1.2, 2.0 and 2.1.
- Select the **Reuse ICE Matches** check box if you want to compare current content with the content available last time an XLIFF file was generated and reuse all existing translations.
- Select the **Modified Files Only** check box to exclude from the generated XLIFF files those topics that have not changed since the last translation cycle.
- Select the **Use Translation Memories** check box if you want to recover translations for untranslated segments using the memories associated with the project.
- Select the **Paragraph Segmentation** check box if you need paragraph instead of sentence segmentation (default).  
*One advantage of paragraph segmentation over sentence segmentation in translation is that it can capture more context and coherence within a text unit.*
- Select the **Generate Word Count** check box if you want a statistic analysis to be generated for each target language.
- Select the **Ignore Tracked Changes** check box to ignore tracked changes from Oxygen XML Editor, improving text segmentation.



12. Select the **Ignore SVG Files** check box to exclude SVG files that contain translatable text.
13. Select the **Embed Skeleton** check box if you need XLIFF files that can be converted back to DITA in any computer using [OpenXLIFF Filters](#) or [XLIFF Manager](#).
14. Select the check boxes corresponding to the target languages that you want to process.
15. Click the **Generate XLIFF** button.

The **Process Log** dialog appears:



Click the **Cancel** button to interrupt XLIFF generation if needed.

## Results

An XLIFF file is generated for each selected target language. XLIFF files and optional word counts are placed in the selected XLIFF Folder.

## Import XLIFF

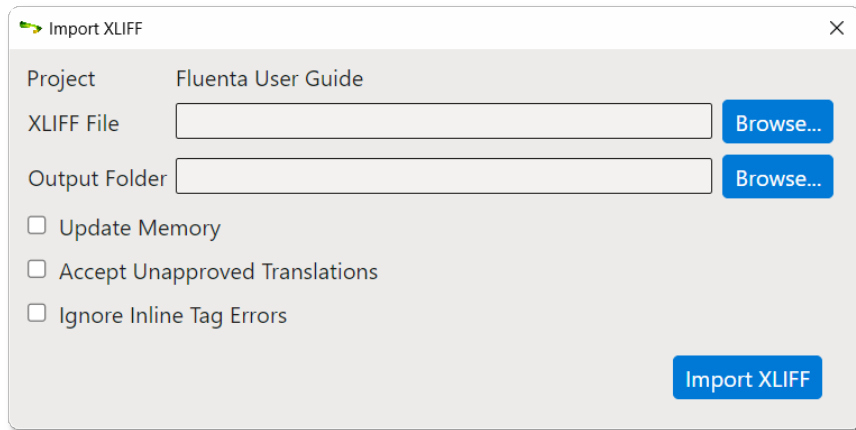
### About this task

Once you receive a translated [XLIFF](#) from your [Language Service Provider](#), you need to import it into Fluenta to generate a translated version of your project.

### Procedure

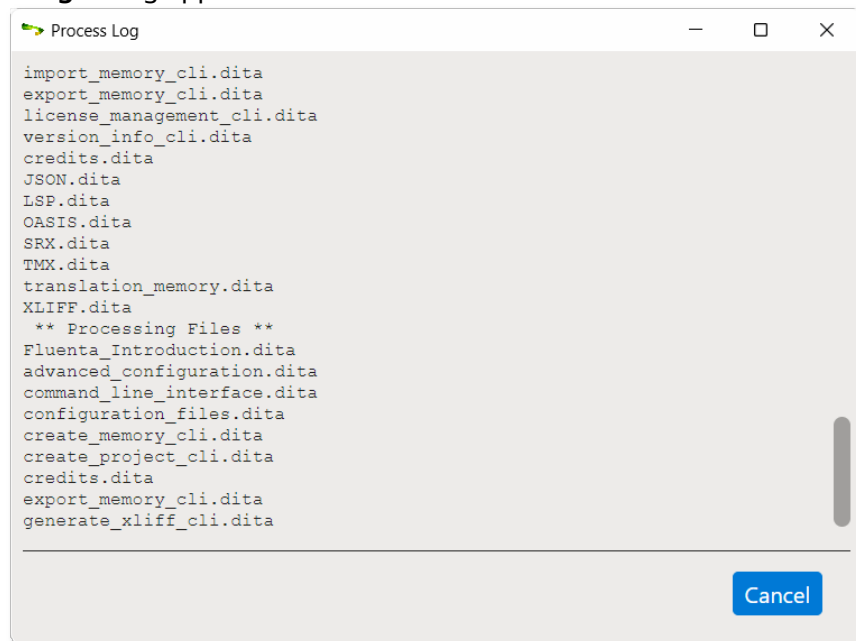
1. In **Projects** view, select the project that will receive the translated data.
2. In main menu, select **Projects** → **Import XLIFF** or click the **Import XLIFF** button in the **Projects** view toolbar.

The **Import XLIFF** dialog appears:



3. Type the name of the XLIFF file to import in the **XLIFF File** text box or use the **Browse...** button next to it to select an XLIFF file from the file system.
4. Type the name of the folder where the translated content should be placed in the **Output Folder** text box or use the **Browse...** button to select a folder in the file system.
5. Select the **Update Memory** check box if you want to store the imported translations in the project memory.
6. Select the **Accept Unapproved Translations** check box if the XLIFF file being imported does not have all translations marked as approved and you want to accept the existing drafts.
7. Select the **Ignore Inline Tag Errors** check box if you want to try importing an XLIFF file despite its errors with inline tags (may break the DITA content and make publication in target language impossible). If the check box is left blank and errors are found, a detailed report of errors in HTML format is automatically generated and displayed in the default browser.
8. Click the **Import XLIFF** button.

The **Process Log** dialog appears:




Click the **Cancel** button to interrupt XLIFF import if needed.

## Results

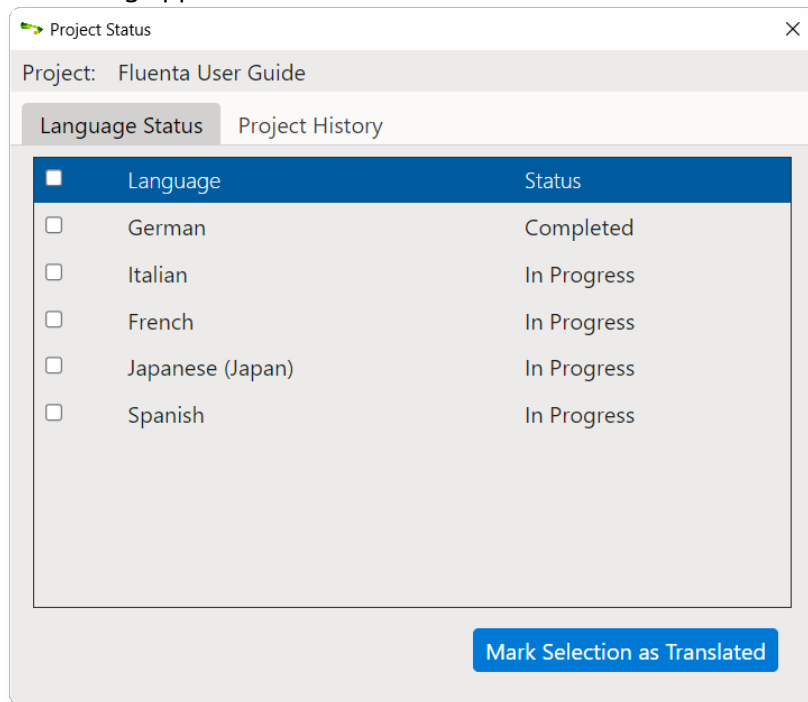
The XLIFF file is imported into the selected project. A translated version of the project content is created in the indicated output folder.

## Project Status

Follow these steps to check the translation status of your projects.

1. In **Projects** view, select the project that you want to examine.
2. In main menu, select **Projects** → **Project Status** or click the  **Project Status** button in the **Projects** view toolbar.

The **Project Status** dialog appears:

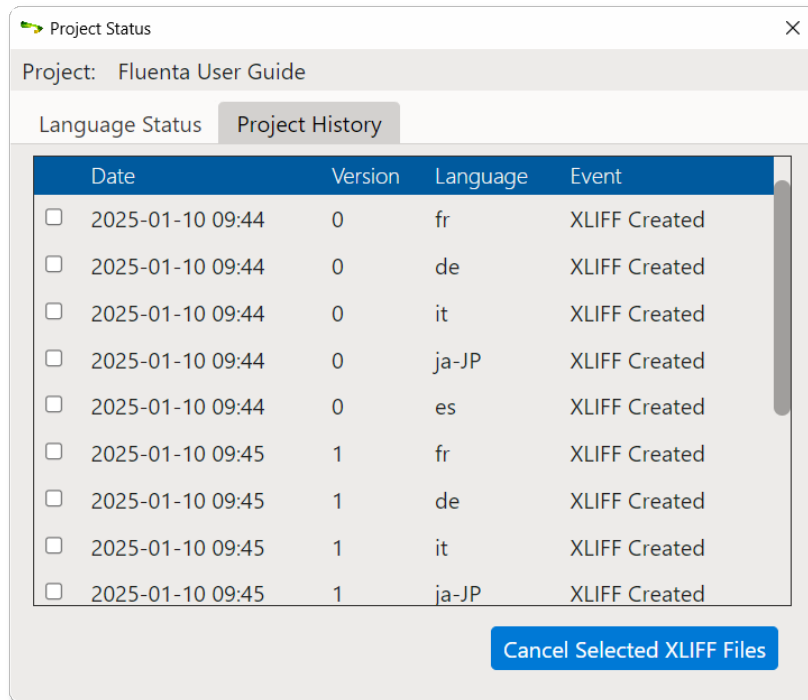


The **Status** column shows the current translation state for all target languages. Possible values are:

- **Untranslated:** no XLIFF file has been generated yet for the target language.
- **In Progress:** an XLIFF file has been generated for the language and the translated version has not been imported yet.
- **Completed:** the last XLIFF file generated for the target language has been imported.

You can mark one or more languages as completed by selecting the corresponding check boxes and clicking the **Mark Selection as Translated** button.

Click the **Project History** tab to display a list of events associated with the project.



Select the check box of an **XLIFF Created** event to discard one or more XLIFF files that have not been imported yet and click the **Cancel Selected XLIFF Files** button to adjust project status accordingly.

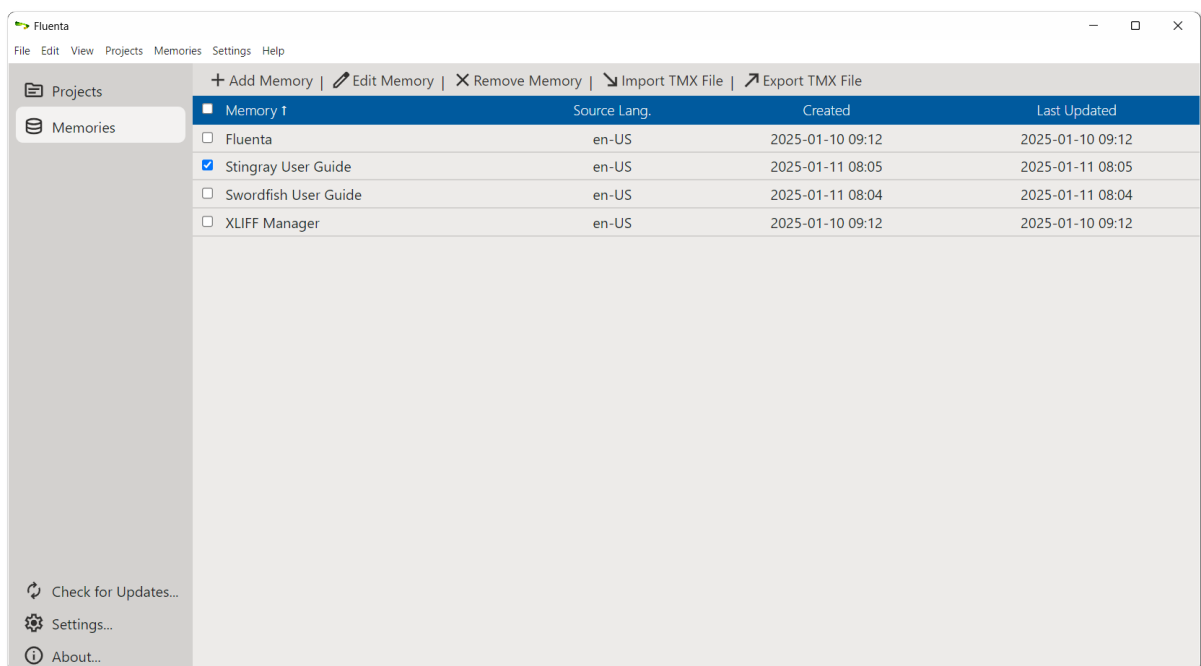
## Translation Memories

Translation Memory (TM) is a language technology that enables the translation of segments (paragraphs, sentences or phrases) of documents by searching for similar segments in a database and suggesting matches that are found in the databases as possible translations.

When you create a project, a new translation memory is automatically created and associated with your project. When you import a translated XLIFF file, the memory associated with the project can be populated with the translations included in the XLIFF file.

When you generate a new XLIFF file after adding new content to your project, Fluenta can reuse the data stored in the associated memories to translate the newly added content, reducing translation costs.

The **Memories View** in Fluenta contains a list of your existing translation memories, and the buttons in its toolbar allow you to manage all memory-related tasks.



## Add Memory

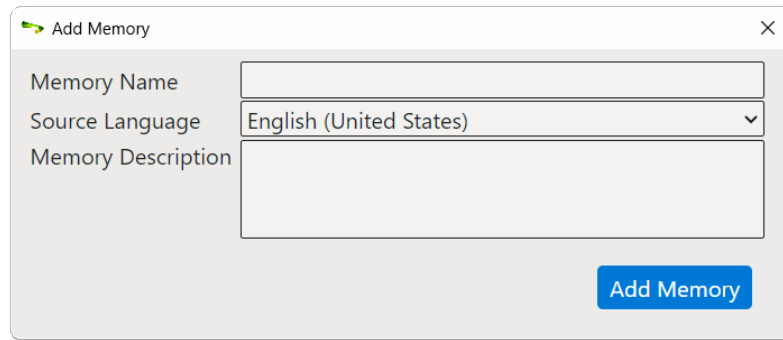
### About this task

Follow these steps to create a new translation memory.

### Procedure

1. In main menu, select **Memories** → **Add Memory** or click the **+ Add Memory** button in **Memories** view toolbar.

The **Add Memory** dialog appears:

A dialog box titled "Add Memory" with a close button (X) in the top right corner. It contains three input fields: "Memory Name" (a text box), "Source Language" (a dropdown menu showing "English (United States)"), and "Memory Description" (a larger text box). A blue "Add Memory" button is located at the bottom right.

2. Type a name for the new memory in the **Memory Name** text box.
3. Select the source language for the new memory in the **Source Language** drop-down list.
4. Optionally, enter a description in the **Memory Description** text box.
5. Click the **Add Memory** button.

## Results


A new memory is created and the list of memories in the **Memories** view is updated to reflect the changes.

## Edit Memory

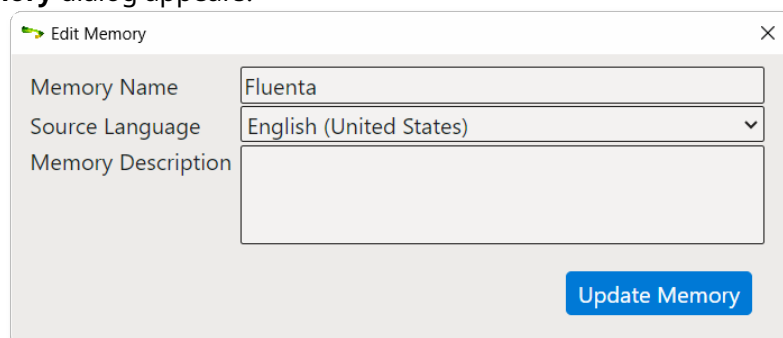
### About this task

Follow these steps to edit the name, source language or description of a translation memory.

### Procedure

1. In **Memories** view, select the memory that you want to edit by clicking on it.
2. In main menu, select **Memories** → **Edit Memory** or click the  **Edit Memory** button in **Memories** view toolbar.

The **Edit Memory** dialog appears:

A dialog box titled "Edit Memory" with a close button (X) in the top right corner. It contains three input fields: "Memory Name" (a text box with "Fluenta" entered), "Source Language" (a dropdown menu showing "English (United States)"), and "Memory Description" (a larger text box). A blue "Update Memory" button is located at the bottom right.

3. Edit all fields as required.
4. Click the **Update Memory** button.

## Results

Selected memory data is updated and the list of memories in the **Memories** view is updated to reflect the changes.

## Import TMX File

---

### About this task

You can populate translation memories with data from **TMX** files. Use this feature if you have legacy TM data that you want to reuse in Fluenta.

### Procedure

1. In **Memories** view, select the translation memory that will receive the imported data.
2. In main menu, select **Memories** → **Import TMX File** or click the ↗**Import TMX File** button in the **Memories** view toolbar.
3. In the file selection dialog that appears, select the TMX file to import.

## Results

The selected TMX file is imported into the selected translation memory.

## Export Memory as TMX File

---

### About this task

The content of Fluenta memories can be exported in **TMX** format for exchanging with other tools or for backup purposes.

### Procedure

1. In **Memories** view, select the translation memory that you want to export.
2. In main menu, select **Memories** → **Export TMX File** or click the ↗**Export TMX File** button in the **Memories** view toolbar
3. In the file selection dialog that appears, select a name and location for the TMX file that will contain the memory data.

## Results

Memory data is exported in TMX format in the selected file.

## Advanced Configuration

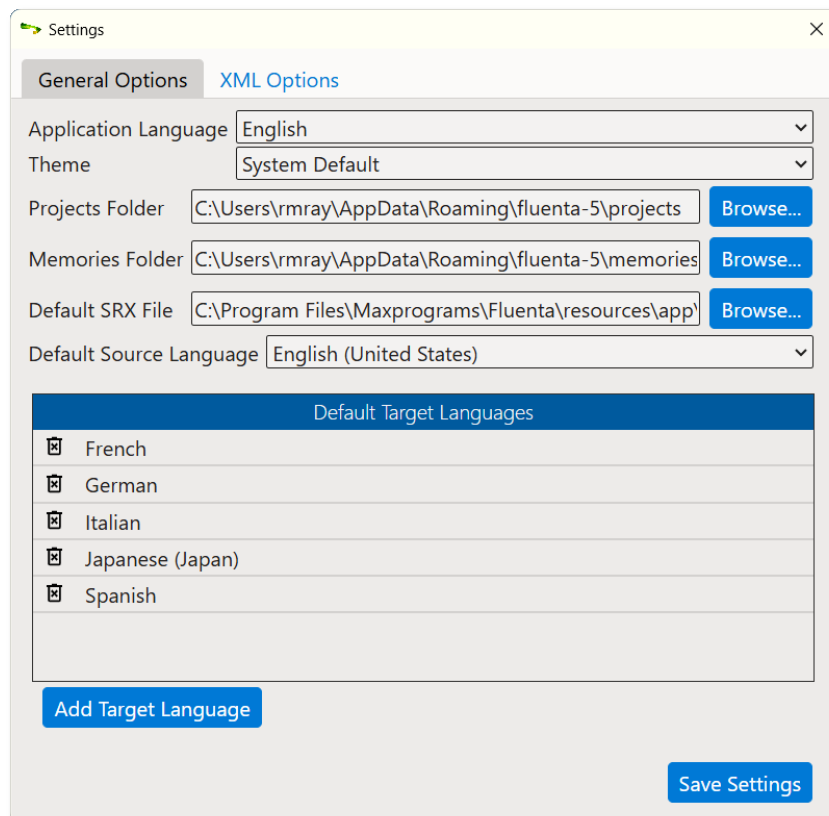
Fluenta default settings can be changed in the **Settings** dialog.

- To open the **Settings** dialog on Windows or Linux, in main menu select **Settings** → **Preferences**.
- To open the **Settings** dialog on macOS, select **Apple** → **Preferences** in the menu.

You can also open the **Settings** dialog clicking the ⚙ **Settings...** button on the left panel.

## General Options

The following picture shows the **General Options** tab of the **Settings** dialog:



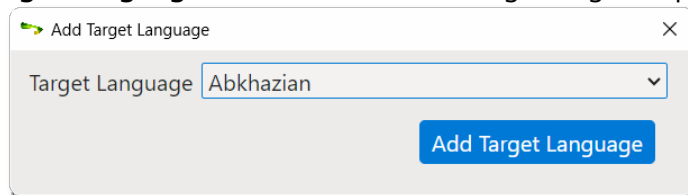
- Use the **Application Language** drop-down to select the language to use in the Graphical User Interface (GUI).
- Use the **Theme** drop-down to control the application's default appearance.
  - Select **System Default** to let your Operating System control the default text/background color combination (this option is not supported in some Linux distributions).
  - Select **Dark** to work with light text on dark background.
  - Select **Light** to work with dark text on light grey background.
  - Select **High Contrast** for a dark theme with enhanced contrast.
- If you want to change storage locations for your projects or memories, enter the desired paths on the **Projects Folder** and **Memories Folder** text boxes or click the associated **Browse...** buttons to select folders from the file system.



**Note**

Avoid using folders that could be modified by other programs while you work.

- To change the default **SRX** (Segmentation Rules eXchange) file that Fluenta uses for segmenting XLIFF files:
  1. Type the name of the SRX file in the **Default SRX File** text box or use the **Browse...** button to select an SRX file from the file system.
- Use the **Default Source Language** drop-down to select a default source language for your projects.
- To add new target languages to the default set:
  1. click the **Add Target Language** button and the following dialog will appear:



2. Select a language from the **Target Language** drop-down list.
  3. Click the **Add Target Language** button.
- To remove unnecessary languages from the default set, click the corresponding **Remove Language** buttons (🗑) on the leftmost column.

Click the **Save Settings** button after making any change to the default settings.

## XML Options

Fluenta needs to know two things for working with XML files:

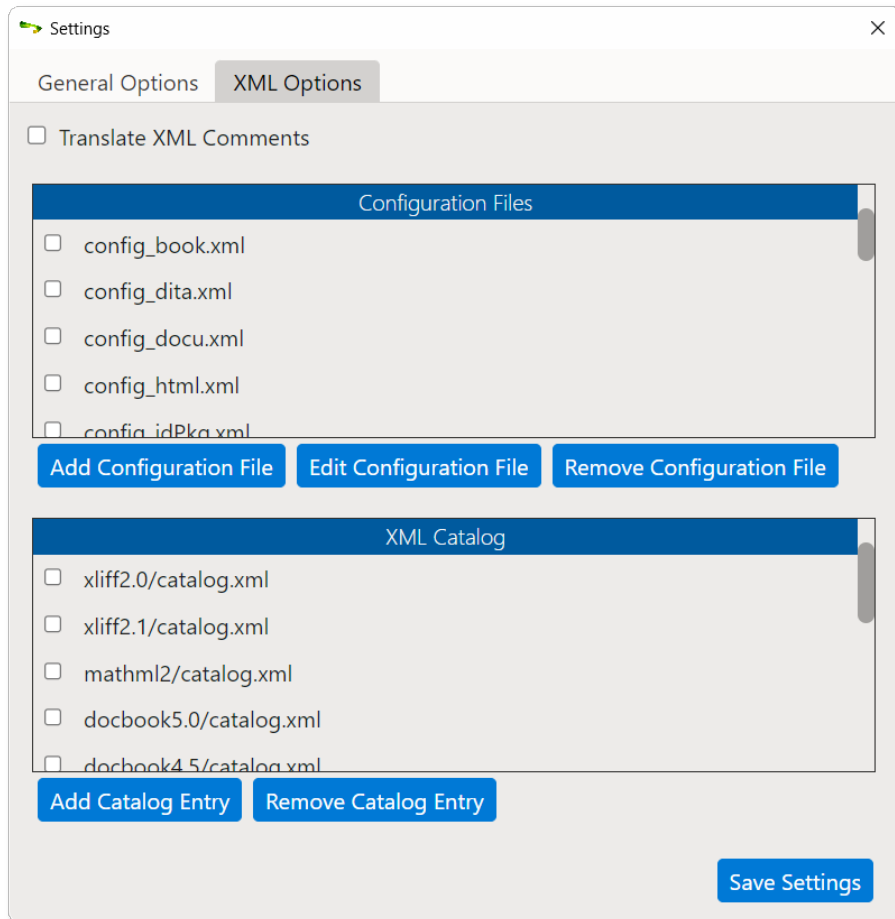
- How to locate the grammar rules and entities declared in an XML file, if any.
- What elements and attributes contain translatable text.

XML catalogs that follow the specification published at

<https://www.oasis-open.org/committees/entity/spec-2001-08-06.html> by OASIS are used to resolve the location of XML DTDs and Schemas.

Special XML files are used to configure the elements and attributes that contain translatable text. These files are used by the internal XML filter to extract text for processing. The configuration files are created and maintained using the application's graphical user interface.

The following picture shows the **XML Options** tab of the **Settings** dialog:



## Configuration Files

Fluenta includes a configuration file for the 600+ elements defined in [Appendix B.6 of DITA 1.3 All-Inclusive Edition](#).

A configuration file for [Scalable Vector Graphics \(SVG\)](#) is also shipped in fluenta installers.

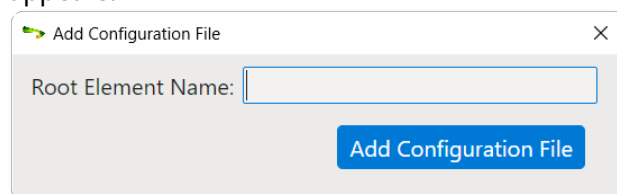
Additional configurations can be added by the user as required.

## Add Configuration File

### Procedure

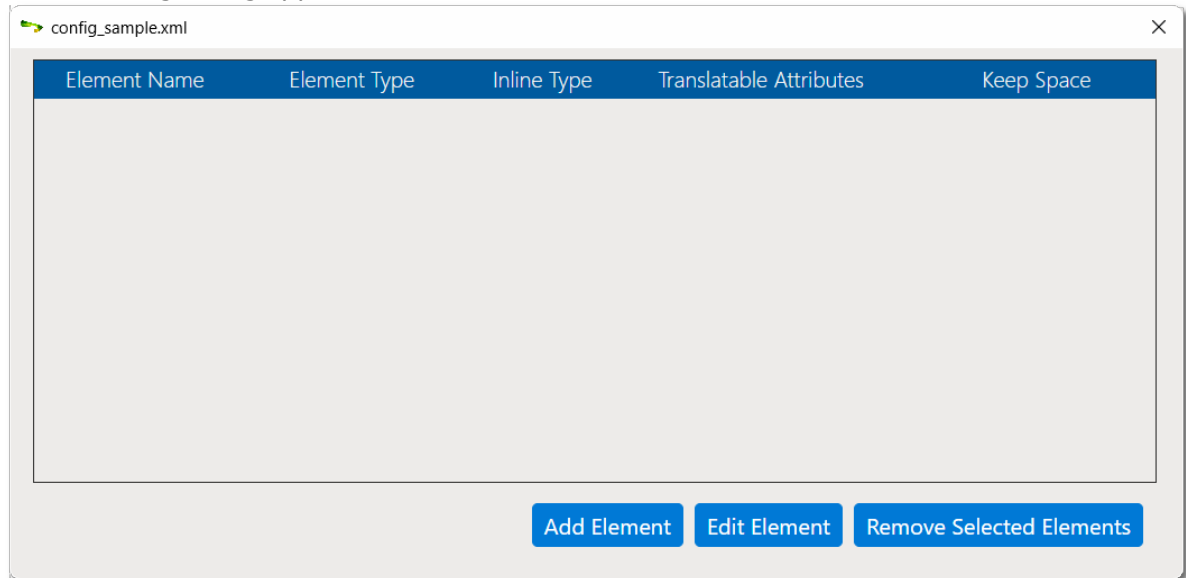
1. In **XML Options** tab of the **Settings** dialog, click the **Add Configuration File** button.

The following dialog appears:



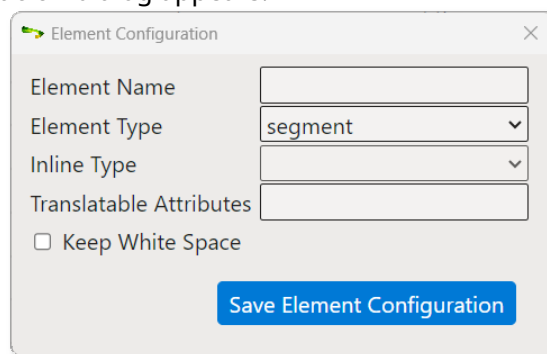
2. Type the name of the root element of your XML files in the **Root Element** text box. The name of the root element is used to name the configuration file.
3. Click the **Add Configuration File** button.

The following dialog appears:



- Click the **Add Element** button to add the configuration of an element.

The **Element Configuration** dialog appears:



- Type the name of the element being added in the **Element Name** text box.
- Select the type of element in the **Element Type** drop-down list. Available types are:
  - **segment**: the selected element starts a new section of translatable text.
  - **inline**: the selected element represents a change in formatting options and does not start a new section of translatable text.
  - **ignore**: the selected element and its children should be ignored.
- If the element type is "**inline**", select the kind of formatting represented by the element in the **Inline Type** drop-down list.
- If the element has translatable attributes, enter their names separated by a ";" in the **Translatable Attributes** text box.
- If white space needs to be preserved when extracting text, select the **Keep White Space** check box.
- Click the **Save Element Configuration** button.
- Repeat the previous steps until all required elements have been configured.

## Results

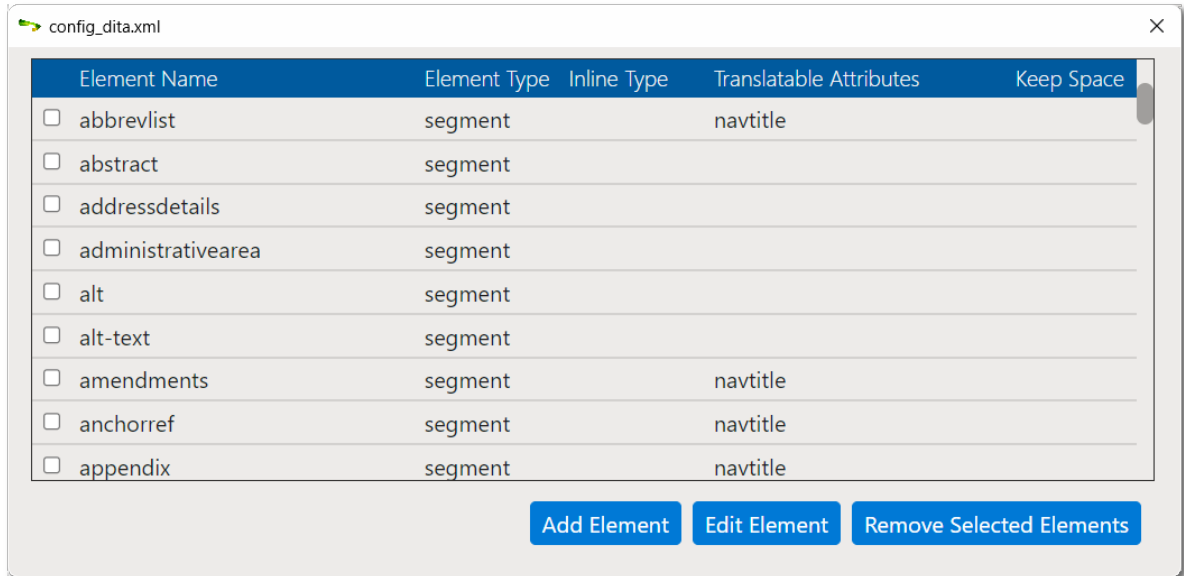
A new configuration file for the XML filter is created.

## Edit Configuration File

### Procedure

1. In **XML Options** tab of the **Settings** dialog, select the configuration file to edit.
2. Click the **Edit Configuration File** button.

The following dialog appears:



3. Use the buttons in the dialog to update the configuration file.
  - Use the **Add Element** button to add a new element to the configuration file.
  - Use the **Edit Element** button to modify the properties of an existing element.
  - Use the **Remove Selected Elements** button to delete all selected elements from the configuration file.

The configuration file is automatically updated as you add, edit or remove elements.

4. Repeat the previous step until all elements are properly configured.
5. Close the configuration dialog.

## Remove Configuration File

### Procedure

1. In **XML Options** tab of the **Settings** dialog, select the configuration file to remove.
2. Click the **Remove Configuration File** button.

## Results

Selected configuration file is removed and the list of configuration files is updated to reflect the change.

## XML Catalog

---

The application includes a default XML catalog with DTDs and XML Schemas for the most relevant formats and supported document types. Additional DTDs and XML Schemas can be added by the user as needed.

Select the **XML Options** tab in the **Preferences** dialog to add or remove entries from the catalog.

### Add Catalog Entry

Click the **Add Catalog Entry** button and a file selection dialog will appear. Locate the catalog in the file system and select it.

### Remove Catalog Entry

Select the catalog entry to remove in the catalogs table. Click the **Remove Catalog Entry** button and the application's catalog will be updated to reflect the change.

# Subscriptions

Fluenta is available in two modes:

- Source Code
- Yearly Subscriptions for installers and support

## Source Code

Source code of Fluenta is free. Anyone can download the source code from [Github.com](https://github.com), compile, modify and use it at no cost in compliance with the accompanying license terms.

## Subscriptions

The version of Fluenta included in the official installers from [Fluenta Download Page](#) can be used at no cost for 30 days requesting a free evaluation key.

Subscription keys are available at [Maxprograms Online Store](#). Subscription keys cannot be shared or transferred to different machines.

Subscription version includes unlimited direct email support at [tech@maxprograms.com](mailto:tech@maxprograms.com)

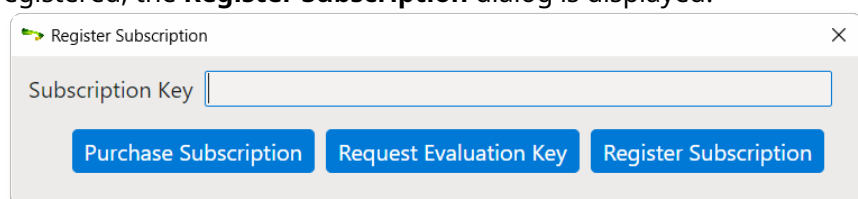
## Differences Summary

	Source Code	Subscription Based
Ready To Use Installers	No	Yes
Notarized macOS launcher	No	Yes
Signed launcher and installer for Windows	No	Yes
Headless mode batch scripts for running from CLI (Command Line Interface)	No	Yes
Technical Support	<ul style="list-style-type: none"><li>• Peer support at <a href="#">Groups.io</a></li></ul>	<ul style="list-style-type: none"><li>• Direct email at <a href="mailto:tech@maxprograms.com">tech@maxprograms.com</a></li><li>• Peer support at <a href="#">Groups.io</a></li></ul>

## First Registration - Evaluation Request

### Procedure

1. When you open Fluenta from subscription installers for the first time, or when the subscription key is not registered, the **Register Subscription** dialog is displayed:



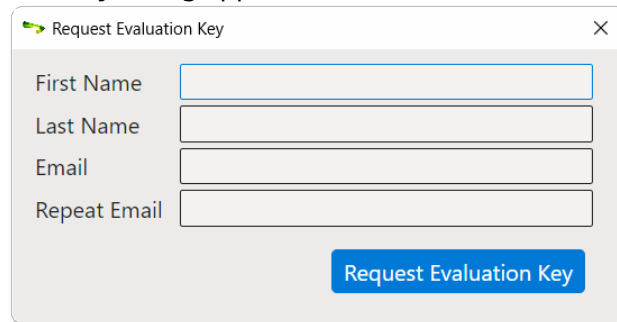
The image shows a 'Register Subscription' dialog box. It has a title bar with a green icon and a close button. Inside, there is a text input field labeled 'Subscription Key'. Below the input field are three buttons: 'Purchase Subscription', 'Request Evaluation Key', and 'Register Subscription'.

2. If you have a subscription key, type it in the **Subscription Key** text box and click the **Register Subscription** button.

On registration success, the main screen will be presented.

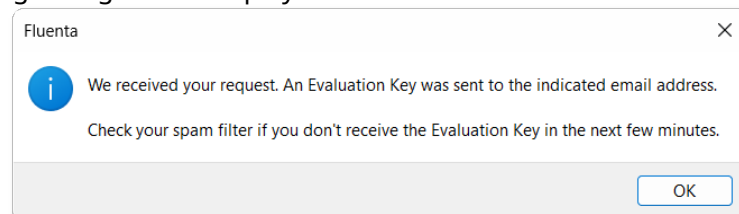
3. If you don't have a subscription key and wish to test the program during 30 days, click the **Request Evaluation Key** button.

The **Request Evaluation Key** dialog appears:

A dialog box titled "Request Evaluation Key" with a close button (X) in the top right corner. It contains four text input fields labeled "First Name", "Last Name", "Email", and "Repeat Email". Below these fields is a blue button labeled "Request Evaluation Key".

4. Enter the requested data and click the **Request Evaluation Key** button.

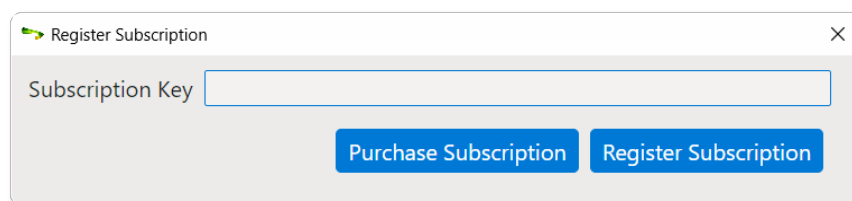
On success, an email with the requested evaluation key will be sent to the indicated address and the following dialog will be displayed:

A dialog box titled "Fluenta" with a close button (X) in the top right corner. It features a blue information icon (i) on the left. The text inside reads: "We received your request. An Evaluation Key was sent to the indicated email address. Check your spam filter if you don't receive the Evaluation Key in the next few minutes." At the bottom right is an "OK" button.

Check your email and register the Evaluation Key using the instructions shown above.

## Register a Subscription Key

When you start the program after the trial period has ended or a subscription key has expired, the **Register Subscription** dialog appears:

A dialog box titled "Register Subscription" with a close button (X) in the top right corner. It contains a text input field labeled "Subscription Key". Below the field are two blue buttons: "Purchase Subscription" and "Register Subscription".

If you don't have a subscription key, click the **Purchase Subscription** button to open [Maxprograms Online Store](#) in your default browser and purchase a key.

Type your subscription key in the **Subscription Key** text box and then click the **Register Subscription** button. On success, the **Register Subscription** dialog closes and you will be able to continue using Fluenta.

## Command Line Interface

The subscription-based version of Fluenta can be integrated into publishing and editing workflows via its Command Line Interface (CLI).

The methods exposed by Fluenta's CLI are:

- Create a project;
- Remove project;
- Retrieve the list of projects;
- Generate XLIFF files and optional word counts;
- Import an XLIFF file;
- Create a memory;
- Remove memory
- Retrieve the list of memories;
- Import a TMX file into a memory;
- Export a memory as TMX file;
- Register a subscription key;
- Retrieve version information.

On macOS and Linux use `fluenta_cli.sh` for executing Fluenta without the graphical user interface.

- On macOS the default location for the script is  
`/Applications/Fluenta.app/Contents/Resources/app/fluenta_cli.sh`
- On Linux the default location for the shell script is `/opt/Maxprograms/Fluenta/fluenta_cli.sh`

On Windows use `fluenta_cli.cmd` (available in the default installation directory) to process command line parameters.



### Note

Before you can use the Command Line Interface (CLI), you must start Fluenta from its Graphical User Interface (GUI) at least once.

## Create Project

Use the following parameters to create a project from the command line:

Parameter	Value	Description	Required
-add	JSON file	Full path to a <a href="#">JSON</a> file containing the data required for creating the project.	Yes

Example:

```
fluenta_cli.cmd -add addProject.json
```

Where the content of `addProject.json` should be like:



```
{
  "id": 12345678,
  "title": "JSON test",
  "description": "project created using JSON and CLI",
  "map": "D:\\sample content\\en\\User_Guide-use-only.ditamap",
  "srcLang": "en-US",
  "tgtLang": [
    "es",
    "fr"
  ]
}
```

## JSON Value Pairs

Member	Data Type	Description	Required
id	Number	Project identifier	Yes
title	String	Short text description for the project. This is the description displayed in Fluenta's GUI	Yes
description	String	A longer text description for the project	Yes
map	String	Path to the project DITA map	Yes
srcLang	String	Language code for the original DITA content	Yes
tgtLang	String Array	An array containing the target languages for the project	Yes
memories	Number Array	An array containing the ids of existing memories to be associated with the project	No

## Remove Project

Use the following parameters to remove a project from the command line:

Parameter	Value	Description	Required
-del	projectId	Project identifier	Yes

Example:

```
fluenta_cli.cmd -del 12345678
```

## Retrieve Project List

Use the following parameter to retrieve the list of projects from the command line:

Parameter	Value	Description	Required
-getProjects			Yes

Example:

```
./fluenta_cli.sh -getProjects
```

## Output example:

```
{
  "projects": [
    {
      "languageStatus": { "es": "2" },
      "tgtLanguages": [ "es" ],
      "lastUpdate": "2025-01-18 09:03",
      "srcLanguage": "en-US",
      "memories": [ 1734259114482 ],
      "description": "Downloaded from https://github.com/oxygenxml/userguide",
      "id": 1734259114482,
      "history": [
        {
          "date": "2024-12-15 08:09",
          "build": 0,
          "language": "es",
          "type": "0"
        },
        {
          "date": "2024-12-21 10:29",
          "build": 0,
          "language": "es",
          "type": "1"
        }
      ],
      "title": "Oxygen User Guide",
      "creationDate": "2024-12-15 07:38",
      "map": "userguide-master/DITA/UserManual.ditamap",
      "status": "4"
    },
    {
      "languageStatus": { "es": "3" },
      "tgtLanguages": [ "es" ],
      "lastUpdate": "2025-01-14 10:16",
      "srcLanguage": "en-US",
      "memories": [ 1736860617048 ],
      "description": "",
      "id": 1736860617048,
      "history": [],
      "title": "Fluenta User Guide",
      "creationDate": "2025-01-14 10:16",
      "map": "Documents/GitHub/FluentaPRO/docs/en/fluenta.ditamap",
      "status": "0"
    }
  ]
}
```

The possible values for the `status` field are:

- **0**: New project
- **1**: In progress
- **2**: Completed

- **3:** Untranslated
- **4:** Translated

The possible values for the `type` field in `history` array are:

- **0:** XLIFF Created
- **1:** XLIFF Imported
- **2:** XLIFF Canceled

## Generate XLIFF Files

Use the following parameters to generate XLIFF files for a project from the command line:

Parameter	Value	Description	Required
-generateXLIFF	JSON file	Full path to a <a href="#">JSON</a> file containing the data required for generating XLIFF files.	Yes
-verbose		Selects whether progress information is logged or not in stdout.	No

Example:

```
fluenta_cli.cmd -generateXLIFF genXLIFF.json -verbose
```

Where the content of `genXLIFF.json` should be like:

```
{
  "id": 12345678,
  "xliffFolder": "C:\\sample data\\XLIFF",
  "tgtLang": [
    "es",
    "fr"
  ],
  "ditaval": "C:\\sample data\\filter.ditaval",
  "useICE": true,
  "modifiedFilesOnly": true,
  "useTM": true,
  "generateCount": false,
  "version": 2.0,
  "paragraph": false,
  "ignoreTrackedChanges": false,
  "ignoreSVG": true,
  "embedSkeleton": false
}
```

## JSON Value Pairs

Member	Data Type	Description	Required
id	Number	Project identifier	Yes
xliffFolder	String	Path to the folder where XLIFF files and optional word counts should be stored	Yes

Member	Data Type	Description	Required
tgtLang	String Array	An array containing the target languages for the XLIFF files	Yes
ditaval	String	Path to a .ditaval file for filtering the content to be extracted	No
useICE	Boolean	Selects whether existing ICE matches should be reused. Default <code>false</code>	No
modifiedFilesOnly	Boolean	Selects whether topics that have not changed since the last translation cycle should be included in the generated XLIFF files. Default value: <code>false</code>	No
useTM	Boolean	Selects whether translation memories should be used. Default value: <code>false</code>	No
generateCount	Boolean	Selects whether word counts should be generated. Default value: <code>false</code>	No
version	String	Selects the version of XLIFF to generate. Values: 1.2, 2.0 or 2.1. Default value: 2.0	No
paragraph	Boolean	Selects whether paragraph segmentation should be used instead of sentence segmentation. Default value: <code>false</code>	No
ignoreTrackedChanges	Boolean	Selects whether tracked changes from Oxygen XML Editor should be ignored. Default value: <code>false</code>	No
ignoreSVG	Boolean	Selects whether translatable SVG files should be ignored when generating XLIFF. Default value: <code>false</code>	No
embedSkeleton	Boolean	Selects whether skeletons should be embedded in the generated XLIFF files, allowing conversion back to DITA in any computer using <a href="#">OpenXLIFF Filters</a> or <a href="#">XLIFF Manager</a> . Default value: <code>false</code>	No

## Import XLIFF File

Use the following parameters to import a translated XLIFF file into a project from the command line:

Parameter	Value	Description	Required
-importXLIFF	JSON file	Full path to a <a href="#">JSON</a> file containing the data required for importing the XLIFF file	Yes
-verbose		Selects whether progress information is logged or not in stdout.	No

Example:

```
fluenta_cli.cmd -importXLIFF impXLIFF.json -verbose
```

Where the content of `impXLIFF.json` should be like:

```
{
  "id": 12345678,
  "xliffFile": "C:\\sample data\\XLIFF\\spanish.xlf",
  "outputFolder": "C:\\sample data\\es\\",
  "updateTM": true,
  "acceptUnapproved": true,
  "ignoreTagErrors": false
}
```

## JSON Value Pairs

Member	Data Type	Description	Required
id	Number	Project identifier	Yes
xliffFile	String	Path to the XLIFF file to be imported	Yes
outputFolder	Boolean	Path to the folder where the translated DITA files should be stored	Yes
updateTM	Boolean	Selects whether the memory associated with the project should be updated with the translations in the XLIFF file	Yes
acceptUnapproved	Boolean	Selects whether segments that are translated but not approved should be treated as approved. Default value: <code>false</code>	No
ignoreTagErrors	Boolean	Selects whether inline tag errors should be ignored. If set to <code>false</code> and errors are found, a detailed report in HTML format is generated and stored in the folder that contains the XLIFF file. Default value: <code>false</code>	No

## Create Memory

Use the following parameters to create a memory from the command line:

Parameter	Value	Description	Required
-addMem	JSON file	A JSON file containing the data required for creating the memory	Yes

Example:

```
fluenta_cli.cmd -addMem addMemory.json
```

Where the content of `addMemory.json` should be like:

```
{
  "id": 12345678,
  "title": "JSON test",
}
```

```
{
  "description": "memory created using JSON and CLI",
  "srcLang": "en-US"
}
```

## JSON Value Pairs

Member	Data Type	Description	Required
id	Number	Memory identifier	Yes
title	String	Short text description for the memory. This is the description displayed in Fluenta's GUI.	Yes
description	String	A longer text description for the memory.	Yes
srcLang	String	Source language code.	Yes

## Remove Memory

Use the following parameters to remove a memory from the command line:

Parameter	Value	Description	Required
-delMem	memoriId	Memory identifier	Yes

Example:

```
fluenta_cli.cmd -delMem 12345678
```

## Retrieve Memory List

Use the following parameter to retrieve the list of memories from the command line:

Parameter	Value	Description	Required
-getMemories			Yes

Example:

```
./fluenta_cli.sh -getMemories
```

Output example:

```
{ "memories": [
  {
    "lastUpdate": "2025-01-07 16:06",
    "name": "Oxygen User Guide",
    "srcLanguage": "en",
    "description": "",
    "id": 1734259114482,
    "creationDate": "2024-12-15 07:38"
  },
  {
    "lastUpdate": "2025-01-14 10:16",
    "name": "Fluenta User Guide",
    "srcLanguage": "en-US",
```

```
"description": "",  
"id": 1736860617048,  
"creationDate": "2025-01-14 10:16"  
}  
1}
```

## Import TMX File

Use the following parameters to import a [TMX](#) file into a memory from the command line:

Parameter	Value	Description	Required
-importTmx	memId	The id of the memory.	Yes
-tmx	tmxFile	Path to the TMX file to be imported	Yes

Example:

```
fluenta_cli.cmd -importTmx 12345678 -tmx "c:\sample data\updated.tmx"
```

## Export TMX File

Use the following parameters to export a memory as a [TMX](#) file from the command line:

Parameter	Value	Description	Required
-exportTmx	memId	The id of the memory to be exported	Yes
-tmx	tmxFile	Path to the TMX file to be generated	Yes

Example:

```
fluenta_cli.cmd -exportTmx 12345678 -tmx "c:\sample data\exported.tmx"
```

## Subscription Management

Use the following parameters to register a subscription key from the command line:

Parameter	Value	Description	Required
-reg	key	The subscription key to be registered	Yes

Example:

```
fluenta_cli.cmd -reg HAL9000
```



### Note

Registering a subscription key requires an Internet connection.

## Retrieve Version Information

Use the following parameter to retrieve Fluenta version and build information from the command line:

Parameter	Value	Description	Required
-version			Yes

Example:

```
./fluenta_cli.sh -version
```

Output example:

```
{
  "build": "20250118_1109",
  "version": "5.0.0"
}
```



# Licenses

---

Fluenta uses components from these open source projects:

- **BCP47J**
  - Available from <https://github.com/rmraya/BCP47J>
  - Published under Eclipse Public License 1.0
- **DTDParser**
  - Formerly available from <http://wutka.com/dtdparser.html>
  - Published under two licenses: Apache Style License and LGPL 2.1
- **Electron**
  - Available from <https://www.electronjs.org/>
  - Published under MIT License
- **JSON-java**
  - Available from <https://github.com/stleary/JSON-java>
  - Released to Public Domain
- **jsoup**
  - Available from <https://jsoup.org>
  - Published under MIT License
- **MapDB**
  - Available from <https://mapdb.org>
  - Published under Apache License
- **OpenXLIFF Filters**
  - Available from <https://www.maxprograms.com/products/openxliff.html>
  - Published under Eclipse Public License 1.0
- **SQLite**
  - Available from <https://sqlite.org/>
  - Released to Public Domain
- **SQLite JDBC**
  - Available from <https://github.com/xerial/sqlite-jdbc>
  - Published under Apache License
- **Swordfish**
  - Available from <https://www.maxprograms.com/products/swordfish.html>

- Published under [Eclipse Public License 1.0](#)
- **TypesBCP47**
  - Available from <https://github.com/rmraya/TypesBCP47>
  - Published under [Eclipse Public License 1.0](#)
- **TypesXML**
  - Available from <https://github.com/rmraya/TypesXML>
  - Published under [Eclipse Public License 1.0](#)
- **XMLJava**
  - Available from <https://github.com/rmraya/XMLJava>
  - Published under [Eclipse Public License 1.0](#)

# Glossary

## JSON

JavaScript Object Notation (JSON) is an open standard for the serialization of structured data in text format . It is derived from the object literals of JavaScript, as defined in the ECMAScript Programming Language Standard, Third Edition. JSON specification is available at <https://tools.ietf.org/html/rfc7159>.

## Language Service Provider (LSP)

A company or individual specialized in providing translation and localization services.

## OASIS

OASIS ([Organization for the Advancement of Structured Information Standards](#)) is a not-for-profit consortium that drives the development, convergence and adoption of open standards for the global information society.

## SRX

Segmentation Rules eXchange (SRX) is an XML-based open standard, originally published by LISA (Localization Industry Standards Association), for describing how translation and other language-processing tools segment text for processing.

## TMX

Translation Memory eXchange (TMX) is an open standard originally published by LISA (Localization Industry Standards Association). The purpose of TMX is to allow easier exchange of translation memory data between tools and/or translation vendors with little or no loss of critical data during the process.

## Translation Memory

Translation Memory (TM) is a language technology that enables the translation of segments (paragraphs, sentences or phrases) of documents by searching for similar segments in a database and suggesting matches that are found in the databases as possible translations.

## XLIFF

XLIFF (XML Localization Interchange File Format) is an open standard developed by [OASIS](#) (Organization for the Advancement of Structured Information Standards). The purpose of this vocabulary is to store localizable data and carry it from one step of the localization process to the other, while allowing interoperability between tools.