

MADCAP FLARE 2024

Translation and Localization

Copyright © 2024 MadCap Software. All rights reserved.

Information in this document is subject to change without notice. The software described in this document is furnished under a license agreement or nondisclosure agreement. The software may be used or copied only in accordance with the terms of those agreements. No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or any means electronic or mechanical, including photocopying and recording for any purpose other than the purchaser's personal use without the written permission of MadCap Software.

MadCap Software
9171 Towne Center Drive, Suite 335
San Diego, California 92122
858-320-0387
www.madcapsoftware.com

THIS PDF WAS CREATED USING MADCAP FLARE.

CONTENTS

CHAPTER 1

Introduction 5

CHAPTER 2

General Information for Translation 7

- Translation Life Cycle 8
- Bi-Directional Authoring 9
- Translation Best Practices 11
- Single Language vs. Multilingual Outputs 31
- Language Skins 59
- Translation and Localization Features Per Output Type 66

CHAPTER 3

Methods for Translation 68

- Method Comparison 69
- Method 1: Full-Service Translation 70
- Method 2: Translation With Lingo 72
- Method 3: Translation With Third-Party Tool 74
- Method 4: Translation in Flare Project 76
- Method 5: Translation of Output Files 86

CHAPTER 4

Other Information for Translation	87
Exporting Projects for Translation	88
Multilingual Support for Salesforce	90
Multilingual Support for ServiceNow®	92
Multilingual Support for MadCap Connect to Zendesk	95
Selecting a Language for the Interface	98

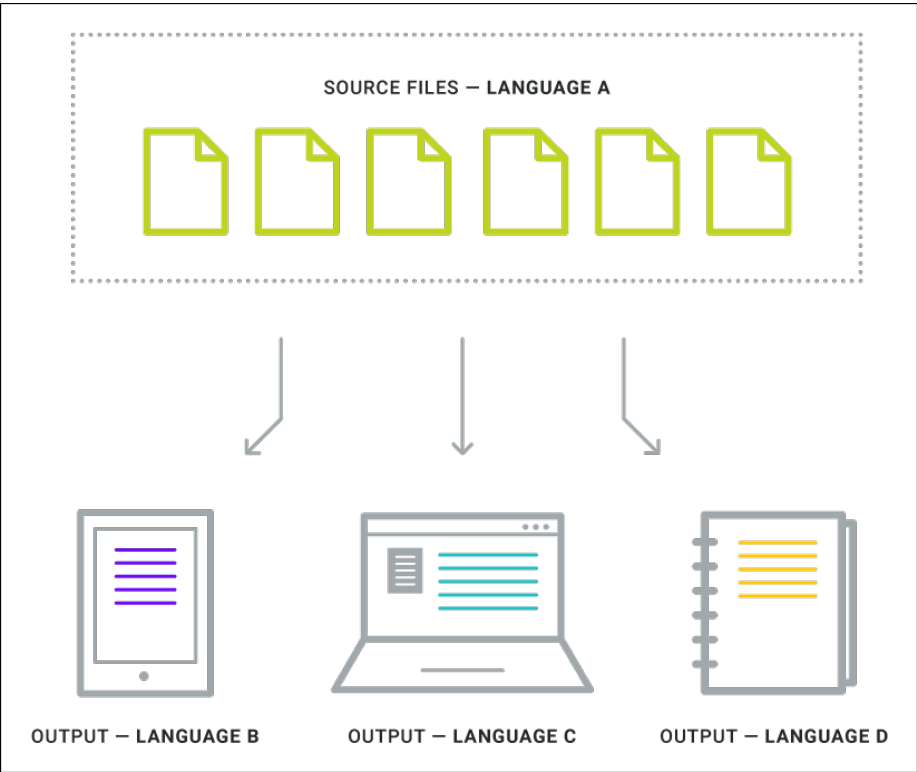
APPENDIX

PDFs	99
Tutorials	99
Cheat Sheets	100
User Guides	101

CHAPTER 1

Introduction

Translation and localization services provide the ability for us to communicate and understand each other around the world better. Translation changes the language of your content, while localization adapts content for a specific region. MadCap products (e.g., Flare, Lingo, Capture, etc.) are designed to streamline the translation process so you can easily provide multilingual documentation to your global users.



General Information

- "Translation Life Cycle" on page 8
- "Bi-Directional Authoring" on page 9
- "Translation Best Practices" on page 11
- "Single Language vs. Multilingual Outputs" on page 31
- "Language Skins" on page 59
- "Translation and Localization Features Per Output Type" on page 66

Methods

"Methods for Translation" on page 68

- "Method 1: Full-Service Translation" on page 70
- "Method 2: Translation With Lingo" on page 72
- "Method 3: Translation With Third-Party Tool" on page 74
- "Method 4: Translation in Flare Project" on page 76
- "Method 5: Translation of Output Files" on page 86

Other Information

- "Exporting Projects for Translation" on page 88
- "Multilingual Support for Salesforce" on page 90
- "Multilingual Support for MadCap Connect to Zendesk" on page 95
- "Selecting a Language for the Interface" on page 98

General Information for Translation

There are various pieces of general information you should know if you plan to use this feature.

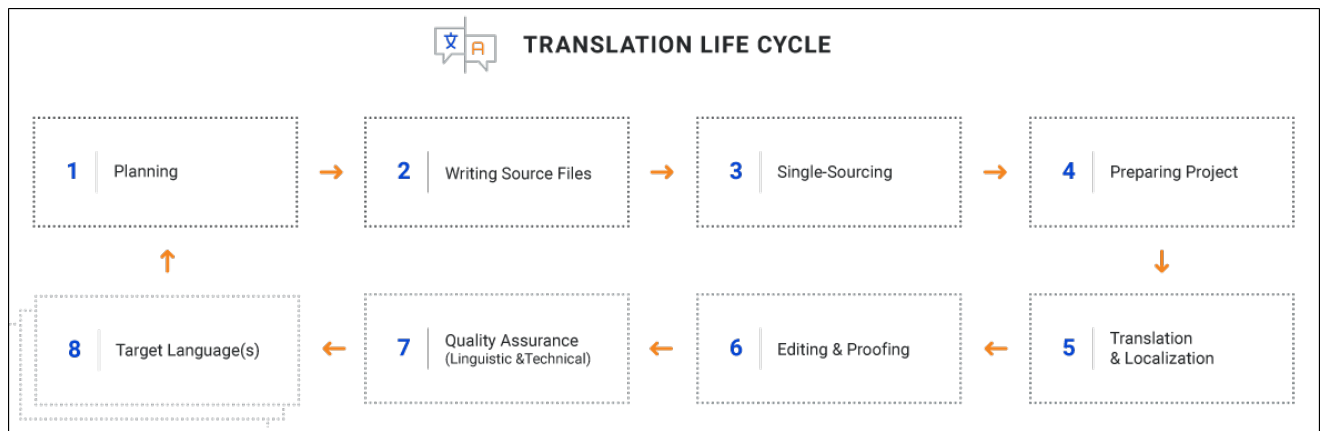
This chapter discusses the following:

- Translation Life Cycle 8
- Bi-Directional Authoring 9
- Translation Best Practices 11
- Single Language vs. Multilingual Outputs 31
- Language Skins 59
- Translation and Localization Features Per Output Type 66

I Translation Life Cycle

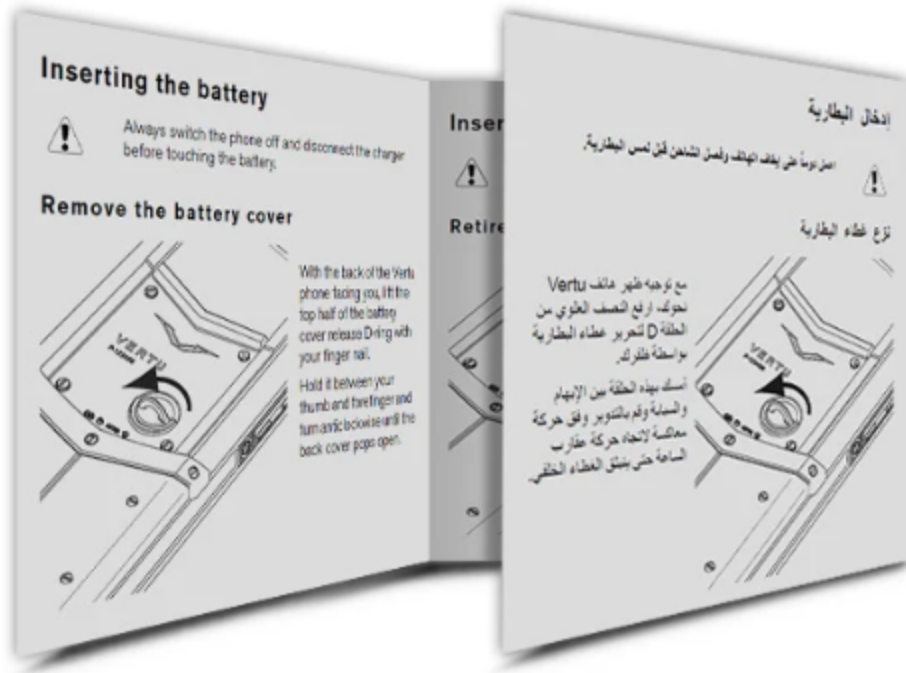
From the beginning to the end of the translation life cycle, a project requires a lot of communication and collaboration with others involved with the translation process for successful results.

Members of a translation team can include, but are not limited to, single or multiple authors, product managers, language service providers (LSPs) or translators, and in-country reviewers (ICRs). Each member who is a part of the project contributes to the overall translation life cycle.



Bi-Directional Authoring

Flare supports bi-directional language authoring and publishing, and is fully Unicode capable, providing translation into both single- and double-byte languages. This includes English, French, German, Japanese, Chinese, Arabic, Hebrew, Persian, and more.



Bi-Directional XML Editor

Full authoring of right-to-left (RTL) and left-to-right (LTR) text is supported in the XML Editor.

- RTL text is readable and can be edited in the XML Editor.
- CSS and HTML direction properties are supported (e.g., "direction: rtl" and dir="rtl"). The following occur if these properties are set to "right."
 - Text is right-aligned.
 - Tables flow from RTL (i.e., cell 1 is at the top right, cell 2 is to the left of it).
 - Bulleted lists and numbered lists are to the right of their content instead of left.
- If an RTL language is selected for the project or topic, the default direction for content is right-to-left. If an LTR language is selected, the default direction is left-to-right.

I Translation Best Practices

Preparing documentation for translation and localization might be daunting at first, but there are best practices you can implement to make the task easier, cost effective, and ready for an efficient translation. An overarching point to keep in mind as you write, is to keep the project and content as simple as possible. If the original content is clean and of high quality, then the translation and localization will most likely be clean and of high quality.

General Information

- "Content Strategy for Translation" on the next page
- "Optimizing Content in Flare" on page 14

Guidelines

- "Communicating With Translators" on page 17
- "Translation Writing Guidelines" on page 18
- "Using Flare Features Wisely for Translation" on page 20
- "Formatting for Translation" on page 24
- "Using Images for Translation" on page 25

Other Information

- "Inverting Styles, Page Layouts, and Image Callouts" on page 26
- "Inverting Hotspot Images" on page 28
- "Table Styles and Right-to-Left Languages" on page 30

General Information for Translation Best Practices

There are various pieces of general information you should know if you plan to use this feature.

Content Strategy for Translation

Content strategy has different meanings for different people, but let's think of it as a way to connect project goals to audience needs. For a translation project, language is just another factor in the overall content strategy. This involves making workflow and content choices throughout the translation life cycle. Upfront planning is essential for efficient translations and for keeping costs down.

✔ **TIP** From the beginning, think if your project needs to be translated now, or if the need could come up later. It is better to plan for it (even if it has a remote chance of happening), rather than not giving it any thought at all and having to backtrack or start over.

Project Planning

Take the time to define and document your project's scope to meet your end goal in the most effective way possible. By asking and answering questions such as the following, your project and the best way to implement it, will start to take shape.

- What is the source language, and what are the languages needed?
- What do the published outputs need to look like?
- What content is needed, and what content should be shared?
- What should the project structure look like, so it is easy to translate?
- How should the project be prepared for translations?
- What translation workflow will move through the company?
- Who is the translator and what is the budget for translations?
- Are the translation and review processes accounted for in the schedule?

Communication Tools

MadCap Flare

The fact that you are reading Flare Help means that you are on the right track for a successful translation experience. Using the right tool for content development streamlines the translation process, and provides you with the necessary features to get the job done properly.

Flare is a powerful single-sourcing system that allows you to control the amount of content created. Since most translations are billed on a price per word basis, reusing content reduces the word count, which lowers translation costs.

MadCap Lingo

One of the best ways to go about getting a project translated is to use Flare with MadCap Lingo because the two products are tightly integrated for translation. Localized files stay within the actual project which helps to prevent content or formatting corruption. Also, translators can use previous translations created in other tools by importing Translation Memory eXchange (TMX) files.

After opening a project in Lingo, a translator immediately sees a list of all of the files (e.g., topics, snippets, variables), index markers, and concept markers for translation. Then, after translating the content using Lingo, those results are easily exported to a new project in that language.

Optimizing Content in Flare

Flare enables you to create a well-constructed project. Customizing project structure, leveraging single-sourcing features, and preparing the project for translations all contribute to optimizing content.


Structuring a Project


How you structure a project for translation in Flare can vary, and how it is done does not impact the translator. In general, if the setup works in the source language, then it will work for translations.

While in the planning stages of your project, determine the best way to structure your project. Should you have a single project versus multiple? Should projects be separated or linked together? How much content will be shared? Flare can accommodate many different multilingual scenarios.

It is recommended to set up a separate project for each language for the following reasons:

- Lingo exports single language projects. It works well to keep translated files in their own project.
- A single project for each language enables you to maintain many outputs for that language.
- It is a convenient way to keep projects separated if more than one translator is employed.
- It is a simple and clean structure for shared content and single-sourcing.

 **TIP** Communicate with your language service provider (LSP) or translator for advice on how to configure your Flare project with translation in mind.

 **NOTE** Any way you structure your project, it is recommended to send the project's source files for translation, not the output files.

Single-Sourcing to Optimize Content

Using Flare's single-sourcing features are beneficial when translating content. This includes, but is not limited to: topic-based authoring, snippets, variables, and conditions. Single-sourcing lets the source language project work for you.

- Writing content once, and reusing it across your project.
- Translating only the source files, regardless of how many times you reuse it.
- Producing multiple outputs. Usually the file names in a translated project remain in the source language. This enables you to create other output types using the translated files in the project without the assistance of a translator.
- Simplifying maintenance. With a translated project, making minor adjustments (e.g., product name, address) for a republish is easy.
- Promoting content quality control.

✔ **TIP** Using Flare with other Madcap products such as Lingo and Capture streamlines the process even more because they are designed to optimize the translation workflow as a suite.

What's Next?

Once you have established a plan, you can move forward with writing, single-sourcing, and preparing your project for translation. Although implementing best practices is not required, doing so might help by:

- Improving the overall translation experience.
- Cutting down on translation costs.
- Making the translation and localization easier for the translator.
- Streamlining your workflow.
- Creating the most efficient and effective multilingual project.
- Learning about common translation challenges and how to adjust for them.

See "Guidelines for Translation Best Practices" on the next page.

Guidelines for Translation Best Practices

The foundation of your translation project is formed when you define your workflow and project structure. The next steps are to build on that foundation with well-constructed content. In the following topics, suggestions are offered on how to best prepare your content and project for translation.


The objective is to create translation-ready content, which can save you time and money, and increase the overall readability and quality of your target translations.

Communicating With Translators

Communication is key! You want your project to get translated without delays or difficulties, and translators want to provide you with accurately translated files. One way to achieve quality translation results is to establish a healthy relationship with your language service provider (LSP) or translator.

Establishing Translator Relations

- **Learn About Your Translator** Ideally, you want to find a reliable translation partner who is interested in a long-term business relationship. It is cost-effective to stick with a translator, rather than switch each time you have a translation project to do. It might be beneficial to find a translator who is familiar with Flare.
- **Communicate With the Translator About the Process** Talk about services provided, scheduling, pricing, pre-translation and post translation analysis, project updates during translation, reviewing and editing content, formatting, final delivery of files, etc. Be sure to include the translator in your process.
- **Talk With the Translator About Your Project** If just starting a project, ask for advice on how to best configure it. If your project is established, be sure to explain the project structure to the translator.
- **Provide Context to the Translator** Translators do not know the project like you do, so any contextual resources you can provide helps the translator pinpoint the meaning and tone for translation. Reference materials such as a final PDF of the document, glossary terms, style guide, branding guide, abbreviations, list of terms not for translation, screenshots, notes, or training guide can provide insight.

 **NOTE** MadCap Software offers MadTranslations for complete end-to-end translation and localization services. For more information, see:

<https://www.madcapsoftware.com/services/translation/>

Translation Writing Guidelines

When content is translated to another language, it is important to write source files with the cleanest and simplest language possible. This is advantageous because the quality of translated output depends directly on the quality of the source files.

✔ **TIP** The following guidelines are intended for all authors, but they also assume English as the source language and offer tips on avoiding translation issues based on English quirks. If you are using a primary language other than English, it is recommended to find translation guidelines that apply specifically to your source language.

General Guidelines for Translation

- **Simple Sentence Structure** Keeping sentences concise increases readability and reduces word count. In addition, writing as clearly as possible in your source language can affect the quality of machine translation.
 - Write in a direct, active voice.
 - Avoid repetition.
 - Avoid ambiguous content.
 - Avoid noun strings, idioms, gerunds, and cultural references.
 - Use standard word order (e.g., subject, verb, object).
 - Establish consistent naming conventions.
 - Use consistent terminology.

📄 **NOTE** When writing for a software user interface (UI), the UI labels can influence terminology.

- **Mitigate Errors** The content should be clean; meaning that the content data should be error-free. Since translations are essentially a copy of the original project in a different language, errors could carry over into the new translated project.
 - Make sure the project is technically complete (e.g., fix broken links, references, condition tags, formatting, missing files).
 - Make sure the project is grammatically correct (e.g., watch for bad punctuation, awkward sentence segmentation).
 - Make sure files are logically structured and complete in the project.
- **International Formats** Depending on where you are in the world, different standards exist for writing measurements, temperatures, phone numbers, currency, dates. etc.
 - Spell out the name of the month for clarity on international dates (e.g., 05/11/2020 might reference May, or November).
 - Use metric for units of measure. Most of the world uses the metric versus the imperial measurement system. If your source language is accustomed to the imperial system it might be worth adopting the metric system to simplify the translation process (eliminating the need for excessive conditions).

✔ **TIP** It is recommended to create and use an in-house style guide to help define and dictate consistency, terminology, and writing style for a project. This is especially effective among a team of writers.

Using Flare Features Wisely for Translation

There are many advantages in using Flare's single-sourcing tools such as snippets, variables, and conditions for reusing content and avoiding repetition in your project. However, considering that most languages are grammatically different from each other, these features should be used with caution. Try to use them smartly, and do not overuse them.

Because sentence structure can change a lot depending on the target language, be aware of the following general guidelines when using these powerful features.

Snippets

- Do not use snippets in partial sentences. This can cause problems in translation because translators see each part of the sentence separately. Each snippet is its own file in a long list of other files, making it difficult for translators to know its context.
- Snippets are best for single elements, rather than using as a variable that fits into a sentence, unless the variable does not change (e.g., product name, brand name, or proper noun).

Variables

- Keep variables simple. Translators see variables embedded in sentences, and not the context of the variable.

✔ **TIP** An advantage to using MadCap Lingo for managing files for translation is that it allows you to select and flatten variables to convert them to text. By doing this, translators retrieve and view the real text only, rather than the tag.

- Avoid using common nouns, or generic terms as variables (e.g., device, machine, instruction guide, manual).
- Only use variables for proper nouns, dates, version numbers, addresses, or non-textual information for targets.
- Avoid using inline variables. It is better to put a condition on a whole sentence. Using inline variables can present improper conjugation and noun-verb agreement problems for languages that are gender-specific.

☆ EXAMPLE

If translating from English to Spanish, consider the following sentence: *The <variable> is tall.*

The variable is either girl, or boy. In Spanish, the article and the adjective change, based on the gender of the noun.

The translation with girl as the variable is *la niña es alta*, and with boy it is *el niño es alto*.

The cleaner solution is to write two sentences, and apply a condition to each one.

Conditions

- Avoid conditions that change sentence structure; meaning do not put more than one condition in a single sentence. Instead, write two simple sentences and condition each. For some languages, the grammar and word form can change depending on singular vs plural elements, the gender of the noun, or verb tense, etc.

☆ **EXAMPLE** In this example, a condition is applied with an “s” to add a plural form to a sentence. In the Flare image below, notice the use of singular and plural in one sentence with condition tags.



The machine is designed for destructing process off-gas up to 25, 50, 75 or 100 m³/h, respectively.

Keep in mind that other languages do not allow the translator to simply add an “s” to make a word plural. In the computer-assisted translation (CAT) tool, the translator is faced with figuring out a rather complicated text string. The following image is what they see in the CAT tool.



```
The machine<pt val="<MadCap:conditionalText MadCap:conditions="DM_conditions/DM_SizeConds.DM25-100" xmlns:MadCap...>s
</pt val="<MadCap:conditionalText">
<pt val="<MadCap:conditionalText MadCap:conditions="DM_conditions/DM_SizeConds.DEFX7" xmlns:MadCap="h...> is
</pt val="<MadCap:conditionalText">
<pt val="<MadCap:conditionalText MadCap:conditions="DM_conditions/DM_SizeConds.DM25-100" xmlns:MadCap...> are
</pt val="<MadCap:conditionalText">
<pt val="<MadCap:conditionalText"> designed for destructing process off-gas up to
<pt val="<MadCap:conditionalText MadCap:conditions="DM_conditions/DM_SizeConds.DM25-100" xmlns:MadCap...> 25, 50, 75 or
100 <pt val="<MadCap:conditionalText">
<pt val="<MadCap:conditionalText MadCap:conditions="DM_conditions/DM_SizeConds.DEFX7" xmlns:MadCap="h...> 7
</pt val="<MadCap:conditionalText"> m<pt val="<sup> 3 </pt val="</sup>/h
</pt val="<MadCap:conditionalText MadCap:conditions="DM_conditions/DM_SizeConds.DM25-100" xmlns:MadCap...>, respectively
</pt val="<MadCap:conditionalText">.
```

The better way to approach this sentence is to create two sentences and apply one condition tag to the entire sentence.

Some might wonder if creating two sentences instead of one will increase translation costs. But if you write two cleaner sentences from the start, that enables the translator to interpret what is going on better in order to produce accurately translated material. This simplified method reduces the likelihood of rework, thereby reducing costs in the long run.

- Avoid using condition tags for internal only notes. If you do, label them as such at the topic level (not within topics) so they get omitted from word count. It is better to create notes with Flare annotations.

Tag Placement

As you can imagine, a lot of tags applied to content intended for translation can reduce content clarity and increase its complexity. Translators have to work around tags, but authors can help the efficiency of translations by optimizing tag placement.

- Reduce the use of, or avoid using inline tags (e.g., no closed tags in the middle of a word).
- Turn hyperlinks into cross-references when possible. Cross-references are easier to handle, and require less integration.
- Avoid inline styles. Create style classes in the stylesheet instead.

Formatting for Translation

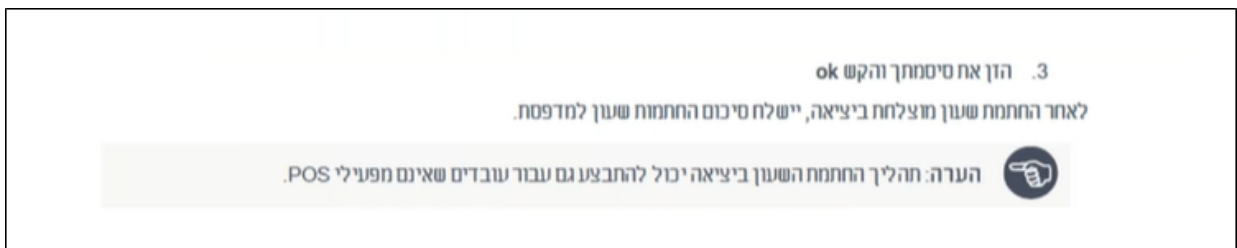
After having files translated, you might have to make adjustments to text or layouts to fine-tune formatting for the target language publications. If a Flare project is set up well and optimized, that can help to mitigate formatting issues later on.

When it comes to formatting translated documentation, keep in mind the following:

- **Use CSS Stylesheets and Avoid Inline Formatting** Applying inline styles (or local formatting) instead of formatting content using a stylesheet can make it hard to apply changes quickly. When inline styles are set, those styles override properties in the stylesheet, and so changes to that inline formatting must be done manually; making the process more time-consuming and expensive.
- **Text Expansion** Many times, formatting translated files is necessary due to text expansion. This is when translated text is significantly shorter or longer (up to 30% or more) than the source text. Chinese or Japanese tends to be shorter, while German tends to be much longer.

✔ **TIP** Ask your translator in advance to perform a pseudo translation (a process that tests translating files into another language). This should include a text expansion of at least 30%. This might detect troublesome formatting areas in your documentation where there might be text expansion.

- **Directional Icons** Make sure that icons are reusable in the target locations. For example, a hand icon pointing to the right is appropriate for left-to-right languages, but it would have to be adapted for a different reading direction, to accommodate right-to-left languages (such as Hebrew or Arabic).



- **A4 Paper Size** It is important to identify what publications are needed from your targets, and ask yourself if they might work overseas. Be aware that international paper sizes differ from those used in North America and a few other countries. Because of this, make a template page format in the A4 size, and make page layout designs flexible. Consider using external anchors to margins, specifying height or width for a header and footer, and not using hard-coded widths and heights for the body frame.

Using Images for Translation

When it comes to working with images for a translation project, it is highly beneficial to use both the MadCap Flare and Capture products together. Capture caters to the translation workflow to mitigate common issues that might otherwise occur with images. Images not only illustrate what you need to say, but they serve as valuable tools for reducing the overall word count in a document. Consider the following when using images for translation:

- **Use MadCap Capture** Since the Capture application is closely integrated with Flare, they are both optimized for translation. Capture is the only application that makes translating graphic callouts easy and possible. It does this by writing the image callout text to its own XML file (i.e., a PROPS file), which accompanies the image file. A translator can translate the text in the XML file, and if using Capture to view the image, can account for any text expansion right away. This text is also added to your project's translation memory.
- **Do Not Embed Text** Try not to embed text in graphics that need to be translated. Instead write the text in a table, as a bulleted list below the graphic, or don't use text at all.
- **Text Expansion** Be mindful that the appearance of your graphics might change if the translated text is significantly shorter or longer than the source text. You can opt to simply not add text to your images, or you could employ MadCap Capture to work with Flare, since the two applications handle this problem seamlessly.

If you have inserted MadCap Capture images that contain objects with text, you can auto-size those objects when the output is generated. This is done by selecting an option in the Advanced tab of the Target Editor. The original image file and its associated properties (PROPS) file remain unchanged. Only the output image is affected.

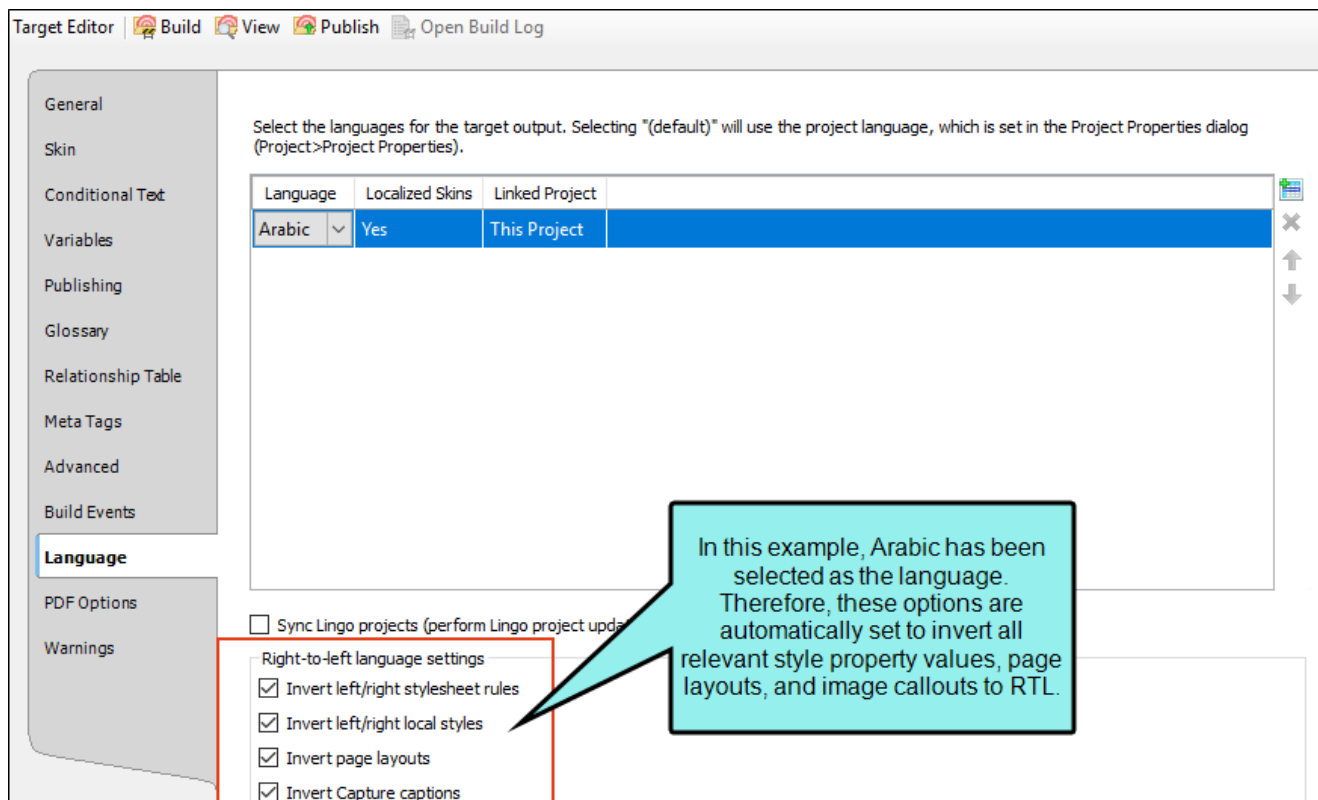
- **Provide Files to Translator** If you are using other programs such as PhotoShop, Illustrator, Visio, etc., be sure to provide translators with source files and all layered files for images. This allows translators to create images to the target language properly.
- **Use Simple Images** Generic and neutral images are appropriate for a worldwide audience.

Other Information for Translation Best Practices

In addition to the primary concepts, there is some additional information that might help you when working with this feature.

Inverting Styles, Page Layouts, and Image Callouts

In the Language tab for the target there are multiple options that are selected by default when you choose a right-to-left (RTL) language at the project or target level. The options are used to automatically invert the following: (1) language-related style rules locally, (2) language-related style rules in the stylesheet, (3) image callouts from MadCap Capture, and (4) page layout settings.



The options that are seen depend on which output type you are using.

- **PDF/Word** All four options are shown.
- **HTML5/HTML Help/EPUB/WebHelp/WebHelp Plus** Local styles, CSS styles, and image callout options are shown.
- **DITA** No options are shown.

These options can be quite useful after you receive the project back from a translator. When you generate the output, the inversion of the styles and page layouts takes place.

☆ **EXAMPLE** You have a project in English, a left-to-right (LTR) language, and you need to have it translated into Arabic, a right-to-left language.

For your p style, suppose you have a left margin of 5 px and a right margin of 30 px.

Those settings work fine for your targets using the LTR language. But for the RTL language, you need the settings to be flipped so that the left margin is 30 px and the right margin is 5 px.

So after you receive the project back from the translator, you make sure the RTL language in Flare is set at either the project or target level.

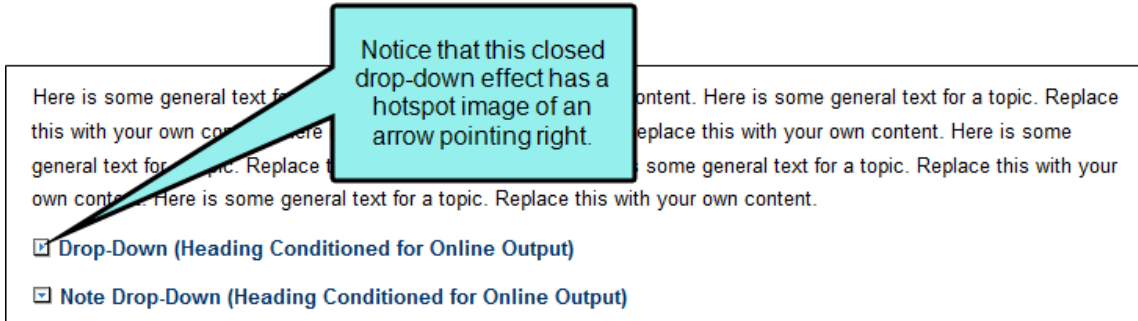
As a result, the inversion options become selected automatically, and when you generate the output, paragraphs will have a left margin of 30 px and a right margin of 5 px.

📄 **NOTE** If your selected language is LTR, these options cannot be accessed in the Target Editor.

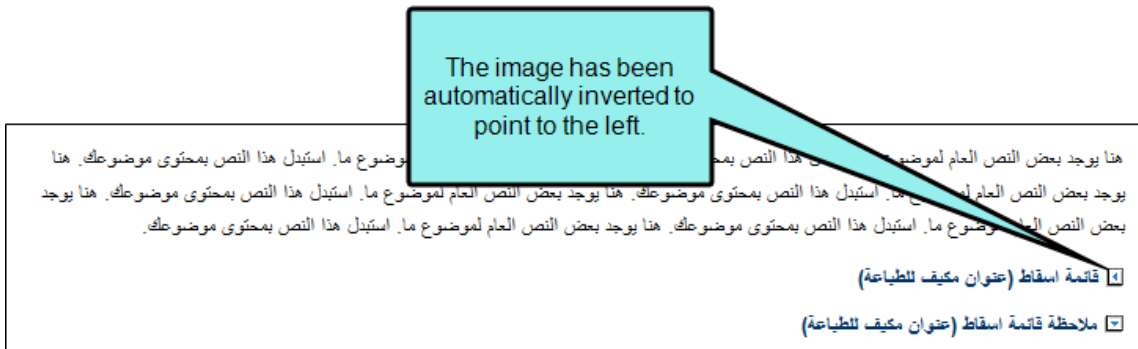
Inverting Hotspot Images

There are certain features in Flare that use special hotspot images (e.g., drop-down and expanding text effects). Flare's default images for these features are designed to work with left-to-right (LTR) outputs. If you are using a right-to-left (RTL) language, Flare automatically inverts the default images so that they make sense for that direction.

☆ **EXAMPLE** You have an English project with drop-down effects, like this:



If you look at the same topic in an Arabic project, it will look like this:



This inversion occurs in any of the following features that use hotspot images.

- Editing Drop-Down Text.
- Editing Expanding Text.
- Editing Toggles.
- Editing Keyword Links.
- Editing Related Topics Links.
- Editing Shortcut Controls.

This is also done in HTML5 skins and Standard skins for any items that use images where background settings can be specified.



NOTE Custom images that you create and add are not inverted.

Table Styles and Right-to-Left Languages

If you are using a right-to-left (RTL) language, table styles in your project need to write additional cascading stylesheet (CSS) rules behind the scenes in order to work correctly with RTL tables. Because this can potentially double the size of the table style file, this behavior does not happen by default if you create and save a new table style. However, the behavior kicks in automatically in the following two scenarios.

- If you open a topic in the XML Editor and an RTL table references an old table stylesheet, Flare updates and saves the table stylesheet in your Content Explorer folder.
- If you generate output and an RTL table references an old table stylesheet, the table style in the output is updated.

I Single Language vs. Multilingual Outputs

When translating projects, you can use single language or multilingual output.

Single Language Output

With single language output, one target is generated for each language. In most cases, this means having a separate Flare project for each language, and the target would generate output for that language. A less common situation is where you have one Flare project and a multilingual author is translating content for each language. A unique target for each language would be created and separate output generated for each language.

Multilingual Output

With multilingual output, one target is generated for multiple languages. There are a couple of ways to do this:

- **Multilingual Targets** Each translator provides you with a Flare or Lingo project that contains the translated files. In the original Flare project, you would use the Target Editor's Language tab to point to each of the translated projects. The generated target combines all of the translations into the same output. See "Creating Multilingual Targets" on the next page.
- **Stitching PDFs** You drag translated PDFs in different languages into a table of contents (TOC), which serves as an outline for a PDF target. When you generate the PDF target, the result is a single PDF with one language following another. Translated PDFs can also be added to the TOCs for online output. See "Stitching PDFs" on page 52.


Creating Multilingual Targets

Supported In:



At the target level, you can link translated Flare or Lingo projects to your current Flare project. When the target is generated, the output will consist of content in each of those languages. Also, with HTML5 output, you can include a drop-down for users to manually switch between languages. See "Multilingual Target Examples" on page 39.


How to Create a Multilingual Target

1. From the Project Organizer, double-click a target.
2. In the Target Editor, select the **Language** tab.
3. Link to translated projects in other languages.
 - a. Click . Another row is added to the grid.
 - b. In the **Linked Project** column, click the link to select the Lingo or Flare project you want to link to your current project.

If you have not added a linked project yet, the link looks like an ellipsis; if you have added a linked project, the link displays the project's file name.


Language	Localized Skins	Linked Project
(default) [English (United States)]	Yes	This Project
Slovak	No	...






 **NOTE** You can link to many separate Lingo projects, or you can link multiple rows in this tab to the same multilingual Lingo project, or both.

- c. In the dialog, find and select the project you want to link to your current project. Click **Open**.
- d. If necessary, select the linked project's language from the **Language** drop-down. Flare automatically detects the linked project's default language settings, so you only need to update this setting if you are pointing to a multilingual Lingo project, or if it is incorrect in the linked project.

Additionally, if default translations are available for the selected language's skin, it is noted in the **Localized Skins** column. If you have a custom language skin for the selected language in your linked project, Flare automatically uses your custom translations. See "Adding Language Skins" on page 63.


 **NOTE** If you link to a multilingual Lingo project, only the available languages from a multilingual project can be selected from the drop-down.

 **NOTE** The default target output language in the Target Editor always shows the currently selected project language. This language updates automatically if you change the project language.

- e. (Optional) Continue adding rows and linked projects for each additional target output language.
 - f. (Optional) Use the  and  buttons to rearrange your projects. They display in the output in the order they are listed on the Language tab (e.g., if you include a drop-down for users to choose each language in HTML5 output, or if you are working in a PDF target where one language will follow another in the output).
4. (Optional) If you have added links to one or more Lingo projects, you can select **Sync Lingo projects** if you want to synchronize updates with those projects.

If this option is enabled, the application detects whether any of the Lingo and Flare source files are out of sync. If they are, the Lingo project is automatically updated and these changes are also brought into the parent Flare project. This is different from the usual process, where

the translator would normally update the Lingo project manually and translate the changed or new files.

 **WARNING** You might use this feature if you want to quickly see any updated files in your parent Flare project, including non-translated content such as images. However, enabling this option is typically not recommended, because there is always the risk of updating the Lingo project (and therefore also the parent Flare project) with content that has not yet been translated.


5. (Optional) If you selected a right-to-left (RTL) language, you will see the following options at the bottom, which are enabled by default for RTL languages. Leave them selected to automatically invert the following: (1) language-related style rules locally, (2) language-related style rules in the stylesheet, (3) image callouts from MadCap Capture, and (4) page layout settings.

- **Invert left/right local styles**
- **Invert left/right stylesheet rules**
- **Invert Capture captions**
- **Invert Page Layouts**

The options that are seen depend on the output type you are using.

- **PDF/Word** All four options display.
- **HTML5/HTML Help/EPUB/Eclipse Help/WebHelp/WebHelp Plus** Local styles, CSS styles, and image callout options display.
- **DITA** No options display.

These options are useful after you receive the project back from a translator. When you generate the output, the inversion of the styles and page layouts takes place.


 **EXAMPLE** You have a project in English, a left-to-right (LTR) language, and you need to have it translated into Arabic, a RTL language.

For your paragraph (p) style, suppose you have a left margin of 5 px and a right margin of 30 px.

☆ Those settings work fine for targets using the LTR language. But for the RTL language, you need the settings to be flipped so that the left margin is 30 px and the right margin is 5 px.

So after you receive the project back from the translator, make sure the RTL language in Flare is set at either the project or target level.

As a result, the inversion options are automatically selected. When output is generated the paragraphs have a left margin of 30 px and a right margin of 5 px.

 **NOTE** The option to invert page layouts controls every aspect of the page layout file (e.g., it inverts not only frames and content within them, but also styles applied to frame content). The other invert options have no effect on page layouts.

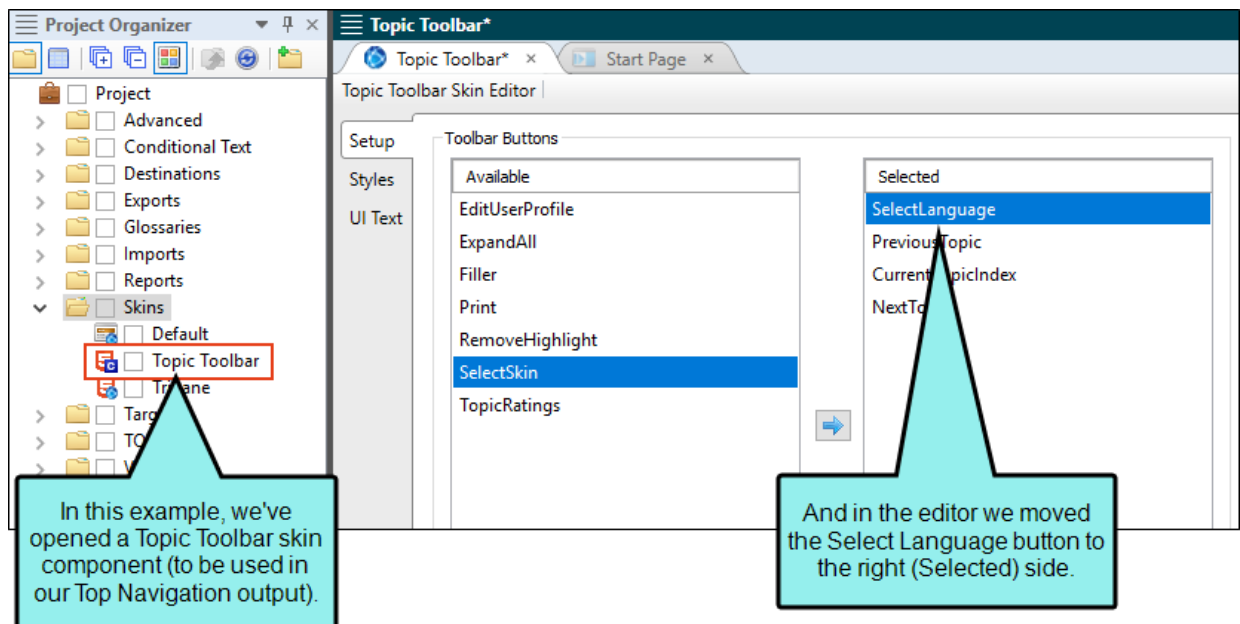
 **NOTE** If your selected language is LTR, these options cannot be accessed in the Target Editor.

6. Click  to save your work.

How to Add a Language Drop-Down for HTML5

With a multilingual target, Flare creates a subfolder in your output folder for each language. When you view the output, it is automatically displayed based on your browser's default language. However, when producing HTML5 output, you also have the option of including a drop-down in the skin. This allows users to switch manually from one language to another.



1. Do one of the following, depending on the output type:
 - **Side and Top Navigation** Add a Topic Toolbar skin component to your project . Open the skin component, and select the **Setup** tab.
 - **Tripane** Open the regular skin and select the **Toolbar** tab.
2. Move the **Select Language** button from the **Available** section to the **Selected** section.

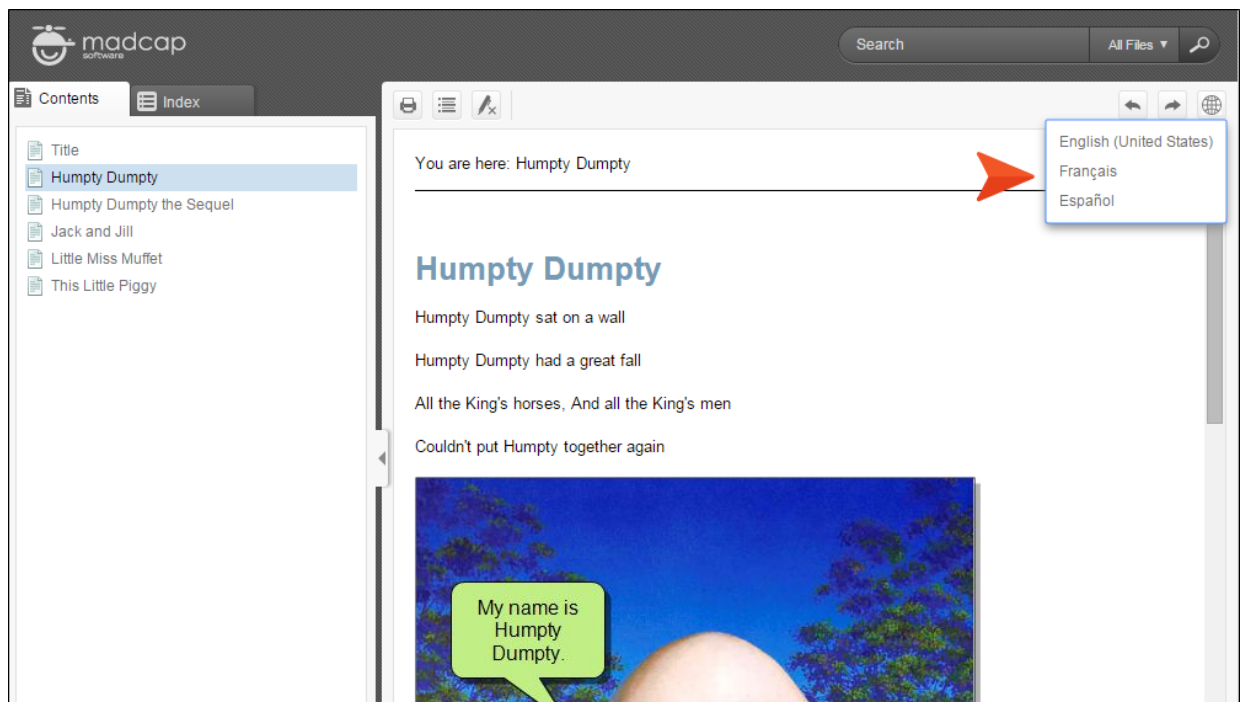


3. For Side Navigation and Top Navigation output, insert a Topic Toolbar proxy into your template page(s), and select the Topic Toolbar skin component when doing so.


If you decided not to create a Topic Toolbar skin component in Step 1, you can click **Select buttons** in the Topic Toolbar Proxy dialog and enable the **Select Language** button from there.


If you are creating Tripane output, you do not need to insert a proxy.


4. Click  to save your work. When you generate the multilingual target, you will see the language button  in the output, and you can click it to select any of the languages that are linked in the target.





What's Noteworthy?


 **NOTE** You must link each language to a Flare or Lingo project before you can close the Target Editor. If you make changes to your linked projects and their links are no longer valid when you build the project, a warning message displays, and it will be unable to build.


 **NOTE** When you build a target that is set up for multilingual output, and you link directly to a Lingo project, the export process runs automatically in Lingo so that the parent Flare project grabs the necessary translated Flare projects. If the Lingo export process encounters warnings, these will not display with the other build warnings in Flare's interface. Instead, you must open the build log to find any such warnings.


 **NOTE** In order to build multilingual output that links directly to multilingual Lingo projects, you must have Lingo 10 or newer installed on the same computer.

 **NOTE** Since Flare generates output from the linked targets in each of your project files, each linked project must have its own target file (e.g., if you are creating PDF output, each linked project needs its own PDF target). If you do not have a target file, a warning message displays before the build starts and you will be unable to build. Target files must be in the same relative location in each project.

 **NOTE** Be sure that file names are the same for your parent project and for each of your linked projects. This is important for HTML5 targets, so you can switch between languages using the Select Language button in the output.

 **NOTE** Multilingual Elasticsearch publishing requires all targets to use the same GUID. This is set on the Search tab of the Target Editor. If the GUIDs in the targets are different, you might have empty targets on the portal or experience a publishing failing with error 400. If necessary, open targets in the Internal Text Editor to modify the GUID.

 **NOTE** When generating localized HTML Help targets, it is sometimes necessary to set the Windows system locale to match the language that the project is set to. It is necessary to do this when the project contains topic file names with non-English characters. To do this in Windows: 1. Open the **Control Panel**; 2. select **Regional and Language Settings**; 3. select the **Advanced** tab; 4. from the drop-down in the section called **Language for non-Unicode programs**, choose the same language that the Flare project is set to; 5. restart Windows.

 **NOTE** If you are building Eclipse Help, you will need to open your Output folder to open your desired language output.

Multilingual Target Examples

Following are examples of how to create multilingual targets in two ways—by linking to Flare projects and by linking to Lingo projects.

Linking to Flare Projects

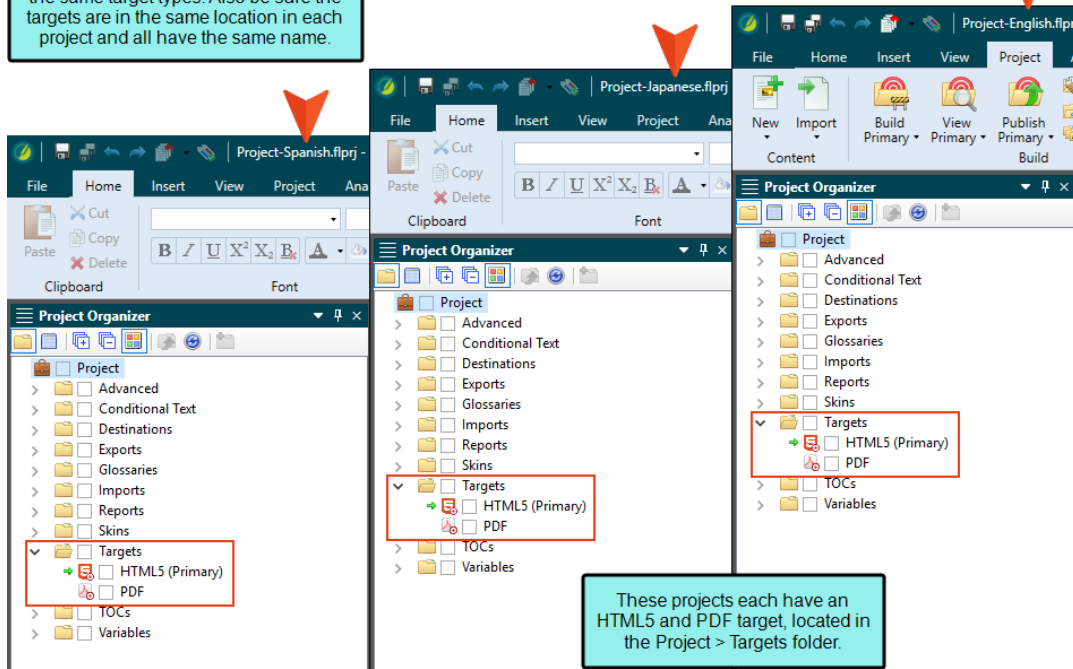
In the example below, a Flare target points to other Flare projects that have been translated. This example illustrates the process of adding a button (via a skin) so users can switch between languages in online output. It also shows how PDF output is generated from a multilingual project.

☆ **EXAMPLE** You want to create HTML5 and PDF outputs in US English, Spanish, and Japanese. You have already worked with a translator to translate your documents, so you have three different Flare projects: one for each language.


Before you can create your multilingual output in Flare, you make sure each Flare project has both an HTML5 and a PDF target available. This allows your parent project to build from the targets in the other two linked projects. You open the Project Organizer in each project, and then open the Targets folder. If you already have the targets you need, you do not need to create any new ones. If you do not have both targets in all three projects, you can create the missing targets from the Add File dialog.

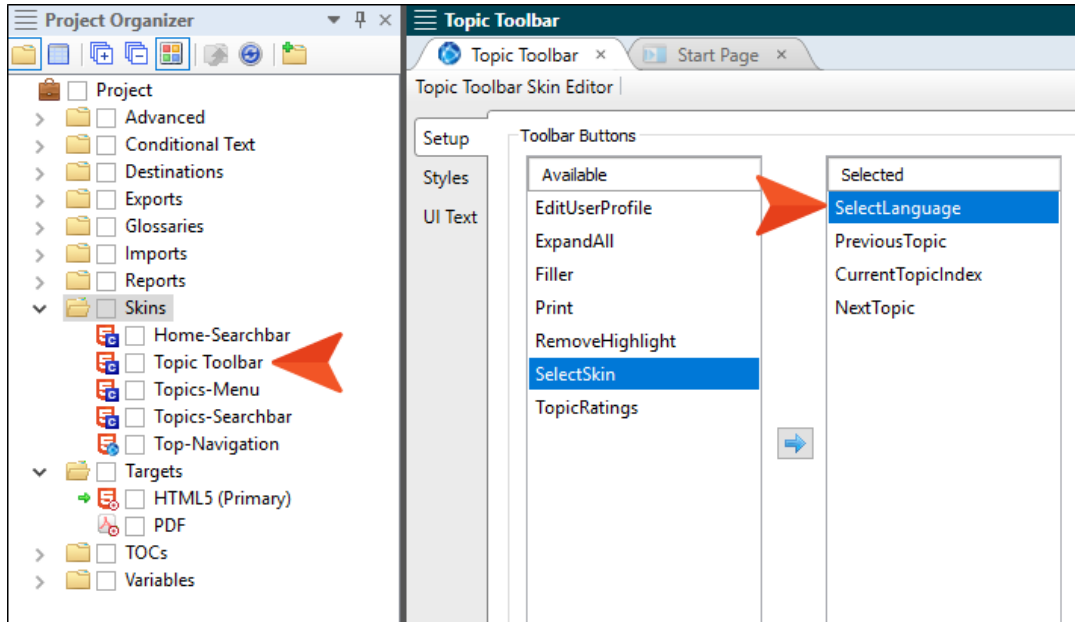


Make sure that all of your projects have the same target types. Also be sure the targets are in the same location in each project and all have the same name.




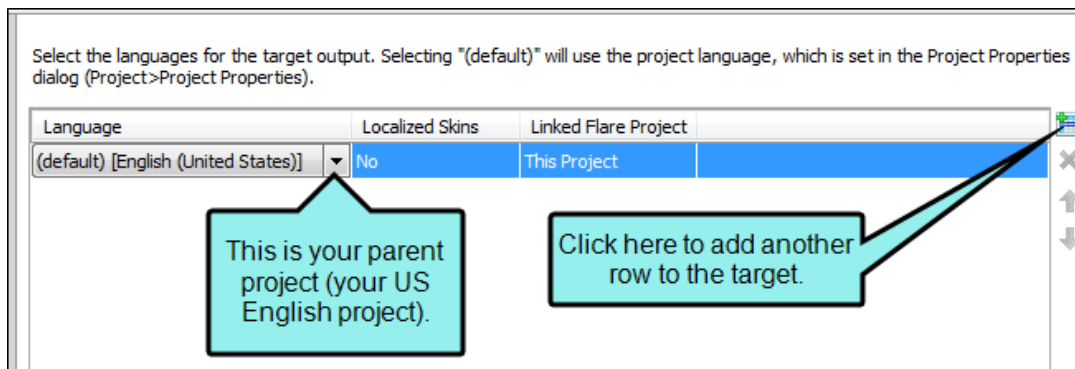
Because you want to build multilingual HTML5 output where users can manually switch between languages, you need to make sure that each of your projects has the Select Language button in its HTML5 toolbar.

So you open the Project Organizer. In the **Skins** folder, you open the **Topic Toolbar** skin component. On the **Setup** tab, you select the **Select Language** button, click  to add it to your skin, and then save your work.



Next, you insert a Topic Toolbar proxy in the template page(s), and when doing so, you choose the Topic Toolbar skin component that you created.

Now you can prepare your multilingual target. In the parent project (in this case, the US English project), you expand the **Targets** folder and open the **HTML5** target. In the Target Editor, you select the **Language** tab. This is where you will link your other two projects. On the right side of the tab, you click  to add a new row.



☆ Then you click the link (ellipsis) in the **Linked Flare Project** column.

Select the languages for the target output. Selecting "(default)" will use the project language, which is set in the Project Properties dialog (Project>Project Properties).

Language	Localized Skins	Linked Flare Project
(default) [English (United States)]	No	This Project
Afrikaans	No	...

You don't need to select a language yet... Flare will detect your project's default settings when you link a new project.

Click here to link a new Flare project to your parent project.

In the dialog that opens, you navigate to the Spanish project and click **Open**. This links the Spanish project to the US English project. The Language drop-down automatically updates when Flare detects the default Spanish language settings in your linked project.

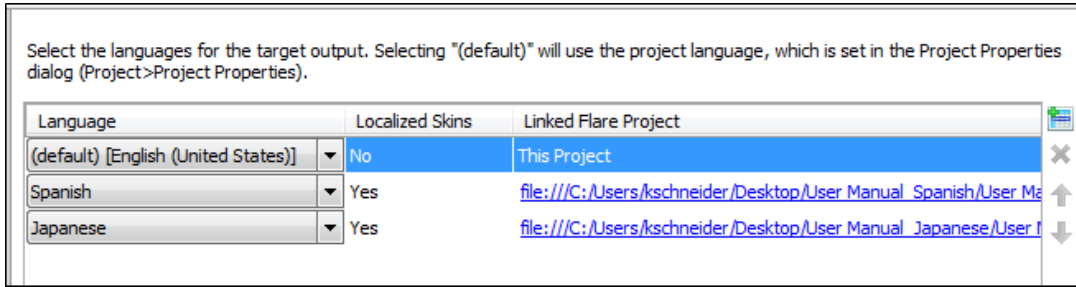
Select the languages for the target output. Selecting "(default)" will use the project language, which is set in the Project Properties dialog (Project>Project Properties).


Language	Localized Skins	Linked Flare Project
(default) [English (United States)]	No	This Project
Spanish	Yes	file:///C:/Users/kschneider/Desktop/User Manual_Spanish/User Me

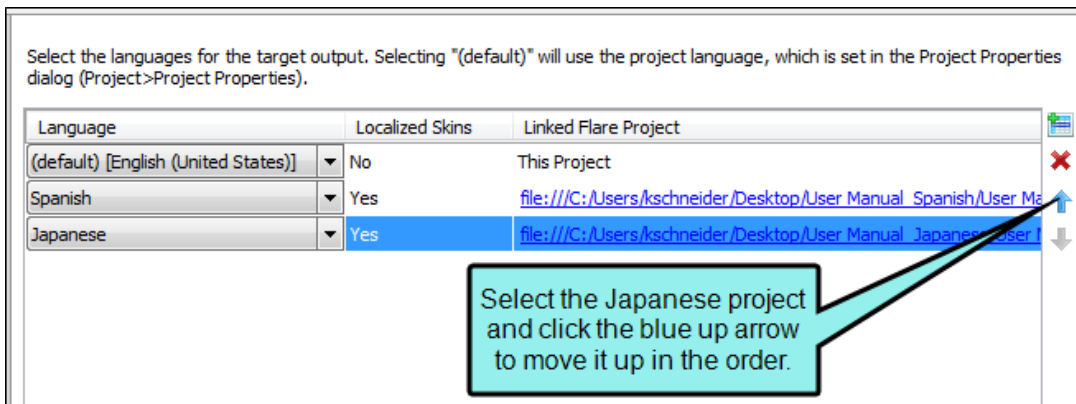
When you link the project...

...Flare detects that its default language is Spanish.

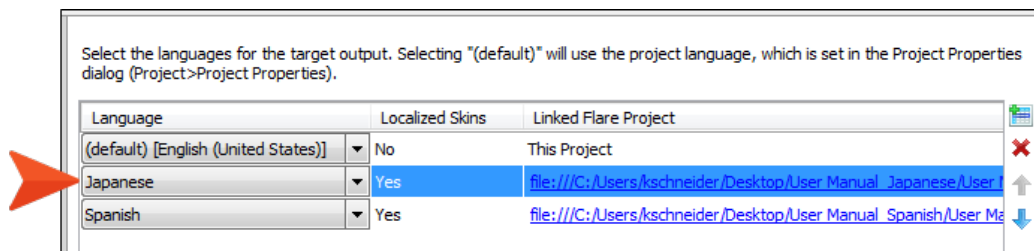
☆ Then you link to the Japanese project in the same way.



You want the Japanese project to display second in the Help system (after US English), so you select the Japanese project and click  to rearrange the projects.

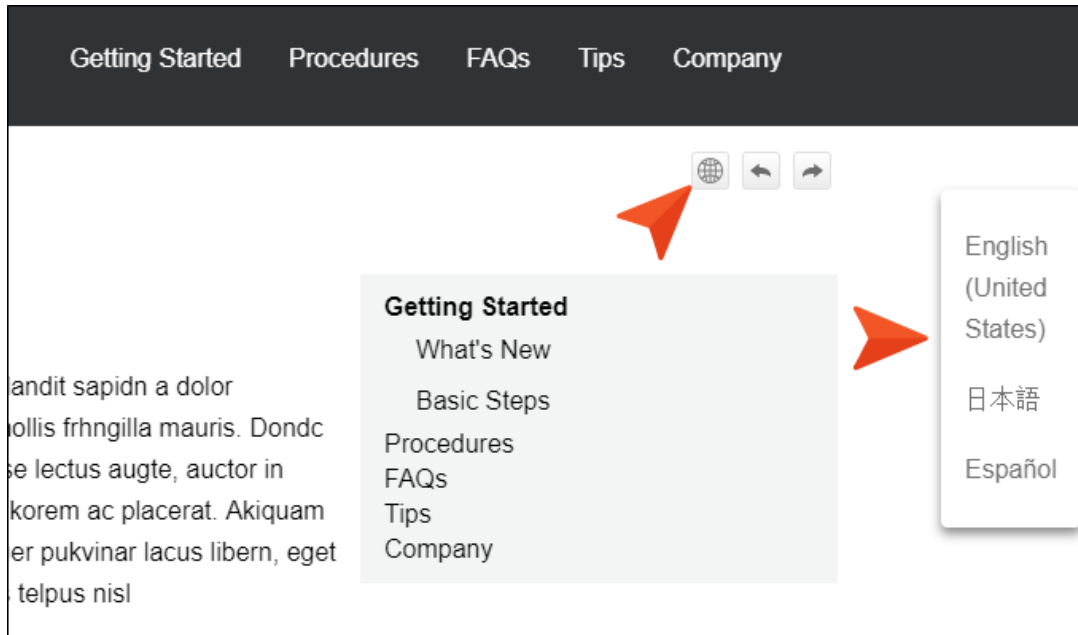


Now the Japanese project is listed before the Spanish project.



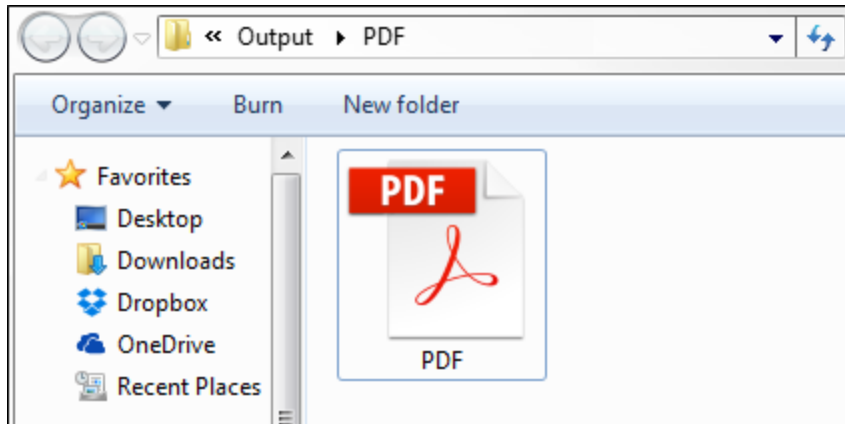
You are ready to build, so you save your work and build the target. When you open the HTML5 output, it defaults to the US English Help because that is your browser's default setting.

☆ But remember the Select Language button you added to your Topic Toolbar skin component? You can use that button to change the language setting using a drop-down menu.

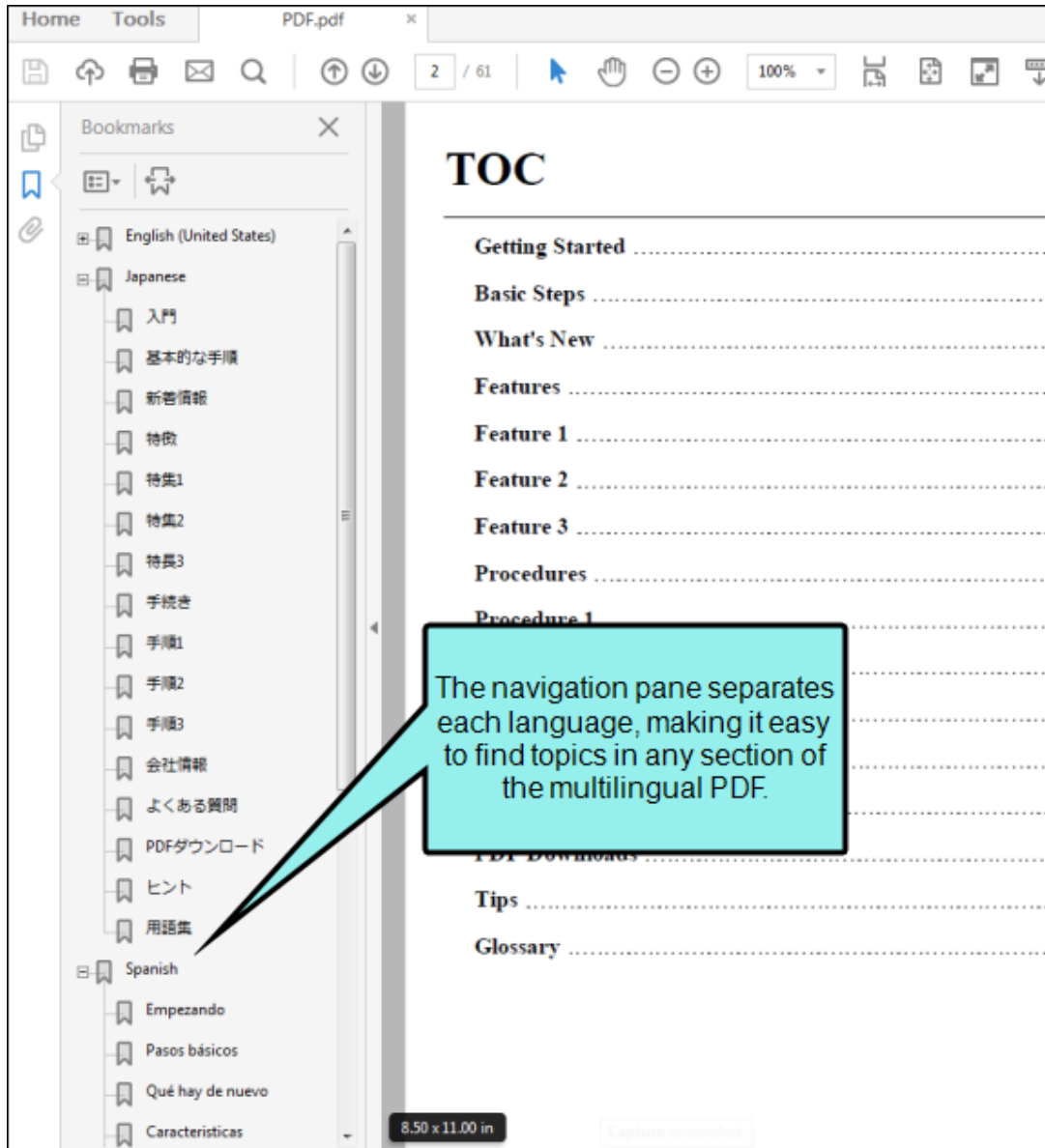


Next you open the PDF target in the Target Editor and adjust it in the same way you did the HTML5 output. Since you want the Japanese output to display immediately after the US English output, you add that project first, then you add the Spanish project.

- ☆ You save the PDF target and build it. When you look in the Output folder, you only see one document. This is because Flare stitched all three documents together into a single PDF.



- ☆ When you open the PDF, you see all three languages in the document's table of contents. Each language has its own title page so you can find it quickly.



Linking to Lingo Projects

In the following example, a Flare target points directly to Lingo projects. This is a more direct approach because it doesn't require the creation of additional Flare projects.

☆ **EXAMPLE** You want your US English Flare project to be translated into Arabic, French, German, and Spanish. You have one translator who knows French, German, and Spanish, and you have a second translator who knows Arabic.

Because the first translator knows three languages, there is no reason to have multiple Lingo projects for each language. Instead, the translator adds all three languages to a single Lingo project and translates the files.

Meanwhile, the second translator creates a Lingo project and uses it to translate the Flare project into Arabic.

Both translators keep their Lingo projects on a server where you have access them. When the translation work is finished, you open the target in Flare, and on the **Language** tab, add a row for each language.

In the first row, you link to the Lingo project that was used for the French, German, and Spanish translations. After you add the first row, you notice that you can select any of the three languages.



Select the languages for the target output. Selecting "(default)" will use the project language, which is set in the Project Properties dialog (Project>Project Properties).

Language	Localized Skins	Linked Project	
(default) [English (United States)]	Yes	This Project	✖
French	Yes	../../../../My Translated Projects/Nursery Rhyme	↑
French			↓
German			
Spanish			

Any of the three languages can be selected.

We've linked to a Lingo project containing multiple languages (French, German, and Spanish).

Sync Lingo projects (perform Lingo project updates as needed)

- ☆ For the first row, you select **French**, and then add a second row. This time, you can only select German or Spanish (because French has already been used).

Select the languages for the target output. Selecting "(default)" will use the project's default language. Selecting a language will use the project's localized skins. Selecting "(default)" will use the project's default skins. Selecting a language will use the project's localized skins. Selecting "(default)" will use the project's default skins. Selecting a language will use the project's localized skins.

Language	Localized Skins	Linked Project
(default) [English (United States)]	Yes	This Project
French	Yes	../../../../My Translated Projects/Nursery Rhyme
German	Yes	../../../../My Translated Projects/Nursery Rhyme
Spanish	Yes	../../../../My Translated Projects/Nursery Rhyme

Sync Lingo projects (perform Lingo project updates as needed)

- ☆ For the second row, you choose **German**, and then you add a third row. **Spanish** is automatically chosen for that row, because it is the only one left.

Select the languages for the target output. Selecting "(default)" will use the project language, which is set in the Project Properties dialog (Project>Project Properties).

Language	Localized Skins	Linked Project	
(default) [English (United States)]	Yes	This Project	✖
French	Yes	../../../../My Translated Projects/Nursery Rhyme	↑
German	Yes	../../../../My Translated Projects/Nursery Rhyme	↓
Spanish	Yes	../../../../My Translated Projects/Nursery Rhyme	

- ☆ Finally, you create a fourth row and link it to the Lingo project used for the **Arabic** translation.

Select the languages for the target output. Selecting "(default)" will use the project language, which is set in the Project Properties dialog (Project>Project Properties).

Language	Localized Skins	Linked Project	
(default) [English (United States)]	Yes	This Project	✖
French	Yes	../../../../My Translated Projects/Nursery Rhyme	↑
German	Yes	../../../../My Translated Projects/Nursery Rhyme	↓
Spanish	Yes	../../../../My Translated Projects/Nursery Rhyme	
Arabic	Yes	../../../../My Translated Projects/Nursery Rhyme	

Stitching PDFs

You can stitch existing PDFs into some types of output by adding links to them in a table of contents (TOC). This is an alternative to stitching PDFs via a multilingual target (see "Creating Multilingual Targets" on page 32).

How to Stitch PDFs Into Flare Output

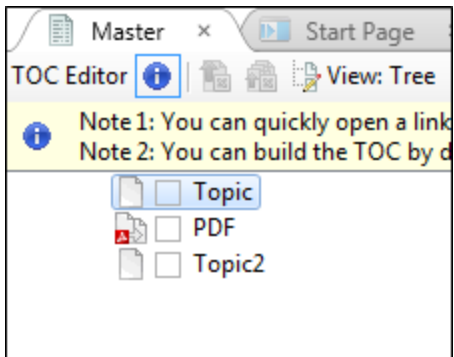
1. Add the finished PDF(s) to the Content Explorer in your project. The easiest way to do add PDFs to a Flare project is to simply paste them into the Content Explorer. You also might consider using the External Resources feature, which allows you to keep the PDFs in sync over time with the original copies if changes are made to them. For more about External Resources, see the online Help.
2. In Flare open a TOC file.
3. In the Content Explorer or File List window pane, click the PDF file and drag it to the TOC.
4. Generate the target.

PDF Targets

If you generate a PDF target, the stitched PDF pages will be inserted at the location represented by the entry in the TOC.

If you insert a finished PDF in the middle of topic entries in the TOC, the topics after the inserted PDF will increment page numbers accordingly.

☆ **EXAMPLE** You have a TOC like this, with a finished PDF inserted between two topic entries:



☆ In the output, the stitched PDF is sandwiched after the first topic and before the second one.

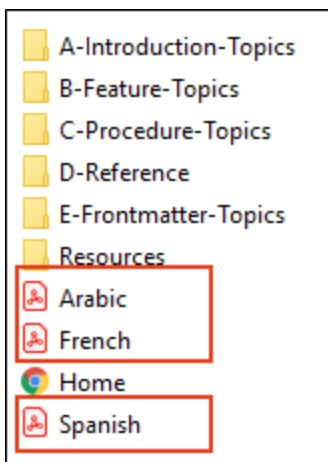


☆ If the existing PDF had not been included in the TOC, the second topic would normally have shown page 3 (page 1 = generated TOC, page 2 = first topic, page 3 = second topic). But because the stitched PDF has been included, the second topic starts on page 6 (page 1 = generated TOC, page 2 = first topic, pages 3-5 = stitched PDF, page 6 = second topic).

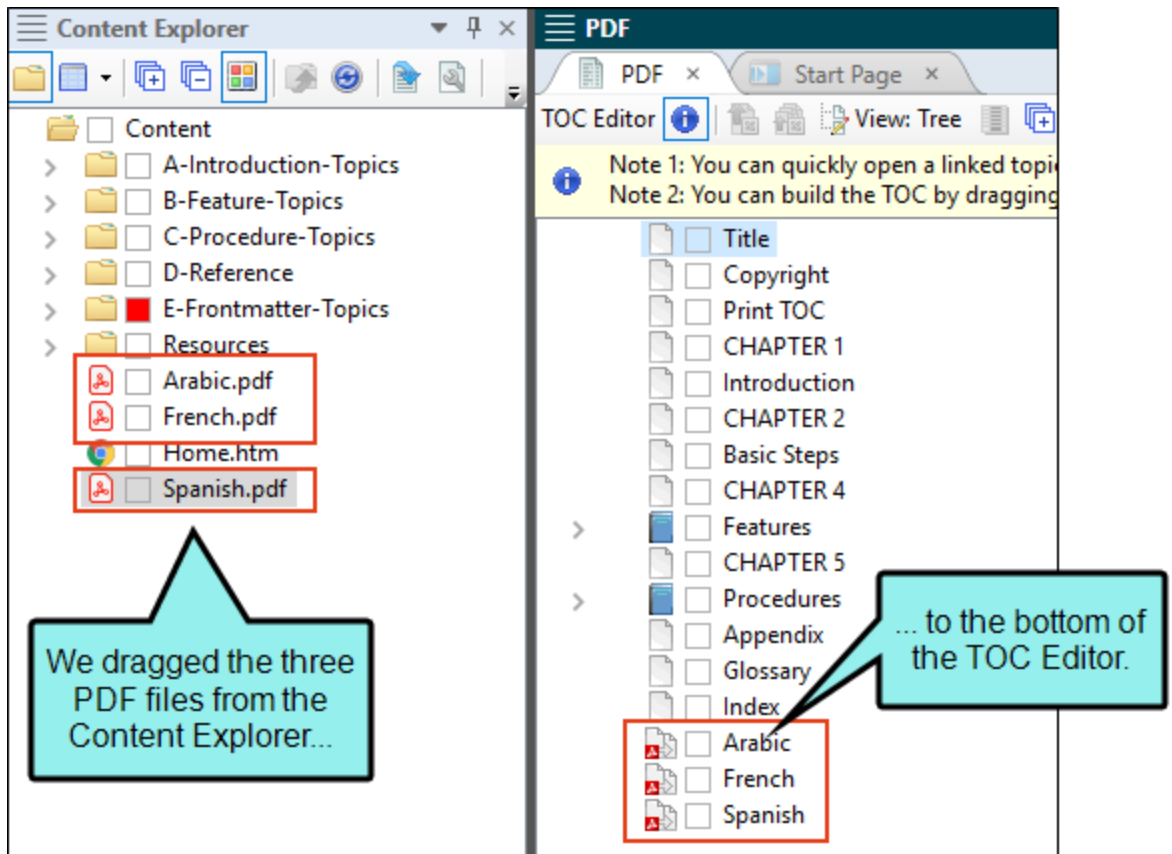
The PDF stitching feature can be especially useful if you have created multiple PDF versions of your documentation in different languages. Each existing PDF could be a version of the content in a unique language.

☆ **EXAMPLE** You have an English project, which you send away to be translated into Arabic, French, and Spanish. At the end of the translation process, you've got three PDF files, one for each of those languages.

In Windows, you copy those three PDFs into the Content subfolder where your project is located.

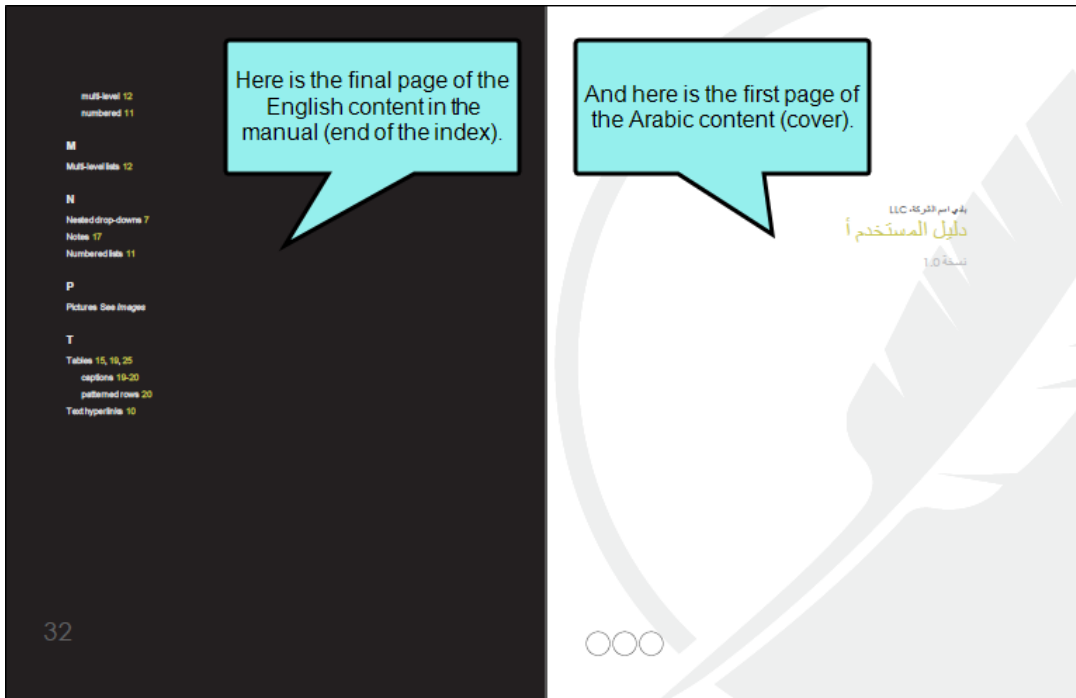


- ☆ In Flare you open the TOC that you are using to generate the PDF output for your English content. Then you drag and drop the three PDFs to the bottom of that TOC.



You don't necessarily need to put the PDFs at the end of the TOC; they can actually be placed anywhere. But we put them at the bottom because we want the final stitched PDF to move in order from English to Arabic to French to Spanish.

- ☆ After generating the final PDF target, the other PDFs are stitched into the output along with the English content.



- 📄 **NOTE** Generating a PDF output using a multilingual target uses the same process as PDF stitching. However, creating a multilingual target automates the process and lets you work from multiple Flare projects at once. Because you do not need to generate each PDF every time the document is updated, producing output from a multilingual PDF target saves considerable time.


Online Targets


If you generate online targets, the linked PDF appears in the TOC like other items.



When the end user clicks that TOC item, the linked PDF file opens in a browser.

What's Noteworthy?

 **NOTE** A stitched PDF cannot link to any of the other content generated by the target, and vice versa. This is one reason the feature is called "stitching" instead of "merging."

 **NOTE** It is not necessary to have project content within the TOC at all. You could simply add links to existing PDFs in a TOC, stitching them together in the output.

Language Skins

Supported In:



Flare supports regular skins and language skins; and out of the box Flare offers several common languages in regular skin files. A regular skin controls how the interface looks for online outputs, whereas a language skin is used to display the interface in a specific language, mostly for online output but there are also some print-based uses for it. The following are reasons why you would use a language skin in addition to a regular skin.

- You need to work in a language that Flare does not support with regular skin files.
- You can enter an alternate translation to the default user interface (UI) text string in a regular skin, but if you use a language skin for this, the alternate translation can be used for all skins in a project.
- You need to modify UI text strings that are not available in a regular skin, such as a cross-reference, drop-downs, etc.

General Information

- "Localized vs. Non-Localized Skins" below

Main Activities

- "Adding Language Skins" on page 63
- "Editing Language Skins" on page 65

Localized vs. Non-Localized Skins

Flare provides completed language skins for certain languages, such as French, German, and Spanish. These are identified as being "localized" skins when you select a project or target language.

For other languages, you can create language skins so they can be applied to a project or target. After doing this, the skin is identified as "localized." These files are stored in the Advanced folder of the Project Organizer.

☆ **EXAMPLE** You select French as the project language.

Use the table below to set the language of the project

Languages with installed spell check dictionaries are displayed in bold.

Hide languages with no spell check dictionary

Language	Localized Skins
English (United States)	Yes
English (Zimbabwe)	No
Estonian	No
Faeroese	No
Filipino (Philippines)	No
Finnish	Yes
French	Yes
French (Belgium)	No
French (Canada)	No
French (France)	No
French (Luxembourg)	No
French (Monaco)	No

Notice that French has a "Yes" next to it in the Localized Skins column. This means that the language skin is already translated into French for you.

If you generate output, the French skin is automatically used, so the output looks something like this:

YOUR LOGO

Glossaire

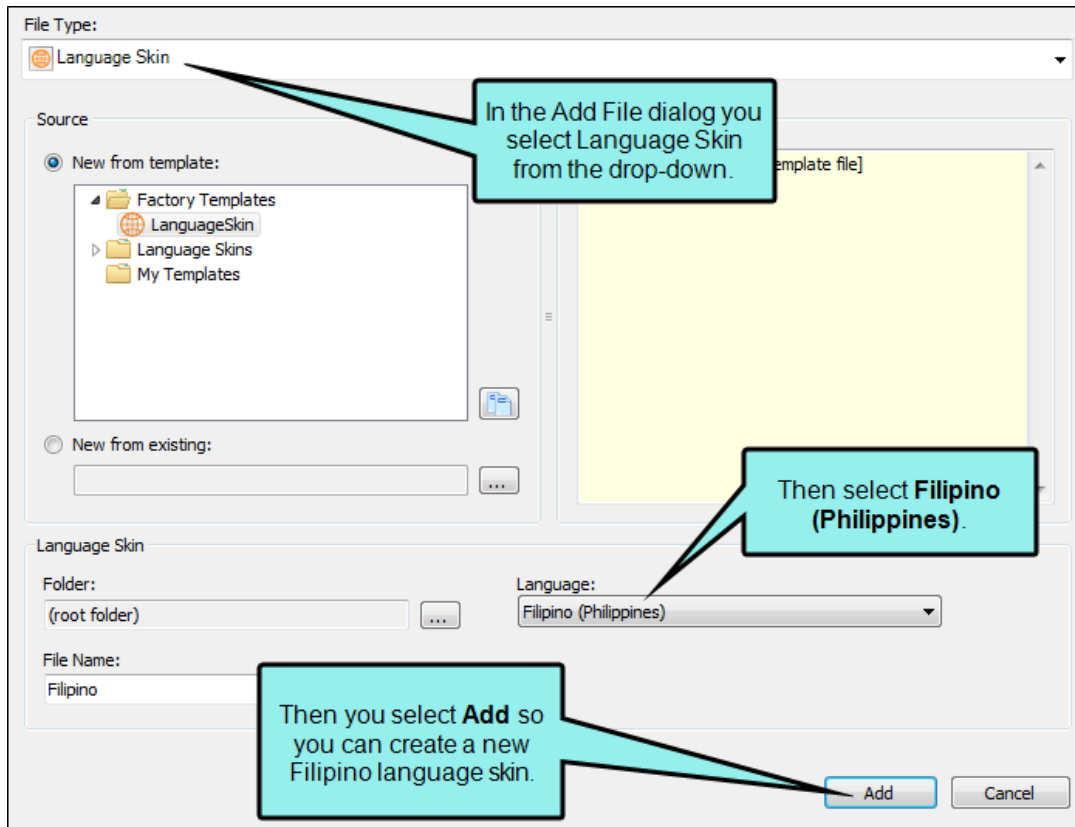
Rechercher dans le glossaire

- Bateau
- Vélo
- Voiture

Notice that the labels for the glossary tab and filter are already displayed in French. You do not need to do anything.

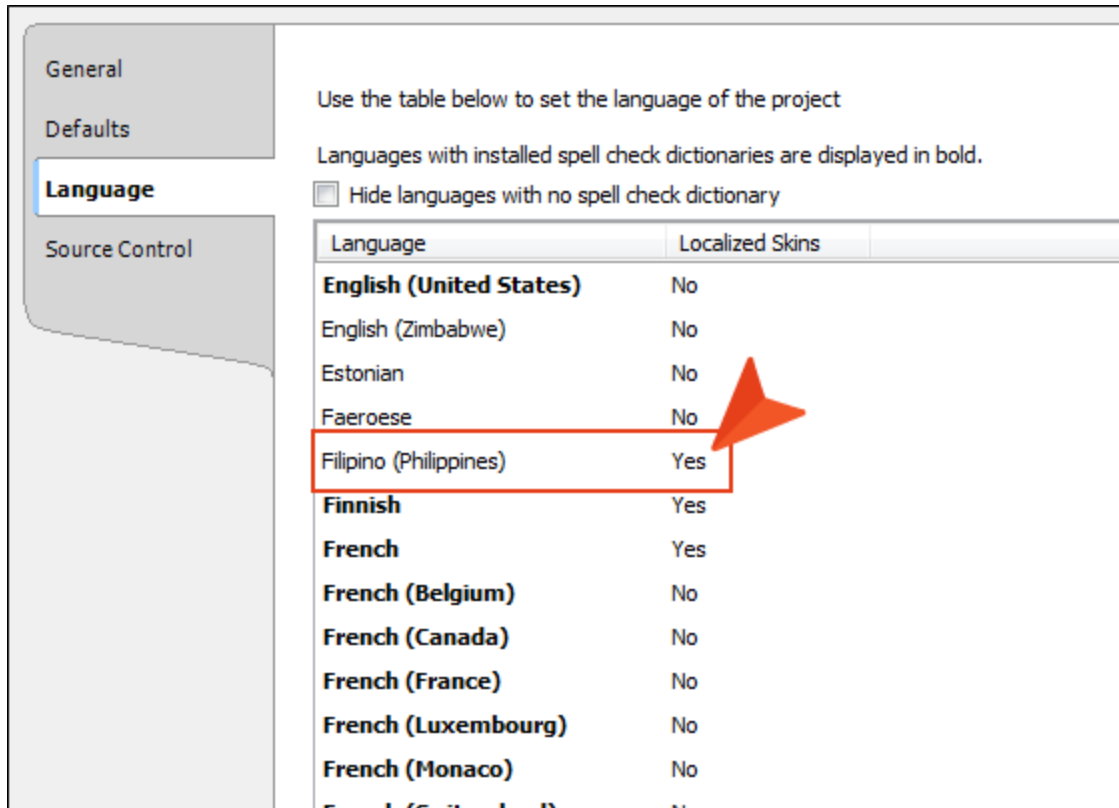
- ☆ Now suppose that you want to use Filipino as the project language. Notice in the example above that Filipino does not have a "Yes" next to it. This means that you need to create a language skin for it from the Add File dialog (**Project > New > Advanced > Add Language Skin**).

In the Add File dialog, you can select a language to use for your language skin.



After you do this, the new language skin opens in the Language Skin Editor, which you can use to edit text labels and make other changes. See "Editing Language Skins" on page 65.

- ☆ After you are finished, you can select the language for the project or a specific target. When you do this, you'll notice that the language now has a "Yes" next to it in the Localized Skins column.



Adding Language Skins

If you want to work with a language skin, you have to add a language skin file to your project.



How to Add a Language Skin File


1. Do one of the following, depending on the part of the user interface you are using:
 - **Ribbon** Select **Project > New > Advanced > Add Language Skin**.
 - **Right-Click** In the Project Organizer, right-click on the **Advanced** folder and from the context menu select **Add Language Skin**.


The Add File dialog opens.

2. In the **File Type** field at the top, make sure **Language Skin** is selected.
3. In the **Source** area select to create a new file from a template (most common) or from an existing file.
4. From the **Language** drop-down, select the language you want to use for the language skin.

 **NOTE** The **Language** drop-down defaults to the current project language.

5. In the **File Name** field, type a new name for the language skin.
6. (Optional) If you want to apply condition tags to the file, expand the **Attributes** section at the bottom of the dialog. Next to the **Condition Tags** field, click  and select the conditions you want to apply. Click **OK**.
7. (Optional) If you want to apply file tags, expand the **Attributes** section at the bottom of the dialog. Next to the **File Tags** field, click  and select the file tags you want to apply. Click **OK**.
8. Click **Add**. The language skin is added to the Advanced folder in the Project Organizer. The Language Skin Editor opens to the right, with the new language skin shown. Depending on the language you selected when creating the language skin, the skin may include default translations for some interface text strings.

 **IMPORTANT** Creating multiple language skins for a single language is not recommended. Flare applies the first language skin it finds for a language—alphabetically—in the Advanced folder of the Project Organizer. As a best practice, you should maintain all of your translations for a language in a single language skin to avoid confusion and incorrect translations.

 **NOTE** If you have created a new language skin for a language, Flare will use it when you build the project. The language skin must reside in the project that uses that language.

What's Next?


You can edit the language skin, adjusting the text on the buttons and labels.

Editing Language Skins

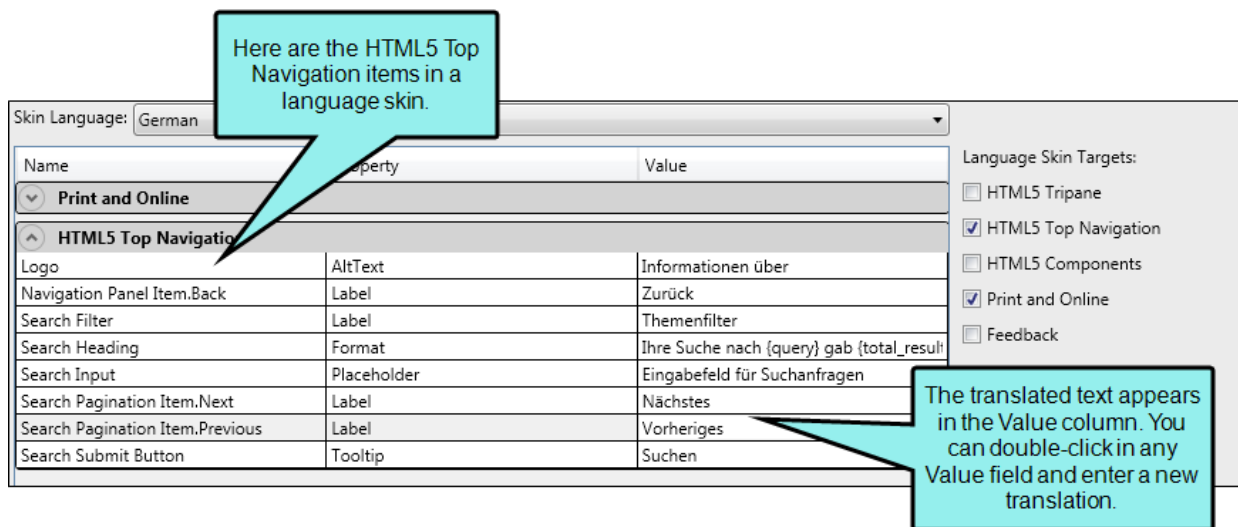
After you add a language skin file, you need to edit it to make sure it provides all the correct text strings.

How to Edit a Language Skin

1. Open the Project Organizer.
2. Double-click the **Advanced** folder.
3. Double-click a language skin to open it.
4. In the Language Skin Targets area, select the check boxes for the target types you want to modify. Translatable values appear in the grid. Values are grouped by the skin type(s) in which they appear, so you do not need to translate the value more than once.

 **NOTE** For more information about the purpose of each of these items in the output, see the online Help.

5. Double-click in any of the **Value** fields and change the text as necessary.



Here are the HTML5 Top Navigation items in a language skin.

Name	Property	Value
▼ Print and Online		
^ HTML5 Top Navigation		
Logo	AltText	Informationen über
Navigation Panel Item.Back	Label	Zurück
Search Filter	Label	Themenfilter
Search Heading	Format	Ihre Suche nach {query} gab {total_result}
Search Input	Placeholder	Eingabefeld für Suchanfragen
Search Pagination Item.Next	Label	Nächstes
Search Pagination Item.Previous	Label	Vorheriges
Search Submit Button	Tooltip	Suchen

Language Skin Targets:

- HTML5 Tripane
- HTML5 Top Navigation
- HTML5 Components
- Print and Online
- Feedback

The translated text appears in the Value column. You can double-click in any Value field and enter a new translation.

6. Click  to save your work.

Translation and Localization Features Per Output Type

Following are translation features supported in each output type.

	HTML5	PDF	Word	Clean XHTML	Eclipse Help	EPUB	HTML Help
Author and Generate Output in Left-to-Right (LTR) and Right-to-Left (RTL) Languages	✓	✓	✓	✓	✓	✓	✓
Invert Image Callouts to RTL	✓	✓	✓	✓	✓	✓	✓
Invert Page Layouts to RTL	✗	✓	✓	✗	✗	✗	✗
Invert Styles to RTL	✓	✓	✓	✓	✓	✓	✓
Language Skins	✓ ¹	✓	✓	✗	✗	✓	✓

	HTML5	PDF	Word	Clean XHTML	Eclipse Help	EPUB	HTML Help
Multilingual Output (Link to External Projects in Target)	✓	✓	✓	✓	✓	✓	✓
Output Interface (Display in LTR and RTL)	✓ ²	✗	✗	✗	✓	✗	✗
PDF Stitching ³	✓	✓	✗	✗	✗	✗	✓
Separate UI Text Tab in Skin Editor for Localization	✓	✗	✗	✗	✗	✗	✗

¹As an alternative to using language skins, you can use the UI Text tab in the Skin Editor to perform all of the localization tasks for HTML5 targets. Be aware that translations in the Skin Editor are prioritized over translations in a language skin, so you should try to perform your localization tasks in one place to avoid conflicts.

²This is controlled on the UI Text tab in a regular HTML5 skin.

³ PDF stitching allows you to add PDFs to Flare output, but you cannot link to content in that PDF. In online outputs, the PDF is simply added to the TOC and can be opened from there; it is not technically "merged" into the output.

Methods for Translation

The approach to employ for your translation project depends on your resources, time, and budget. In your planning, be sure to account for a workflow that best meets your needs, and one that streamlines your efforts.

This chapter discusses the following:

- Method Comparison69
- Method 1: Full-Service Translation70
- Method 2: Translation With Lingo72
- Method 3: Translation With Third-Party Tool74
- Method 4: Translation in Flare Project76
- Method 5: Translation of Output Files86

Method Comparison

Benefits	Method 1: Full-Service	Method 2: Lingo	Method 3: Third-Party Tool	Method 4: Flare Project	Method 5: Output Files
Streamlined Process	✓	✓	✓	✗	✗
Full-Service	✓	✗	✗	✗	✗
ISO Certified	✓	✗	✗	✗	✗
Source Files Translated	✓	✓	✓	✓	✗
Lingo Project	✓	✓	✓	✗	✗
Translation Memory	✓	✓	✓	✗	✗
Reporting	✓	✓	✓	✗	✗
Termbases	✓	✓	✓	✗	✗
Integration With Flare	✓	✓	✓	✓	✗
Cost Effective	✓	✓	✓	✓	✗
Easy Maintenance	✓	✓	✓	✗	✗

I Method 1: Full-Service Translation

The easiest approach to getting material translated into another language is engaging [MadTranslations](#), MadCap's full-service translation and localization division. You send your Flare project to MadTranslations, and at the end of the process you get a translated Flare project back.

Pros

MadTranslations offers complete end-to-end translation and localization services for any language.

This is a great option if you want to be guided through the translation process, and want it all done for you. When the project is done, you receive a fully functional Flare project.

MadTranslations is ISO certified.

Cons

Full-service might sound costly but it proves to be more cost effective considering quality translation services and industry professionalism in return.

Full-Service Workflow

The overview process in using MadTranslations as your translation approach is simple in that your focus remains on the source language project.

1. **Author** Creates source project in source language with Flare.
2. **Author** Sends a completed Flare project (FLPRJZIP) to MadTranslations, and tells MadTranslations what target(s) need to be translated into which languages.
3. **MadTranslations** Provides a quote.
4. **Author** Approves the quote.
5. **MadTranslations** Translates the material and delivers the translated Flare project(s) back to the author. If required by the client, the translator also provides the Lingo project and translation memories (i.e., TMX files) created during the project.

I Method 2: Translation With Lingo

Using MadCap Lingo is a highly streamlined approach for translations. The translator can use Lingo as a computer-assisted translation (CAT) tool and translation management tool.

Pros

MadCap products are designed to work together for maximum efficiency. This includes not only Flare and Lingo, but the entire suite: Central, Capture, Mimic, etc.

When using Flare and Lingo some benefits are: managing your content, extracting source files for translation (automatically), running quality assurance reports, outsourcing to various vendors, and leveraging termbases and translation memory.

Content is easily integrated back into Flare.

Lingo can be used as a computer assisted translation (CAT) tool, as a translation management tool, or both.

Lingo ensures that only the necessary files get translated as it identifies the strings and files that need to be translated per output.

Cons

When you receive translated files back, you might need to make minor adjustments to localized styles or layouts (e.g., text expansion) for publishing. Communicate with the translator about these needs.

Translation With Lingo Workflow

The overview process in using Lingo for translation keeps the workflow aligned for optimal efficiency.

1. **Author** Creates Flare project in source language.
2. **Author** Zips Flare source files, and sends to the translator.
3. **Translator** Creates a new Lingo project using the Flare project provided. The translator selects the source language and the target language. The translator uses Lingo to translate segments.
4. **Translator** Exports the translated Lingo project into a Flare project in the target language and sends it to the author.
5. **Author** From the translated Flare project, the author generates one of the following using the Target Editor: single target, multilingual target, or PDF stitching.

I Method 3: Translation With Third-Party Tool

When Flare source files are packaged with MadCap's Lingo application the files for translation are prepared, managed, and exported into the XLIFF format that translators can understand. With this method, you use Lingo along with Flare to put files into a format that translators can use with any computer-assisted translation (CAT) tool.

Pros

Translators are able to work with files bundled by Lingo, and use in a CAT tool of choice.

Translated XLIFF files are easily integrated back into Lingo, and finally into Flare.

Cons

The files necessary for translation are not bundled automatically, and need to be selected manually for extraction.

When you receive translated files back, you might need to make minor adjustments to localized styles or layouts (e.g., text expansion) for publishing. Communicate with the translator about these needs.


Translation With Third-Party Tool Workflow

There are many CAT tools available for a translator to use for translations (including Lingo).

1. **Author** Creates Flare project in source language.
2. **Author** Creates Lingo project based on Flare project. For details, see the Lingo online Help.
3. **Author** Creates bundle of files in Lingo. This zips the files into the XLIFF file format.
4. **Author** Provides the translator with the bundled files.
5. **Translator** Uses a CAT tool to translate segments.
6. **Translator** When done, the translator zips the translated XLIFF files, and sends them back to the author,
7. **Author** Imports the XLIFF files into the Lingo project.
8. **Author** Exports the Lingo project into a Flare project in the target language.
9. **Author** In the translated Flare project, the author generates one of the following using the Target Editor: single target, multilingual target, or PDF stitching.

Method 4: Translation in Flare Project

Even though Flare is not a computer-assisted translation (CAT) tool, Flare provides the capability for you to type content into Flare as the set primary language of the project. The major caveat to this approach is that you have to be highly proficient in one or more languages, and your operating system must be enabled for the language desired.

 **NOTE** Although you can set the language for your project in Flare, this does not mean that Flare automatically outputs translated content.

Pros

Flare's language attributes allow you to apply tags at different levels (e.g., project, target, topic, content) to organize and identify text better when there might be multiple languages in one project. See "Selecting a Language" on the next page.

Cons


You would have to manage multiple languages in a project.

You must have multilingual skills.

Whether you are fluent in another language or not, it is advised to zip your source files or bundle them using Lingo, for sending to a professional in-country reviewer (ICR) to review and quality check your work.


Selecting a Language

When creating a project, you can assign language attributes at different levels (i.e., project, target, topic, content). The broader the level where the language is set, the lower precedence it has. Meaning, the content level has priority over everything. The topic level outweighs the target and project levels, and the target level has priority over the project level.

 **NOTE** Although you can set the language for your project in Flare, this does not mean that Flare automatically outputs translated content.

Language for a Project


Flare has a default language set when you create a new project, and you can change it to another language if necessary. The language for the project determines the dictionary and skin language for the project.


 **NOTE** You might have to enable another language for your operating system, and switch keyboard language settings.

How to Select a Language for a Project

1. If you have not created the project yet, select a primary language from the Start New Project Wizard. If the project already exists, continue with the next step.
2. Do one of the following, depending on the part of the user interface you are using:
 - **Ribbon** Select **Project > Project Properties**.
 - **Keyboard Shortcut** Press **CTRL+SHIFT+P** on the keyboard.

The Properties dialog opens.

3. Select the **Language** tab.
4. From the list, select a language. (If the language is displayed in bold, Flare sets the dictionary and spelling functions to the specified language.)
5. Click **OK**.
6. Click  to save all files.

 **NOTE** Spell check is based on the language setting for the project, target, topic, or selected content. However, you can also set language-based spell check settings for a style in the Advanced view of the Stylesheet Editor by entering a language code in the **mc-language** property for a style or class.


Language for a Target

At the target level, you can link translated Flare or Lingo projects to your current Flare project. When the target is generated, the output will consist of content in each of those languages. Also, with HTML5 output, you can include a drop-down for users to manually switch between languages.

How to Select Languages for a Target


1. From the Project Organizer, double-click a target.
2. In the Target Editor, select the **Language** tab.
3. From the **Language** drop-down, you can change the language.


4. (Optional) If you want to link to translated projects in other languages, do the following:

- a. Click . Another row is added to the grid.
- b. In the **Linked Project** column, click the link to select the Lingo or Flare project you want to link to your current project.

If you have not added a linked project yet, the link looks like an ellipsis; if you have added a linked project, the link displays the project's file name.


Language	Localized Skins	Linked Project
(default) [English (United States)]	Yes	This Project
Slovak	No	...






 **NOTE** You can link to many separate Lingo projects, or you can link multiple rows in this tab to the same multilingual Lingo project, or both.

- c. In the dialog, find and select the project you want to link to your current project. Click **Open**.
- d. If necessary, select the linked project's language from the **Language** drop-down. Flare automatically detects the linked project's default language settings, so you only need to update this setting if you are pointing to a multilingual Lingo project, or if it is incorrect in the linked project.


Additionally, if default translations are available for the selected language's skin, it is noted in the **Localized Skins** column. If you have a custom language skin for the selected language in your linked project, Flare automatically uses your custom translations. See "Adding Language Skins" on page 63.

 **NOTE** If you link to a multilingual Lingo project, only the available languages from a multilingual project can be selected from the drop-down.

 **NOTE** The default target output language in the Target Editor always shows the currently selected project language. This language updates automatically if you change the project language.

- e. (Optional) Continue adding rows and linked projects for each additional target output language.
 - f. (Optional) Use the  and  buttons to rearrange your projects. They display in the output in the order they are listed on the Language tab (e.g., if you include a drop-down for users to choose each language in HTML5 output, or if you are working in a PDF target where one language will follow another in the output).
5. (Optional) If you have added links to one or more Lingo projects, you can select **Sync Lingo projects** if you want to synchronize updates with those projects.

If this option is enabled, the application detects whether any of the Lingo and Flare source files are out of sync. If they are, the Lingo project is automatically updated and these changes are also brought into the parent Flare project. This is different from the usual process, where the translator would normally update the Lingo project manually and translate the changed or new files.

 **WARNING** You might use this feature if you want to quickly see any updated files in your parent Flare project, including non-translated content such as images. However, enabling this option is typically not recommended, because there is always the risk of updating the Lingo project (and therefore also the parent Flare project) with content that has not yet been translated.

6. (Optional) If you selected a right-to-left (RTL) language, you will see the following options at the bottom, which are enabled by default for RTL languages. Leave them selected to automatically invert the following: (1) language-related style rules locally, (2) language-related style rules in the stylesheet, (3) image callouts from MadCap Capture, and (4) page layout settings.
- Invert left/right local styles
 - Invert left/right stylesheet rules
 - Invert Capture captions
 - Invert Page Layouts

The options that are seen depend on the output type you are using.

- **PDF/Word** All four options display.
- **HTML5/HTML Help/EPUB/Eclipse Help/WebHelp/WebHelp Plus** Local styles, CSS styles, and image callout options display.
- **DITA** No options display.

These options are useful after you receive the project back from a translator. When you generate the output, the inversion of the styles and page layouts takes place.

☆ **EXAMPLE** You have a project in English, a left-to-right (LTR) language, and you need to have it translated into Arabic, a RTL language.

For your paragraph (p) style, suppose you have a left margin of 5 px and a right margin of 30 px.

Those settings work fine for targets using the LTR language. But for the RTL language, you need the settings to be flipped so that the left margin is 30 px and the right margin is 5 px.

So after you receive the project back from the translator, make sure the RTL language in Flare is set at either the project or target level.


As a result, the inversion options are automatically selected. When output is generated the paragraphs have a left margin of 30 px and a right margin of 5 px.


📄 **NOTE** The option to invert page layouts controls every aspect of the page layout file (e.g., it inverts not only frames and content within them, but also styles applied to frame content). The other invert options have no effect on page layouts.


📄 **NOTE** If your selected language is LTR, these options cannot be accessed in the Target Editor.


7. Click  to save your work.


What's Noteworthy?


 **NOTE** You must link each language to a Flare or Lingo project before you can close the Target Editor. If you make changes to your linked projects and their links are no longer valid when you build the project, a warning message displays, and it will be unable to build.


 **NOTE** When you build a target that is set up for multilingual output, and you link directly to a Lingo project, the export process runs automatically in Lingo so that the parent Flare project grabs the necessary translated Flare projects. If the Lingo export process encounters warnings, these will not display with the other build warnings in Flare's interface. Instead, you must open the build log to find any such warnings.


 **NOTE** In order to build multilingual output that links directly to multilingual Lingo projects, you must have Lingo 10 or newer installed on the same computer.

 **NOTE** Since Flare generates output from the linked targets in each of your project files, each linked project must have its own target file (e.g., if you are creating PDF output, each linked project needs its own PDF target). If you do not have a target file, a warning message displays before the build starts and you will be unable to build. Target files must be in the same relative location in each project.

 **NOTE** Be sure that file names are the same for your parent project and for each of your linked projects. This is important for HTML5 targets, so you can switch between languages using the Select Language button in the output.

 **NOTE** Multilingual Elasticsearch publishing requires all targets to use the same GUID. This is set on the Search tab of the Target Editor. If the GUIDs in the targets are different, you might have empty targets on the portal or experience a publishing failing with error 400. If necessary, open targets in the Internal Text Editor to modify the GUID.


 **NOTE** When generating localized HTML Help targets, it is sometimes necessary to set the Windows system locale to match the language that the project is set to. It is necessary to do this when the project contains topic file names with non-English characters. To do this in Windows: 1. Open the **Control Panel**; 2. select **Regional and Language Settings**; 3. select the **Advanced** tab; 4. from the drop-down in the section called **Language for non-Unicode programs**, choose the same language that the Flare project is set to; 5. restart Windows.


 **NOTE** If you are building Eclipse Help, you will need to open your Output folder to open your desired language output.

Language for a Topic

Selecting language attributes at the topic level is useful for organizing or identifying content for a certain language, if you are managing multiple languages in one project. When you apply a language attribute to a topic, the topic is flagged as containing content for the specified language.

How to Select a Language for a Topic


1. From the Content Explorer, right-click a topic, and select **Properties**.
2. From the Properties dialog, select a language. (If the language is displayed in bold, Flare sets the dictionary and spelling functions to the specified language.)
3. Click **OK**.
4. Click  to save your work.

 **NOTE** Using Flare's analysis tool, you can generate a language report. This report shows each file where a language tag is found, the content tag to which it has been applied (e.g., html, span), and the language used.

Language for a Content Segment

Selecting language attributes at the content level can be used for organizing or identifying text for a certain language, but only if you are managing multiple languages in one project. When you apply a language attribute to a segment, it is flagged as containing content for the specified language.

How to Select a Language for Content

1. Open the topic.
2. Highlight the text where you want to change the language.
3. Select **Home > Language**.
4. From the Select Language dialog, choose a language. (If the language is displayed in bold, Flare sets the dictionary and spelling functions to the specified language.)
5. Click **OK**.
6. Click  to save your work.

I Method 5: Translation of Output Files

You might be considering just having the output files translated, which might eliminate the worry of having to filter out a variety of project files. Although this approach is possible, it is not practical in the long run, and it does not benefit from single-sourcing.

It is not recommended to translate output files for the following reasons:

- Translation is based on output files (online or PDF). You are responsible for extracting all the translatable files manually.
- Sending one output file might seem like a quick way to get a project translated, but you run the risk of higher translation costs later.
- You cannot create other output types using translated files in the project (without the assistance from a translator).
- For HTML5 targets, some search files get compiled during the build. If translating the output files, any JavaScript files in the data folder would need to be parsed, and translated without context of where each word is located in the content.
- After the translation of output files, you are left with many different files for each output. This can be a maintenance challenge for updating and publishing.

✔ **TIP** It is recommended to always have the source files translated. By translating the source project (and not the output files), you increase the integrity of the project, you avoid potential workflow and maintenance challenges, and you reduce your overall translation costs.

Other Information for Translation

In addition to the primary concepts, there is some additional information that might help you when working with this feature.

This chapter discusses the following:

- Exporting Projects for Translation 88
- Multilingual Support for Salesforce90
- Multilingual Support for ServiceNow®92
- Multilingual Support for MadCap Connect to Zendesk95
- Selecting a Language for the Interface 98

Exporting Projects for Translation

When you need to send a Flare project to someone for translation, you can do so by exporting the entire project, or just the parts of it that you need translated. For more information see the Flare online Help.

☆ **EXAMPLE** You have a Flare project with seven targets, and you need to translate the content associated one of those targets from English to French. You could send the entire Flare project to the translator, but that would mean the translator would be getting files associated with all seven targets, not just the one requiring translation. So you decide to export only the portion of the Flare project that needs to be translated.

First, from the **Project** ribbon you click **Export Project**. On the first page of the wizard, you click in the **Export From** drop-down and select **Using Target**.

Export Project
Export Project Settings

Source Project:
C:\Users\pstoedlein\Documents\My Projects_Export Project\ExportProject3\ExportPro

New Project Name:
ExportProject3

New Project Path:
C:\Users\pstoedlein\Documents\My Proj...s

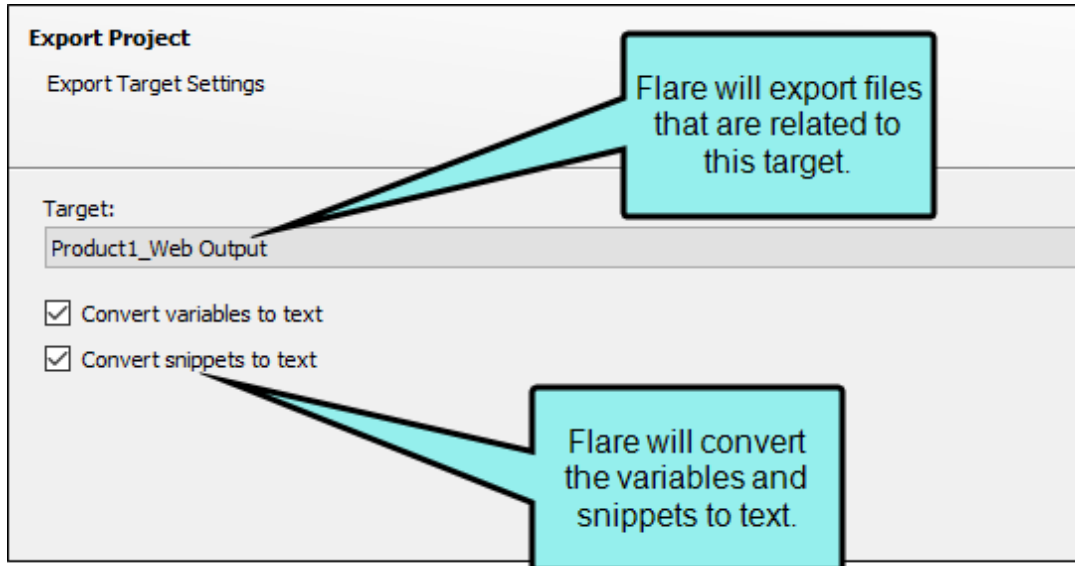
Export From:
Using Target

Output:
Project Files

Save settings

This option lets you export to a new Flare project based on a particular target in the source project.

- ☆ On the next page of the wizard, you select the target whose files you want to export. In this case, let's say the target in question is named "Product1_Web Output." In addition, you tell Flare to convert variables and snippets to text so that they become part of the topics, rather than separate files.



After clicking **Finish**, the relevant files and content are exported to a new, smaller Flare project, which you send to the translator. Only the files and content necessary to produce the Product1_Web Output target are included in the export. Therefore, the translator receives only the files requiring translation.

I Multilingual Support for Salesforce

You can publish directly in different languages to Salesforce®. The list of available languages depends on the organization, but below you can see all the languages that Salesforce® currently supports and their Flare equivalents.

If you publish a project in an unsupported language, the articles will default to the organization's default language.

Supported Languages

Salesforce® Language	Flare Language
Chinese (Simplified)	Chinese (China)
Chinese (Traditional)	Chinese (Taiwan)
Danish	Danish i.e., Danish (Denmark) is incorrect
English	English (United States) i.e, English or any other similar is incorrect
Finnish	Finnish
French	French i.e., French (Canada) or any other similar is incorrect
German	German i.e., German (Germany) or any other similar is incorrect

Salesforce® Language	Flare Language
Italian	Italian i.e., Italian (Italy) or any other similar is incorrect
Japanese	Japanese i.e., Japanese (Japan) is incorrect
Korean	Korean i.e., Korean (Korea) is incorrect
Portuguese (Brazil)	Portuguese (Brazil) i.e., Portuguese or any other similar is incorrect
Russian	Russian i.e., Russian (Russia) or any other similar is incorrect
Spanish	Spanish i.e., Spanish (Spain) or any other similar is incorrect
Spanish (Mexico)	Spanish (Mexico) i.e., Spanish (Spain) or any other similar is incorrect
Swedish	Swedish i.e., Swedish (Sweden) or any other similar is incorrect
Thai	Thai

I Multilingual Support for ServiceNow®

You can publish directly in different languages to ServiceNow®.

To do this, make sure the following have been done before publishing Flare output to ServiceNow®:

- Work with a ServiceNow® administrator to make sure you have the following permissions:
 - **v_plugin** Read, write access is required to determine the plugin install status for a localized upload.
 - **sys_translated_text** Create, read, write access is required for the record- and field-level properties for creating category translation strings.
- On ServiceNow®, make sure the plugin for each language is installed.
- You might need to switch user interface language in ServiceNow®.
- In Flare, you must have a target that is configured for multilingual output.

Supported Languages

ServiceNow® Language	Flare Language
Chinese (Simplified)	Chinese (China)
Chinese (Traditional)	Chinese (Taiwan)
Czech	Czech
Dutch	Dutch (Belgium), Dutch (Netherlands)
English	English, English (United States), English (Australia), English (Belize), English (Canada), English (Ireland), English (Jamaica), English (New Zealand), English (Philippines), English (Trinidad), English (South Africa), English (United Kingdom), English (Zimbabwe)

ServiceNow® Language	Flare Language
Finnish	Finnish
French (Canada)	French (Canada)
French	French, French (Belgium), French (France), French (Luxembourg), French (Monaco), French (Switzerland)
German	German, German (Pre-reform), German (Austria), German (Germany), German (Lichtenstein), German (Luxembourg), German (Switzerland)
Hebrew	Hebrew, Hebrew (Israel)
Hungarian	Hungarian
Italian	Italian, Italian (Italy), Italian (Switzerland)
Japanese	Japanese, Japanese (Japan)
Korean	Korean, Korean (Korea)
Norwegian	Norwegian, Norwegian (Bokmål), Norwegian (Norway), Norwegian (Nynorsk)
Portuguese (Brazilian)	Portuguese (Brazil)
Portuguese	Portuguese, Portuguese (Portugal)
Russian	Russian, Russian (Moldova), Russian (Russia)

ServiceNow® Language**Flare Language**

Spanish

Spanish, Spanish (Argentina), Spanish (Bolivia), Spanish (Colombia), Spanish (Costa Rica), Spanish (Dominican Republic), Spanish (Ecuador), Spanish (El Salvador), Spanish (Guatemala), Spanish (Honduras), Spanish (Mexico), Spanish (Nicaragua), Spanish (Panama), Spanish (Paraguay), Spanish (Peru), Spanish (Puerto Rico), Spanish (Spain), Spanish (Uruguay), Spanish (Venezuela)

Swedish

Swedish, Swedish (Finland), Swedish (Sweden)

Thai

Thai

Turkish

Turkish

I Multilingual Support for MadCap Connect to Zendesk

Zendesk offers multilingual support for articles published to their Help Center dashboard. If you publish your project in an unsupported language, the articles will be published in the default language set in your Zendesk Help Center dashboard.

You can publish your output to Zendesk in the following languages supported by Flare.

Supported Languages

Languages Supported in Zendesk

Arabic	English (New Zealand)	Greek	Portuguese
Arabic (Egypt)	English (Phillipines)	Hebrew	Portuguese (Brazil)
Bulgarian	English (South Africa)	Hindi	Romanian
Catalan	English (United Kingdom)	Hungarian	Russian
Chinese (China)	Estonian	Icelandic	Serbian
Chinese (Hong Kong)	Filipino	Indonesian	Slovak
Chinese (Taiwan)	Finnish	Italian	Slovenian
Croatian	French	Japanese	Spanish

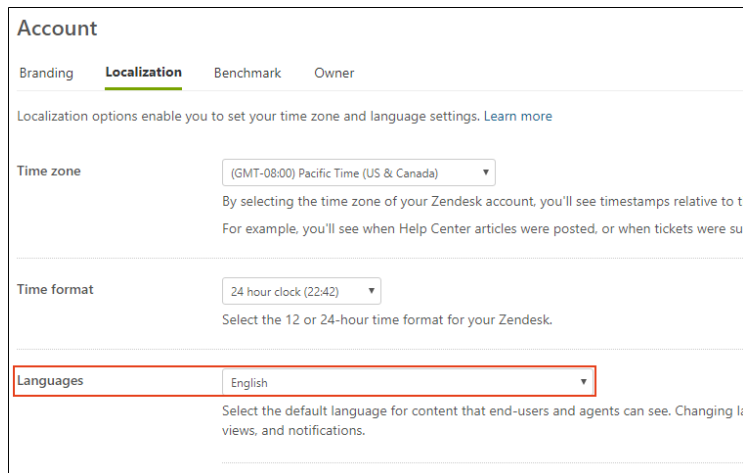
Languages Supported in Zendesk

Czech	French (Belgium)	Korean	Spanish (Mexico)
Danish	French (Canada)	Latvian	Spanish (Spain)
Dutch	French (France)	Lithuanian	Swedish
English	French (Switzerland)	Malay	Thai
English (Australia)	German	Norwegian	Turkish
English (Canada)	German (Austria)	Persian	Ukrainian
English (Ireland)	German (Switzerland)	Polish	Vietnamese

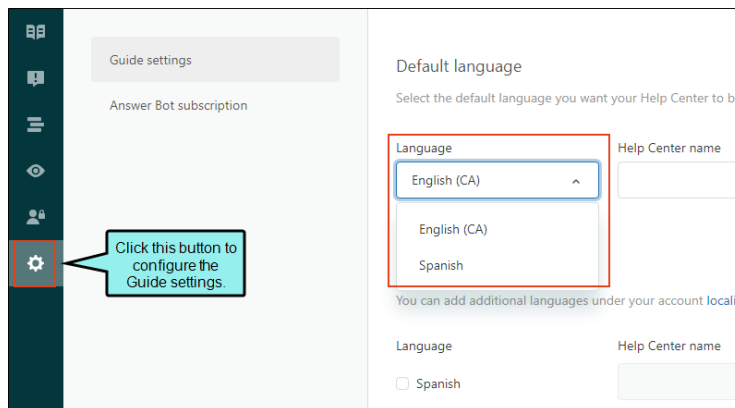
How to Set the Language for Zendesk Output

1. Open your target.
2. Click the **Language** tab.
3. From the **Language** field, select the language you wish to use in your output.
4. Click **Publish**.
5. After you publish your output to Zendesk, go to your Zendesk Help Center dashboard.
6. Go to your account settings in Zendesk.
7. In the Zendesk account settings, click the **Localization** tab.


- In the **Languages** section, click the drop-down arrow to select another language.



- Click **Save tab**.
- In Zendesk Help Center, click **Guide settings**.
- Click the drop-down arrow in the **Language** section to apply the language to be used for your Zendesk dashboard.



- In the upper-right corner, click **Update**.

 **NOTE** For each language used to publish your project, you need to set up the default categories and sections for each language in Zendesk.

I Selecting a Language for the Interface

You can select one of the following languages for displaying the Flare interface: English, French, German, Japanese, or Chinese.

How to Select a Language for the Interface

1. Launch Flare. The Select UI Language dialog opens.

If you have previously selected to disable this dialog, you can enable it:

- a. Select **File > Options**. The Options dialog opens.
 - b. Select the **General** tab.
 - c. Select **Show Select UI Language Dialog on Startup**.
 - d. Click **OK**.
2. Select the language.
 3. (Optional) If you do not want the dialog to display at launch, click **Show this dialog on startup** to remove the check mark.
 4. Click **OK**. The interface displays according to the language selection.

APPENDIX

PDFs

The following PDFs are available for download from the online Help.

I Tutorials

Getting Started Tutorial

Autonumbers Tutorial

Back-to-Top Button Tutorial

Context-Sensitive Help Tutorial

Custom Toolbar Tutorial

eLearning Tutorial—Basic

eLearning Tutorial—Advanced

Image Tooltips Tutorial

Lists Tutorial

Meta Tags Tutorial

Micro Content Tutorial—Basic

Micro Content Tutorial—Advanced

Responsive Output Tutorial

Single-Sourcing Tutorial

Snippet Conditions Tutorial

Styles Tutorials

Tables Tutorial

Word Import Tutorial

| Cheat Sheets

Context-Sensitive Help Cheat Sheet

Folders and Files Cheat Sheet

Learning & Development Cheat Sheet

Lists Cheat Sheet

Micro Content Cheat Sheet

Print-Based Output Cheat Sheet

Search Cheat Sheet

Shortcuts Cheat Sheet

Structure Bars Cheat Sheet

Styles Cheat Sheet

I User Guides

Accessibility Guide

Analysis and Reports Guide

Architecture Guide

Autonumbers Guide

Branding Guide

Condition Tags Guide

Context-Sensitive Help Guide

Eclipse Help Guide

eLearning Guide

Getting Started Guide

Global Project Linking Guide

HTML5 Guide

Images Guide

Import Guide

Indexing Guide

Key Features Guide

Lists Guide

*MadCap Central Integration
Guide*

Meta Tags Guide

Micro Content Guide

Navigation Links Guide

Plug-In API Guide

Print-Based Output Guide

Project Creation Guide

QR Codes Guide

*Reviews & Contributions With
Contributor Guide*

Scripting Guide

Search Guide

SharePoint Guide

Skins Guide

Snippets Guide

Source Control Guide: Git

*Source Control Guide:
Perforce Helix Core*

*Source Control Guide:
Subversion*

*Source Control Guide: Team
Foundation Server*

Styles Guide

Tables Guide

Tables of Contents Guide

Targets Guide

Template Pages Guide

Templates Guide

Topics Guide

Touring the Workspace Guide

*Transition From FrameMaker
Guide*

*Translation and Localization
Guide*

Variables Guide

Videos Guide

What's New Guide