MADCAP FLARE 2019 r2

Language Support
Copyright 2019 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
9191 Towne Center Drive, Suite 150
San Diego, California 92122
858-320-0387
www.madcapsoftware.com

THIS PDF WAS CREATED USING MADCAP FLARE.
CONTENTS

CHAPTER 1
Introduction ................................................................. 5
Selecting a Language ...................................................... 7
Authoring Content .......................................................... 7
Language Skins ............................................................ 8
Multilingual Output ......................................................... 10
Invert Styles, Page Layouts, and Image Callouts .................... 30
Invert Hotspot Images ..................................................... 32
Table Styles and RTL Languages ........................................ 34
Dictionaries .................................................................. 35
Translating Flare Projects ................................................ 36
Multilingual Support for Salesforce® Integration ....................... 37
Multilingual Support for MadCap Connect to Zendesk ............ 39
Exporting Projects for Translation ....................................... 42
Stitching PDFs ............................................................... 44

CHAPTER 2
Selecting a Language ........................................................ 48
Selecting a Language for Your Content ............................... 49
Selecting a Language for the Interface ................................. 53
<table>
<thead>
<tr>
<th>Chapter</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>Language Skins</td>
<td>54</td>
</tr>
<tr>
<td></td>
<td>Completed Versus User-defined Language Skins</td>
<td>55</td>
</tr>
<tr>
<td></td>
<td>Creating User-Defined Language Skins</td>
<td>60</td>
</tr>
<tr>
<td></td>
<td>Editing Language Skins</td>
<td>63</td>
</tr>
<tr>
<td></td>
<td>Changing Text Strings in HTML5 Skins</td>
<td>65</td>
</tr>
<tr>
<td>4</td>
<td>Importing Dictionaries</td>
<td>66</td>
</tr>
<tr>
<td>Appendix</td>
<td>PDFs</td>
<td>68</td>
</tr>
</tbody>
</table>
CHAPTER 1

Introduction

Flare supports authoring and output for left-to-right (LTR) as well as right-to-left (RTL) languages. This includes English, French, German, Japanese, Chinese, Arabic, Persian, Hebrew, multi-byte languages, and more. Flare supports bi-directional languages not only at the topic level, but all the way down to the paragraph and sentence level.

There are multiple ways to produce output in different languages. For example, you might translate content directly in a Flare project, perhaps creating a different topic for each language. Alternatively, you can send the Flare files to a translator, who can use a CAT (computer-assisted translation) tool in order to translate the text strings. MadCap Lingo can be used as a CAT tool or for project management, packaging your Flare files to be sent to a translator, who uses a third-party CAT tool for the translation.

This chapter discusses the following:

- Selecting a Language ............................................................... 7
- Authoring Content ................................................................. 7
- Language Skins ............................................................... 8
- Multilingual Output .................................................................10
- Invert Styles, Page Layouts, and Image Callouts ...................30
- Invert Hotspot Images ..............................................................32
- Table Styles and RTL Languages ........................................34
- Dictionaries ............................................................... 35
- Translating Flare Projects .........................................................36
- Multilingual Support for Salesforce® Integration ..................37
- Multilingual Support for MadCap Connect to Zendesk ...........39
Selecting a Language

Broadly speaking, selecting a language can mean a couple of different things. First, it can mean specifying a language for the content that you are producing. However, it can also mean choosing a language for the Flare interface as you do your work. See “Selecting a Language” on page 48.

Authoring Content

Full authoring of LTR and RTL text is supported in Flare’s XML Editor.

The XML Editor supports the following related to RTL:

- RTL text is readable and can be edited in the XML Editor.
- Flare supports CSS and HTML direction properties (e.g., "direction: rtl" and dir="rtl"). If these properties are set to "right," the following occur.
  - Text is right-aligned.
  - Tables flow from the right to left (i.e., cell 1 is at the top right, cell 2 is to the left of it).
  - List item bullets and numbers are to the right of their content instead of left.
- If an RTL language has been selected for the project or topic, the default direction for content is right-to-left. On the other hand, if an LTR language is selected, the default direction is left-to-right.
Language Skins

A language skin can be used to display the interface in a specific language for online outputs. Language skins also control some text-based items, like cross-reference links. Unlike regular skins, language skins hold only text used for the output; they do not control the look of the output. See "Language Skins" on page 54.

The language set at the project or target level affects the skin used for the output if you are generating WebHelp outputs. Flare provides completed language skins for certain languages, such as French, German, Spanish, and more. You can customize these language skins with your own translations, as well. For other languages, you can create language skins. See "Creating User-Defined Language Skins" on page 60 and "Editing Language Skins" on page 63.

EXAMPLE

Let’s say you select French as the project language. If you generate HTML5 Tripane, the French skin is automatically used, so the output looks something like this.
**NOTE** Typically, language skins are designed for online outputs only (HTML5, WebHelp, and WebHelp Plus).

However, if you are editing text strings in a language skin for toolbar items, those strings will be translated in Microsoft HTML Help output as well.

In addition, the "Formats/Cross-Reference" items in the Language Skin Editor are used for both print-based and online outputs. Those items are intended to set text for context-sensitive cross-references.

Another print-based feature controlled in the language skin is the heading text for auto-generated glossaries, indexes, and proxies.

Language skins also control features found in MadCap Feedback, such as comment labels and profile items.
Multilingual Output

If you have translated your project into other languages, you can link Lingo projects or additional translated Flare projects to your master project to create multilingual output. When you build the target, Flare will included translated content from each linked project.

HTML5 and WebHelp

Flare will create a subfolder in your Output folder for each language. When you view the output, Flare will display the output for your browser’s default language. In HTML5 outputs, you can use a drop-down menu to switch between each generated language.

EXAMPLE

HTML5 output will look like this.
WebHelp output will look like this.

NOTE Be sure that the file names are the same in your master project and each of your linked projects. This is especially important for HTML5 targets, so you can switch between languages using the Select Language button in the output.
**NOTE** If you are using an HTML5 output, be sure to enable the Select Language button in the skin for your master project and each linked project. This will allow you to change languages using a drop-down.

1. If you are creating Top Navigation output, you should have a Topic Toolbar skin component in your project. Open the skin component, and select the **Setup** tab.
   
   If instead you are creating Tripane output, open the regular skin and select the **Toolbar** tab.

2. If necessary, move the **Select Language** button from the **Available** section to the **Selected** section.

After you add the Select Language button to your Topic Toolbar, you also need to be sure the toolbar appears in your master page. If the topic toolbar does not appear in each of your master pages, you will not be able to switch languages from those page types.
PDF

The output is merged (or “stitched”) into a single document, with each language appearing in the order you specified. You will see a single file in your Output folder.

EXAMPLE

PDF output will look like this.

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. Additionally, because you do not need to generate each PDF separately every time the document is updated, generating output from a multilingual PDF target saves considerable time.
NOTE You can access individual PDFs for each language in the Temporary folder in your Output folder.
Other Output Types

Flare will create subfolders for each language in your Output folder. You can merge printed output types manually (e.g., in Word) if you need to create a single file.

EXAMPLE

Word output will look like this.

![EXAMPLE Image]

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

Following are two examples of multilingual output.

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

In the second 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 — Linking to Flare Projects

Let’s say 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, make sure each Flare project has both an HTML5 and a PDF target available. This allows your master project to build from the targets in the other two linked projects. 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.
Because you want to build multilingual HTML5 output, you need to make sure that each of your projects has the Select Language button in its HTML5 toolbar, and that you've added the Topic Toolbar proxy to a master page. If the button isn't there, users won't be able to switch from one language to another in your output.

Open the Project Organizer. In the Skins folder, open the Topic Toolbar skin component. On the Setup tab, select the Select Language button, click to add it to your skin, and then save your work.
Now you can prepare your multilingual target. Open your master project (in this case, the US English project), and open the Targets folder. Open the HTML5 target. In the Target Editor, select the Language tab. This is where you will link your other two projects. On the right side of the tab, click to add a new row.

Click the link in the Linked Flare Project column.
In the dialog that opens, navigate to your 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.

Add your Japanese project in the same way.
You want the Japanese project to appear second in the Help system, so you select the Japanese project and click the blue up arrow to rearrange the projects.

Now the Japanese project is listed before the Spanish project.

Now the projects are in the order you want to use.
You are ready to build, so 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.

The output will then display in Japanese.

Next you create the PDF target. Open the PDF target in the Target Editor and create it in the same way you did the HTML5 output. Since you want the Japanese output to appear first, you add that project first, then you add the Spanish project.
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.

Flare stitches all three languages 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.

The navigation pane separates each language, making it easy to find topics in any section of the multilingual PDF.
EXAMPLE — Linking to Lingo Projects

Let’s say want your 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 second 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 your target in Flare, and on the Language tab, you 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.
For the first row, you select **French**. Then you add a second row. This time, you can only select German or Spanish (because French has already been used).

We've added a second row, linking to the same Lingo project.

The first row has French selected.

From the second row, we can select either German or Spanish.
For the second row, you choose **German**. Then you add a third row. **Spanish** is automatically chosen for that row, because it is the only one left.
Finally, you create a fourth row and link it to the Lingo project used for the Arabic translation.

**NOTE** You can use a tool like MadCap Lingo (or another computer-aided translation tool) to translate your Flare content. Although you can set the language for your project in Flare, this does not mean that Flare automatically outputs translated content.

**NOTE** You must link each language to a Flare or Lingo project before you can close the Target Editor. Similarly, if you make changes to your linked projects and their links are no longer valid when you build the project, you will see a warning message before the build starts and you will be unable to build.
**NOTE** In order to build output that links directly to multilingual Lingo projects, the Flare user must have at least Lingo 10 installed on the same computer.

**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 master Flare project can grab 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** Because Flare generates output from the linked targets in each of your project files, each linked project must have its own target file for the output you want to build (e.g., if you are creating PDF output, each linked project needs its own PDF target). If you do not have a needed target file in one of your linked projects, you will see a warning message before the build starts and you will be unable to build. Target files must be in the same relative location in each project.

**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.

**NOTE** If you are using right-to-left language settings in a linked project, you must enable these settings in your master project target. Language settings in the master project target control those for each of the linked projects, regardless of the settings in your linked projects.
Invert 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 an 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. See “Selecting a Language” on page 48.

In this example, Arabic has been selected as the language. Therefore, these options are automatically set to invert all relevant style property values, page layouts, and image callouts to RTL.
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**

Let’s say you have a project in English (a left-to-right 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.
Invert 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 LTR outputs. However, if you are using an RTL language, Flare automatically inverts the default images so that they make sense for that direction as well.

☆ EXAMPLE

Let's say 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. For more information see the Flare online Help.

- Drop-Down Text
- Expanding Text
- Togglers
- Concept Links
- Keyword Links
- Related Topics
- Shortcuts

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. If you want to have inverted versions of an image, you need to use two different skins—one for the LTR images and a second for the RTL images.
Table Styles and RTL Languages

If you are using an RTL language, table styles in your project need to write additional 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 folder.
- If you generate output and an RTL table references an old table stylesheet, the table style in the output will be updated.
Dictionaries

Several languages come with built-in spell checking support in Flare. You can see which dictionaries are installed on your computer by opening the Options dialog (File > Options) and selecting the Spelling tab.

You can download more dictionaries from openoffice.org. After you download a new dictionary, you can install it by clicking Import Dictionaries at the bottom of the Options dialog (File > Options) in Flare. See "Importing Dictionaries" on page 66.
Translating Flare Projects

After completing a project in one language, you might need to have it translated into another language. Following are the two most common ways to do this.

- **MadCap Lingo** One of the easiest ways to translate a Flare project is for a translator to open that project within MadCap Lingo, which is tightly integrated with Flare. Because of this integration, there is no need to transfer localized files outside of the actual project, which helps prevent content and formatting corruption. In addition, translators can leverage all previous translations created in other tools by importing Translation Memory eXchange (TMX) files.

  After opening your project in Lingo, a translator can immediately see a list of all of the files (e.g., topics, snippets, variables), index markers, and concept markers than need to be localized. Then, after translating the content in the Lingo interface, the translator can export the results to a new Flare project in that language. For more information, please refer to the documentation provided with MadCap Lingo.

- **Third-Party Translation Tools** Each file in a Flare project is XML-based and accessible in third-party tools. You can even use MadCap Lingo to package your project files and send them to a translator who is using another tool. That way, the translator receives all of the files requiring translation. It makes things easier for translators, who receive a complete list of files that they can import into the third-party tool; no guessing is required. It also makes things easier for you as the author, because you do not need to worry about forgetting to send often-missed Flare project files that require translation and can easily be overlooked. For more information, please refer to the documentation provided with MadCap Lingo.
Multilingual Support for Salesforce® Integration

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.

LANGUAGE EQUIVALENCY

<table>
<thead>
<tr>
<th>Salesforce® Language</th>
<th>Flare Language</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chinese (Simplified)</td>
<td>Chinese (China)</td>
</tr>
<tr>
<td>Chinese (Traditional)</td>
<td>Chinese (Taiwan)</td>
</tr>
<tr>
<td>Danish</td>
<td>Danish</td>
</tr>
<tr>
<td></td>
<td>i.e., Danish (Denmark) is incorrect</td>
</tr>
<tr>
<td>English</td>
<td>English (United States)</td>
</tr>
<tr>
<td></td>
<td>i.e., English or any other similar is incorrect</td>
</tr>
<tr>
<td>Finnish</td>
<td>Finnish</td>
</tr>
<tr>
<td>French</td>
<td>French</td>
</tr>
<tr>
<td></td>
<td>i.e., French (Canada) or any other similar is incorrect</td>
</tr>
<tr>
<td>German</td>
<td>German</td>
</tr>
<tr>
<td></td>
<td>i.e., German (Germany) or any other similar is incorrect</td>
</tr>
<tr>
<td>Italian</td>
<td>Italian</td>
</tr>
<tr>
<td></td>
<td>i.e., Italian (Italy) or any other similar is incorrect</td>
</tr>
<tr>
<td>Japanese</td>
<td>Japanese</td>
</tr>
<tr>
<td></td>
<td>i.e., Japanese (Japan) is incorrect</td>
</tr>
<tr>
<td>Salesforce® Language</td>
<td>Flare Language</td>
</tr>
<tr>
<td>------------------------------</td>
<td>--------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Korean</td>
<td>Korean</td>
</tr>
<tr>
<td></td>
<td>i.e., Korean (Korea) is incorrect</td>
</tr>
<tr>
<td>Portuguese (Brazil)</td>
<td>Portuguese (Brazil)</td>
</tr>
<tr>
<td></td>
<td>i.e., Portuguese or any other similar is incorrect</td>
</tr>
<tr>
<td>Russian</td>
<td>Russian</td>
</tr>
<tr>
<td></td>
<td>i.e., Russian (Russia) or any other similar is incorrect</td>
</tr>
<tr>
<td>Spanish</td>
<td>Spanish</td>
</tr>
<tr>
<td></td>
<td>i.e., Spanish (Spain) or any other similar is incorrect</td>
</tr>
<tr>
<td>Spanish (Mexico)</td>
<td>Spanish (Mexico)</td>
</tr>
<tr>
<td></td>
<td>i.e., Spanish (Spain) or any other similar is incorrect</td>
</tr>
<tr>
<td>Swedish</td>
<td>Swedish</td>
</tr>
<tr>
<td></td>
<td>i.e., Swedish (Sweden) or any other similar is incorrect</td>
</tr>
<tr>
<td>Thai</td>
<td>Thai</td>
</tr>
</tbody>
</table>
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**

<table>
<thead>
<tr>
<th>Arabic</th>
<th>English (New Zealand)</th>
<th>Greek</th>
<th>Portuguese</th>
</tr>
</thead>
<tbody>
<tr>
<td>Arabic (Egypt)</td>
<td>English (Phillipines)</td>
<td>Hebrew</td>
<td>Portuguese (Brazil)</td>
</tr>
<tr>
<td>Bulgarian</td>
<td>English (South Africa)</td>
<td>Hindi</td>
<td>Romanian</td>
</tr>
<tr>
<td>Catalan</td>
<td>English (United Kingdom)</td>
<td>Hungarian</td>
<td>Russian</td>
</tr>
<tr>
<td>Chinese (China)</td>
<td>Estonian</td>
<td>Icelandic</td>
<td>Serbian</td>
</tr>
<tr>
<td>Chinese (Hong Kong)</td>
<td>Filipino</td>
<td>Indonesian</td>
<td>Slovak</td>
</tr>
<tr>
<td>Chinese (Taiwan)</td>
<td>Finnish</td>
<td>Italian</td>
<td>Slovenian</td>
</tr>
<tr>
<td>Croatian</td>
<td>French</td>
<td>Japanese</td>
<td>Spanish</td>
</tr>
<tr>
<td>Czech</td>
<td>French (Belgium)</td>
<td>Korean</td>
<td>Spanish (Mexico)</td>
</tr>
<tr>
<td>Danish</td>
<td>French (Canada)</td>
<td>Latvian</td>
<td>Spanish (Spain)</td>
</tr>
<tr>
<td>Dutch</td>
<td>French (France)</td>
<td>Lithuanian</td>
<td>Swedish</td>
</tr>
<tr>
<td>English</td>
<td>French (Switzerland)</td>
<td>Malay</td>
<td>Thai</td>
</tr>
<tr>
<td>English (Australia)</td>
<td>German</td>
<td>Norwegian</td>
<td>Turkish</td>
</tr>
<tr>
<td>English (Canada)</td>
<td>German (Austria)</td>
<td>Persian</td>
<td>Ukrainan</td>
</tr>
<tr>
<td>English (Ireland)</td>
<td>German (Switzerland)</td>
<td>Polish</td>
<td>Vietnamese</td>
</tr>
</tbody>
</table>
HOW TO SET THE LANGUAGE FOR ZENDESK OUTPUT

1. Open your target.
2. Click the Language tab.
3. Click the drop-down in the Language field and 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.
8. In the Languages section, click the drop-down arrow to select another language.

9. Click Save tab.
11. Click the drop-down arrow in the **Language** section to apply the language to be used for your Zendesk dashboard.

![Language settings](image)

12. 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.
Exporting Projects for Translation

You can export an entire Flare project, or parts of one, to another location. One reason you might want to use this feature is to quickly and easily archive projects, especially if you have an extremely large Flare project and need to archive only parts of it. Another use for this feature is translation. If you only need a portion of a master project to be translated, you don’t want to send the translator all of the files, but rather a smaller version of the project containing only the files requiring translation. For more information see the Flare online Help.

EXAMPLE — Translation Using Target

You have a Flare project with seven targets you need to translate the content associated one of those targets from English to French. You could send your 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.
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 your 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. 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.
Stitching PDFs

You can stitch existing PDFs into your output by adding links to them in a table of contents (TOC). This is supported in PDF output and all of the online targets.

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

Let’s say 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.

We dragged the three PDF files from the Content Explorer...

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. Additionally, because you do not need to generate each PDF separately every time the document is updated, generating output from a multilingual PDF target saves considerable time.

**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 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.

NOTE If you have inserted MadCap Capture images that contain objects with text, you can auto-size those objects automatically when the output is generated. This can be 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. You might use this option in case you accidentally cut off text in your image callouts or if they are translated into another language that requires more characters in the translation.
Selecting a Language

Broadly speaking, selecting a language can mean a couple of different things. First, it can mean specifying a language for the content that you are producing. However, it can also mean choosing a language for the Flare interface as you do your work.

This chapter discusses the following:

Selecting a Language for Your Content .............................................. 49
Selecting a Language for the Interface ................................................. 53
Selecting a Language for Your Content

You can select a language at any of the following levels:

- Project
- Target
- Topic
- Content within a topic or snippet

The broader the level where the language is set, the lower precedence it has in case of conflicts. In other words, the content level has precedence over everything. The topic level has precedence over the target and project levels. And the target level has precedence over the project level.

You can also set multiple languages at the target level, then link translated Flare projects to your current project. This lets you create multilingual output.

**HOW TO SELECT A LANGUAGE FOR THE ENTIRE PROJECT**

1. If you have not created the project yet, start a new project and select the language when you come to the appropriate page of the wizard. Otherwise, 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 Project Properties dialog opens.

3. Select the Language tab.

4. From the list, select a language (those in bold font are supported in spell check).

5. Click OK.

6. Click to save all files.
HOW TO SELECT LANGUAGES FOR A SPECIFIC TARGET

1. Open the 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 add additional languages to your target output, 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. Linking to a Lingo project lets you skip the step in Lingo of exporting translated content out to additional Flare projects.
      If you have not added a linked project yet, the link will look like an ellipsis; if you have already added a linked project, the link will display the project’s file name.
      
      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 you can do both.
   c. In the dialog that opens, 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 will automatically detect the linked project's default language settings, so you should 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 will be noted in the Localized Skins column. If you have a custom language skin for the selected language in your linked project, Flare will automatically use your custom translations. See “Creating User-Defined Language Skins” on page 60.
      
      NOTE If you link to a multilingual Lingo project, only the remaining languages from a multilingual project can be chosen from the drop-down. In other words, if you’ve already selected one of the languages for a previous row, it is no longer available when you add more language rows.
The default target output language in the Target Editor always shows the currently selected project language. This language will update 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 will appear in your output in the order they are listed on the Language tab.

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 master 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 master 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 master Flare project) with content that has not yet been translated.

6. (Optional) If you selected a right-to-left (RTL) language, you may 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 which output type you are using.

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

7. Click  to save your work.

### HOW TO SELECT A LANGUAGE FOR SPECIFIC CONTENT WITHIN A TOPIC OR SNIPPET

1. Open the topic or snippet.
2. Highlight the content where you want to change the language.
3. Select **Home > Language**.
4. Select the language from the list (those in bold font are supported in spell check).
5. Click **OK**.
6. Click  to save your work.
Selecting a Language for the Interface

You have the option of viewing the interface in English, French, German, Japanese, or Chinese. All of these languages are available from the same version of Flare (you do not need to purchase different versions for different languages). When you initially launch Flare, you are asked to select your language preference. Based on your selection, the user interface is displayed in the selected language, with the option to switch your preferences at any time from the Select UI Language dialog. If a particular element is not available in the selected language, that element displays in English.

HOW TO SELECT A LANGUAGE FOR THE FLARE INTERFACE

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

   If you have previously selected to disable this dialog when launching Flare, you can overrule that selection. To do this:

   a. Select File > Options. The Options dialog opens.
   b. Select the General tab.
   c. Click the check box Show Select UI Language Dialog on Startup.
   d. Click OK.
   e. Launch Flare again.

2. Select the language that you want to use from the drop-down list.

3. (Optional) If you do not want the dialog to display when you launch Flare, click Show this dialog on startup to remove the check mark. If you later decide that you do want to see this dialog when launching Flare, open the Options dialog and click the box to open this dialog on startup.

4. Click OK. Flare is displayed according to your language selections.
A regular skin controls the interface that users see for online outputs. They are stored in the Skins folder of the Project Organizer. By default, the text in these skins is in English, and they are used not only for editing text but also to control the look of the skin in the output.

A language skin can be used to display the interface in a specific language for online outputs. Language skins also control some text-based items, like cross-reference links. Unlike regular skins, language skins hold only text used for the output; they do not control the look of the output.

This chapter discusses the following:

- Completed Versus User-defined Language Skins ........................... 55
- Creating User-Defined Language Skins ................................. 60
- Editing Language Skins ............................................................ 63
- Changing Text Strings in HTML5 Skins ...................................... 65
Completed Versus User-defined Language Skins

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

For other languages, you can create language skins so that they can be applied to a project or target too (see "Creating User-Defined Language Skins" on page 60). After you create a user-defined language skin, it too is identified as "localized." User-defined language skins are stored in the Advanced folder of the Project Organizer.

### EXAMPLE

You select French as the project language.

<table>
<thead>
<tr>
<th>Language</th>
<th>Localized Skins</th>
</tr>
</thead>
<tbody>
<tr>
<td>English (United States)</td>
<td>Yes</td>
</tr>
<tr>
<td>English (Zimbabwe)</td>
<td>No</td>
</tr>
<tr>
<td>Estonian</td>
<td>No</td>
</tr>
<tr>
<td>Focroce</td>
<td>No</td>
</tr>
<tr>
<td>Filipino (Philippines)</td>
<td>No</td>
</tr>
<tr>
<td>Finnish</td>
<td>Yes</td>
</tr>
<tr>
<td>French (Belgium)</td>
<td>No</td>
</tr>
<tr>
<td>French (Canada)</td>
<td>No</td>
</tr>
<tr>
<td>French (France)</td>
<td>No</td>
</tr>
<tr>
<td>French (Luxembourg)</td>
<td>No</td>
</tr>
<tr>
<td>French (Monaco)</td>
<td>No</td>
</tr>
<tr>
<td>French (Switzerland)</td>
<td>No</td>
</tr>
</tbody>
</table>

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:

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 63).
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.
NOTE Typically, language skins are designed for online outputs only (HTML5, WebHelp, and WebHelp Plus).

However, if you are editing text strings in a language skin for toolbar items, those strings will be translated in Microsoft HTML Help output as well.

In addition, the "Formats/Cross-Reference" items in the Language Skin Editor are used for both print-based and online outputs. Those items are intended to set text for context-sensitive cross-references.

Another print-based feature controlled in the language skin is the heading text for auto-generated glossaries, indexes, and proxies.

Language skins also control features found in MadCap Feedback, such as comment labels and profile items.

NOTE The default target output language in the Target Editor always shows the currently selected project language. This language will update automatically if you change the project language.
Creating User-Defined Language Skins

There are two pieces to the generated output: the content and the user interface (e.g., navigation buttons and labels). If you are generating an online Help output (HTML5, WebHelp, or WebHelp Plus), you can select a specific language skin for displaying the user interface. You can then edit the language skin as necessary, adjusting text for buttons and labels. Normally, you would edit a skin in the Skin Editor. However if you want to display the output user interface in a particular language, you can use the Language Skin Editor instead.

Flare provides language skins for certain languages, such as French, German, and Spanish (i.e., many values are already translated). These are languages that are already identified as being "localized" skins. However, you can edit these default translations by creating a language skin for the language. For languages that do not provide default translations, you can create language skins so that they can be applied to the output too.

EXAMPLE

Let’s say that you want to display the user interface of the HTML5 output in Bulgarian. Flare does not provide a language skin in Bulgarian, so you can create your own skin, entering the Bulgarian text in the skin where appropriate. After you create the user-defined language skin with Bulgarian text, you can select it for use in the HTML5 output interface.
HOW TO CREATE A USER-DEFINED LANGUAGE SKIN

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 one of the following:
   - **New from template** Choose either a factory template file or one of your own custom template files as a starting point. The new file will take on all of the settings contained in the template. If you want to use the factory template provided by Flare, expand the Factory Templates folder and click on a template file. If you want to use your own custom template file, expand the appropriate folder and click on a file. For more information about templates, see the online Help.

   **NOTE** If you have existing language skins in your AppData folder, you can use them as templates for new language skins. Select New as template, then select an existing language skin from the Language Skins folder.

   - **New from existing** Choose an existing file of the same type as a starting point for your new file. As with template files, your new file will take on all of the settings contained in the file you select. To use this option, click use the Open File dialog to find a file, and double-click it.

4. (Optional) If you want to place the file into a subfolder previously created in the Content Explorer or Project Organizer, in the Folder field click and select the subfolder. Otherwise, keep the default location.

5. 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.

6. In the File Name field, type a new name for the language skin.
7. (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.

8. (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.

9. 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 UI text strings.

**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.
Editing Language Skins

If you create a new user-defined language skin, you need to edit it to make sure it provides all the correct text strings. You can also open a completed language skin to edit the translated text if you want.

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. (Optional) From the Skin Language drop-down, select the language whose values you want to edit.

   ❯ **NOTE** Selecting a new language resets the existing translations to the default values for the new language.

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

7. Click to save your work.
Changing Text Strings in HTML5 Skins

As 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. This tab not only lets you enter text in other languages, but it lets you change the substance of text strings altogether, whether they are in your primary language or another.

**HOW TO CHANGE TEXT STRINGS IN AN HTML5 SKIN OR COMPONENT**

1. Open an HTML5 skin or a skin component.
2. Select the **UI Text** tab.
3. From the **Language** drop-down, select the language for the skin.
4. In the grid below, type text in the **Value** field for each relevant row. If you want to insert a variable, you can click ▼ The variable will appear as syntax in the field, but in the output the variable definition will be shown.
5. Click ▼ to save your work.

**NOTE** You can also translate interface values in your stylesheet, language skin, or using a legacy language skin. When you set these values in multiple places, they are prioritized as follows:

1. Stylesheet
2. HTML5 skin
3. Language skin
4. Legacy language skin, still in your AppData\Roaming folder

As a best practice, you should try to translate each interface value in a single location to prevent conflicts.

**NOTE** If you use the Language Skin Editor to edit an HTML5 skin value, your changes will be reflected in the HTML5 skin file (on the UI Text tab of the Skin Editor). However, changes made in the Skin Editor are not reflected in the Language Skin Editor.
Importing Dictionaries

You can see which dictionaries are installed on your computer by opening the Options dialog and selecting the Spelling tab.

You can also use the Options dialog to import more dictionaries that you have downloaded.
HOW TO IMPORT A DICTIONARY

1. Download a dictionary from OpenOffice.org.
2. Select File > Options
   The Options dialog opens.
3. Select the Spelling tab.
4. Click Import Dictionaries.
5. In the dialog that opens, locate and double-click the dictionary that you downloaded. The dictionary is added to the Options dialog.
6. Click OK.

NOTE Dictionaries are stored in your AppData Windows folder.
APPENDIX

PDFs

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

TUTORIALS

Getting Started Tutorial
Product Foldout Tutorial
Side Navigation Tutorial
Top Navigation Tutorial

USER GUIDES

Accessibility Guide
Analysis and Reports Guide
Architecture Guide
Autonumbers Guide
Condition Tags Guide
Context-Sensitive Help Guide
Eclipse Help Guide
Getting Started Guide
Global Project Linking Guide

HTML Help Guide
HTML5 Guide
Images Guide
Import Guide
Indexing Guide
Key Features Guide
Language Support Guide
MadCap Central Integration Guide
Master Pages Guide
Micro Content Guide  Source Control Guide: Perforce
Movies Guide  Source Control Guide: Subversion
Navigation Links Guide  Source Control Guide: Team Foundation Server
Plug-In API Guide  Styles Guide
Print-Based Output Guide  Tables Guide
Project Creation Guide  Tables of Contents Guide
QR Codes Guide  Targets Guide
Reports Guide  Templates Guide
Reviews & Contributions Guide  Topics Guide
Search Guide  Touring the Workspace Guide
SharePoint Guide  Transition From FrameMaker Guide
Skins Guide  Variables Guide
Snippets Guide  What's New Guide
Source Control Guide: Git

CHEAT SHEETS

Folders and Files Cheat Sheet
Print-Based Output Cheat Sheet
Shortcuts Cheat Sheet
Structure Bars Cheat Sheet
Styles Cheat Sheet