MadCap Software

Language Support Guide

Flare 10

Copyright 2014 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
7777 Fay Avenue
La Jolla, California 92037
858-320-0387
www.madcapsoftware.com

THIS GUIDE WAS CREATED USING MADCAP FLARE.

CONTENTS

CHAPTER 1 About Language Support 5

Selecting a Language 6
Authoring Content 7
Language Skins 8
Invert Styles, Page Layouts, and Image Callouts at the Target Level 9
Invert Hotspot Images 11
Table Styles and RTL Languages 13
Dictionaries 14
Translating Flare Projects 15
Exporting Projects for Translation 16

CHAPTER 2 Selecting a Language 19

Selecting a Language for the Interface 23

CHAPTER 3 Language Skins 25

Completed Versus User-Defined Language Skins 26
Tasks for Using Language Skins 30
Creating User-Defined Language Skins 31
Editing Language Skins 32
Changing Text Strings in HTML5 Skins 34

CHAPTER 4 Importing Dictionaries 35

APPENDIX PDF Guides 39

CHAPTER 1

About Language Support

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, double-byte languages, and more. RTL languages are supported in all outputs except DotNet Help and FrameMaker.

This chapter discusses the following:

Selecting a Language 6
Authoring Content 7
Language Skins 8
Invert Styles, Page Layouts, and Image Callouts at the Target Level 9
Invert Hotspot Images 11
Table Styles and RTL Languages 13
Dictionaries 14
Translating Flare Projects 15
Exporting Projects for Translation 16

Selecting a Language

You can select a language for a project, target, topic, or even specific content. Selecting a language affects spell checking as well as the language skin used. See "Selecting a Language" on page 19.

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.

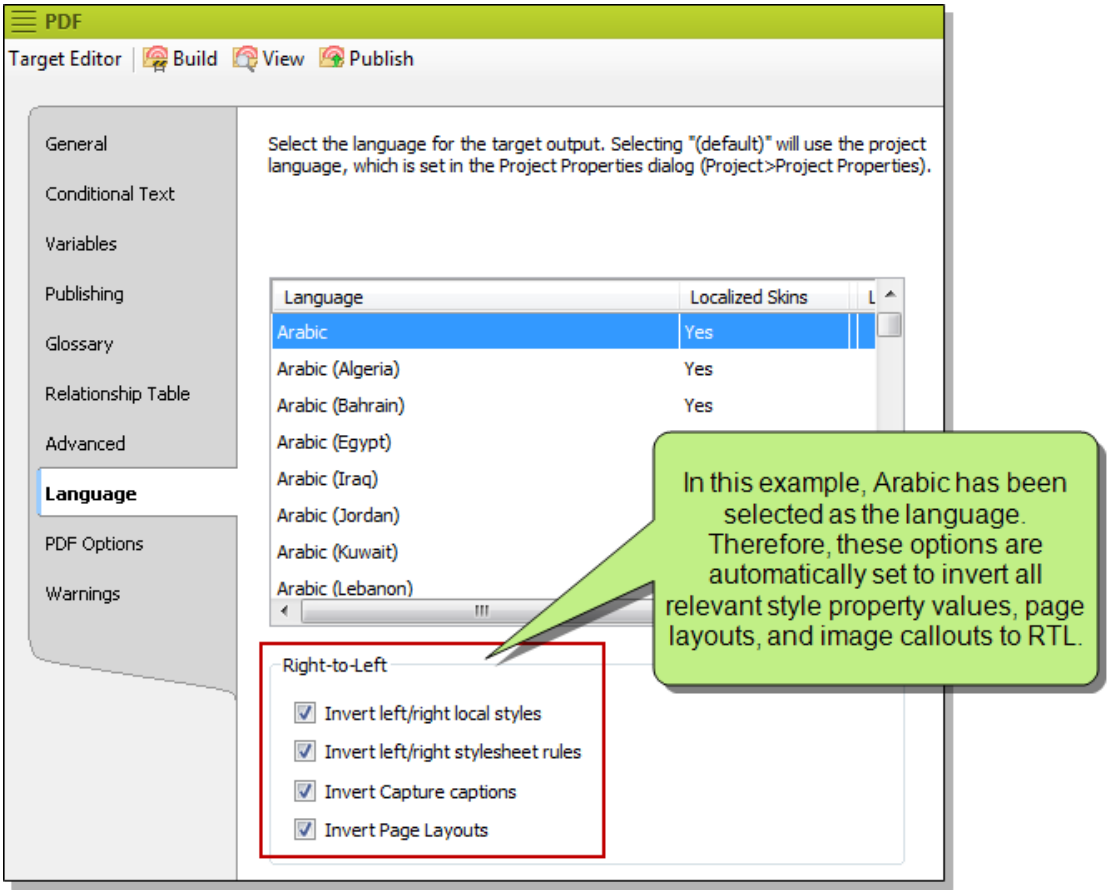
Language Skins

A language skin can be used to display the interface in a specific language for most WebHelp outputs (WebHelp, WebHelp AIR, WebHelp Mobile, and WebHelp Plus). Language skins hold only text used for the output; they do not control the look of the output. To control the appearance of the output, you must use a regular skin in addition to the language skin. See "Language Skins" on page 25.

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. For other languages, you can create language skins. See "Creating User-Defined Language Skins" on page 31 and "Editing Language Skins" on page 32.

Invert Styles, Page Layouts, and Image Callouts at the Target Level

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 19.



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

- **PDF/XPS/XHTML/Word** All four options are shown.
- **HTML5/HTML Help/EPUB/All WebHelps** Local styles, CSS styles, and image callout options are shown.
- **DITA/DotNet Help/FrameMaker** No options are shown (FrameMaker and DotNet Help do not support RTL languages).

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.

E X A M P L E

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.

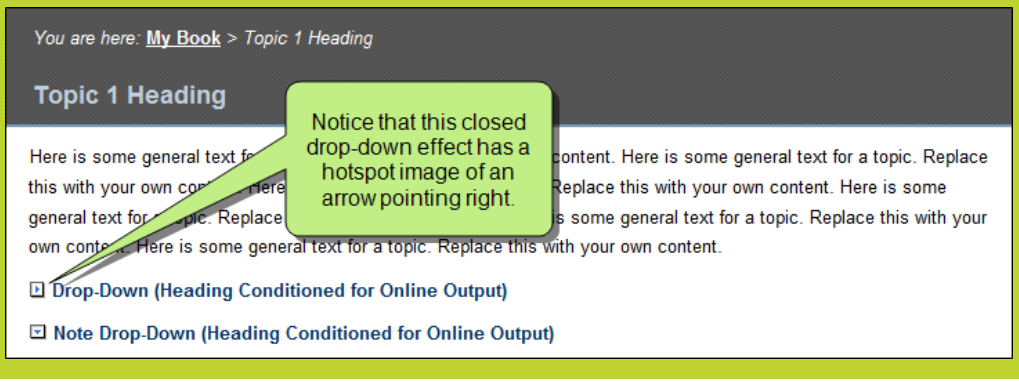
Note: If your selected language is LTR, these options cannot be accessed in the Target Editor.

Invert Hotspot Images

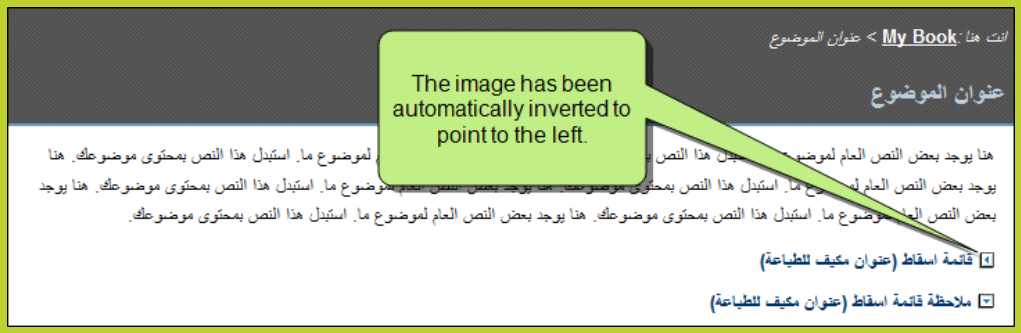
There are certain features in Flare that use special hotspot images (e.g., drop-down and expanding text effects). Flare's default images for these features are designed to work with LTR outputs. However, if you are using an RTL language, Flare automatically inverts the default images so that they make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

E X A M P L E

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 and selecting the Dictionaries 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 in Flare. See "Importing Dictionaries" on page 35.

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 the translator, who receives a complete list of files that he or she 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.

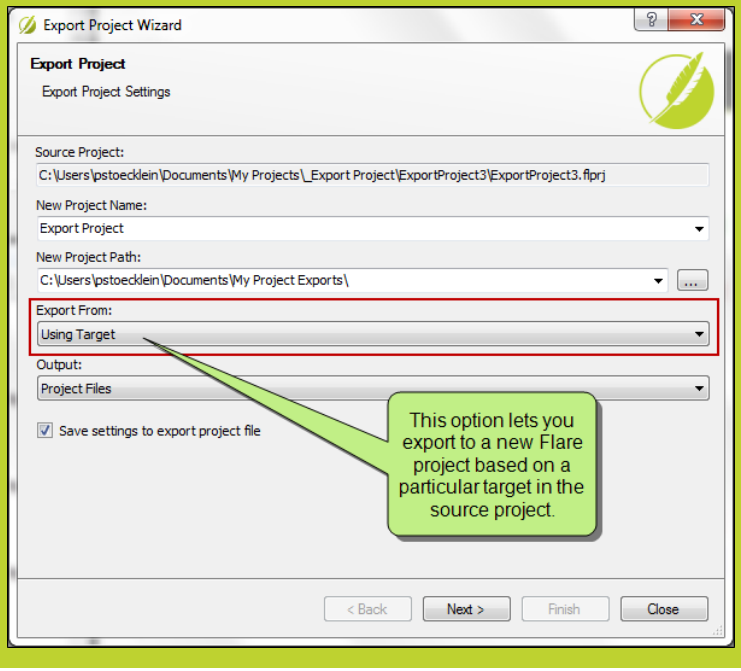
Exporting Projects for Translation

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.

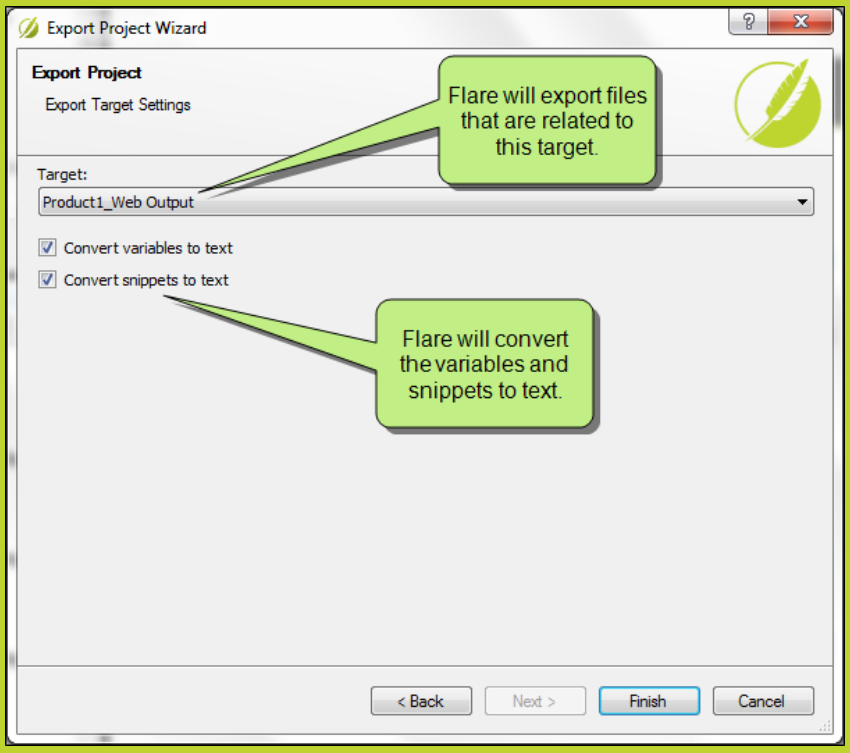
EXAMPLE—TRANSLATION

Let's say 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.

Note: The freely distributable MadCap Help Viewer also supports multiple languages in the interface (English, French, German, and Japanese). Therefore, if you build and distribute DotNet Help, users will be able to select from the available languages to view the interface for your online Help.

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 the external version of MadCap Analyzer, 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.

CHAPTER 2

Selecting a Language

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.

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 the **Project** ribbon. In the **Properties** section select **Project Properties**.

   You can use the Options dialog to switch between ribbons and the classic tool strip layout. For more information see the online Help.

   Keep in mind that the smaller the application window becomes, the more the options in a ribbon shrink. Therefore, you might only see a small icon instead of text, or you might see only a section name displayed with a down arrow to access the options in it. You can hover over small icons to see tooltips that describe them. You can also enlarge the application window or click one of the section drop-downs in the ribbon to locate a hidden feature.

   » **Menu** Select **Project>Project Properties**.

   » **Project Toolbar** Click Project Properties...

   To open the Project toolbar from the menu view, select **View>Toolbars>Project**.

   » **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. Select **File>Save>Save All** (if using ribbons), select **File>Save All** (if using menus), or press **CTRL+SHIFT+S**.

HOW TO SELECT A LANGUAGE FOR A SPECIFIC TARGET

1. Open the target.
2. In the Target Editor, select the **Language** tab.
3. Select the language from the list (those in bold font are supported in spell check).
4. If you selected a right-to-left (RTL) language, you may see the following options, 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/XPS/XHTML/Word** All four options are shown.
   - **HTML5/HTML Help/EPUB/Eclipse Help/All WebHelps** Local styles, CSS styles, and image callout options are shown.
   - **DITA/DotNet Help/FrameMaker** No options are shown (FrameMaker and DotNet Help do not support RTL languages).
5. Click to save your work.

HOW TO SELECT A LANGUAGE FOR A SPECIFIC TOPIC

1. Make sure the Content Explorer is open.
2. Find and select the topic.
3. In the local toolbar of the Content Explorer, click . The Properties dialog opens.
4. Select the **Language** tab.
5. From the list, select a language (those in bold font are supported in spell check).
6. Click **OK**.

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. Do one of the following, depending on the part of the user interface you are using:
   - **Ribbon** Select the **Home** ribbon. In the **Attributes** section select **Language**.
   - **Menu** Select **Format>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, or Japanese. 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 the order of your language preferences. Based on your selection, the user interface is displayed in the appropriate language, with the option to later switch your preferences at any time from the Select UI Language dialog. Flare's user interface displays in the first language listed in your preferences. If, for any reason, a particular element is not available in that language, the element displays in the next language in the list, and so on.

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. Do one of the following, depending on the part of the user interface you are using:

      - **Ribbon** Select **File>Options**.
      - **Menu** Select **Tools>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 move in the list. (The language displayed highest in the list will be the first choice in displaying the interface. If that language is not available for a particular element, the next language in the list is used, and so on.)
3. Click the **Up** or **Down** button until that language is in the proper location in the list.
4. Repeat steps 2 and 3 until the list displays the languages in the order that you want.
5. (Optional) If you do not want the dialog to display the next time 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.
6. Click **OK**. Flare is displayed according to your language selections.

Note: The MadCap Help Viewer also supports multiple languages in the interface. Therefore, if you build and distribute DotNet Help, users will be able to select from the available languages to view the interface for your online Help.

CHAPTER 3

Language Skins

This feature is supported in all outputs, except HTML5.

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 (e.g., color, navigation elements).

On the other hand, a language skin can be used to display the interface in a specific language for most WebHelp outputs (WebHelp, WebHelp AIR, WebHelp Mobile, and WebHelp Plus). Language skins hold only text used for the output; they do not control the look of the output. To control the appearance of the output, you must use a regular skin in addition to the language skin.

This chapter discusses the following:

Completed Versus User-Defined Language Skins 26
Tasks for Using Language Skins 30
Creating User-Defined Language Skins 31
Editing Language Skins 32
Changing Text Strings in HTML5 Skins 34

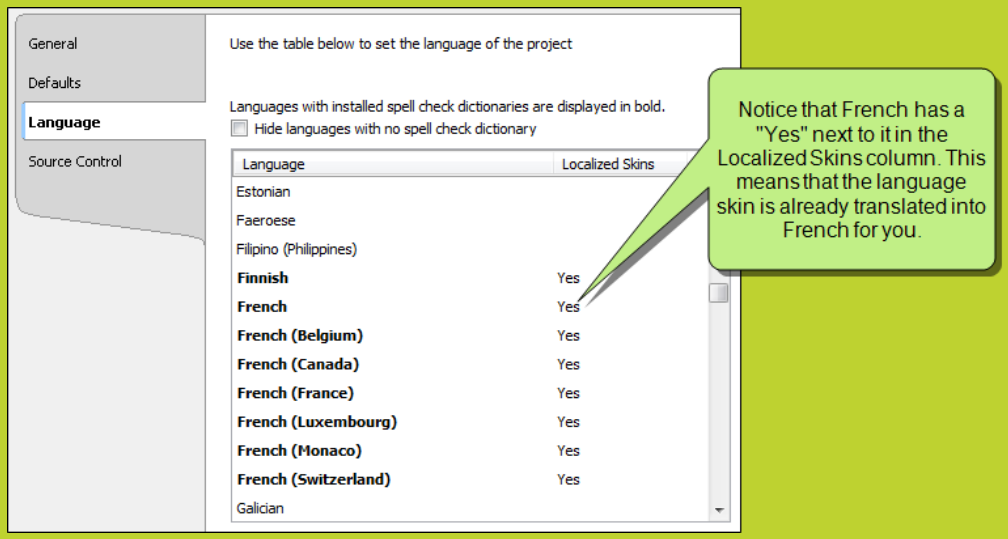
Completed Versus User-Defined Language Skins

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.

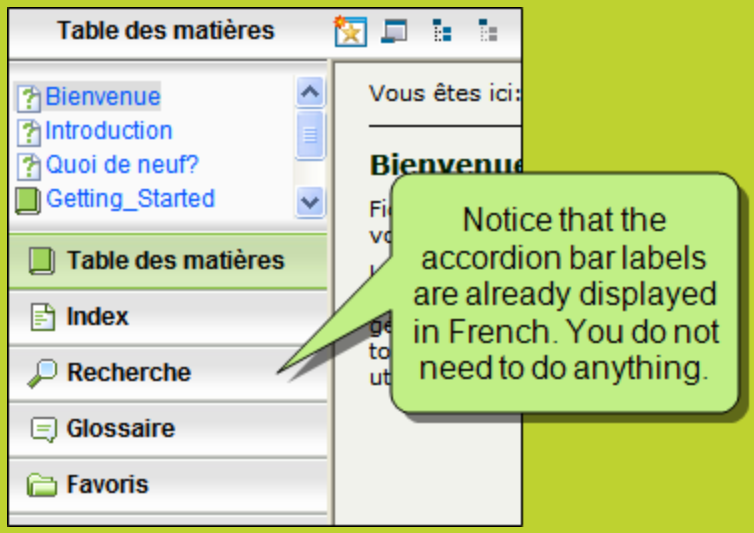
For other languages, you can create language skins so that they can be applied to a project or target too (see "Creating User-Defined Language Skins" on page 31). After you create a user-defined language skin, it too is identified as "localized." User-defined language skins that you create are stored in your application data folder, outside of your Flare project (e.g., C:\Users\myname\AppData\Roaming\MadCap Software\Flare\Language Skins).

E X A M P L E

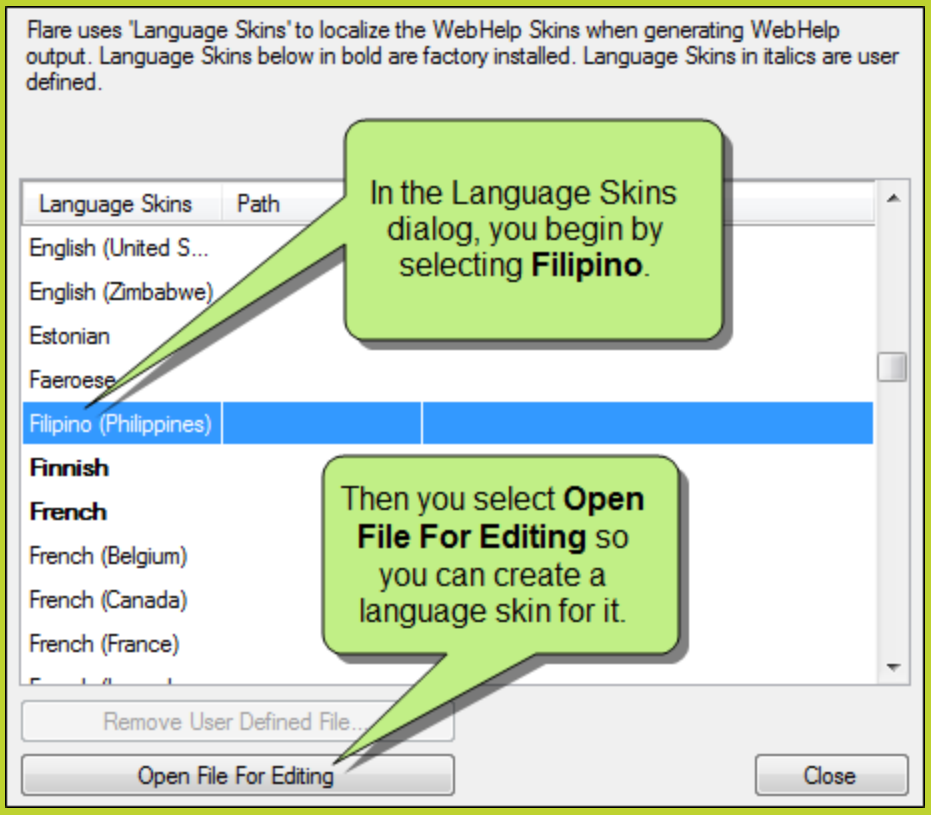
Let's say you select French as the project language.



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



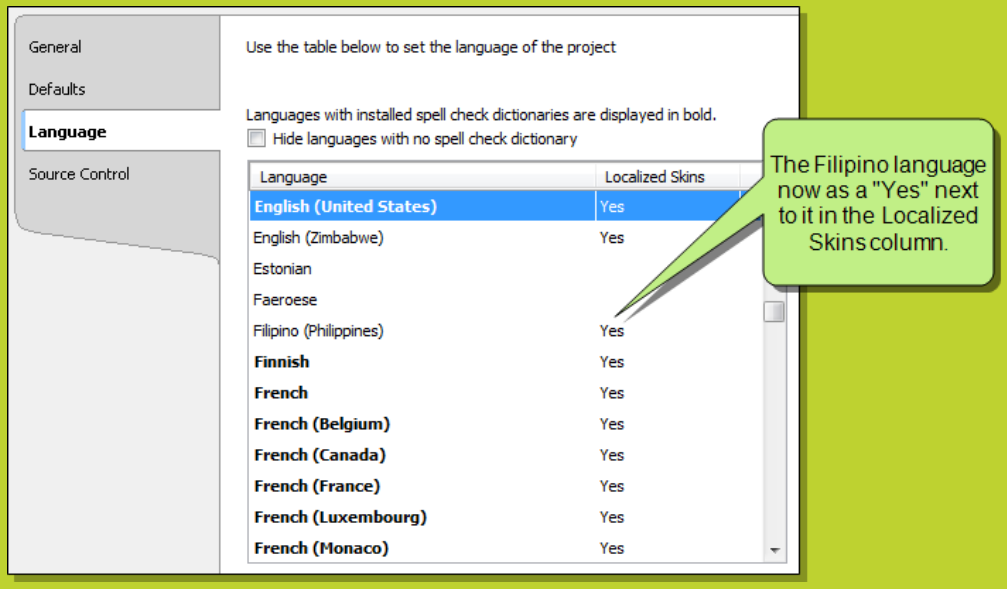
Now suppose that you want to use Filipino as the project language. Notice in the example above that Filipino does not have a "Yes" next to it. This means that you need to create a language skin for it from the Language Skins dialog (**Tools>Manage Language Skins**).



When you do this another dialog opens, asking you to select a language to use as a starting point. Some languages are relatively close to others. Therefore, by selecting a language close to the one you want, you probably do not need to do as much editing.

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 32).

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.



Tasks for Using Language Skins

Following are the basic steps involved with language skins.

1. **(Optional) Create** If you are using a language for which Flare has already provided a completed language skin, you do not need to complete this step. However, if you want to use another language, you can create a skin for it. See "Creating User-Defined Language Skins" on the following page.

2. **(Optional) Edit**If you create a new user-defined language skin, you need to edit it to make sure it provides all the correct translations. You can also open a completed language skin to edit the translated text if you want. See "Editing Language Skins" on page 32.

   Note: If you want to make other changes to a skin to affect how it looks in the output, you can edit a regular skin in the Project Organizer.

3. **Select**You need to tell Flare which language you are working with so that it knows which skin to use in the output. This can be done when you are first creating the project. You can also change the language for the entire project later by using the Project Properties option. You can also override the project language and set it for the individual targets or topics. See "Selecting a Language" on page 19.

Note: Typically, language skins are designed for most WebHelp outputs only (WebHelp, WebHelp AIR, WebHelp Mobile, and WebHelp Plus).

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

In addition, the "Formats/Cross-Reference" items in the Language Skin Editor are used for print-based outputs (Adobe FrameMaker, Adobe PDF, Microsoft Word, Microsoft XPS, XHTML) and EPUB, not for WebHelp or the other online outputs. Those items are intended to set text for context-sensitive cross-references in print output.

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

The lone exception to language skins is HTML5. Unlike other output types, separate language skins are not used for HTML5 targets. Instead, a UI Text tab is located in the Skin Editor, which lets you perform all of the localization tasks that you would normally accomplish in a language skin. See "Changing Text Strings in HTML5 Skins" on page 34.

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 one of the WebHelp output formats (HTML5, WebHelp, WebHelp AIR, WebHelp Mobile, or WebHelp Plus) output, 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 completed language skins for certain languages, such as French, German, and Spanish (i.e., they are already translated for you). These are languages that are already identified as being "localized" skins. For other languages, you can create language skins so that they can be applied to the output too.

E X A M P L E

Let's say that you want to display the user interface of the WebHelp 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 WebHelp 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 the **Tools** ribbon. In the **Templates** section select **Manage Language Skins**.
   - **Menu** Select **Tools>Manage Language Skins**.

   The Language Skins dialog opens.
2. From the list, select a language in normal font (not bold, not italics).
3. Click **Open File For Editing**. The Select Template Language Skin dialog opens.
4. From the list, select an existing language skin to use as a template in creating the new one. Select a language that is closest to the new language skin. That way, you may not have to change as much when editing the skin.
5. Click **OK**.
6. Click **OK** in the message that opens. The skin is created and the Language Skin Editor opens.

Editing Language Skins

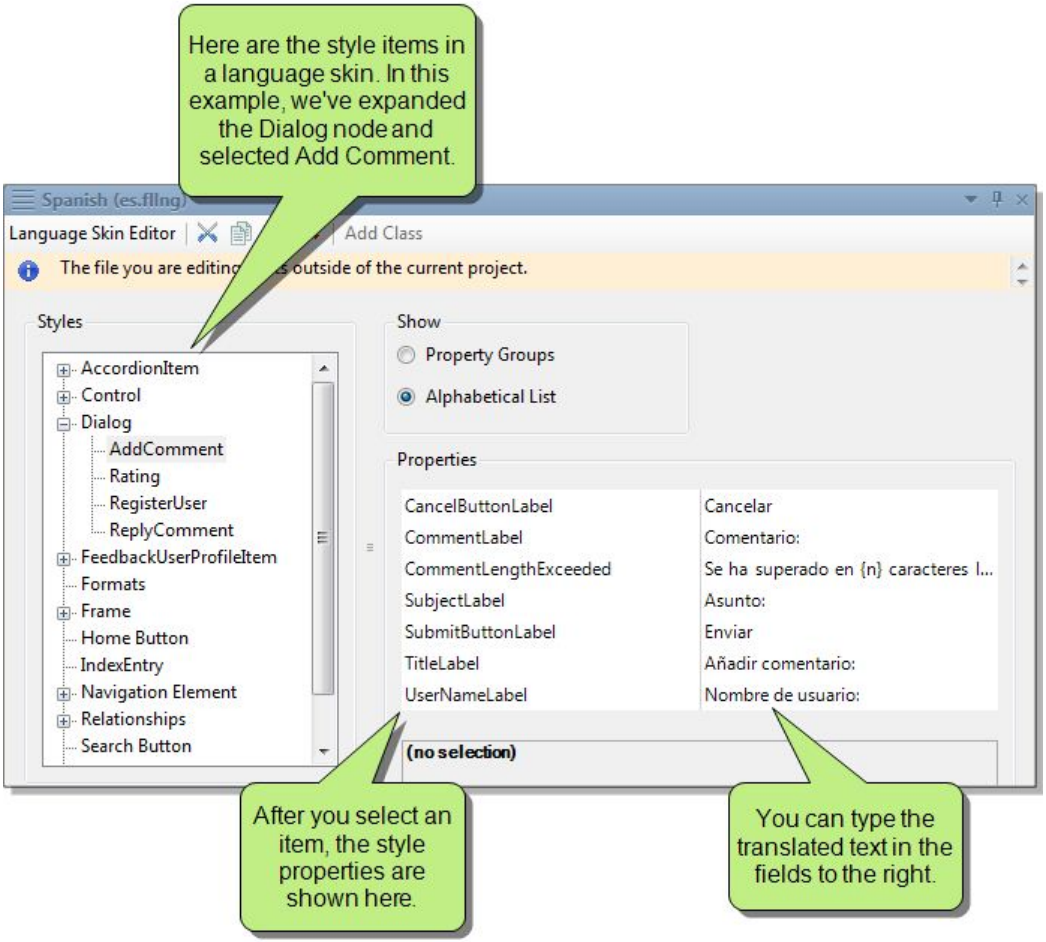
If you create a new user-defined language skin, you need to edit it to make sure it provides all the correct text strings. You can also open a completed language skin to edit the translated text if you want.

HOW TO EDIT A LANGUAGE SKIN

1. Do one of the following, depending on the part of the user interface you are using:
   - **Ribbon** Select the **Tools** ribbon. In the **Templates** section select **Manage Language Skins**.
   - **Menu** Select **Tools>Manage Language Skins**.

   The Language Skins dialog opens.
2. From the list, select a language in bold or italic font.
3. Click **Open File For Editing**. If you selected a language in bold, click **OK** in the message that displays. The Language Skin Editor opens.
4. In the **Styles** expand the item that you want to modify (e.g., Accordion Item, Control, Dialog), if necessary. Then select a style under the node. If a node cannot be expanded (e.g., Formats, Home Button), click on the node itself.

   Note: For more information about the purpose of each of these style items in the output, see the online Help.
5. Select the **Property Groups** or **Alphabetical List** option to display the properties for the style item. It is usually easiest to work with properties in an alphabetical list.
6. In the **Properties** section, set the values for the properties that you want modify. Click in the field to the right of the property and type the translated text for that item. (If you are viewing by property group instead of an alphabetical list, you need to expand the property group to see the properties under it.)



7. Click to save your work.

Changing Text Strings in HTML5 Skins

Unlike other output types, separate language skins are not used for HTML5 targets. Instead, a UI Text tab is located in the Skin Editor, which lets you perform all of the localization tasks that you would normally accomplish in a language skin. 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

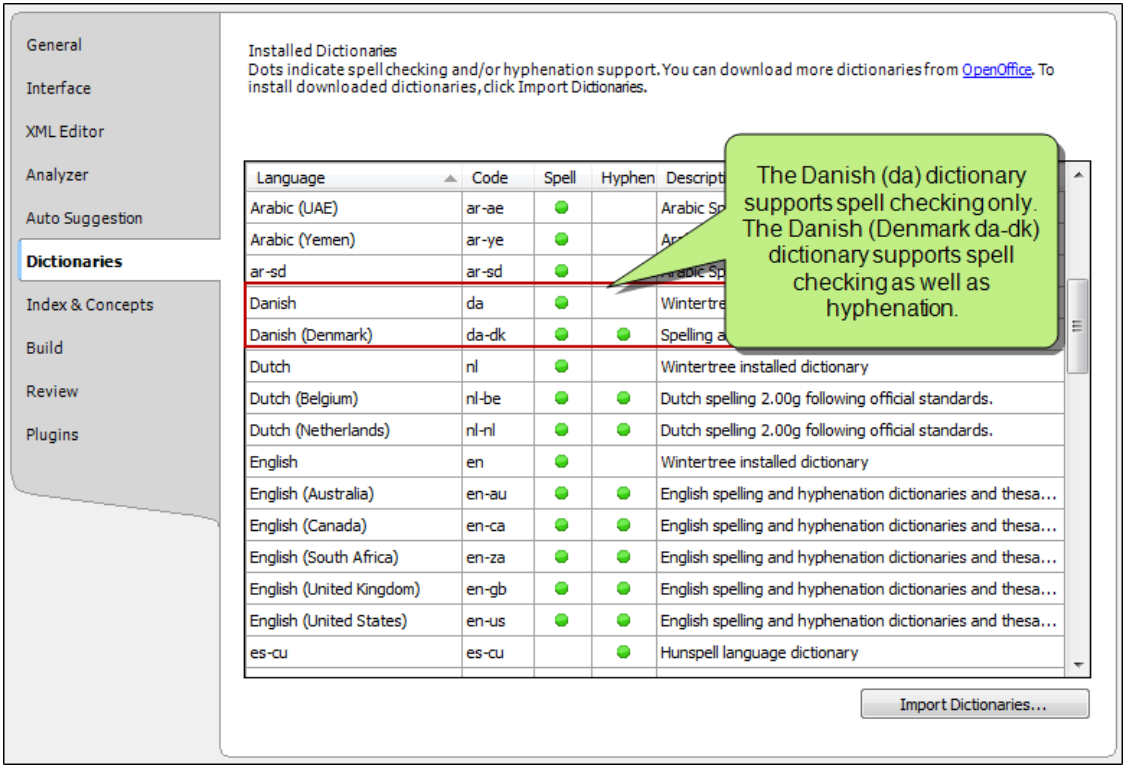
1. Open an HTML5 skin.
2. Select the **UI Text** tab.
3. From the **Language** drop-down, select the language for the skin.
4. In the grid below, type text in the **Value** field for each relevant row.
5. Click to save your work.

CHAPTER 4

Importing Dictionaries

When you create a new project in Flare, you select a language. Several of the languages listed in the Wizard (i.e., those in bold) come with built-in spell checking support. You can see which dictionaries are installed on your computer by opening the Options dialog and selecting the Dictionaries tab.

Green dots indicate which dictionaries are supported by spell checkers and hyphenators. A *hyphenator* automatically inserts separators in an editor based on particular word patterns, as outlined in the dictionary file. Hyphenators are important only for certain languages.


General
Interface
XML Editor
Analyzer
Auto Suggestion
Dictionaries
Index & Concepts
Build
Review
Plugins
Installed Dictionaries
Dots indicate spell checking and/or hyphenation support. You can download more dictionaries from OpenOffice. To install downloaded dictionaries, click Import Dictionaries.
Language
Code
Spell
Hyphen
Descript
Arabic (UAE)
ar-ae
Arabic (Yemen)
ar-ye
ar-sd
ar-sd
Danish
da
Danish (Denmark)
da-dk
Dutch
nl
Wintertree installed dictionary
Dutch (Belgium)
nl-be
Dutch spelling 2.00g following official standards.
Dutch (Netherlands)
nl-nl
Dutch spelling 2.00g following official standards.
English
en
Wintertree installed dictionary
English (Australia)
en-au
English spelling and hyphenation dictionaries and thesa...
English (Canada)
en-ca
English spelling and hyphenation dictionaries and thesa...
English (South Africa)
en-za
English spelling and hyphenation dictionaries and thesa...
English (United Kingdom)
en-gb
English spelling and hyphenation dictionaries and thesa...
English (United States)
en-us
English spelling and hyphenation dictionaries and thesa...
es-cu
es-cu
Hunspell language dictionary
The Danish (da) dictionary supports spell checking only. The Danish (Denmark da-dk) dictionary supports spell checking as well as hyphenation.
Import Dictionaries...

You can also use the Options dialog to import more dictionaries that you have downloaded.

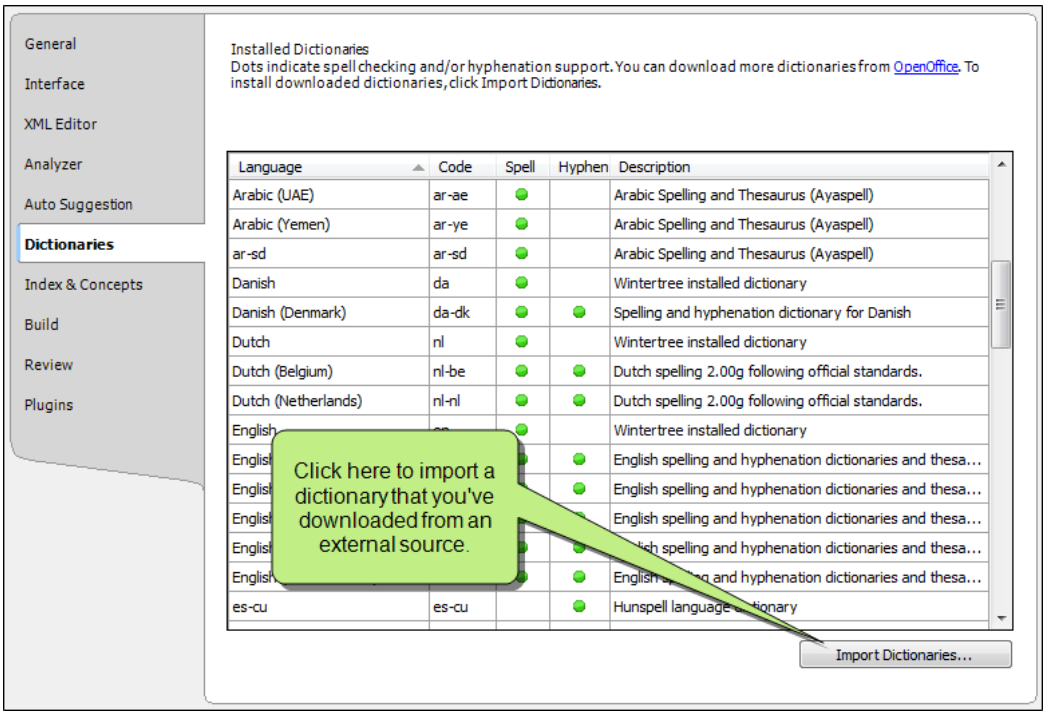
HOW TO IMPORT A DICTIONARY

1. Download a dictionary from **OpenOffice.org**.
2. Do one of the following, depending on the part of the user interface you are using:
   - **Ribbon** Select **File>Options**.

     You can use the Options dialog to switch between ribbons and the classic tool strip layout. For more information see the online Help.

     Keep in mind that the smaller the application window becomes, the more the options in a ribbon shrink. Therefore, you might only see a small icon instead of text, or you might see only a section name displayed with a down arrow to access the options in it. You can hover over small icons to see tooltips that describe them. You can also enlarge the application window or click one of the section drop-downs in the ribbon to locate a hidden feature.
   - **Menu** Select **Tools>Options**.

   The Options dialog opens
3. Select the **Dictionaries** tab.
4. At the bottom of the tab, 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: For more information about hyphenation patterns in different languages, see http://hun-spell.sourceforge.net/tb87nemeth.pdf.

Note: Dictionaries are stored in your AppData Windows folder.

APPENDIX

PDF Guides

The following PDF guides are available for download from the online Help.

Accessibility Guide
Analyzer Guide
Autonumbers Guide
Condition Tags Guide
Context-sensitive Help Guide
DotNet Help Guide
Eclipse Help Guide
Getting Started Guide
Global Project Linking Guide
HTML Help Guide
HTML5 Guide
Images Guide
Importing Guide
Index Guide
Key Features Guide
Language Support Guide
Movies Guide
Navigation Links Guide
Print-based Output Guide
Project Creation Guide
Pulse Guide
QR Codes Guide
Reports Guide
Reviews & Contributions Guide
Search Guide
SharePoint Guide
Shortcuts Guide
Skins Guide

Snippets Guide

Source Control Guide

Styles Guide

Tables Guide

Tables of Contents Guide

Targets Guide

Templates Guide

Topics Guide

Touring the Workspace Guide

Transition From FrameMaker Guide

Variables Guide

WebHelp Outputs Guide

What's New Guide