

MADCAP LINGO 12

What's New

Copyright © 2026 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
1660 17th Street, Suite 201
Denver, Colorado 80202
858-320-0387
www.madcapsoftware.com

THIS PDF WAS CREATED USING MADCAP FLARE.

CONTENTS

CHAPTER 1

Introduction 5

CHAPTER 2

Command Line Enhancements 7

 How to Create a Project Using the Command Line 8

 Library of Command Line Options 10

CHAPTER 3

File List Progress Column 20

CHAPTER 4

Microsoft Translator Region 22

CHAPTER 5

Pseudo Translate 24

CHAPTER 6

QA Reports Ignore Case 27

CHAPTER 7

TM Suggestion Highlight 29

APPENDIX

PDFs	31
Cheat Sheets	31
User Guides	31

CHAPTER 1

Introduction

Following are the new features available in Lingo 12.

For more information about each feature discussed in this manual, open the Help system and refer to the "What's New" topic. Links are provided in some feature descriptions, taking you to topics that contain additional information and steps.



"Command Line Enhancements" on page 7

- Create projects from the command line
- Reference JSON files
- Many more commands available



"File List Progress Column" on page 20

- See the progress of each file
- Shown in percentages



"Microsoft Translator Region" on page 22

- Select a specific region when using Microsoft Translator
- Helps prevent issues with machine translation



"Pseudo Translate" on page 24

- Pre-translate files with pseudo translate
- Helps to see how translations might affect page structure



"QA Reports Ignore Case" on page 27

- Improve report results by ignoring case
- Use for inconsistency in translation of repeated source segments
- Also used for repeated target segments with varying source



"TM Suggestion Highlight" on page 29

- Changed content is highlighted when using translation memory
- Helps with usability

Command Line Enhancements

Many more commands are now available, allowing you to accomplish much more without needing to have Lingo open. This includes the ability to create a project using the command line, as well as referencing a JSON file (via the -config command).

This chapter discusses the following:

- How to Create a Project Using the Command Line 8
- Library of Command Line Options 10

I How to Create a Project Using the Command Line

1. Open Notepad (to create and save a batch file) or open your command prompt (to run a command immediately).

✓ **TIP** The best way to use the command line feature is to create a batch file with the necessary commands in it. Then you can use a scheduling tool (such as the Task Scheduler utility in Windows) to run the batch file automatically whenever you want.

2. Type the path to the Lingo.app folder where you installed Lingo, and press **ENTER**.

```
cd\Program Files\MadCap Software\MadCap Lingo 12\Lingo.app
```

📄 **NOTE** Alternatively, you can use `cd c:\Program Files\[rest of the path]` instead of `cd\Program Files\[rest of the path]`.

3. Type any of the available commands (see "Library of Command Line Options" on page 10), entering the appropriate value in place of the section in brackets. (Additional commands could be added if necessary.)

```
lingoproject.exe -sourceFile [source file] -projectDirectory [folder  
for new Lingo project] -projectName [name for new Lingo project] -  
sourceLanguage [source file language] -targetLanguages [new Lingo  
project language(s)] -function Create
```

☆ EXAMPLE

You have a Flare project called "Cool Project" that is located in your C:\FlareProjects folder. You want to use the files in that project to create a new Lingo project named "Cool Project Translated." The source language in the Flare project is English (United States), and the target languages for the Lingo project will be Spanish and French. You want to store this new Lingo project in your C:\LingoProjects folder. You would type this:

```
lingoproject.exe -sourceFile "C:\FlareProjects\Cool  
Project.flprj" -projectDirectory C:\LingoProjects -projectName  
"Cool Project Translated" -sourceLanguage en-us -  
targetLanguages es;fr -function Create
```



NOTE If there are spaces anywhere in the path/name of your project file, source file, export path, or target language, you need to use quotation marks around it.



NOTE The commands used by the compiler are case-sensitive.

NOW WHAT?

If you are using the command prompt window for any of the commands just described, the project is created immediately.

If you are creating a batch file in Notepad, the project will be created when you run that file.

Library of Command Line Options

Following is a collection of the various command line options that you can use for Lingo.

Global

-config

The location of a config file containing arguments for the console application. Any arguments passed directly on the command line will override the config values. The config file should be in JSON format with the name of the argument as the key with a string value.



EXAMPLE You might have a JSON file named "config.json" that you've stored in your C:\LingoCMD folder. The JSON file contains the following:

```
{
  "sourceFile" : "C:/LingoCMD/Flare/Flare.flprj",
  "projectDirectory" : "C:/LingoCMD/",
  "projectName" : "Lingo",
  "sourceLanguage" : "en-us",
  "targetLanguages" : "es",
  "function" : "create",
  "flattenedVariables" : "Variables.CompanyName",
  "selectedTargets" :
    "Project/Targets/HTML5.fltar",
}
```

In the command line, you could type this:

```
lingoproject.exe -config
"C:\LingoCMD\config.json"
```

-function

The operation you want Lingo to perform. It precedes each of the following: Bundle, Create, Export, MergeBundle, MT, Sync, TM, Update.



EXAMPLES

- function Bundle
- function Create
- function Export
- function MergeBundle
- function MT
- function Sync
- function TM
- function Update

-log

Flag to create log file.

-logFile

The location of the log file to create.

-sourceFile

The source project or file(s). Semicolon-delimited list.



EXAMPLE -sourceFile "C:\SourceFiles\Nursery Rhymes.flprj;C:\Local\text.txt"

Create Lingo Project

Main Commands

-projectDirectory

Path where you want to store the new Lingo project.



EXAMPLE -projectDirectory C:\Folder

-projectName

Name of the new Lingo project.



EXAMPLE -projectName LingoProject

-sourceLanguage

Source language code of the project.



EXAMPLE -sourceLanguage en-us

-targetLanguages

Target language code(s) of the project.



EXAMPLE -targetLanguages es;fr

Other Commands

-client

The client for whom the project is being created.

-domain

The source domain or subject.

-fileFilters	File filter URLs to use. Semicolon-delimited list.
-maintainFiles	Maintain non-translatable files in an imported folder in the exported folder.
-mtProviders	Machine translation (MT) providers to connect to. Pipe-delimited connection strings.
-termbaseConnections	Termbases to connect to. Pipe-delimited connection strings.
-tmConnections	Translation memories (TMs) to connect to. Pipe-delimited connection strings.
-useMT	Flag to pre-translate the project using MT.
-useTranslationMemory	Flag to pre-translate the project using the connected TMs.
Other Commands for DITA Files	
-flattenConrefs	Flag to convert inline conrefs to plain text.
-selectedAudience	Filter to include the selected audience attribute.
-selectedOtherProps	Filter to include the selected other props attribute.
-selectedPlatform	Filter to include the selected platform attribute.
-selectedProps	Filter to include the designated property attribute.

Other Commands for Flare Files

-createLanguageSkin Flag to create language skin for target languages.

-flattenedSnippets Flag to convert inline snippets to plain text.

-flattenedVariables Variables to convert to plain text. Semicolon-delimited variable names.



EXAMPLE -flattenedVariables
General.Variable;Flare.Variable

-selectedConditions Conditional text to exclude from translation. Semicolon-delimited list.



EXAMPLE -selectedConditions
Default.PrintOnly;Default.ScreenOnly

-selectedTargets Filter content by selected target(s). Semicolon-delimited list.



EXAMPLE -selectedTargets
Project/Targets/HTML5.fltar;Project/Targets/PDF.fltar

Other Commands for Source Control

-bind

Flag to bind the project to source control.



EXAMPLE -bind true

-database

The source control database URL.



EXAMPLE -database
http://blah3l7tfs12:8080/tfs/NewCollection

-databaseRoot

The source control path to the project.



EXAMPLE -databaseRoot \$/Flare/Lingo.liprj/

-providerType

The source control provider to bind the project to.



EXAMPLE -providerType "Microsoft Team Foundation
Server"

-sourceControlProjectFile

The URL(s) for the file(s) to import from the source control client.



EXAMPLE -sourceControlProjectFile
https://gitstage.madcapcentral.com/abc/project.git/pr
oject.flprj

<code>-sourceDistributedEmail</code>	The git user email for the client used to import source files.  EXAMPLE <code>-sourceDistributedEmail ldobler@fictionsoft.com</code>
<code>-sourceDistributedName</code>	The git user name for the client used to import source files.  EXAMPLE <code>-sourceDistributedName "Lloyd Dobler"</code>
<code>-sourceDistributedSavePer</code>	The save per setting for the git client for the client used to import source files.
<code>-sourcePerforceServer</code>	The Perforce server URL for the client used to import source files.
<code>-sourcePerforceUser</code>	The Perforce user for the client used to import source files.
<code>-sourceProviderType</code>	The source control provider to bind the project to.  EXAMPLE <code>-sourceProviderType Git</code>
<code>-sourceSubversionServer</code>	The Subversion server URL for the client used to import source files.
<code>-sourceTeamFoundationServer</code>	The TFS server URL for the client used to import source files.

-sourceUser

The source control user for the client used to import source files.



EXAMPLE `-sourceUser apiuser1`

-sourceUserPass

The source control password for the client used to import source files.



EXAMPLE `-sourceUserPass mypassword121!`

-user

The source control user.



EXAMPLE `-user ldobler@fictionsoft`

-userSecret

The source control password, or Personal Access Token (PAT) when binding to Git.



EXAMPLE `-userSecret mypassword875!`

Bundle / Merge Lingo Project

Main Commands

-bundleFiles

The filenames (relative paths inside project e.g. Content/Topic.htm) that will be included in the bundle. If omitted, all files with segments and images will be included.

-bundleType	Lingo or XLIFF bundle type. Default is Lingo bundle.
-merge	Flag to merge bundle.
-mergeFile	Bundle URL.
-project	Path and name of the Lingo project to bundle from or merge to.
	 EXAMPLE -sourceFile "C:\SourceFiles\Nursery Rhymes.flprj"
Export Lingo Project	
Main Commands	
-appendLanguage	Used to apply a language code (e.g., "es" for Spanish) to exported non-project files, instead of applying code to folder.
	 EXAMPLE -appendLanguage true
-exportTo	Path where you want to store translated project or file after exporting from Lingo.
	 EXAMPLE -exportTo C:\Users\[username]\Desktop\Exports

-language

Language code of the target language to export.



EXAMPLE `-language es;fr`

-project

Path and name of the Lingo project to export.



EXAMPLE `-project "C:\SourceFiles\Nursery
Rhymes.flprj"`

-sync

Perform a project update. If an export path is omitted and a source file specified, an update will be performed.

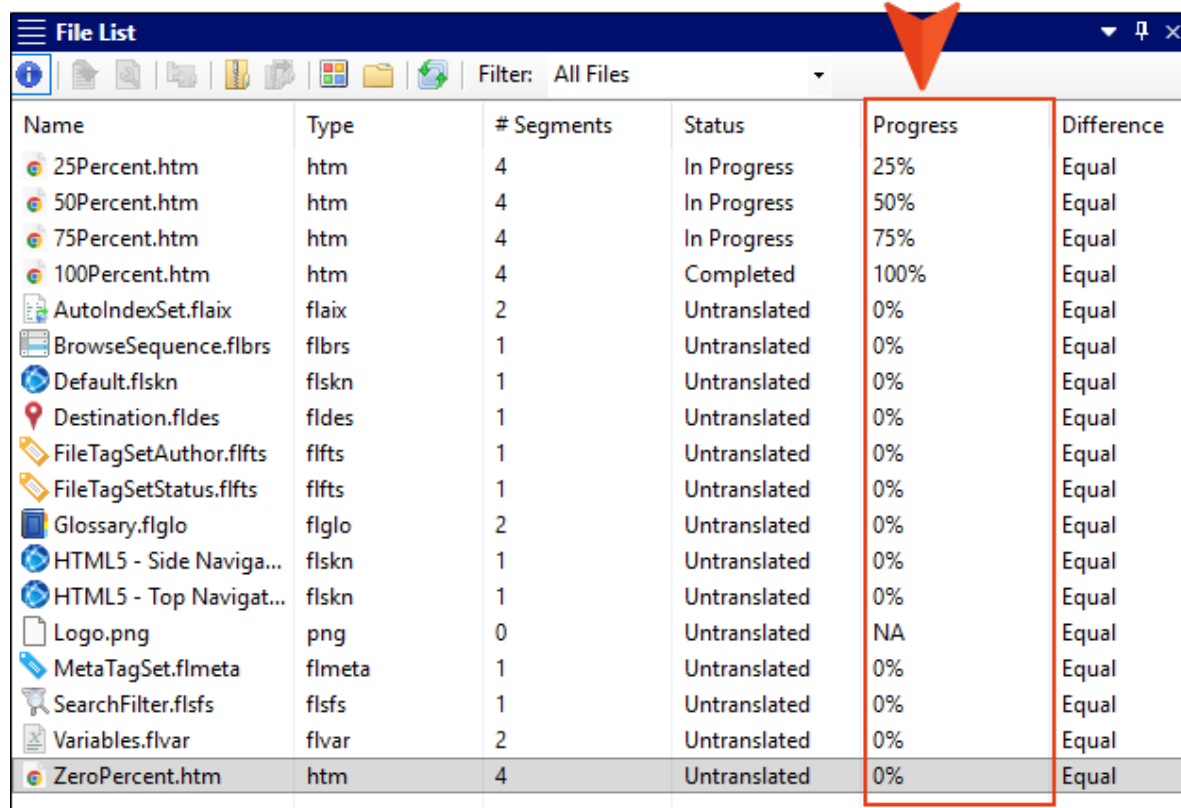


EXAMPLE `-sync true`

CHAPTER 3

File List Progress Column

There is a new Progress column shown in the File List window pane, which you can open from View > File List. Until now, translators could use this window pane to see the status of each file (e.g., In Progress, Completed, Untranslated), but they could not determine how much work was left. Now that a percentage is displayed in the Progress column, a translator can tell at a glance how far along each file is.



Name	Type	# Segments	Status	Progress	Difference
25Percent.htm	htm	4	In Progress	25%	Equal
50Percent.htm	htm	4	In Progress	50%	Equal
75Percent.htm	htm	4	In Progress	75%	Equal
100Percent.htm	htm	4	Completed	100%	Equal
AutoIndexSet.flaix	flaix	2	Untranslated	0%	Equal
BrowseSequence.flbrs	flbrs	1	Untranslated	0%	Equal
Default.flskn	flskn	1	Untranslated	0%	Equal
Destination.flides	flides	1	Untranslated	0%	Equal
FileTagSetAuthor.flfts	flfts	1	Untranslated	0%	Equal
FileTagSetStatus.flfts	flfts	1	Untranslated	0%	Equal
Glossary.flglo	flglo	2	Untranslated	0%	Equal
HTML5 - Side Naviga...	flskn	1	Untranslated	0%	Equal
HTML5 - Top Navigat...	flskn	1	Untranslated	0%	Equal
Logo.png	png	0	Untranslated	NA	Equal
MetaTagSet.flmeta	flmeta	1	Untranslated	0%	Equal
SearchFilter.flfsfs	flfsfs	1	Untranslated	0%	Equal
Variables.flvar	flvar	2	Untranslated	0%	Equal
ZeroPercent.htm	htm	4	Untranslated	0%	Equal

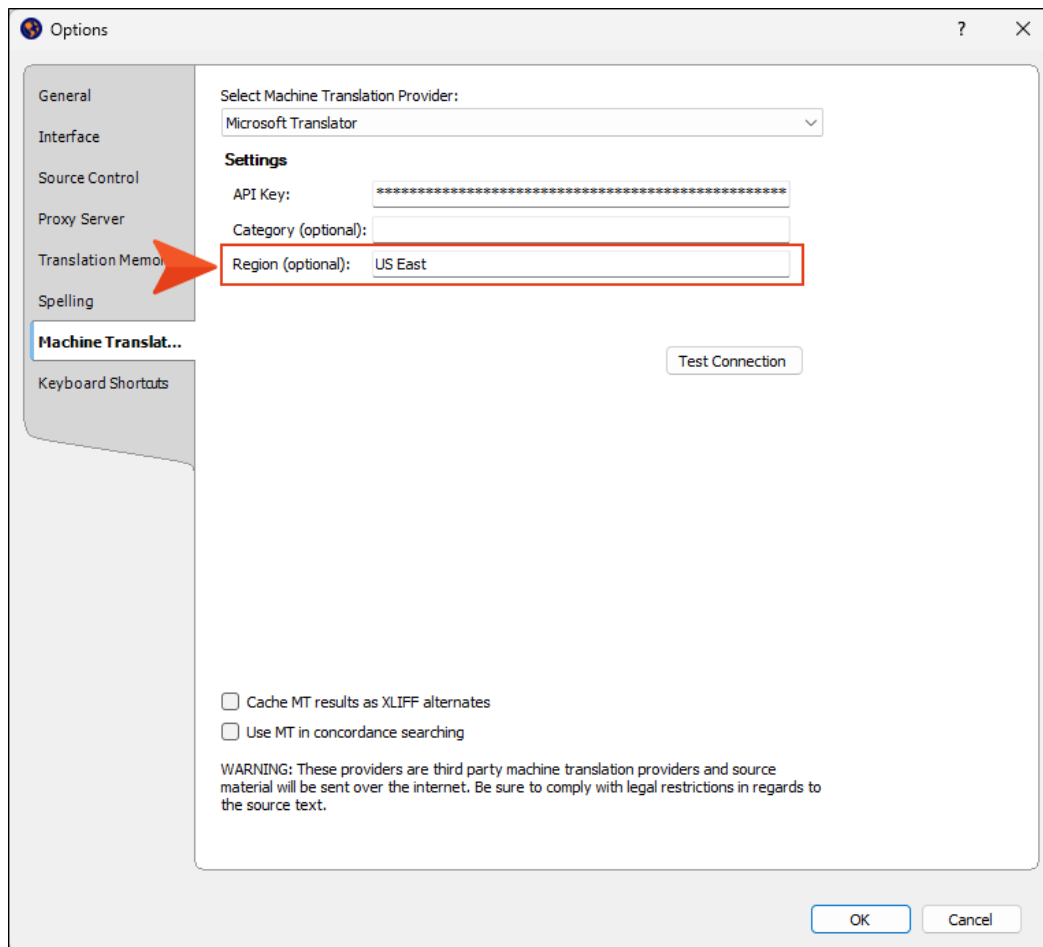


NOTE If the project was created in a version before Lingo 12, this column will display NA. However, if you update and save a file, then this column will display a percentage.

CHAPTER 4

Microsoft Translator Region

If you are using Microsoft Translator for machine translation (MT), you can now choose a specific region when doing so. This is done in the Options dialog (File > Options).

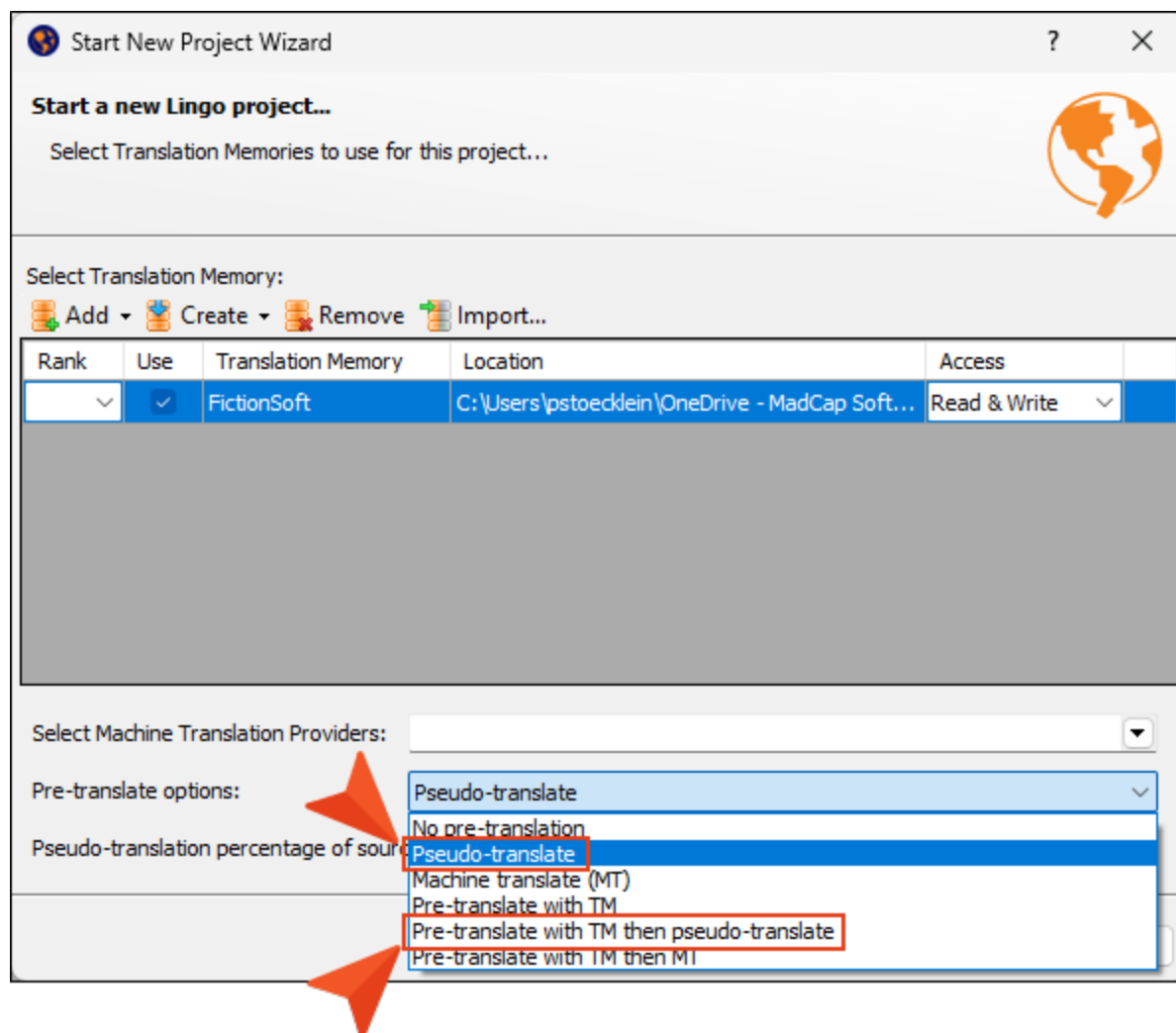


The reason for this new field is that some Microsoft Translator keys require a region, whereas others do not. If you have a key that requires a region, you need to enter it into this field for the key to be authenticated and MT to work. If you have a key that does not require a region, you can leave this field blank.

Pseudo Translate

Lingo now allows you to use pseudo translation when creating a new project. When this feature is enabled, the new project is pre-translated using random characters while preserving the tag structure. This helps you visualize content placement and layout in the final translated output, as it allows control over how much larger or smaller the target segments will be compared with the source.

You can choose a simple pseudo-translate option, or you can an alternative option where the project is pre-translated with translation memory (TM) and then is pseudo-translated.



Another field indicates how much larger or smaller the pseudo-translated version will be compared with the original content. For example, in the following image, the pseudo-translated content will be 110% the size of the content in the source language (i.e., 10% larger). In most cases, translated projects tend to be 10% to 20% larger than the source. By adjusting this value and performing a pseudo-translation, you can get an idea of how certain structural features, such as page breaks, might affect the actual translation in the end.

Start New Project Wizard

?

×

Start a new Lingo project...

Select Translation Memories to use for this project...



Select Translation Memory:

 Add

 Create

 Remove

 Import...

Rank	Use	Translation Memory	Location	Access
▼	☒	FictionSoft	C:\Users\pstoedklein\OneDrive - MadCap Soft...	Read & Write ▼

Select Machine Translation Providers:

▼

Pre-translate options:

Pseudo-translate

▼

Pseudo-translation percentage of source:

110

▲▼

< Back

Next >

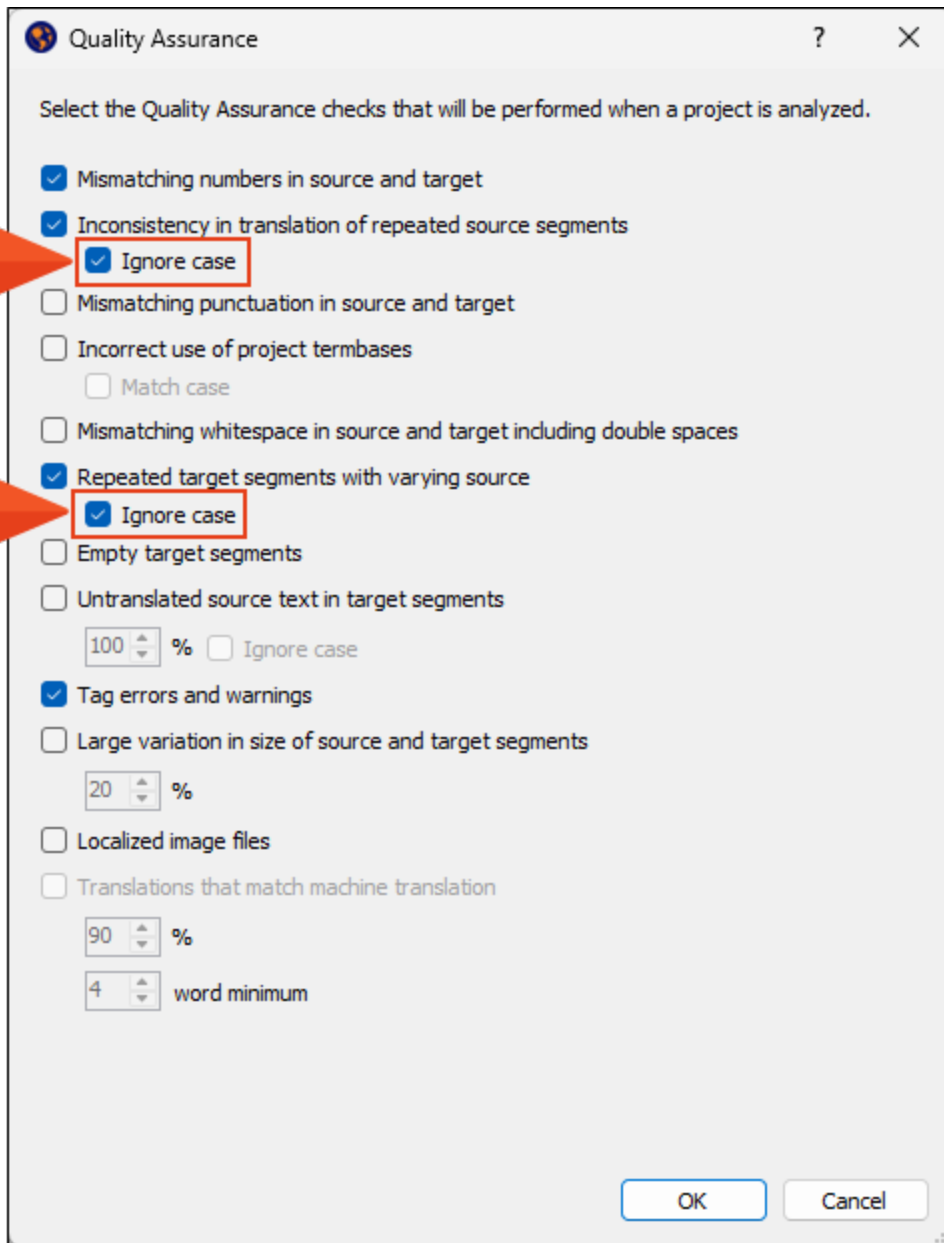
Finish

Cancel

QA Reports Ignore Case

A new option to ignore the case has been added to the following two QA reports:

- **Inconsistency in translation of repeated source segments** This report shows segments where two identical source segments do not have identical translations.
- **Repeated target segments with varying source** This report shows segments where identical target segments do not have the same source segment.

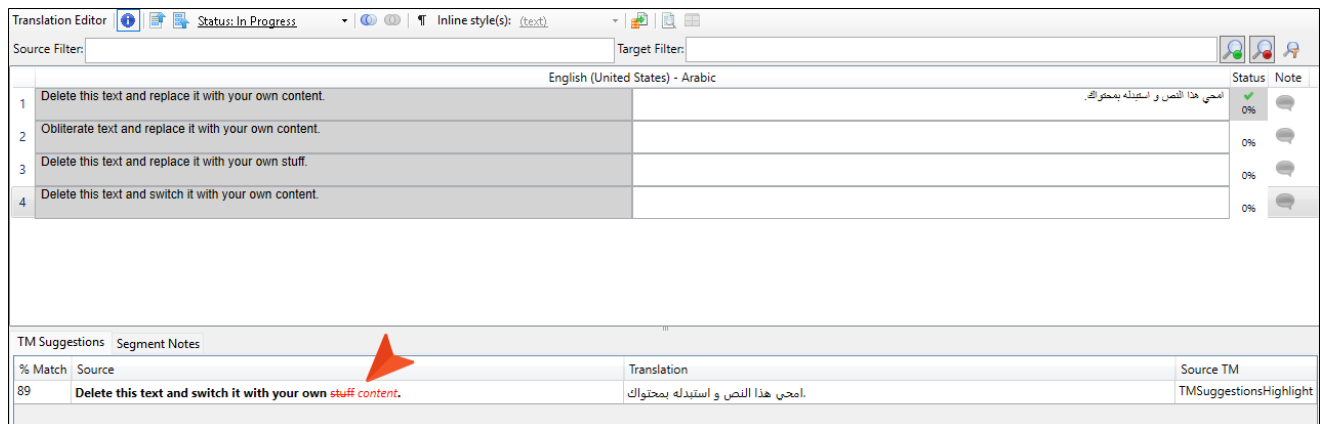


By enabling the **Ignore case** option, you might improve the results you receive from these reports.

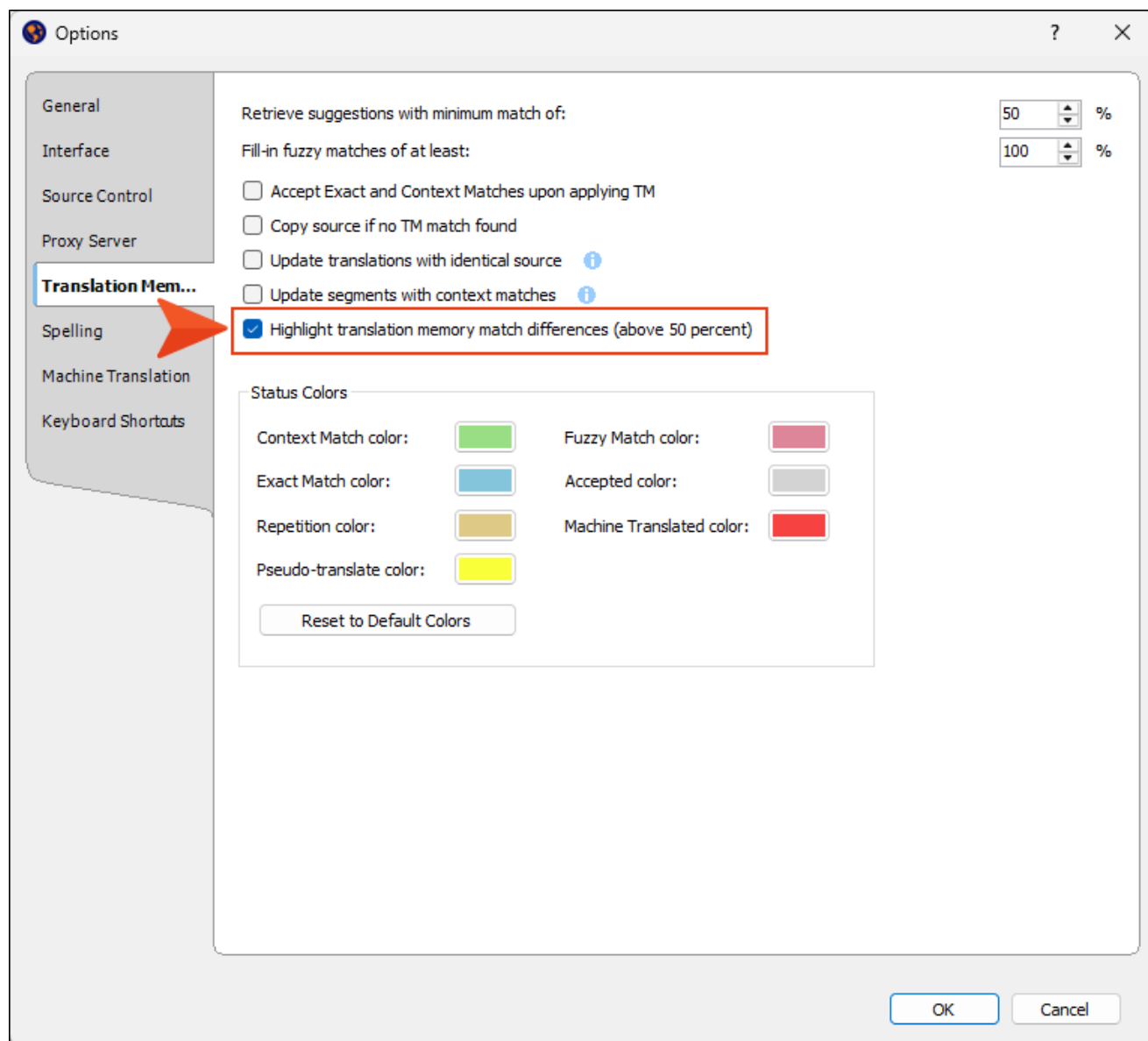
CHAPTER 7

TM Suggestion Highlight

When using translation memory (TM) suggestions, highlighting of changed content helps with usability. Text that is the same, is shown in black. Text that has been removed is in red and crossed out. Text that is new is in red and italicized.



In order for this feature to function properly, the related option must be enabled in the Options dialog (File > Options > Translation Memory). (This option is selected by default.)



APPENDIX

PDFs

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

I Cheat Sheets

Shortcuts Cheat Sheet

I User Guides

Alignment Guide

Getting Started Guide

Key Features Guide

Source Control Guide: Git

Source Control Guide: Perforce Helix Core

Source Control Guide: Subversion

Source Control Guide: Team Foundation Server

Termbases Guide

Touring the Workspace Guide

Translation Guide

What's New Guide