

ORACLE

Oracle Forms Translations

XLIFF Extract and Merge Tool

February, 2023 Version 12.2.1.19.0 Rev 0
Copyright © 2023, Oracle and/or its affiliates
Public

Purpose statement

This document provides an overview of a features and usage information related to the Forms XLIFF Extract and Merge translation tool. Additional usage information is available in the tool's built-in help text.

Table of contents

Purpose statement	2
Introduction	4
Getting Started	5
Usage Instructions	5
Using the Extraction Tool	5
Adding Translations	7
Using the Merge Tool	8
Considerations	10
Appendix	11
Extraction Usage Details	11
Merge Usage Details	12
References	13

Introduction

Oracle Forms is an application development technology with over four decades of history. It is used by companies of all sizes and can be found on most continents around the world. As such, the need for multi-language support in the applications developed with this technology is very important.

In earlier Forms releases, tools like Translation Hub and Translation Builder were included to assist with the creation of language specific versions of the applications created with Forms. Unfortunately, these tools were tightly tied to specific Oracle Database versions and were only available for Microsoft Windows 32-bit platforms. Further, the tools and output from these tools was proprietary, therefore causing the developer to be forever forced to use the provided tooling.

XLIFF (XML Localization Interchange File Format) is an XML-based text format created to standardize localization data. It specifies elements and attributes to store content extracted from various original file formats and its corresponding translation.

The Forms XLIFF Extract and Merge Tool will allow Forms application developers to extract industry standard XLIFF files from Forms application source files (FMB, MMB, and OLB). These files can be edited using most XLIFF 1.0 compatible tools or a plain text editor. Once the desired changes have been made to the XLIFF file, the same Forms tool can be used to merge the changes into a new Forms source file, which can then be used to generate new Forms executable files (FMX, MMX) as necessary.

Getting Started

Any third party product images or product references mentioned in this document are for example purposes only and are not an endorsement of that product nor will they be supported by Oracle.

The Forms XLIFF Extract and Merge Tool is a Java based utility built using the Forms JDAPI (Java Development API). The tool operates in a similar manner to the Forms XML Converter in that a file is passed into the tool and a resulting file is returned. In the case of XLIFF Extract and Merge, extracted files contain all the translatable strings from within the processed modules (FMB, MMB, and OLB). This output XLIFF file can be translated into the desired language using a plain text editor or an XLIFF 1.0 compatible third party tool. More about XLIFF tools will be covered later in this guide. Once a translated XLIFF file is available, the Merge function of the Forms tool can be used to import the translated strings back into a Forms module. For FMB and MMB files, in addition to translated versions of these module types being created by the Merge function, creating FMX and MMX files is also possible during merging. Carefully review the usage information exposed by the tool (and the [Appendix](#) of this guide) for details.

Usage Instructions

This tool can be used on any platform where Oracle Forms 12.2.1.19 is supported for use. Do not attempt to use this tool in any Forms version other than the one in which the tool was included. The tool requires the use of a certified Oracle Java 8 or newer certified version.

The *Forms XLIFF Extract and Merge Tool* includes two convenience scripts to make using the tool easier. These scripts are included in the domain and should be used to operate this tool.

- On Unix/Linux
 - frmf2xlf.sh (Forms module to xliiff)
 - frmxf2f.sh (xliiff to Forms module)
- On Microsoft Windows
 - frmf2xlf.bat (Forms module to xliiff)
 - frmxf2f.bat (xliiff to Forms module)

On Unix/Linux, the Forms XLIFF utility requires access to a running display. Be sure to properly

Warning:

Some functionality of this tool can be destructive. Be sure to create backup copies of any Forms source files (i.e. .fmb, .mmb, .olb) before using them with this tool. Deleted or altered source files are not recoverable.

Using the Extraction Tool

Step 1:

Open a shell and change directories to the directory where the convenience scripts are located. This will be in the domain, Forms instance directory. Example:

```
C:\Oracle\Middleware\user_projects\domains\base_domain\config\fmwconfig\components\FORMS\instances\forms1\bin
```

Step 2:

In this shell, set the ORACLE_HOME environment variable. The value should point to the top directory where Oracle Forms is installed. Example:

```
Windows: set ORACLE_HOME=C:\Oracle\Middleware\Oracle_Home
```

```
Unix/Linux: export ORACLE_HOME=/u01/Oracle/Middleware/Oracle_Home
```

Step 3:

This step will extract translatable strings from a Forms module and generate an .xlf file containing the translatable strings in XLIFF format.

In the open shell, run the Extraction script (frmf2x1f) without any arguments in order to expose all the available options and some examples. Alternatively, refer to the [Appendix](#) in this document.

```
Windows: frmf2x1f.bat
```

```
Unix/Linux: frmf2x1f.sh
```

Extraction example:

```
frmf2x1f.bat -o C:\Users\oracle\xliff\output *.fmb
```

In the above example, -o and its value tells the tool where to create the output XLIFF file(s). The *.fmb indicates that all files with the extension “.fmb” in the current directory should be processed. Alternatively, a single file name or list of specific file names can also be provided. The output XLIFF file(s) will be named to match the module name. The file extension will be “.xlf”. Use care in cases where multiple file formats (FMB, MMB, OLB) are stored in the same directory and may have the same name as a file of a different format. As an example, if an xlf is generated from emp.fmb but the same directory also contains a menu named emp.mmb a collision in file names could occur when the output .xlf file is created. The second one processed will overwrite the first. The Extract part of the tool does include a -f switch, which can be used when an alternative file name is desired, but only if a single file is being processed (i.e. no wildcards or list of files).

Warning:

Although the extraction process does allow the use of wildcards in file naming (e.g. *.fmb) of the input file(s), it is not recommended that wildcards be used in file extensions. Doing so could result in overwriting existing files with the same base name but different file extensions (e.g. test.fmb and test.mmb would both result in test.xlf).

Step 4:

Create backup copies of the XLIFF file(s) created by the extraction tool and store them as originals. The created files will be used to insert the desired translated strings and then later be used in the merging process to create the translated Forms module(s). Create one copy for each target language, for example bitester_sp.xlf, bitester_de.xlf, bitester_ja.xlf. Alternatively, create an xlf file that will be stored along with the translated Forms source file in a language specific directory. For example, create a directory “/sales_sp” and use this directory to contain the Forms source file and corresponding xlf file.

Adding Translations

Although an XLIFF editing tool is not required to edit the XLIFF files, using one is recommended. An XLIFF editor is not provided with this tool. Any third party XLIFF editor that supports XLIFF v1.0 can be used with files created by the Forms XLIFF Extract and Merge Tool. There are numerous XLIFF editors available [online](#). Some are free to use, whereas others may have cost associated with their use. A plain text editor can be used, but is not recommended. Editing the files manually may result in unintentionally introducing invalid entries thereby causing the merge function to fail or not properly create the new Forms module(s).

Adding Translations with Third Party Editors

Using a third party XLIFF translation editor, use the created XLIFF files from Step 4 of the extraction procedure to insert any desired translation strings. The translated XLIFF file(s) should now be language specific. As mentioned in Step 4 of the extraction, it is recommended that the translated file(s) be saved with a name that includes reference to the language it represents and/or stored in a directory that clearly indicates the language of the files it contains. For example if the module name is “bitester” and the translation was to Spanish, name the XLIFF file bitester_sp.xlf.

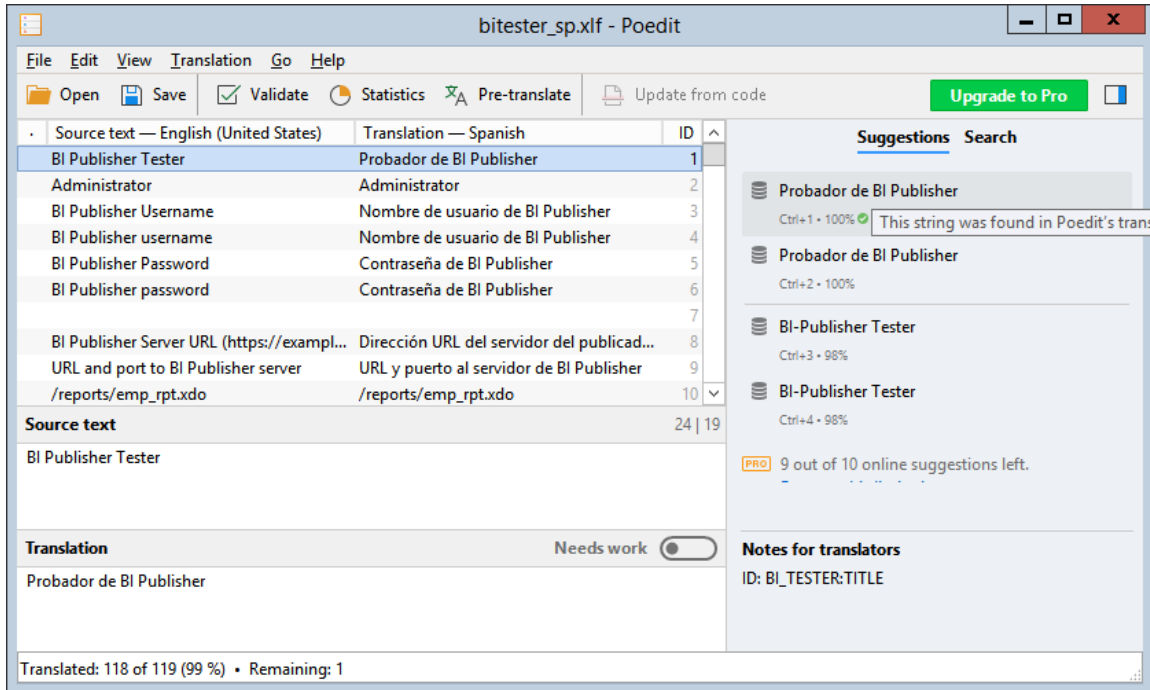


Figure 1 - “bitester.fmb” XLIFF file being translated to Spanish with Poedit (third party editor)

Adding Translations Manually

If using a third party XLIFF editor is not possible, the extracted XLIFF file(s) can be edited in a plain text editor, although doing so is not recommended.

Open the created .xlf file in a plain text editor (do not use Microsoft Word or any other editor that may introduces formatting). Edit the “target-language” value using the appropriate language code. Then edit the strings between the “<target> ... </target>” tags to the desired new language text. Repeat this for each object requiring a language change. Do not alter any other contents in the file. When all the desired changes have been made to the file, save the changes.

English to Spanish Example:

Before:

```
...
<file datatype="fmb" original="bitester.fmb" source-language="en-US" target-language="en-US">
...
<source>BI Publisher Username</source>
<target>BI Publisher Username</target>
```

After:

```
...
<file datatype="fmb" original="bitester.fmb" source-language="en-US" target-language="es">
...
<source>BI Publisher Username</source>
<target>Nombre de usuario de BI Publisher</target>
```

Using the Merge Tool

Step 1:

Using the updated XLIFF file(s) that contain the desired translations created above, use the merge tool to create new Forms module(s). Be sure to backup copies of the original Forms source files have been created and safely stored before continuing.

If not using the same shell that was used previously with the Extraction task, be sure the shell being used has ORACLE_HOME properly set and the current directory is the directory where the tool scripts are being stored or the scripts are included in PATH.

In the open shell, run the Merge script (frmX1f2f) without any arguments in order to expose all the available options and some examples. Alternatively, refer to the [Appendix](#) in this document.

Windows: frmX1f2f.bat

Unix/Linux: frmX1f2f.sh

Merge example:

```
frmxl2f.bat -k -o -t C:\Users\oracle\xliff\output\SP -F bitester_sp.xlf -m  
C:\Users\oracle\xliff\output\tmp C:\Users\oracle\xliff\output\bitester.fmb
```

In the above example:

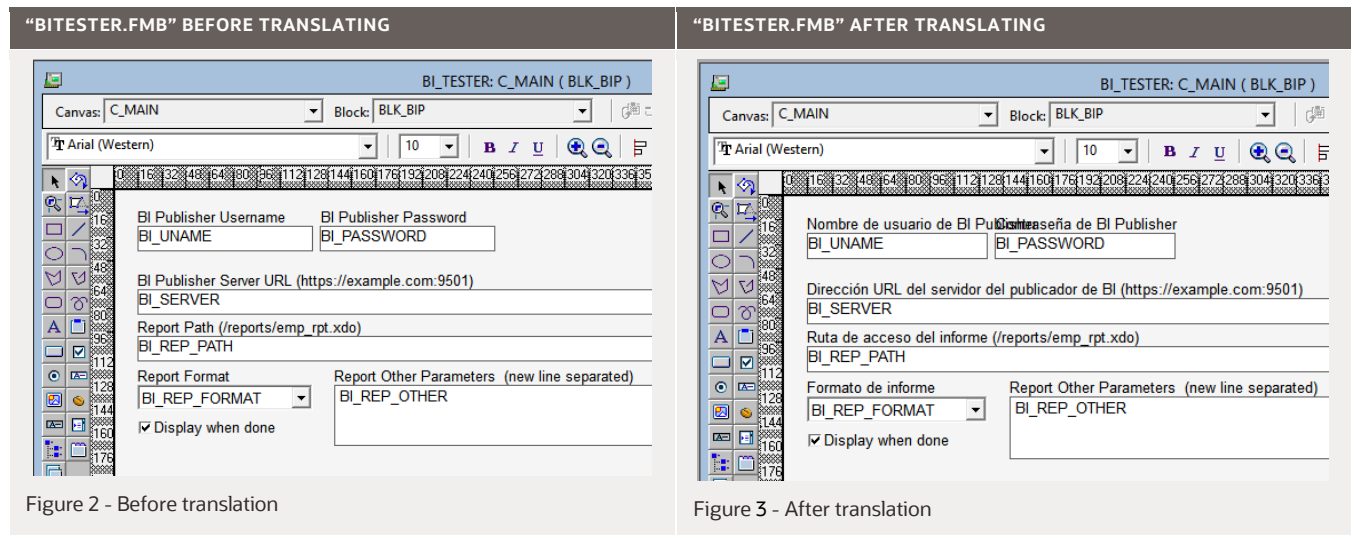
- -k indicates that after processing completes, do not delete the source file (fmb, mmb, olb) generated by this call. The -k is needed when not using -o. Using both -k and -o would create a new source module but not create a corresponding executable module (e.g. fmx, mmx).
- -o indicates that you do not want a Forms executable file (fmx, mmx) generated from the translated source file that will be created.
- -t indicates the directory where the translated XLIFF file is stored. In this example, a "SP" (Spanish) sub-directory was created and used to store the translations that were created for Spanish.
- -F refers to the specific XLIFF file, by name that should be used for this operation.
- -m is the "merge" or temporary directory used to perform the operation. This directory is where the output FMB will be created. The last entry is the source module (FMB) that is used along with the merge operation to create the eventual output module.
- The path and file name at the end is the input file (original) used as the basis to create the translated module. This should be the same file used to perform the extraction.

The use of Wildcard symbols or listing more than one module name can be used to specify the Forms source modules with the merge operation. However this is not recommended.

Step 2:

Open the resulting module or run the resulting executable module (if one was created) and verify that all the expected translations were created.

Here are the before and after examples in the Form Builder of the module used in the above example.



Considerations

1. Because the tool is capable of creating and deleting/overwriting files, be sure to create backup copies of all files being used during the process. This should include original Forms source files (e.g. fmb, mmb, olb), extracted XLIFF files created by the extraction tool, and XLIFF files that contain the new language specific translation strings.
2. By using a third party XLIFF Translation Editor, spelling errors and accidentally introducing invalid formatting of the XLIFF file are less likely to occur.
3. Using the DTD validation option is recommended when manually editing XLIFF files. The -d (lower case) option must be used when extracting the original XLIFF file if performing the DTD validation is planned when merging the changes occurs. The -D (upper case) is used during merging to perform DTD validation.
4. The Forms XLIFF Extract and Merge Tool is based on XLIFF v1.0. Pay close attention to this when choosing a third party editor. Using a third party editor that is not compatible with XLIFF v1.0 may result in the generation of incompatible XLIFF files. As a result, attempts to merge and create a new Forms module may fail.

Appendix

Extraction Usage Details

If running the tool through the provided `frmf2xlf` script, substitute the name of the script for "Extract" in the below examples.

Command line arguments:

Flag	Meaning	XLIFF Property
----	-----	-----
-d	Include DTD information	n/a
-f	Output XLIFF filename	<code>xliff.extractFilename</code>
-o	Output directory path	<code>xliff.extractOutputPath</code>
-p	Properties file	n/a
-h	HyperHub preview support	n/a
-v	Display version and help	n/a

e.g. : `Extract -o translations/US form1.fmb form2.fmb`
-> Extract XLIFF documents from `form1` and `form2` and saves them in `translations/US` directory.

e.g. : `Extract -p xliff.props -f test1.xlf form1.fmb`
-> Extract an XLIFF document called `test1.xlf` from `form1` and saves it using the properties set in property file "`xliff.props`".

e.g. : `Extract -p xliff.props form1.fmb`
-> Extract an XLIFF document from `form1` and saves it using the properties set in property file "`xliff.props`".

e.g. : `Extract -v`
-> Display tool version number

Using the `-h` flag will cause the resulting XLIFF file to contain additional non-translatable information which is needed if the XLIFF document will be used with the HyperHub previewer.

`NLS_LANG` is specified as a system property using the `-D` Java syntax so that the `NLS_LANG` environment setting can be picked up by the script '`frmf2xlf`'. `NLS_LANG` is used to set the source and target language attributes in the xliiff file. If `NLS_LANG` is not set then the default is '`american_america.we8iso8859p1`'

Merge Usage Details

If you are running the tool through the provided frmxlf2f script, substitute the name of the script for "Merge" in the below examples.

Command line arguments :

Flag	Meaning	XLIFF Property
-a	Pre-merge directory	xliff.preMergeDirectory
-c	Database connection string	xliff.connectionString
-d	Skip dependency analysis	n/a
-f	Forms path	xliff.formsPath
-k	Keep FMB/MMB files	n/a
-l	Log merge operations	n/a
-m	Merge (output) directory	xliff.mergeDirectory
-o	Don't create FMX/MMX file	n/a
-p	Properties file	n/a
-t	Translation path	xliff.translationPath
-v	Display tool version number	n/a
-x	Specify FMX/MMX filename (for single file operations only)	xliff.fmxFilename
-D	Include DTD validation	n/a
-F	XLIFF filename (for single file operations only)	xliff.mergeFilename

e.g. : Merge -t trans/FR;trans/US -p xliff.properties myform.fmb
-> Merge myform.fmb and dependencies with a French translation using the properties defined in "xliff.properties"

e.g. : Merge -v
-> Display tool version number

Note that filename used in -x will also be used for the FMB/MMB if -k is specified. Additionally, -x operates by renaming the file(s) after they are written (and thus files with the original name in the destination directory will still be overwritten in this case).

Note that if the XLIFF file does not contain DTD information then using the -D option will fail.

References

Oracle Forms

<https://www.oracle.com/application-development/technologies/forms/forms.html>

XLIFF 1.0 Specification

<http://www.oasis-open.org/committees/xliff/documents/contribution-xliff-20010530.htm>

Connect with us

Call **+1.800.ORACLE1** or visit **oracle.com**. Outside North America, find your local office at: **oracle.com/contact**.

 blogs.oracle.com

 facebook.com/oracle

 twitter.com/oracle

Copyright © 2023, Oracle and/or its affiliates. All rights reserved. This document is provided for information purposes only, and the contents hereof are subject to change without notice. This document is not warranted to be error-free, nor subject to any other warranties or conditions, whether expressed orally or implied in law, including implied warranties and conditions of merchantability or fitness for a particular purpose. We specifically disclaim any liability with respect to this document, and no contractual obligations are formed either directly or indirectly by this document. This document may not be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without our prior written permission.

This device has not been authorized as required by the rules of the Federal Communications Commission. This device is not, and may not be, offered for sale or lease, or sold or leased, until authorization is obtained.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group. 0120

Disclaimer: If you are unsure whether your data sheet needs a disclaimer, read the revenue recognition policy. If you have further questions about your content and the disclaimer requirements, e-mail REVREC_US@oracle.com.
