

Bible Online Learner: Localization Guide

Claus Tøndering
Ezer IT Consulting

5 October 2015

Copyright © 2015 by Claus Tøndering, claus@ezer.dk

The document is made available under a Creative Commons Attribution 4.0 International License
(see <http://creativecommons.org/licenses/by/4.0/>)

Contents

Contents	2
1 Introduction	3
1.1 Requirements	3
2 Editing PHP Files	4
2.1 Special Characters	4
2.2 Email Text	5
2.3 Overview of the PHP Files	6
3 Editing JSON Files	7
3.1 Sorting Information	7
3.2 Overview of the JSON Files	8
4 Final Actions	9

Introduction

This document tells you how to create a translation of the user interface of Bible Online Learner (Bible OL).

You are expected to be familiar with the way Bible OL is used, both by teachers and by students.

1.1 Requirements

Before you start, you must have a copy of the following files:

```
class_lang.php
common_lang.php
ETCBC4.en.prop.json
ETCBC4-translit.en.prop.json
file_manager_lang.php
font_lang.php
form_validation_lang.php
intro_text_lang.php
js_lang.php
login_lang.php
menu_lang.php
owner_lang.php
nestle1904.en.prop.json
privacy_lang.php
shebanq_lang.php
statistics_lang.php
text_lang.php
userclass_lang.php
users_lang.php
```

These files are so-called “flat” text files. This means that they contain no formatting. When you edit them, you must use a proper text editor, not a word processor such as Microsoft Word. Many text editors are available. On computers running Windows, the *Notepad* program can be used, although it has an annoying tendency to add “.txt” to the end of file names.

It is preferable that your text editor can read and write text using the UTF-8 encoding. In Windows Notepad you can select the text encoding in the “Save As” dialog.

Editing PHP Files

Most of the files have file names that end in “.php”. These are files that provide translations for the parts of the system that are not directly linked to a Greek or Hebrew database.

A typical entry in one of the PHP files looks like this:

```
$lang['file_exists_overwrite'] = 'The file already exists. Do you want to replace it?';
```

When making your translation, you must not change anything to the left of the equals (=) sign. The text to the right of the equals sign is the translation. As shown above, it must be enclosed in single quotation marks (') and end with a semicolon.

So if you are making a translation into French, you should change the above line to something like:

```
$lang['file_exists_overwrite'] = 'Le fichier déjà existe. Voulez-vous le remplacer?';
```

You are allowed to break the lines and insert extra spaces if you need to; it makes no difference to the displayed text. So the following entry would also be fine:

```
$lang['file_exists_overwrite'] = 'Le fichier déjà existe.  
Voulez-vous le remplacer?';
```

There is, however, a few cases where line breaks and spaces must not be inserted. This is described in Section 2.2.

In some cases the translation is followed by the characters // and some text. This is a comment that may provide you with additional information. You need not translate it. For example:

```
$lang['user_profile_deleted'] = 'User Profile Deleted'; // Text in title bar
```

Here the comment “Text in title bar” indicates the reason why the English words are capitalized. This is irrelevant in languages that do not use upper case in titles.

2.1 Special Characters

Punctuation Marks

If the English string contains punctuation marks – in particular if it ends with a colon – you should include similar punctuation in your own language if it makes sense. Be particularly conscious about quotation mark styles, which can vary greatly from language to language:

English: “abc” or ‘abc’
German: »abc« or „abc“
French: « abc »

But regardless of your language, the translated string must be enclosed in single quotation marks (') and end with a semicolon (;).

Apostrophe or Quotation Mark

If you need to include the character ' in the text, you must precede it with the character \. This tells the system that the ' does not end the string. For example:

```
$lang['dont_care'] = 'Don\'t care';
```

HTML Code

In a few places the translated string may contain HTML commands. For example:

```
$lang['select_number_preset'] = 'Select number of questions<br>using preset passages';
```

HTML commands are enclosed in < and > characters. You should not change them in the translation.

%s, %d, and {0}

In some cases the system needs to insert a value into the text string. This may, for example, be the name of a file. Such values are indicated by %s, %d, or {0} in the text strings. For example:

```
$lang['user_name_used'] = 'The user name "%s" is already in use';
```

or

```
$lang['use_qo_selection'] = 'Do you also wish to use {0} for sentence unit selection?';
```

Be sure to include the %s, %d, or {0} in the translated string to indicate where the system should insert a value.

2.2 Email Text

A few of the text strings are not displayed on web pages but are sent to users in emails. The rules for writing these translations are a little different:

1. The strings are delimited by double quotes (") rather single quotes (').
2. If you need to break a line in the PHP file, you must end the first line with a double quote and start the second line with a period followed by a double quote.
3. If you want a line break in the email you must write the string \n at the place where the line break is needed.

The following example illustrates these three rules:

```
$lang['email_text'] = "This is the text of an "
                    . "email.\nThis is on a "
                    . "new line.\n";
```

This will cause the actual text of the email to be:

```
This is the text of an email.
This is on a new line.
```

(Strictly speaking, the reason double quotes are used instead of single quotes is that \n does not work within single quotes.)

2.3 Overview of the PHP Files

The following table lists the name of each PHP file and gives a little information about which parts of the program use the translations in each file.

File	Examples of use
<code>class_lang.php</code>	Class management under the “Classes” menu item.
<code>common_lang.php</code>	Strings used in many different parts of the system.
<code>file_manager_lang.php</code>	Exercise management under the “Manage exercises” menu item.
<code>font_lang.php</code>	Font management under the “Font preferences” menu item.
<code>form_validation_lang.php</code>	Error messages for data input forms.
<code>intro_text_lang.php</code>	Front page text.
<code>js_lang.php</code>	Text display, exercise editing, exercise execution.
<code>login_lang.php</code>	Login mechanism.
<code>menu_lang.php</code>	The menus.
<code>owner_lang.php</code>	Ownership of exercises and classes.
<code>privacy_lang.php</code>	Privacy statement.
<code>shebanq_lang.php</code>	Strings used when importing queries from SHEBANQ.
<code>statistics_lang.php</code>	Statistics display under the “Statistics” menu item.
<code>text_lang.php</code>	Text and exercise display.
<code>userclass_lang.php</code>	Management of a user’s membership of a class.
<code>users_lang.php</code>	User management under the “Users” menu item.

The listed examples of use are not exhaustive.

Editing JSON Files

A few of the files have file names that end in “.json”. These are files that provide translations for the grammatical terms that relate directly to the Hebrew and Greek databases. The format of these files is completely different from the format of the PHP files.

A typical section in one of the JSON files might look like this:

```
"part_of_speech_t": {  
  "prps": "Personal pronoun",  
  "verb": "Verb",  
  "adjv": "Adjective"  
},
```

When making your translation, you must not change anything to the left of a colon (:). The text to the right of the colon is the translation. As shown above, it must be enclosed in double quotation marks (") and end with a comma. If, however, the following line starts with a }, the comma *must* be omitted, as illustrated with the string "Adjective" above.

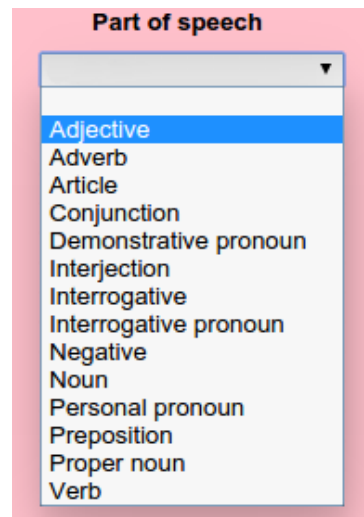
So if you are making a translation into French, you should change the above lines to something like:

```
"part_of_speech_t": {  
  "prps": "Pronom personnel",  
  "verb": "Verbe",  
  "adjv": "Adjectif"  
},
```

3.1 Sorting Information

In the above example the values of the type *part_of_speech_t* have English translations such as “Personal pronoun”, “Verb”, and “Adjective” (and several others that were omitted above).

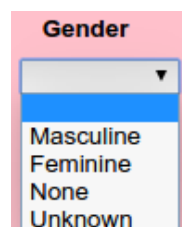
When these values are displayed by the system, they are normally sorted alphabetically:



But sometimes you want another sorting order. If the translation starts with “#1”, “#2” etc. these numbers indicate the sort order. For example:

```
"gender_t": {
  "f": "#2 Feminine",
  "m": "#1 Masculine",
  "NA": "#3 None",
  "unknown": "#4 Unknown"
},
```

Here, the strings “#1”, “#2”, etc. indicate the sorting order. Thus:



Be aware that the preferred sorting order in English may not be the preferred sorting order in your language.

3.2 Overview of the JSON Files

The following table lists the JSON files.

File	Use
ETCBC4.en.prop.json	Grammatical terms for the Hebrew Old Testament.
ETCBC4-translit.en.prop.json	Grammatical terms for the transliterated version of Hebrew Old Testament. (The contents of this file is very similar to the preceding file.)
nestle1904.en.prop.json	Grammatical terms for the Greek New Testament.

Final Actions

When you have edited all the files, you must send them to me at claus@ezer.dk. I will then enter them into the system and you can verify that they are correct. Unfortunately, there is no way for you to see the result before I have installed the files.

Please be aware that Bible OL is a dynamic program whose features change frequently. As new features are added to the system, I will contact you and ask you to translate text for the new features.