

Community Builder Language Package Guide

Updated for CB 1.2.3

Introduction

This document has been created to assist people in creating CB Language plugins.

Overview

CB 1.2.3 can “speak” different languages with the assistance of CB Language plugins. A CB Language plugin is a special kind of CB plugin. Your CB 1.2.3 distribution package contains a built-in default language plugin with English language strings.

CB ‘queries’ the CMS system regarding which is the ‘active’ language. In Joomla 1.5.X, the website administrator has installed one or more ‘Language’ extensions (Joomla 1.5.X: Extensions→Language Manager) and has selected one as the default language. In Joomla 1.5.X, users are permitted to select their preferred language assuming of course that the website administrator has installed additional Joomla 1.5.X languages. CB 1.2.3 also supports this user-specific language personalization allowing users to select their language preference via a drop-down language select field when editing their CB profile. JoomFish component can also be used for front-end language changing and CB will support this also.

To assist translators in the CB Language plugin creation process, the CB Team has created an **english_cb123_utf8.zip** package as a CB 1.2.3 English Language plugin. This package when decompressed will reveal an ‘english’ folder with the following contents:

- english (folder)
 - index.html
 - english.xml
 - english.php
 - admin_language.php
 - calendar-local.js
 - cbteampugins_language.php
 - images (folder)
 - index.html
 - nophoto.jpg
 - pendphoto.jpg
 - tnnophoto.jpg
 - tnpndphoto.jpg

Translation Process

The process is relatively easy and can be summarized in the following 12 steps:

1. setting up your language folder template based on English package
2. finding what language name to use
3. renaming folder
4. renaming english.xml file
5. renaming the english.php file
6. editing renamed xml file

7. editing-translating renamed php file
8. editing-translating calendar-local.js
9. editing-translating cbteampugins_language.php
10. editing-translating admin_language.php
11. packaging
12. testing

General Notes:

Most of the language plugins will most likely be used on Joomla 1.5.X installations. This means that they must be utf-8 ready. All editing should be performed with an editor that is capable of encoding utf-8 characters and all basis files should be opened and saved using the "UTF-8 without BOM characters" character-set. Windows notepad++ free editor supports this.

Also please take care of the following cases:

- in a 'single-quoted string' using a ' should be like this: 'today, it\'s nice outside'
- for \ use \\
- in "a double-quoted string", using a " should be \"
- newlines \n and returns \r can be used only in "double-quoted strings"
- single-quoted strings 'single' and "double-quoted" are equivalent in PHP, however, single-quoted are faster and use less memory since they don't allow for PHP variables. Thus if you do not need to use newlines and returns, we recommend single-quoting.
- in double-quoted strings \$ should also be escaped with \\$
- strings like %s is transformed by CB to a variable string – so %6\$s means the 6th variable string of that string; so by translating e.g.:
'All %s of user %s' into 'User %2\$s \'s all %1\$s'
a string like "All books of user Joe" would become "User Joe 's all books"
that might be useful for languages reversing order of sentences
- %s means string, %d decimal
more information can be found here:
<http://us.php.net/manual/en/function.sprintf.php>
www.php.net/sprintf

Generating a CB Language plugin for another character-set, as e.g. the ISO-8859-1 charset used in most Joomla 1.0.X CMS or a Mambo CMS is easy: Just open the UTF-8 file language.php and in UTF-8 mode, and save it using the ISO-8859-1 (Western Latin-1) characters-set. The XML file, the calendar-local.js file and the cbteampugins_language.php must remain utf-8 encoded and always saved without BOM characters

We recommend generating and publishing both versions (UTF-8 and ISO) of your CB language plugin.

Step 1: Setting up your language folder template

Unpack the english_cb123_utf8.zip package to reveal the English folder with the contents outlined in the Overview section.

This is your starting point – you will be modifying this setup to create your language package.

- english (folder)
 - index.html

- english.xml
- english.php
- admin_language.php
- calendar-local.js
- cbteampugins_language.php
- images (folder)
 - index.html
 - nophoto.jpg
 - pendphoto.jpg
 - tnnophoto.jpg
 - tnpndphoto.jpg

Step 2: Finding what language name to use

In order for CB to be able to trigger off of the Joomla active language setting, we must make sure that the proper language name is selected. To find the appropriate name we need to locate the Joomla xml language file located in the Joomla Language folder (located in your Joomla root folder on your website). For default (English-GB) Joomla 1.5.X installations you should be able to locate a folder named **en-GB** in your language folder off your website root location. If you have installed an extra Joomla language pack, you should see an additional folder (e.g., **el-GR** if Joomla Greek Language pack is installed). Your Joomla language folder (e.g., **el-GR**), will contain many .ini files but only two .xml files. You need to locate the one that matches your language folder name (e.g., **el-GR.xml**) and open it with your text editor and locate the **<backwardLang>XXXXX </backwardLang>** xml tag line. The content XXXXX of this tag is the language name you must use throughout your CB Language plugin (e.g., **greek**).

Following this process you should be able to easily locate the proper language name you need to use. Test this process out – for the en-GB installed Joomla language you should be able to identify that the CB language name is **english**.

Step 3: Rename your template folder

Now that we have locate the language name we should be using we should rename the ‘english’ folder we created in step 1 to our language name (e.g., greek). So, if we are constructing a Greek CB Language plugin our template area should now look like:

- **greek** (folder)
 - index.html
 - english.xml
 - english.php
 - calendar-local.js
 - cbteampugins_language.php
 - admin_language.php
 - images (folder)
 - index.html
 - nophoto.jpg
 - pendphoto.jpg
 - tnnophoto.jpg
 - tnpndphoto.jpg

Step 4: Renaming your xml file

Rename your template english.xml file using your language name. So, if we are constructing a Greek CB Language plugin our template area should now look like:

- **greek** (folder)
 - index.html
 - **greek.xml**
 - english.php
 - calendar-local.js
 - cbteampugins_language.php
 - admin_language.php
 - images (folder)
 - index.html
 - nophoto.jpg
 - pendphoto.jpg
 - tnnophoto.jpg
 - tnpndphoto.jpg

Step 5: Renaming your php file

Rename your template english.php file using your language name. So, if we are constructing a Greek CB Language plugin our template area should now look like:

- **greek** (folder)
 - index.html
 - **greek.xml**
 - **greek.php**
 - calendar-local.js
 - cbteampugins_language.php
 - admin_language.php
 - images (folder)
 - index.html
 - nophoto.jpg
 - pendphoto.jpg
 - tnnophoto.jpg
 - tnpndphoto.jpg

Step 6: Editing your language.xml file

Open your renamed language xml file (see step 4) with your preferred text editor and change all english words to your language name. All green highlighted words must be changed and you can also optionally edit all yellow highlighted text.

```
<?xml version="1.0" encoding="utf-8"?>
<cbinstall version="1.0" type="plugin" group="language">
  <name>english</name>
  <author>MamboJoe and Beat</author>
  <creationDate>2009-01-24</creationDate>
  <copyright>(C) 2005 - 2009 joomlapolis.com</copyright>
  <license>http://www.gnu.org/licenses/old-licenses/gpl-2.0.html GNU/GPL version 2</license>
  <authorEmail>beat@joomlapolis.com</authorEmail>
```

```
<authorUrl>www.joomlapolis.com</authorUrl>
<version>1.2</version>
<description>Provides the English language for Community Builder core functions.</description>
<files>
    <filename plugin="english">english.php</filename>
    <filename>index.html</filename>
    <filename>admin_language.php</filename>
    <filename>cbteampugins_language.php</filename>
    <filename>calendar-locals.js</filename>
    <filename>images/index.html</filename>
    <filename>images/nophoto.jpg</filename>
    <filename>images/pendphoto.jpg</filename>
    <filename>images/tnnophoto.jpg</filename>
    <filename>images/tnpendphoto.jpg</filename>
</files>
<params>
</params>
    <database>
</database>
</cbinstall>
```

Save the updated file. Make sure that your editor does not create any backup file – if it does remove it. This file should be saved as utf-8 encoded without BOM characters.

Step 7: Editing/translating our **language.php** file

Open your renamed **language.php** file (e.g., **greek.php** - see step 5) with your preferred text editor and change all DEFINE strings to your language. All green highlighted words must be changed. All DEFINE strings must be translated to have a complete translation. Be careful to edit only the English strings and take care to leave the surround quotes ‘ ‘ intact.

```
...
//Field Labels
DEFINE('_UE_HITS','Hits');
DEFINE('_UE_USERNAME','Username');
DEFINE('_UE_Address','Address');
DEFINE('_UE_City','City');
DEFINE('_UE_State','State');
DEFINE('_UE_PHONE','Phone #');
...
```

Save the updated file. Make sure that your editor does not create any backup file – if it does remove it. If you are building this CB plugin for Joomla 1.5.X sites, then you should encode this file as utf-8 encoded without BOM characters.

Step 8: Editing/translating the calendar-local.js file

Open your calendar-local.js file with your preferred text editor and translate all the calendar language strings to your language. All green highlighted words must be changed. All double quoted strings

must be translated to have a complete translation. Be careful to edit only the English strings and take care to leave the surround double quotes "" intact.

```
...
// full day names
Calendar._DN = new Array
("Sunday",
 "Monday",
 "Tuesday",
 "Wednesday",
 "Thursday",
 "Friday",
 "Saturday",
 "Sunday");
...
```

Save the updated file. Make sure that your editor does not create any backup file – if it does remove it. This file should be saved as utf-8 encoding without BOM characters.

Step 9: Editing-translating the cbteampugins_language.php

Open your cbteampugins_language.php file with your preferred text editor and translate all the language strings to your language. All green highlighted words must be changed. Only the right set of words must be translated. Be careful to edit only the English part of the strings (leave any %s codes intact) and take care to leave the surround single quotes " intact.

Once again be careful – this file is different than the previous Step 7 file. As you see it appears that the same thing is duplicated but only the second instance needs translation.

```
...
// 1.2 Stable:
// ProfileBook plugin: (new method: UTF8 encoding here):
CBTxt::addStrings( array(
'Profile Book' => 'Profile Book',
'Name' => 'Name',
'Entry' => 'Entry',
'Profile Book Description' => 'Profile Book Description',
'Created On: %s' => 'Created On: %s',
...

```

Save the updated file. Make sure that your editor does not create any backup file – if it does remove it. This file should be saved as utf-8 encoding without BOM characters.

Step 10: Editing-translating the admin_language.php

Open your admin_language.php file with your preferred text editor and translate all the language strings to your language. All green highlighted words must be changed. Only the right set of words

must be translated. Be careful to edit only the English part of the strings (leave any %s codes intact) and take care to leave the surrounding single quotes " intact.

```
...
CBTxt::addStrings( array(

// XML files:

// cb.authortab.xml
'Provides an User Tab that shows all articles written by the user.' => 'Provides an User Tab that shows all
articles written by the user.',

// cb.connections.xml
// some also used in other files (php and xml)
'Provides CB Core Connections functionality' => 'Provides CB Core Connections functionality',
'Display Settings' => 'Display Settings',
'User Profile Status :' => 'User Profile Status :',
'see Field: Connections: Parameters' => 'see Field: Connections: Parameters',
'Connections Status Settings :' => 'Connections Status Settings :',

...

```

Step 11: Packaging

You are now finished and you can package your plugin for testing. Just compress your language folder (e.g., greek) to a compressed file named language_cb123_encoding.zip (example: greek_cb123_utf8.zip or greek_cb123_iso.zip).

Step 12: Testing

Try and install your CB Language Plugin package (see step 11) using the CB Plugin management backend. See that your language does in fact translate CB based on the active Joomla language.

Your language package is complete!

Please upload to our JoomlaPolis Directory language category area for others to use. Also consider joining the CB Language WorkGroup to officially support your language translation from within – contact nant on JoomlaPolis to join!
