



# Anleitung Übersetzung

Eine neue Sprache hinzufügen



## Datum

30. September 2016

## Version

5.0

## Inhaltsverzeichnis

1. Einleitung.....	3
2. Kurzbeschreibung.....	3
3. Vorbereitungen.....	3
4. Texte übersetzen.....	4
4.1 Übersetzung mit dem Lokalisierungseditor.....	4
4.2 Übersetzung direkt in der Übersetzungsdatei.....	4
5. Länder und Regionen.....	5
5.1 Ländernamen.....	5
5.2 Regionen von Ländern.....	5
6. Buttons übersetzen.....	5
7. Willkommenseite (in HTML) anpassen.....	6
8. Optional - HTML-Templates anpassen.....	6
9. Text-Templates anpassen.....	6
10. Das neue Sprachset in Betrieb nehmen.....	6
11. Ausschliesslich diese Sprache verwenden.....	6
12. Suchmaschinenoptimierungstags des PhPepperShops anpassen.....	6
13. Wohin mit dem Sprachset?.....	6
14. Weitere Details / Informationen.....	7



PepperShop wird von Glarotech entwickelt und vertrieben.  
Seit 1998 ist das innovative Unternehmen im Internet tätig  
und auf E-Commerce spezialisiert. Sie als Kunde profitieren  
vom direkten Draht zu den Herstellern der Produkte.

Glarotech GmbH  
Toggenburgerstrasse 156  
CH-9500 Wil

[info@glarotech.ch](mailto:info@glarotech.ch)  
Tel. +41 (0)71 923 08 58  
[www.glarotech.ch](http://www.glarotech.ch)

## 1. Einleitung

Seit der Version 1.5 kann der PhPepperShop kundenseitig mehrsprachig betrieben werden. Diese Anleitung beschreibt die nötigen Arbeiten, welche ausgeführt werden müssen, damit der PhPepperShop in einer weiteren Sprache betrieben werden kann. Dieses Dokument ist mit PhPepperShop ab Version 5.0 und neuer einsetzbar. Für ältere Shopversionen stehen andere Versionen dieser Anleitung zur Verfügung.

Der PhPepperShop setzt sogenannte Sprachsets ein. Diese befinden sich im Ordner `{shop_verzeichnis}/shop/language`. Hier findet man für jede Sprache einen Ordner.

Ein Sprachset besteht aus folgenden Dateien:

```
{sprachcode}
|-- countries.inc.php
|-- html_templates
|  |-- impressum.tpl.html
|  |-- start_content.tpl.html
|-- index.html
|-- localetext.inc.php
|-- regions.inc.php
|-- text_templates
|  |-- artikel_an_lager.txt
|  |-- artikel_noch_nicht_bewertet.txt
|  |-- bestellung_pending_abgelehnt.txt
|  |-- bestellung_storniert.txt
|  |-- bestellung_versendet.txt
|  |-- e_content_download_versendet.txt
|  |-- haendler_abholung_info.txt
|  |-- kunde_zugangsdaten.txt
|  |-- kundeninfo_vor_session_delete.txt
|  |-- loyalty_gutschein_versand.txt
|  |-- mahnung_1.txt
|  |-- mahnung_2.txt
|  |-- mail_newsletter_opt_in.txt
|  |-- mail_newsletter_reminder.txt
|  |-- password_vergessen_bekommen.txt
|  |-- password_vergessen_bekommen_plain.txt
|  |-- password_vergessen_link.txt
|  |-- vorauskasse_erinnerung.txt
```

Einige Dateien im Unterordner `text_templates` gehören zu Modulen und sind ev. nicht vorhanden.

Den grössten Teil der Dateien machen die sprachspezifischen Buttons aus. Weit wichtiger aber, und ziemlich unscheinbar, ist die Datei `localetext.inc.php`, sie beherbergt die übersetzten Texte der kundenseitigen Darstellung des PhPepperShops. Daneben findet man auch noch die HTML-Templates, welche lokalisiert sind (standardmässig nur das Impressum und die Startseite). Als drittes, zu übersetzendes Element, gibt es die Datei `countries.inc.php`, welche die Namen aller Länder dieser Welt beherbergt. Nun folgt noch die `regions.inc.php` mit den Regionaldaten dieser Welt. Im Shop dient sie als Basis der Lieferlandauswahl.

## 2. Kurzbeschreibung

Bei einer Übersetzung ist im Wesentlichen folgendes zu tun:

- Übersetzung der Texte `localetext.inc.php`, `countries-` + `regions.inc.php`, und `*.txt` Files
- Übersetzung der Willkommenseite (`start_content.tpl.html`)
- Erstellung der Grafiken (Buttons)
- Anpassen des Shops und Aktivierung des neuen Sprachsets

## 3. Vorbereitungen

Wir gehen davon aus, dass man sich im Verzeichnis `{shopdir}/shop/language` befindet. `{shopdir}` steht dabei für das Verzeichnis, in welchem der PhPepperShop installiert ist.

Am einfachsten kopiert man einen vorhandenen Sprachset-Ordner (z.B. de für Deutsch, en für Englisch) und benennt die Kopie gleich so, wie der entsprechende ISO-639-1 Sprachcode der neuen Sprache heisst z.B. fr (für Französisch) oder nl (für Holländisch). ISO-Quelle z.B. wie folgt:

[http://www.loc.gov/standards/iso639-2/php/English\\_list.php](http://www.loc.gov/standards/iso639-2/php/English_list.php)

## 4. Texte übersetzen

### 4.1 Übersetzung mit dem Lokalisierungseditor

Im PepperShop gibt es einen Online-Editor zur Bearbeitung der im Shop verwendeten Übersetzungstexte. Dazu wechselt man in die Shop Administration und dort ins Menü Sprachen. Hier findet man zuunserst den Link 'Alle übersetzten Texte anpassen'. Wenn man diesen Link anklickt kann man die neue Sprache wählen und dann bequem alle Angaben und Übersetzungen im Editor durchführen. Da der Editor noch neu ist, empfehlen wir häufiges Zwischenspeichern.

### 4.2 Übersetzung direkt in der Übersetzungsdatei

Nun geht man in den neu erstellten Ordner und öffnet die Datei `localetext.inc.php`. Info: Alle PepperShop Sourcecode Dateien benutzen UNIX-Zeilenumbrüche, dementsprechend 'durcheinander gewürfelt' sieht die Darstellung auf Windows Rechnern aus. Ein Editor, welcher PHP-Dateien gut darstellt und UNIX-Zeilenumbrüche verarbeiten kann, findet man hier zum Download:

<https://notepad-plus-plus.org/>

**ACHTUNG: ALLE DATEIEN VERWENDEN FOLGENDE ENCODIERUNG: UTF-8 (OHNE BOM), UNIX ZEILENENDE**

Bevor man sich an die eigentliche Übersetzung macht, sollte man sich den Beginn der Datei gut anschauen. Hier sollte man die Sprache korrekt angeben und auch den eigenen Namen und die EMail Adresse der / des Übersetzers angeben.

In der `localetext.inc.php` muss man nun alle Textpassagen übersetzen und die Sprachinfos anpassen (charset, ...). Dies macht man, indem man jede Zeile die mit `$locale_strings` beginnt wie folgt übersetzt. In der eckigen Klammer folgt eine Referenz, an dieser darf man *nichts* ändern. Nach dem Gleichzeichen folgt der zu übersetzende Text(teil). Hier muss darauf geachtet werden, dass das Hochkomma am Anfang des Textteils und am Ende des Textteils auf keinen Fall verloren geht. Zum besseren Verständnis sind bei gewissen Textteilen dahinter nach jeweils zwei Slashes (//) noch Kommentare angebracht worden. Beispiel einer Zeile:

```
$locale_strings['artikelweiterempfehlen'] = 'Diesen Artikel weiterempfehlen';
```

Viele Zeilen besitzen sogenannte Platzhalter in den zu übersetzenden Texten. Platzhalter sehen wie folgt aus: `%s`. Für was die Platzhalter stehen steht jeweils im Kommentar ganz rechts, hinter der Zeile geschrieben. Die Platzhalter **DÜRFEN NICHT GELÖSCHT WERDEN**. Sie müssen im übersetzten Text wieder am richtigen Ort platziert werden.

Info: Spezialfälle sind die letzten Abschnitte Kunden Zusatzfelder, lokalisierte AGBs (Allgemeine Geschäftsbedingungen), Kundeninformationen und Artikel Zusatzfelder ganz unten in der jeweiligen `localetext.inc.php`. Die (Kunden) Zusatzfelder und die AGBs lassen sich für die Default-Sprache bequem über das Webinterface tätigen (Zusatzfelder → Kundenattribute, AGBs → Kundenattribute, AGBs → Kundenattribute, AGBs → Allgemeine Shopeinstellungen / Kundeninformationen). Wenn der Shop vom Kunden in den jeweiligen Sprachmodus geschaltet wird und er in diesen Variablen Angaben findet, welche entweder leer sind oder sich von den in der Datenbank gespeicherten Werten unterscheiden, so benutzt der Shop die in der `localetext.inc.php` angegebenen Werte.

## 5. Länder und Regionen

### 5.1 Ländernamen

In den Dateien `countries.inc.php` (Länder) muss man den Namen des Landes in der neuen Sprache angeben. Es wird ISO-3166-2 verwendet. Dabei muss nur der Name in den Hochkommata auf der rechten Seite eingegeben werden, hier zwei Beispiele:

```
$country_strings['AD'] = 'Andorra';  
$country_strings['CI'] = 'Côte d'Ivoire';
```

Im obigen Beispiel sieht man, dass einfache Hochkommata mit einem Backslash-Zeichen escaped werden müssen. Aus ' wird \ '.

### 5.2 Regionen von Ländern

*Beschreibung:* Es gibt Regionen-Vorlagen in vielen verschiedenen Sprachen als Übersetzungsdateien im PO-Format. Falls man eine PO-Übersetzungsdatei direkt importieren möchte, stellt PepperShop ein PHP basiertes Converterprogramm zur Verfügung um daraus eine `regions.inc.php` für die jeweilige Sprache zu erstellen: `{shop_dir}/shop/language/iso_3166_po_converter.php`. Dieser Converter portiert ein gettext PO-File in einen PHP-Array für Regionen.

*Anwendung:* In der Datei `{shop_dir}/shop/language/iso_3166_po_converter.php` muss `$file` definiert werden:

```
// *****  
$file = './NL.po';  
// *****
```

In diesem Beispiel steht `$file = './NL.po'`; . Der Shop erwartet also eine Datei `NL.po` im selben Verzeichnis. Nun kann man einfach eine Datei gleich benennen oder mehrere hineinlegen und `$file` nach jedem Aufruf anpassen.

Verwendbare `.po`-Quelldateien (ISO-3166-2) findet man einfach via GIT: `git clone git://git.debian.org/git/iso-codes/iso-codes.git`. Nun erhält man viele `Regions`-Dateien im PO-Format. Diese Dateien dann entsprechend bereitlegen und dann im Browser den `iso_3166_po_converter.php` aufrufen - fertig ist die neue `regions.inc.php`.

#### Manuelle Übersetzung:

In der Datei `regions.inc.php` werden die Regionen der Länder übersetzt. Hier gilt dasselbe vorgehen, wie oben bei den Ländern beschrieben. Auch hier einige Beispiele:

```
$region_strings['en']['Bayern'] = 'Bayern';  
$region_strings['en']['Bremen'] = 'Bremen';  
$region_strings['en']['Hamburg'] = 'Hamburg';
```

## 6. Buttons übersetzen

Ist die Übersetzung der Texte in `localetext.inc.php` vollbracht, sollte man die Buttons im `Buttons` Unterverzeichnis anpassen, insbesondere das Bild `lang_pic.gif`, welches ein Piktogramm für die Sprache enthält. Die Buttons-Änderungen sollten über das Administrationsinterface gemacht werden. Man darf nicht vergessen, dass der Ordner `Buttons` und alle darin befindlichen Dateien ausser der `index.html` für alle lesbar markiert werden müssen (nur UNIX-Systeme) - das heisst der Ordner `Buttons` muss die Dateizugriffsrechte `777` haben, die Bilder darin dann `666` oder `777`, dies ist wichtig, sonst kann man unter Umständen die Bilder später nicht über die Shop-Administration verwalten (ändern).

Die Texte der Buttons kann man bei der Übergabe eines neuen Sprachsets auch einfach an Glarotech GmbH mitgeben. Wir erstellen dann ein lokalisiertes Button-Set mit den Standard-Buttons.

## 7. Willkommenseite (in HTML) anpassen

Was man hingegen direkt via HTML-Editor erledigen kann, ist die Anpassung der Datei mit dem Namen `start_content.html`. Dies ist die Willkommensdatei die der Kunde sieht, wenn er den Shop betritt (und diese Sprache vorgewählt hat). Man kann dieses HTML-Template auch direkt über die Shopadministration bearbeiten.

## 8. Optional - HTML-Templates anpassen

Dies ist im Moment nicht nötig, soll an dieser Stelle aber erwähnt werden, weil in den HTML-Templates auch sprachspezifische Änderungen enthalten sein können.

Hierzu kann man einfach die entsprechenden HTML-Dateien im Unterverzeichnis `html_templates` bearbeiten.

## 9. Text-Templates anpassen

Im Unterordner `text_templates` befinden sich Textdateien, welche als Vorlagen für Mails dienen – z.B. um den Kunden zu benachrichtigen, dass seine bestellten Waren ausgeliefert wurden. Auch diese Dateien haben UNIX-Zeilenumbrüche, siehe dazu den Downloadlink zu einem geeigneten Editor in Kapitel 4.2.

## 10. Das neue Sprachset in Betrieb nehmen

Nun ist das 'Sprachset' bereit. Wir können es im Shop aktivieren. Dazu geht man ins Administrationsmenü und dort in das Untermenü Sprachen. Hier sieht man nun neu aufgelistet und als verfügbar markiert, die neue Sprache. Mit einem Klick auf das Kästchen unterhalb von 'aktiviert' und nach dem Speichern steht die neue Sprache dann zur Verfügung.

## 11. Ausschliesslich diese Sprache verwenden

Wenn man nun im eigenen Shop ausschliesslich diese Sprache verwenden möchte, muss man in die Datei `{shopdir}/shop/config.inc.php` gehen und bei `DEFAULT_LANGUAGE` die Konstante mit dem ISO-639-1 Code der Sprache definieren (siehe auch Kommentar auf der Zeile dort).

## 12. Suchmaschinenoptimierungstags des PhPepperShops anpassen

Beim ausschliesslichen Betrieb mit der neu übersetzten Sprache:

In der Datei `{shopdir}/shop/shopindex.php` und `katalog.php` bei `DC.Coverage` (2 mal und 4 mal) das Land hinzufügen. Dies ist nötig, damit die Suchmaschinenindizierung des PhPepperShops die Dublin Core Metatags für das Land richtig setzt.

## 13. Wohin mit dem Sprachset?

Wer ein Sprachset erstellt hat, kann es uns komprimiert in einem der unten aufgeführten Formate an [developers@peppershop.com](mailto:developers@peppershop.com) zusenden. Wir werden es danach allen Sponsoren zur Verfügung stellen und auf unserer Sprachpaketeliste posten (<http://www.phpeppershop.com/sprachsets/sprachsets.php>).

Falls es zuviel Aufwand ist, das ganze Sprachset zu erstellen, reicht es uns, die übersetzte `localetext-`

`t.inc.php` zuzusenden zusammen mit den Mail-Vorlagen und im E-Mail noch die übersetzten Schriftzüge auf den Buttons zu nennen. Wir erstellen die Buttons dann hier.

## 14. Weitere Details / Informationen

Weitere Informationen findet man in der `info.txt` Datei, welche in jedem PhPepperShop im Verzeichnis `{shopdir}/shop/language/` liegt. Darin ist auch der Übersetzungsmechanismus erklärt.

Bei offenen Fragen stehen die PhPepperShop Foren offen: [www.peppershop.com/forum/](http://www.peppershop.com/forum/).