

Mehrsprachige Sites mit Elementor und WPML

Elementor ist ein Page Builder, mit dem Sie das Design und den Inhalt Ihrer Site steuern können. Verwenden Sie Elementor mit WPML und Sie können jedes gewünschte Design erstellen und den gesamten Inhalt Ihrer Website einfach übersetzen.

Auf dieser Seite

- [Einstieg](#)
- [Übersetzen von Elementor-Seiten mit WPML](#)
- [Übersetzen von Elementor Global Widgets](#)
- [Übersetzen von Elementor-Vorlagen](#)
- [WPML-Sprachumschalter-Widget](#)
- [Liste der übersetzbaren Elementor-Elemente](#)
- [WPML-fähige Elementor-Add-Ons](#)
- [Bekannte Probleme mit Elementor](#)
- [Holen Sie sich Hilfe von unserem Support](#)
- [Holen Sie sich WPML](#)

Einstieg

Beginnen Sie mit der Installation und Aktivierung der folgenden Plugins:

- [Elementor-Plugin](#), kostenlose oder Pro-Version
- WPML-Kern-Plugin und WPML-String-Übersetzungs-Add-on

Wenn Sie neu bei WPML sind, lesen Sie unseren [Leitfaden für die ersten Schritte](#). Es führt Sie schnell durch alle Übersetzungsoptionen, die Sie verwenden können.

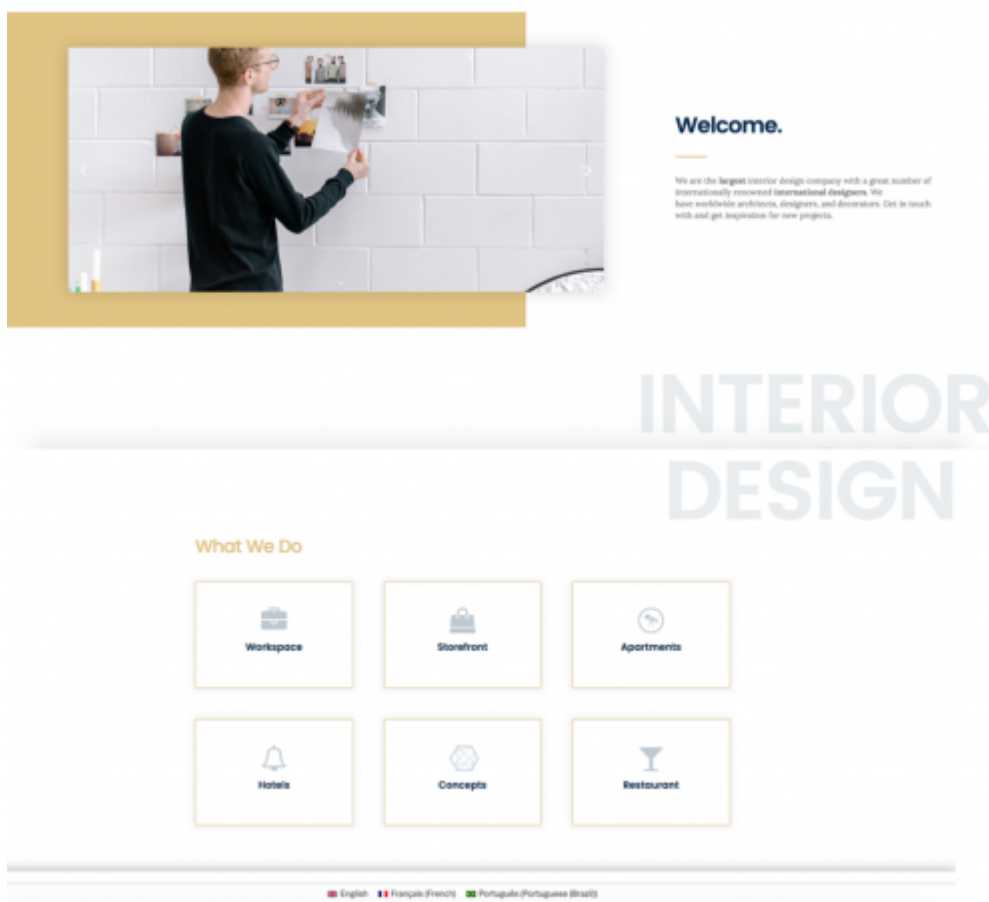
Eine dieser Optionen ist [Alles übersetzen](#) – der schnellste Weg zum Übersetzen und der einfachste Weg, Ihre Inhalte auf dem neuesten Stand zu halten. Diese Funktion übersetzt automatisch

Seiten, Beiträge, benutzerdefinierte Beitragstypen, benutzerdefinierte Felder, Taxonomien, WooCommerce-Produkte und mehr, während Sie Inhalte hinzufügen und bearbeiten.

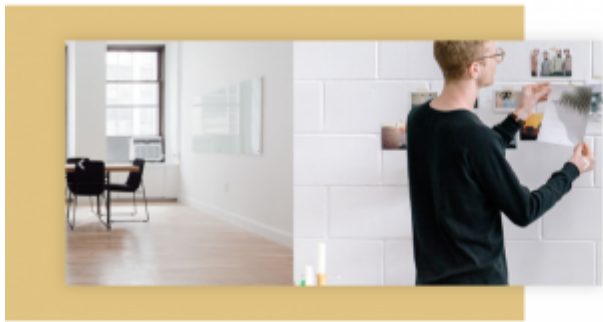
Übersetzen von Elementor-Seiten mit WPML

Wenn Sie nicht aktiviert haben **Alles** übersetzen, um Ihre mit Elementor erstellten Beiträge und Seiten automatisch zu übersetzen, können Sie sie genauso übersetzen wie normale WordPress-Seiten und -Beiträge. Weitere Sie in unserer Dokumentation [Informationen zum Übersetzen von Seiten, Beiträgen und benutzerdefinierten Beitragstypen finden](#).

Seite im Originalinhalt



Seite ins Französische übersetzt

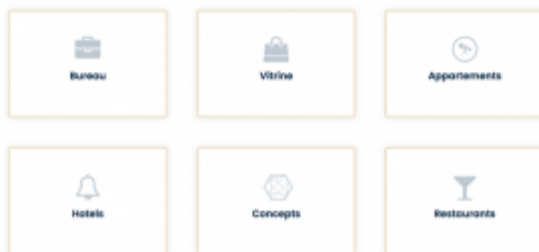


Bienvenue

Nous sommes la plus grande entreprise de décoration d'intérieur.
Nous avons les plus grands architectes internationaux.

DÉCORATION D'INTÉRIEUR

Ce qu'on fait



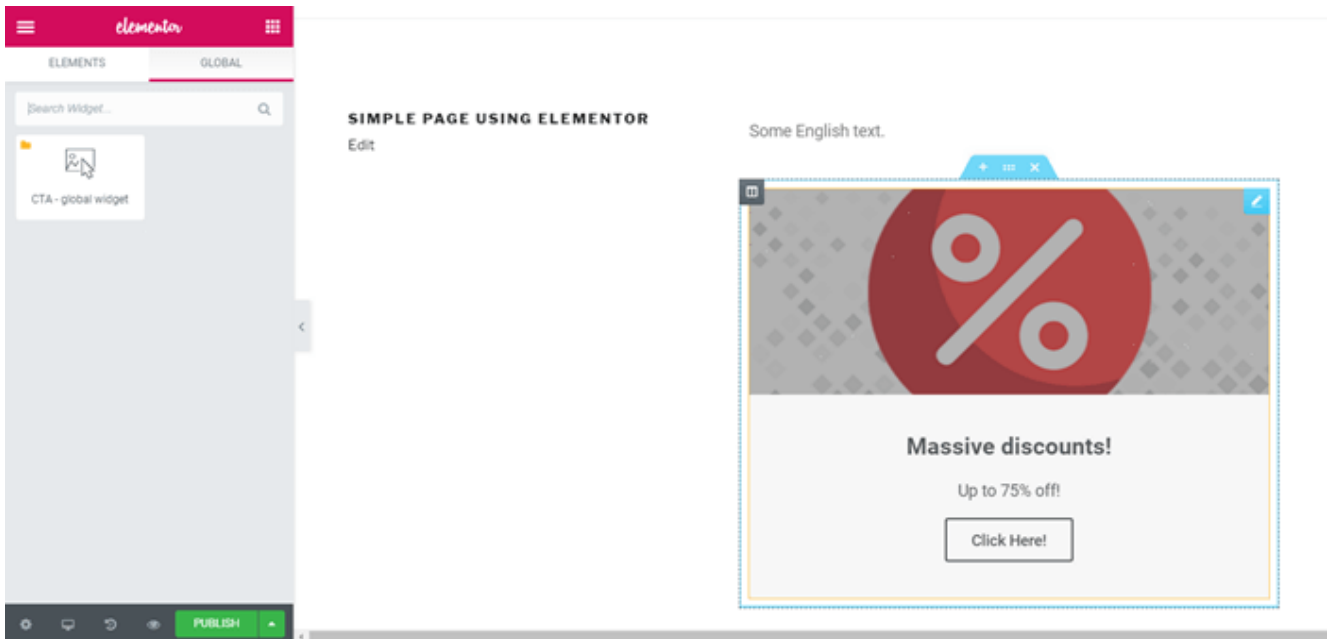
English (Anglais) Français Portugais (Português - do Brasil)

Übersetzen von Elementor Global Widgets

[Globale Widgets](#) sind eine Funktion von Elementor Pro. Es gibt Ihnen die zentrale Kontrolle über ein Widget, das in verschiedenen Bereichen Ihrer Website angezeigt wird. Das Aktualisieren des globalen Widget-Inhalts an einem Ort aktualisiert ihn überall.

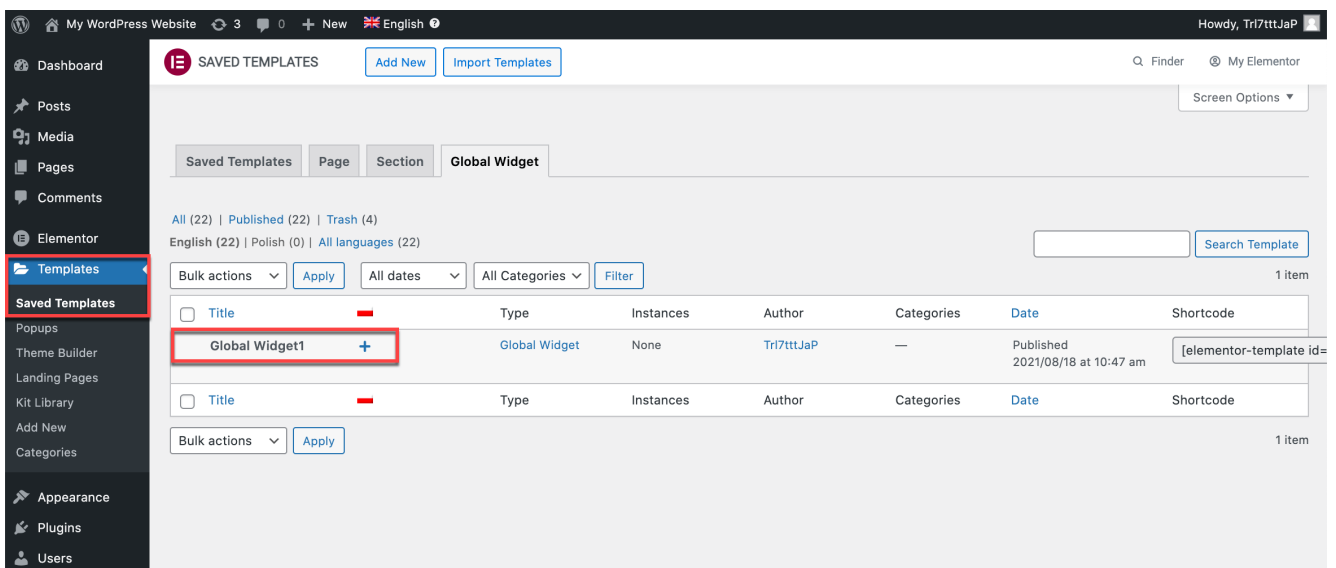
Wenn **Alles übersetzen** aktiv ist, werden globale Widgets automatisch übersetzt. Wenn **einige übersetzen** aktiv ist, führen Sie die folgenden Schritte aus, um sie zu übersetzen.

Wir haben mit Elementor eine einfache Seite erstellt, ein hinzugefügt Call-to-Action- **Schaltflächen-** Widget und [als globales Widget gespeichert](#) .



Speichern eines Elements als globales Widget
Verwenden Sie die folgenden Schritte:

1. Um die Texte des globalen Widgets so zu übersetzen, dass alle gewünschten Sprachen bei der Wiederverwendung des Widgets verfügbar sind, navigieren Sie zu **Vorlagen** → **Gespeicherte Vorlagen** und klicken Sie auf die **Globales Widget** Registerkarte .
2. Klicken Sie auf das **Plus-** Symbol unter der Sprache und neben der Vorlage, die Sie übersetzen möchten.

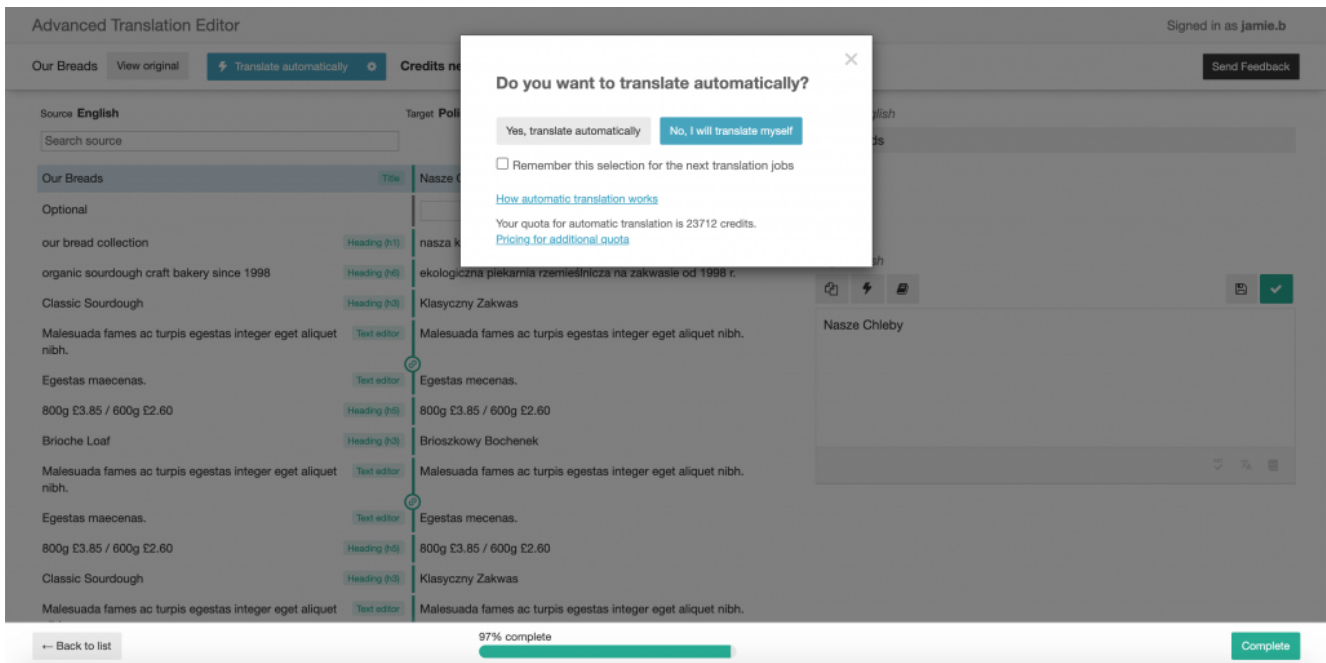


Übersetzen des globalen Widgets aus dem Elementor-Vorlagenmenü

3. Sie werden auf die Seite des erweiterten Übersetzungseeditors weitergeleitet, auf der Sie in einem

Popup gefragt werden, ob Sie den Inhalt automatisch übersetzen oder selbst übersetzen möchten. Wenn Sie auf Ja klicken, sind Ihre Übersetzungen in wenigen Augenblicken fertig.

4. Nachdem Sie die Übersetzungen überprüft und gespeichert haben, klicken Sie auf **Fertig stellen**.



Erweiterter Übersetzungseditor von WPML

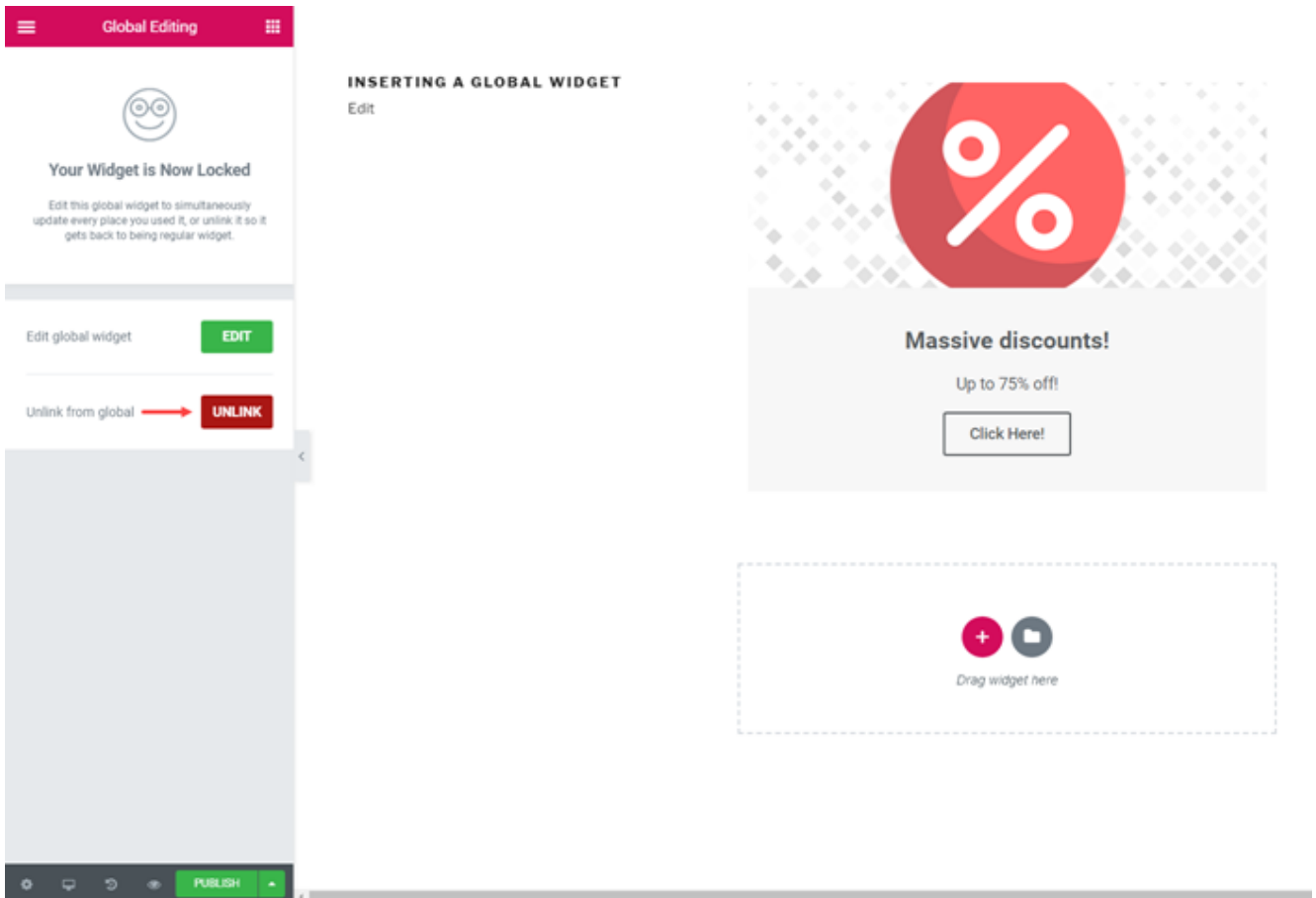
Wenn Sie in Zukunft den Inhalt eines globalen Widgets aktualisieren, können Sie die Übersetzungen mit den gleichen Schritten aktualisieren.

Einfügen eines übersetzten globalen Widgets in eine Seite

Sobald Ihr globales Widget übersetzt ist, können Sie es in jede Seite Ihrer Website einfügen. einfach [Übersetzen die Seite, zu der](#) Sie es hinzugefügt haben, und WPML kümmert sich darum, die übersetzte Version des globalen Widgets im Frontend anzuzeigen.



Eine übersetzte Seite und ein globales Widget im Frontend
Wenn Sie ein globales Widget in eine Seite und die **einfügen
Verknüpfung aufheben** , wird es zu einem normalen Widget. Um
ein normales Widget zu übersetzen, übersetzen Sie die Seite,
auf der es angezeigt wird, und der Widget-Text wird in den
erweiterten Übersetzungsektor aufgenommen .



Ein globales Widget einfügen und die Verknüpfung aufheben

Übersetzen von Elementor-Vorlagen

Mit Elementor können Sie Vorlagen erstellen, die in jede Seite Ihrer Website eingefügt werden können.

Um einen Abschnitt, eine Seite, eine Fußzeile, eine einzelne oder eine Archivvorlage zu übersetzen, stellen Sie zunächst sicher, dass Sie sie veröffentlichen. Befolgen Sie dann diese Schritte:

1. Gehen Sie zu **WPML → Translation Management** , wählen Sie **Vorlage aus** im ersten Dropdown-Menü und klicken Sie auf die **Filter** Schaltfläche . Die Liste wird aktualisiert und zeigt alle Ihre Elementor-Vorlagen an.
2. Wählen Sie die Vorlagen aus, die Sie übersetzen möchten.
3. Wählen Sie die Übersetzungsaktion für jede Sprache aus (übersetzen, duplizieren, nichts tun).
4. Klicken Sie auf **Ausgewählten Inhalt übersetzen** . Bitte

beachten Sie, dass diese Schaltfläche mit der Beschriftung beschriftet ist, wenn Sie nicht der einzige Übersetzer auf der Site sind **Ausgewählten Inhalt in den Übersetzungskorb hinzufügen** .

The screenshot shows the WPML Translation Management interface. At the top, there are tabs for 'Dashboard', 'Translators', 'Automatic Translation', and 'Jobs'. Below this, there is a section titled '1. Select items for translation' which contains a filter bar with various dropdown menus and a 'filter' button. Below the filter bar is a table with columns for 'Title', 'Type', 'Date', and 'Actions'. The first row in the table is 'Call to action', which is selected with a checkbox. Below the table, there is a section titled '2. Select translation options' which contains radio buttons for 'Translate', 'Duplicate content', and 'Do nothing' for both 'All Languages' and 'Spanish'. A 'Translate selected content' button is located at the bottom of this section. Red circles with numbers 1, 2, 3, and 4 highlight the filter bar, the 'Call to action' checkbox, the 'Translate' radio button for Spanish, and the 'Translate selected content' button respectively.

Senden einer Elementor-Vorlage zur Übersetzung mit WPML Translation Management
Wenn Sie Alles übersetzen verwenden, werden Ihre Vorlagen in einer Minute übersetzt.

Wenn Sie selbst übersetzen oder andere Übersetzer verwenden, gehen Sie zur **Übersetzungskorb** Registerkarte , wählen Sie aus, wer die Vorlage übersetzen soll, und klicken Sie auf , um Elemente zur Übersetzung zu senden.

Das folgende Bild zeigt, wie unsere übersetzte Beispielseite im Frontend aussieht.

TODAS LAS CONSULTAS

Para todas las consultas de servicio, complete el formulario y nos comunicaremos con usted lo antes posible.

Estaremos encantados de responder a todas sus preguntas y aseguramos de que tenga una experiencia increíble en nuestro salón.

Nombre completo*

Número de teléfono*

Dirección de correo electrónico*

Tema*

Mensaje

ENVIAR MENSAJE

Eine übersetzte Elementor-Vorlage im Frontend
Besuchen Sie unsere spezielle Seite, um mehr über die zu erfahren [Verwendung des Übersetzungsmanagements zum Senden von Inhalten zur Übersetzung](#).

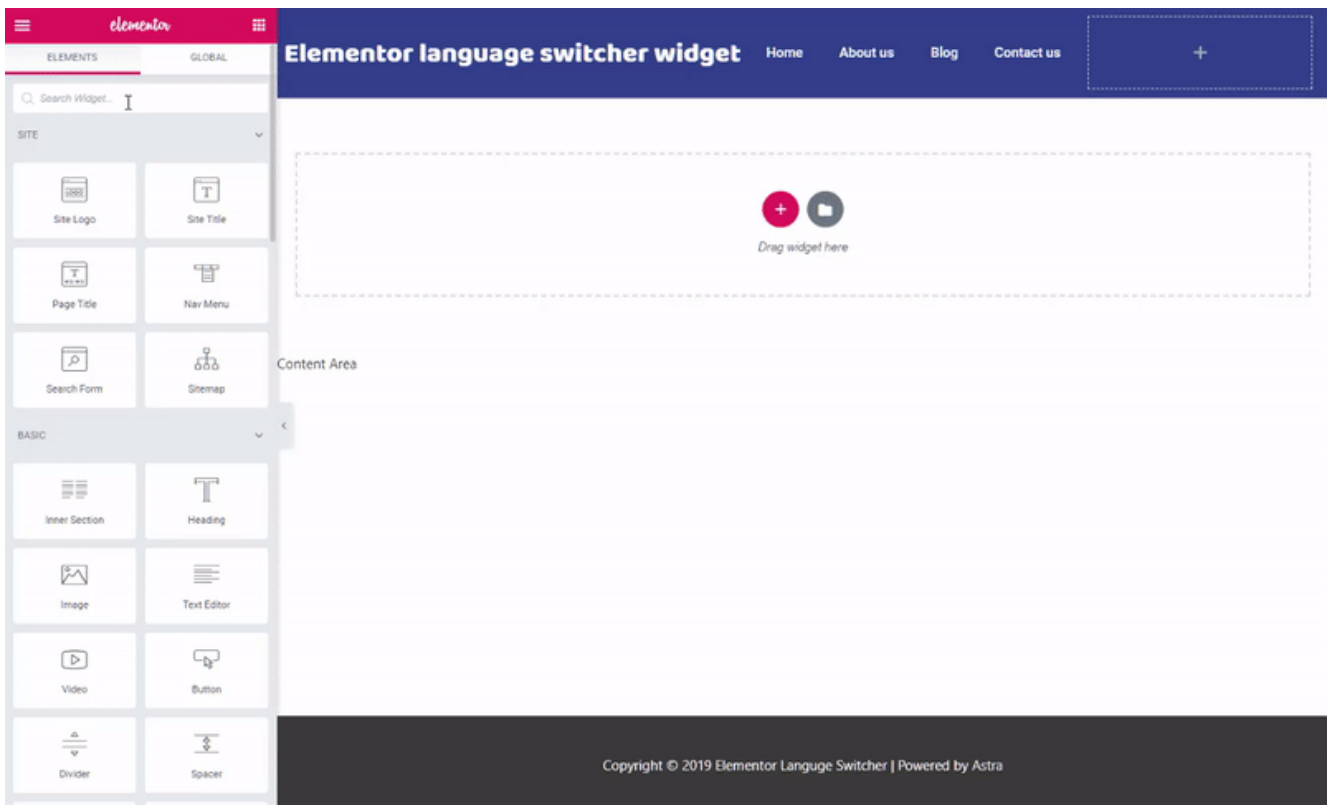
WPML - Sprachumschalter-Widget

Mit dem WPML-Sprachumschalter-Widget erhalten Sie alle Standardanpassungsoptionen für den [WPML-Sprachumschalter](#), wie z. B. die Anzeige der Flagge und des nativen Sprachnamens. Darüber hinaus können Sie den Sprachumschalter mit den Stilooptionen von Elementor gestalten. Sie können beispielsweise die Typografie und den Text ändern oder die Ränder und den Abstand steuern.

Sie können das WPML Language Switcher-Widget überall auf Ihrer Website hinzufügen. Dazu gehören Kopfzeile, Fußzeile, Beiträge und Seiten.

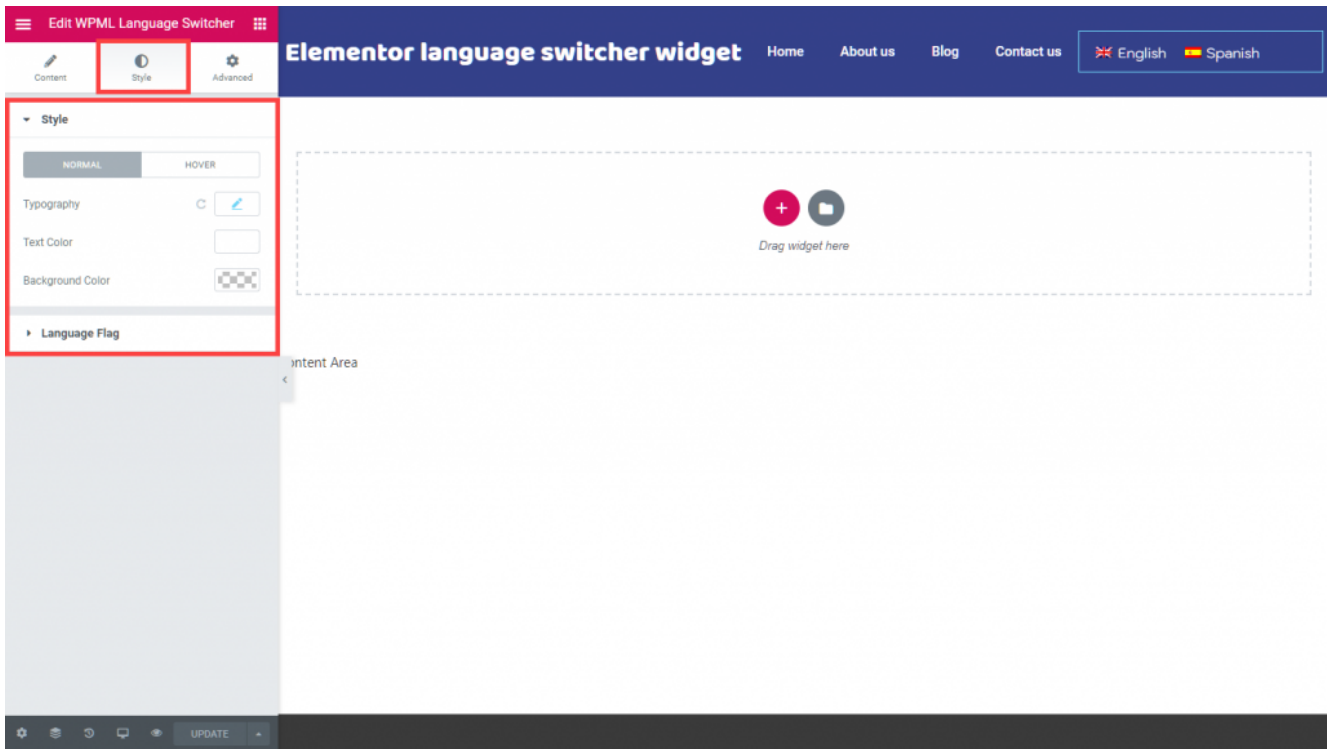
Um beispielsweise den WPML-Sprachumschalter zu einer Header-Vorlage hinzuzufügen:

1. Gehen Sie zu **WPML** → **Sprachen** und aktivieren Sie den benutzerdefinierten Umschalter im **Benutzerdefinierte Sprachumschalter** Abschnitt . Klicken Sie auf die Sie die **Anpassen gewünschten** Schaltfläche und stellen Optionen ein.
2. Bearbeiten Sie die Kopfzeile mit Elementor. Suchen Sie nach dem **WPML Language Switcher**- Widget und ziehen Sie es dann hinüber, um es der Kopfzeile hinzuzufügen.



WPML - Sprachumschalter-Widget zum Header hinzufügen

3. Passen Sie den Sprachumschalter auf der **Stil an** Registerkarte . Sie können weitere Anpassungen vornehmen, indem Sie zur navigieren **Erweitert** Registerkarte .



Styling-Optionen für WPML-Sprachumschalter-Widgets
 So sieht der angepasste Sprachumschalter unseres Headers im Frontend aus:

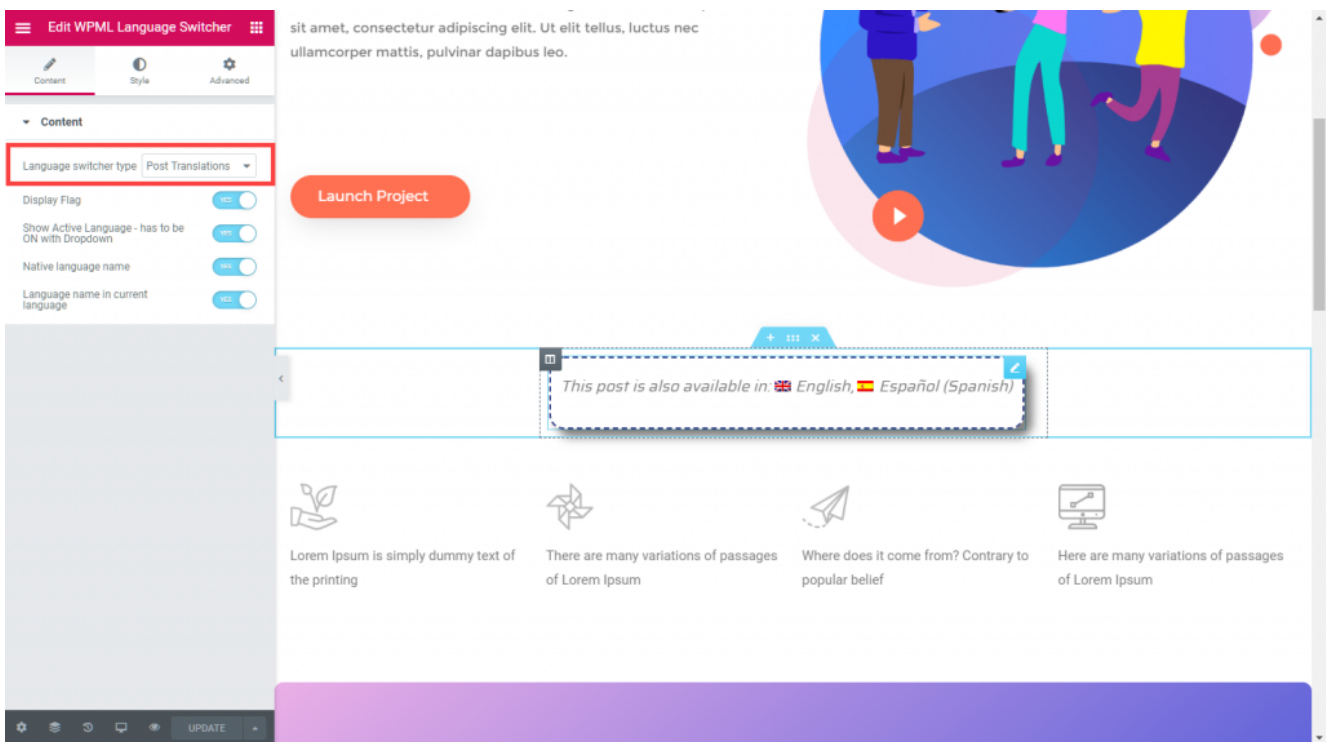


Der Header-Sprachumschalter im Frontend
 Wenn Sie die Sprachumschalter in eine Fußzeile Vorlage hinzufügen möchten, einfach die eingestellte **Sprachumschalter Typ** auf **Footer** .

Sie können den Sprachumschalter auch in Ihren Seiten und

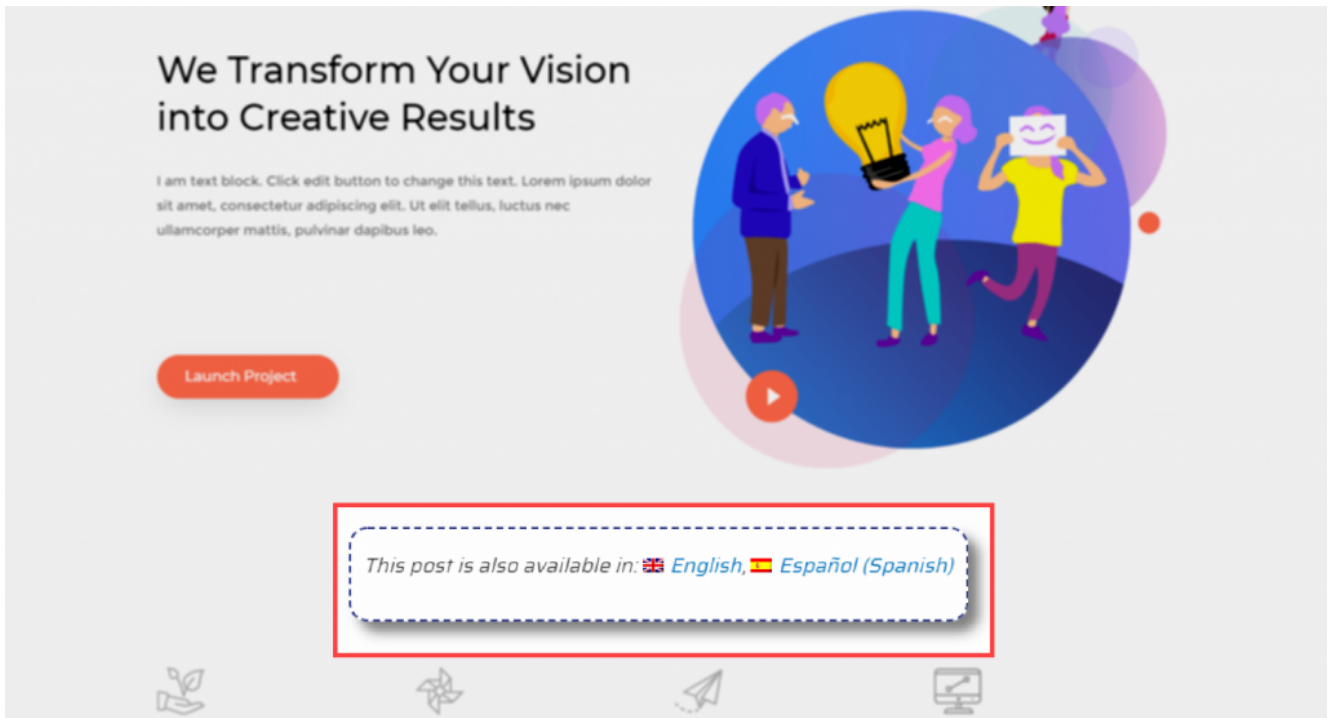
Beiträgen platzieren:

1. Bearbeiten Sie eine Seite oder einen Beitrag mit Elementor und fügen Sie das **WPML Language Switcher**-Widget an Ihrem gewünschten Ort hinzu
2. Setzen Sie den **Sprachumschaltertyp** auf **Post Translations**



Einstellen des Sprachumschaltertyps auf Post Translations

So sieht unser angepasster Sprachumschalter im Frontend aus:



WPML-Sprachumschalter in einem Beitrag oder einer Seite

Liste aller übersetzbaren Elementor-Elemente

Wenn Sie ein benutzerdefiniertes Elementor-Widget verwenden, müssen [Sie WPML-Unterstützung hinzufügen](#) dafür .

WPML unterstützt derzeit die folgenden Elementor-Widgets:

- Überschrift
- Text Editor
- Video
- Taste
- Symbol
- Preisliste
- Preisliste
- Flip-Box
- Folien
- Bildbox
- Symbolfeld
- Symbolliste
- Zähler

- Fortschrittsanzeige
- Zeugnis
- Registerkarten
- Akkordeon
- Umschalten
- Alarm
- HTML
- Form

WPML-fähige Elementor-Add-Ons

Der Prozess der Übersetzung von benutzerdefinierten Elementor-Elementen, die von Drittanbieter-Plugins bereitgestellt werden, ähnelt der Übersetzung der Standard-Elementor-Elemente. Eine Liste der empfohlenen Verzeichnis für empfohlene [WPML-Ready Elementor Addons finden Sie](#) in unserem Plugins.

Wir empfehlen auch, unseren Artikel über die [besten Elementor-Addons für mehrsprachige Websites zu lesen](#).

Bekannte Probleme

WPML funktioniert mit diesem Plugin gut, aber manchmal kann es kleinere Probleme geben, an denen wir arbeiten. Dies wird erwartet, da beide Plugins häufige Updates bereitstellen.

Aktuelle Probleme:

- [Elementor – WordPress-Widget-Titel können nicht übersetzt werden](#)
- [Seiten, die ein Elementor-Bildkarussell-Widget mit unerwarteten Werten enthalten, stürzen beim Übersetzungsversuch ab](#)
- [Elementor – Navigation zum vorherigen und nächsten Beitrag zeigt Beiträge in gemischten Sprachen an](#)

Hilfe von unserem technischen Support erhalten

Falls Sie Hilfe bei der Übersetzung Ihrer mit Elementor und WPML erstellten Site benötigen, besuchen [Sie das Support-Forum von WPML](#).

Bereit anzufangen?

Sie können WPML 30 Tage lang testen und sehen, wie es Ihnen gefällt. Wenn Sie damit nicht zufrieden sind, erstatten wir Ihnen Ihr Geld ohne Rückfragen innerhalb der 30-tägigen Testphase. [Kaufen und herunterladen](#)

Themes WPML Konfigurations-Datei – wpml-config.xml

WPML kann eine Konfigurationsdatei lesen, die ihm mitteilt, was in Themes und Plugins übersetzt werden muss. Die Datei heißt `wpml-config.xml` und befindet sich im Stammordner des Plugins oder Themes.

Inhalt

1. [Der Zweck der Sprachkonfigurationsdatei](#)
2. [Automatische Generierung der Datei wpml-config.xml](#)
3. [Struktur und Syntax](#)
 - [Inhalt des Seitenerstellers](#)
 - [Benutzerdefinierte Felder](#)
 - [Benutzerdefinierte Bedingungen](#)
 - [Benutzerdefinierte Typen](#)

- [Benutzerdefinierte Taxonomien](#)
 - [Gutenberg blocks](#)
 - [Admin-Texte / wp_options](#)
 - [Konfiguration des Sprachumschalters](#)
4. [Verwenden der WPML-Sprachkonfigurationsdatei mit untergeordneten Designs](#)

Zweck

Um die Kompatibilität mit WPML zu erreichen, sollten Sie auch eine Konfigurationsdatei erstellen, die Ihnen bei der Aufrechterhaltung der Kompatibilität in Ihren zukünftigen Versionen hilft. WPML kann alles auf Ihrer WordPress-Site übersetzen, aber Sie müssen ihr mitteilen, was übersetzt werden muss. Das macht diese Datei.

Gehen Sie zur **WPML → Einstellungen** Seite .

Settings

Multilingual Content Setup Translation Notifications Custom XML Configuration

[General settings](#) [How to translate posts and pages](#) [Posts and pages synchronization](#) [Translated documents options](#) [Custom posts slug translation options](#)
[Translation pickup mode](#) [XLIFF file options](#) [Custom Fields Translation](#) [Custom Term Meta Translation](#) [Post Types Translation](#) [Taxonomies Translation](#) [Media Translation](#)
[Translate Link Targets](#)

WPML can read a configuration file that tells it what needs translation in themes and plugins. The file is named wpml-config.xml and it's placed in the root folder of the plugin or theme.
[Learn more](#)

General settings Disable the possibility of creating accounts for users on wpml.org

How to translate posts and pages

New content

Use WPML's Advanced Translation Editor
Advanced Translation Editor is enabled and active

Use WPML's Classic Translation Editor

Seite WPML-Einstellungen

Diese Seite sagt WPML alles, was es wissen muss, einschließlich der zu übersetzenden oder zu synchronisierenden

benutzerdefinierten Felder, welche benutzerdefinierten Beiträge und Taxonomien mehrsprachig sein sollten und sogar welche Admin-Texte zu übersetzen sind.

Die Sprachkonfigurationsdatei enthält diese Informationen, damit nicht jeder Benutzer sie manuell auf der Admin-Seite eingeben muss.

Für einige der Themes und Plugins hosten wir die Sprachkonfigurationsdateien auf unseren Servern. Eine Liste können Sie einsehen [hier](#). Es ist so eingestellt, dass es die lokale Sprachkonfigurationsdatei überschreibt, die sich im Stammordner des Themes oder Plugins befindet.

Mit WPML können Sie auch Design- [manuell alle Einstellungen überschreiben, die von und Plugin-Sprachkonfigurationsdateien geladen werden](#). Dies gilt sowohl für Sprachkonfigurationsdateien im Stammordner des Themes oder Plugins als auch für die auf unseren Servern gehosteten Sprachkonfigurationsdateien.

Automatische Generierung der Datei wpml-config.xml

Wenn Sie mit der Erstellung von XML-Dateien nicht vertraut sind, hat unser Team das [Multilingual Tools-Plugin erstellt](#), das diese Aufgabe erleichtert. Obwohl es ursprünglich als Werkzeug gedacht war, um und Plugin-Autoren zu helfen, ihre Produkte mehrsprachig zu machen, kann es leicht verwendet werden, um Ihre eigene zu generieren **Theme- wpml-config.xml-Datei** .

Um mehr über das Generieren Ihrer zu erfahren **wpml-config.xml-Datei** , besuchen Sie die des [Seite Multilingual Tools-Plugins](#) . Sehen Sie sich insbesondere den **Abschnitt Wie erzeuge ich Sprachkonfigurationsdateien mit mehrsprachigen Tools an?** Sektion.

Sobald Sie die Konfigurationsdatei haben, fügen Sie sie dem Stammverzeichnis Ihres Theme-Ordners hinzu. Wenn Sie bereits eine haben, überschreiben Sie sie nicht. Bearbeiten Sie stattdessen Ihre ursprüngliche XML-Datei und fügen Sie den mit dem generierten Code hinzu **Multilingual Tools**- Plugin .

Bitte beachten Sie, dass dieses Plugin nicht für den Einsatz auf den Live-Produktionsseiten vorgesehen ist.

Um dieses Tutorial zu lesen und Sprachkonfigurationsdateien für Ihre Themes und Plugins zu , können Sie mit diesem Beispiel beginnen – [erstellen wpml-config.zip](#) .

Sie müssen sie bearbeiten, können aber die Abschnitte und die Struktur dieser Datei verwenden.

Struktur und Syntax

Der Inhalt der wpml **-config.xml** Datei muss in diese Tags verpackt werden

```
<wpml-config>
```

und

```
</wpml-config>
```

Derzeit können vier Datentypen und zwei WPML-Konfigurationseinstellungen für die Übersetzung in dieser Konfigurationsdatei konfiguriert werden:

1. [Inhalt des Seitenerstellers](#)
2. [Benutzerdefinierte Felder](#)
3. [Benutzerdefinierte Bedingungen](#)
4. [Benutzerdefinierte Typen](#)
5. [Benutzerdefinierte Taxonomien](#)
6. [Gutenberg blocks](#)
7. [Admin-Texte / wp_options](#)
8. [Konfiguration des Sprachumschalters](#)

1. Inhalt des Seitenerstellers

Mit WPML können Sie die , die der `wpml-config.xml` **Shortcodes Inhaltsübersetzung** Datei verwenden, um zu definieren hinzugefügt werden müssen.

1.1 Zeichenfolgen übersetzen

Betrachten wir ein Beispiel, bei dem Sie einer Seite mit Visual Composer ein Texttrennzeichen hinzufügen. Dieses Trennzeichen hat einen Titel und sein Shortcode sieht so aus:

Um den Titel dieses zu übersetzen, müssen wir unserer einige Zeilen **Texttrennzeichens wpml-config.xml hinzufügen** Datei . Auf diese Weise „weiß“ WPML, dass der Titel dieses Trennzeichens übersetzt werden muss. Dies folgt tatsächlich der gleichen Logik, die für die benutzerdefinierten Beitragstypen, benutzerdefinierten Felder und andere verwendet wird.

Der folgende Code ist ein Beispiel dafür, was wir hinzufügen müssen `wpml-config.xml` in diesem Fall der Datei . Beispiel für das Hinzufügen eines Page Builder-Shortcodes zur Datei `wpml-config.xml`

```
1
2
3
4
5
6
7 <wpml-config> <shortcodes>   <shortcode>   <tag>vc_text_separator</tag>   <attributes>   <attribute>title</attribute>   </attributes>   </shortcode></shortcodes> </wpml-config>
8
9
10
11
12
```

Gehen wir den Aufbau des obigen Beispiels durch:

- Beginnen Sie mit dem Shortcodes-Tag. Alle Shortcodes auf Ihrer Website, die übersetzt werden müssen, sollten unter diesem Tag platziert werden.
- Verwenden Sie dann das Shortcode-Tag, um alle Tags zu umschließen, die zu einem einzigen Shortcode gehören.
- Das Tag namens **tag** wird verwendet, um den Namen des Shortcodes zu definieren. In diesem Fall ist es der **vc_text_separator** . Sie können benutzerdefinierte

Optionslabels hinzufügen, die sowohl im angezeigt werden [erweiterten Übersetzungseeditor](#) als auch im [klassischen Übersetzungseeditor](#). Diese Etiketten sind auch beim Export in XLIFF-Dateien enthalten. Siehe das folgende Beispiel zu Tag- und Attributlabels.

- Shortcodes können ein oder mehrere Attribute haben, daher packen wir sie in das Attribute-Tag (Plural) und verwenden das Attribut-Tag (Singular), um den Titel jedes Attributs zu definieren.

Page Builder enthalten (manchmal) Designelemente mit Linkattributen.

Sie können interne Links automatisch auf die übersetzte Version dieses Beitrags verweisen lassen, indem Sie die **Shortcode- wpml-config.xml** neue Link-Option der Datei verwenden: **type="link"** .

Sie können das **Codierungsattribut** mit diesem verwenden. Es behandelt spezielle Codierung, die verschiedene Seitenersteller verwenden. Das Codierungsattribut ist normalerweise spezifisch für den verwendeten Page Builder. Es akzeptiert die folgenden Werte:

- **base64** – Visual Composer-Roh-HTML-Shortcode. Der HTML- wird als gespeichert **Code base64-** String im Shortcode .
- **vc_link Linkformatierung** – Spezielle für Visual Composer.
- **av_link Linkformatierung** – Spezielle für Enfold.
- **allow_html_tags** – Normalerweise werden HTML-Tags aus Shortcode-Attributen entfernt. Setzen Sie die Kodierung auf **allow_html_tags**, wenn das Shortcode-Attribut HTML-Tags zulassen soll. Dies sollte mit Vorsicht verwendet werden, da das Zulassen von HTML-Tags in einigen Situationen die Formatierung durcheinander bringen und ein Sicherheitsproblem darstellen könnte.

Beispiel für das Hinzufügen eines Shortcode-Link-Attributs:

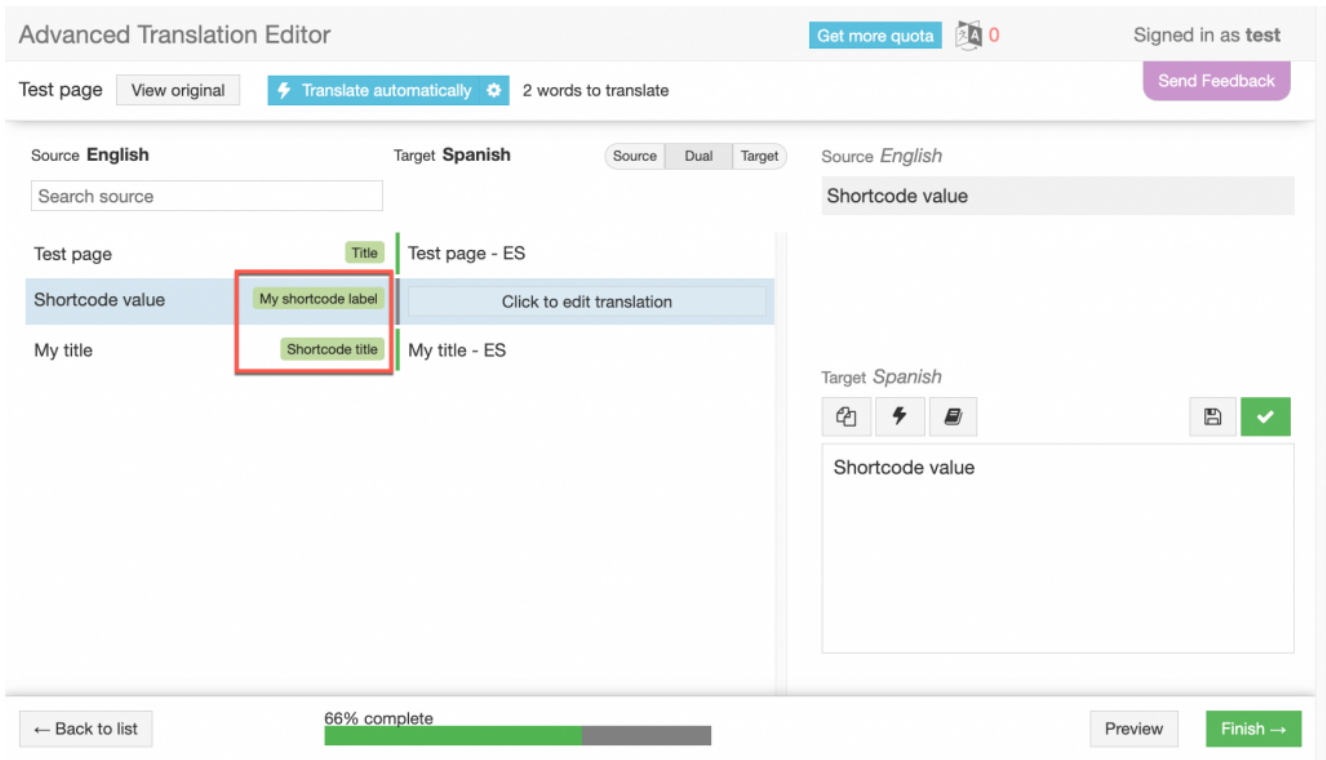
type="link"

```
1
2
3
4
5 <wpml-config> <shortcode>
6 <tag>av_button</tag> <attributes> <attribute>label</attribute> <attribute
7 type="link" encoding="av_link">link</attribute> </attributes></shortcode> </wpml-config>
8
9
10
11
```

Wenn Sie über die Verwendung von **urlencodierten Shortcodes nachdenken** , lesen Sie die Seite [unbedingt Übersetzen von urlencodierten Shortcodes](#) . Beispiel für das Hinzufügen von Tag- und Attributlabels

```
1
2
3
4 <wpml-config> <shortcodes> <shortcode> <tag label="My shortcode
5 label">my_shortcode</tag> <attributes> <attribute label="Shortcode
6 title">title</attribute> </attributes> </shortcode> </shortcodes></wpml-config>
7
8
9
10
```

Durch das Hinzufügen von Labels zu Tags und Attributen können Sie benutzerdefinierte Labels im erweiterten Übersetzungseeditor oder im klassischen Übersetzungseeditor anzeigen. Dies kann dem Übersetzer helfen, den Kontext der Zeichenfolge besser zu verstehen.



Definieren von benutzerdefinierten Shortcode-Labels im erweiterten Übersetzungseditor

1.2 Medien übersetzen

Sie können WPML Media Translation verwenden, um Bilder in Seitenerstellungsinhalten zu übersetzen. Dies geschieht durch Konvertierung von Bild-IDs und Bild-URLs. Es ist notwendig, dem Page Builder, der Shortcodes verwendet, mitzuteilen, wie diese Konvertierung durchzuführen ist. Der folgende Code ist ein Beispiel, das wir hinzufügen müssen **wpml-config.xml** in diesem Fall der Datei `wpml-config.xml` für die Bildübersetzung

```

1
2
3
4
5
6
7
8 <wpml-config> <shortcode> <!-- Convert media IDs in `gallery_ids` attribute, and ignore the shortcode content --> <tag ignore-
9 content="1">et_pb_gallery</tag> <attributes> <attribute type="media-
10 ids">gallery_ids</attribute> <attributes></shortcode><shortcode> <!-- Convert the media URL in the content, translate some string
11 attributes for `title` and `alt`, convert media URL in `src` attribute --> <tag type="media-
12 url">et_pb_image</tag> <attributes> <attribute>title_text</attribute> <attribute>alt</attribute> <attribute
13 type="media-url">src</attribute> </attributes></shortcode> </wpml-config>
14
15
16
17
18
19
20

```

Sie können die folgenden Werte verwenden:

- **ignore-content** – Kann innerhalb eines **Tag-** Elements verwendet werden. Dieser Wert ist optional und kann entweder **0** oder **1 sein** . Sie können dieses Attribut verwenden, um Abwärtskompatibilität für neue Medien-Shortcodes zu erreichen. Wenn der Wert auf gesetzt ist, wird **1** der Inhalt des Shortcodes nicht verarbeitet.
- **type** – Kann innerhalb eines **Tag-** Elements verwendet werden. Sie können auch Shortcode-Inhalte verwenden, die die URL von Medien als optionalen Wert in **media-url enthalten** .
- **type** – kann auch innerhalb eines **Attributelements** verwendet werden. In diesem Fall kann es einen der folgenden optionalen Werte haben:
 - **media-ids** – eine durch Kommas getrennte Liste von Medien-IDs.
 - **media-url** – URL des Mediums
 - **link** – verweist auf eine andere Seite der Site und WPML konvertiert sie automatisch in die URL der Seitenübersetzung

1.3 Seitenerstellungs-Widgets übersetzen

Ab WPML 4.4.4 können Sie nun Seitenerstellungs-Widgets in Ihrer Sprachkonfigurationsdatei registrieren. Auf unserer Dokumentationsseite erfahren Sie, [wie Sie Seitenerstellungs-Widgets für die Übersetzung registrieren](#) .

2. Benutzerdefinierte Felder

Der Name des benutzerdefinierten Felds muss angegeben werden und auch die von WPML erwartete Aktion: übersetzen, kopieren, einmal kopieren, ignorieren.

Beispiel für das Hinzufügen von benutzerdefinierten Feldern zur Sprachkonfigurationsdatei

1	
2	
3	
4	<wplm-config> <custom-fields> <custom-field
5	action="copy">quantity</custom-field> <custom-field
6	action="translate">custom-title</custom-field> <custom-field
7	action="copy">weight</custom-field> <custom-field action="copy-once">bg-
8	color</custom-field> <custom-field action="translate">custom-
9	description</custom-field> <custom-field action="ignore">date-
10	added</custom-field></custom-fields> </wplm-config>
11	
12	

Dieser Block muss unter dem Tag `<wplm-config>` verschachtelt werden.

Sie können die folgenden Übersetzungsoptionen für benutzerdefinierte Felder festlegen:

- **translate** : ermöglicht Ihrem Benutzer, den Wert des benutzerdefinierten Felds zu übersetzen. Diese Felder werden auf dem Bildschirm des Übersetzungseditors angezeigt und können an jeden der professionellen Übersetzungsdienste gesendet werden.
- **copy** : Diese Aktion kopiert den benutzerdefinierten Feldwert der Standardsprache in die sekundären Sprachen. Dies bedeutet, dass die Aktualisierung des benutzerdefinierten Feldwerts der Standardsprache immer in die Sekundärsprache kopiert wird. Die zum festgelegten benutzerdefinierten Felder werden **Kopieren** auf dem Bildschirm des Übersetzungseditors nicht angezeigt.
- **einmal kopieren** : Der Wert des benutzerdefinierten Felds wird beim anfänglichen Übersetzungsprozess in die Sekundärsprache kopiert. Die benutzerdefinierten Felder, die die **einmalige** Aktion verwenden, werden nicht auf dem Bildschirm des Übersetzungseditors angezeigt. Der Benutzer kann jedoch den benutzerdefinierten Feldwert der Sekundärsprache mithilfe des Nachbearbeitungsbildschirms so ändern, dass er sich von

der Standardsprache unterscheidet. Es wird bevorzugt, die benutzerdefinierten Felder festzulegen, die Einstellungen wie Hintergrundfarbe, Schriftfarbe, Schriftgröße und andere enthalten. , zum **einmaligen Kopieren** . Dadurch kann der Benutzer andere Einstellungen für übersetzte Inhalte vornehmen als für die Beiträge und Seiten in der Standardsprache. Der Benutzer möchte beispielsweise festlegen, dass die Hintergrundfarbe desselben Elements in der Standardsprache gelb und in der Sekundärsprache blau ist. Beachten Sie, dass das Bearbeiten eines auf eingestellten **einmaliges Kopieren** Felds das Feld nicht als markiert **Aktualisierungsbedarf** . Dies liegt daran, dass das Feld nicht in eine vorhandene Übersetzung kopiert wird, sondern erst beim Erstellen der Übersetzung kopiert wird.

- **ignorieren** : Diese Aktion verhindert, dass das benutzerdefinierte Feld in die Sekundärsprache kopiert wird.

Sie können den Attribute hinzufügen **<custom-field>** -Tags . Diese Attribute passen die Anweisungstexte im [erweiterten Übersetzungseeditor](#) oder im [klassischen Übersetzungseeditor an](#) .

- **style** kann **line** , **textarea** oder **visual sein** , um eine einzelne Zeile, einen Textbereich oder WYSIWYG anzuzeigen.
- **Beschriftung** wird neben dem Feld angezeigt.
- **group** gibt an, ob das benutzerdefinierte Feld zu einer Gruppe gehört und wie die Bezeichnung der Gruppe lauten soll. Wenn sich ein Feld in einer Gruppe befindet:

* das Feld wird aus dem benutzerdefinierten Feldbereich entfernt

* das Feld wird dem Abschnitt der zugehörigen Gruppe hinzugefügt Beispiel für benutzerdefinierte Feldattribute

```

1
2
3 <wpml-config> <custom-fields> <custom-field action="translate"
4 style="line" label="Title">custom-title</custom-field> <custom-field
5 action="translate" style="textarea" label="Description">custom-
6 description</custom-field> <custom-field action="translate"
7 style="visual" label="Some content" group="Custom group">custom-
8 wysiwyg</custom-field> </custom-fields> </wpml-config>
9

```

Sie können auch optionale hinzufügen **Codierungsattribute** , um die Codierung vom Standardwert (zu ändern **keine** Codierung) .

Das Codierungsattribut akzeptiert die folgenden Werte:

- **json**
- **base64**
- **URL-Code**

Das Attribut ermöglicht Ihnen die Verwendung mehrerer Codierungen. In diesem Fall müssen die Werte durch ein Komma getrennt werden, zB encoding="json,base64". Beispiel für das Hinzufügen von benutzerdefinierten Feldern zur Sprachkonfigurationsdatei, um die Codierung zu ändern

```

1
2
3
4 <wpml-config> <custom-fields> <custom-field action="translate"
5 encoding="json,base64">keywords</custom-field></custom-fields> </wpml-config>
6
7

```

Wenn dieser **wpml-config.xml**- Inhalt verwendet wird, werden diese benutzerdefinierten Felder im angezeigt [erweiterten Übersetzungs-Editor](#) oder im [klassischen Übersetzungs-Editor](#) , wie in der folgenden Abbildung gezeigt. Erweiterter Übersetzungseditor Klassischer Übersetzungseditor

Benutzerdefinierte Feldbezeichnungen im erweiterten Übersetzungseditor

Benutzerdefinierte Feldbeschriftungen im klassischen

2.1 Unterschlüssel in benutzerdefinierten Feldern übersetzen

WPML übersetzt standardmäßig alle Unterschlüssel in benutzerdefinierten Feldern. Es ist möglich, ein solches Verhalten zu umgehen, indem Sie angeben, welche Unterschlüssel übersetzt werden sollen. Codebeispiel für die Übersetzung von Unterschlüsseln in benutzerdefinierten Feldern

```
1
2
3
4
5
6
7
8
9
10 <wpml-config> <custom-fields> <custom-field action="translate">with_attributes</custom-
11 field> <custom-field action="translate">with_deep_attributes</custom-field> <custom-field
12 action="translate">no_attributes</custom-field></custom-fields> <custom-fields-texts> <key
13 name="with_attributes"> <key name="attribute1" /> <key name="attribute2"
14 /> </key> <key name="with_deep_attributes"> <key name="attribute1" /> <key
15 name="attribute2"> <key name="level1"> <key
16 name="level2"></key> </key> </key> </key></custom-fields-texts> </wpml-config>
17
18
19
20
21
22
23
24
25
```

Es ist auch möglich, Platzhalter auf die gleiche Weise zu verwenden, wie sie für [Admin-Texte verwendet werden](#) :

- alle Unterfelder ab, die mit dem **Gleichen Sie Titel beginnen** und Code verwenden:
<key name="title-*" />
- Übereinstimmung mit allen Unterfeldern und wird im Allgemeinen verwendet, um einen Array-Index mithilfe von Code abzugleichen:
<key name="*" />
- Bekommen [{"title": "First title"}, {"title": "Second title"}] Code verwenden Beispiel einer

```
Konfigurationsdatei mit Platzhaltern 1 2 3 4 5 6 7 8 9
10 11 <wpml-config> <custom-fields-texts> <key
name="key-name"> <key name="*"> <key name="title"
/> </key> </key></custom-fields-texts> </wpml-config>
```

3. Benutzerdefinierte Bedingungen

Der benutzerdefinierte Begriffsname muss angegeben werden und auch die von WPML erwartete Aktion: übersetzen, kopieren, einmal kopieren, ignorieren. Beispiel für das Hinzufügen von benutzerdefinierten Begriffen zur Sprachkonfigurationsdatei

```
1
2
3
4 <wpml-config> <custom-term-fields> <custom-term-field
5 action="copy">term_meta_A</custom-term-field> <custom-term-field
6 action="translate">term_meta_B</custom-term-field> <custom-term-field
7 action="ignore">term_meta_C</custom-term-field> <custom-term-field action="copy-
8 once">term_meta_D</custom-term-field></custom-term-fields> </wpml-config>
9
10
```

Dieser Block muss unter dem Tag `<wpml-config>` verschachtelt werden.

Sie können die folgenden Übersetzungsoptionen für benutzerdefinierte Felder festlegen:

- **translate** : ermöglicht Ihrem Benutzer, den Wert des benutzerdefinierten Begriffs zu übersetzen. Diese Begriffe werden auf dem Bildschirm des Übersetzungseditors angezeigt und können an jeden der professionellen Übersetzungsdienste gesendet werden.
- **copy** : Diese Aktion kopiert den benutzerdefinierten Begriffswert der Standardsprache in die Sekundärsprachen. Dies bedeutet, dass die Aktualisierung des benutzerdefinierten Begriffswerts der Standardsprache immer in die Sekundärsprache kopiert wird. Die zu benutzerdefinierten Terme werden **kopierenden** nicht auf dem Bildschirm des Übersetzungseditors angezeigt.

- **einmal kopieren** : Der Wert des benutzerdefinierten Begriffs wird beim ersten Übersetzungsprozess in die Sekundärsprache kopiert. Die benutzerdefinierten Begriffe, die die **einmalige** Aktion verwenden, werden nicht auf dem Bildschirm des Übersetzungseitors angezeigt. Der Benutzer kann jedoch den benutzerdefinierten Begriffswert der Sekundärsprache mithilfe des Nachbearbeitungsbildschirms so ändern, dass er sich von der Standardsprache unterscheidet.
- **ignorieren** : Diese Aktion verhindert, dass der benutzerdefinierte Begriff in die Sekundärsprache kopiert wird.

4. Benutzerdefinierte Typen

Die benutzerdefinierten Beitragstypen, die WPML übersetzen soll. Benutzerdefinierte Typen in der Sprachkonfigurationsdatei

```
1
2
3      <wpml-config> <custom-types> <custom-type
4      translate="1">book</custom-type> <custom-type
5      translate="1">DVD</custom-type></custom-types> </wpml-config>
6
7
8
```

Sie können dem Tag das Attribut „display-as-translated“ hinzufügen, um die Beitragstypen in der Standardsprache anzuzeigen, wenn keine Übersetzung vorhanden ist. Verwenden des Attributs Als übersetzt anzeigen mit benutzerdefinierten Typen

```
1
2
3 <wpml-config> <custom-types> <custom-type translate="1"
4   display-as-translated="1">movie</custom-type></custom-
5     types> </wpml-config>
6
7
```

5. Benutzerdefinierte Taxonomien

Die benutzerdefinierten Taxonomien, die Ihr Plugin möglicherweise verwendet und die bereits bei WP registriert sind. Benutzerdefinierte Taxonomien in der Sprachkonfigurationsdatei

```
1
2
3   <wpml-config> <taxonomies> <taxonomy
4     translate="1">genre</taxonomy> <taxonomy
5     translate="1">type</taxonomy> <taxonomy
6   translate="0">publisher</taxonomy></taxonomies> </wpml-config>
7
8
9
```

Hinweis: Die Taxonomien, die keine Übersetzung benötigen, können in dieser Liste einfach weggelassen werden.

Sie können dem Tag das Attribut „display-as-translated“ hinzufügen, um die Taxonomien in der Standardsprache anzuzeigen, wenn keine Übersetzung vorhanden ist. Verwenden der Anzeige als übersetztes Attribut mit Taxonomien

```
1
2
3   <wpml-config> <taxonomies> <taxonomy translate="1" display-as-
4   translated="1">wild-west</taxonomy></taxonomies> </wpml-config>
5
6
7
```

6. Gutenberg blocks

Gutenberg ist der neue Editor, der mit der WordPress **5.0** kommt Version . In diesem Editor erstellen Sie Inhalte mithilfe von Blöcken.

Sie können angeben, welche Teile Ihres Gutenberg-Blocks übersetzt werden müssen, indem Sie Einstellungen zu **wpml-config.xml** hinzufügen .

[Xpath](#) wird verwendet, um Textteile zu definieren, die übersetzt werden müssen.

Nehmen wir an, wir haben ein Bild, das mit dem folgenden Code angezeigt wird: Beispiel für ein zu übersetzendes Bild

```
1 <!-- wp:image {"id":3} --><figure class="wp-block-image"><figcaption>Image
Caption</figcaption></figure><!-- /wp:image -->
```

Wir wollen das übersetzen figcaption und der altAttributwerte dieses Bildes.

Dazu muss der folgende Code in eingefügt werden **wpml-config.xml** : Beispiel für Einstellungen für die Übersetzung des Gutenberg-Blocks

```
1
2
3
4
5 <wpml-config> <gutenberg-blocks> <gutenberg-block type="core/image"
6 translate="1"> <xpath//figure/figcaption/> <xpath//figure/img/@alt/> </gutenberg-block></gutenberg-blocks> </wpml-config>
7
8
9
10
```

Bitte beachten dass der **Typ** ist core/image und nicht wp:image da dies der von der Block-API zurückgegebene Wert ist.

Sie können angeben, welche Gutenberg-Blockfelder Verknüpfungen sind. WPML ersetzt dann alle Links durch ihre Übersetzungen, sofern sie verfügbar sind. Übersetzung von Links in Gutenberg-Blöcken

```
1
2
3
4 <wpml-config> <gutenberg-blocks> <gutenberg-block type="core/some-block"
5 translate="1"> <xpath type="link">//a/@href</xpath> </gutenberg-
6 block></gutenberg-blocks> </wpml-config>
7
8
9
```

Übersetzung von Blockattributen

Hier ein Beispiel für ein Format zur Definition eines Editorblocks: Format zur Definition eines Editorblocks

```
1
2
3
4 <wpml-config> <gutenberg-blocks> <gutenberg-block
5 type="my-plugin/my-block" translate="1"> <xpath
6 label="My Block">//p</xpath> <key name="title"
7 /> <key name="foo"> <key name="bar1"
8 /> <key name="bar2"
9 /> </key> <key name="/^[^_]\S+$/> search-
10 method="regex" /> <key name="something" search-
11 method="wildcards" /> </gutenberg-block></gutenberg-
12 blocks> </wpml-config>
13
14
15
16
```

Sie können das **Schlüsselement** genauso verwenden, wie es bei der [Admintexte / wp_options](#) Konfiguration verwendet wird. Dies bedeutet auch, dass Sie Schlüsselemente innerhalb von übergeordneten Schlüsselementen haben können.

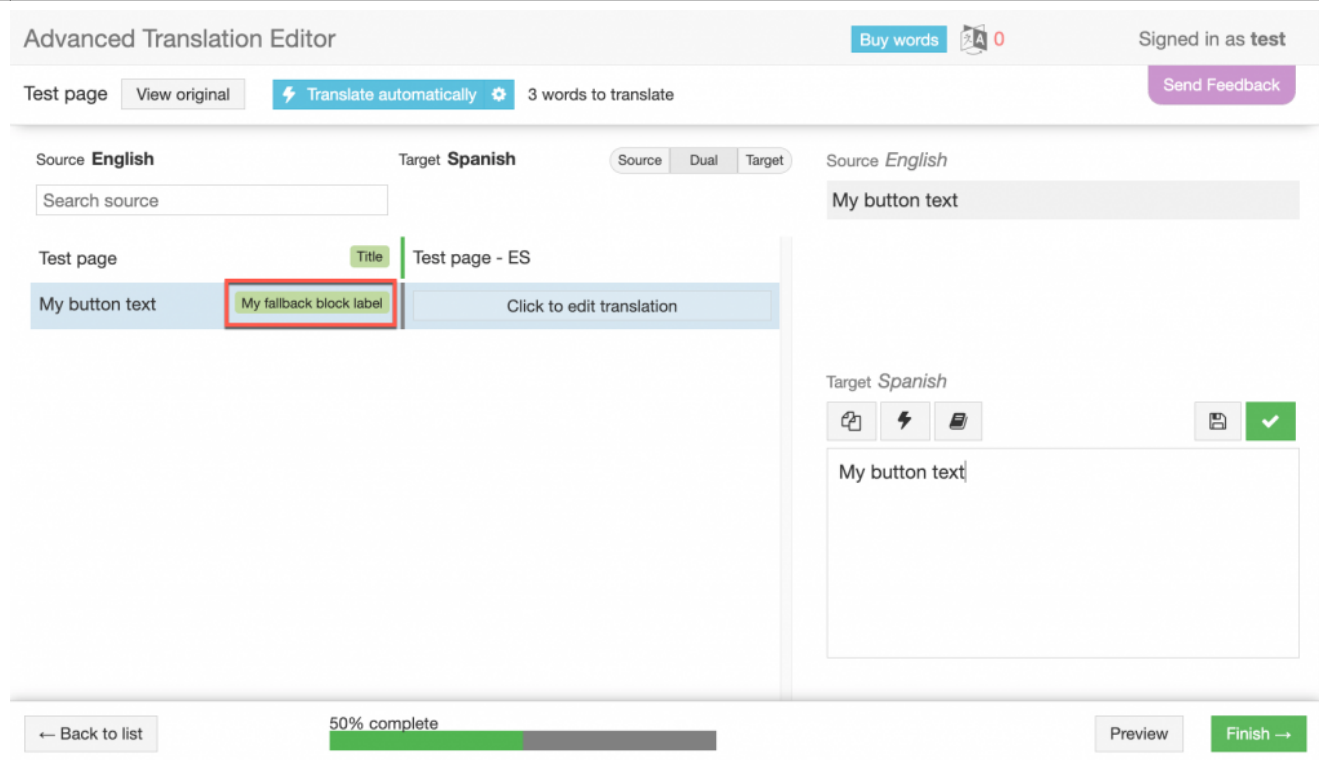
Sie können das **Label**- Attribut verwenden, um optionale benutzerdefinierte Labels hinzuzufügen, die im erweiterten Übersetzungseditor neben den Blockelementen angezeigt werden. Wenn das Label-Attribut Teil des **gutenberg-block**- Tags ist,

wird es als Fallback-Label für Elemente des Blocks verwendet, für die kein bestimmtes Label definiert ist. Beispiel für ein Fallback-Label

```

1
2
3
4     <wpml-config>   <gutenberg-blocks>           <gutenberg-block type="my-plugin/mybutton" translate="1" label="My fallback block
5 label">             <xpath>//span</xpath>           <xpath>//a@href</xpath>           </gutenberg-block>   </gutenberg-blocks></wpml-config>
6
7
8

```



Fallback-Label im erweiterten Übersetzungseditor

Das Attribut **search-method** kann einen von zwei Werten haben:

- Platzhalter (Standard)
- regex.

Wildcards können genauso verwendet werden, wie sie für Admin-Texte verwendet werden. Dies bedeutet, dass im Namensattribut ein Sternchen (*) verwendet werden kann. Hier ist ein Beispiel für einen Block: Beispiel für einen Block

```

1 2
3 4 <!-- wp:my-plugin/my-block {           "myp": {           "mypTitle":"The
5 6 title",           "mypContent":"The Content",           "_mypSystem":"Meta
7 8           attribute to not translate"           }           }/-->
9

```

Wir können die Blockdefinition mit einem Platzhalter setzen:

Blockdefinition mit Platzhalter

```
1
2
3
4     <wpml-config> <gutenberg-blocks>   <gutenberg-block type="my-
5 plugin/my-block" translate="1">       <key name="myp">           <key
6   name="myp*" />           </key>   </gutenberg-block></gutenberg-
7                                   blocks> </wpml-config>
8
9
10
11
```

Dadurch können wir „Der Titel“ und „Der Inhalt“ übersetzen, da dies die einzigen Attribute sind, die mit **myp** beginnen .

Die **Regex** ermöglicht es uns, einen regulären Ausdruck im name-Attribut zu haben. Dies kann bei komplexen Konfigurationen äußerst nützlich sein. Hier ist ein Beispiel für einen Block: Komplexe Konfiguration – Beispiel

```
1
2
3
4     <!-- wp:my-plugin/my-block   {           "data": {           "title": "The
5 title",           "_title": "e980759463943209f6f1ae09a239e353",           "content": "The
6 Content",           "_content": "1502a7b825dfe7b789c63830609f1701"           }   }/-->
7
8
9
10
```

Wir können die Blockdefinition mit einem regulären Ausdruck setzen: Blockdefinition mit einem regulären Ausdruck

```
1
2
3
4     <wpml-config> <gutenberg-blocks>   <gutenberg-block type="my-plugin/my-
5 block" translate="1">       <key name="data">           <key
6 name="/^[^_]\S+$/" search-method="regex" />           </key>   </gutenberg-
7                                   block></gutenberg-blocks> </wpml-config>
8
9
10
11
```

Dadurch können wir „Der Titel“ und „Der Inhalt“ übersetzen, da dies die einzigen Attribute sind, die nicht mit **_ beginnen** .

Namensraum blockieren

Sie können eine globale Konfiguration für den Blocknamensraum haben. Der Blockname ist kein erforderlicher Bestandteil der Definition mehr.

Wenn die Blockkonfiguration für alle Blöcke in einem Namensraum gleich ist, kann sie wie folgt geschrieben werden:
Blöcke in einem Namensraum

```
1
2
3
4 <wpml-config> <gutemberg-blocks> <gutemberg-block type="my-plugin"
5   translate="1"> <key name="data"> <key
6     name="/^[^_]\S+$/> search-method="regex"
7 /> </key> </gutemberg-block></gutemberg-blocks> </wpml-config>
8
9
10
11
```

7. Admin-Texte / wp_options

Strings, die Teil der Optionen sind, die die Plugins oder Designs in der **wp_options speichern** Tabelle .

Wenn Themes und Plugins [get_option verwenden](#) , lesen sie Werte aus der *wp_options* Tabelle . WPML kann diese Aufrufe filtern und die Werte dieser Optionen übersetzen.

Dies funktioniert, wenn der **Datensatz wp_option** ein einfacher String ist, aber auch, wenn es sich um ein serialisiertes Array handelt.

Um eine einzelne Option zu übersetzen, fügen Sie einen **Schlüsseleintrag hinzu** unter **admin-texts** . Um ein serialisiertes Array zu übersetzen, fügen Sie mehrere

Schlüssel unter einem **Schlüssel hinzu** , wie Sie in sehen können **my_plugin_options** im Beispiel unten . Admintexte in der Sprachkonfigurationsdatei

```
1
2
3
4
5
6
7         <wpml-config> <admin-texts> <key
8   name="my_plugins_options"> <key name="option_name_1"
9     /> <key name="option_name_2" /> <key
10  name="options_group_1"> <key name="sub_option_name_11"
11    /> <key name="sub_option_name_12" /> </key> <key
12  name="options_group_2"> <key name="sub_option_name_21"
13 /> <key name="sub_option_name_22" /> </key> </key> <key
14 name="simple_string_option"/></admin-texts> </wpml-config>
15
16
17
18
19
```

Es ist möglich, **den Platzhalter *** in Unterschlüsseln wie im folgenden Fall zu verwenden. Beispiel für die Verwendung von Platzhaltern

```
1
2
3
4         <wpml-config> <admin-texts> <key
5   name="testing_option"> <key name="*" /> </key></admin-
6     texts> </wpml-config>
7
8
9
```

Es entspricht diesem Code:

Der entsprechende Code

```
1
2
3
4
5
6         <wpml-config> <admin-texts> <key
7 name="testing_option"> <key name="sub_key_1"/> <key
8 name="sub_key_2"> <key name="sub_sub_21"> <key
9     name="sub_sub_211"/> </key> <key
10     name="sub_sub_21"/> </key> <key
11 name="more_sub_keys"/> </key></admin-texts> </wpml-config>
12
13
14
15
16
```

Bitte beachten Sie, dass der Platzhalter * **NICHT** in übergeordneten Schlüsseln funktioniert:

Der Platzhalter funktioniert nicht in übergeordneten Schlüsseln

```
1 2
3 4 <wpml-config> <admin-texts> <key name="some_option_*"
5 6     /> <key name="*" /></admin-texts> </wpml-config>
7 8
```

8. Konfiguration des Sprachumschalters

Aktiviert eine bestimmte Konfiguration für den in WPML integrierten Sprachumschalter. Es kann auch zum Zurücksetzen der Sprachumschalterkonfiguration verwendet werden, wenn sie vom Backend (von ihren Anfangswerten) geändert wurde.

Um neue Änderungen , klicken Sie **Standard** unten auf der Seite auf die Schaltfläche **anzuzeigen WPML → Sprachen**

wiederherstellen . Beispiel für das Hinzufügen einer benutzerdefinierten Konfiguration für den integrierten WPML-Sprachumschalter

1 2
3 4
5 6
7 8
9 10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80

```
<wpml-config> <language-switcher-settings> <key name="additional_css">{inline CSS styles}</key> <key
name="link_empty">{0 or 1}</key> <key name="copy_parameters">{parameter1, parameter2}</key> <key
name="sidebars"> <key name="{sidebar slug}"> <key name="display_flags">{0 or 1}</key> <key
name="display_names_in_current_lang">{0 or 1}</key> <key name="display_names_in_native_lang">{0 or
1}</key> <key name="display_link_for_current_lang">{0 or 1}</key> <key name="widget_title">{widget
title}</key> <key name="template">{template slug}</key> <!-- color picker keys --> <key
name="background_normal">#{hex color}</key> <key name="border_normal">#{hex color}</key> <key
name="font_current_normal">#{hex color}</key> <key name="font_current_hover">#{hex
color}</key> <key name="background_current_normal">#{hex color}</key> <key
name="background_current_hover">#{hex color}</key> <key name="font_other_normal">#{hex
color}</key> <key name="font_other_hover">#{hex color}</key> <key
name="background_other_normal">#{hex color}</key> <key name="background_other_hover">#{hex
color}</key> </key> </key> <key name="statics"> <key name="footer"> <key name="show">{0 or
1}</key> <key name="display_flags">{0 or 1}</key> <key name="display_names_in_current_lang">{0 or
1}</key> <key name="display_names_in_native_lang">{0 or 1}</key> <key
name="display_link_for_current_lang">{0 or 1}</key> <key name="template">{template slug}</key> <!--
color picker keys --> <key name="background_normal">#{hex color}</key> <key
name="border_normal">#{hex color}</key> <key name="font_current_normal">#{hex color}</key> <key
name="font_current_hover">#{hex color}</key> <key name="background_current_normal">#{hex
color}</key> <key name="background_current_hover">#{hex color}</key> <key
name="font_other_normal">#{hex color}</key> <key name="font_other_hover">#{hex color}</key> <key
name="background_other_normal">#{hex color}</key> <key name="background_other_hover">#{hex
color}</key> </key> <key name="post_translations"> <key name="show">{0 or
1}</key> <key name="display_flags">{0 or 1}</key> <key name="display_names_in_current_lang">{0 or
1}</key> <key name="display_names_in_native_lang">{0 or 1}</key> <key
name="display_link_for_current_lang">{0 or 1}</key> <key name="display_before_content">{0 or
1}</key> <key name="display_after_content">{0 or 1}</key> <key name="template">{template
slug}</key> <key name="availability_text">{string like "See post translations:
%s"}</key> </key> <key name="shortcode_actions"> <key name="display_flags">{0 or
1}</key> <key name="display_names_in_current_lang">{0 or 1}</key> <key
name="display_names_in_native_lang">{0 or 1}</key> <key name="display_link_for_current_lang">{0 or
1}</key> <key name="template">{template slug}</key> <!-- color picker keys --> <key
name="background_normal">#{hex color}</key> <key name="border_normal">#{hex color}</key> <key
name="font_current_normal">#{hex color}</key> <key name="font_current_hover">#{hex
color}</key> <key name="background_current_normal">#{hex color}</key> <key
name="background_current_hover">#{hex color}</key> <key name="font_other_normal">#{hex
color}</key> <key name="font_other_hover">#{hex color}</key> <key
name="background_other_normal">#{hex color}</key> <key name="background_other_hover">#{hex
color}</key> </key> </key></language-switcher-settings> </wpml-config>
```

Nicht alle dieser Abschnitte müssen in der Konfigurationsdatei vorhanden sein, sondern nur diejenigen, die für Ihr Plugin oder Design gelten.

Verwenden der WPML-Sprachkonfigurationsdatei mit untergeordneten Designs

Wenn Sie ein Child-Theme verwenden, überschreibt die Sprachkonfigurationsdatei des Parent-Themes die im Child-Theme gefundene. WPML bietet eine Konfigurationsseite, auf der Sie dies problemlos mit benutzerdefinierten Einstellungen überschreiben können.

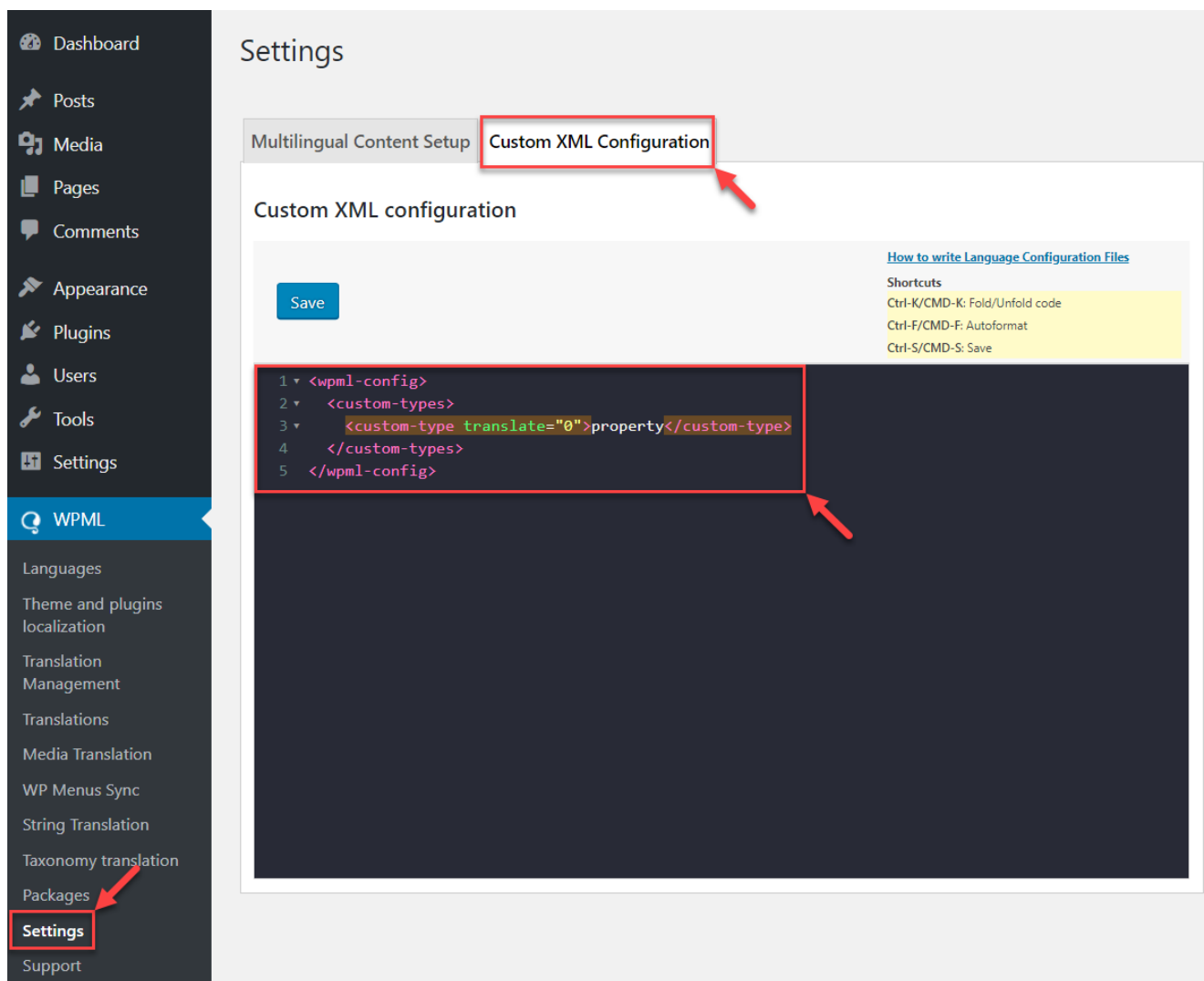
Betrachten wir ein Beispiel, bei dem die Sprachkonfigurationsdatei des übergeordneten Themas den benutzerdefinierten Beitragstyp „Eigenschaft“ als übersetzbar festlegt.

```
<wpml-config>
  <custom-fields>
    <custom-field action="translate">description</custom-field>
    <custom-field action="copy">price</custom-field>
    <custom-field action="copy">size</custom-field>
    <custom-field action="copy">build-year</custom-field>
  </custom-fields>
  <custom-types>
    <custom-type translate="1">property</custom-type>
  </custom-types>
  <taxonomies>
    <taxonomy translate="1">property-type</taxonomy>
  </taxonomies>
</wpml-config>
```

Die Konfigurationsdatei der übergeordneten Sprache legt den benutzerdefinierten Beitragstyp der Eigenschaft zum Übersetzen fest

Wenn Sie ein Child-Theme verwenden und den benutzerdefinierten Beitragstyp „Eigenschaft“ so einstellen möchten, dass er nicht

übersetzbar ist, navigieren Sie zur **WPML** → **Einstellungen** Seite und klicken Sie auf die **Benutzerdefinierte XML-Konfiguration** Registerkarte . Verwenden Sie den Editor, um den benutzerdefinierten Beitragstyp „Eigenschaft“ so einzustellen, dass er nicht übersetzbar ist. Setzen Sie einfach den des *translate*- Wert Attributs auf **0** statt auf **1** .



The screenshot shows the WordPress Settings page for WPML. The left sidebar contains a menu with 'Settings' highlighted. The main content area is titled 'Settings' and has a sub-tab 'Custom XML Configuration' selected. Below this, there is a 'Custom XML configuration' section with a 'Save' button and a code editor. The code editor contains the following XML snippet:

```
1 <wpl-config>
2   <custom-types>
3     <custom-type translate="0">property</custom-type>
4   </custom-types>
5 </wpl-config>
```

Red arrows point to the 'Custom XML Configuration' tab and the code editor. A yellow box highlights the 'Shortcuts' section on the right, which includes: Ctrl-K/CMD-K: Fold/Unfold code, Ctrl-F/CMD-F: Autoformat, and Ctrl-S/CMD-S: Save.

Überschreiben der Sprachkonfigurationseinstellungen des übergeordneten Themes

Die Einstellungen auf der **Benutzerdefinierte XML-Konfiguration** Registerkarte haben Vorrang vor den Einstellungen in der Sprachkonfigurationsdatei im übergeordneten Thema.

Themes und Plugins WPML kompatibel.

Jedes WordPress-Theme und jedes Plugin kann mit WPML kompatibel werden.

Folgen Sie diesem Tutorial, um zu erfahren, wie Sie die WPML-Kompatibilität testen und die erforderlichen Änderungen vornehmen, um Ihren Code vollständig mehrsprachig zu machen.



Wichtig: Denken Sie daran, eine Sprachkonfigurationsdatei einzuschließen

Wenn Sie fertig sind, müssen Sie eine [XML-Sprachkonfigurationsdatei erstellen](#) und sie zum Stammverzeichnis Ihres Themes hinzufügen.

Auf diese Weise können Endbenutzer mehrsprachige Websites mit Ihrem Thema betreiben, ohne selbst etwas konfigurieren zu müssen.

Ohne eine solche Sprachkonfigurationsdatei werden Ihre Endbenutzer Stunden damit verschwenden, Ihr Design zurückzuentwickeln und zu erraten, welche Texte übersetzt werden müssen. Dies führt zu mehr Unterstützungsarbeit für Sie und für uns. Wenn Sie Hilfe beim Erstellen einer Sprachkonfigurationsdatei benötigen, informieren Sie unser

Theme-Kompatibilitätsteam und es wird Ihnen helfen.

Auf dieser Seite

1. [Installieren von WPML-Komponenten](#)
2. [Fallstudie – Sydney-Thema](#)
3. [Übersetzen der Kopfzeile](#)
4. [Menüs übersetzen](#)
5. [Übersetzen der Dienste von Sydney Theme und anderer benutzerdefinierter Post-Typen](#)
6. [Übersetzen von benutzerdefinierten Theme- oder Plugin-Elementen](#)
7. [Den Post-Body übersetzen](#)
8. [Widgets übersetzen](#)
9. [Übersetzen der Fußzeile](#)
10. [Erstellen einer Sprachkonfigurationsdatei für Ihr Theme](#)
11. [GetText-Unterstützung für hartcodierte Zeichenfolgen](#)
12. [WooCommerce-Kompatibilität](#)
13. [Tipps und Debugging](#)

1. Installieren von WPML-Komponenten

WPML enthält mehrere Komponenten. Einige sind für Ihre Theme-Kompatibilitätsarbeit erforderlich und andere sind für Site-Administratoren gedacht, die später den Inhalt der Site verwalten.

Sie sollten die wesentlichen WPML-Komponenten aktivieren, mit denen Sie Ihr Theme vollständig übersetzen können:

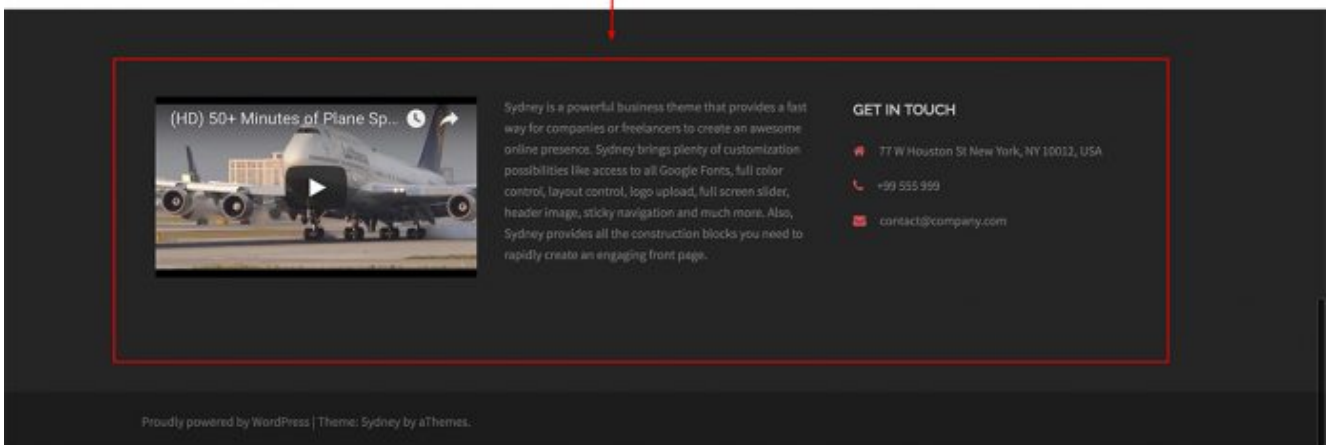
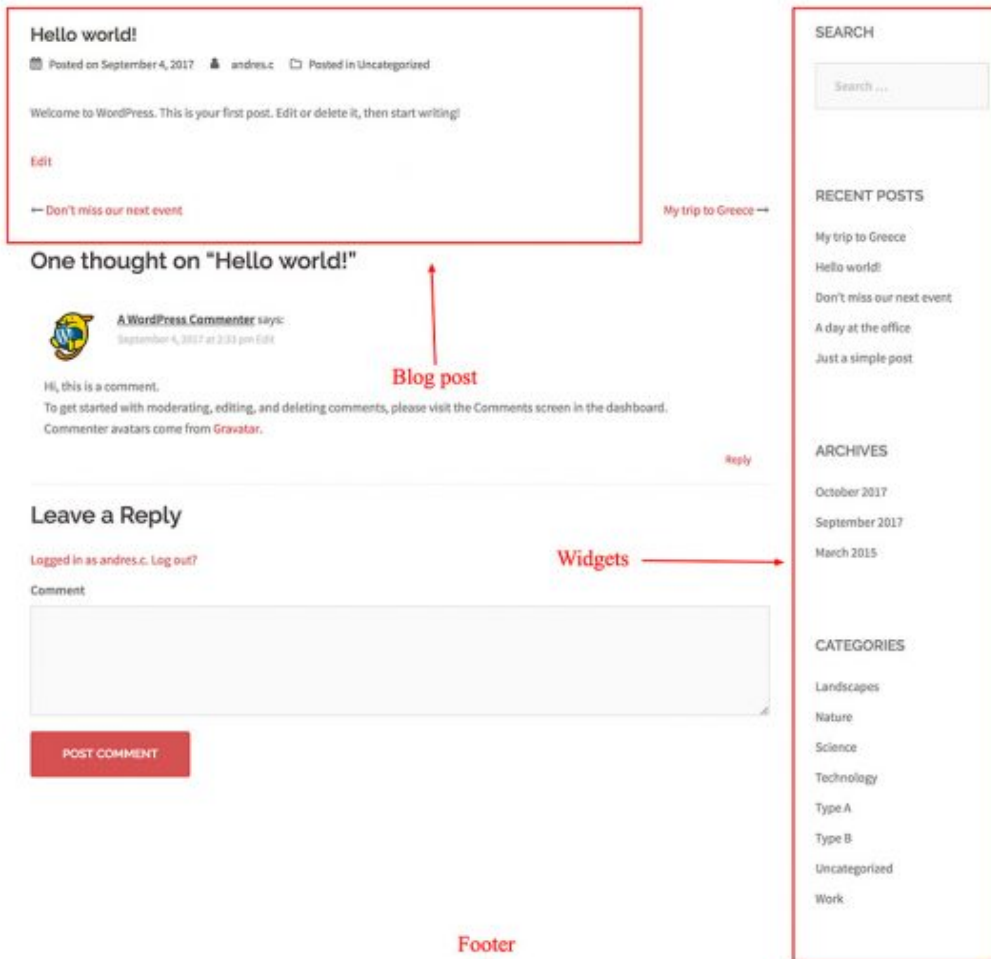
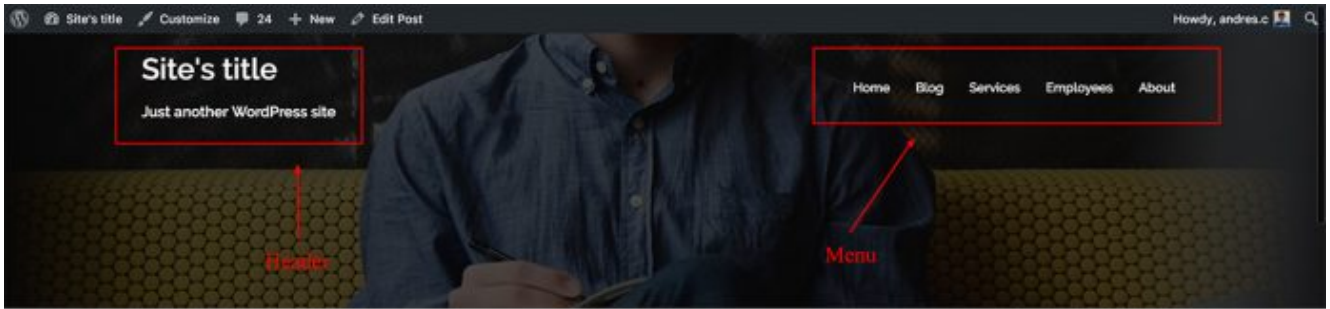
- WPML-Kern-Plugin
- WPML-String-Übersetzungs-Add-on

Um sie zu erhalten, gehen Sie zu Ihrem [WPML-Konto](#) und klicken Sie auf **Downloads** .

Natürlich können Sie sich gerne über die anderen Komponenten des [WPML-Kerns und der Addons informieren](#) .

2. Fallstudie – Sydney-Thema

Schauen wir uns dieses Beispiel einer Website an, die mit dem Sydney-Thema von aThemes erstellt wurde:

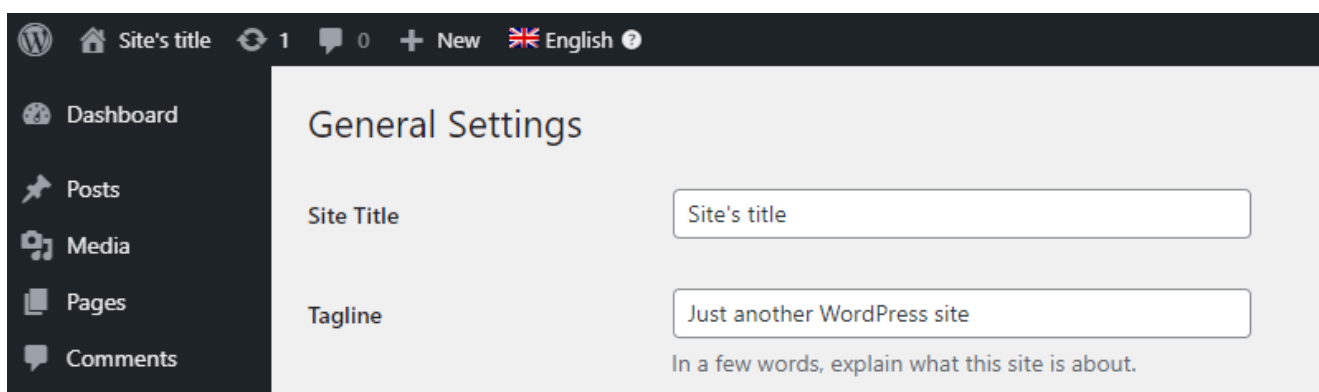


Beispiel für eine Website, die mit dem Sydney-Thema erstellt wurde

Während wir die verschiedenen Elemente dieser Site durchgehen, lernen wir, wie man Rich Themes oder Plugins mit WPML übersetzt.

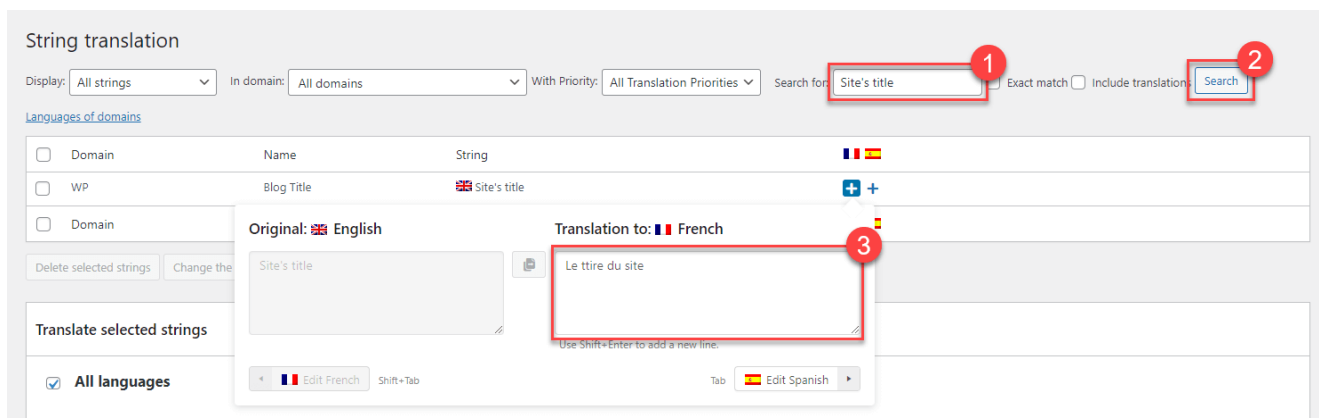
3. Übersetzen der Kopfzeile

Der Header des Themes oder Plugins stammt aus den allgemeinen WordPress-Einstellungen, wie in der folgenden Abbildung dargestellt:



Seite „Allgemeine WordPress-Einstellungen“

Da es nicht Teil eines Beitrags ist, müssen wir die WPML verwenden [String Translation](#) von , um es zu übersetzen. Gehen Sie zur **WPML zB → String Translation** Seite , suchen Sie den String anhand seines Inhalts („Site-Titel“) und übersetzen Sie ihn.



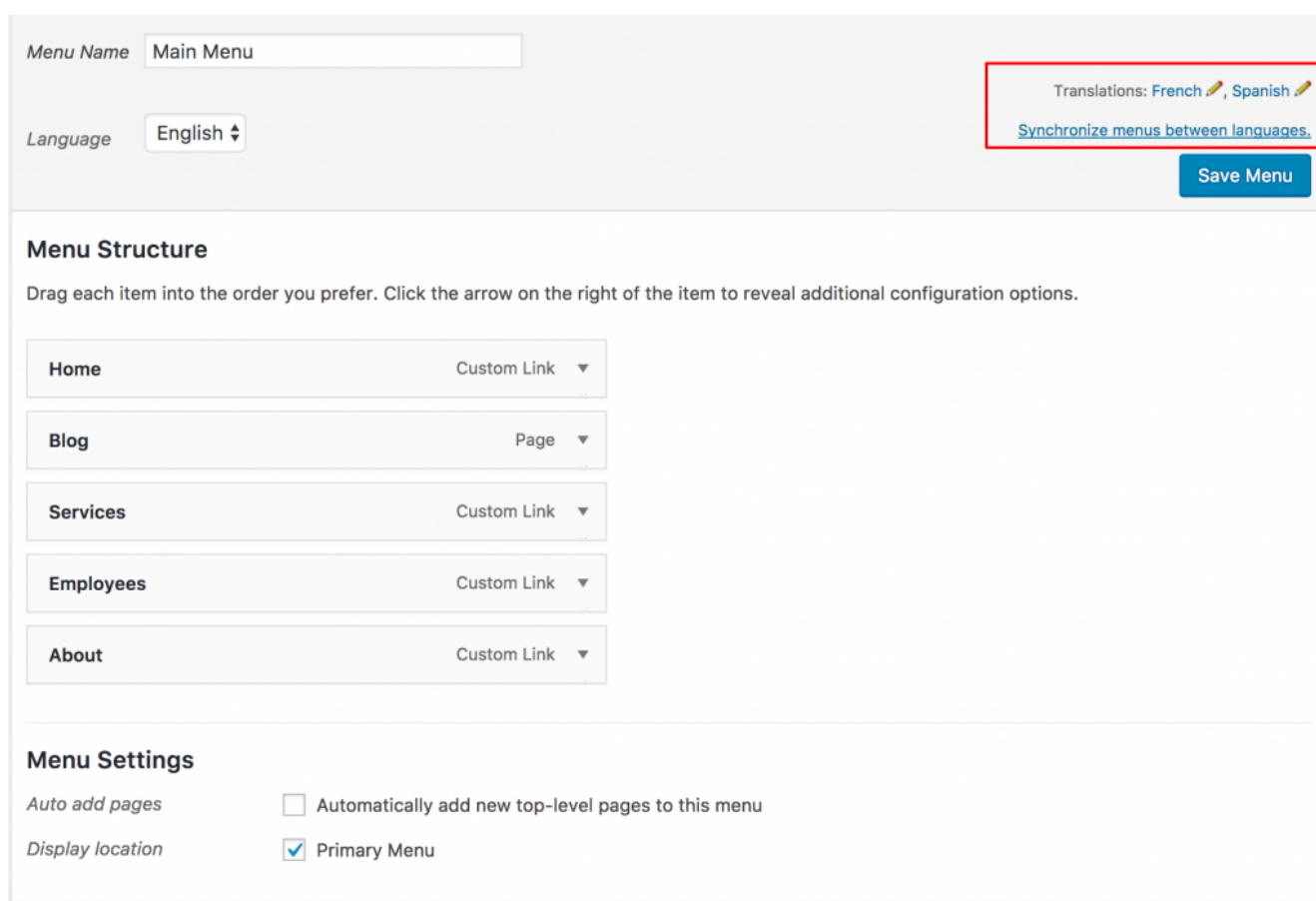
Übersetzen anderer Texte, die aus dem Thema und den Plugins stammen, mit der String-Übersetzungsseite von WPML

Sie müssen nichts im Theme oder Plugin tun, um dies zu

erreichen. WPML verarbeitet es automatisch. Wir haben gezeigt, wie Sie den Site-Namen übersetzen, da wir die String-Übersetzung von WPML für viele andere Elemente in Ihrem Theme oder Plugin verwenden werden.

4. Menüs übersetzen

WPML übersetzt WordPress-Menüs, ohne dass Sie etwas tun müssen. Solange Ihr Theme das Standard-WordPress-Menüsystem verwendet, übersetzt WPML die Menüs.



The screenshot shows the WordPress menu editor interface. At the top, the 'Menu Name' is 'Main Menu' and the 'Language' is 'English'. A red box highlights the translation options: 'Translations: French', 'Spanish', and a link 'Synchronize menus between languages.'. A 'Save Menu' button is visible. Below this is the 'Menu Structure' section with a list of menu items: 'Home' (Custom Link), 'Blog' (Page), 'Services' (Custom Link), 'Employees' (Custom Link), and 'About' (Custom Link). At the bottom is the 'Menu Settings' section with two options: 'Auto add pages' (unchecked) and 'Display location' (checked, set to 'Primary Menu').

Automatische Übersetzung von Menüs, die das Standard-WordPress-Menüsystem verwenden

Wenn Sie zur **Erscheinungsbild** → **Menüs gehen** Seite, sehen Sie die Steuerelemente für die Menüübersetzung von WPML. Erfahren Sie mehr über das [Übersetzen von Menüs, um](#) zu sehen, wie dies funktioniert.

5. Übersetzen der Dienste von Sydney Theme und anderer benutzerdefinierter Post-Typen

Das Beispielthema, das wir uns ansehen, verwendet einen benutzerdefinierten Beitragstyp, „Dienste“. Es ist einfach, es mit WPML zu übersetzen, mit wenig Konfiguration. Dies gilt auch für die Beitragstypen „Testimonials“, „Kunden“ und „Projekte“.

Wir müssen WPML mitteilen, dass der benutzerdefinierte Beitragstyp „Services“ mehrsprachig ist. Dies bedeutet, dass WPML Benutzern anbietet, es zu übersetzen.

WPML ermöglicht drei Modi für übersetzbare Inhalte:

- Übersetzbar, nur übersetzte Elemente anzeigen
- Übersetzbar, Übersetzung verwenden, falls verfügbar oder Fallback auf Standardsprache
- Nicht übersetzbar

Dies gibt Ihnen große Flexibilität bei der Entscheidung, welche Inhalte übersetzt werden und wie die Übersetzungen angezeigt werden (oder nicht).

Wir möchten die „Dienstleistungen“ übersetzen, damit wir unser Angebot in verschiedenen Sprachen zeigen können.

Navigieren Sie zu **WPML → Einstellungen** und scrollen Sie nach unten zu **Post Types Translation** . Wählen Sie eine von zwei **übersetzbaren** in den Zeilen für die benutzerdefinierten Beitragstypen, die übersetzbar sein sollen, Optionen aus, und klicken dann auf **Sie Speichern** .

Post Types Translation

Post types	Translatable only show translated items	Translatable use translation if available or fallback to default language	Not translatable	Translate automatically
Posts (post)	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="checkbox"/>
Pages (page)	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="checkbox"/>
Services	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="checkbox"/>
Employees	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="checkbox"/>
Testimonials	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="checkbox"/>
Clients	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="checkbox"/>

[Save](#)

Auswählen, welche benutzerdefinierten Beitragstypen übersetzbar sein sollen

6. Übersetzen von benutzerdefinierten Theme- oder Plugin-Elementen

Viele Themes und Plugins haben einzigartige Funktionen, die Texte in der speichern `wp_options` Tabelle. Die **Folienbeschriftungsfunktion** von Sydney macht genau das.

The screenshot displays the configuration interface for the Sydney theme's slide. The left sidebar contains a 'FIRST SLIDE' section with an image upload area and two text input fields: 'Title for the first slide' (containing 'Welcome to Sydney') and 'Subtitle for the first slide' (containing 'Feel free to look around'). The main preview area shows the slide content: 'Site's title' (Just another WordPress site), a navigation menu (Home, Blog, Services, Employees, About), and the main text 'Welcome to Sydney' with the subtitle 'Feel free to look around' and a 'CLICK TO BEGIN' button. Red arrows indicate the mapping between the text in the preview and the input fields in the sidebar.

Übersetzen von Texten, die in der Tabelle `wp_options` enthalten sind

Das Theme oder Plugin speichert diese Texte in der `wp_options` Tabelle und wir müssen WPML anweisen, sie zu übersetzen. Wir fügen diese Informationen der [Sprachkonfigurationsdatei hinzu](#). Dort teilen wir WPML mit, welche Einträge in der `wp_options` Tabelle übersetzt werden müssen.

Diese Technik ist gut, wenn die Tasten für die Optionen festgelegt sind (wie in den meisten Themes). Wenn Ihr Theme eine Reihe von Einträgen verwendet – die mit Benutzereingaben wachsen können – müssen Sie diese Einträge dynamisch registrieren. Verwenden Sie dazu die WPML [API-Funktionen](#) von .

6.1 Übersetzungen Ihres Plugins hinzufügen

Um Ihrem Plugin Übersetzungen hinzuzufügen, empfehlen wir Ihnen, einen erstellen **Sprachenordner** im Hauptordner Ihres Plugins zu und alle Ihre **.mo- abulegen** Dateien dort .

Dann muss WordPress sie laden und korrekt mit der Textdomäne deines Plugins verknüpfen. Dazu müssen Sie sie mit einer [load_plugin_textdomain- Theme- aufrufen](#) Funktion direkt nach dem Header Ihrer /Plugin-Datei . Dieser Code sollte in einer Datei abgelegt werden, die sich im Stammordner Ihres Plugins befindet:

7. Den Post-Body übersetzen

Mit WPML können Benutzer Inhalte problemlos übersetzen. Die Nachbearbeitungsbildschirme enthalten die Übersetzungssteuerelemente von WPML, mit denen Sie neue Übersetzungen erstellen und vorhandene bearbeiten können.

Hinzufügen von Übersetzungen bei Verwendung des Blockeditors

Language of this post

English ▾

Translation Priority

Optional ▾ [edit terms](#)

Translate this Document

Use WPML's Translation Editor

	Translate	Duplicate
French	+	<input type="checkbox"/>
German	+	<input type="checkbox"/>

Media attachments

Duplicate uploaded media to translations

Duplicate featured image to translations

Hinzufügen von Übersetzungen bei Verwendung des Classic-Editors

Language ▲

Language of this page

English ▼

Translation Priority

Optional ▼ [edit terms](#)

Translate this Document

	Translate	Duplicate
French	+	<input type="checkbox"/>
Spanish	+	<input type="checkbox"/>

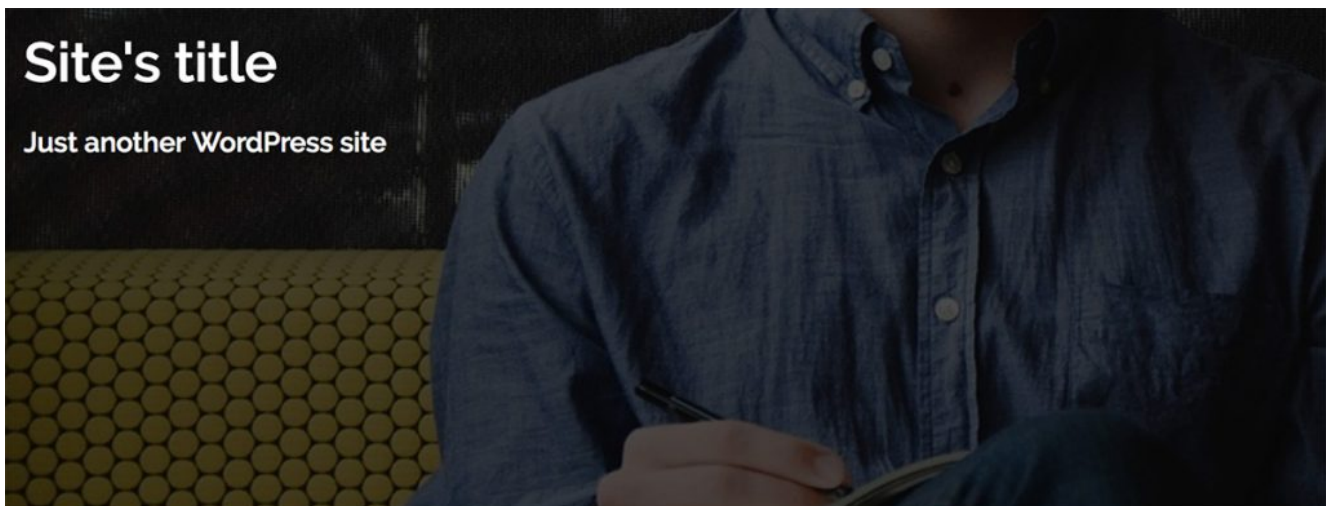
Media attachments

Duplicate uploaded media to translations

Duplicate featured image to translations

Sie müssen nichts im Theme oder Plugin tun, um dies zu erreichen. Das Übersetzen von Inhalten ist eine Kernfunktion von WPML.

Was Sie jedoch überprüfen müssen, ist, dass jeder Text, den Ihr Design der Ausgabe hinzufügt, übersetzbar ist. Beispielsweise sollten die hervorgehobenen Texte in diesem Screenshot in **gettext**- Aufrufe eingeschlossen werden.



Hello world!

Posted on September 4, 2017 by andres.c Posted in Uncategorized

Welcome to WordPress. This is your first post. Edit or delete it, then start writing!

Edit

Sicherstellen, dass der Text, den Ihr Design der Ausgabe hinzufügt, übersetzbar ist

Wenn Sie **gettext zum ersten Mal verwenden**, erfahren Sie mehr darüber in unseren [FAQ zur Übersetzung von Thementexten](#).

8. Widgets übersetzen

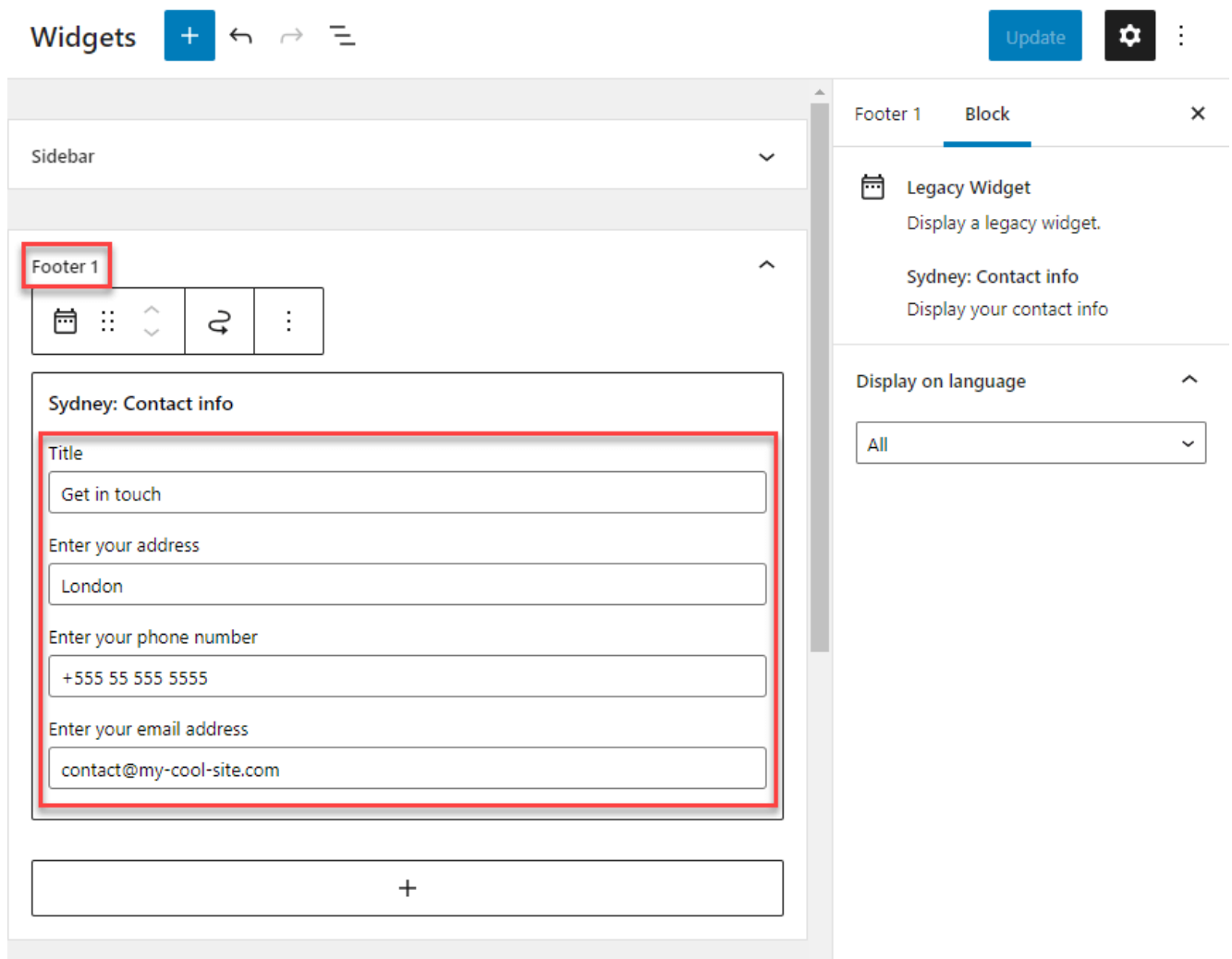
Mit WPML können Benutzer den Inhalt von Text-Widgets übersetzen. Es übersetzt auch die Titel aller anderen Widgets. Wenn Ihr Theme oder Plugin eigene benutzerdefinierte Widgets erstellt, stellen Sie sicher, dass Sie deren Titel durch die Standard-WordPress-Filter leiten. Der Code wird wie folgt aussehen:

```
1         function widget( $args, $instance )
2         { extract($args); $title = apply_filters(
3           'widget_title', empty($instance['title']) ? '' :
4             $instance['title'], $instance );}
```

Auf diese Weise ermöglicht WPML den Benutzern, die Titel Ihrer Widgets über den Bildschirm „String Translation“ zu übersetzen. Weitere Informationen finden Sie auf der Seite zum [Übersetzen von Widgets](#) .

9. Übersetzen der Fußzeile

Wie viele andere Themes verwendet auch dieses Widgets, um die Fußzeilentexte zu speichern.



The screenshot shows the WordPress widget editor interface. At the top, there is a 'Widgets' header with a plus sign, navigation arrows, and a menu icon. On the right, there are 'Update' and 'Settings' buttons. The main area is divided into two panes. The left pane, titled 'Sidebar', shows a 'Footer 1' widget selected and highlighted with a red box. Below the widget title is a control bar with icons for adding, deleting, and moving widgets. The main content area of the widget is also highlighted with a red box and contains the following text and input fields:

- Title: Get in touch
- Enter your address: London
- Enter your phone number: +555 55 555 5555
- Enter your email address: contact@my-cool-site.com

The right pane, titled 'Footer 1 Block', shows the widget's configuration options. It includes a 'Legacy Widget' section with the text 'Display a legacy widget.' and 'Sydney: Contact info' with the text 'Display your contact info'. Below this is a 'Display on language' section with a dropdown menu set to 'All'.

Widget-Text mit der String Translation von WPML übersetzen
WPML ermöglicht es Benutzern, die Titel und Inhalte Ihrer Widgets über den String Translation-Bildschirm zu übersetzen.

10. Erstellen einer Sprachkonfigurationsdatei für Ihr Theme

Wenn Sie fertig sind und Ihr Theme vollständig WPML-kompatibel ist, möchten Sie es wahrscheinlich anderen so einfach wie möglich machen, mehrsprachige Websites mit dem Theme zu betreiben.

Erstellen Sie eine [Sprachkonfigurationsdatei](#) für Ihr Theme und speichern Sie sie im Stammverzeichnis des Themes. Die Datei teilt WPML mit, welche benutzerdefinierten Beitragstypen, Taxonomien, Felder und Optionen übersetzbar sind.

Diese winzige XML-Datei spart Ihren Kunden Stunden und ermöglicht es ihnen, mehrsprachige Websites ohne Aufwand zu betreiben. Ihre Kunden werden alles zum Laufen bringen, ohne dass Sie konfigurieren müssen, was übersetzt werden soll. Dies hilft Ihnen auch, Support-Arbeit zu sparen.

11. GetText-Unterstützung für hartcodierte Zeichenfolgen

Obwohl dies nicht unbedingt mit WPML zusammenhängt, sollten Sie daran denken, alle Strings in Ihrem Theme in GetText-Aufrufe einzuschließen. Das bedeutet, dass hartcodierte Texte in der richtigen Sprache auf der Site erscheinen. Sie sollten dies sowohl für die Theme- (oder Plugin-) Texte tun, die im WordPress-Admin-Bereich erscheinen, als auch für die Texte, die im Frontend angezeigt werden.

Alle WordPress-Themes und -Plugins, egal ob sie mehrsprachig sind oder nicht, sollten GetText verwenden, um hartcodierte Zeichenfolgen zu übersetzen. Die WordPress-Standardthemen verwenden GetText sehr genau und sind eine großartige Referenz, wenn Sie gerade erst damit anfangen.

Das Umbrechen von Texten in GetText-Aufrufen ist ein großes Thema für sich. Wenn Sie damit nicht vertraut sind oder Hilfe beim Debuggen der Lokalisierung Ihres Codes benötigen, sehen Sie sich diese an:

- [Anleitung zur Verwendung von GetText](#)
- [Debugging-Tipps für GetText](#)

WPML lässt sich vollständig in GetText integrieren. Sobald Sie hartcodierte Strings in GetText-Aufrufe verpackt haben, können Sie sie direkt vom WordPress-Administrator über den String-Übersetzungsbildschirm von WPML übersetzen. WPML kann auch .po- exportieren und importieren, **sodass** Dateien für Sie die Strings Ihres Themes alle aus WordPress heraus übersetzen können.

12. WooCommerce-Kompatibilität

Themen, die einen WooCommerce-Bereich enthalten, müssen möglicherweise einigen anderen Vorschlägen folgen. Hauptsächlich sollten Sie **get_options verwenden** , um die IDs von WooCommerce-Seiten zu erhalten. Schauen Sie sich unseren Leitfaden an [, um WooCommerce-Themes mehrsprachig und währungsbereit zu machen](#) .

13. Tipps und Debugging

Wenn Sie bereits mit den WPML-Grundlagen vertraut sind und an der Kompatibilität mit Ihrem Theme oder Plugin arbeiten, werfen Sie einen Blick in den [Debugging-Theme-Kompatibilitätsleitfaden](#) . Sie finden Tipps und Ideen, wie Sie Ihr Theme mit WPML kompatibel machen.

Wenn Sie unsere Hilfe benötigen, melden Sie sich einfach. Besuchen Sie unser [technisches Support-Forum](#) , erklären Sie, was nicht stimmt und wir helfen Ihnen weiter.