

Les traductions dans PrestaShop 1.5

Table of contents

- [Les traductions dans PrestaShop 1.5](#)
 - [Traduction du front-office](#)
 - [Traduction du back-office](#)
 - [Traduction des messages d'erreur](#)
 - [Traduction des noms de champs](#)
 - [Traduction des modules installés](#)
 - [Traduction des fichiers PDF](#)
 - [Traduction de modèles de courriers](#)
 - [Objet du message](#)
 - [Modèles de message](#)
 - [Sprintf](#)
 - [Dans les fichiers PHP](#)
 - [Dans les fichiers modèles \(nouveau 1.5\)](#)

Les traductions dans PrestaShop 1.5

PrestaShop dispose de plusieurs types de traduction :

- Traduction du front-office
- Traduction du back-office
- Traduction des messages d'erreur
- Traduction des noms de champs
- Traduction des modules installés
- Traduction des fichiers PDF
- Traduction des modèles de courriers

Pour chaque type de traduction, l'outil de traduction interne à PrestaShop analyse un jeu spécifique de dossiers afin de récupérer toutes les chaînes traduisibles qu'ils contiennent, et de les présenter au traducteur.

Traduction du front-office

Les dossiers suivants sont analysés :

- `/themes/nom_du_theme/`
- `/themes/nom_du_theme/override/` (nouveau 1.5)

Les chaînes traduisibles utilisent la syntaxe Smarty suivante :

```
{l s='There are no products in this category.'}
```

Deux options peuvent être ajoutées à cette syntaxe :

- `"js=1"` : indique que cette chaîne peut être contenue dans du code JavaScript.
- `"sprintf='any string or number'"` : voir les exemples dans la section "Sprintf" de ce document (nouveau 1.5).

Toutes les traductions front-end sont stockées dans le fichier suivant, où `iso_code` est le code ISO 3166-1 (http://www.iso.org/iso/country_codes.htm) pour la langue utilisée pour traduire les chaînes (de, fr, en, it, es, etc.) :

- `/themes/name_of_the_theme/lang/iso_code.php`

Les traductions sont stockées dans un tableau PHP nommé `$_LANG`, et prenant la forme suivante, en utilisant un hash MD5 de la chaîne :

- `$_LANG['address_19f823c6453c2b1ffd09cb715214813d'] = 'Champ requis';`

La clé d'identification est construite en combinant le nom du fichier de modèle duquel la chaîne est tirée, un signe souligné, et le hash MD5 de la chaîne elle-même.

Si une même chaîne apparaît à différents endroits du contrôleur, elle apparaîtra aussi souvent dans l'outil de traduction.

Au contraire, si une même chaîne apparaît tant dans le contrôleur que dans le modèle, elle n'apparaîtra qu'une fois pour ce contrôleur.

Traduction du back-office

Les dossiers suivants sont analysés :

- /admin/
 - **Fichiers spécifiques** : header.inc.php, footer.inc.php, index.php, login.php, password.php, et functions.php
- /admin/themes/ (nouveau 1.5)
 - **Tous les fichiers modèles de tous les sous-dossiers sont analysés.**
- /controllers/admin/ (nouveau 1.5)
- /override/controllers/admin/ (nouveau 1.5)
- /classes/helper/ (nouveau 1.5)
- /classes/ (nouveau 1.5)
 - **Fichiers spécifiques** : AdminController.php, PaymentModule.php

Les chaînes traduisibles utilisent la syntaxe suivante :

- **Contrôleurs et classes** :

```
$this->l('Add new root category')
```

- **Les fichiers PHP à la racine du dossier /admin** :

```
translate('Customer name:')
```

- **Modèles du back-office** :

```
{l s='Add tag'}
```

Trois options peuvent être ajoutées à cette syntaxe :

- "js=1" : indique que cette chaîne peut être contenue dans du code JavaScript.
- "slashes=1" : vous permet d'ajouter des backslash (\) à une chaîne.
- "sprintf='any string or number'" : voir les exemples dans la section "Sprintf" de ce document (nouveau 1.5).

Toutes les traductions back-end sont stockées dans le fichier suivant, où `iso_code` est le code ISO 3166-1 (http://www.iso.org/iso/country_codes.htm) pour la langue utilisée pour traduire les chaînes (de, fr, en, it, es, etc.) :

- /translations/**iso_code**/admin.php

Les traductions sont stockées dans un tableau PHP nommé `$_LANGADM`, et prenant la forme suivante, en utilisant un hash MD5 de la chaîne :

- `$_LANGADM['AdminAccess151648106e4bf98297882ea2ea1c4b0e'] = 'Mise à jour réussie';`

La clé d'identification est construite en combinant le nom du contrôleur duquel la chaîne est tirée et le hash MD5 de la chaîne elle-même.

Notez que :

- Tous les modèles contenus dans le dossier `/admin/themes/default/templates/controllers/` sont préfixés avec "Admin" puis le nom du fichier.
- Tous les fichiers du dossier `/classes/helper/` sont préfixés avec "Helper".
- Tous les fichiers à la racine de `/admin/themes/default/templates/` sont préfixés avec "AdminController".

Si une même chaîne apparaît à différentes endroits du contrôleur, elle apparaîtra aussi souvent dans l'outil de traduction.

Au contraire, si une même chaîne apparaît tant dans le contrôleur que dans le modèle, elle n'apparaîtra qu'une fois pour ce contrôleur.

Traduction des messages d'erreur

Les dossiers suivants sont analysés :

- `/` : tous les fichiers à la racine du site.
- `/classes/`
- `/controllers/`
- `/override/classes/`
- `/override/controllers/`
- `/admin/`
- `/admin/tabs/`
- `/modules/` : tous les fichiers de tous les modules sont analysés.
- `/controllers/admin/` (nouveau 1.5)
- `/controllers/front/` (nouveau 1.5)
- `/override/controllers/admin/` (nouveau 1.5)
- `/override/controllers/front/` (nouveau 1.5)
- `/override/classes/` (nouveau 1.5)

Les chaînes traduisibles utilisent la syntaxe suivante :

```
Tools::displayError('An error occurred while creating archive.')
```

Par défaut, la chaîne est traitée par la fonction `htmlentities()`, afin de convertir à la volée les caractères spéciaux en entités HTML. Vous pouvez spécifier que `htmlentities()` ne doit pas être utilisé en ajoutant une option :

```
Tools::displayError('An error occurred while creating archive.', false)
```

Cette méthode ne doit pas être utilisée au sein de modules, car les chaînes de traduction seraient alors stockées dans le fichier `/translations/iso_code/errors.php` au lieu du fichier `/modules/nom_du_module/translations/iso_code.php`.

Toutes les traductions des messages d'erreur sont stockées dans le fichier suivant, où `iso_code` est le code ISO 3166-1 (http://www.iso.org/iso/country_codes.htm) pour la langue utilisée pour traduire les chaînes (de, fr, en, it, es, etc.) :

- `/translations/iso_code/errors.php`

Les traductions sont stockées dans un tableau PHP nommé `$_ERRORS`, et prenant la forme suivante :

- `$_ERRORS['00569f4db559dc94d9a954a090c22b85'] = 'Impossible d\'écrire';`

La clé d'identification est construite en utilisant le hash MD5 de la chaîne elle-même.

Si une chaîne apparaît dans plusieurs fichiers, elle n'apparaîtra qu'une fois dans le fichier de traduction.

Traduction des noms de champs

Les dossiers suivants sont analysés :

- `/classes/` et tous ses sous-dossiers

Les chaînes utilisées sont celles renvoyées par la méthode `getValidationRules()` de l'objet `ObjectModule`, pour chaque classe.

Toutes les traductions des messages d'erreur sont stockées dans le fichier suivant, où `iso_code` est le code ISO 3166-1 (http://www.iso.org/iso/country_codes.htm) pour la langue utilisée pour traduire les chaînes (de, fr, en, it, es, etc.) :

- `/translations/iso_code/fields.php`

Les traductions sont stockées dans un tableau PHP nommé `$_FIELDS`, et prenant la forme suivante :

- `$_FIELDS['Address_2df2ca5cf808744c2977e4073f6b59c8'] = 'Téléphone mobile';`

La clé d'identification est construite en combinant le nom de la classe d'où provient la chaîne, un signe souligné, et le hash MD5 de la chaîne elle-même.

Traduction des modules installés

En mode de production, seuls les modules installés peuvent être traduits. Si vous souhaitez traduire tous les modules, qu'ils soient installés ou non, vous devez activer le mode Debug de PrestaShop.

Les dossiers suivants sont analysés :

- `/modules/` et tous les sous-dossiers des divers modules.

Les chaînes traduisibles utilisent la syntaxe suivante :

- Pour tous les fichiers PHP :

```
$this->l('Add new root category')  
  
...or...  
  
$module->l('Add new root category')
```

- Pour les fichiers template :

```
{l s='Add tag'}
```

Trois options peuvent être ajoutées à cette syntaxe :

- "js=1" : indique que cette chaîne peut être contenue dans du code JavaScript.
- "mod='blockcms'": ajoute le nom du module
- "sprintf='any string or number'": voir les exemples dans la section "Sprintf" de ce document (nouveau 1.5).

La méthode `Tools::displayError()` ne doit pas être utilisée au sein de modules, car les traductions de chaînes seraient alors enregistrées dans le fichier `/translations/iso_code/errors.php` plutôt que dans le fichier `/modules/name_of_the_module/translations/iso_code.php`.

Depuis la version 1.5 de PrestaShop, vous pouvez traduire les modules en fonction du thème. De fait, la liste des fichiers de traduction est comme suit :

- `/modules/nom_du_module/translations/iso_code.php` : pour le thème par défaut.
- `/themes/nom_du_theme/modules/nom_du_module/translations/iso_code.php` : pour tout autre thème.

Les traductions sont stockées dans un tableau PHP nommé `$_MODULE`, et prenant la forme suivante :

- `$_MODULE['<{blockcms}prestashop>blockmobilecms_d1aa22a3126f04664e0fe3f598994014'] = 'Promotions';`

La clé d'identification est construite en combinant le nom du module d'où provient la chaîne originale (ex. : "blockcms), suivi du nom du thème (ex. : "prestashop"), puis du nom du fichier (ex. : "blockmobilecms"), d'un signe souligné, et enfin du hash MD5 de la chaîne elle-même.

Si une même chaîne apparaît dans plusieurs fichiers, elle n'apparaîtra qu'une fois dans l'outil de traduction.

Traduction des fichiers PDF

Les dossiers suivants sont analysés :

- `/classes/PDF.php`
- `/override/classes/PDF.php`
- `/classes/pdf/` (nouveau 1.5)
- `/override/classes/pdf/` (nouveau 1.5)
- `/pdf/` (nouveau 1.5)
- `/themes/nom_du_theme/pdf/` (nouveau 1.5)

Les chaînes traduisibles utilisent la syntaxe suivante :

- Pour les fichiers de classes, `ClassName::l()` :

```
HTMLTemplateInvoice::l('Invoice ' )  
  
...ou...  
  
HTMLTemplateDeliverySlip::l('Delivery')
```

- Pour les modèles :

```
{l s='Delivery Address'}
```

Deux options peuvent être ajoutées à cette syntaxe :

- "pdf=true" : ajoute le nom du module.
- "sprintf='any string or number'" : voir les exemples dans la section "Sprintf" de ce document (nouveau 1.5).

Toutes les traductions des messages d'erreur sont stockées dans le fichier suivant, où `iso_code` est le code ISO 3166-1 (http://www.iso.org/iso/country_codes.htm) pour la langue utilisée pour traduire les chaînes (de, fr, en, it, es, etc.) :

- `/translations/iso_code/pdf.php`

Dans la version 1.5

Si vous utilisez le thème par défaut, les traductions sont stockées au sein des traductions par défaut : `translations/iso_code/pdf.php`

Si vous utilisez tout autre thème, les traductions sont stockées dans le dossier du thème : `themes/votre_theme/pdf/lang/iso_code.php`

Les traductions sont stockées dans un tableau PHP nommé `$_LANGPDF`, et prenant la forme suivante :

- `$_LANGPDF['PDF_invoicecb5efbba6a6babef9082ca6976928ca7'] = 'Bon de livraison n°';`

La clé d'identification est construite en combinant "PDF_invoice" (v1.4) ou "PDF" (v1.5) avec le hash MD5 de la chaîne elle-même.

Si une même chaîne apparaît à différents endroits du contrôleur, elle apparaîtra aussi souvent dans l'outil de traduction.

Au contraire, si une même chaîne apparaît tant dans le contrôleur que dans le modèle, elle n'apparaîtra qu'une fois pour ce contrôleur.

Traduction de modèles de courriers

Quand il s'agit de toucher aux modèles de courriers électronique, il existe deux types de traduction : le modèle lui-même, et l'objet (ou le titre) du courrier. Ce sont deux choses totalement différentes.

Objet du message

Les dossiers suivants sont analysés :

- `/classes/`
- `/controllers/`
- `/admin/`
- `/admin/tabs/`
- `/modules/nom_du_module/` : si un module contient un dossier `/mails/`, tous les fichiers du module sont analysés.
- `/controllers/admin/` (nouveau 1.5)
- `/controllers/front/` (nouveau 1.5)
- `/override/classes/` (nouveau 1.5)
- `/override/controllers/admin/` (nouveau 1.5)

- `/override/controllers/front/` (nouveau 1.5)

Les chaînes traduisibles utilisent la syntaxe suivante :

```
Mail::l('Your new admin password', (int)$id_lang)
```

Le second paramètre est obligatoire. S'il est oublié, le message sera envoyé avec un objet qui n'est pas dans la langue par défaut de la boutique.

Toutes les traductions de courriers sont stockées dans les fichiers suivants, où `iso_code` est le code ISO 3166-1 (http://www.iso.org/iso/country_codes.htm) pour la langue utilisée pour traduire les chaînes (de, fr, en, it, es, etc.) :

- `/mails/iso_code/lang.php` : pour la traduction par défaut.
- `/themes/nom_du_theme/mails/iso_code/lang.php` : pour les traductions de chaque thème.

Les traductions sont stockées dans un tableau PHP nommé `$_LANGMAIL`, et prenant la forme suivante :

- `$_LANGMAIL['Welcome!'] = 'Bienvenue !';`

The identification key is built by using the original string directly.

La clé d'identification est construite en la chaîne elle-même.

Modèles de message

Les dossiers suivants sont analysés :

- `/mails/iso_code/` : pour les modèles par défaut.
- `/themes/nom_du_theme/mails/iso_code/` : pour les modèles de message de chaque thème.

Les modèles de message traduits sont ceux disponibles en anglais. Par exemple :

- Si le fichier `/mails/en/account.html`, alors il sera traduisible pour chaque langue activée dans le back-office
- Si le fichier `/themes/nom_du_theme/mails/en/account.txt` n'existe pas, alors il ne sera pas traduisible via le back-office.

Sprintf

La fonction `sprintf()`, disponible par défaut dans PHP, doit être utilisée correctement dans tous les fichiers de PrestaShop.

Dans les fichiers PHP

Bonne manière :

```
sprintf($this->l('Empty string found, please edit: "%s"', $string), $string);  
sprintf(Tools::displayError('Please create a "%1$s.php" file in "%2$s"', $string1, $string2)
```

Mauvaise manière :

```
$this->l('Empty string found, please edit:').''.$string.''';  
Tools::displayError('Please create a').' '.$string1.'.php" '.Tools::displayError('file in').''.$string2.''';
```

Dans les fichiers modèles (nouveau 1.5)

Bonne manière :

```
{l s='renamed the /admin folder (e.g. /admin123%d)' sprintf=$number}  
{l s='renamed the /%1$s folder (e.g. /admin123%2$d)' sprintf=[$string, $number]}
```

Mauvaise manière :

```
{l s='renamed the /admin folder (e.g. /admin123'){$number}}  
{l s='renamed the' } /{$string} {l s='folder (e.g. /admin123'){$number}}
```