Language Files

Translate the language files to your locale.

Writing Translations

Translatable content locations

Language strings are stored in files within the Lang/<locale> directory. The locale is the code as set when you created the language.

```
/addons /Languages /English /Lang /en core.php
/Spanish /Lang /es core.php
```

File structure

Each language files must return an array of language strings:

File Encoding
Please ensure that all of your language files are saved with UTF-8 encoding otherwise the help desk may not function correctly.

```
<?php return [ 'product_name' => 'SupportPro',];
```

We recommend copying the language files from the included English language pack and then translating all the strings within those files.

Updating Translations

Language files are cached for performance reasons. If you make changes to a translation file after the language has been enabled, you must manually clear the system cache to see the changes in some parts of the application. This can be done by browsing to Settings -> Utilities -> System Cleanup or using the [redacted] command via CLI.

Using Translations

Language strings can be accessed from any part of the application using Lang::get() function. If you need to access language files from within a template the syntax changes slightly to Lang.get(). For example, if you would like to access the product_name language string in the core.php file from within a PHP file:

```
echo Lang::get('core.product_name');
```

Alternatively, from within a template file:

```
{{ Lang.get('core.product_name') }}
```

Variable Replacement

If you need to dynamically control words within language strings, this is possible using placeholders. All placeholders are prefixed by the : character.

```
<?phpreturn [    'product_name' => 'SupportPro,    '
welcome'     => 'Welcome to :product_name',];
```

We can then assign the placeholder value when accessing the

language string:

```
echo Lang::get('core.welcome', [ 'product_name' => Lang::get( 'core.product_name') ]); // Welcome to SupportPro
```

Pluralisation

Often we make use of singular and plural forms of language strings. Pluralisation is possible via the pipe (|) character:

```
<?phpreturn [    'product' => 'Product|Products'];
```

Depending on whether we want to access the singular of plural form, the string can be accessed using the Lang::choice() function. The second parameter specifies which form we would like to retrieve:

```
echo Lang::choice('core.product', 1); // Productecho Lang::choice( 'core.product', 2); // Products
```

Next Section

Online URL: https://docs.supportpro.vn/article/language-files-195.html