

# Languages

## Overview

Language translation files are placed inside the appropriate language languages directory within a template.

```
/templates
.. /foo
.. .. /languages
.. .. .. /{LanguageName}
.. .. .. .. {LanguageName}.tpl_{TemplateName}.ini
```

## Setup

As previously mentioned, language files are setup as key/value pairs. A key is used within the template's code and the translator retrieves the associated string for the given language. The following code is an extract from a typical language file.

```
; Template Test (en-US)
TPL_TEST_HERE_IS_LINE_ONE = "Here is line one"
TPL_TEST_HERE_IS_LINE_TWO = "Here is line two"
TPL_TEST_MYLINE = "My Line"
```

Translation keys can be upper or lowercase or a mix of the two and may contain underscores but no spaces. HUBzero convention is to have keys all uppercase with words separated by underscores, following a pattern of TPL\_{TemplateName}\_{Text} for naming. Adhering to this naming convention is not required but is strongly recommended as it can help avoid potential translation collisions.

See the [Languages](#) overview for details.

## Loading

The appropriate language file for a template is preloaded, as long as it follows the naming conventions detailed above, when the template is rendered. As such, no manual loading is necessary. However, if you wish to load an alternate language file, you can do so by calling

## LANGUAGES

---

`Lang::load($extension);`. The following example demonstrates, from a template layout, loading a language file for the component 'com\_test' out of the `/hub/app/languages` directory.

```
<?php
// No direct access
defined('_HZEXEC_') or die();

Lang::load('com_test', PATH_APP . '/languages');
?>
<html>
...
```

### Translating Text

Below is an example of accessing the translate helper:

```
<p><?php echo Lang::txt("TPL_TEST_MY_LINE"); ?></p>
```

Strings or keys not found in the current translation file will output as is.

See the [Languages](#) overview for details.