

# Languages

## Overview

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

```
/hubzero
  /language
    /{LanguageName}
      {LanguageName}.plg_{GroupName}_{PluginName}.ini
```

**Note:** Plugin language files contain data for both the front-end and administrative back-end.

## Setup

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

```
; Plugin - System - Test (en-US)
PLG_SYSTEM_TEST_HERE_IS_LINE_ONE = "Here is line one"
PLG_SYSTEM_TEST_HERE_IS_LINE_TWO = "Here is line two"
PLG_SYSTEM_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 `PLG_{PluginGroup}_{PluginName}_{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 plugin is **not** preloaded when the plugin is instantiated as many plugins may not have language files at all. As such, one must specifically load any file(s) if

## LANGUAGES

---

they are needed. This can be done by setting the `$_autoloadLanguage` property to true or by manually loading the desired language files by calling the `loadLanguage()` method. This method accepts the name of the plugin (e.g., `plg_{PluginGroup}_{PluginName}`) and an optional base path to start from (e.g., `PATH_APP`, `PATH_CORE`).

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

class plgSystemTest extends HubzeroPluginPlugin
{
    /**
     * Affects constructor behavior. If true, language files will be load
     ed automatically.
     *
     * @var boolean
     */
    protected $_autoloadLanguage = true;
}
```

Note that the string passed to the `loadLanguage()` method matches the pattern for the naming of the language file itself, minus the language prefix and file extension.

## Translating Text

Below is an example of accessing the translate helper:

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

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

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