



# Language Library

The Language Library provides handlers to retrieve language files and lines of text for purposes of internationalization.

In your revlgniter system folder you'll find one called **language** containing sets of language files. You can create your own language files as needed in order to display error and other messages in other languages.

Language files are typically stored in your **system/language** directory. Alternately you can create a folder called **language** inside your **application** folder and store them there. revlgniter will look first in your **application/language** directory. If the directory does not exist or the specified language is not located there revlgniter will instead look in your global **system/language** folder.

**Note:** Each language should be stored in its own folder. For example, the English files are located at: **system/language/english**

## Creating Language Files

Language files must be named with **Lang.lc** as the file extension. For example, let's say you want to create a file containing error messages. You might name it: **errorLang.lc**

Within the file you will assign each line of text to an array called **gLang** with this prototype:

```
put "The actual message to be shown" into gLang["language_key"]
```

**Note:** It's a good practice to use a common prefix for all messages in a given file to avoid collisions with similarly named items in other files. For example, if you are creating error messages you might prefix them with **error\_**

```
put "You must submit an email address" into gLang["error_email_missing"]
put "You must submit a URL" into gLang["error_url_missing"]
put "You must submit a username" into gLang["error_username_missing"]
```

## Loading A Language File

In order to fetch a line from a particular file you must load the file first. Loading a language file is done with the following code:

```
get rigLangLoadLang("filename", "language")
```

Where **filename** is the name of the file you wish to load (without the file extension), and **language** is the language set containing it (ie, english). If the second parameter is missing, the default language set in your **application/config/config.lc** file will be used.

## Fetching a Line of Text

Once your desired language file is loaded you can access any line of text using this function:

```
get rigLangLangLine("language_key")
```

Where **language\_key** is the array key corresponding to the line you wish to show.

## Using language lines as form labels

See the **rigLangLine()** function of the [Language helper](#).

## Auto-loading Languages

If you find that you need a particular language globally throughout your application, you can tell revlgniter to **auto-load** it during system initialization. This is done by opening the application/config/autoload.lc file and adding the language(s) to the autoload array.