Loader Library

Loader, as the name suggests, is used to load elements. These elements can be Libraries, Stacks, View files, Models, Helpers, Files, Language files, Custom Configuration files or LiveCode Builder (library) Extensions.

Note: This library is initialized automatically by the system so there is no need to do it manually.

The following handlers are available in this library:

rigLoaderLoadLibrary "libraryName", tConfig

This handler is used to load core libraries. Where **libraryName** is the name of the library you want to load.

For example, if you would like to send email with revIgniter, the first step is to load the email library within your controller:

```
rigLoaderLoadLibrary "Email"
```

Once loaded, the library will be ready for use.

Library files can be stored in subdirectories within the main "libraries" folder, or within your personal **application/libraries** folder. To load a file located in a subdirectory, simply include the path, relative to the "libraries" folder. For example, if you have a file located at:

```
libraries/flavors/Chocolate.lc
```

You will load it using:

```
rigLoaderLoadLibrary "flavors/Chocolate"
```

You may nest the file in as many subdirectories as you want.

Setting options

The second (optional) parameter allows you to optionally pass configuration settings. You will typically pass these as an array:

```
put "html" into tConfig["mailtype"]
put "utf-8" into tConfig["charset"]
put "1" into tConfig["priority"]

rigLoaderLoadLibrary "Email", tConfig
```

Config options can usually also be set via a config file. Each library is explained in detail in its own page, so please read the information regarding each one you would like to use.

rigLoadStack "stackName", tConfig, "behind / front"

Use this handler to load revIgniter stack files located at system/stacks as well as your own stacks which should be located at **application/stacks**. The first parameter is the name of the stack (the suffix can be omitted) you want to load. Use the optional second parameter to pass configuration settings in form of an array. The optional third parameter is used to define if the loaded stack should sit in front or behind the home stack, default is behind.

Like library files, stacks can be stored in subdirectories. Include the path, relative to the "stacks" folder to load a stack located in a subdirectory. Assuming your stack is located at:

```
stacks/flavors/Strawberry.livecode
```

Load the stack using:

```
rigLoadStack "flavors/Strawberry"
```

You may nest the stack in as many subdirectories as you want.

Setting options

The optional second parameter allows you to pass configuration settings. You will typically pass these as an array:

```
put "strawberry" into tConfig["flavor"]
put "big" into tConfig["size"]

rigLoadStack "Icecream", tConfig
```

Stack config options can usually also be set via a config file. Please read about using stacks in the Using Stacks chapter.

rigLoadView("fileName", TRUE/FALSE)

This function is used to load your View files. If you haven't read the Views section of the user guide it is recommended that you do since it shows you how this function is typically used.

The first parameter is required. It is the name of the view file you would like to load. Note: The .lc file extension does not need to be specified unless you use something other than .lc.

The second **optional** parameter lets you change the behavior of the function so that it returns data as a string rather than sending it to your browser. This can be useful if you want to process the data in some way. If you set the parameter to TRUE (boolean) it will return data. The default behavior is FALSE, which sends it to your browser. Remember to assign it to a variable if you want the data returned:

```
put rigLoadView("myfile", TRUE) into tString
```

rigLoadModel "modelName"

```
rigLoadModel "modelName"
```

If your model is located in a sub-folder, include the relative path from your models folder. For example, if you have a model located at application/models/blog/queries.lc you'll load it using:

```
rigLoadModel "blog/queries"
```

rigLoadDatabase("options", TRUE/FALSE, TRUE/FALSE)

This function lets you load the database library. The three parameters are **optional**. Please see the database section for more info.

rigLoadScaffolding "tableName"

This handler lets you enable scaffolding. Please see the Scaffolding section for more info.

rigLoadHelper "fileName"

This handler loads helper files, where **fileName** is the name of the file, without the Helper.lc extension.

rigLoadFile("filepath/filename", TRUE/FALSE)

This is a generic file loading function. Supply the filepath and name in the first parameter and it will open and read the file. By default the data is sent to your browser, just like a View file, but if you set the second parameter to TRUE (boolean) it will instead return the data as a string.

rigLoadLanguage "fileName", "language"

This handler is an alias of the language loading function: rigLangLoadLang()

rigLoadConfig "fileName"

This handler is an alias of the config file loading function: rigLoadConfigFile()

rigLoadExtension "folderName"

This handler loads LiveCode Builder extensions, where **folderName** is the name of a folder (containing a "module.lcm" file) located in application/extensions/ or in system/extensions/. Please follow the naming convention used by LiveCode Ltd. This means such a folder name should be composed of a reverse domain, followed by "library" and a library name all delimited by a period.