

☐ [revIgniter User Guide Version 2.1.7](#)

Search User Guide



1. [revIgniter Home](#)
2. [User Guide Home](#)
3. [HTML Table Library](#)
4. [Table of Contents Page](#)

- [revIgniter](#)
- [User Guide Home](#)
- [Table of Contents Page](#)
- [Basic Info](#)
- [Server Requirements](#)
- [License Agreement](#)
- [Change Log](#)
- [Credits](#)
- [Installation](#)
- [Downloading revIgniter](#)
- [Installation Instructions](#)
- [Upgrading from a Previous Version](#)
- [Troubleshooting](#)
- [Introduction](#)
- [Getting Started](#)
- [revIgniter at a Glance](#)
- [revIgniter Cheatsheets](#)
- [Supported Features](#)
- [Application Flow Chart](#)
- [Model-View-Controller](#)
- [HMVC - Extending MVC](#)
- [Architectural Goals](#)
- [General Topics](#)
- [revIgniter URLs](#)
- [Controllers](#)
- [Reserved Names](#)
- [Views](#)
- [Models](#)
- [Helpers](#)
- [Using revIgniter Libraries](#)
- [Creating Your Own Libraries](#)
- [Creating Core Libraries](#)
- [Using Stacks](#)
- [Hooks - Extending the Core](#)
- [Auto-loading Resources](#)
- [Common Handlers](#)
- [Scaffolding](#)
- [URI Routing](#)
- [Modules](#)
- [Extensions](#)
- [Error Handling](#)
- [Caching](#)
- [Profiling Your Application](#)
- [Managing Applications](#)
- [Security](#)
- [CLI](#)
- [Tutorial](#)
- [Simple Chat Application](#)
- [Library Reference](#)
- [Authentication Library](#)
- [Benchmarking Library](#)
- [Calendar Library](#)
- [Captcha Library](#)
- [Config Library](#)
- [Database Library](#)
- [Email Library](#)
- [Encryption Library](#)
- [File Uploading Library](#)
- [Form Validation Library](#)
- [FTP Library](#)
- [HTML Table Library](#)
- [Image Manipulation Library](#)
- [Input and Security Library](#)
- [jQuery Library](#)
- [Language Library](#)

- [Loader Library](#)
- [Modules Library](#)
- [Output Library](#)
- [Pagination Library](#)
- [Session Library](#)
- [Trackback Library](#)
- [URL Library](#)
- [User Agent Library](#)
- Helper Reference
- [Array Helper](#)
- [Asset Helper](#)
- [Cookie Helper](#)
- [Date Helper](#)
- [Download Helper](#)
- [Email Helper](#)
- [File Helper](#)
- [Form Helper](#)
- [Form-Mail Helper](#)
- [Galleria Helper](#)
- [HTML Helper](#)
- [JWT Helper](#)
- [Language Helper](#)
- [Markdown Helper](#)
- [OTP Helper](#)
- [QueryToJSON Helper](#)
- [QueryValues Helper](#)
- [SiteLinks Helper](#)
- [String Helper](#)
- [URL Helper](#)
- [XML Helper](#)

Table of Contents  
Close

# HTML Table Library

The Table Library provides handlers that enable you to auto-generate HTML tables from arrays or database result sets.

## Initializing the Library

Like most other libraries in revIgniter, the Table library is initialized in your controller using the *rigLoaderLoadLibrary* handler:

```
rigLoaderLoadLibrary "Table"
```

## Examples:

Here is an example showing how you can create a table from a multi-dimensional array. Note that the first array index will become the table heading (or you can set your own headings using the *rigSetTableHeading* handler described in the handler reference below).

Copy

```
rigLoaderLoadLibrary "Table"

put "Name" into tData[1][1]
put "Color" into tData[1][2]
put "Size" into tData[1][3]

put "Fred" into tData[2][1]
put "Blue" into tData[2][2]
put "Small" into tData[2][3]

put "Mary" into tData[3][1]
put "Red" into tData[3][2]
put "Large" into tData[3][3]

put "John" into tData[4][1]
put "Green" into tData[4][2]
put "Medium" into tData[4][3]

put rigGenerateTable(tData) into gData["testTable"]
```

Here is an example of a table created from a database query result. The Table library will automatically generate the headings based on the table names (or you can set your own headings using the *rigSetTableHeading* handler described in the handler reference below).

Copy

```
rigLoaderLoadLibrary "Table"

put rigDbQuery("SELECT * FROM mytable") into tQuery
put tQuery["resultarray"] into tResult

put rigGenerateTable(tResult) into gData["testTable"]
```

Here is an example showing how you might create a table using discrete parameters:

Copy

```
rigLoaderLoadLibrary "Table"

rigSetTableHeading "Name,Color,Size"

rigAddTableRow "Fred,Blue,Small"
rigAddTableRow "Mary,Red,Large"
rigAddTableRow "John,Green,Medium"

put rigGenerateTable() into gData["testTable"]
```

Here is the same example, except instead of individual parameters, arrays are used:

Copy

```
rigLoaderLoadLibrary "Table"

put "Name" into tHeading[1]
put "Color" into tHeading[2]
put "Size" into tHeading[3]
rigSetTableHeading tHeading

put "Fred" into tRow[1]
put "Blue" into tRow[2]
put "Small" into tRow[3]
rigAddTableRow tRow

put "Mary" into tRow[1]
put "Red" into tRow[2]
put "Large" into tRow[3]
rigAddTableRow tRow

put "John" into tRow[1]
put "Green" into tRow[2]
put "Medium" into tRow[3]
rigAddTableRow tRow

put rigGenerateTable() into gData["testTable"]
```

## Changing the Look of Your Table

The Table Library permits you to set a table template with which you can specify the design of your layout. Here is the template prototype:

Copy

```
put "<table border=" & quote & "0" & quote && "cellpadding=" & quote & "4" & quote && "cellspacing=" & quote & "0" & quote & ">" into tTmpl
put "<tr>" into tTmpl["headingRowStart"]
put "</tr>" into tTmpl["headingRowEnd"]
put "<th>" into tTmpl["headingCellStart"]
put "</th>" into tTmpl["headingCellEnd"]

put "<tr>" into tTmpl["rowStart"]
put "</tr>" into tTmpl["rowEnd"]
put "<td>" into tTmpl["cellStart"]
put "</td>" into tTmpl["cellEnd"]

put "<tr>" into tTmpl["rowAltStart"]
put "</tr>" into tTmpl["rowAltEnd"]
put "<td>" into tTmpl["cellAltStart"]
put "</td>" into tTmpl["cellAltEnd"]

put "</table>" into tTmpl["tableClose"]

rigSetTableTemplate tTmpl
```

**Note:** You'll notice there are two sets of "row" blocks in the template. These permit you to create alternating row colors or design elements that alternate with each iteration of the row data.

You are NOT required to submit a complete template. If you only need to change parts of the layout you can simply submit those elements. In this example, only the table

opening tag is being changed:

Copy

```
put "<table border=" & quote & "1" & quote && "cellpadding=" & quote & "2" & quote && "cellspacing=" & quote & "1" & quote && "class=" &
rigSetTableTemplate tTpl
```

## Handler Reference

### rigGenerateTable()

Returns a string containing the generated table. Accepts an optional parameter which can be an array like a database query result array.

### rigSetTableCaption

Permits you to add a caption to the table.

```
rigSetTableCaption "Colors"
```

### rigSetTableHeading

Permits you to set the table heading. You can submit an array or discrete params:

```
rigSetTableHeading "Name,Color,Size"
```

```
put "Name" into tHeading[1]
put "Color" into tHeading[2]
put "Size" into tHeading[3]
```

```
rigSetTableHeading tHeading
```

The second optional parameter is a boolean used to uppercase the first character of the table heading words. This is useful if you retrieve lowercase field names from a database table as table heading.

```
rigSetTableHeading tQuery["fieldnames"], TRUE
```

### rigAddTableRow

Permits you to add a row to your table. You can submit an array or discrete params:

```
rigAddTableRow "Blue,Red,Green"
```

```
put "Blue" into tRow[1]
put "Red" into tRow[2]
put "Green" into tRow[3]
```

```
rigAddTableRow tRow
```

### rigMakeTableColumns()

This function takes a one-dimensional array as input and creates a multi-dimensional array with a depth equal to the number of columns desired. This allows a single array with many elements to be displayed in a table that has a fixed column count. Consider this example:

```
put "one,two,three,four,five,six,seven,eight,nine,ten,eleven,twelve" into tList
split tList using comma
```

```
put rigMakeTableColumns(tList, 3) into tNewList
```

```
put rigGenerateTable(tNewList) into gData["testTable"]
```

# Generates a table with this prototype

```
<table border="0" cellpadding="4" cellspacing="0">
<tr>
<td>one</td><td>two</td><td>three</td>
</tr><tr>
<td>four</td><td>five</td><td>six</td>
</tr><tr>
<td>seven</td><td>eight</td><td>nine</td>
</tr><tr>
<td>ten</td><td>eleven</td><td>twelve</td></tr>
</table>
```

### rigSetTableTemplate

Permits you to set your template. You can submit a full or partial template.

```
put "<table border=" & quote & "1" & quote && "cellpadding=" & quote & "2" & quote && "cellspacing=" & quote & "1" & quote && "class=" & quote & "table" & quote && ">" & quote & "</table>" & quote && "
```

```
rigSetTableTemplate tTpl
```

### **rigSetEmptyCellVal**

Lets you set a default value for use in any table cells that are empty. You might, for example, set a non-breaking space:

```
rigSetEmptyCellVal "&nbsp;"
```

### **rigClearTableVars**

Lets you clear the table heading and row data. If you need to show multiple tables with different data you should call this function after each table has been generated to empty the previous table information. Example:

```
rigLoaderLoadLibrary "Table"

rigSetTableHeading "Name,Color,Size"

rigAddTableRow "Fred,Blue,Small"
rigAddTableRow "Mary,Red,Large"
rigAddTableRow "John,Green,Medium"

put rigGenerateTable() into gData["testTable"]

rigClearTableVars

rigSetTableHeading "Name,Day,Delivery"

rigAddTableRow "Fred,Wednesday,Express"
rigAddTableRow "Mary,Monday,Air"
rigAddTableRow "John,Saturday,Overnight"

put rigGenerateTable() into gData["newTestTable"]
```

Previous Topic: [FTP Library](#) :: [Top of Page](#) :: [User Guide Home](#) :: Next Topic: [Image Manipulation Library](#)

[revIgniter](#) :: Copyright © 2009 - 2020 :: [dimensionB Bitter u. Bitter GmbH](#)