



downloaded directly from CodeCanyon) is located. Select the plugin file and click on “Open” in your popup window.

4. Click on “Install Now”

5. Wait until WordPress finishes the upload and installation process.

6. Click on “Activate Now”

### Manual Upload via FTP:

1. Extract the ZIP file you received from CodeCanyon. The file should include a folder named “ts-advanced-tables-vc”

2. Using any FTP program, navigate to your WordPress installation and find the folder “wp-content”. Open that folder and (inside) find the folder “plugins”. Now open that folder and you should see a listing of all currently installed plugins for your WordPress installation.

3. Upload the complete plugin folder you extracted in step 1 into the “plugins” folder you just navigated to via FTP. Ensure, that the plugin folder is uploaded directly into the “wp-content/plugins” folder and not accidentally into one of the existing folders for one of the existing plugins.

4. Once the upload is finished, log into your WordPress admin section and go to the plugins section. You should now see “Tabenator - Advanced Tables for Visual Composer” listed as available plugin. All that is left is to activate it.

If you have problems installing the plugin, follow the step by step instructions you can find [here](#).

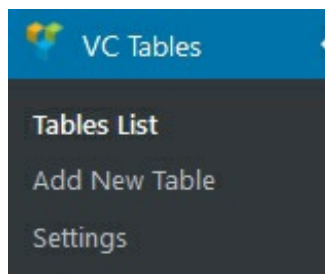
<input type="checkbox"/> Tabenator - Advanced Tables for Visual Composer	A plugin to easily create and display advanced tables and to use them within Visual Composer.
<a href="#">Activate</a>   <a href="#">Edit</a>   <a href="#">Delete</a>	Version 1.0.0   By Tekanewa Scripts   <a href="#">Visit plugin site</a>

## After Installation & Activations

If you have followed all these steps correctly, and your system passed the memory check, the plugin should now be installed and functional.

After you successfully activated the plugin, you should also see a new menu items in your WordPress backend, named “VC Tables” (see screenshot), giving you access to the settings page, table generator and shortcode generator.

**CONGRATULATIONS, because you are done with the installation!**



## Error Messages

Depending upon your individual WordPress setup and internal size limits, you might encounter some problems/errors when attempting to install or activate the plugin.

## Upload / Post Size Limits

*When using the standard WordPress routine, please ensure that your WordPress upload and post size limits are large enough in order to accommodate the size of the plugin file. Otherwise, you will have to upload the plugin via FTP or to increase the restricting limits first.*

Are you sure you want to do this?

If during the standard install routine you encounter a white page or a message asking you “Are you sure you want to do this?”, your upload / post size limits are most likely not sufficient for the file size and you should opt for the FTP method

If you have problems increasing your upload and/or post size limits, you might find more information [here](#).

## User Roles Manager

If the table element from this plugin does not show up in Visual Composer, when editing a page or post, it is most likely that you are using the so-called “User Roles Manager” that is part of Visual Composer itself (not part of this addon).

If you do use that native Visual Composer feature, you need to assign all newly added elements to the respective user roles that are actually allowed to use the elements.

You can access the “User Roles Manager” from within the settings page for Visual Composer itself; either by clicking on the sub-menu item “Role Manager”, or the matching tab on the main settings page.

Once you are within the settings for the “User Roles Manager”, scroll to the section “Elements”, which will provide you with a listing of all elements that are registered with Visual Composer. Simply mark the elements the respective user role is allowed to use, save the settings, and provided the logged in user has the required user role, all assigned elements should be available when editing a page or post.

## Plugin Settings

### General Settings

The general settings section will give you access to plugin options that have a global impact on the plugin itself and the features it provides. Every setting option includes a short description as to what the option does.

As with every plugin or theme ... the more features one enables, the higher the resulting impact on WordPress performance (memory consumption, load times, server load, etc.) can be. While the impact of a single plugin is usually easy to handle for WordPress and the server, the cumulative (compounded) impact of all active plugins and the theme is what really impacts overall performance, so only enable the plugins you actually need for your websites features and to keep performance at its best.

**Naturally, this plugin is specifically designed to only load the resources it actually needs on any given page (if any).**

### Format Settings

The format settings section specifically deals with settings that are relevant to the table editor itself, allowing you to define some default settings that will be used for every table you create.

Some of the default settings you define here can be changed for individual table cells when using the table editor (such as alignments, format locale and currency symbol), while other settings have a global impact and can not be adjusted for individual cells.

By default, the plugin will attempt to detect the WordPress language setting (locale) you are using, and will set the table editor locale to the same language, provided that language is one of the included editor locale this plugin currently provides for.

If your specific WordPress language is currently not supported as locale for the table editor, please select one of the available languages/locales that best fit the specific number formatting that is used in your particular country.

## Translation Settings

When using some of the included advanced table scripts (“TableSaw”, “FooTable”, “DataTable”), the plugin will use some text strings on the frontend, based on the scope of your table (sorting, search, pagination, etc.).

The language settings section allows you to change those text strings to better fit the actual language of your site. All text strings entered here serve as default strings for all tables, but you can change the text strings for each table individually when adding the table to your site in Visual Composer (via the element settings panel), as well as when using the general shortcode generator.

If a specific input for a text string shows a help icon next to it, clicking on that icon will provide some more information about that particular text string, as the string is using at least one placeholder that will be replaced with actual values (based on the actual table used) once that table is rendered on the frontend.

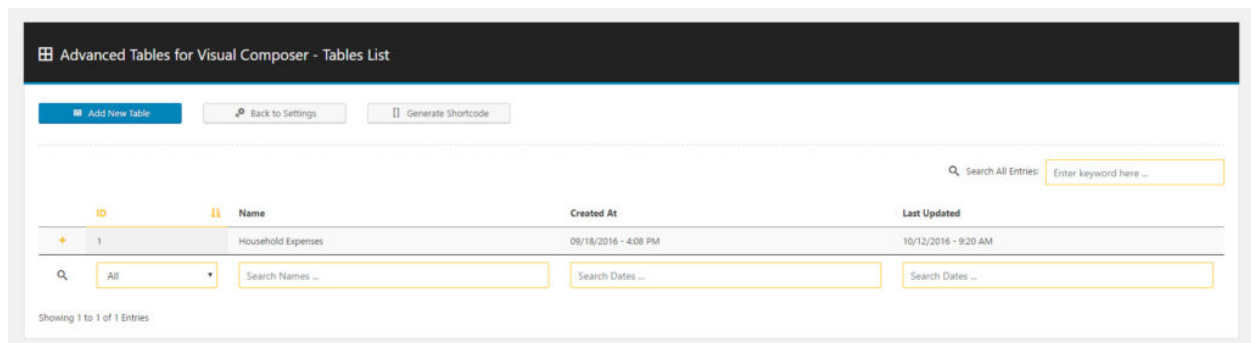
## Table Listing

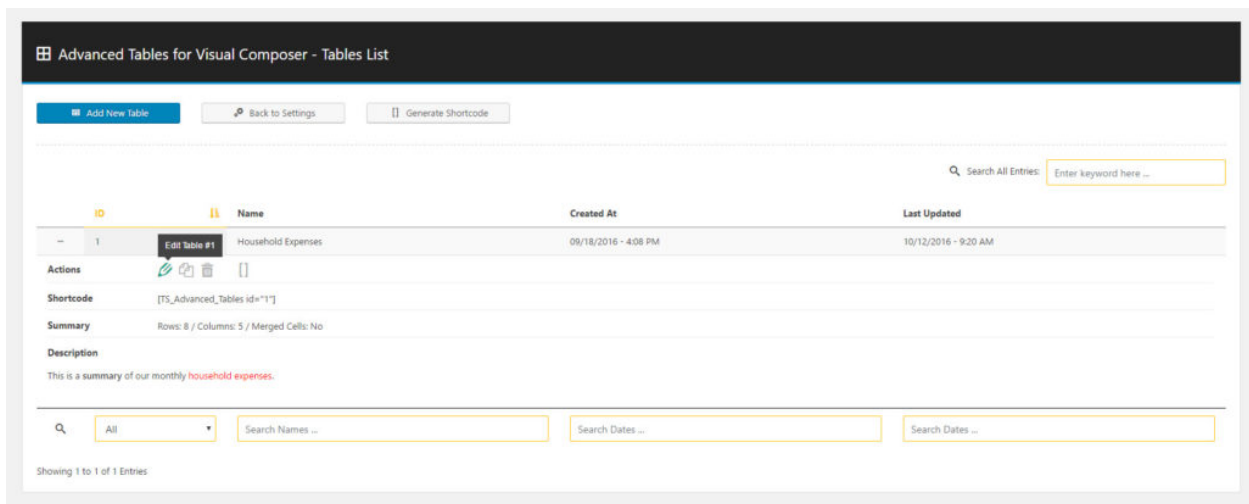
### Existing Tables

The “Tables List” page will provide you with a listing of all created tables, allowing you to edit those tables again, or to delete or clone them. The page will also provide you with a shortcode generator that can be used to generate a full shortcode for a specific table, which can be useful if you want to use a table within a post type which Visual Composer has not been activated for.

When opening the “Tables List” page, all existing tables will be displayed in a table listing, which by default is sorted in ascending order, using the table ID as sort criteria.

If you prefer a different default sort order or sort criteria, you can change those settings in the general settings section within the plugins settings page. Of course, you can always change sort order and sort criteria manually, using the provided controls in the header of the tables list.





## Table Actions

The “Tables List” will initially show the table ID and name, as well as the date the table was first created, and the date it was last updated (modified).

Clicking on the provided “plus” icon (or anywhere within the table row) will open or hide additional table information, which will also include “action” links for that table, to either edit the table, clone it, or to delete the table altogether. The “shortcode” icon within the action links will open a shortcode generator popup for the table, allowing you to generate a (full) custom shortcode.

## Table Editor

### Editor Basics

Whether you create a brand new table, or edit/modify an existing one, the table editor will be the same.

The table editor is basically a collection of a variety of 3rd party scripts, adjusted to work in tandem in order to create the editor itself. Due to the large number of files required for the editor to work, it is not feasible to provide the editor within an element in Visual Composer itself, as the risk of conflicts between the editor files and any other files loaded when editing a page or post with Visual Composer is simply too big.

Furthermore, most pages on your website will most likely not actually include a table, but if the editor would be incorporated within an element in Visual Composer, you would be loading the required editor files every time you edit a page or post, even if you don’t actually plan to embed a table on that particular page or post.

Aside from the aforementioned potential conflicts, that would negatively impact the performance of Visual Composer and WordPress alike, which is the main reason, why the table editor has been placed into a dedicated section within the plugin itself.

The editor itself is made up of two distinct sections:

- General table information, such as table name, ID, date created and an optional information summary. The table ID and date created will be assigned automatically to the table the moment you create it.

- The actual editor, including the raw data text area, editor toolbar and the spreadsheet mimicking table structure. This section also includes a toggle button, which will give you access to some additional options, allowing you to enable/disable support for some advanced editor features, some of them still to be considered in “BETA” stage.
- The editor also provides a “distraction-free” (fullscreen) mode, which can be toggled via the corresponding icon in the editor toolbar. If supported by your browser, the fullscreen mode can also be exited using the “ESC” key on your keyboard. When in fullscreen mode, the editor will be limited to the (optional) search bar, the raw data field, editor toolbar and the table itself. All other features present on the editor page will be hidden.

## Table Spreadsheet

To make it as easy as possible to provide content for a table, the table editor will mimic a layout you might be familiar with from your desktop spreadsheet software, such as Microsoft Excel or Apple Numbers.

**But while the look might be the same, the table editor is neither able, nor designed to provide you with the same scope of features as a normal spreadsheet software. The main purpose of this editor is to simply enable you to quickly provide content to a table cell, using a familiar layout.**

By default, the editor will generate the spreadsheet with 10 rows and 5 columns, but you can always add more rows and/or columns to the table, simply by either using the provided controls in the toolbar, or via the context menu, which can be accessed with a right mouse click on any given cell.

You do NOT need to delete any unused columns or rows, as the plugin will only render those rows and columns on the frontend, that actually have content (empty rows or columns between other rows or columns with content will of course be shown on the frontend as well).

In order to provide content to a specific empty cell, you simply select that cell within the spreadsheet, and start typing. If you want to edit the existing content of a cell, you simply make a double click on that cell.

---

## Editor Toolbar

In order to provide easy access to some basic styling and control options for a given cell, the editor is providing you with a corresponding toolbar above the table spreadsheet.



The scope of provided toolbar options partially depends upon the content type of the cell, as well as enabled editor features in general.

Each toolbar option will provide you with a corresponding tooltip, indicating the associated action for that option, although the utilized icons should hopefully explain the underlying use already. Some toolbar options will open a submenu or color picker, while most will trigger the associated action via the main icon directly.

## Content Types

The table editor allows for 6 distinct content types (or cell formats). The content type for a cell can be changed by using the # icon in the editor toolbar, after a cell has been selected.

## **Text**

The “Text” content type is the most generic cell format and also the default format for all cells. It basically allows you to use any text, including basic HTML code, as content for a cell.

Naturally, cells within a table are not meant to hold complex HTML structures, as that would interfere with table layout and styling, table responsiveness, as well as any advanced table features (such as search, sorting, pagination).

## **Number**

The “Number” content type is meant for any numeric content that is not representing a currency or percentage (as those have their own content types). The number of decimal points for numeric values is defined globally in the plugin settings, and will be the same for all values using the “Number” content type.

You are, however, able to change the locale for the number format, which will affect which characters are used to indicate the decimals and thousand separators, using the listing of available locales that can be accessed using the “Format Locale” option in the editor toolbar.

## **Currency**

The “Currency” content type is meant for any numeric content that is representing a currency and requires the display of a currency symbol. The number of decimal points for currency values is defined globally in the plugin settings, and will be the same for all values using the “Currency” content type. The global plugin settings also determine whether the percentage symbol is shown before or after the value (with or without spacing).

You are, however, able to change the locale for the currency format, which will affect which characters are used to indicate the decimals and thousand separators, using the listing of available locales that can be accessed using the “Format Locale” option in the editor toolbar. You are also able to change the currency symbol itself for each cell individually.

## **Percent**

The “Percent” content type is meant for any numeric content that is representing a percentage and requires the display of a percentage symbol. The number of decimal points for percentage values is defined globally in the plugin settings, and will be the same for all values using the “Percent” content type. The global plugin settings also determine whether the percentage symbol is shown before or after the value (with or without spacing).

You are, however, able to change the locale for the percentage format, which will affect which character is used to indicate the decimals, using the listing of available locales that can be accessed using the “Format Locale” option in the editor toolbar.

## **Date**

The “Date” content type is meant for any content that is representing an actual date. The format in which the date is displayed is defined globally in the plugin settings, and will be the same for all values using the “Date” content type.

All cells using the “Date” content type will provide a date picker in the cell itself, accessible by a small triangle icon in the upper right corner of the cell.

## Time

The “Time” content type is meant for any content that is representing an actual time. The format in which the time is displayed is defined globally in the plugin settings, and will be the same for all values using the “Time” content type.

## Editor Performance

In order to preserve the original cell content, the table is storing format information (alignments, formatting, number of decimals, locale, currency symbol, etc.) as meta data for each table cell individually. This is important in order to ensure that the unformatted content is still available when rendering the table on the frontend itself, but particularly for any sorting routines applied to the columns.

This means that when the table editor is rendering in order to edit an existing table, the plugin will have to process the individually stored metadata for each cell and apply the information to each cell. This process is done “client side”, meaning it is done on your local computer, using your browser.

Therefore, the speed at which the editor can process your table for the editor is directly dependent upon your computer’s CPU and memory power, as well as the utilized browser. Naturally, the larger the table (number of columns and/or rows) and the more meta data is attached to the individual cells, the longer the process can potentially take.

If your browser seems to “freeze” during the table processing (because it is a very large table with a lot of meta data), let the browser continue “working its way through” and don’t refresh the editor page. The processing will eventually be completed so that the browser will become accessible again.

In general, the editor is designed to handle a moderate amount of data at a reasonable speed. And while there is no built-in limit in the amount of possible rows or columns, the overall amount of cell raw and meta data, in conjunction with your computers processing power, will in effect determine how far you can “push” the editor, before performance becomes intolerable.

## BETA Features

As stated before, the editor is basically just a collection of several otherwise independent scripts, made to work together in order to create the editor. Those scripts are obtained from 3rd party sources, as developing a custom solution for all of them would not be possible without investing a significant amount of time and resources, resulting in a much higher item price for the plugin.

The table editor allows you to enable some advanced editor functions that are still to be considered in “BETA” mode. That means, it is currently not possible to ensure and guaranty full functionality of those features and full support will hopefully be provided with a future update, as soon as the underlying scripts have been updated by their corresponding developers.

**As such, please use all editor features marked as “BETA” with caution and only if absolutely necessary.**

---

## Provide Row + Column Drag Support (BETA)



This BETA feature basically allows you to move columns or rows into different positions within the table, moving the contained data with it. This routine is usually working without problems, provided the table does not use any merged cells (which is also a BETA feature).

## Provide Cell Merge Support (BETA)

Merging cells obviously changes the underlying HTML structure of a table, resulting in a different number of columns in different rows. That will immediately affect how a table reacts when it is displayed on smaller screens (“responsiveness”), as the only way to make tables responsive is to either make column widths smaller (up to the point that the content in those columns can’t be read anymore), or to hide/collapse/shift columns around.

The latter is the recommended approach to make tables responsive, but it does not work with merged cells, as the merged area can include columns that are meant to be hidden/collapsed/shifted around, but also include columns that are still meant to be visible.

Furthermore, advanced table features such as column sorting or pagination simply does not work with merged cells, as tables are sorted by a specific column and a merged area can cover multiple columns, therefore breaking the sorting routine.

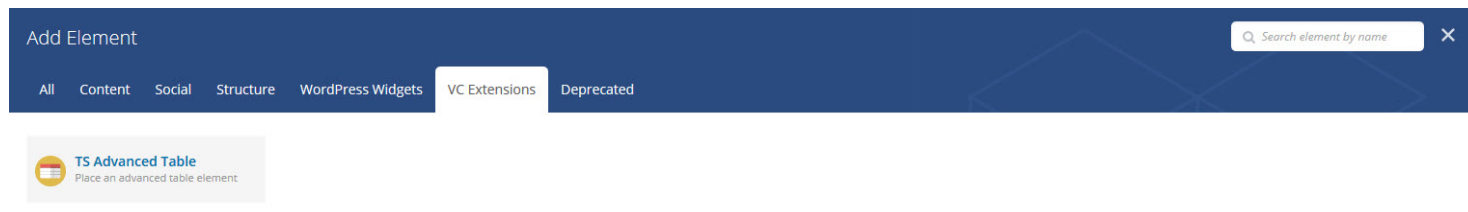
**As such, this feature isn’t as much in real “BETA” mode (as merging cells does work correctly), as it is to be considered a non advisable usage if one wants to make the table responsive and/or use advanced table features.**

## Table Embedding

### Visual Composer Element

Since the name of this plugin is “Advanced Tables for Visual Composer”, the plugin will of course provide you with a dedicated element to actually embed any of the created tables into your page or post.

The new element is named “**TS Advanced Table**” and will be placed under “**VC Extensions**” within the Visual Composer element selection panel.



The settings panel for the element is extensive, allowing you to finetune almost every aspect and feature of the table. In order to keep options manageable, the element makes extensive usage of so-called “conditional” options, where some setting options are show or hidden based on the value of another setting option.

### Shortcode Generator

There are two access points for the shortcode generator, when viewing the “Tables List” page:

- The dedicated “Generate Shortcode” button will open the shortcode generator, requiring you to select a specific table first, before the actual table options will be made available.
- The shortcode action icon that is shown for each created table will open the same shortcode generator, but will preselect the specific table the action icon was associated with.

Just like the settings panel for the Visual Composer element, the shortcode generator is extensive as well, as it includes the same setting options, allowing you to fine tune almost every aspect and feature of the table.

In order to keep options manageable, the generator also makes extensive usage of so-called “conditional” options, where some setting options are show or hidden based on the value of another setting option. The screenshot to the left shows the shortcode generator with ALL available options.

Once you made your selections in the shortcode generator, you simply need to click on the “Generate Shortcode” button at the bottom of the generator popup, and the table shortcode will be generated based on your selections.

Once you do, the generator popup will close and the generated shortcode will be placed into a dedicated textarea at the top of the screen, allowing you to copy the shortcode, so you can embed it elsewhere. For ease of use, a “Copy to Clipboard” button will also be provided, to automatically place the shortcode into the browser clipboard (HTML5 browsers only).

The shortcode for the table has been copied to your clipboard!

```
[TS_Advanced_Tables id="1" table_scope="none" show_name="true" show_info="true" show_comments="true" preloader_use="false" header_use="true" header_rows="1" footer_use="false" responsive_type="none" margin_top="0" margin_bottom="0"]
```

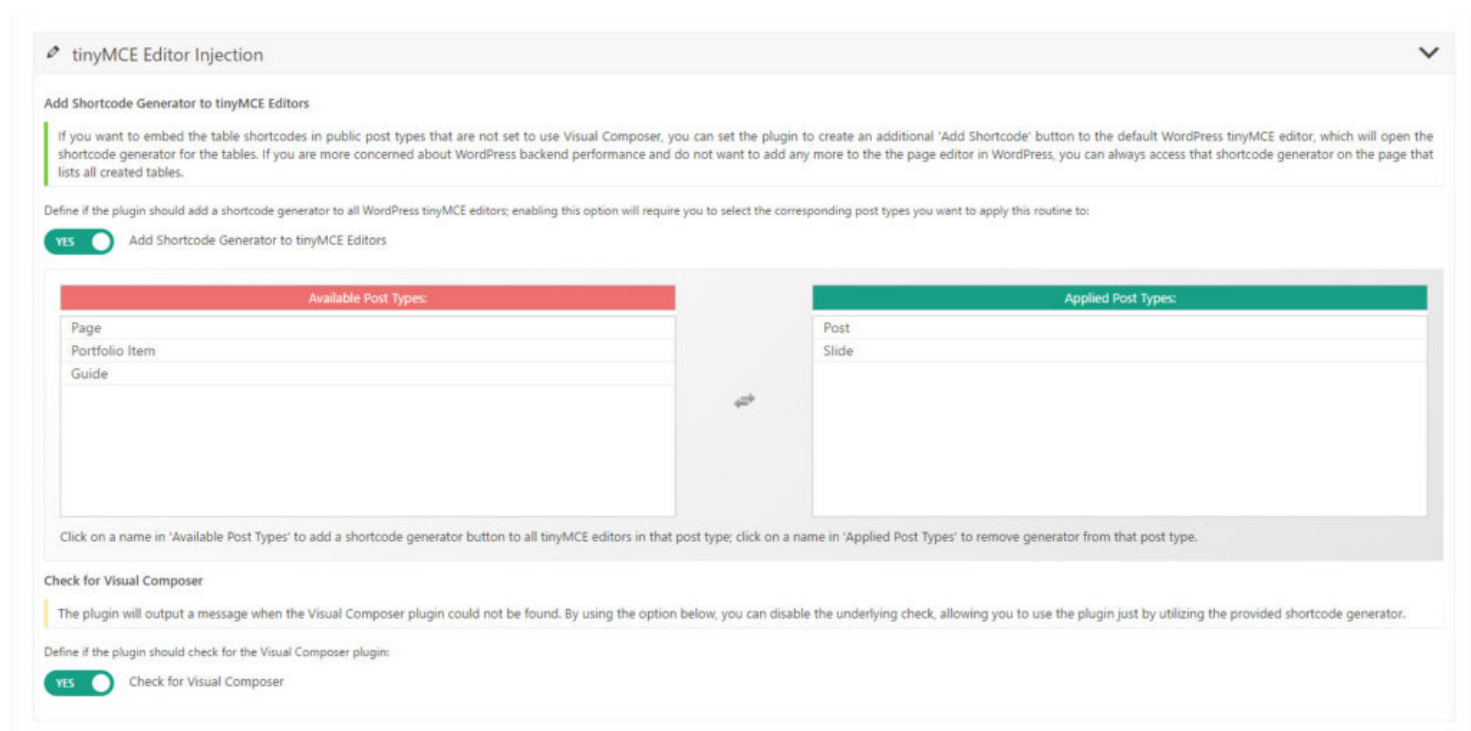
Copy to Clipboard

## tinyMCE Editor Button

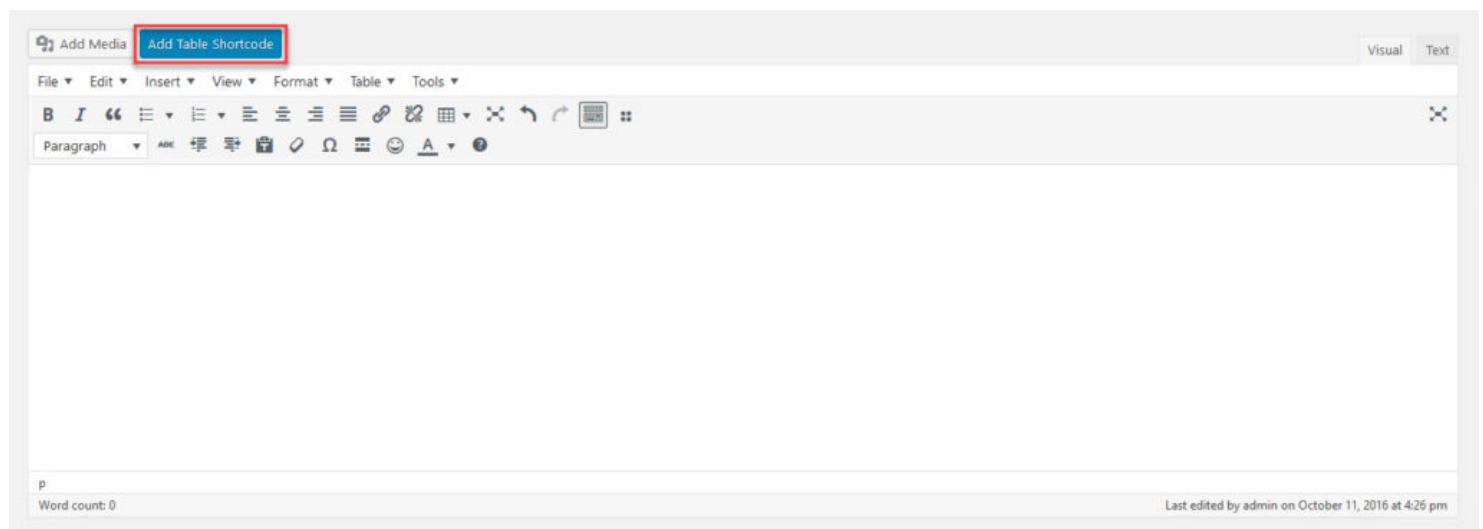
By default, the plugin will only provide its shortcode generator on the “Tables List” page, as the intended way to embed a table into a page or post is to use the provided element in Visual Composer. And for all other occasions/needs, the built-in shortcode generator on the “Tables List” page is usually sufficient.

However, if you really want to, you can add a shortcode generator button to every standard tinyMCE editor that is generated by WordPress for posts or pages. In order to do so, you simply need to enable that feature in the plugin’s settings page (“General Settings” -> “tinyMCE Editor Injection”).

After enabling the feature, the next step is to select the post types you want to apply the feature to, using the provided input.



After enabling that feature and saving the settings, you should find a new shortcode generator button for each standard tinyMCE editor that is generated by WordPress for each of the assigned post types.



## Table Features

### Features Summary

Aside from some basic table stylings, this plugin allows you to apply an advanced table script to your table, which will add some additional features to your table, such as column sorting, table search, export options, and others (based on selected script).

In order for most advanced table routines to work, your table should be designed without using any merged cells, as

merged cells can stretch over multiple columns and/or rows, which in turn prevents routines such as column sorting or responsive behavior, where columns are collapsed, from working correctly.

If you are using number formatting (including currencies and percentages) within your table, it might be required to assign columns to specific content formats in order for column sorting to work correctly. You will find corresponding input for that in the element settings in Visual Composer, as well as the shortcode generator.

The utilized advanced table scripts are 3rd party solution and this plugin has no control over any errors or issues those scripts might still have. If you notice any errors or issues with any of those scripts, please contact me, so I can inform the respective author about it, hoping s/he will eventually release an update that fixes the error or issue. As such, no time frame can be given for fixes to any such errors or issues, as the availability of such a fix will depend upon the respective author.

## **TableSaw Features**

The “TableSaw” script is a lightweight script to quickly and easily add a sorting routine and responsive layout to your table, without sacrificing speed and performance. It will only add a minimum styling to the table and is most useful for small to medium sized tables, as it does NOT provide a row pagination feature.

It’s purpose is to assign a responsive behavior to the table, as well as a basic optional sorting feature for the table columns.

The TableSaw script requires a table header in order to work correctly. Even if you set the table to be displayed without header but using the TableSaw script, the plugin will always use the first row as header in order to prevent the TableSaw script from failing!

You will find matching settings for the TableSaw script in the built-in shortcode generator as well, as both the Visual Composer element and the standalone shortcode generator give you access to the exact same setting option for your table.

All setting options include a respective description right within the element settings in Visual Composer or the Shortcode Generator. Some more specific options are explained below:

### **Sort: Main Control**

When adding a sorting feature to the table via the TableSaw script, you also have the option to add a global sort control above the table itself (one can still sort the columns simply by clicking on a respective column header directly).

### **Responsiveness: Mode Switch**

While you must assign a specific responsive mode to the table (“Swipe” or “Column Toggle”), you can also provide your viewers with a global control option to switch between the available modes (“Swipe”, “Column Toggle” or “Stack”). The “Stack” mode is NOT available as default responsive mode due to overly large layout it generates, causing excessive scrolling, therefore leaving that option to the end viewer.

### **Responsiveness: Mini Map**

The responsive modes “Swipe” and “Column Toggle” have the option to show a so-called mini map to the end user, indicating which columns of the table are currently visible to the viewer, and which ones are currently hidden. The mini map is just a line of individual dots, each dot representing one column within the table. Dots with a black

background indicate a visible column, while dots with a grey background indicate hidden columns.

## **FooTable Features**

The “FooTable” script is a medium heavy script to quickly and easily add a sorting routine, row pagination, search option and responsive layout to your table. Due to its row pagination option, the script is perfectly suited for large tables.

The “FooTable” script does NOT support footer rows, as the table footer is a reserved space for the row pagination controls. Even with row pagination disabled, the script will still not support custom footer rows.

When using the optional sort feature with the “FooTable” script, it is highly recommended to manually assign a respective column format, using the provided inputs in the Visual Composer element or Shortcode Generator. Particularly columns with “date” content (in some formats) will not sort correctly, if you haven’t assigned that column to be a “date” column, due to some inconsistencies with the “FooTable” content parser.

## **DataTable Features**

The “DataTable” script is the heaviest advanced table script, but also the most complex ones in terms of features. It allows you to quickly and easily add a sorting routine, row pagination, search option, export buttons and responsive layout to your table

## **Tips & Tricks**

### **Wrong Sort Order**

By default, the underlying advanced table scripts will attempt to determine the column content format automatically, which is important for any utilized sorting routine.

If the column sorting does not provide correct results on your end, your first step should be to check that you didn’t mix content formats within the same column. For example, any sorting routine can only sort/process content in the same format within a column, so mixing “text/html” content with numeric content will prevent any sorting to work correctly.

As a second step, try to manually assign the column in question to a specific format (in case the auto-detection is failing for some reason), which you can do using the provided dedicated inputs for that in the Visual Composer element settings panel, or the shortcode generator.

## COLUMNS FORMAT SETTINGS

Advanced table scripts will attempt to parse the column content to determine the format of the content, which is relevant for any sorting feature to work correctly. Using the inputs below, you can manually assign columns to specific formats.

### Columns: HTML/Text Content

Define the columns that contain HTML and/or text content only; separate columns with comma and use only numeric values.

### Columns: Numeric Content

Define the columns that contain standard numbers only; separate columns with comma and use only numeric values.

### Columns: Currency Content

Define the columns that contain currency numbers only; separate columns with comma and use only numeric values.

### Columns: Percent Content

Close

Save changes

**Advanced Tables for Visual Composer** ✕

**Columns Format Settings**

Advanced table scripts will attempt to parse the column content to determine the format of the content, which is relevant for any sorting feature to work correctly. Using the inputs below, you can manually assign columns to specific formats.

Define the columns that contain HTML and/or text content only; separate columns with comma and use only numeric values.

Columns: HTML/Text Content	<input type="text"/>	?
Columns: Numeric Content	<input type="text"/>	?
Columns: Currency Content	<input type="text"/>	?
Columns: Percent Content	<input type="text"/>	?
Columns: Date Content	<input type="text"/>	?
Columns: Time Content	<input type="text"/>	?

**Table Features**

When using advanced table scripts, please ensure that you do not use merged cells within the last header row, the table body and the table footer, as advanced features such as sorting of columns do not work with merged cells.

**Table Column Settings**

Generate Shortcode

## Responsive Behavior

Based on your selected table scope/theme and the assigned responsive behavior, the table will auto-adjust its layout once certain screen sizes and/or table widths have been detected.

While some responsive layouts don't require any additional settings, beyond a width setting at which the layout should be triggered, the "FooTable" and "DataTable" scripts in particular allow you (or even require you) to provide some more information about which columns should be targeted at certain breakpoints.

In order to do so, you will find matching inputs in the Visual Composer element or shortcode generator, assigning columns to be collapsed (hidden) whenever a certain table width is reached. This will allow you to fine tune which columns will remain visible to your viewers (the important columns, so to say), while the less important columns will be collapsed (hidden).

The user can view collapsed (hidden) columns at any time by using the provided toggle button for each row, which will show all collapsed (hidden) columns in a subtable below the row.

When using the "FooTable" or "DataTable" script, and if your table is collapsing columns in an order you do not want it to be, simply assign columns to be collapsed (hidden) at specific breakpoints only.