# Visual Composer: A Guide To Drag & Drop Page Building

Prestashop is great, and being able to customize your Prestashop run website is just the icing on the cake. However not all of us are coding gurus. Heck, some Prestashop lovers don’t know anything about HTML or CSS. But this shouldn’t stop you from being able to customize your Prestashop theme. This is where page builders come in.

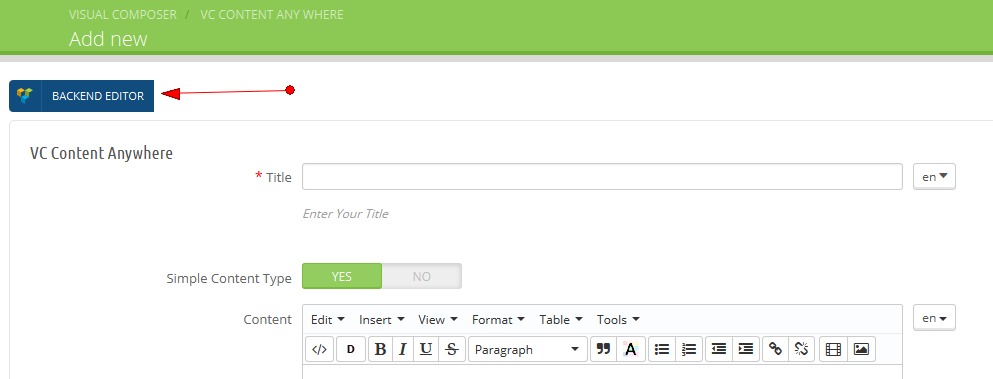
Drag & drop page builders are widely available as modules and are built into many popular themes. They make it possible for users of all levels to make changes to the general layout of a theme, and it can make adding extra flourishes (icons, skill-bars, pricing tables, etc.) super easy. Specifically we want to introduce you to our new favorite page building Prestashop module – the [Visual Composer](http://www.google.com)

**What Is Visual Composer**

Visual Composer is a fantastic module developed by Smartdatasoft that adds a drag & drop page builder to your Prestashop theme. We popped it into our [Total Multipurpose Prestashop Theme](http://www.google.com), but you can also see it in almost every new top seller on Themeforest. It gives you the tools you need to be able to create complex page layouts without needing to touch a single line of code.

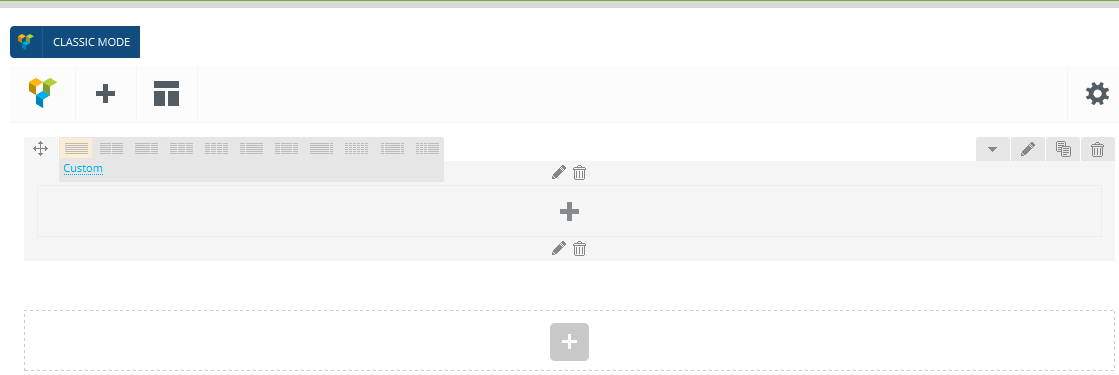
**How Visual Composer Works,And What It Can Do**

It’s a Module, so just install and activate to take advantage of all it’s wonderful features. Once activated, Visual Composer gives you control over the layout of your pages. Just add a new page, and click on the blue **Backend Editor** button to access the builder.



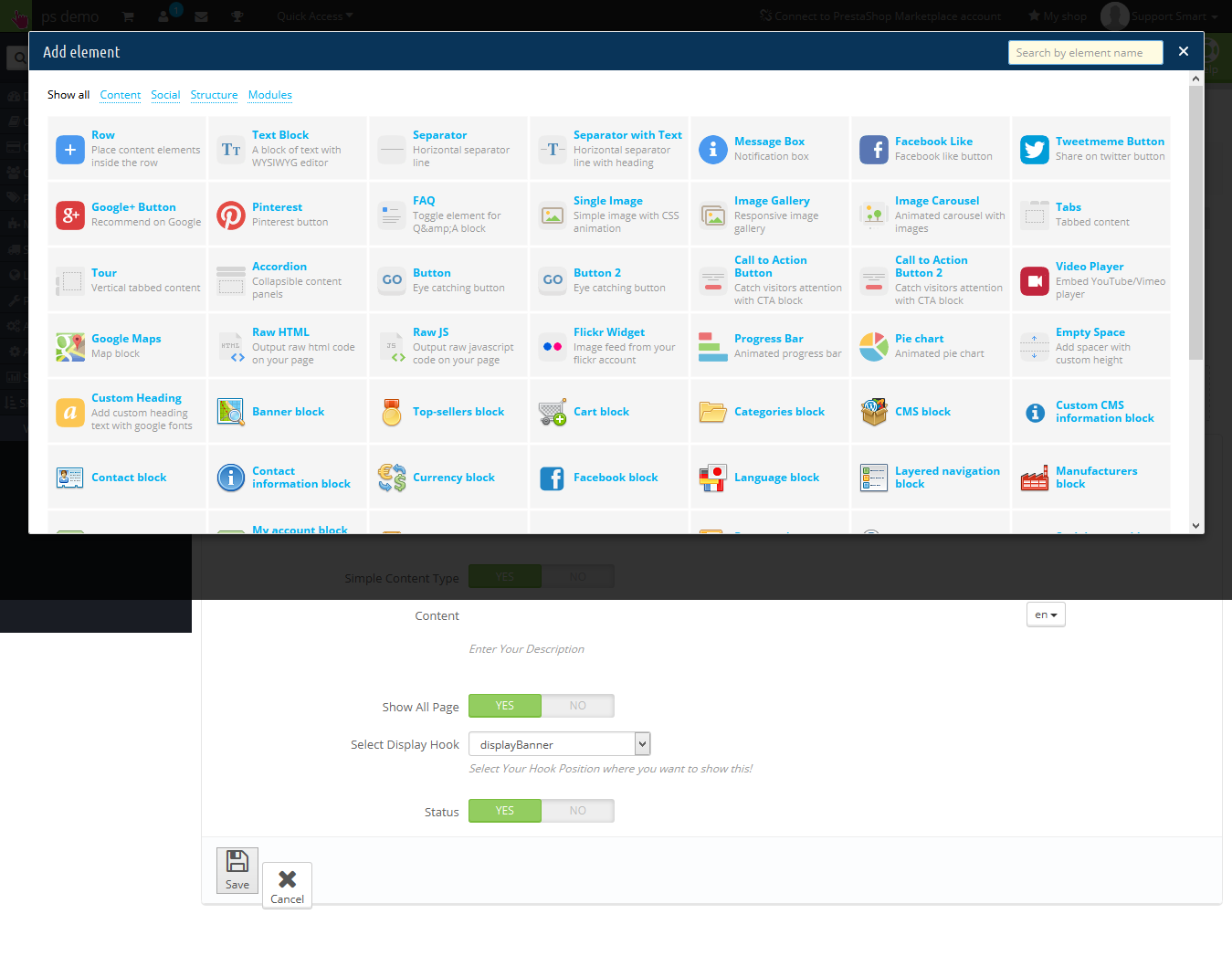
From here the possibilities are only limited by your (or your clients’) imagination. You can add rows, columns, buttons, text, images and so much more using all of the Visual Composer’s options. Plus there’s an option to save your layouts as a templates to be reused over & over again.

### **Rows**



Included in the builder are page building elements. You can use these elements as literal building blocks for your page. First **Add A Row**. Rows make it possible to add columns in one click. Just hover on the yellow paragraph tab at the top left corner of your row and select a layout. The row automatically breaks into columns according to the layout you choose.

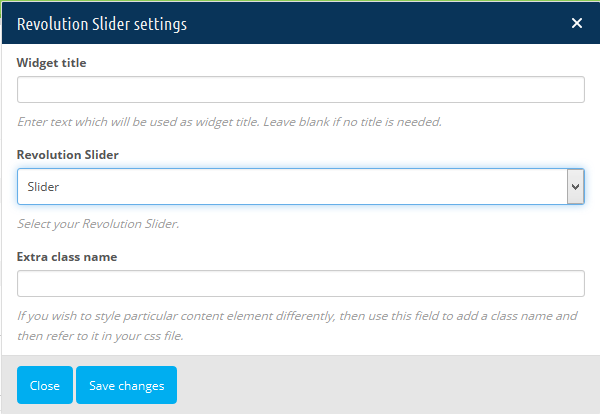
### **Page Elements**



The next step is to begin adding your page features. Just click the **Add Element** button to open a pop-up menu. There are lots of options for adding text boxes, headings, social sharing, images and galleries, toggles or tabs, accordions, recent CMS (which also works with the custom cms types of your theme in addition to the blog CMS), buttons and even sidebar widgets.

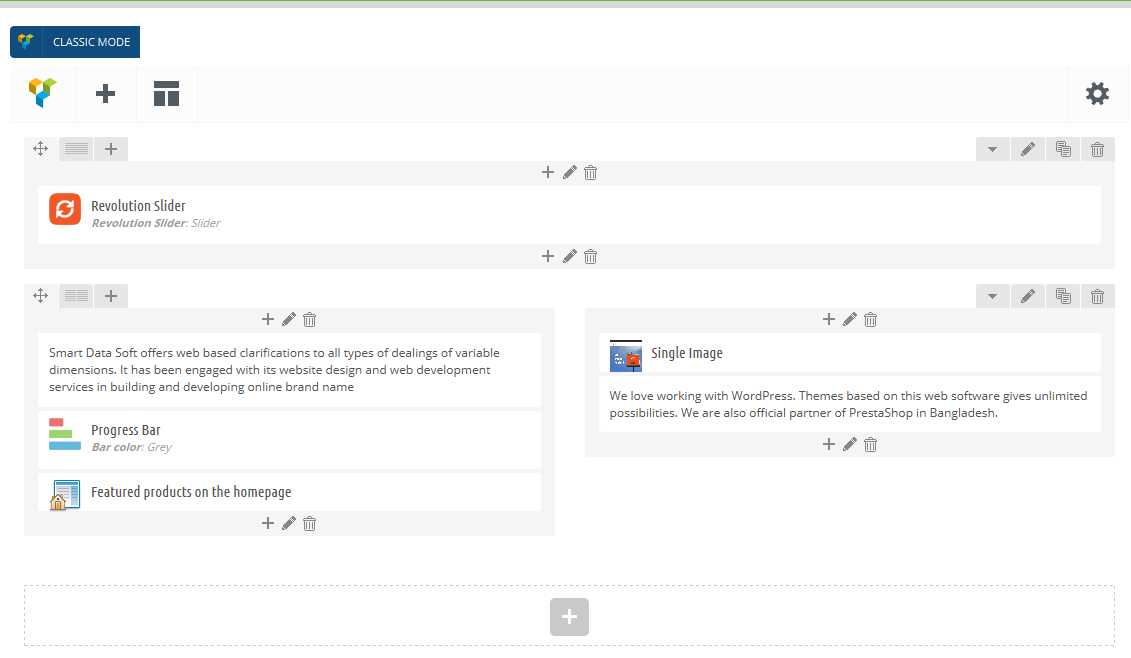
Also, depending on the Prestashop theme and compatible modules you are using with the Visual Composer your will see more or less page elements. The above example is from our [Total Drag & Drop Prestashop theme](http://www.google.com), so there are an additional 30 page builder elements we created and styled exclusively for Total. This includes custom dividers, icon boxes, milestones, carousels and more.

Click on any element to open the options for that element (which ranges from choosing colors and alignments, to make decisions about slider transitions).



The builder also supports four awesome third party Modules: Revolution Slider, Smartshortcode. If you have any of these module installed, a page element will automatically appear for theme. When you choose to insert a page element for one of these modules, Visual Composer displays a drop-down of the respective sliders/forms you’ve created, or displays an entry box for you to add your slider ID. No more copying and pasting codes!

### **Drag & Drop**

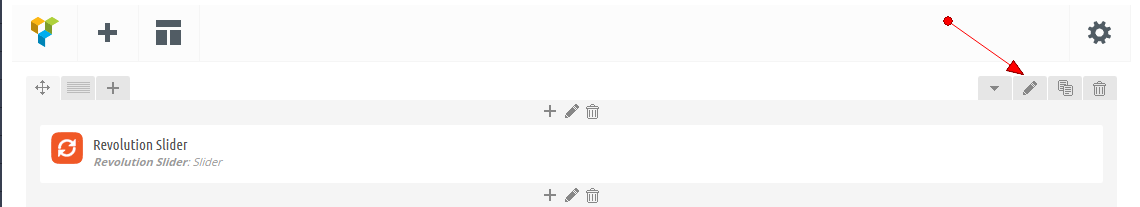


At any point in the design process you can drag & drop any row or page element to rearrange them. You can also duplicate any row or element, which makes adding repetitive content quick and easy (great for similar columns, FAQs, toggles, etc.).

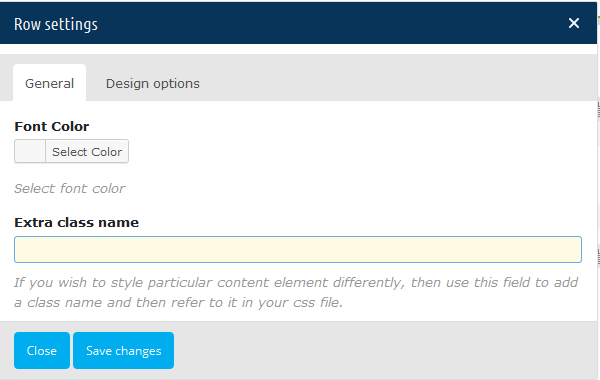
Once you’ve created a design you love you can save it as a template. This comes in handy when creating a custom page layout or sidebar the you’d like to reuse. Just click on the **Templates** button, give your layout a name, and hit save. When you use your template it will automatically populate your page with the layout and elements you saved.

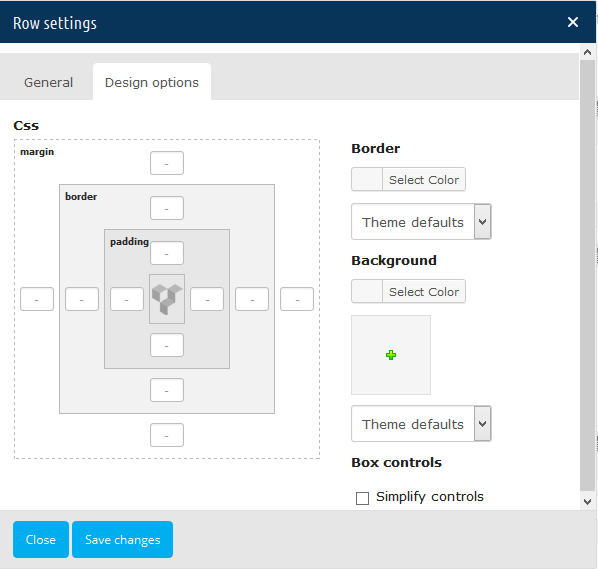
### Row Options: Format, Clone & Delete Rows

There are three added features added to the far right corner of every row. The **X** can be used to delete an entire row and its contents, the **Papers** can be used to clone or duplicate an entire row and all of its contents, and the **Pencil** is used to access the formatting options



Again, depending on the theme you are using with the Visual Composer the formatting options available to you may be different. While some theme developers choose to leave the Visual Composer as-is, many customize the add-on modules to fit the needs or style of their theme.





The screenshot above is from our Total Drag & Drop Prestashop theme and includes all the options we’ve custom added. We chose to separate the row styling options into tabs to make it easier for users to change the backgrounds, borders, padding and other elements of each individual row.

### **Frontend Editor**

The powerful Visual Composer now includes a frontend editor, which makes it easier than ever to get started building your own pages and layouts. The only caveat is that not all themes support the front-end editor just yet. Make sure to double check a theme’s feature list, or to contact the theme developer directly, to see if the Visual Composer frontend editor is compatible with your Prestashop theme.

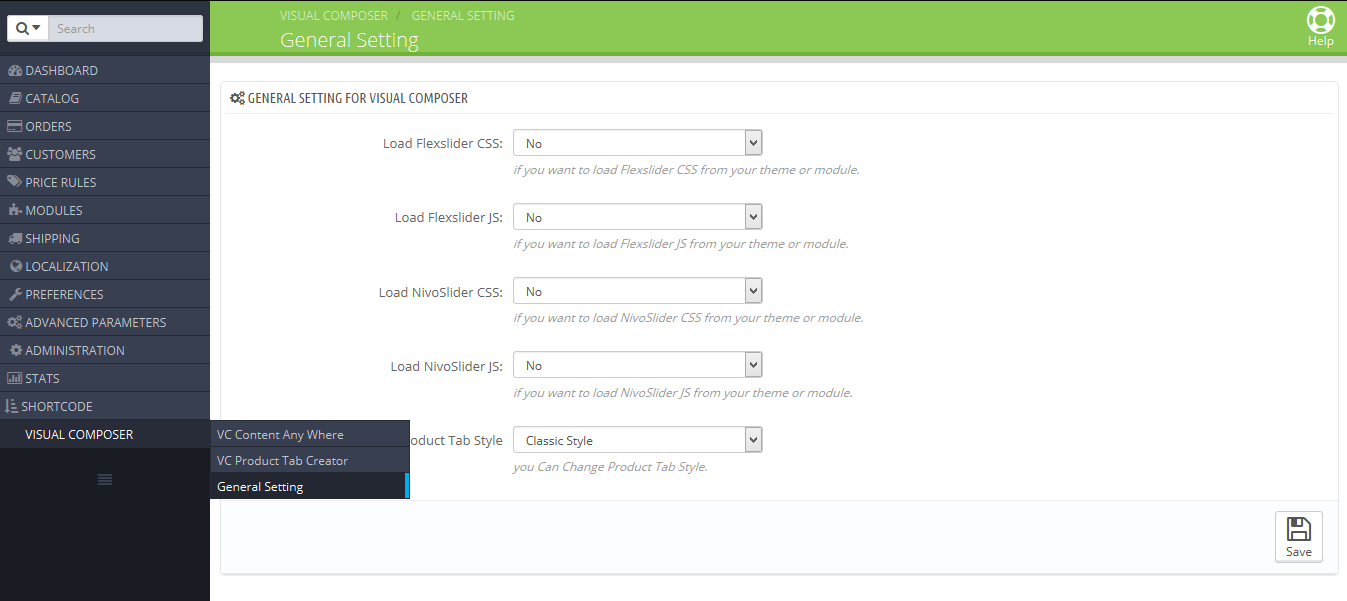
### Extensions

The last feature I’d like to touch on is Visual Composer’s extendibility. If you happen to be an advanced coder or theme developer you can integrate your own short codes into the module. In fact, we added in our own Simple Short codes Premium to give Total users an additional page building elements. This gives users even more options when building pages!

If you’re not a coder, checkout [CodeCanyon](http://www.wpexplorer.com/out/codecanyon) and do a quick search for Visual Composer Add-on. There are lots of great premium Modules that add extra features to the Visual Composer

### Visual Composer Settings Options

I keep referring to the Visual Composer as a page builder, but it works great with CMS and custom CMS types as well. All you have to do is enable the option. Navigate to **Menu > Visual Composer**. You should see a screen similar to this:



The Module will automatically display the various content types included in your Prestashop theme. Just select which you’d like to use with the Visual Composer page builder. The example above is from our Total theme, which is why there are added options for portfolio, staff and testimonials as well as forums, topics, reply and products supported by Total).

*-Content From:*[*http://www.wpexplorer.com/visual-composer-guide*](http://www.wpexplorer.com/visual-composer-guide)

# Visual Composer Pagebuilder for Prestashop

# [Introduction](https://wpbakery.atlassian.net/wiki/display/VC/Introduction)

Thank you for purchasing Visual Composer Prestashop Module.

Visual Composer is very unique Module, it will help you managing your content at the Prestashop sites and create stunning layouts in few minutes without coding. Now a days many websites have complex grid layouts with columns, tabs, sliders and etc. In the past, to create that type of layouts you should be HTML guru or Shortcodes operator ninja.

But not anymore! We have prepared a short video for you to get you up and running in no time. See yourself how easy it is to create and manage complex layouts combined with professionally designed Modules made easy!

In this section you will find list of useful resources related to Visual Composer Prestashop Module, including link to official support, CodeCanyon sales page and official website.

* [Resources](https://wpbakery.atlassian.net/wiki/display/VC/Support+and+Resources#SupportandResources-Resources)
* [Official Support](https://wpbakery.atlassian.net/wiki/display/VC/Support+and+Resources#SupportandResources-OfficialSupport)

## Resources

* [Official website](http://vc.wpbakery.com/)
* [Official sales point on CodeCanyon](http://codecanyon.net/item/visual-composer-page-builder-for-wordpress/242431?ref=wpbakery)
* [Official demo](http://go.wpbakery.com/vcdemo)
* [Video Academy](http://vc.wpbakery.com/video-academy/)
* [Facebook page](https://www.facebook.com/wpbakery)
* [Twitter account](https://twitter.com/wpbakery)

## Official Support

To get your support related question answered in the fastest timing, please head over to our [support page](http://support.wpbakery.com/) and open Support ticket. To open support ticket you should have account in the ticket system. Registration should take you couple of minutes to complete and is completely automatic process. Prepare your [Purchase code](https://wpbakery.atlassian.net/wiki/pages/viewpage.action?pageId=524307) and fill in**"Create account in Ticket system"** form located at [support page](http://support.wpbakery.com/), then follow on screen instructions.

Before applying for support please make sure you understand the rules of support and go through all steps described and listed in [Support Policy](https://wpbakery.atlassian.net/wiki/display/VC/Support+Policy) in order to get your issues solved as soon as possible

# [Support Policy](https://wpbakery.atlassian.net/wiki/display/VC/Support+Policy)

## **Opening Support Ticket**

In order to get response in reasonable time we are kindly asking you to provide following information while opening support ticket:

* Your website URL
* Login credentials
* Version of Visual Composer used
* Detailed description of actions taken before issue
* Screenshot of the problem
* Any other information that you find useful

All information provided is confidential and will not be provided to any third party.

## **What You Can Expect From Our Support?**

There are certain things that you can expect from our Help Center:

* We always have bug-related and technical support in priority
* We are happy to receive proposals and ideas
* Tiny custom requests can be served (low priority, depending on workload and good will)

## **What You Can Not Expect From Our Support?**

There are certain things that are out of scope and will not be handled by our team:

* Generic Prestashop questions
* Issues related to 3rd party add-on Modules
* Customization requests
* Any tickets that contain abuse or violence
* Requests from indirect purchases (Visual Composer came as part of some other product)

## **E-mail Policy**

With leaving your e-mail you agree to receive promotional e-mails and newsletters from Smartdatasoft.

Thank you for finding time to read this policy and remember that we are here to help you out.

Best wishes, Smartdatasoft

# [Release Notes](https://wpbakery.atlassian.net/wiki/display/VC/Release+Notes)

We are constantly working on a new features and improvements for Visual Composer in order to keep your site and your business at the top. In this section you can find information on each and every update we created during lifetime of Visual Composer.

|  |
| --- |
|  | | * Color picker z-index increased * Proper image gallery re-initialization after dragging * Negative values can be saved in Design options now * Single image now has circle and round styles * Shortcode mapper parsing shortcodes correctly now * Quotes (") can be used in the Custom heading element * Row stretching works as expected now * Position after cloning element is correct now * Auto rotate tabs works now * "h3 element converted to p tag if template is saved in frontend editor" fixed * "1/3 + 2/3" layout is added in grid builder * 3D shadow look improved for single image element * vc\_load\_default\_templates\_action returned * Custom query works properly in grid builder now |
|  | | * Extra CSS class can be used in grid elements now * Single image styles fixed * added action "vc\_load\_default\_templates\_action" * vc\_add\_default\_templates() fixed * Added compatibility for older Layer slider * Background disappearing in row on child element deleting fixed * Message box backwards compatibility added * Grid builder ajax stability improved * Css tpl for Design options regenerated * Custom query in Grid builder handles correctly ampersand and quote marks * Compatibility with jwplayer improved  [Licensing](https://wpbakery.atlassian.net/wiki/display/VC/Licensing) This section will explain all you have to know about licensing for Visual Composer and point you to necessary resources. The most important thing you need to know about licensing of Visual Composer is that we totally rely on licensing and copyrights developed by Envato who has an amazing licensing standards. At the moment Envato, so as Smartdatasoft, offers several types of licenses. **Available Licenses**  * [Regular License](https://wpbakery.atlassian.net/wiki/display/VC/Regular+License) * [Extended License](https://wpbakery.atlassian.net/wiki/display/VC/Extended+License) |

# [Regular License](https://wpbakery.atlassian.net/wiki/display/VC/Regular+License)

common type of licensing for Visual Composer. Regular license allows you to use item in one single end product which end user is not charged to access or use. You can acquire regular license for your own website, or if you are a developer - for one client by transferring the license to your client when the project is finished.

**Important:** Distribution of source files is not permitted.

In order to read more about [Regular license](http://codecanyon.net/licenses/regular) please visit CodeCanyon.

# [Extended License](https://wpbakery.atlassian.net/wiki/display/VC/Extended+License)

Extended License allows use of the item in one single end product, which you or one client may sell to end users (charge end users to access or use the end product), but user do not have access to the source files in any case.

**Important:** Distribution of source files is not permitted.

Extended license allows you, with the permission from Smartdatasoft, to integrate Visual Composer into your Prestashop theme to sell on ThemeForest. In order to read more about [Extended license](http://codecanyon.net/licenses/extended) please visit CodeCanyon.

# [Getting Started](https://wpbakery.atlassian.net/wiki/display/VC/Getting+Started)

Thank you for purchasing Visual Composer and congratulations with the huge win which you are going to experience once you start using our amazing page builder which will bring you new experience in building your Prestashop pages and also will save you tons of time.

This section will guide you through initial setup and activation processes in order to get your copy of Visual Composer up and running. Please follow sections step by step to get familiar with installation, activation and update processes. Moreover, as you may already know - Visual Composer works not only with your Prestashop pages, but also CMS and even custom cms types. See how you can activate Visual Composer for specific custom cms type and handle your custom CMS without any trouble.

## Contents

* [Module Installation](https://wpbakery.atlassian.net/wiki/display/VC/Plugin+Installation)
* [Activation](https://wpbakery.atlassian.net/wiki/display/VC/Activation)
* [Update](https://wpbakery.atlassian.net/wiki/display/VC/Update)
* Content

# [Module Installation](https://wpbakery.atlassian.net/wiki/display/VC/Plugin+Installation)

Visual Composer installation doesn't differ from any other module installation process, so you might be familiar with this process already. If not, please follow instructions below.

1. Login to your website
2. Go to Modules -> Add New Module
3. Click Upload link
4. Browse to the Module’s zip file (it is located in the archive you've downloaded from CodeCanyon) and choose that file.
5. Click "Install " button
6. Wait while module is uploaded to your server

Congratulations, you are all set! To start creating CMS Pages->Add new, Visual Composer Content Anywhere , Product Tab creator Add New and switch over to the Visual Composer mode. Also make sure to check Settings->Visual Composer for available settings

# [Learning More](https://wpbakery.atlassian.net/wiki/display/VC/Learning+More)

After installing and enabling Visual Composer Module you can start creating your layouts right away. In this section you will learn all necessary skills and find valuable information for becoming an expert in Visual Composer. Soon enough you will realize(if you haven't already) that Visual Composer is amazingly easy to use and it brings your 'web design' and 'content management' skills to the brand new level.

## **Contents**

* [Basic Concept](https://wpbakery.atlassian.net/wiki/display/VC/Basic+Concept)
* [Content Elements](https://wpbakery.atlassian.net/wiki/display/VC/Content+Elements)
* [General Settings](https://wpbakery.atlassian.net/wiki/display/VC/General+Settings)
* [Custom CSS](https://wpbakery.atlassian.net/wiki/display/VC/Custom+CSS)
* [Element Design Options](https://wpbakery.atlassian.net/wiki/display/VC/Element+Design+Options)
* [Responsive Settings](https://wpbakery.atlassian.net/wiki/display/VC/Responsive+Settings)
* [Templates](https://wpbakery.atlassian.net/wiki/display/VC/Templates)
* [Predefined Layouts](https://wpbakery.atlassian.net/wiki/display/VC/Predefined+Layouts)
* [Grid Builder](https://wpbakery.atlassian.net/wiki/display/VC/Grid+Builder)

# [Update](https://wpbakery.atlassian.net/wiki/display/VC/Update)

To update Visual Composer from older version, please follow this steps.

1. First, login to CodeCanyon and under your profile click Downloads link.
2. Re-download Visual Composer.
3. Now navigate to the Modules menu in your Prestashop back-office area.
4. Upload new version that downloaded few seconds earlier like as install new module.

# [Basic Concept](https://wpbakery.atlassian.net/wiki/display/VC/Basic+Concept)

There's one concept/design pattern you should know - in the Visual Composer editor mode you'll have "Add element" button, which you should click or drag to the working canvas, to add content block or row to your page. If you are adding row element to your page pay attention that it will automatically add column element within row. You can later add more columns within row and change their behavior. Read more about rows and columns in Elements section.

Clicking the "Add element" button will show the content element selection menu. Content blocks are divided into categories (Show all, Content, Social, Structure, Modules, etc.).

# [Content Elements](https://wpbakery.atlassian.net/wiki/display/VC/Content+Elements)

This is a list of 40+ available content elements that can be placed on the working canvas or inside of the columns. Think of them as bricks. With those bricks you are building your layout. Most of the content elements have options to set them click pencil icon. To save changes click save button.

## Contents

* [Contents](https://wpbakery.atlassian.net/wiki/display/VC/Content+Elements#ContentElements-Contents)
* [Row](https://wpbakery.atlassian.net/wiki/display/VC/Content+Elements#ContentElements-Row)
* [Column](https://wpbakery.atlassian.net/wiki/display/VC/Content+Elements#ContentElements-Column)
* [Text Block](https://wpbakery.atlassian.net/wiki/display/VC/Content+Elements#ContentElements-TextBlock)
* [Separator](https://wpbakery.atlassian.net/wiki/display/VC/Content+Elements#ContentElements-Separator)
* [Separator with Text](https://wpbakery.atlassian.net/wiki/display/VC/Content+Elements#ContentElements-SeparatorwithText)
* [Message Box](https://wpbakery.atlassian.net/wiki/display/VC/Content+Elements#ContentElements-MessageBox)
* [Facebook 'Like'](https://wpbakery.atlassian.net/wiki/display/VC/Content+Elements#ContentElements-Facebook%27Like%27)
* [Tweetmeme Button](https://wpbakery.atlassian.net/wiki/display/VC/Content+Elements#ContentElements-TweetmemeButton)
* [Google +](https://wpbakery.atlassian.net/wiki/display/VC/Content+Elements#ContentElements-Google+)
* [Pinterest](https://wpbakery.atlassian.net/wiki/display/VC/Content+Elements#ContentElements-Pinterest)
* [FAQ (Toggle)](https://wpbakery.atlassian.net/wiki/display/VC/Content+Elements#ContentElements-FAQ%28Toggle%29)
* [Single Image](https://wpbakery.atlassian.net/wiki/display/VC/Content+Elements#ContentElements-SingleImage)
* [Image Gallery](https://wpbakery.atlassian.net/wiki/display/VC/Content+Elements#ContentElements-ImageGallery)
* [Image Carousel](https://wpbakery.atlassian.net/wiki/display/VC/Content+Elements#ContentElements-ImageCarousel)
* [Tabs](https://wpbakery.atlassian.net/wiki/display/VC/Content+Elements#ContentElements-Tabs)
* [Tour](https://wpbakery.atlassian.net/wiki/display/VC/Content+Elements#ContentElements-Tour)
* [Accordion](https://wpbakery.atlassian.net/wiki/display/VC/Content+Elements#ContentElements-Accordion)
* [Cms Grid](https://wpbakery.atlassian.net/wiki/display/VC/Content+Elements#ContentElements-PostGrid)
* [Masonry Cms Grid](https://wpbakery.atlassian.net/wiki/display/VC/Content+Elements#ContentElements-MasonryPostGrid)
* [Media Grid](https://wpbakery.atlassian.net/wiki/display/VC/Content+Elements#ContentElements-MediaGrid)
* [Masonry Media Grid](https://wpbakery.atlassian.net/wiki/display/VC/Content+Elements#ContentElements-MasonryMediaGrid)
* [Cms Carousel](https://wpbakery.atlassian.net/wiki/display/VC/Content+Elements#ContentElements-PostCarousel)
* [Cms Slider](https://wpbakery.atlassian.net/wiki/display/VC/Content+Elements#ContentElements-PostSlider)
* [Button](https://wpbakery.atlassian.net/wiki/display/VC/Content+Elements#ContentElements-Button)
* [Button 2.0](https://wpbakery.atlassian.net/wiki/display/VC/Content+Elements#ContentElements-Button2.0)
* [Call to Action](https://wpbakery.atlassian.net/wiki/display/VC/Content+Elements#ContentElements-CalltoAction)
* [Call to Action 2.0](https://wpbakery.atlassian.net/wiki/display/VC/Content+Elements#ContentElements-CalltoAction2.0)
* [Video Player](https://wpbakery.atlassian.net/wiki/display/VC/Content+Elements#ContentElements-VideoPlayer)
* [Google Maps](https://wpbakery.atlassian.net/wiki/display/VC/Content+Elements#ContentElements-GoogleMaps)
* [Raw HTML](https://wpbakery.atlassian.net/wiki/display/VC/Content+Elements#ContentElements-RawHTML)
* [Raw JS](https://wpbakery.atlassian.net/wiki/display/VC/Content+Elements#ContentElements-RawJS)
* [Flickr Widget](https://wpbakery.atlassian.net/wiki/display/VC/Content+Elements#ContentElements-FlickrWidget)
* [Progress Bar](https://wpbakery.atlassian.net/wiki/display/VC/Content+Elements#ContentElements-ProgressBar)
* [Pie Chart](https://wpbakery.atlassian.net/wiki/display/VC/Content+Elements#ContentElements-PieChart)
* [Empty Space](https://wpbakery.atlassian.net/wiki/display/VC/Content+Elements#ContentElements-EmptySpace)
* [Custom Heading](https://wpbakery.atlassian.net/wiki/display/VC/Content+Elements#ContentElements-CustomHeading)
* [Icon Element](https://wpbakery.atlassian.net/wiki/display/VC/Content+Elements#ContentElements-IconElement)
* [Woo Elements](https://wpbakery.atlassian.net/wiki/display/VC/Content+Elements#ContentElements-WooElements)
* [Default Display Module of PS](https://wpbakery.atlassian.net/wiki/display/VC/Content+Elements#ContentElements-DefaultWidgetsofWP)
* [3rd Party Display Modules Support](https://wpbakery.atlassian.net/wiki/display/VC/Content+Elements#ContentElements-3rdPartyPluginsSupport)

## Row

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| https://wpbakery.atlassian.net/wiki/download/attachments/524364/add_row.png?version=1&modificationDate=1415895046870&api=v2 | | “Row“ is the main content element of Visual Composer. Rows are used to divide your page into the logic blocks with columns, columns later will hold your content blocks. Rows can be divided into the layouts (eg. 1/2 + 1/2, 1/3 + 1/3 + 1/3, and so on). Your page can have unlimited number of rows.   To change row's position, click and drag row's drag handler (top left row's corner) and drag row around (vertical axis). | | |
|  |
| Font color | | | Select font color for containing elements |
| Extra class name | | | Add class name in order to refer to this element in CSS |
| Design Options | | | Control borders, background and other styling options |

**Important:** Row allows you build complex layouts by inserting inner row within root level row/column. Take into account that it is not allowed to insert inner row within inner row.

## Column

“Columns“ are part of the row and they hold your content elements inside them. Columns can be reordered. Click and drag column around (horizontal axis). That way you can group elements in logical groups and then drag them around with your mouse (to re-position).

|  |
| --- |
| Font color | | Select font color for containing elements |
| Extra class name | | Add class name in order to refer to this element in CSS |
| Design Options | | Control borders, background and other styling options |
| Width & Responsiveness | | Control width, offset and visibility of element on different devices |

## Text Block

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| https://wpbakery.atlassian.net/wiki/download/attachments/524364/text.png?version=1&modificationDate=1415895047063&api=v2 | | “Text block“ allows you to insert paragraph type text and format it using TinyMCE editor. Moreover text block allows adding media(images and videos). | | |
|  |
| Text | | | Add/Edit content using editor TinyMCE |
| CSS animations | | | Add animation to your element |
| Extra class name | | | Add class name in order to refer to this element in CSS |
| Design Options | | | Control borders, background and other styling options |

## Separator

|  |  |  |  |
| --- | --- | --- | --- |
| https://wpbakery.atlassian.net/wiki/download/attachments/524364/separator.png?version=1&modificationDate=1415895047036&api=v2 | | “Separator“ add separator line to your page. | |
|  |
| Color | | | Choose color of your separator line, including Custom color chosen with Color picker or defined as RGB code | |
| Separator alignment | | | Select separator alignment. | |
| Style | | | Choose line styling - solid, dotted, dashed etc. | |
| Border | | | Border with in pixels. | |
| Element width | | | Define line width within column in percents | |
| Extra class name | | | Add class name in order to refer to this element in CSS | |

## Separator with Text

|  |  |  |  |
| --- | --- | --- | --- |
| https://wpbakery.atlassian.net/wiki/download/attachments/524364/separator_text.png?version=1&modificationDate=1415895047042&api=v2 | | “Separator with Text“ add separator line to your page together with text which can be positioned in the centre, left or right sides of line. | |
|  |
| Title | | | Text which will appear on the line |
| Title position | | | Allows positioning of your text in the centre, left or right side of the line |
| Separator alignment | | | Select separator alignment. |
| Color | | | Choose color of your separator line, including Custom color chosen with Color picker or defined as RGB code |
| Style | | | Choose line styling - solid, dotted, dashed etc. |
| Border | | | Border with in pixels. |
| Element width | | | Define line width within column in percents |
| Extra class name | | | Add class name in order to refer to this element in CSS |

## Message Box

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| https://wpbakery.atlassian.net/wiki/download/attachments/524364/message_box.png?version=1&modificationDate=1415895046974&api=v2 | | “Message Box“ add noticeable message to your website in order to show success, error, warning etc. | | |
|  |
| Message Box Presets | | | Select predefined message box designs or choose "Custom" for custom styling. |
| Style | | | Select message box style. |
| Shape | | | Select message box shape. |
| Color | | | Select message box color. |
| Icon library | | | Select icon library. |
| Icon | | | Select icon from library. |
| Message text | | | Add/Edit content using editor TinyMCE |
| CSS animation | | | Add animation to your element. |
| Extra class name | | | Add class name in order to refer to this element in CSS |

## Facebook 'Like'

|  |  |  |  |
| --- | --- | --- | --- |
| https://wpbakery.atlassian.net/wiki/download/attachments/524364/facebook.png?version=1&modificationDate=1415895046909&api=v2 | | “Facebook Like“ add social 'Like' button. | |
|  |
| Button type | | | Select type of your button from predefined layouts | |

## Tweetmeme Button

|  |  |  |  |
| --- | --- | --- | --- |
| https://wpbakery.atlassian.net/wiki/download/attachments/524364/twitter.png?version=1&modificationDate=1415895047077&api=v2 | | “Tweetmeme button“ add social 'Tweet' button. | |
|  |
| Button type | | | Select type of your button from predefined layouts | |

## Google +

|  |  |  |  |
| --- | --- | --- | --- |
| https://wpbakery.atlassian.net/wiki/download/attachments/524364/google_plus.png?version=1&modificationDate=1415895046947&api=v2 | | “Google+ button“ add social “Google+“ button. | |
|  |
| Button size | | | Select size of the button | |
| Annotation | | | Select type of annotation presented together with button | |

## Pinterest

|  |  |  |  |
| --- | --- | --- | --- |
| https://wpbakery.atlassian.net/wiki/download/attachments/524364/pinterest.png?version=1&modificationDate=1415895047000&api=v2 | | “Pinterest“ add social 'Pinit' button. | |
|  |
| Button layout | | | Select type of button layout | |

## FAQ (Toggle)

|  |  |  |  |
| --- | --- | --- | --- |
| https://wpbakery.atlassian.net/wiki/download/attachments/524364/faq.png?version=1&modificationDate=1415895046918&api=v2 | | “FAQ“ add collapsible toggle to your page. | |
|  |
| Toggle title | | | Title of toggle which will be displayed near open/collapse button | |
| Toggle content | | | Content of toggle editable using editor TinyMCE | |
| Style | | | Toggle style. | |
| Color | | | Toggle icon color. | |
| Size | | | Toggle size. | |
| Default state | | | Ability to set default state of toggle open or closed(collapsed) | |
| CSS animation | | | Add animation to your element | |
| Extra class name | | | Add class name in order to refer to this element in CSS | |

## Single Image

|  |  |  |  |
| --- | --- | --- | --- |
| https://wpbakery.atlassian.net/wiki/download/attachments/524364/single_image.png?version=1&modificationDate=1415895047049&api=v2 | | Add simple image to your layout and style it. Add CSS animation or apply Design Options. | |
|  |
| Widget title | | | Title of element which will be displayed above widget | |
| Image | | | Select image from media library. You can also upload images using media library | |
| CSS animation | | | Add animation to your element | |
| Image size | | | Set size of your image. Set large, small etc. or set your own width x height in pixels | |
| Image alignment | | | Set alignment of element within a column | |
| Image style | | | Set image style - choose from square, rounded, border etc. | |
| Link to Large Image | | | Set if you want to open large image on click | |
| Image link | | | Add link to your image | |
| Extra class name | | | Add class name in order to refer to this element in CSS | |

## Image Gallery

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| https://wpbakery.atlassian.net/wiki/download/attachments/524364/gallery.png?version=1&modificationDate=1415895046934&api=v2 | | Add responsive Image gallery. Choose your gallery type from Flex Slider, Nivo Slider or Image grid. | | |
|  |
| Widget title | | | Title of element which will be displayed above widget |
| Auto rotate slides | | | Select interval between auto-rotation of images or disable it |
| Images | | | Select images from media library. You can also upload images using media library |
| Image size | | | Set size of your image. Set large, small etc. or set your own width x height in pixels |
| On click | | | Define action for onclick event if needed |
| Extra class name | | | Add class name in order to refer to this element in CSS |

## Image Carousel

|  |  |  |  |
| --- | --- | --- | --- |
| https://wpbakery.atlassian.net/wiki/download/attachments/524364/image_carrousel.png?version=1&modificationDate=1415895046961&api=v2 | | Add fixed size animated image carousel. | |
|  |
| Widget title | | | Title of element which will be displayed above widget | |
| Auto rotate slides | | | Select interval between auto-rotation of images or disable it | |
| Images | | | Select images from media library. You can also upload images using media library | |
| Image size | | | Set size of your image. Set large, small etc. or set your own width x height in pixels | |
| On click | | | Define action for onclick event if needed | |
| Slider mode | | | Slides will be positioned horizontally (for horizontal swipes) or vertically (for vertical swipes) | |
| Slider speed | | | Duration of animation between slides (in ms) | |
| Slides per view | | | Set numbers of slides you want to display at the same time on slider's container for carousel mode. Supports also "auto" value, in this case it will fit slides depending on container's width. "auto" mode isn't compatible with loop mode | |
| Slider autoplay | | | Enables autoplay mode | |
| Hide pagination control | | | If YES pagination control will be removed | |
| Hide prev/next buttons | | | If YES prev/next control will be removed | |
| Partial view | | | If YES part of the next slide will be visible on the right side | |
| Slider loop | | | Enables loop mode | |
| Extra class name | | | Add class name in order to refer to this element in CSS | |

## Tabs

|  |  |
| --- | --- |
| https://wpbakery.atlassian.net/wiki/download/attachments/524364/tabs.png?version=1&modificationDate=1415895047056&api=v2 | “Tabs“ is a complex element which consists of inner section collections(tabs) and its structure is similar to row and column hierarchy. You can create as many inner sections within element as you wish and then place any type of content within.  Tabs and tab element has different parameters to operate with. |

**Tabs Parameters**

|  |
| --- |
| Widget title | | Title of element which will be displayed above widget |
| Auto rotate tabs | | Auto rotate tabs each X seconds |
| Extra class name | | Add class name in order to refer to this element in CSS |

**Tab Parameters**

|  |
| --- |
| Title | | Title of tab which will be displayed in Tab header |
| Tab ID | | unique tab id which can be used within link (can not be modified) |

**Important:** Tabs are considered as complex elements which do not allow to insert other complex elements within.

## Tour

|  |  |
| --- | --- |
| https://wpbakery.atlassian.net/wiki/download/attachments/524364/tours.png?version=1&modificationDate=1415895047070&api=v2 | “Tour“ is a complex element which consists of inner sections(tabs) collection and its structure is similar to row and column hierarchy. You can create as many inner sections within element as you wish and then place any type of content within.  Tour and inner tab element has different parameters to operate with. |

**Tour Parameters**

|  |
| --- |
| Widget title | | Title of element which will be displayed above widget |
| Auto rotate tabs | | Auto rotate tabs each X seconds |
| Extra class name | | Add class name in order to refer to this element in CSS |

**Tab Parameters**

|  |
| --- |
| Title | | Title of tab which will be displayed in Tab header |
| Tab ID | | unique tab id which can be used within link (can not be modified) |

**Important:** Tour are considered as complex elements which do not allow to insert other complex elements within.

## Accordion

|  |  |
| --- | --- |
| https://wpbakery.atlassian.net/wiki/download/attachments/524364/accordion.png?version=1&modificationDate=1415895046849&api=v2 | “Accordion“ is a complex element which consists of inner sections collection and its structure is similar to row and column hierarchy. You can create as many inner sections within element as you wish and then place any type of content within.  Accordion and inner section element has different parameters to operate with. |

**Tour Parameters**

|  |
| --- |
| Widget title | | Title of element which will be displayed above widget |
| Active section | | Enter section number to be active on load or enter false to collapse all sections |
| Allow collapsible all | | Select to allow all sections to be collapsible |
| Extra class name | | Add class name in order to refer to this element in CSS |

**Tab Parameters**

|  |
| --- |
| Title | | Title of section which will be displayed in Section header |

**Important:** Accordion are considered as complex elements which do not allow to insert other complex elements within.

SINCE 4.4

## Grid

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| https://wpbakery.atlassian.net/wiki/download/attachments/524364/post_grid.png?version=1&modificationDate=1415895047015&api=v2 | | Add CMS of your Prestashop site in grid view. Choose CMS type, build your own unique query and define what kind of information to display. | | |
|  |
| Data source | | | General | Select content type for your grid. |
| Narrow data source | | | General | Enter categories, tags or custom taxonomies. |
| Total items | | | General | Set max limit for items in grid or enter -1 to display all (limited to 1000). |
| Display Style | | | General | Select display style for grid. |
| Show filter | | | General | Append filter to grid. |
| Grid elements per row | | | General | Select number of single grid elements per row. |
| Gap | | | General | Select gap between grid elements. |
| Include only | | | General | Add CMS, pages, etc. by title. **Note:** active only if "Data source" is set to "List of IDs" |
| Custom query | | | General | Build custom query according to [Prestashop Codex](http://codex.wordpress.org/Function_Reference/query_posts). **Note:** active only if "Data source" is set to "Custom query" |
| Items per page | | | General | Number of items to show per page. Note: active only if "Display style" is set to "Load more button", "Lazy loading" or "Paginator" |
| Order by | | | Data settings | Select order type. If "Meta value" or "Meta value Number" is chosen then meta key is required. |
| Sorting | | | Data settings | Select sorting order. |
| Offset | | | Data settings | Number of grid elements to displace or pass over. |
| Exclude | | | Data settings | Exclude CMS, pages, etc. by title. |
| Grid element template | | | Item design | Select predefined grid element template.  **Note:** It is allowed to create new template or modify selected. Predefined templates will be cloned. |
| Filter by | | | Filter | Select filter source. |
| Style | | | Filter | Select filter display style. |
| Alignment | | | Filter | Select filter alignment. |
| Color | | | Filter | Select filter color. |
| Filter size | | | Filter | Select filter size. |
| Button style | | | Load more button | Select button style. |
| Button color | | | Load more button | Select button color. |
| Button size | | | Load more button | Select button size. |
| Arrows design | | | Pagination | Select design for arrows. |
| Arrows position | | | Pagination | Arrows will be displayed inside or outside grid. **Note:** active only if "Arrow design" is chosen |
| Arrows color | | | Pagination | Select color for arrows. **Note:** active only if "Arrow design" is chosen |
| Pagination style | | | Pagination | Select pagination style. |
| Pagination color | | | Pagination | Select pagination color. **Note:** active only if "Pagination style" is chosen |
| Loop pages? | | | Pagination | Allow items to be repeated in infinite loop (carousel). |
| Autoplay delay | | | Pagination | Enter value in seconds. Set -1 to disable autoplay. |
| Animation In | | | Pagination | Select "animation in" for page transition. |
| Animation Out | | | Pagination | Select "animation out" for page transition. |

SINCE 4.4

## Masonry Grid

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| https://wpbakery.atlassian.net/wiki/download/attachments/524364/masonry_post_grid_ico.png?version=1&modificationDate=1421671664727&api=v2 | | Add CMS of your Prestashop site in masonry grid view. Choose CMS type, build your own unique query and define what kind of information to display. | | |
|  |
| Data source | | | General | Select content type for your grid. |
| Narrow data source | | | General | Enter categories, tags or custom taxonomies. |
| Total items | | | General | Set max limit for items in grid or enter -1 to display all (limited to 1000). |
| Display Style | | | General | Select display style for grid. |
| Show filter | | | General | Append filter to grid. |
| Grid elements per row | | | General | Select number of single grid elements per row. |
| Gap | | | General | Select gap between grid elements. |
| Include only | | | General | Add CMS, pages, etc. by title. **Note:** active only if "Data source" is set to "List of IDs" |
| Custom query | | | General | Build custom query according to Prestashop Forum and Documentation. |
| Items per page | | | General | Number of items to show per page. Note: active only if "Display style" is set to "Load more button", "Lazy loading" or "Paginator" |
| Order by | | | Data settings | Select order type. If "Meta value" or "Meta value Number" is chosen then meta key is required. |
| Sorting | | | Data settings | Select sorting order. |
| Offset | | | Data settings | Number of grid elements to displace or pass over. |
| Exclude | | | Data settings | Exclude CMS, pages, etc. by title. |
| Grid element template | | | Item design | Select predefined grid element template.  **Note:** It is allowed to create new template or modify selected. Predefined templates will be cloned. |
| Filter by | | | Filter | Select filter source. |
| Style | | | Filter | Select filter display style. |
| Alignment | | | Filter | Select filter alignment. |
| Color | | | Filter | Select filter color. |
| Filter size | | | Filter | Select filter size. |
| Button style | | | Load more button | Select button style. |
| Button color | | | Load more button | Select button color. |
| Button size | | | Load more button | Select button size. |

SINCE 4.4

## Media Grid

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| https://wpbakery.atlassian.net/wiki/download/attachments/524364/media_grid_ico.png?version=1&modificationDate=1421671665041&api=v2 | | Add media from Media library to your Prestashop site in grid view. | | |
|  |
| Images | | | General | Select images from media library. | |
| Display Style | | | General | Select display style for grid. | |
| Show filter | | | General | Append filter to grid. | |
| Grid elements per row | | | General | Select number of single grid elements per row. | |
| Gap | | | General | Select gap between grid elements. | |
| Include only | | | General | Add CMS, pages, etc. by title. **Note:** active only if "Data source" is set to "List of IDs" | |
| Custom query | | | General | Build custom query according to Prestashop Forum and Documentation. **Note:** active only if "Data source" is set to "Custom query" | |
| Items per page | | | General | Number of items to show per page. Note: active only if "Display style" is set to "Load more button", "Lazy loading" or "Paginator" | |
| Grid element template | | | Item design | Select predefined grid element template.  **Note:** It is allowed to create new template or modify selected. Predefined templates will be cloned. | |
| Filter by | | | Filter | Select filter source. | |
| Style | | | Filter | Select filter display style. | |
| Alignment | | | Filter | Select filter alignment. | |
| Color | | | Filter | Select filter color. | |
| Filter size | | | Filter | Select filter size. | |
| Button style | | | Load more button | Select button style. | |
| Button color | | | Load more button | Select button color. | |
| Button size | | | Load more button | Select button size. | |
| Arrows design | | | Pagination | Select design for arrows. | |
| Arrows position | | | Pagination | Arrows will be displayed inside or outside grid. **Note:** active only if "Arrow design" is chosen | |
| Arrows color | | | Pagination | Select color for arrows. **Note:** active only if "Arrow design" is chosen | |
| Pagination style | | | Pagination | Select pagination style. | |
| Pagination color | | | Pagination | Select pagination color. **Note:** active only if "Pagination style" is chosen | |
| Loop pages? | | | Pagination | Allow items to be repeated in infinite loop (carousel). | |
| Autoplay delay | | | Pagination | Enter value in seconds. Set -1 to disable autoplay. | |
| Animation In | | | Pagination | Select "animation in" for page transition. | |
| Animation Out | | | Pagination | Select "animation out" for page transition. | |

SINCE 4.4

## Masonry Media Grid

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| https://wpbakery.atlassian.net/wiki/download/attachments/524364/masonry_media_grid_ico.png?version=1&modificationDate=1421671664346&api=v2 | | Add media from Media library to your Prestashop site in masonry grid view. | | |
|  |
| Images | | | General | Select images from media library. | |
| Display Style | | | General | Select display style for grid. | |
| Show filter | | | General | Append filter to grid. | |
| Grid elements per row | | | General | Select number of single grid elements per row. | |
| Gap | | | General | Select gap between grid elements. | |
| Include only | | | General | Add CMS, pages, etc. by title. **Note:** active only if "Data source" is set to "List of IDs" | |
| Custom query | | | General | Build custom query according to Prestashop Forum and Documentation . **Note:** active only if "Data source" is set to "Custom query" | |
| Items per page | | | General | Number of items to show per page. Note: active only if "Display style" is set to "Load more button", "Lazy loading" or "Paginator" | |
| Grid element template | | | Item design | Select predefined grid element template.  **Note:** It is allowed to create new template or modify selected. Predefined templates will be cloned. | |
| Filter by | | | Filter | Select filter source. | |
| Style | | | Filter | Select filter display style. | |
| Alignment | | | Filter | Select filter alignment. | |
| Color | | | Filter | Select filter color. | |
| Filter size | | | Filter | Select filter size. | |
| Button style | | | Load more button | Select button style. | |
| Button color | | | Load more button | Select button color. | |
| Button size | | | Load more button | Select button size. | |

## Carousel

|  |  |  |
| --- | --- | --- |
| https://wpbakery.atlassian.net/wiki/download/attachments/524364/post_carrousel.png?version=1&modificationDate=1415895047008&api=v2 |  | Add CMS of your Prestashop site in carousel view. Choose CMS type, build your own unique query and define what kind of information to display.  **Important:** do not place CMS carrousel element within same type of CMS. Such option will trigger infinite loop. |

## Slider

|  |  |
| --- | --- |
| https://wpbakery.atlassian.net/wiki/download/attachments/524364/posts_slider.png?version=1&modificationDate=1415895047022&api=v2 | Create slider from your Prestashop site CMS, pages or custom CMS. Display featured images and take full control over their settings and styling. |

## Button

|  |  |  |  |
| --- | --- | --- | --- |
| https://wpbakery.atlassian.net/wiki/download/attachments/524364/button.png?version=1&modificationDate=1415895046878&api=v2 | | Add button with or without icon to your layout. Control size, link and text of the button. | |
|  |
| Text on the button | | | Text which will appear on the button | |
| URL (Link) | | | Button link | |
| Color | | | Color of the button | |
| Icon | | | Set icon which you want to display on button. Icon is not mandatory and by default field value is 'None' | |
| Size | | | Choose one of the predefined sizes for your button | |
| Extra class name | | | Add class name in order to refer to this element in CSS | |

## Button 2.0

|  |  |  |  |
| --- | --- | --- | --- |
| https://wpbakery.atlassian.net/wiki/download/attachments/524364/button.png?version=1&modificationDate=1415895046878&api=v2 | | Add button with multiple color and styling options. | |
|  |
| Text on the button | | | Text which will appear on the button | |
| URL (Link) | | | Button link | |
| Button Alignment | | | Select button alignment. | |
| Style | | | Select predefined style of the button, including square, rounded, outline, round, etc. | |
| Color | | | Color of the button | |
| Size | | | Choose one of the predefined sizes for your button | |
| Extra class name | | | Add class name in order to refer to this element in CSS | |

## Call to Action

|  |  |  |  |
| --- | --- | --- | --- |
| https://wpbakery.atlassian.net/wiki/download/attachments/524364/call_to_action.png?version=1&modificationDate=1415895046885&api=v2 | | Create call to action block with text, button and control its styling. | |
|  |
| Text | | | Text which will appear on call to action block | |
| Text on the button | | | Text which will appear on the button | |
| Color | | | Color of the button | |
| Icon | | | Set icon which you want to display on button. Icon is not mandatory and by default field value is "None" | |
| Size | | | Choose one of the predefined sizes for your button | |
| Button position | | | Choose one of predefined button positions | |
| CSS animation | | | Add animation to your element | |
| Extra class name | | | Add class name in order to refer to this element in CSS | |

## Call to Action 2.0

|  |  |  |  |
| --- | --- | --- | --- |
| https://wpbakery.atlassian.net/wiki/download/attachments/524364/call_to_action.png?version=1&modificationDate=1415895046885&api=v2 | | Create call to action block with heading, text, button and control its styling. | |
|  |
| Heading first line | | | Text which will appear as a heading on call to action block | |
| Heading second line | | | Text which will appear as a second heading on call to action block | |
| CTA style | | | Choose styling of call to action block from round, square, etc. | |
| Element width | | | Call to action element width in percents within a column | |
| Text align | | | Choose text align within call to action block | |
| Custom Background Color | | | Choose custom background color for call to action block with color picker which allow control opacity | |
| Promotional text | | | Add/Edit text of call to action block using editor TinyMCE | |
| Text on the button | | | Text which will appear on the button | |
| URL (Link) | | | Button link | |
| Style | | | Select predefined style of the button, including square, rounded, outline, round, etc. | |
| Color | | | Color of the button | |
| Size | | | Choose one of the predefined sizes for your button | |
| Button position | | | Choose one of predefined button positions | |
| CSS animation | | | Add animation to your element | |
| Extra class name | | | Add class name in order to refer to this element in CSS | |

## Video Player

|  |  |  |  |
| --- | --- | --- | --- |
| https://wpbakery.atlassian.net/wiki/download/attachments/524364/video.png?version=1&modificationDate=1415895047083&api=v2 | | Insert video in your layout. Choose from multiple formats accepted by Prestashop. | |
|  |
| Widget title | | | Title of widget which will be displayed above widget | |
| Video link | | | Link to the video. More about supported formats at Prestashop Forum and Documentation. | |
| CSS animation | | | Add animation to your element | |
| Extra class name | | | Add class name in order to refer to this element in CSS | |
| Design Options | | | Control borders, background and other styling options | |

## Google Maps

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| https://wpbakery.atlassian.net/wiki/download/attachments/524364/google_map.png?version=1&modificationDate=1415895046941&api=v2 | | Insert Google Maps within your iframe. Create Google Maps using [Google Map Engine](https://mapsengine.google.com/map/) and easily insert generated code within your Prestashop layout. | | |
|  |
| Widget title | | | Title of widget which will be displayed above widget |
| Map embed iframe | | | 1. Visit Google maps to create your map. 2. Find location 3. Click "Share" and make sure map is public on the web 4. Click folder icon to reveal "Embed on my site" link 5. Copy iframe code and paste it |
| Map height | | | Enter map height in pixels |
| Extra class name | | | Add class name in order to refer to this element in CSS |

## Raw HTML

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| https://wpbakery.atlassian.net/wiki/download/attachments/524364/html.png?version=1&modificationDate=1415895046955&api=v2 | | Insert your custom HTML content if necessary | | |
|  |
| Raw HTML | | | Enter your HTML content |

## Raw JS

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| https://wpbakery.atlassian.net/wiki/download/attachments/524364/js.png?version=1&modificationDate=1415895046968&api=v2 | | Insert your custom JS code if necessary | | |
|  |
| Raw JS | | | Enter your JS code |

## Flickr Widget

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| https://wpbakery.atlassian.net/wiki/download/attachments/524364/flickr.png?version=1&modificationDate=1415895046925&api=v2 | | Display your photo stream from Flickr on your Prestashop site with Flickr Widget. In order to place images you will need to determine your Flickr ID - which you can retrieve here: <http://idgettr.com/> | | |
|  |
| Widget title | | | Title of widget which will be displayed above widget |
| Flickr ID | | | Enter Flickr ID |
| Number of photos | | | Choose number of photos to display |
| Type | | | Choose photo stream type |
| Display | | | Choose photo display order |
| Extra class name | | | Add class name in order to refer to this element in CSS |

## Progress Bar

|  |  |  |  |
| --- | --- | --- | --- |
| https://wpbakery.atlassian.net/wiki/download/attachments/524364/progress_bar.png?version=1&modificationDate=1415895047028&api=v2 | | Display progress bar with custom values. Style your whole progress bar or separate values according to your needs. | |
|  |
| Widget title | | | Title of widget which will be displayed above widget |
| Graphic values | | | Development|#e75956 |
| Units | | | Enter measurement units (if needed) Eg. %, px, points, etc. Graph value and unit will be appended to the graph title |
| Bar color | | | Select bar background color |
| Add Stripes? | | | Add stripes to your progress bars |
| Add animation? | | | Add animation to stripes. Will be visible with striped bars |
| Extra class name | | | Add class name in order to refer to this element in CSS |

## Pie Chart

|  |  |  |  |
| --- | --- | --- | --- |
| https://wpbakery.atlassian.net/wiki/download/attachments/524364/piechart.png?version=1&modificationDate=1415895046981&api=v2 | | Display pie chart with your custom value and styling parameters. | |
|  |
| Widget title | | | Title of widget which will be displayed above widget | |
| Pie value | | | Input graph value here. Choose range between 0 and 100 | |
| Pie label value | | | Input integer value for label. If empty "Pie value" will be used | |
| Units | | | Enter measurement units (if needed) Eg. %, px, points, etc. Graph value and unit will be appended to the graph title | |
| Bar color | | | Select pie chart color | |
| Extra class name | | | Add class name in order to refer to this element in CSS | |

SINCE 4.3

## Empty Space

|  |  |  |  |
| --- | --- | --- | --- |
| https://wpbakery.atlassian.net/wiki/download/attachments/524364/empty_space.png?version=1&modificationDate=1415895046901&api=v2 | | Insert empty space between elements with custom height. | |
|  |
| Height | | | Enter empty space height | |
| Extra class name | | | Add class name in order to refer to this element in CSS | |

SINCE 4.3

## Custom Heading

|  |  |  |  |
| --- | --- | --- | --- |
| https://wpbakery.atlassian.net/wiki/download/attachments/524364/custom_heading.png?version=1&modificationDate=1415895046893&api=v2 | | Create custom heading with Google Fonts, choose one of predefined fonts and apply styling.  **Important:** If you are using non-latin characters be sure to activate them under Settings/Visual Composer/General Settings. | |
|  |
| Text | | | Enter your content |
| Element tag | | | Select element tag |
| Text align | | | Select text alignment |
| Font size | | | Select font size. By default the size defined by your theme for specific tag will be used |
| Line height | | | Select line height to set space between lines |
| Text color | | | Select color of your text |
| Font Family | | | Select Google Font family from predefined list |
| Font style | | | Select font styling available for chosen font family. For example light, normal, bold, italic, etc. |
| Google Fonts preview | | | See how you chosen Font Family and Font style looks like |
| Extra class name | | | Add class name in order to refer to this element in CSS |

SINCE 4.4

## Icon Element

|  |  |  |  |
| --- | --- | --- | --- |
| https://wpbakery.atlassian.net/wiki/download/attachments/524364/icon_icon.png?version=1&modificationDate=1415964149709&api=v2 | | With Visual Composer you can easily add icons from following libraries:   * Font Awesome * Open icons * Typicons * Entype * Linecons | |
|  |
| Icons source | | | Select icon source (Library) | |
| Icon | | | Select icon from chosen Icon source (Library) | |
| Color | | | Select color of your icon | |
| Background style | | | Select background style for your icon (default is 'None') | |
| Background color | | | Select color of icon's background (only if Background style is chosen) | |
| Size | | | Select icon size from Mini, Small, Medium, Large, Extra Large | |
| Icon alignment | | | Select icon alignment | |
| URL (link) | | | Icon link | |
| CSS animation | | | Add animation to your element | |
| Extra class name | | | Add class name in order to refer to this element in CSS | |

|  |
| --- |
| Widget title | | Title of widget which will be displayed above widget |
| Extra class name | | Add class name in order to refer to this element in CSS |

## 3rd Party Modules Support

Occasionally Visual Composer is extended with support for popular third party Modules. Currently this third party Modules are supported:

1. Revolution Slider
2. SmartShortCode
3. SmartBlog
4. Also Display all available hooks Modules

After placing content element to the working canvas (page), click pencil icon to see options available for this particular shortcode (content element).

# [General Settings](https://wpbakery.atlassian.net/wiki/display/VC/General+Settings)

General Settings of Visual Composer can be managed in your Back-office - Settings/Visual Composer. General Settings allows editing of general configuration.

* Controlling CSS and JS Load.
* Changing Product Page Tab Style.
* You can change Image type and Image Size.

# [Element Design Options](https://wpbakery.atlassian.net/wiki/display/VC/Element+Design+Options)

Starting from version 4.1.0 Visual Composer offers additional design controls for Row, Columns, Text block, Single image and Video block which can be found under element param window 'Design Options' tab.

Within Design Options it is possible to control:

* Margins, Paddings and Border for overall element or just left, right, top or bottom part
* Border style
* Border color
* Background color
* Background image and position style

# [Predefined Layouts](https://wpbakery.atlassian.net/wiki/display/VC/Predefined+Layouts)

You can select from predefined started templates and [if you are a developer you can modify that list](https://wpbakery.atlassian.net/wiki/pages/viewpage.action?pageId=524300) . Templates are featured at the Visual Composer welcome page and are accessible by clicking Templates button in Visual Composer top bar and then navigating to "Default templates" tab.

# [Visual Composer "How To's"](https://wpbakery.atlassian.net/wiki/pages/viewpage.action?pageId=4030510)

In this section you will find a quick tips in form of video tutorials on how to operate with Visual Composer.

## Contents

* [How to add and edit images?](https://wpbakery.atlassian.net/wiki/pages/viewpage.action?pageId=4030535)
* [How to add Image Gallery?](https://wpbakery.atlassian.net/wiki/pages/viewpage.action?pageId=4030517)
* [How to add old style cms grid with Grid Builder?](https://wpbakery.atlassian.net/wiki/pages/viewpage.action?pageId=4030552)
* [How to add Text Block?](https://wpbakery.atlassian.net/wiki/pages/viewpage.action?pageId=4030531)
* [How to check responsiveness?](https://wpbakery.atlassian.net/wiki/pages/viewpage.action?pageId=4030519)
* [How to clone elements?](https://wpbakery.atlassian.net/wiki/pages/viewpage.action?pageId=4030529)
* [How to control elements on different devices?](https://wpbakery.atlassian.net/wiki/pages/viewpage.action?pageId=4030539)
* [How to create empty space between elements?](https://wpbakery.atlassian.net/wiki/pages/viewpage.action?pageId=4030543)
* [How to drag and drop elements?](https://wpbakery.atlassian.net/wiki/pages/viewpage.action?pageId=4030527)
* [How to edit cms title?](https://wpbakery.atlassian.net/wiki/pages/viewpage.action?pageId=4030533)
* [How to insert inner row?](https://wpbakery.atlassian.net/wiki/pages/viewpage.action?pageId=4030521)
* [How to minimize row?](https://wpbakery.atlassian.net/wiki/pages/viewpage.action?pageId=4030537)
* [How to switch "Guide Mode" on/off in Frontend editor?](https://wpbakery.atlassian.net/wiki/pages/viewpage.action?pageId=4030523)
* [How to use Custom Heading element?](https://wpbakery.atlassian.net/wiki/pages/viewpage.action?pageId=4030541)
* [How to work with Tabs, Tours and Accordions?](https://wpbakery.atlassian.net/wiki/pages/viewpage.action?pageId=4030515)

# [FAQ](https://wpbakery.atlassian.net/wiki/display/VC/FAQ)

Here you can find answers to the Frequently Asked Question about Visual Composer.

## Contents

* [Does Visual Composer works with any Prestashop theme?](https://wpbakery.atlassian.net/wiki/pages/viewpage.action?pageId=1441797)
* [Where can I find my purchase code?](https://wpbakery.atlassian.net/wiki/pages/viewpage.action?pageId=524307)
* [Where can I find API Key?](https://wpbakery.atlassian.net/wiki/pages/viewpage.action?pageId=524301)
* [How to deactivate and reactivate license?](https://wpbakery.atlassian.net/wiki/pages/viewpage.action?pageId=524321)
* [I keep getting this error: "The module does not have a valid header"](https://wpbakery.atlassian.net/wiki/pages/viewpage.action?pageId=524340)
* [I'm trying to install Visual Composer, but Prestashop is saying "The package could not be installed. No valid modules were found."](https://wpbakery.atlassian.net/wiki/pages/viewpage.action?pageId=524328)
* [When installing Visual Composer I get error:"Fatal error: Cannot redeclare ... (previously declared in ...)"](https://wpbakery.atlassian.net/wiki/pages/viewpage.action?pageId=524367)
* [There's a new version of Visual Composer available, how can I get it and update it on my site?](https://wpbakery.atlassian.net/wiki/pages/viewpage.action?pageId=524338)
* [When trying to update I receive message 'Update Package not available'](https://wpbakery.atlassian.net/wiki/pages/viewpage.action?pageId=524336)
* [Can I update Visual Composer if I have purchased it in the theme?](https://wpbakery.atlassian.net/wiki/pages/viewpage.action?pageId=524337)
* [After clicking Visual Composer button I see spinning icon for indefinite time...](https://wpbakery.atlassian.net/wiki/pages/viewpage.action?pageId=524315)
* [I can not use Visual Composer on CMS and Custom CMS](https://wpbakery.atlassian.net/wiki/display/VC/I+can+not+use+Visual+Composer+on+Posts+and+Custom+Posts)
* [How to insert Google Maps?](https://wpbakery.atlassian.net/wiki/pages/viewpage.action?pageId=524325)
* [How to manage CSS with Visual Composer?](https://wpbakery.atlassian.net/wiki/pages/viewpage.action?pageId=524319)
* [How to reduce vertical space between rows](https://wpbakery.atlassian.net/wiki/display/VC/How+to+reduce+vertical+space+between+rows)
* [Content is not saved after the cms/page update](https://wpbakery.atlassian.net/wiki/pages/viewpage.action?pageId=524327)
* [Can I get white label version of Visual Composer?](https://wpbakery.atlassian.net/wiki/pages/viewpage.action?pageId=524306)
* [Can I include Visual Composer in my end-product?](https://wpbakery.atlassian.net/wiki/pages/viewpage.action?pageId=524322)
* [Can I include Visual Composer in my theme?](https://wpbakery.atlassian.net/wiki/pages/viewpage.action?pageId=524323)
* [What about GPL and Split License?](https://wpbakery.atlassian.net/wiki/pages/viewpage.action?pageId=524330)
* [Can I use Visual Composer on several Prestashop sites (Multisite/Developers license)?](https://wpbakery.atlassian.net/wiki/pages/viewpage.action?pageId=524305)
* [Can I insert 3rd party short codes?](https://wpbakery.atlassian.net/wiki/pages/viewpage.action?pageId=524334)
* [Can I add my own short codes sets?](https://wpbakery.atlassian.net/wiki/pages/viewpage.action?pageId=524317)
* [I have a layout that I created with Visual Composer and now I want to "clone" it for use on the other page/cms, how can I do that?](https://wpbakery.atlassian.net/wiki/pages/viewpage.action?pageId=524318)
* [Call to undefined method WP\_Error::resize()](https://wpbakery.atlassian.net/wiki/pages/viewpage.action?pageId=524326)
* [Where Can I Get Official Support?](https://wpbakery.atlassian.net/wiki/pages/viewpage.action?pageId=524302)
* [How to make full width row?](https://wpbakery.atlassian.net/wiki/pages/viewpage.action?pageId=3604483)
* [How to make custom grid element?](https://wpbakery.atlassian.net/wiki/pages/viewpage.action?pageId=3604488)
* [How to modify predefined grid element?](https://wpbakery.atlassian.net/wiki/pages/viewpage.action?pageId=3604489)
* [How to add cms or media grid?](https://wpbakery.atlassian.net/wiki/pages/viewpage.action?pageId=3604491)

# [Inner API](https://wpbakery.atlassian.net/wiki/display/VC/Inner+API)

## Methods available

* [vc\_add\_default\_templates()](https://wpbakery.atlassian.net/wiki/pages/viewpage.action?pageId=524300)
* [vc\_add\_param()](https://wpbakery.atlassian.net/wiki/pages/viewpage.action?pageId=524335)
* [vc\_add\_params()](https://wpbakery.atlassian.net/wiki/pages/viewpage.action?pageId=524370)
* [vc\_disable\_frontend()](https://wpbakery.atlassian.net/wiki/pages/viewpage.action?pageId=524293)
* [vc\_enabled\_frontend()](https://wpbakery.atlassian.net/wiki/pages/viewpage.action?pageId=524361)
* [vc\_is\_as\_theme()](https://wpbakery.atlassian.net/wiki/pages/viewpage.action?pageId=524371)
* [vc\_is\_updater\_disabled()](https://wpbakery.atlassian.net/wiki/pages/viewpage.action?pageId=524359)
* [vc\_map\_update()](https://wpbakery.atlassian.net/wiki/pages/viewpage.action?pageId=524310)
* [vc\_map()](https://wpbakery.atlassian.net/wiki/pages/viewpage.action?pageId=524332)
* [vc\_remove\_element()](https://wpbakery.atlassian.net/wiki/pages/viewpage.action?pageId=524329)
* [vc\_remove\_param()](https://wpbakery.atlassian.net/wiki/pages/viewpage.action?pageId=524316)
* [vc\_set\_as\_theme()](https://wpbakery.atlassian.net/wiki/pages/viewpage.action?pageId=524297)
* [vc\_set\_shortcodes\_templates\_dir()](https://wpbakery.atlassian.net/wiki/pages/viewpage.action?pageId=524294)
* [vc\_shortcodes\_theme\_templates\_dir()](https://wpbakery.atlassian.net/wiki/pages/viewpage.action?pageId=524360)

At first you have to need registered "vcBeforeInit" hook and write all inner API methods in this hook method . Otherwise inner API method do not work properly.

N.B. If any inner API do not work, Must have to check " vcBeforeInit" hooked is registered to that modules.

# [vc\_add\_default\_templates()](https://wpbakery.atlassian.net/wiki/pages/viewpage.action?pageId=524300)

To add more predefined templates to the Visual Composer interface call vc\_add\_default\_templates() function in your jscomposer.php file. Predefined templates are available since Visual Composer 4.3 version. **Note:** vc\_add\_default\_templates() should be hooked in vc\_load\_default\_templates\_action action.

|  |
| --- |
| <?php vc\_add\_default\_templates( $data); ?> |

## Params

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | | Param Name | Type | Description | | $data | Array | Information about your new predefined template | |

### Modify default templates

To modify default templates list you can use vc\_load\_default\_templates hook and filter/modify array which stores default templates in it.

|  |
| --- |
|  |

# [vc\_add\_param()](https://wpbakery.atlassian.net/wiki/pages/viewpage.action?pageId=524335)

|  |
| --- |
| <?php vc\_add\_param( $shortcode, $attributes); ?> |

|  |  |  |
| --- | --- | --- |
| Param Name | Type | Description |
| $shortcode | Array | Shortcode base. For [my\_shortcode] shortcode base is my\_shortcode. For default "Message box" element, base is "vc\_message" |
| $attributes | Array | Array should be defined according to [vc\_map() available options](https://wpbakery.atlassian.net/wiki/pages/viewpage.action?pageId=524332#vc_map%28%29-ParametersofparamsArray). |
|  |  |  |

Code explanation:

1. First we create array which "describes" new attribute that should be added to "Message box" element. **Note:** during the initial vc\_map() function call "vc\_message" was used as a base attribute, base is required to properly "target" function call
2. Then we call vc\_add\_param() function with new attribute and base

# [vc\_add\_params()](https://wpbakery.atlassian.net/wiki/pages/viewpage.action?pageId=524370)

This helper function is similar  to [vc\_add\_param()](https://wpbakery.atlassian.net/wiki/pages/viewpage.action?pageId=524335), but it can accept multiple params that should be added to your content element. Basically this function will use a foreach loop for all the attributes and then [vc\_add\_param()](https://wpbakery.atlassian.net/wiki/pages/viewpage.action?pageId=524335) will be called for each single param. List of [available parameters](https://wpbakery.atlassian.net/wiki/pages/viewpage.action?pageId=524332#vc_map%28%29-Availabletypevalues).

|  |
| --- |
| <?php vc\_add\_params( $shortcode, $attributes); ?> |

|  |  |  |
| --- | --- | --- |
| Param Name | Type | Description |
| $shortcode | String | Shortcode base. For [my\_shortcode] shortcode base is my\_shortcode. For default "Message box" element, base is "vc\_message" |
| $attributes | Array | Multidimensional array should be defined according to [vc\_map() available options](https://wpbakery.atlassian.net/wiki/pages/viewpage.action?pageId=524332#vc_map%28%29-ParametersofparamsArray). |
|  |  |  |

# [vc\_disable\_frontend()](https://wpbakery.atlassian.net/wiki/pages/viewpage.action?pageId=524293)

To disable frontend editor on your site call vc\_disable\_frontend() function from your functions.php file. **Note:** This functions is available since Visual Composer 4.0 version.

|  |
| --- |
| <?php vc\_disable\_frontend(); ?> |

## Params

None.

## Example

How to disable frontend editing mode in Visual Composer.

|  |
| --- |
| <?php  vc\_disable\_frontend();  ?> |
|  |

# [vc\_enabled\_frontend()](https://wpbakery.atlassian.net/wiki/pages/viewpage.action?pageId=524361)

Returns (bool) true if front end editor is enabled or (bool) false if theme author previously used [vc\_disable\_frontend()](https://wpbakery.atlassian.net/wiki/pages/viewpage.action?pageId=524293) to disabled front end editor on the site.

|  |
| --- |
| <?php vc\_enabled\_frontend(); ?> |

## Params

None.

## Example

Check if frontend editing mode is enabled on your site.

|  |
| --- |
| <?php  if( vc\_enabled\_frontend() ) {      // Front end editor mode is enabled. Do something.  }  ?> |

# [vc\_is\_as\_theme()](https://wpbakery.atlassian.net/wiki/pages/viewpage.action?pageId=524371)

Helper function that will return (bool) true if Visual Composer is integrated in to the theme. This is true if theme author called [vc\_set\_as\_theme()](https://wpbakery.atlassian.net/wiki/pages/viewpage.action?pageId=524297) from theme's functions.php file.

|  |
| --- |
| <?php vc\_is\_as\_theme(); ?> |

## Params

None.

## Example

Check if Visual Composer is running in a "theme mode".

|  |
| --- |
| <?php  if( vc\_is\_as\_theme() ) {      //Do something  }  ?> |

# [vc\_map\_update()](https://wpbakery.atlassian.net/wiki/pages/viewpage.action?pageId=524310)

After vc\_map() function call all mapping data is stored in memory, if you need to modify one of the existing parameters you can use vc\_map\_update() function to do that. This is useful if you want to make a simple change to the existing content elements. For example: Change content element name.

|  |
| --- |
| <?php vc\_map\_update( $shortcode, $settings); ?> |

## Params

|  |  |  |
| --- | --- | --- |
| Param Name | Type | Description |
| $shortcode | String | Shortcode base. For [my\_shortcode] shortcode base is my\_shortcode. For default "Message box" element, base is "vc\_message" |
| $settings | Array | Array with list of settings that should be updated for desired content element. [List of settings](https://wpbakery.atlassian.net/wiki/pages/viewpage.action?pageId=524332#vc_map%28%29-Parameters) that can be changed |

## Example

How to change default name and category values for "Message box" content element.

|  |
| --- |
| <?php  $settings= array(    'name'=> 'new name',    'category'=> 'New category name'  );  vc\_map\_update( 'vc\_message', $settings); // Note: 'vc\_message' was used as a base for "Message box" element  ?> |

# [vc\_map\_update()](https://wpbakery.atlassian.net/wiki/pages/viewpage.action?pageId=524310)

After vc\_map() function call all mapping data is stored in memory, if you need to modify one of the existing parameters you can use vc\_map\_update() function to do that. This is useful if you want to make a simple change to the existing content elements. For example: Change content element name.

|  |
| --- |
| <?php vc\_map\_update( $shortcode, $settings); ?> |

## Params

|  |  |  |
| --- | --- | --- |
| Param Name | Type | Description |
| $shortcode | String | Array with list of settings that should be updated for desired content element. List of settings that can be changed |
| $settings | Array | Array with list of settings that should be updated for desired content element. List of settings that can be changed |
|  |  |  |

## Example

How to change default name and category values for "Message box" content element.

|  |
| --- |
| <?php  $settings= array(    'name'=> 'new name',    'category'=> 'New category name'  );  vc\_map\_update( 'vc\_message', $settings);  // Note: 'vc\_message' was used as a base for "Message box" element  ?> |

# [vc\_map()](https://wpbakery.atlassian.net/wiki/pages/viewpage.action?pageId=524332)

To add your shortcode to the Visual Composer content elements list, vc\_map() function should be called with an array of special attributes describing your shortcode.

|  |
| --- |
| <?php vc\_map( $params); ?> |

## Contents

* [Contents](https://wpbakery.atlassian.net/wiki/pages/viewpage.action?pageId=524332#vc_map%28%29-Contents)
* [Parameters](https://wpbakery.atlassian.net/wiki/pages/viewpage.action?pageId=524332#vc_map%28%29-Parameters)
* [params Array](https://wpbakery.atlassian.net/wiki/pages/viewpage.action?pageId=524332#vc_map%28%29-paramsArray)
  + [Parameters of params Array](https://wpbakery.atlassian.net/wiki/pages/viewpage.action?pageId=524332#vc_map%28%29-ParametersofparamsArray)
    - [Available type values](https://wpbakery.atlassian.net/wiki/pages/viewpage.action?pageId=524332#vc_map%28%29-Availabletypevalues)
* [Param Dependencies](https://wpbakery.atlassian.net/wiki/pages/viewpage.action?pageId=524332#vc_map%28%29-param-dependenciesParamDependencies)
* [Nested Content Elements](https://wpbakery.atlassian.net/wiki/pages/viewpage.action?pageId=524332#vc_map%28%29-NestedContentElements)
* [Example](https://wpbakery.atlassian.net/wiki/pages/viewpage.action?pageId=524332#vc_map%28%29-Example)
  + [Add existing shortcode](https://wpbakery.atlassian.net/wiki/pages/viewpage.action?pageId=524332#vc_map%28%29-Addexistingshortcode)

## Parameters

**$params** Associative array which holds instructions for Visual Composer and is used in "mapping" process.

|  |  |  |
| --- | --- | --- |
| Param Name | Type | Desceiption |
| weight | Integer | Content elements with greater weight will be rendered first in "Content Elements" grid (Available from 3.7 version) |
| show\_settings\_on\_create | Boolean | Set it to false if content element's settings page shouldn't open automatically after adding it to the stage |
| params | Array | List of shortcode attributes. Array which holds your shortcode params, these params will be editable in shortcode settings page |
| name | String | Name of your shortcode for human reading inside element list |
| js\_view | String | Set custom backbone.js view controller for this content element |
| icon | String | URL or CSS class with icon image. |
| html\_template | String | Path to shortcode template. This is useful if you want to reassign path of existing content elements through your module. Another way to change html markup |
| group | |  |  | | --- | --- | |  | String | | |  |  | | --- | --- | |  | Group your params in groups, they will be divided in tabs in the edit element window *(Available from Visual Composer 4.1)* | |
| front\_enqueue\_js | String | Absolute url to javascript file (useful for storing your custom backbone.js views), this js will be loaded in the js\_composer frontend edit mode (it allows you to add more functionality to your shortcode in js\_composer frontend edit mode). (Available from Visual Composer 4.2.2) |
| front\_enqueue\_css | String | Absolute url to css file if you need to load custom css file in the frontend editing mode. (Available from Visual Composer 4.2.2) |
| description | String | Short description of your element, it will be visible in "Add element" window |
| custom\_markup | String | Custom html markup for representing shortcode in visual composer editor |
| class | String | CSS class which will be added to the shortcode's content element in the page edit screen in Visual Composer backend edit mode |
| category | String | Category which best suites to describe functionality of this shortcode. Default categories: Content, Social, Structure. You can add your own category, simply enter new category title here |
| base | String | Shortcode tag. For [my\_shortcode] shortcode base is my\_shortcode |
| admin\_enqueue\_js | String | Absolute url to javascript file, this js will be loaded in the js\_composer edit mode (it allows you to add more functionality to your shortcode in js\_composer edit mode) |
| admin\_enqueue\_css | String | Absolute url to css file if you need to add custom css for element block in js\_composer constructor mode |

## params Array

Here you should describe all you shortcode's attributes that should be editable with Visual Composer interface. Each shortcode's attribute should be described in separate array element.

Defining "Text" attribute:

|  |
| --- |
| array(    "type"=> "textfield",    "holder"=> "div",    "class"=> "",    "heading"=> "Text",    "param\_name"=> "foo",    "value"=> "This is test param for creating new project",    "description"=> "Enter foo.",  ) |

### Parameters of params Array

|  |  |  |
| --- | --- | --- |
| Param Name | Type | Desceiption |
| type | String | |  |  | | --- | --- | |  | Attribute type. In the "Available type values" table you can see all available variations | |
| holder | String | |  |  | | --- | --- | |  | HTML tag name where Visual Composer will store attribute value in Visual Composer edit mode. Default: hidden input | |
| params | Array | List of shortcode attributes. Array which holds your shortcode params, these params will be editable in shortcode settings page |
| class | String | Class name that will be added to the "holder" HTML tag. Useful if you want to target some CSS rules to specific items in the backend edit interface |
| js\_view | String | Set custom backbone.js view controller for this content element |
| heading | String | Human friendly title of your param. Will be visible in shortcode's edit screen |
| param\_name | String | Must be the same as your parameter name |
| group | |  |  | | --- | --- | |  | String | | |  |  | | --- | --- | |  | Group your params in groups, they will be divided in tabs in the edit element window *(Available from Visual Composer 4.1)* | |
| value | Array or String | Default attribute's value |
| description | String | |  |  | | --- | --- | |  | Human friendly description of your param. Will be visible in shortcode's edit screen | |
| admin\_label | Boolean | Show value of param in Visual Composer editor |
| dependency | Array | Define param visibility depending on other field value. Please read Param Dependencies |
| edit\_field\_class | String | Set param container width in content element edit window. According to Bootstrap logic eg. col-md-4. (Available from Visual Composer 4.0) |
| weight | Integer | Params with greater weight will be rendered first. (Available from Visual Composer 4.4) |
| group | String | Use it to divide your params within groups (tabs) |
| admin\_enqueue\_js | String | Absolute url to javascript file, this js will be loaded in the js\_composer edit mode (it allows you to add more functionality to your shortcode in js\_composer edit mode) |

#### Available type values

There are multiple attribute types available, this field is used to "tell" Visual Composer how to handle your attribute in content element edit screen.

|  |
| --- |

|  |  |
| --- | --- |
| Value | Desceiption |
| textarea\_html | Text area with default Prestashop Editor. **Important:** only one html textarea is permitted per shortcode and it should have "content" as a param\_name |
| textfield | Simple input field |
| textarea | Simple textarea field |
| dropdown | Dropdown input field with set of available options. Array containing the drop down values (either should be a normal array, or an associative array) |
| attach\_image | Single image selection |
| attach\_images | Multiple images selection |
| posttypes | Checkboxes with available post types |
| colorpicker | |  |  | | --- | --- | |  | Color picker | |
| exploded\_textarea | Text area, where each line will be imploded with comma (,) |
| widgetised\_sidebars | Dropdown element with set of available widget regions, that are registered in the active wp theme |
| textarea\_raw\_html | Text area, its content will be coded into base64 (this allows you to store raw js or raw html code) |
| vc\_link | Link selection. Then in shortcodes html output, use $href = vc\_build\_link( $href ); to parse link attribute |
| checkbox | Creates checkboxes, can have 1 or multiple checkboxes within one attribute |
| loop | Loop builder. Lets your users to construct loop which can later be used during the shortcode's output |
| base | Shortcode tag. For [my\_shortcode] shortcode base is my\_shortcode |
| css | Basic CSS style editor for your content element. Check "Add "Design Options" Tab with CSS Editor to Your Element" page for more details) |
|  | attribute types can be extended with new custom types. Tutorial: Create new param type |

## Param Dependencies

It is possible to link param with another param by defining dependency param. Dependency param is array with defined param name and values which will allow to display this linked param. Also it is possible to show param field if linked field is not empty and define callback function which will be called when value of linked element is changed. Next table contains field list and explanation of dependency param array keys.

|  |  |  |
| --- | --- | --- |
| Key Name | Type | Desceiption |
| element | String | Param name (linked field) which will be observed for changes. Must be the same as param\_name param for shortcode attribute |
| value | Array | |  |  | | --- | --- | |  | List of linked element's values which will allow to display param | |
| not\_empty | Boolean | Display field if value of linked field is not empty |
| callback | String | |  |  | | --- | --- | |  | javascript function name. Function will be called when value of linked field is changed | |
| js\_view | String | Set custom backbone.js view controller for this content element |
| icon | String | URL or CSS class with icon image. More info [here](https://wpbakery.atlassian.net/wiki/pages/viewpage.action?pageId=524332) |
| html\_template | String | Path to shortcode template. This is useful if you want to reassign path of existing content elements through your module. Another way to [change html markup](https://wpbakery.atlassian.net/wiki/pages/viewpage.action?pageId=524332) |
| admin\_enqueue\_css | String | Absolute url to css file if you need to add custom css for element block in js\_composer constructor mode |

## Nested Content Elements

If you are looking for a way to create content elements with "nested" structure, [have a look at this example](https://wpbakery.atlassian.net/wiki/pages/viewpage.action?pageId=524362).

## Example

In this example you will be guided throw the whole process of adding your own shortcode to the Visual Composer editor. We also prepared a nice and clean "starter" for you, feel free to use it as a starting point for your own Visual Composer shortcode.

## Nested Content Elements

If you are looking for a way to create content elements with "nested" structure, [have a look at this example](https://wpbakery.atlassian.net/wiki/pages/viewpage.action?pageId=524362).

# [vc\_remove\_element()](https://wpbakery.atlassian.net/wiki/pages/viewpage.action?pageId=524329)

To remove content element from Visual Composer editor you should call vc\_remove\_element() function from your theme's jscomposer.php file with shortcode's base passed as parameter.

|  |
| --- |
| <?php jscomposer::vc\_remove\_element( $shortcode); ?> |

## Params

|  |  |  |
| --- | --- | --- |
| Param Name | Type | Description |
| $shortcode | String | Shortcode base. For [my\_shortcode] shortcode base is my\_shortcode. For default "Message box" element, base is "vc\_message" |

## Example

To remove default "Message box" element from Visual Composer use this code:

|  |
| --- |
| <?php  vc\_remove\_element( "vc\_message"); //Note: "vc\_message" was used in the vc\_map() function call as a base parameter for "Message box" element  ?> |

# [vc\_remove\_param()](https://wpbakery.atlassian.net/wiki/pages/viewpage.action?pageId=524316)

To remove parameter from existing content element you can use vc\_remove\_param() function.

|  |
| --- |
| <?php vc\_remove\_param( $shortcode, $attribute\_name); ?> |

## Params

|  |  |  |
| --- | --- | --- |
| Param Name | Type | Description |
| $shortcode | String | Shortcode base. For [my\_shortcode] shortcode base is my\_shortcode. For default "Message box" element, base is "vc\_message" |
| $attribute\_name | String | Parameter name (param\_name) from "params" array in your [vc\_map function call](https://wpbakery.atlassian.net/wiki/pages/viewpage.action?pageId=524332#vc_map%28%29-ParametersofparamsArray) |

# [vc\_set\_as\_theme()](https://wpbakery.atlassian.net/wiki/pages/viewpage.action?pageId=524297)

You as a theme author may want to hide certain Visual Composer settings tabs, for example: "Design options" tab under Settings->Visual Composer page and "Hola! Please activate your copy of Visual Composer to receive automatic updates.". To do that, you should call vc\_set\_as\_theme() function from your functions.php file.

|  |
| --- |
| <?php vc\_set\_as\_theme( $disable\_updater= false ); ?> |

## Params

|  |  |  |
| --- | --- | --- |
| Param Name | Type | Description |
| $disable\_updater | Boolean | You can pass optional argument to disable automatic updates notifications and hide "Product License" tab in Settings->Visual Composer |

# [vc\_set\_default\_editor\_post\_types()](https://wpbakery.atlassian.net/wiki/pages/viewpage.action?pageId=524296)

To enable Visual Composer page builder for theme defined post types by default, you can make use of vc\_set\_default\_editor\_post\_types() api function.

|  |
| --- |
| <?php vc\_set\_default\_editor\_post\_types( $list); ?> |

Params

|  |  |  |
| --- | --- | --- |
| Param Name | Type | Description |
| $list | Array | List of post types for which Visual Composer should be available by default |

# [Developers "How To's"](https://wpbakery.atlassian.net/wiki/pages/viewpage.action?pageId=524341)

Periodically we update our "How To" section with new materials.

* [Add "Design Options" Tab with CSS Editor to Your Element](https://wpbakery.atlassian.net/wiki/pages/viewpage.action?pageId=2654215)
* [Add Custom Row Layouts (Presets)](https://wpbakery.atlassian.net/wiki/pages/viewpage.action?pageId=524342)
* [Change CSS Class Names in Output](https://wpbakery.atlassian.net/wiki/display/VC/Change+CSS+Class+Names+in+Output)
* [Change Shortcode's HTML Output](https://wpbakery.atlassian.net/wiki/display/VC/Change+Shortcode%27s+HTML+Output)
* [Create New Param Type](https://wpbakery.atlassian.net/wiki/display/VC/Create+New+Param+Type)
* [Nested Shortcodes (container)](https://wpbakery.atlassian.net/wiki/pages/viewpage.action?pageId=524362)
* [Param position control with vc\_add\_param](https://wpbakery.atlassian.net/wiki/display/VC/Param+position+control+with+vc_add_param)
* [Remove "Edit with Visual Composer" from Prestashop Admin Bar](https://wpbakery.atlassian.net/wiki/pages/viewpage.action?pageId=2031624)
* [Remove Content Element from Grid Builder](https://wpbakery.atlassian.net/wiki/display/VC/Remove+Content+Element+from+Grid+Builder)
* [Set Content Element Icon](https://wpbakery.atlassian.net/wiki/display/VC/Set+Content+Element+Icon)
* [Update Single Param Values](https://wpbakery.atlassian.net/wiki/display/VC/Update+Single+Param+Values)

# [Become a VC Tester](https://wpbakery.atlassian.net/wiki/display/VC/Become+a+VC+Tester)

Smartdatasoft is looking for a customers who are willing to receive latest updates of Visual Composer before official release for testing in order to help your beloved Module to be shipped in a top quality.

## Contents

* [Contents](https://wpbakery.atlassian.net/wiki/display/VC/Become+a+VC+Tester#BecomeaVCTester-Contents)
* [What is Happening?](https://wpbakery.atlassian.net/wiki/display/VC/Become+a+VC+Tester#BecomeaVCTester-WhatisHappening?)
* [Our Requirements](https://wpbakery.atlassian.net/wiki/display/VC/Become+a+VC+Tester#BecomeaVCTester-OurRequirements)
* [What will You Receive?](https://wpbakery.atlassian.net/wiki/display/VC/Become+a+VC+Tester#BecomeaVCTester-WhatwillYouReceive?)
* [How to Apply?](https://wpbakery.atlassian.net/wiki/display/VC/Become+a+VC+Tester#BecomeaVCTester-HowtoApply?)

## What is Happening?

Starting from Visual Composer version 4.5 we are launching "Visual Composer Tester Program" based on our customers in order to rise quality of Visual Composer Module. This is introduced as an additional mechanism for ensuring we are releasing bug free top quality product that fits with our customer real life environment. During program we will involve 4-5 candidates in our testing process and will collaborate with them on regular basis to come up with the best results possible. There are several requirements listed below which must be met in order to apply as a candidate and of course there are also bonuses for those who participate in our program.

## Our Requirements

There are following mandatory requirements you must fit in order to apply:

* You must be a Visual Composer user and direct client of Smartdatasoft (you must have direct license for Visual Composer);
* You must have a valid website with content where you have Visual Composer activated;
* You must have latest Prestashop installation on your site;
* You will need to provide Smartdatasoft with site credentials in order to access your site;
* You must have at least 1 add-on for Visual Composer installed on your site.

In additional there are following optional requirements - in order to apply for tester you must fit with at least 3 out of 7 optional requirements:

* You have a Prestashop theme which was delivered with Visual Composer bundled(now in-built version is disabled or auto-update is set for this version);
* You have Visual Composer Custom CSS used on your Prestashop site;
* You have extended capabilities of Visual Composer by using Visual Composer API;
* You have ShortCodes mapped with ShortCode Mapper;
* You have additional custom predefined templates/layouts added in addition to built-in predefined templates/layouts;
* You have at least one of following modules installed: Revolution Slider, SmartBlog, SmartShortCode.

## What will You Receive?

There are following bonuses that you will receive if applying to our tester program:

* You will receive latest Visual Composer versions even before they will be available on CodeCanyon for public download;
* Your credentials will be mentioned in our release credits.txt (unless you state other);
* You will participate in development process of the most popular page builder module for Prestashop.

## How to Apply?

In order to apply for Visual Composer testing please send inquiry to e-mail address [support@Smartdatasoft.com](mailto:raitis@wpbakery.com) with reference "Visual Composer Tester". In the e-mail you must provide:

* Your name, surname;
* Envato username;
* Website address;
* Site credentials (which Smartdatasoft will use to access your site);
* List of Visual Composer add-ons used on your site;
* List of Optional requirements that applies to you.

Thank you for your interest,

Smartdatasoft Team

# [Outro](https://wpbakery.atlassian.net/wiki/display/VC/Outro)

Thank you for your purchase. If you have any questions that are beyond the scope of this help file, please use our [Support system](http://support.wpbakery.com/) to open support ticket.

Arifur Rahman, Smartdatsoft

Don't be shy - [join us on Facebook](http://www.facebook.com/wpbakery). [Smartdatasoft website](http://wpbakery.com/) ·

*Content From:*<https://wpbakery.atlassian.net/wiki/display/VC/Visual+Composer+Pagebuilder+for+WordPress>