

Blocs Plus Wordpress

Introduction

One of the main additional functions of Blocs Plus is the ability to create Wordpress Themes.

It's important to note, building a Wordpress theme is **not** the same as building a static website, the **most important** difference when creating a Wordpress theme is the fact all of the content is **not** stored within the theme, but is instead, populated from a database.

You can still create static pages as part of your Wordpress theme, but creating pages that will enable clients to edit via the Wordpress backend will require a slightly different way of thinking.

Video Tutorial Series

We have created a free 40 minute video course, the course covers the process of building a Wordpress theme from scratch.

1. [Introduction](#)
2. [Navigation and Hero](#)
3. [Post Loops](#)
4. [Widget Integration](#)
5. [Post Page Template](#)
6. [Page template](#)
7. [Archive Page Template](#)
8. [Author Page Template](#)
9. [Search Integration](#)

Project settings

Blocs has additional options in the main project settings that let you control various aspects of how your theme will work.

Include Wordpress admin bar - This option lets you decide if the Wordpress admin bar is displayed on the front end when a user is logged into Wordpress.

Add support for featured images on pages - this enables the featured image controls for pages in the Wordpress back end.

Logo customiser Controls - When enabled this will allow users to change the logo image found in a navigation Bric, to a custom one, using the Wordpress customiser options found in the Wordpress admin.

Automatic page generation - When enabled this will automatically generate all of the pages in your Blocs site in the Wordpress back end when the theme is first activated. (Please note to re-run this function the theme will need to be disabled and then re-activated).

Theme Preview Thumb - This is the thumbnail used in the Wordpress admin theme selection area, the recommended size is **1200 x 900**. If no thumb image is set, Blocs will automatically generated one during export, based on the home page of your Blocs project.

Site Details

In the Blocs project settings there is a new set of controls that let you add site details, such as version and author. These details are used to populate the theme meta in the style.css file. The theme name meta is taken from the site name field in the project settings (general settings).

Pages

Page names are incredibly important when creating a Wordpress theme. By default Wordpress has specific protected page names set up for key pages in a theme.

[index](#) - The themes home page template - (automatically assigned by Blocs).

[category](#) - The themes category template.

[author](#) - The author theme template.

[single](#) - The blog post template.

[search](#) - The search results template.

[page](#) - The default page template.

Of the above pages, **Single** and **Page** are the most important requirements when creating a Wordpress theme as they will be used in conjunction with the Wordpress blogging and page management backend controls. When you want to create one of these pages you will need to name it accordingly, for example the default blog page template name for a Wordpress theme is `single.php`, so to create this template give it the name **single** via the Blocs page settings.

When a project is set to use Wordpress as the CMS (via project settings), a dropdown button containing all relevant Wordpress template name references will be displayed to the left of the name field on the new page sheet. Selecting one of the options from this dropdown will populate the name field with the relevant Wordpress template name.

Additional page templates

When you export a Wordpress theme, unless the page name is one of the default protected Wordpress theme page names, the page will be generated as an additional page template with the **page_** prefix added to the file name.

Global Areas (header.php | footer.php)

When a project is exported as a Wordpress theme, the top global area becomes the **header.php** and the bottom becomes the **footer.php**.

Navigation

Blocs Plus contains a new menu data feed option, that makes it possible to populate a menu with the data from the Wordpress primary menu. This should be used on the main navigation of a theme.

Columns (Page | Post | Content Loops)

In Blocs Plus the standard column elements will display additional Wordpress loop options that enable them to be setup as a post or page loop item. When a column is set up as a loop item, once hosted, the Wordpress core will populate the columns parent row with multiple items with data from the main Wordpress database.

When setting up a Wordpress loop, take care to ensure the row only has a single column in it with either a post or page loop type assigned.

Within this column you can add elements such as images, text and buttons and assign Wordpress data feeds to them. When a theme is run on a Wordpress install this row will be populated with the relevant number of items found in the Wordpress database.

Post Item - This item will become the template for the post loop item, any elements inside this column with Wordpress post data feeds will be populated accordingly.

Page Item - This item will become the template for the page loop item, any elements inside this column with Wordpress page data feeds will be populated accordingly.

Content Item - This item will become the template for the content loop item (content loop items are typically found on the archive, author, category, tag, and search results page templates), any elements inside this column with Wordpress post data feeds will be populated accordingly.

All - This item will become the template for the standard post, page and all custom post type loop items, any elements inside this column with Wordpress post data feeds will be populated accordingly. This will output all post types and the standard page type that the theme supports.

All Post Types - This item will become the template for the standard post and all custom post type loop items, any elements inside this column with Wordpress post data feeds will be populated accordingly. This will output all post types that the theme supports.

Call Custom Post Types - This item will become the template for all custom post type loop item, any elements inside this column with Wordpress post data feeds will be populated accordingly. This will output all custom post types that the theme supports.

Text Brics

In Blocs Plus the standard core library text Brics will display addition Wordpress specific data feed options.

Title - This will populate the text Bric with the current (post | page) Title.

Content - This will populate the text Bric with the current (post | page) Content.

Publish Date - This will populate the text Bric with the current (post | page) Publish Date.

Edit Date - This will populate the text Bric with the current (post | page) Edit Date.

Author - This will populate the text Bric with the current (post | page) Author.

Excerpt - This will populate the text Bric with the current (post | page) excerpt.

Site Title - This will populate the text Bric with **site title** that is defined in the main Wordpress general settings.

Site Tagline - This will populate the text Bric with **site tagline** that is defined in the main Wordpress general settings.

Images Brics

In Blocs Plus the standard core library image Bric will display addition Wordpress specific data feed options.

Featured Image - This will populate the image with the current (post | page) feature image.

Featured Image (Small) - This will populate the image with the current (post | page) small feature image.

Featured Image (Medium) - This will populate the image with the current (post | page) medium feature image.

Featured Image (Large) - This will populate the image with the current (post | page) large feature image.

Featured Image (Full) - This will populate the image with the current (post | page) full feature image.

Div Container Brics

In Blocs Plus the standard core library div container Brics will display addition Wordpress specific type options.

User is logged in - Any content placed within a div of this type will only be shown when a user is logged in.

User can edit posts - Any content placed within a div of this type will only be shown when a user is logged in and can edit posts.

User can edit pages - Any content placed within a div of this type will only be shown when a user is logged in and can edit pages.

User is admin - Any content placed within a div of this type will only be shown when a user is logged in and is an admin.

User is editor - Any content placed within a div of this type will only be shown when a user is logged in and is an editor.

User is author - Any content placed within a div of this type will only be shown when a user is logged in and is an author.

User is contributor - Any content placed within a div of this type will only be shown when a user is logged in and is a contributor.

User is subscriber - Any content placed within a div of this type will only be shown when a user is logged in and is a subscriber.

User is logged out - Any content placed within a div of this type will only be shown when a user is logged out.

Wordpress Custom Brics

Blocs Plus will install a number of additional custom Brics to use as part of your Wordpress theme.

Widgetized Zone Bric

The widget zone adds a widgetized content area to a page that will allow users to add widgets to via the Wordpress backend once hosted. You can have any number of these zones on your site, they are typically placed in the sidebar or footer of a theme. They allow users to add additional content to a page via the Wordpress back end.

Paginate Bric

The post paginate navigation Bric adds pagination to a page allowing users to quickly browse Wordpress posts in published order. It will automatically populate with data assigned from the core Wordpress database when hosted.

WP Content Bric

The post/page content area will populate with the post/pages content data base feed.

WP Comments Bric

The comments Bric adds a comments feed, can only be used on a page or post page.

Preview Mode

The standard in app preview for Blocs will output the visual layout of your theme and will also populate it with live content if a URL for a site **currently running Wordpress** is provided in the site address in the project settings.

Live Content Previewing

Blocs now has the ability to connect with any site that is currently running Wordpress and can bring various parts of the sites content down into Blocs and use it in the in-app preview and preview in browser modes. It also works on local host setups run with MAMP.

This is really useful if you are rebuilding a site currently run on Wordpress as it means you can design with the sites current live content.

For superior integration, Blocs also has a free Wordpress developer plugin. Installing this into the Wordpress site during the development stage allows even more data such as widgets, custom posts and custom fields to be rendered within the Blocs preview.

When a project has its CMS set to Wordpress and has a valid Wordpress site URL set in project settings, a Wordpress icon will appear in the toolbar, this button gives you an overview of the content available for simulation pulled from the live site.

I've included a mock example that pulls data from the Blocs store. This doesn't have the developer plugin enabled so only pulls some content down, but you can see how to all works.

Wordpress Developer Plugin

This plugin expands the Wordpress JSON API and adds additional endpoints for greater integration with the Blocs design environment. This plugin is optional and only required during the development phase and is not required once a theme is exported from Blocs and installed into Wordpress.

Essentially the plugin sends more detailed simulation data back to Blocs in order to deliver an in app preview, that is closer to the real thing as it uses actual data from the live Wordpress site.

Custom Posts and Fields (Wordpress Custom Data Manager)

Blocs comes with a new window designed to handle the management of custom post and custom field data types. To access the Custom Data Manager navigate to Window > Wordpress Custom Data Manager via the Main Window Menu.

From here you can register new custom posts and custom field types and also apply their core settings.

Custom Posts

When a new custom post type is added to a project, it will become available in the Wordpress loop options found in the column Bric sidebar options.

A custom post type will also require a dedicated page template that has the same name as your post type. It's not necessary to apply the **single-** prefix as Blocs will manage this automatically upon export.

Also take care to re-save your permalink settings within the Wordpress admin, when adding a new custom post page template to your site.

Custom Fields

When a new custom field type is added to a project, it will show up in the text and image data feed dropdown located in the sidebar. It's also possible to display the output of a custom field type, using the custom field short code (see short code section). Short codes are handy for using in areas such as page settings.

Link Interactions

Blocs Plus has a new set of Wordpress interactions which are accessed via the standard interactions section in the sidebar. These controls make the process of hooking up common Wordpress related URL links simple and easy to understand.

These are the current Wordpress link types that are supported:

Permalink - This will be populated with the current page or posts full page URL.

Category URL - This will be populated with a link to the category archive page for the current posts first category.

Tag URL - This will be populated with a link to the tag archive page for the current posts first tag.

Author Archive URL - This will be populated with a link to the post authors archive page.

Edit Post URL - This will be populated with a link to the admin edit interface for the current post.

Login URL - This will be populated with a link to the Wordpress login page.

Custom Classes

Blocs Plus has an additional subclass section in the Class Manager that is dedicated to Wordpress classes. These additional subclasses allow you to style parts off a Wordpress themes user interface, that it is not possible to select within the Blocs design environment.

Posts

No post result title - This is the title shown when no posts are found within a loop.

Comments Bric

Main container - The main parent container for all comments.

Comment container - The container for an individual comment.

Title - The comment title text.

Required * - The required field text.

Author - The author label text.

Avatar - The comment avatar image.

Content container - The comments content container.

Reply Link - The comment reply link.

No comments title - The title when no comments are present.

Paginate Bric

Container - The main parent container for the paginate links.

Link - The individual paginate link.

Current Link - The selected (current) paginate link.

WP Custom Functions - Code Editor

The Blocs Code Editor has an additional input zone (Extra Functions) that lets users add additional php function code to Wordpress themes created with Blocs. Upon export, this additional code is written into a php file (**extra-functions.php**).

Please note, The **Extra Functions** menu item will only appear in the Code Editor dropdown menu, when the project CMS is set to Wordpress.

WP Custom Data Attributes

In Blocs Plus there are some custom data attributes that can be added to elements to apply key Wordpress post and page data to the themes user interface.

blocs-wp-post-thumb="true" - Adding this custom data attribute will assign the current page or post feature image to the element as an inline background-image style. This is most useful for setting feature images as Bloc backgrounds.

WP Short Codes

Blocs Plus adds support for the following short codes. These can be used within text Brics content, custom data attributes, custom Brics and URL interactions.

Site Name

%WP_name% - This will be populated with the blog name that is assigned via the Wordpress admin (`<?php bloginfo('name'); ?>`). Known in the Wordpress backend (General Settings) as the Site Title.

Site Description

%WP_description% - This will be populated with the blog description that is assigned via the Wordpress admin (`<?php bloginfo('description'); ?>`). Known in the Wordpress backend (General Settings) as the Tagline.

Login URL

%WP_login% - This will be populated with the Wordpress login URL (`<?php echo wp_login_url(); ?>`).

Page | Post Details

Page Title

%WP_title% - This will be populated with the current page/post title (`<?php the_title(); ?>`).

Page Content

%WP_content% - This will be populated with the current page/post content (`<?php the_content(); ?>`).

Excerpt

%WP_excerpt% - This will be populated with the current page/post excerpt (`<?php the_excerpt(); ?>`).

Category

%WP_category% - This will be populated with the first category name for the current post (`<?php echo get_the_category()[0]->name;?>`).

Category URL

%WP_categoryURL% - This will be populated with the first category URL for the current post (`<?php echo get_category_link(get_the_category()[0]);?>`).

Categories

%WP_categories% - This will be populated with a list of all category text links, separated by a comma, for the current post (<?php echo the_category(', ');?>).

Tag

%WP_tag% - This will be populated with the first tag name for the current post (<?php \$tags = wp_get_post_tags(\$post->ID); echo \$tags[0]->name;?>).

Tag URL

%WP_tagURL% - This will be populated with the first tag URL for the current post (<?php \$tags = wp_get_post_tags(\$post->ID); echo get_tag_link(\$tags[0]->term_id);?>).

Tags

%WP_tags% - This will be populated with a list of all tag text links, separated by a comma, for the current post (<?php echo get_post_tag_links(', ');?>).

Publish Data

%WP_publishDate% - This will be populated with the current page/post publish date (<?php echo get_the_date();?>).

Edit Date

%WP_editDate% - This will be populated with the current page/post edit date (<?php echo the_modified_date();?>).

Edit Post URL

%WP_editPostURL% - This will be populated with the current page/post edit admin URL (<?php echo get_edit_post_link(get_the_ID());?>).

Comment Count Number Only

%WP_commentCount% - This will be populated with the current post comment count (<?php echo get_comments_number();?>).

Comment Count With Label

%WP_commentCountFull% - This will be populated with the current post comment count followed by the word comment(s) (<?php echo get_comments_number();?>).

Post | Page Images

Thumbnail URL

%WP_thumbURL% - This will be populated with the current page/post thumbnail URL (<?php echo get_the_post_thumbnail_url();?>).

Small Thumbnail URL

%WP_thumbSmURL% - This will be populated with the current page/post small thumbnail URL (<?php echo get_the_post_thumbnail_url('thumbnail');?>).

Medium Thumbnail URL

%WP_thumbMedURL% - This will be populated with the current page/post medium thumbnail URL (<?php echo get_the_post_thumbnail_url('medium');?>).

Large Thumbnail URL

%WP_thumbLgURL% - This will be populated with the current page/post large thumbnail URL (<?php echo get_the_post_thumbnail_url('large');?>).

Full Thumbnail URL

%WP_thumbFullURL% - This will be populated with the current page/post full thumbnail URL (<?php echo get_the_post_thumbnail_url('full'); ?>).

Thumbnail Title

%WP_thumbTitle% - This will be populated with the current page/post thumbnail **title** content. Data is set in the Wordpress Media Library backend. (<?php echo get_post(get_post_thumbnail_id()->post_title; ?>).

Thumbnail Caption

%WP_thumbCaption% - This will be populated with the current page/post thumbnail **caption** content. Data is set in the Wordpress Media Library backend. (<?php echo get_the_post_thumbnail_caption(); ?>).

Thumbnail Description

%WP_thumbDesc% - This will be populated with the current page/post thumbnail description content. Data is set in the Wordpress Media Library backend. (<?php echo get_post(get_post_thumbnail_id()->post_content; ?>).

Archives

Archive Title

%WP_archiveTitle% - This will be populated with the current archive page title (<?php echo single_term_title();?>).

Archive Description

%WP_archiveDesc% - This will be populated with the current archive page description (<?php echo get_the_archive_description(); ?>).

Search

Current Search Query

%WP_searchQuery% - This will be populated with the current search term as text (<?php echo get_search_query(); ?>).

Query Count

%WP_postQueryCount% - This will be populated with the current number of items found for the current page content loop (<?php echo \$wp_query->found_posts; ?>).

Author Avatar

%WP_authorAvatarURL% - This will be populated with the current page/post author thumbnail (200px) URL (<?php echo get_avatar_url(get_the_author_meta('ID'),array('size' => 200));?>).

Author Avatar Medium

%WP_authorMdAvatarURL% - This will be populated with the current page/post author thumbnail (500px) URL (<?php echo get_avatar_url(get_the_author_meta('ID'),array('size' => 500));?>).

Author Avatar Large

%WP_authorLgAvatarURL% - This will be populated with the current page/post author thumbnail (1000px) URL (<?php echo get_avatar_url(get_the_author_meta('ID'),array('size' => 1000));?>).

Links

Perma Link

%WP_link% - This will be populated with the current page/post URL (<?php echo get_permalink(); ?>).

Previous Post Link

%WP_prev_post_link% - This will be populated with a text link to the previous post (<?php previous_post_link('%link', '%title');?>).

Next Post Link

%WP_next_post_link% - This will be populated with a text link to the next post (<?php next_post_link('%link', '%title');?>).

Previous Post URL

%WP_prev_postURL% - This will be populated with the URL of the previous post (<?php echo get_permalink(get_adjacent_post(false, "", false)->ID);?>).

Next Post URL

%WP_next_postURL% - This will be populated with the URL of the next post (<?php echo get_permalink(get_adjacent_post(false, "", true)->ID);?>).

Author details

Author

%WP_author% - This will be populated with the current page/post author name (<?php the_author(); ?>).

Author Link

%WP_authorLink% - This will be populated with a link to the authors post page (<?php echo get_author_posts_url(get_the_author_meta('ID')); ?>).

Author Email

%WP_authorEmail% This will be populated with the current page/post author email (<?php echo \$current_user->user_email; ?>).

Author First Name

%WP_authorFirstName% This will be populated with the current page/post author first name (<?php echo \$current_user->user_firstname; ?>).

Author Last Name

%WP_authorLastName% This will be populated with the current page/post author last name (<?php echo \$current_user->user_lastname; ?>).

Author Bio

%WP_authorBio% - This will be populated with the current page/post author bio content (<?php echo \$current_user->user_description; ?>).

Author Website

%WP_authorWebsite% - This will be populated with the current page/post author website URL (<?php echo \$current_user->user_url; ?>).

Author Display Name

%WP_authorDisplayName% - This will be populated with the current page/post author display name (<?php echo \$current_user->display_name; ?>).

Logged In User

User Name

%WP_userName% - This will be populated with the logged in user name (<?php echo \$current_user->user_login; ?>).

User Email

%WP_userEmail% This will be populated with the logged in user email (<?php echo \$current_user->user_email; ?>).

User First Name

%WP_userFirstName% This will be populated with the logged in users first name (<?php echo \$current_user->user_firstname; ?>).

User Last Name

%WP_userLastName% This will be populated with the logged in users last name (<?php echo \$current_user->user_lastname; ?>).

User Bio

%WP_userBio% - This will be populated with the logged in users bio content (<?php echo \$current_user->user_description; ?>).

User Website

%WP_userWebsite% - This will be populated with the logged in users website URL (<?php echo \$current_user->user_url; ?>).

Display Name

%WP_userDisplayName% - This will be populated with the logged in users display name (<?php echo \$current_user->display_name; ?>).

Custom Field Data

To display post and page custom field data use the following short code. Take care to add your custom field name within the brackets of the shortcode and make sure you add the custom field type to your project via the Wordpress Custom Data Manager within Blocs (Main Menu > Window > Wordpress Custom Data Manager).

%WP_CustomField(custom field name here)% This will be populated with the custom field content set for the post or page (<?php echo get_post_meta(get_the_ID(), 'field name', TRUE);?>).