

# Conductor Plugin

Pre-release User Guide

6.26.14

[slocumthemes.com](http://slocumthemes.com)

# Table of contents

[Table of contents](#)

[Introduction](#)

[How to install Conductor](#)

[Getting started with Conductor](#)

[What is a layout?](#)

[What is a content display?](#)

[Video walkthrough](#)

[Screenshot walkthrough](#)

[Adjusting Conductor Layouts](#)

[Conducting Layouts from the traditional dashboard:](#)

[Conducting Layouts from the live front end customizer:](#)

[Why would you want to do this?](#)

[Video walkthrough](#)

[Screenshot walkthrough](#)

[Conductor Widget Overview](#)

[The 3 panels of the Conductor widget:](#)

[Content settings panel](#)

[Display settings panel](#)

[Advanced settings](#)

[Video walkthrough](#)

[Adding a Conductor widget](#)

[Adding the Conductor Widget from the Widgets subpanel:](#)

[Adding the Conductor Widget from the live front end customizer:](#)

[Why do you want to do this?](#)

[Actions & Filters](#)

[Actions](#)

[Conductor Widget](#)

[Conductor Widget Query](#)

[Conductor Templates](#)

[Conductor Template Functions](#)

[Conductor Admin \(Dashboard\) Hooks](#)

[Filters](#)

[Conductor Widget](#)

[Conductor Widget Query](#)

[Conductor Templates](#)

[Conductor Template Functions](#)

[Conductor Sidebars](#)

[Conductor Options](#)

[Conductor Admin Options](#)

[Conductor Customizer](#)



# Introduction

Thank you for using the Conductor plugin.

Conductor was born from a need we discovered within our custom WordPress work at Slocum Studio. While building content-rich sites for our clients, we were constantly asked for a solution to customize the website's display of content.

From site to site, we were building use cases for an easy-to-use plugin that produced a user friendly widget. Something that was powerful enough for developers, but easy enough for the common author or editor to manipulate.

WordPress is best and originally used for publishing content. However, most WordPress themes display content using the same stale display. Out of the box, WordPress doesn't allow us to curate unique collections of our content to put on display for our visitors.

Conductor changes all of that.

Leverage Conductor to make custom homepage layouts for your content or to make each category archive distinct from the other. This translates to a better visitor experience, an easier consumption of content, and a more powerful way to promote content the way that you want to.

Slocum Studio is very proud of Conductor and we hope you enjoy using it as much as we enjoy producing it.

If you have any questions, feel free to e-mail me: [matt@slocumstudio.com](mailto:matt@slocumstudio.com)

Thank you,

A handwritten signature in black ink, consisting of a stylized 'M' followed by a horizontal line.

Matt Medeiros  
Co-founder Slocum Studio

[slocumstudio.com](http://slocumstudio.com)  
[slocumthemes.com](http://slocumthemes.com)

# How to install Conductor

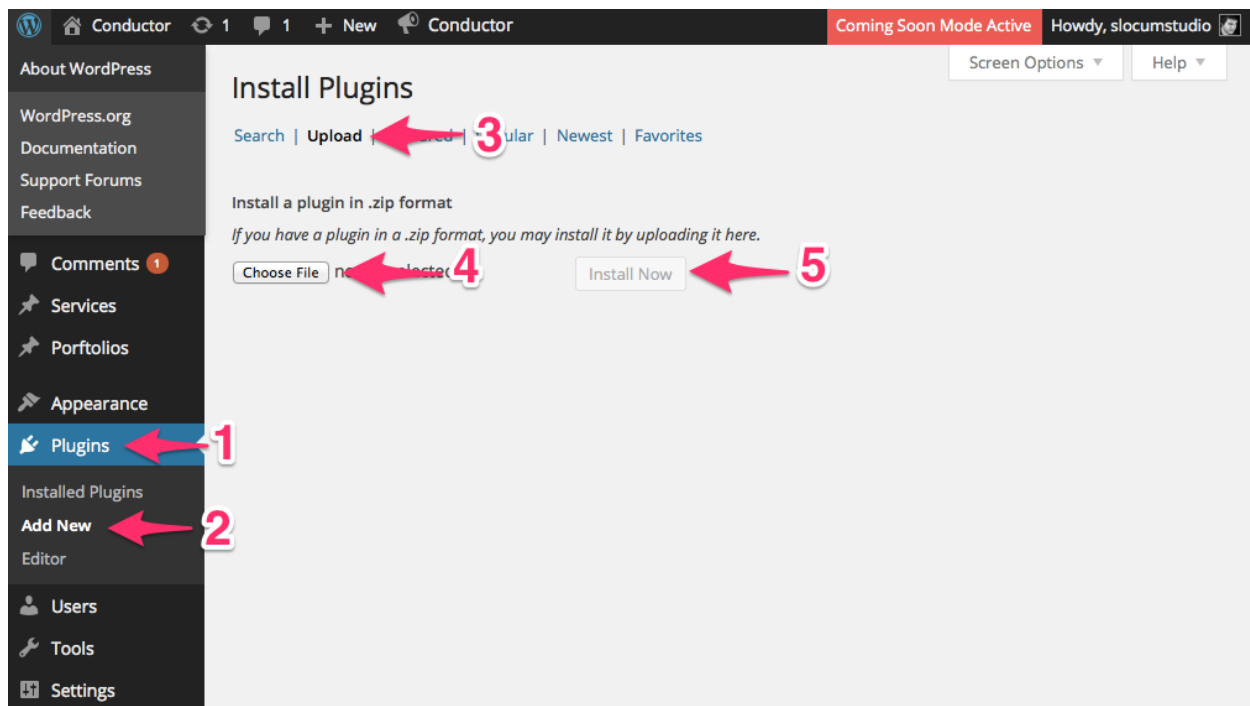
1. Download the plugin from your [customer](#) dashboard.
2. Login to your WordPress dashboard and browse to *Plugins > Add new*
3. Select *Upload*, then choose the **Conductor.zip** from your computer and *Install Now*.
4. If you're ready, select *Activate Plugin* on the final screen.

Now it's time to setup Conductor. View [this](#) article to learn how to get started.

## Video walkthrough

*video here*

## Screenshot walkthrough



# Getting started with Conductor

*How to use this document:*

1. *Read through the getting started guide*
2. *Continue to: [Adjusting Conductor Layouts](#)*

Let's get started!

Conductor allows the WordPress website owner to control the layout and display of content across their website.

**There are two approaches to managing the layouts & content display via Conductor:**

1. Using the traditional WordPress backend via the Widget dashboard & Conductor Options sub panel.
2. Using the live frontend customizer of your WordPress website.

You can use both approaches or settle in with whichever you feel the most comfortable with. There is no *wrong* way to use Conductor.

## What is a layout?



You can learn how to adjust layouts [here](#).

A *layout* allows you to add or remove WordPress sidebars on a page, post, category archive or frontpage.

Why would you want to do this?

You may want to customize your WordPress theme to display a unique homepage and have your *news* category page look different from your *sports* category or the traditional layout. You can mix and match or **conduct** your content layouts as many times as you wish. There are no limits.

## What is a content display?



You can learn how to adjust content displays [here](#).

A *content display* allows you to display your content in different shapes and sizes. Currently, you can display your content in small, medium, or large. We determine this by width percentage:

- Small - 25%
- Medium - 50%
- Large - 100%

Why would you want to do this?

This allows you to mix and match Conductor widgets within your layout. Have your most recent blog post positioned at the top of your site in a large (100%) block. Have your older posts displayed beneath that in a medium block size.

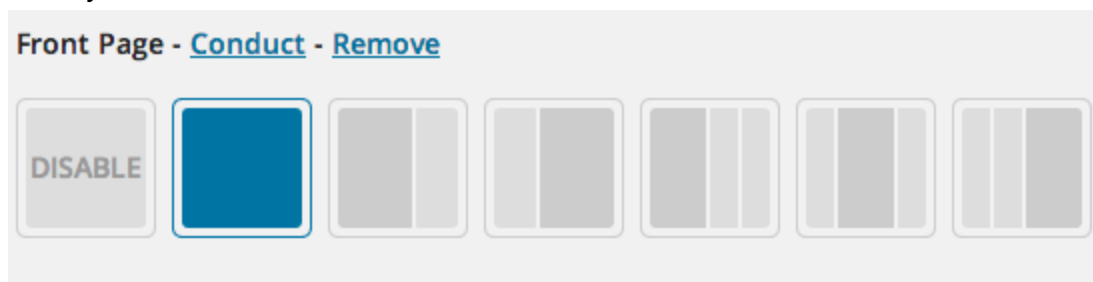
You can mix sizes and stack them on top or below other Conductor Widgets. This is a powerful way to make unique pages across your WordPress website!

## Video walkthrough

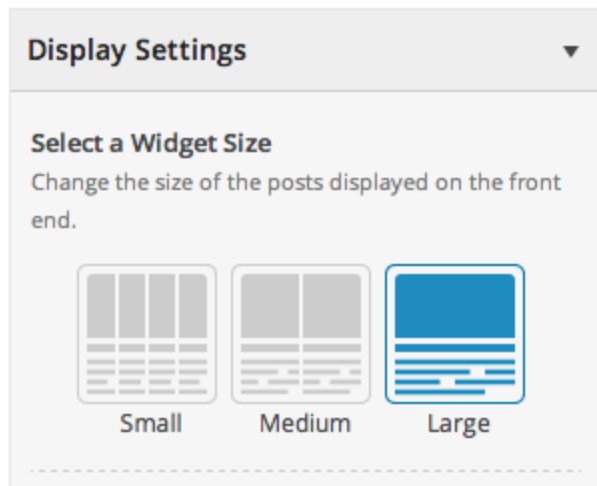
*video here*

## Screenshot walkthrough

*The layout selector:*



*The content display selector:*





# Adjusting Conductor Layouts

*How to use this document:*

1. Read through this guide about *Adjusting Conductor layouts*
2. Continue to: [Conductor Widget Overview](#)

Let's get started!


Conductor allows you to control the layout of your theme from the live front end customizer or via the Conductor options sub panel in your traditional WordPress dashboard.

## Conducting Layouts from the *traditional dashboard*:

1. Click on *Conductor > settings*
2. Enable Conductor Layouts by clicking on the enable switch
3. Under Conductor Layouts you can select which page, post, category archive or frontpage from the dropdown and click **Conduct this content type**.
4. Hover over the layout icons to visually select which layout you would like for the content type you selected.
5. Scroll to the bottom of the page and click Save Options.

When you browse to the front end of your website, the content you selected will have the layout you selected.

## Conducting Layouts from the *live front end customizer*:

1. When viewing your WordPress website while logged in, hover over the home link () in the admin bar and select *Customize*.
2. In the Customizer toolbar on the left, browse to Conductor Layouts.
3. Under Content Layouts you can select which page, post, category archive or frontpage from the dropdown and click **Conduct this content type**.
4. Hover over the layout icons to visually select which layout you would like for the content type you selected.
5. Click on Save or the WordPress customizer will auto-save for you.

## Why would you want to do this?

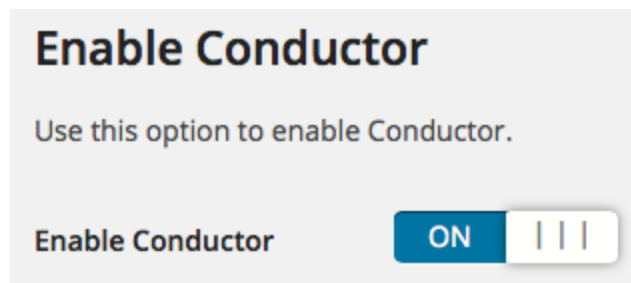
You may want to customize your WordPress theme to display a unique homepage and have your *news* category page look different from your *sports* category or the traditional layout. You can mix and match or **conduct** your content layouts as many times as you wish. There are no limits.

## Video walkthrough

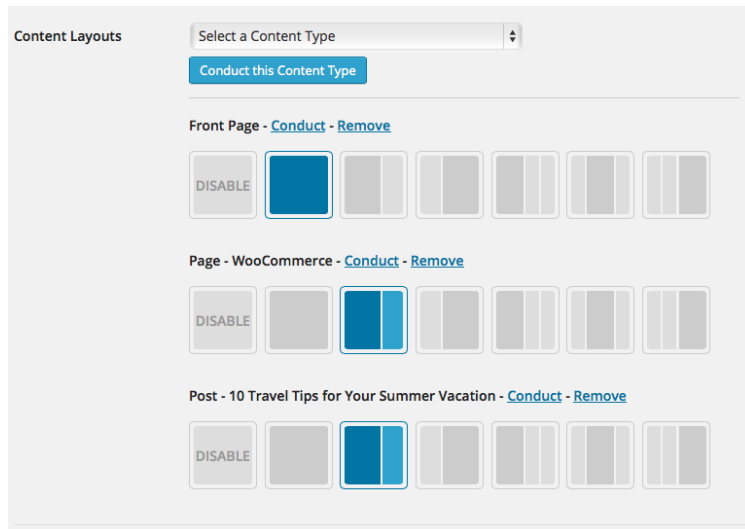
*video here*

## Screenshot walkthrough

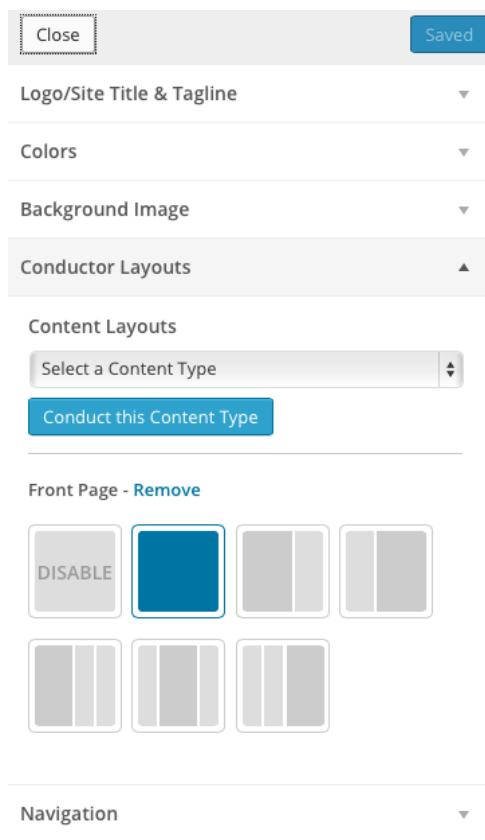
Enable Conductor



Select a Conductor Layout



Select a Conductor Layout in the live front-end customizer



# Conductor Widget Overview

*How to use this document:*

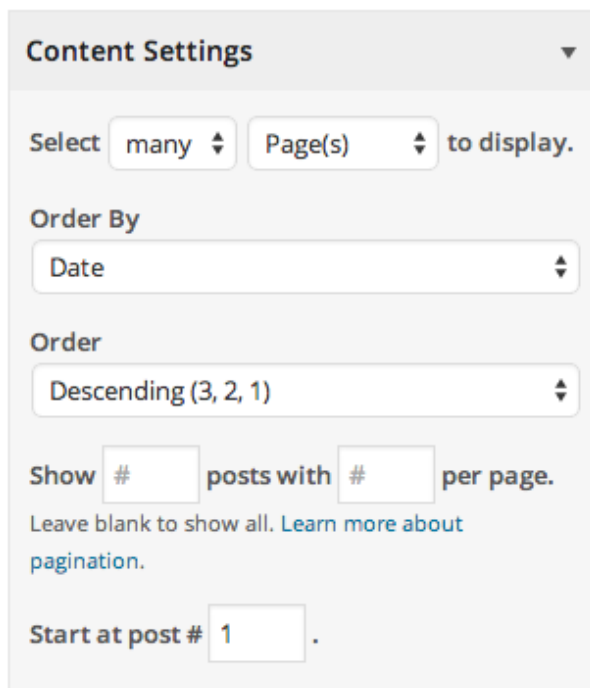
1. Review the settings of the 3 Conductor Widget panels
2. Continue to: [Adding a Conductor Widget](#)

Let's get started!

## The 3 panels of a Conductor widget:

- Content settings -- What content do you want to promote and how?
- Display settings -- What should this content look like when displayed?
- Advanced settings -- Fine tuning for the advanced user.

Let's review the decisions you make along the way:



The screenshot shows the 'Content Settings' panel of a Conductor widget. It includes a dropdown for 'Select many Page(s) to display.', an 'Order By' dropdown set to 'Date', an 'Order' dropdown set to 'Descending (3, 2, 1)', a 'Show # posts with # per page.' section with a link to 'Learn more about pagination.', and a 'Start at post # 1' field.


### Content settings panel

- Select **one** or **many**
- **Posts, Pages** or **Categories**
- Choose to order by
  - Date, Author, Comment Count, Most Comments, Post ID, Parent, Title, or Random
- In **descending** or **ascending** order.
- Show a set **number (#)** with a set **number (#)** to display per page.
- You can also offset by starting at **#**


Display Settings

Select a Widget Size


Change the size of the posts displayed on the front end.



Small



Medium



Large

☐ Hide Widget Title

☒ Hide Post Title(s)

☒ Hide Author Byline

☐ Hide Featured Image(s)

☐ Hide Read More Link

Limit content to  words.

Read More Label

Click here to learn about us

## Display settings panel

- Select from **Small** (25%), **Medium** (50%) and **Large** (100%) to display your widget.
- You can then choose to **hide elements**:
  - Widget title, Post title, Author byline, Featured image and the Read More link.
- If an **excerpt** exists, Conductor displays that by default. If none, a **summary of the content** is displayed.
- Choose to limit the content display to **number (#)** of words.
- Optionally overwrite the default “**Read More**” label.

Advanced Settings

Featured Image Size

mb-1200x475 (1200 x 475)

Featured Images are typically displayed based on the Widget Size option but you can choose a specific size if you'd like.

CSS Class(es)

Space separated list of custom CSS classes which can be used to target this widget on the front-end.

Include Only These Posts

Comma separated list of post IDs. Only these posts will be displayed. Some settings above will be ignored. [How do I find an ID?](#)

Exclude These Posts

Comma separated list of post IDs. Will display all posts based on settings above, except those in this list. [How do I find an ID?](#)

## Advanced settings

- Select a **featured image** size to display.
- Apply a **CSS Class** to further style your individual Conductor widgets
- You can **include** posts by using comma separated post ID's.
- You can **exclude** posts by using comma separated post ID's.

Now that you know how to adjust the settings for a Conductor widget, it's time to [add a Conductor Widget](#) to your sidebars.

## Video walkthrough

*video here*

# Adding a Conductor widget

*How to use this document:*

1. *Read through this guide about Adding a Conductor widget*
2. *If you're a developer, continue to Actions & Filters. If you're not a developer, that is the end of the user guide. Still have questions? E-mail us: [discover@slocumstudio.com](mailto:discover@slocumstudio.com)*

Let's get started!

The Conductor Widget is the heart of the Conductor plugin. You tell the widget what content you want to display and how you want it to look when it's on a page. Conductor stores this information in a widget, which means you can change the order, display, and location of your content on the fly!


*Have you read the [Conductor Widget Overview](#)?*


You can add Conductor widgets from within your traditional WordPress dashboard or from the live front end customizer of your website.

## Adding the Conductor Widget from the *Widgets subpanel*:

1. From within your WordPress dashboard, navigate to *Appearance > Widgets*
2. Click on the *Conductor widget* from your list of available widgets and choose the WordPress sidebar to add it to.
3. Choose the content and display settings within the widget and click save.
4. You can add/remove and stack any number of widgets.

## Adding the Conductor Widget from the *live front end customizer*:

1. When viewing your WordPress website while logged in, hover over the home link (  ) in the admin bar and select *Customize*.
2. In the Customizer toolbar on the left, browse to the desired sidebar location and add the Conductor widget.

*Optionally, you can click on the +Conductor icon (  ) which is represented by a dashed outline for the [Conductor Layouts](#) chosen.*

3. Choose the content and display settings within the widget and click save. The customizer will also auto-save for you.

4. You can add/remove and stack any number of widgets.

### **Why do you want to do this?**

The Conductor widget allows you to display Posts, Pages, and Categories from within your WordPress website. What developers refer to as “queries” is now easier than ever, with no coding necessary and just a few clicks of a button.

Promote any number of content collections, displayed anywhere you have a WordPress sidebar. Remember, Conductor widget works best when you have [Conductor Layouts](#) enabled.



## Actions & Filters

\*\*\* Attention: The following section is for advanced users and developers only.

### Actions

WordPress actions allow developers to “hook” into places within a codebase and execute their own functionality. You can read more about WordPress Actions [here](#).

#### Conductor Widget

The following actions are available within the Conductor Widget:

##### **conductor\_widget\_includes**

Used to include files/functionality specifically for use within the Conductor Widget. This hook is executed upon `__construct()`, or when a widget instance is created.

Arguments passed to callback function: `$this` (WP\_Widget)

Example:

```
add_action( 'conductor_widget_includes', 'my_conductor_widget_includes' );
function my_conductor_widget_includes( $instance ) {
    // Include functionality here...
    include_once( 'my_file.php' );
}
```

##### **conductor\_widget\_settings\_before**

Arguments passed to callback function: `$instance` (widget instance; settings), `$this` (WP\_Widget)

##### **conductor\_widget\_settings\_content\_section\_before**

Arguments passed to callback function: `$instance` (widget instance; settings), `$this` (WP\_Widget)

##### **conductor\_widget\_settings\_feature\_many**

Arguments passed to callback function: `$instance` (widget instance; settings), `$this` (WP\_Widget)

##### **conductor\_widget\_settings\_feature\_one**

Arguments passed to callback function: \$instance (widget instance; settings), \$this (WP\_Widget)

#### **conductor\_widget\_settings\_orderby**

Arguments passed to callback function: \$instance (widget instance; settings), \$this (WP\_Widget)

#### **conductor\_widget\_settings\_content\_section\_after**

Arguments passed to callback function: \$instance (widget instance; settings), \$this (WP\_Widget)

#### **conductor\_widget\_settings\_display\_section\_before**

Arguments passed to callback function: \$instance (widget instance; settings), \$this (WP\_Widget)

#### **conductor\_widget\_settings\_size\_preview**

Arguments passed to callback function: \$size (current widget size output to settings), \$instance (widget instance; settings), \$this (WP\_Widget)

#### **conductor\_widget\_settings\_size\_preview\_{\$size}**

Arguments passed to callback function: \$instance (widget instance; settings), \$this (WP\_Widget)

#### **conductor\_widget\_settings\_display\_section\_after**

Arguments passed to callback function: \$instance (widget instance; settings), \$this (WP\_Widget)

#### **conductor\_widget\_settings\_advanced\_section\_before**

Arguments passed to callback function: \$instance (widget instance; settings), \$this (WP\_Widget)

#### **conductor\_widget\_settings\_advanced\_section\_after**

Arguments passed to callback function: \$instance (widget instance; settings), \$this (WP\_Widget)

#### **conductor\_widget\_settings\_after**

Arguments passed to callback function: \$instance (widget instance; settings), \$this (WP\_Widget)

#### **conductor\_widget\_before**

Arguments passed to callback function: \$instance (widget instance; settings), \$args (widget arguments), \$this (WP\_Widget)

**conductor\_widget\_before**

Arguments passed to callback function: \$instance (widget instance; settings), \$args (widget arguments), \$this (WP\_Widget)

**conductor\_widget\_title\_before**

Arguments passed to callback function: \$instance (widget instance; settings), \$args (widget arguments), \$this (WP\_Widget)

**conductor\_widget\_title\_after**

Arguments passed to callback function: \$instance (widget instance; settings), \$args (widget arguments), \$this (WP\_Widget)

**conductor\_widget\_pagination\_before**

Arguments passed to callback function: \$instance (widget instance; settings), \$args (widget arguments), \$conductor\_widget\_query\_results (current query results), \$this (WP\_Widget)

**conductor\_widget\_pagination\_after**

Arguments passed to callback function: \$instance (widget instance; settings), \$args (widget arguments), \$conductor\_widget\_query\_results (current query results), \$this (WP\_Widget)

**conductor\_widget\_content\_pieces\_other**

Arguments passed to callback function: \$conductor\_widget\_query\_results (current query results), \$instance (widget instance; settings), \$args (widget arguments), \$this (WP\_Widget)

**conductor\_widget\_after**

Arguments passed to callback function: \$instance (widget instance; settings), \$args (widget arguments), \$this (WP\_Widget)

**conductor\_widget\_display\_content**

Arguments passed to callback function: \$post (current post being output), \$instance (widget instance; settings), \$this (WP\_Widget), \$this->conductor\_widget\_query (current query for this widget; query object)

**conductor\_widget\_display\_content\_{\$this->number}**

Arguments passed to callback function: \$post (current post being output), \$instance (widget instance; settings), \$args (widget arguments), \$this (WP\_Widget)

## Conductor Widget Query

The following actions are available within the Conductor widget query classes:

### **conductor\_query\_next\_posts**

Arguments passed to callback function: \$this (Conductor\_Widget\_Query instance)

### **conductor\_widget\_output\_before**

Arguments passed to callback function: \$post (current post being output), \$instance (widget instance; settings)

### **conductor\_widget\_featured\_image\_before**

Arguments passed to callback function: \$post (current post being output), \$instance (widget instance; settings)

### **conductor\_widget\_featured\_image\_after**

Arguments passed to callback function: \$post (current post being output), \$instance (widget instance; settings)

### **conductor\_widget\_post\_title\_before**

Arguments passed to callback function: \$post (current post being output), \$instance (widget instance; settings)

### **conductor\_widget\_post\_title\_after**

Arguments passed to callback function: \$post (current post being output), \$instance (widget instance; settings)

### **conductor\_widget\_author\_byline\_before**

Arguments passed to callback function: \$post (current post being output), \$instance (widget instance; settings)

### **conductor\_widget\_author\_byline\_after**

Arguments passed to callback function: \$post (current post being output), \$instance (widget instance; settings)

### **conductor\_widget\_post\_content\_before**

Arguments passed to callback function: \$post (current post being output), \$instance (widget instance; settings)

### **conductor\_widget\_after\_post\_content**

Arguments passed to callback function: \$post (current post being output), \$instance (widget instance; settings)

**conductor\_widget\_read\_more\_before**

Arguments passed to callback function: \$post (current post being output), \$instance (widget instance; settings)

**conductor\_widget\_read\_more\_after**

Arguments passed to callback function: \$post (current post being output), \$instance (widget instance; settings)

**conductor\_widget\_output\_after**

Arguments passed to callback function: \$post (current post being output), \$instance (widget instance; settings)

---

**Conductor Templates**

The following actions are available within the Conductor page templates:

**conductor\_content\_wrapper\_before**

Arguments passed to callback function: Conductor::get\_conductor\_content\_layout() (content layout for the current page)

**conductor\_content\_before**

Arguments passed to callback function: Conductor::get\_conductor\_content\_layout() (content layout for the current page)

**conductor\_content\_after**

Arguments passed to callback function: Conductor::get\_conductor\_content\_layout() (content layout for the current page)

**conductor\_primary\_sidebar\_before**

Arguments passed to callback function: Conductor::get\_conductor\_content\_layout() (content layout for the current page)

**conductor\_primary\_sidebar\_after**

Arguments passed to callback function: Conductor::get\_conductor\_content\_layout() (content layout for the current page)

**conductor\_secondary\_sidebar\_before**

Arguments passed to callback function: `Conductor::get_conductor_content_layout()` (content layout for the current page)

### **conductor\_secondary\_sidebar\_after**

Arguments passed to callback function: `Conductor::get_conductor_content_layout()` (content layout for the current page)

### **conductor\_content\_wrapper\_after**

Arguments passed to callback function: `Conductor::get_conductor_content_layout()` (content layout for the current page)

---

## **Conductor Template Functions**

The following actions are available within the Conductor page template functions:

### **conductor\_dynamic\_sidebar\_before**

Arguments passed to callback function: `$sidebar_id` (id of the current sidebar), `$content_layout` (content layout associated with current sidebar)

### **conductor\_{\$sidebar}\_dynamic\_sidebar\_before**

Arguments passed to callback function: `$sidebar_id` (id of the current sidebar), `$content_layout` (content layout associated with current sidebar)

### **conductor\_dynamic\_sidebar\_after**

Arguments passed to callback function: `$sidebar_id` (id of the current sidebar), `$content_layout` (content layout associated with current sidebar)

### **conductor\_{\$sidebar}\_dynamic\_sidebar\_after**

Arguments passed to callback function: `$sidebar_id` (id of the current sidebar), `$content_layout` (content layout associated with current sidebar)

---

## **Conductor Admin (Dashboard) Hooks**

The following actions are available within the Conductor admin:

### **conductor\_content\_types\_before**

### **conductor\_content\_types\_built\_in**

**conductor\_content\_types\_after**

**conductor\_options\_notifications**

**conductor\_options\_license\_form\_before**

**conductor\_options\_license\_form\_after**

**conductor\_admin\_options\_sidebar\_before**

**conductor\_admin\_options\_sidebar\_after**

## Filters

WordPress filters allow developers to “hook” into places within a codebase and modify data. You can read more about WordPress Filters [here](#).

### Conductor Widget

The following filters are available within the Conductor Widget:

#### **conductor\_widget\_widget\_options**

Used to adjust the Conductor Widget widget options before the WP\_Widget constructor is called.

Arguments passed to callback function: \$widget\_options (current widget options including CSS class name and description), \$this (WP\_Widget)

Example:

```
add_filter( 'conductor_widget_widget_options', 'my_conductor_widget_widget_options', 10, 2 );
function my_conductor_widget_widget_options( $widget_options, $widget ) {
    // Adjust CSS class name
    $widget_options['classname'] .= ' widget-conductor';

    return $widget_options.
}
```

#### **conductor\_widget\_control\_options**

Arguments passed to callback function: `$control_options` (current widget control options), `$this` (WP\_Widget)

### **conductor\_widget\_defaults**

Arguments passed to callback function: `$defaults` (current widget defaults), `$this` (WP\_Widget)

### **conductor\_widget\_sizes**

Arguments passed to callback function: `$conductor_widget_sizes` (array of widget sizes), `$instance` (widget instance; settings), `$this` (WP\_Widget)

### **conductor\_widget\_available\_image\_sizes**

Arguments passed to callback function: `$avail_image_sizes` (array of image sizes currently available), `$instance` (widget instance; settings), `$this` (WP\_Widget)

### **conductor\_widget\_update**

Arguments passed to callback function: `$new_instance` (new widget instance; settings), `$old_instance` (old, previous, widget instance; settings), `$this` (WP\_Widget)

### **conductor\_widget\_instance**

Arguments passed to callback function: `$instance` (widget instance; settings), `$args` ( array, will be extracted to `$before_widget`, `$after_widget`, `$before_title`, `$after_title`), `$this` (WP\_Widget)

### **conductor\_widget\_query\_args**

Arguments passed to callback function: `$conductor_widget_query_args` (array of query arguments for this widget), `$instance` (widget instance; settings), `$args` ( array, will be extracted to `$before_widget`, `$after_widget`, `$before_title`, `$after_title`), `$this` (WP\_Widget)

### **conductor\_widget\_query**

Arguments passed to callback function: `false` (use default widget query class), `$conductor_widget_query_args` (array of query arguments for this widget), `$instance` (widget instance; settings), `$args` ( array, will be extracted to `$before_widget`, `$after_widget`, `$before_title`, `$after_title`), `$this` (WP\_Widget)

### **conductor\_widget\_css\_classes**

Arguments passed to callback function: `$classes` (an array of CSS classes to be applied to this widget's output), `$instance` (widget instance; settings), `$this` (WP\_Widget)

### **widget\_title**

Arguments passed to callback function: `$instance['title']` (current widget title), `$instance` (widget instance; settings), `$this->id_base` (id base of widget), `$this` (WP\_Widget)



---

## Conductor Widget Query

The following filters are available within the Conductor widget query classes:

### **conductor\_query\_args**

Arguments passed to callback function: \$query\_args (“many” query args; arguments for WP\_Query), \$type (current type of query; single, many or custom type), \$this->widget\_instance (WP\_Widget), \$this (Conductor\_Widget\_Query instance)

### **conductor\_query\_other\_types**

Arguments passed to callback function: \$this->query (current query object), \$type (current type of query; single, many or custom type), \$this->widget\_instance (WP\_Widget), \$this (Conductor\_Widget\_Query instance)

### **conductor\_query\_have\_posts**

Arguments passed to callback function: false (no posts), \$this (Conductor\_Widget\_Query instance)

### **conductor\_query\_current\_post**

Arguments passed to callback function: \$post (current post object; WP\_Post, WP\_User, etc...), \$single (current type of query request; single, many or custom type), \$this (Conductor\_Widget\_Query instance)

### **conductor\_widget\_wrapper\_html\_element (name may change in future versions)**

Arguments passed to callback function: \$element\_tag (‘div’), \$post (current post object; WP\_Post, WP\_User, etc...), \$instance, \$widget (WP\_Widget), \$query (current query object), \$this (Conductor\_Widget\_Query instance)

### **conductor\_widget\_content\_wrapper\_html\_element (name may change in future versions)**

Arguments passed to callback function: \$element\_tag (‘section’), \$post (current post object; WP\_Post, WP\_User, etc...), \$instance, \$widget (WP\_Widget), \$query (current query object), \$this (Conductor\_Widget\_Query instance)

### **conductor\_query\_has\_pagination**

Arguments passed to callback function: \$has\_pagination (Boolean; whether or not the current query has pagination), \$this (Conductor\_Widget\_Query instance)

### **conductor\_query\_paginate\_links\_args**

Arguments passed to callback function: \$paginate\_links\_args (array of arguments for the pagination\_links() function), \$query (current query object), \$echo (Boolean; should pagination be returned or echoed), \$this (Conductor\_Widget\_Query instance)

### **conductor\_widget\_excerpt\_allowable\_tags (may eventually end up depreciated)**

Arguments passed to callback function: \$tags (array of allowed html tags for use in the strip\_tags() function), \$post (current post object)

### **conductor\_widget\_featured\_image\_size**

Arguments passed to callback function: \$conductor\_thumbnail\_size (string; size of featured image to be output), \$instance (widget instance; settings), \$post (current post object)

---

## **Conductor Templates**

The following filters are available within the Conductor page templates:

### **conductor\_content\_wrapper\_element\_before**

Arguments passed to callback function: \$html (<div class="conductor-container container conductor-cf">)

### **conductor\_content\_wrapper\_element\_after**

Arguments passed to callback function: \$html (</div>)

### **conductor\_content\_element\_before**

Arguments passed to callback function: \$html (<div class="conductor-content conductor-cf" . Conductor::get\_conductor\_content\_layout\_sidebar\_id( 'content' ) . "" data-sidebar-id="" . Conductor::get\_conductor\_content\_layout\_sidebar\_id( 'primary' ) . ""><div class="conductor-inner conductor-cf">)

### **conductor\_content\_element\_after**

Arguments passed to callback function: \$html (</div></div>)

### **conductor\_primary\_sidebar\_element\_before**

Arguments passed to callback function: \$html (<div class="conductor-primary-sidebar conductor-sidebar conductor-cf" . Conductor::get\_conductor\_content\_layout\_sidebar\_id( 'primary' ) . "" data-sidebar-id="" . Conductor::get\_conductor\_content\_layout\_sidebar\_id( 'primary' ) . ""><div class="conductor-inner conductor-cf">)

### **conductor\_primary\_sidebar\_element\_after**

Arguments passed to callback function: \$html (</div></div>)

### **conductor\_secondary\_sidebar\_element\_before**

Arguments passed to callback function: \$html (<div class="conductor-secondary-sidebar conductor-sidebar conductor-cf ' . Conductor::get\_conductor\_content\_layout\_sidebar\_id('secondary') . '" data-sidebar-id=" . Conductor::get\_conductor\_content\_layout\_sidebar\_id('secondary') . '"><div class="conductor-inner conductor-cf">)

### **conductor\_secondary\_sidebar\_element\_after**

Arguments passed to callback function: \$html (</div></div>)

---

## **Conductor Template Functions**

The following filters are available within the Conductor page template functions:

### **conductor\_get\_template\_part**

Arguments passed to callback function: \$template (current Conductor template file), \$slug (slug of the requested template), \$name (name, or secondary part of the requested template)

### **conductor\_content\_layout\_has\_sidebar**

Arguments passed to callback function: \$has\_sidebar (Boolean), \$sidebar (sidebar; content, primary, or secondary), \$content\_layout (current or passed content layout)

### **conductor\_is\_sidebar\_active**

Arguments passed to callback function: \$active\_sidebar (Boolean), \$sidebar (sidebar; content, primary, or secondary), \$content\_layout (current or passed content layout)

---

## **Conductor Sidebars**

The following filters are available within the Conductor sidebar functions:

### **conductor\_sidebar\_args**

Arguments passed to callback function: \$sidebar\_args (current sidebar arguments), \$sidebar (current sidebar name; 'content'), \$content\_layout (current or passed content layout), \$content\_layouts (all content layouts saved in database)

---

## Conductor Options

The following filters are available within the Conductor options functions:

### **conductor\_options\_defaults\_{\$option\_name}**

Arguments passed to callback function: \$defaults (current default values for the option name passed), \$option\_name (option name)

### **conductor\_options\_defaults (after the filter above)**

Arguments passed to callback function: \$defaults (current default values for the option name passed), \$option\_name (option name)

### **conductor\_content\_layouts**

Arguments passed to callback function: \$content\_layouts (array of available content layouts)

---

## Conductor Admin Options

The following filters are available within the Conductor admin options functions:

### **conductor\_admin\_options\_localize**

Arguments passed to callback function: \$conductor\_admin\_options\_localize (array of data that will be localized to the conductor-content-layouts script), \$hook (current page hook), \$protocol (https or http)

### **conductor\_sanitize\_option\_built\_in\_content\_types**

Arguments passed to callback function: \$content\_types (array of built in content types for sanitization)

### **conductor\_sanitize\_option\_other\_content\_types**

Arguments passed to callback function: \$temp\_value (current temporary value for this content layout), \$id (content layout id), \$value (content layout value), \$content\_type (what content type this is; category, post\_type, post, page, etc...), \$field\_name (field name; id of post, category, or name of post\_type, or built-in content type, etc...), \$input (all data passed to sanitize function)

### **conductor\_content\_layout\_edit\_link\_url**

Arguments passed to callback function: \$edit\_link\_url (url which links to customizer with correct url query argument attached), \$content\_layout (current content layout), \$content\_layouts (all content layouts saved in database)

**conductor\_options\_navigation\_tabs**

**conductor\_options\_settings**

---

### **Conductor Customizer**

The following filters are available within the Conductor customizer functions:

#### **conductor\_customizer\_preview\_options (options output in the Previewer window of the Customizer)**

Arguments passed to callback function: \$conductor\_options (array of options to be output in the Previewer), Conductor::is\_conductor( false ) (is Conductor?)