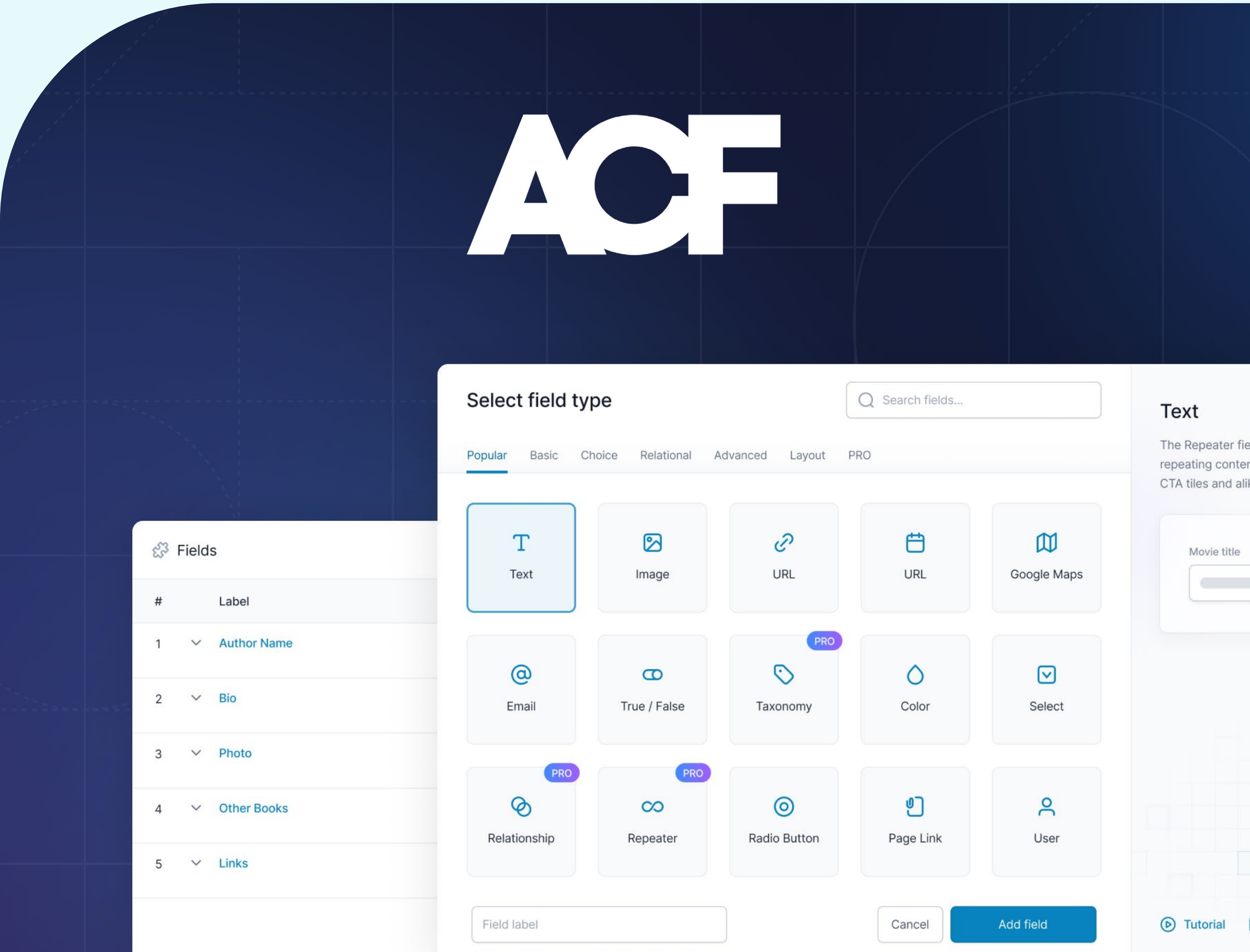
WP engine™

Getting Started With ACF®

Transform your website with custom content



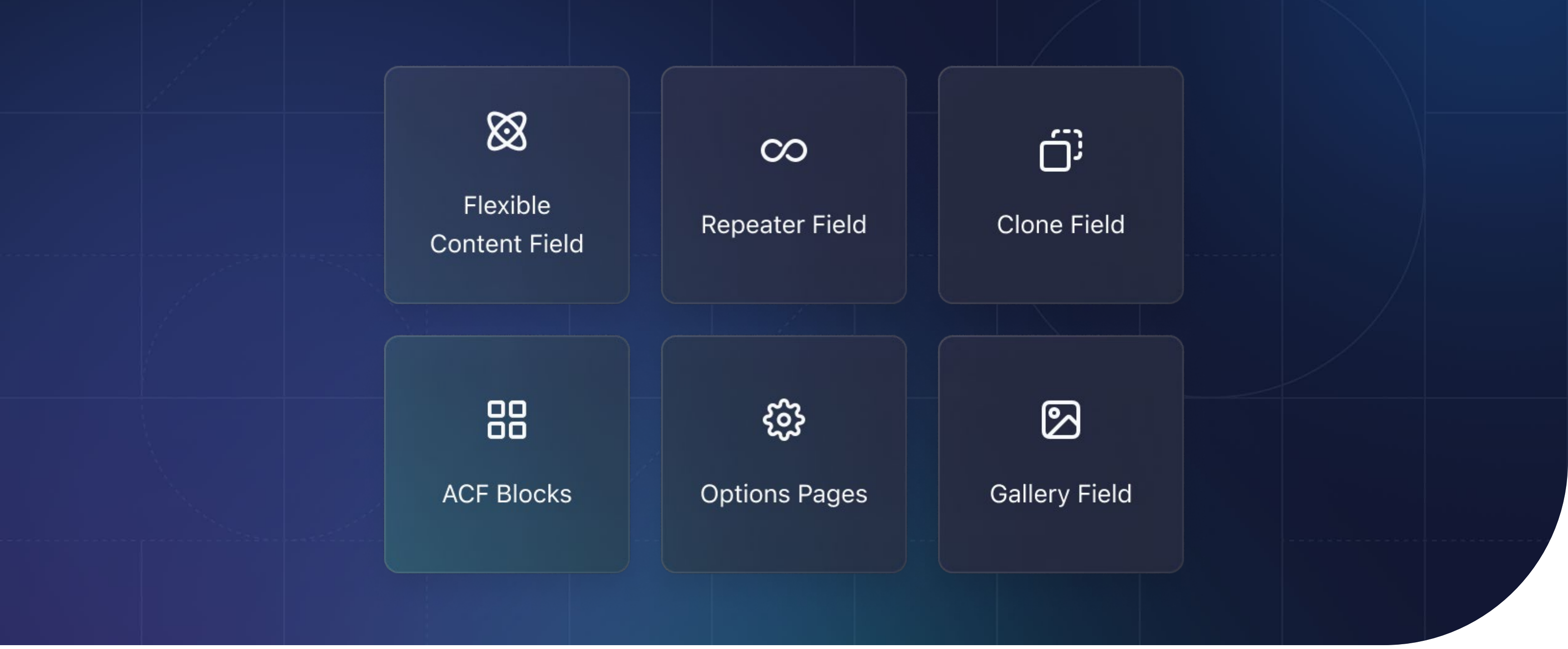
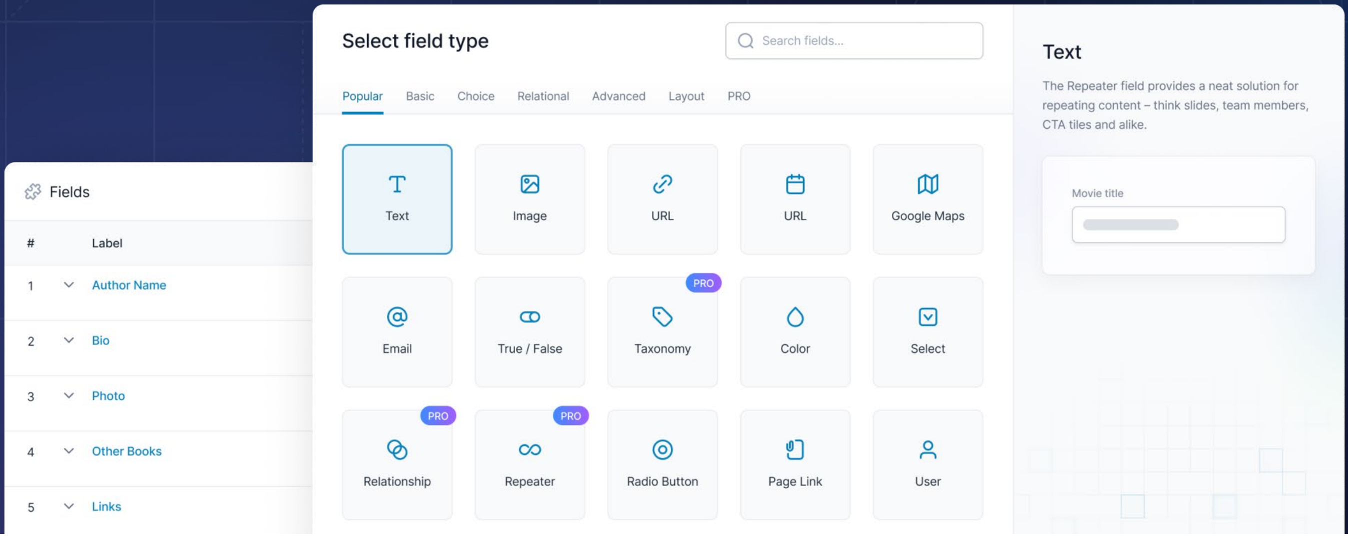


Table of Contents



Text

The Repeater field provides a neat solution for repeating content – think slides, team members, CTA tiles and alike.

#	Label
1	Author Name
2	Bio
3	Photo
4	Other Books
5	Links

Unlock your website's full potential

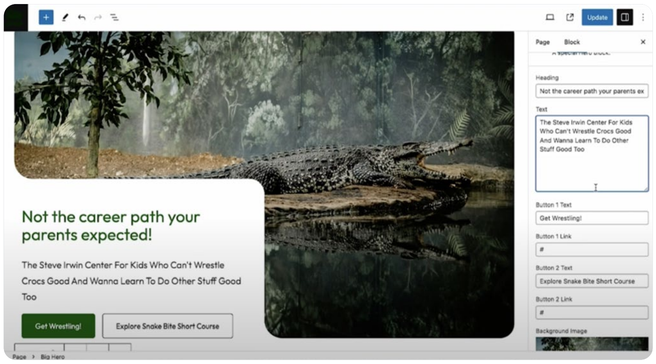
Wordpress® is the world's most widely used content management system (CMS).[1] While it has undeniable strengths, it can also be limited when it comes to handling complex, structured data, and the types of content it will display. This makes it difficult to create the feature-rich digital experiences that end users have come to expect.

This is where Advanced Custom Fields (ACF) comes in, empowering you to create custom, structured data that elevates your website's functionality and user experience. ACF's benefits make it an essential tool for any WordPress developer or site owner looking to create custom, data-driven content.

With ACF, you can:

- Easily create and manage custom fields for posts, users, taxonomy terms, and media attachments
- Model content in a flexible and scalable way, using ACF's intuitive field group and field type system
- Integrate ACF with popular page builders and themes, such as Elementor
- Leverage ACF's extensive library of field types, including Select, Google Map, and Icon Picker fields
- Take advantage of ACF's robust API and hooks system to extend and customize its functionality

[1] WP Engine is a proud member and supporter of the community of WordPress® users. The WordPress® trademarks are the intellectual property of the WordPress Foundation, and the Woo® and WooCommerce® trademarks are the intellectual property of WooCommerce, Inc. Uses of the WordPress®, Woo®, and WooCommerce® trademarks in this website are for identification purposes only and do not imply an endorsement by WordPress Foundation or WooCommerce, Inc. WP Engine is not endorsed or owned by, or affiliated with, the WordPress Foundation or WooCommerce, Inc.



Not the career path your parents expected!

The Steve Irwin Center For Kids Who Can't Wrestle Crocs Good And Wanna Learn To Do Other Stuff Good Too

Heading: Not the career path your parents ex

Text: The Steve Irwin Center For Kids Who Can't Wrestle Crocs Good And Wanna Learn To Do Other Stuff Good Too

Button 1 Text: Get Wrestling!

Button 1 Link: #

Button 2 Text: Explore Snake Bite Short Course

Button 2 Link: #

Background Image

ACF powers a source of structured data that's customized to your needs, while giving content editors a clear, simple, and controlled way to add to that data, improving functionality and elevating the experience of the WordPress site's end users.

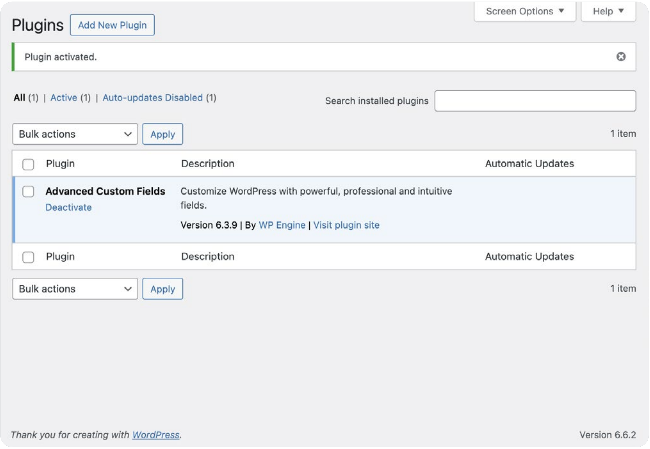
The limitations of default WordPress content modeling

WordPress's native custom fields aren't particularly user-friendly, and they don't play well with more complex data types like dates or dropdowns. While you could code extra areas of the edit screen, this requires valuable development time and effort. ACF revolutionizes the way you approach custom field creation and management, providing a simple and intuitive way to elevate your WordPress content modeling.

ACF includes over [30 different custom field types](), but a complete solution for content modeling must also handle custom post types and taxonomies. ACF does this as well, allowing you to create new custom post types in seconds directly through the plugin's UI, attach or create new field groups, and describe custom taxonomies in just a few clicks.

First steps with ACF

Ready to transform your WordPress sites? [Download the most recent version of ACF](), navigate to Plugins > Add new in your WordPress admin, upload the zip file, and activate the plugin. Start with the free version and [upgrade to PRO]() for the full range of features when you're ready.



Plugin activated.

All (1) | Active (1) | Auto-updates Disabled (1)

Search installed plugins

1 item

	Plugin	Description	Automatic Updates
□	**Advanced Custom Fields** Deactivate	Customize WordPress with powerful, professional and intuitive fields. Version 6.3.9 | By WP Engine | Visit plugin site	
□	Plugin	Description	Automatic Updates

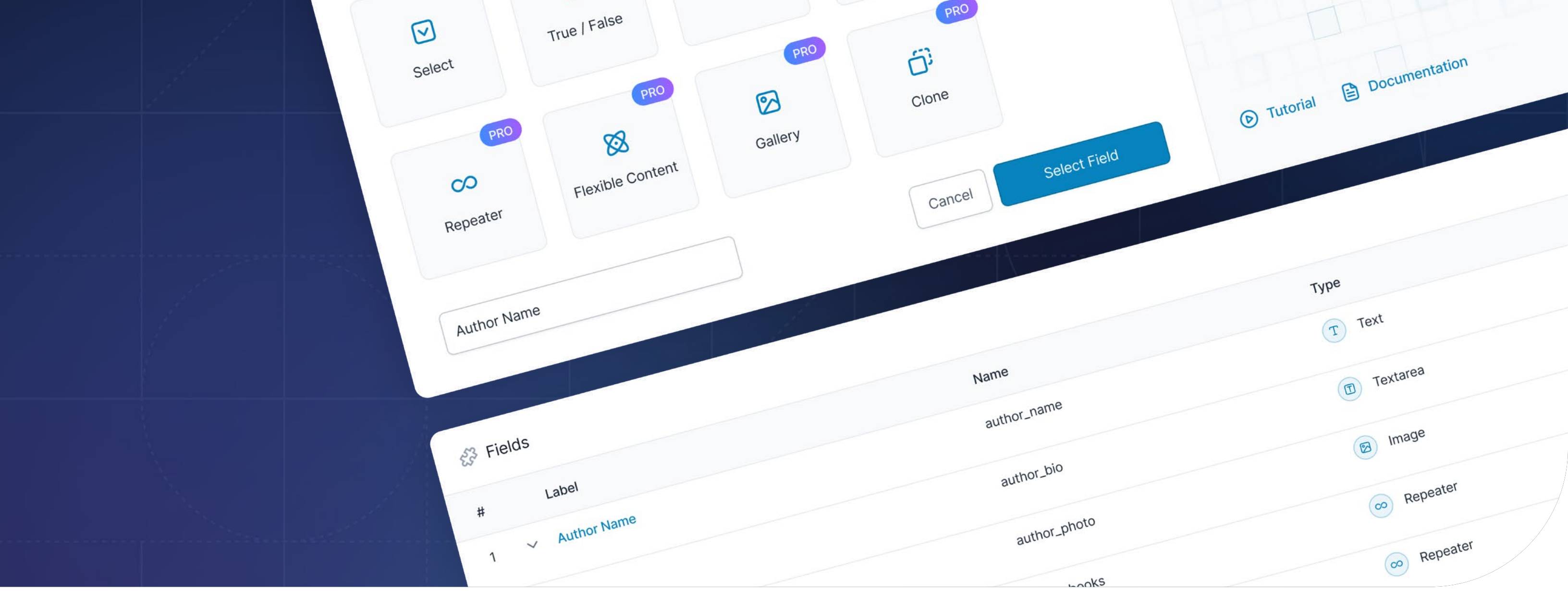
1 item

Thank you for creating with WordPress.

Version 6.6.2

With ACF, you're not just building sites–you're crafting elevated experiences for both editors and end users. By maximizing the power of the WordPress CMS, you can create custom, structured data that elevates your website's functionality and user experience. Whether you're building a site for an auto dealership, an eCommerce platform, or an enterprise client, ACF empowers you to craft tailored solutions that meet your unique needs.

In the following chapters, we'll explore how ACF simplifies the creation and management of custom fields, post types, and taxonomies, providing your WordPress sites with structured data and an intuitive way to edit it.



#	Label	Name	Type
1	Author Name	author_name	Text
		author_bio	Textarea
		author_photo	Image
			Repeater
		books	Repeater

Understanding content modeling in WordPress

WordPress offers a set of default content types, including posts and pages, which can be extended and customized using custom post types, taxonomies, and custom fields. However, this has limitations. For example, it does not provide a built-in way to create complex, hierarchical content models or to define relationships between different content types.

Content in WordPress has two parts. The first is the words, images, etc., added in the WordPress editor. The second part is metadata, which is information *about* the content. This includes the title, author, publication date, and so on.

These different types of content are usually referred to as Post Types. This can be confusing, as one of the Post Types is simply called Post. The other default Post Types are Page, Attachment, Revision, Navigation Menu, Block Templates, and Template Parts.

Posts and Pages are the most common Post Types, followed by Attachments. In brief, a Post is a blog entry, article, or a similar content piece, while a Page is typically used for more static content, such as an "About Us" section. The Attachment Post Type is used to display images or link to files.

By default, WordPress creates different types of metadata for each of these. For example, Posts have date and time stamps, and will often have categories and tags applied. Pages, on the other hand, do not have time stamps, aren't organized with categories and tags, and can be organized into a hierarchy, which you cannot do with Posts.

Going beyond what the WordPress core software offers requires creating custom fields to include more metadata, custom post types to better reflect particular types of content, and custom taxonomies to organize it all. ACF gives you the ability to do all of this, without the need for manual coding.

Custom fields

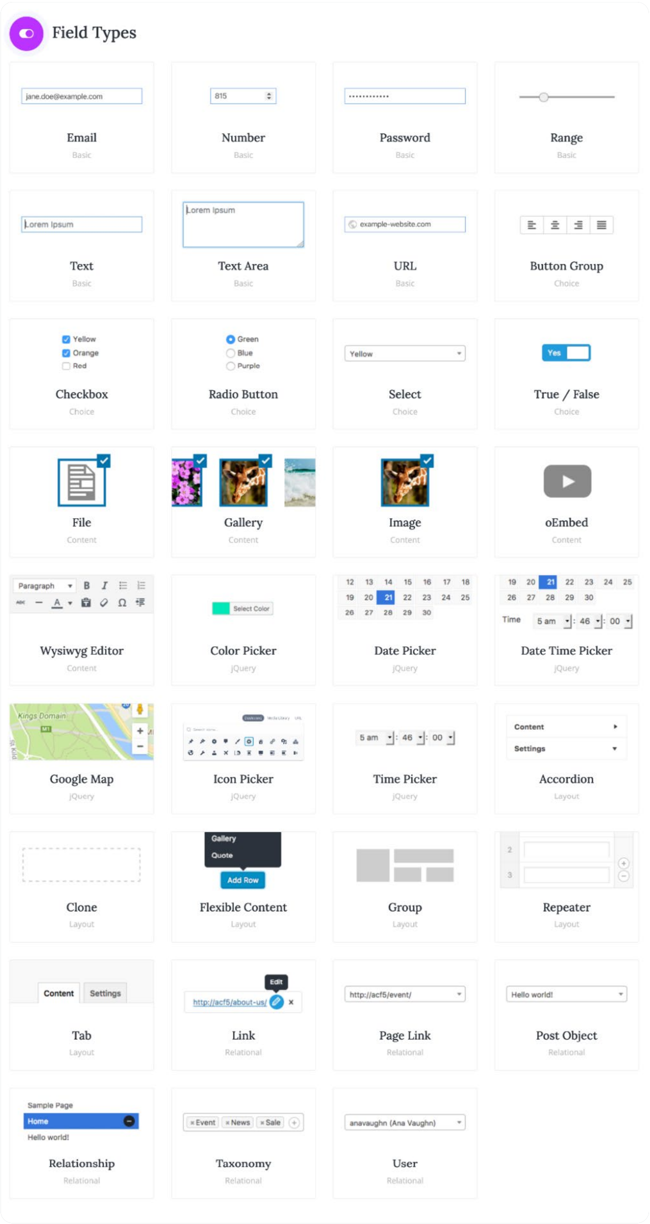
Custom fields allow users to add additional data to WordPress content. They can store extra information about a post, such as a featured image, a video URL, or a custom description. Custom fields can also be added to custom post types, allowing developers to create complex content models with multiple fields and data types.

Custom post types

Custom post types extend the default post and page content types in the WordPress CMS. They allow developers to create custom content types that are tailored to specific needs, such as products, events, or portfolio items. Custom post types can have their own set of fields, taxonomies, and templates, making them a powerful tool for creating complexcontent models.

Taxonomies

Taxonomies are a way to categorize and organize content in WordPress. They can be thought of as a grouping mechanism that helps to classify content into categories, tags, or other custom taxonomies. The WordPress content model provides two default taxonomies: categories and tags. However, developers can create custom taxonomies to suit specific needs, such as product categories or event types.



The ACF advantage

Creating custom fields, post types, and taxonomies for a WordPress site typically requires coding. This applies to even the simplest of custom fields. For example, to create a custom text field in WordPress without using a plugin, you could use the `**add_meta_box()**` function to add a meta box to the post editor, and then use HTML to create the text input field within that meta box. Below is an example of what that might look like.

```
// Add the meta box
add_action('admin_menu', 'my_post_options_box');
function my_post_options_box() {
    add_meta_box('post_info', 'Custom Text Field', 'custom_post_info', 'post', 'side', 'high');
}

// Create the text input field
function custom_post_info() {
    global $post;
    ?>
    <fieldset id="mycustom-div">
        <div>
            <p>
                <label for="custom_text">Custom Text:</label>
                <br />
                <input type="text" name="custom_text" id="custom_text" value="<?php echo get_post_
meta($post→ID, 'custom_text', true); ?>">
            </p>
        </div>
    </fieldset>
    <?php
}

// Save the custom text field data
add_action('save_post', 'custom_add_save');
function custom_add_save($postID) {
    if ($_POST['custom_text']) {
        update_post_meta($postID, 'custom_text', $_POST['custom_text']);
    }
}
```

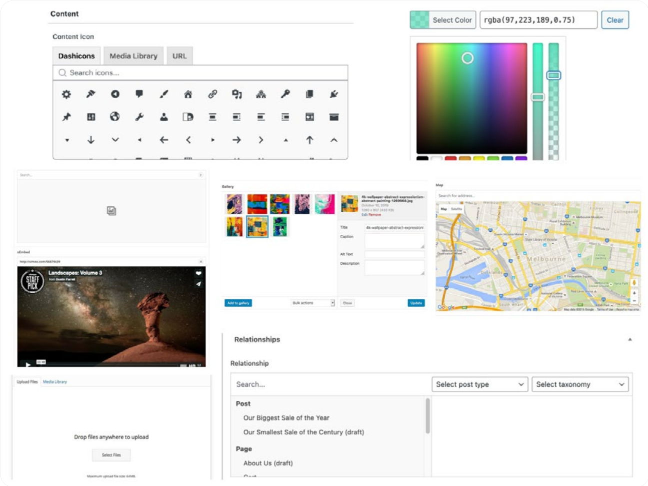
This code adds a meta box to the post editor with a text input field labeled "Custom Text." When the post is saved, the value of the text input field is saved as post metadata. Once you've got the coding itself done, you could add it to a custom plugin or create a template file in a child theme.

It's a lot of work just to get a custom text field to use on a single project! The work itself also tends to disrupt the development workflow. Just think: you're happily building a new WordPress site for yourself or a client when you realize you need a custom field. You drop out of the WordPress backend, fire up your code editor, code the custom field you need, and then it's back into the WordPress admin to continue building the site.

Everything goes well for five minutes, 10 minutes, an hour...and then you realize you need a *different* custom field. This time, it's not just a simple text field, but an URL. Or an Image. Or maybe you need a way to let content editors insert a Google Map into a post, with the marker determined by them at the time. Once again, your workflow is broken, and the code needed for these fields is much more complex.

This is why so many WordPress professionals depend on ACF. With over 30 custom fields built in, they don't need to constantly reinvent the wheel. And since fields are created and modified directly in ACF's UI, they don't need to leave the WordPress backend to do it.

Although easy creation and management of custom fields may be ACF's most well-known capability, ACF also lets you create custom post types and custom taxonomies at will, without leaving the WordPress backend or using another piece of software.



How selected ACF fields might appear when editing a post or page in WordPress. From the top left: Icon Picker, Color Picker, oEmbed, Gallery, Google Map, Image, and Relationship fields.

WordPress provides a solid foundation for content modeling, with built-in support for custom post types, taxonomies, and custom fields. ACF builds on and enhances this foundation while enabling a smoother development workflow.

Creating fields and field groups

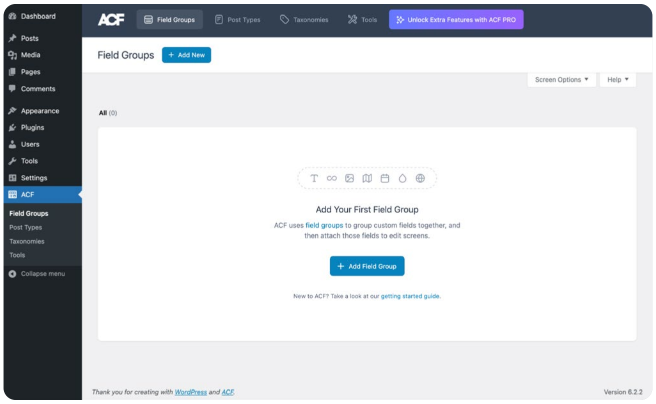
ACF provides a powerful way to add structured data to WordPress sites, enabling you to create custom fields and field groups to store and display data on your site. In this chapter, we'll take a step-by-step approach to creating a field group with ACF, explaining the different field types and how to configure them. We'll also include screenshots and code examples to illustrate the process.

What are field groups?

Field groups are a fundamental concept in ACF, organizing the custom fields used to store and display data on your WordPress sites. You can think of a field group as a container that holds one or more fields, which can be used to capture specific types of data. For example, you might create a field group called "Additional Post Details" that includes fields for a subtitle, author bio, and featured image.

Creating a field group

To create a field group, navigate to **ACF > Field Groups** in your WordPress dashboard and click **Add New**.



Add Your First Field Group

ACF uses field groups to group custom fields together, and then attach those fields to edit screens.

New to ACF? Take a look at our getting started guide.

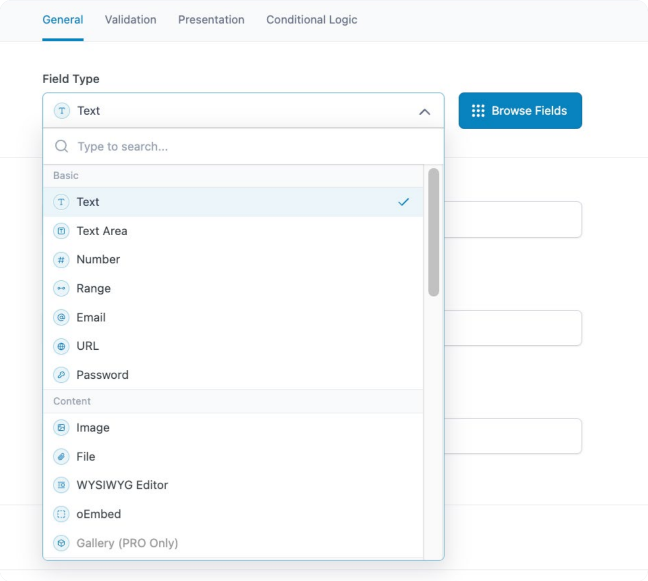
Thank you for creating with WordPress and ACF.

Version 6.2.2

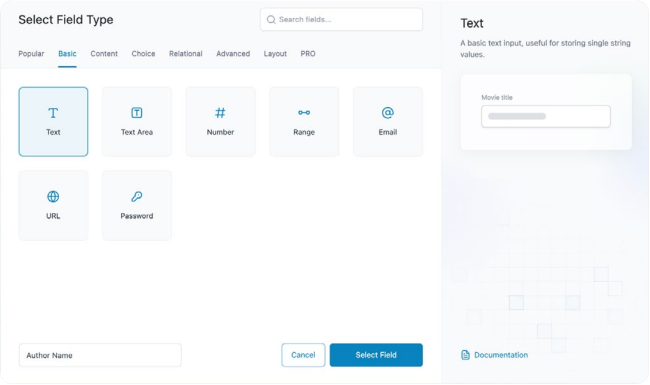
From there, you can add as many fields as you wish, with a great degree of control over each field's validation, presentation, and conditional logic settings. Field groups also include location rules, allowing you to control which post types, pages, or other locations the field group should appear.

Fields

Field groups start with a single field, which defaults to a Text field on creation. You can click the menu under "Field Type" to search or scroll through the available fields.



Clicking on **Browse Fields** opens a modal with fields sorted into tabs, providing a short description, graphical representation, and links to documentation for each field.



Text

A basic text input, useful for storing single string values.

Movie title

Author Name

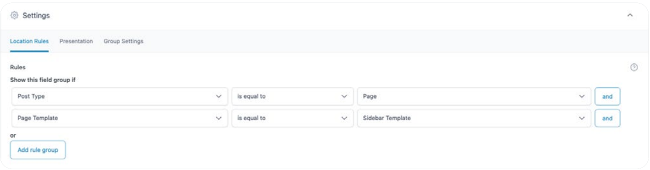
All fields must have a "Field Label" and "Field Name." The Field Label is visible when entering content but is not used by the API, while the Field Name is autogenerated from the field label and is used in the API and templates to retrieve content and data. Other settings change based on the type of field, such as "Default Value."

Settings

The settings panel is located beneath the field group editor, with tabs allowing you to set location rules, alter how the field will look to content editors, and adjust group settings.

Location rules

The Location Rules tab allows you to create rules that determine when and where these fields are added to the edit screen or post object. Location rules can be grouped together to create any combination of `and` and `or` statements.



Rules

Show this field group if

Post Type | is equal to | Page

Page Template | is equal to | Sidebar Template

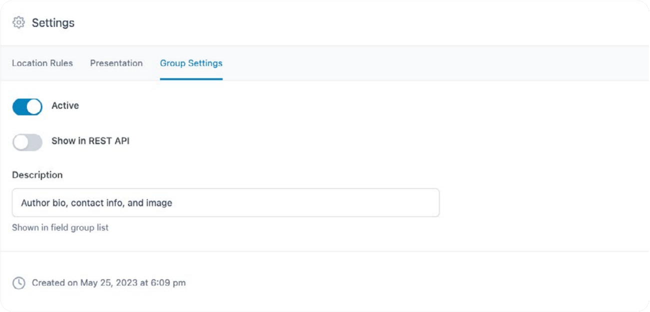
or

Presentation

The Presentation tab provides options to help customize the edit screen where this field group appears. You can also set the "Order No." on this tab, indicating the order in which field groups and their options should be loaded.

Group settings

You can deactivate your field group on this tab by means of the "Active" toggle, which defaults to on. Another toggle allows you to make the field group visible in the REST API, which defaults to off. Finally, you can fill in a description of your field group that will show in the field group list.



Active

Show in REST API

Description

Author bio, contact info, and image

Shown in field group list

Created on May 25, 2023 at 6:09 pm

In the next section, we'll show you how to create a basic ACF field group called "Contributor Info."

Creating a contributor info field group in ACF

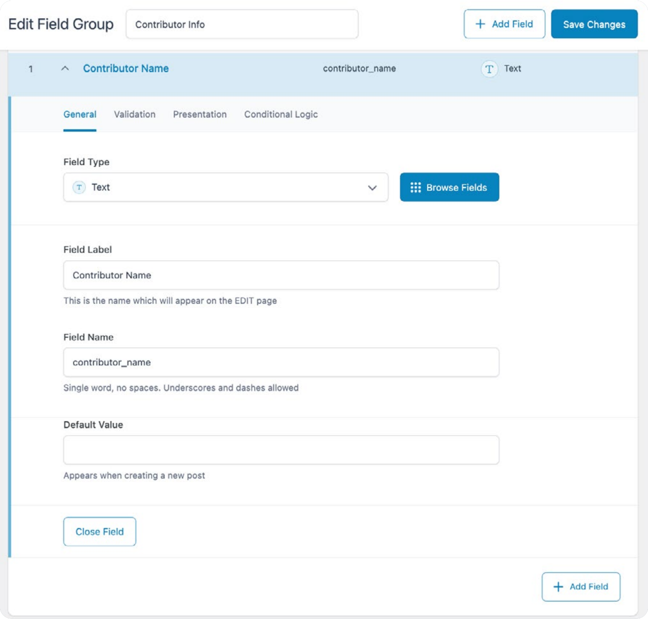
The "Contributor Info" field group is designed to allow new content editors and website contributors to easily add personal details and have these appear alongside their posts. The field group will consist of a Text field for their name, an Image field for their photo, a URL field for their personal site or social media profile, and an Email field so the site's end users can contact them.

Step 1: Create a New Field Group

- Log in to your WordPress admin dashboard and hover over the **ACF** menu item.
- Click on **Field Groups** and then click the **Add New** button.
- Give your field group a title, such as "Contributor Info".

Step 2: Add Fields to the Group

By default, every field group starts with a single Text field. You can change this to any field you wish, but we want a Text field for our "Contributor Info" group, so we'll just leave it as it is and enter "Contributor Name" for the Field Label. Click on the **Field Name** area, and ACF will automatically generate it for you based on the label you entered. You may modify this if you wish.



Edit Field Group: Contributor Info

1 Contributor Name contributor_name Text

Field Type: Text

Field Label: Contributor Name

This is the name which will appear on the EDIT page

Field Name: contributor_name

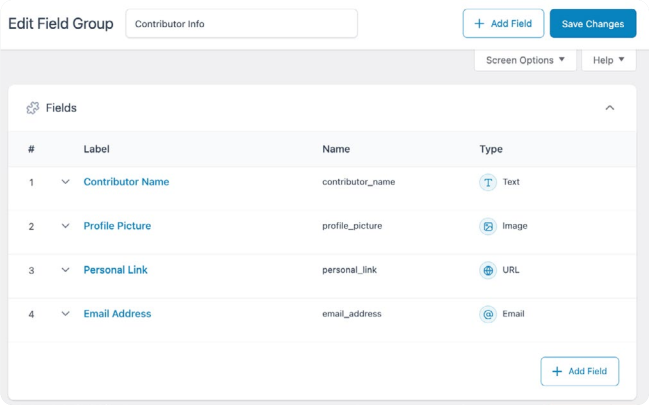
Single word, no spaces. Underscores and dashes allowed

Default Value:

Appears when creating a new post

Click **Add Field**, click the dropdown menu under "Field Type", and select **Image**. We'll give this a Field Label of "Profile Picture". Next, click **Add Field** again, select **URL**, and give it a Field Label of "Personal Link". Click **Add Field** one more time, and this time select **Email**, with a Field Label of "Email Address".

When you're done, your field group should look like this:



Edit Field Group: Contributor Info

#	Label	Name	Type
1	Contributor Name	contributor_name	Text
2	Profile Picture	profile_picture	Image
3	Personal Link	personal_link	URL
4	Email Address	email_address	Email

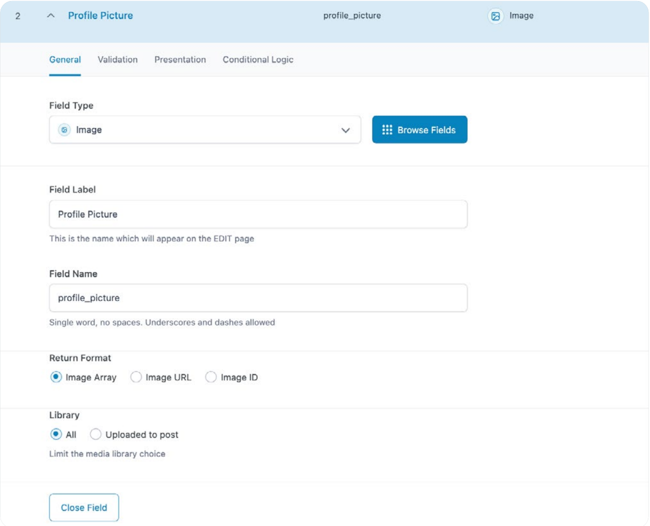
Field configuration: The General tab

The field group editor shows four tabs for any given field: General, Validation, Presentation, and Conditional Logic. The "General" tab is the only one used so far in this guide, as that's where we select the field type and set names and labels.

Fields usually have other settings that can be configured on this tab, with the settings available dependent on the type of field. The Text, URL, and Email fields used above include a "Default Value" setting. Providing a default value in these cases isn't required, but you may find it useful to give content editors a default input in some situations. For example, the "Email Address" field could have a default value of "marketing@ourcompany.com" for contributors who don't want their own email address to show with their posts.

Those fields—and some others, such as Range and Number—have no other settings on the General tab. That's all they need.

The Image field, however, requires different settings to function. "Return Format" specifies how the data will be returned: as an array, a URL, or an ID, with "array" selected by default. The "Library" setting allows you to control if the image selected can be from the site's entire Media Library (the default) or just from images uploaded with this post.



Field Type
Image

Field Label
Profile Picture
This is the name which will appear on the EDIT page

Field Name
profile_picture
Single word, no spaces. Underscores and dashes allowed

Return Format
● Image Array ○ Image URL ○ Image ID

Library
● All ○ Uploaded to post
Limit the media library choice

As noted above, the settings available for any given field are strongly related to the field itself. It's a good practice to review each field's documentation the first time you use it, so you can become familiar with the available settings.

More specific refinements to these fields can be made on the other tabs, which we'll cover in the next section.

Field configuration: The Validation, Presentation, and Conditional Logic tabs

The Validation, Presentation, and Conditional Logic tabs give you fine-grained control over your custom fields, including the types of input they'll accept and how they appear to content editors.

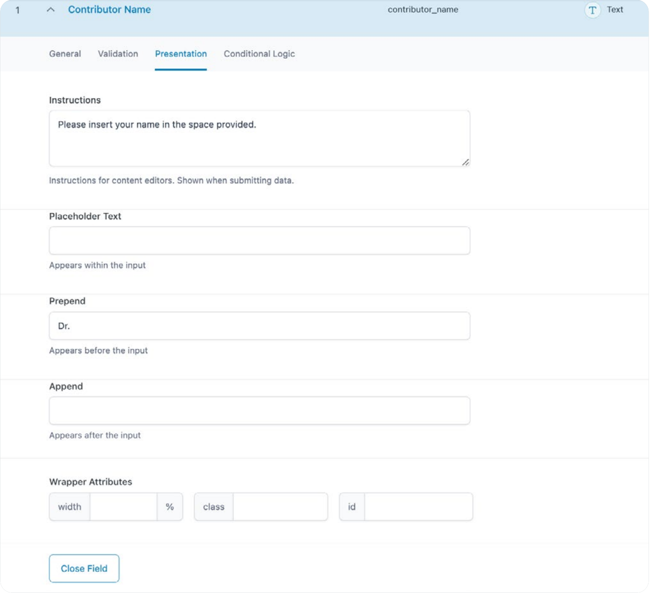
At a bare minimum, the "Validation" tab will include a toggle to mark the field "Required". Toggle this on for any situation where you want to ensure a value is entered in the field.

Most fields will have other validation settings related to the field's expected input. Text and Text Area fields, for example, allow you to specify the maximum number of characters allowed. The Image field, on the other hand, allows you to set minimum and maximum dimensions, as well as limit the input to one or more file types. ACF's validation settings vary across fields, but they're all aligned with the validation rules you'd normally want to set for that type of data. It helps make setting up field validation an intuitive and smooth process.

The "Presentation" tab gives you control over how the field appears to content editors. In some cases, it also gives you more control over how the data will be displayed once the field is rendered.

If we open the "Presentation" tab for the "Contributor Name" field, we can see that it allows us to give our content editors instructions for filling out the field, insert placeholder text, prepend and append text, and define wrapper attributes.

The “Contributor Name” field we created above will work just fine without any of these, but they can be very useful in certain situations. For example, say our “Contributor Info” field group is for a site where all the contributors are medical doctors. We could add “Dr.” to the “Prepend” section, and provide some instructions:



Instructions
Please insert your name in the space provided.
Instructions for content editors. Shown when submitting data.

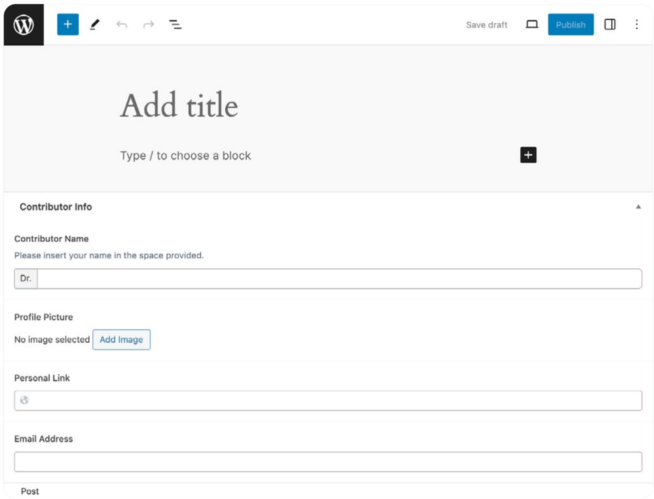
Placeholder Text
Appears within the input

Prepend
Dr.
Appears before the input

Append
Appears after the input

Wrapper Attributes

Now, when our content editor physicians are writing their posts, they’ll see this:



Contributor Name
Please insert your name in the space provided.
Dr.

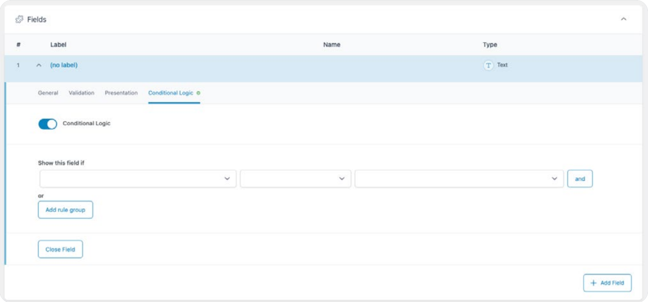
Profile Picture
No image selected

Personal Link

Email Address

The “Conditional Logic” tab allows you to set rules regarding when a custom field should appear, based on the values of other fields in the same field group. At first, the tab will only have a single toggle, allowing you to start setting rules.

Each conditional logic rule consists of three dropdown menus. Clicking the first dropdown menu displays a list of the fields currently in the field group. The second dropdown defines the condition that must be met. The third dropdown further defines these conditions with specific values.



Conditional Logic

Show this field if

Rendering fields

ACF gives you a way to structure data and model content that goes beyond what the default WordPress settings offer. It's up to you exactly how you *show* that content. When you create an ACF field group, the information is stored in the WordPress database. The specific tables used to store this information are `wp_posts` and `wp_postmeta.`

While ACF stores its data in WordPress tables, WordPress does not automatically render ACF fields and field groups. To understand why, we need to discuss how the WordPress core software uses theme files and PHP templates to display content entered into the backend of the site.

Structure of a WordPress theme

WordPress themes consist of various files that work together to create the design and functionality of a website. Themes always include templates, and usually include at least one stylesheet.

Templates: PHP files that generate different parts of the site, such as headers, footers, sidebars, and content areas. These files contain a mix of PHP, HTML, and CSS code that work together to create the final output.

Stylesheets: CSS files that control the visual design and aesthetics of the site.

WordPress themes may also include other files, such as JavaScript files and images. The `functions.php` file is technically optional, but it's almost always present. It's a template in the sense that it is used to generate content and layout. However, instead of directly outputting HTML as template files like `header.php` and `footer.php` would, it provides a set of reusable functions that can be called from other template files. Within `functions.php`, you can define your own functions, add theme support for features like featured images and post formats, and register locations for navigation menus.

Template hierarchy and the Loop

WordPress uses a template hierarchy to decide which template to use based on the type of content being displayed, such as `index.php` for the homepage, `archive.php` for archives, and `single.php` for single posts. Within these template files, the Loop is used to retrieve and display posts. The Loop is a PHP code block that checks if there are posts to display, and if so, it iterates over them using a `while` loop.

The Loop retrieves and displays content, acting as a container that holds the content and allows it to be displayed in a specific format. Think of the Loop like a conveyor belt that brings in content from the database and presents it to the user.

Child theme or custom plugin

To render the data entered into ACF fields, WordPress needs customized templates that tell it exactly where to find the data and how to display it. We do not recommend making changes directly to your theme files. There's a possibility that you'll break your theme, but you may also find all your changes overwritten when the theme is updated.

Instead, we recommend creating a child theme or creating a custom plugin to hold your files. The instructions for doing this are outside of the scope of this guide, but a quick web search should turn up any number of guides.

The next step is to identify the specific template files where you want to display the ACF fields, for example, `single.php.` This is the template the WordPress system uses to display single blog posts, news articles, etc. Once you've determined which files should be modified, you use PHP to include the ACF functions in these templates. For example, you can use `get_field()` to retrieve the value of a field and `the_field()` to display it directly.

Render with developer-friendly functions

Once you've created a field group and added fields to it, it's time to render the fields on the frontend of your WordPress site. ACF provides developer-friendly functions to retrieve and display the values of your custom fields in your theme templates.

For example, you can use the `get_field()` function to retrieve the value of `text_field` from the current post:

```
$value = get_field( "text_field" );
```

You can use the same function to retrieve the value of a specific post by adding the post ID:

```
$value = get_field( "text_field", 123 );
```

This is just the start of what you can do with this one function. For example, you can also check if a value exists for a field using the example below. Please note that any returned values should be escaped using an appropriate escaping function, such as `**wp_kses()**` before they are output.

```
$value = get_field( "text_field" );

if( $value ) {
   echo wp_kses_post( $value );
} else {
   echo 'empty';
}
```

Each ACF field has its own documentation page that shows basic usage and code examples. With over 30 fields to choose from, providing full instructions for how to render every single one is outside the scope of this guide. Instead, we'll show you how to render the fields we created for our "Contributor Info" field group.

Creating and modifying templates

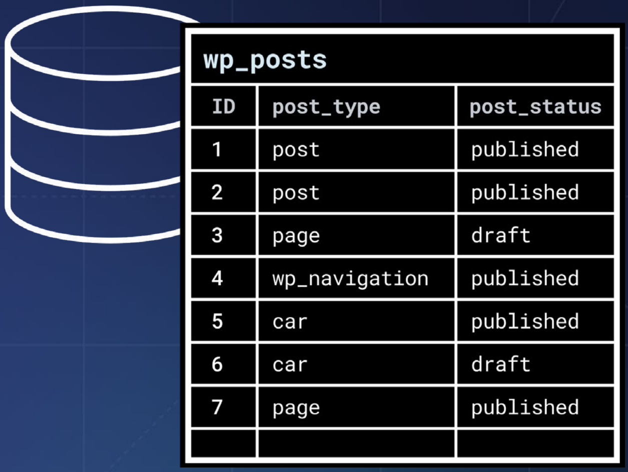
We want our “Contributor Info” fields to show up on every post, so we’re going to place the code in our child theme's `single.php` file, within the Loop:

```
      <?php
      // Start the Loop.
      while ( have_posts() ) :
        the_post();

        get_template_part( 'template-parts/post/content', get_post_format() );

        // Display the Contributor Info field group
        ?>
        <h2>Contributor Info</h2>
        <?php if ( get_field('contributor_name') ) : ?>
          <p><strong>Contributor Name:</strong> <?php the_field('contributor_name'); ?></p>
        <?php endif; ?>
        <?php if ( get_field('profile_picture') ) : ?>
          <p><strong>Profile Picture:</strong> <?php $image = get_field('profile_picture'); ?>
            <img src="<?php echo $image['url']; ?>" alt="<?php echo $image['alt']; ?>"></p>
        <?php endif; ?>
        <?php if ( get_field('personal_link') ) : ?>
          <p><strong>Personal Link:</strong> <a href="<?php the_field('personal_link'); ?>"
target="_blank">Visit Personal Link</a></p>
        <?php endif; ?>
        <?php if ( get_field('email_address') ) : ?>
          <p><strong>Email Address:</strong> <a href="mailto:<?php the_field('email_address');
?>"><?php the_field('email_address'); ?></a></p>
         <?php endif; ?>
            );

      endwhile; // End the loop.
      ?>
```



wp_posts		
ID	post_type	post_status
1	post	published
2	post	published
3	page	draft
4	wp_navigation	published
5	car	published
6	car	draft
7	page	published

Custom post types

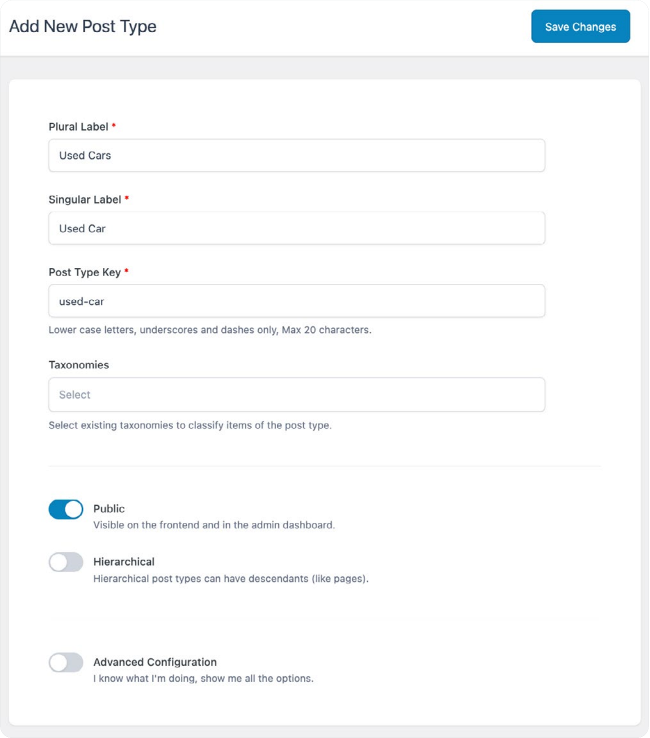
Custom post types are useful for storing data objects other than posts and pages, with WordPress providing an admin UI for managing your data and a URL structure for viewing them on the frontend of the site.

Manually coding post types into the WordPress backend is tedious and isn’t beneficial to development workflows. With ACF, you can create and manage all your custom post types directly in the plugin’s UI, as well as attach or create ACF field groups.

This chapter will guide you through the process of creating a custom post type with ACF and demonstrate how to add custom fields to enhance its functionality.

Step-by-step guide to creating a custom post type with ACF

1. Navigate to **ACF > Post Types** and click **Add New**.
2. Choose a Plural Label and Singular Label. ACF will provide a default Post Type Key, but you can modify this if you wish.
3. Select any existing **Taxonomies** you wish to use to classify the post type.
4. Leave the **Public** toggle on, allowing the post type to show in the admin and be accessible on the site’s frontend.
5. Toggle on **Hierarchical** if you want to allow parent and child relationships, similar to the WordPress “Pages” post type. This defaults to off.
6. Click **Save Changes** to create your post type.


Add New Post Type

Plural Label *
Used Cars

Singular Label *
Used Car

Post Type Key *
used-car
Lower case letters, underscores and dashes only, Max 20 characters.

Taxonomies
Select
Select existing taxonomies to classify items of the post type.

● Public
Visible on the frontend and in the admin dashboard.

○ Hierarchical
Hierarchical post types can have descendants (like pages).

○ Advanced Configuration
I know what I'm doing, show me all the options.

This will create a basic custom post type, but turning on Advanced Configuration gives you access to much deeper customization options. This gives you a wide range of options to customize the content edit screen, visibility, and behavior of your post type. You can define which features are supported, such as titles, featured images, etc., as well as add custom items to the editor. Additionally, you can control how the post type appears in the admin dashboard and frontend, including its visibility in menus and search results. You can also configure labels, URLs, and permissions to suit your requirements. Furthermore, you can integrate your custom post type with the WordPress REST API and manage its behavior in the API. These settings provide immense flexibility to create custom post types that fit your specific use cases.

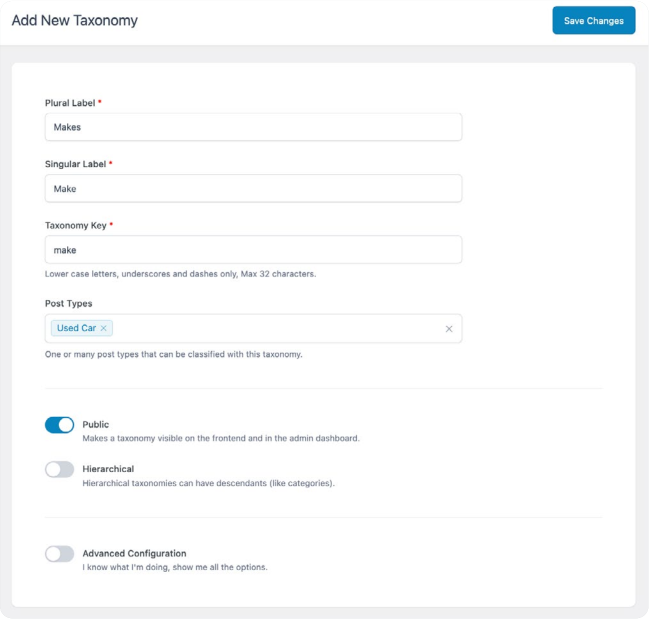
Adding custom fields to post types

Once you have configured the post type and saved your changes, a saved success notice appears along with some helpful links to perform typical next actions, such as adding an existing field group, creating a new one, or creating a custom taxonomy.

Custom taxonomies

By default, the WordPress CMS comes with two taxonomies: categories and tags. However, these might not be sufficient for certain types of websites or content. Custom taxonomies enable you to organize and structure content in a more detailed and specific manner, providing a smoother experience for both content editors and the site's end users. Manually creating a custom taxonomy uses the `register_taxonomy()` function, and fills in an array of arguments that define the taxonomy's properties and the post types it should be associated with.

Creating a custom taxonomy with ACF, on the other hand, is done entirely through the WordPress admin and takes just seconds. The basic creation process boils down to choosing the Plural and Singular Labels and associating the taxonomy with at least one post type. Not only that, but if you followed a link provided when you create a custom post type, you'll find it's already been added:


Add New Taxonomy

Plural Label * Makes

Singular Label * Make

Taxonomy Key * make

Lower case letters, underscores and dashes only, Max 32 characters.

Post Types: Used Car ×

One or many post types that can be classified with this taxonomy.

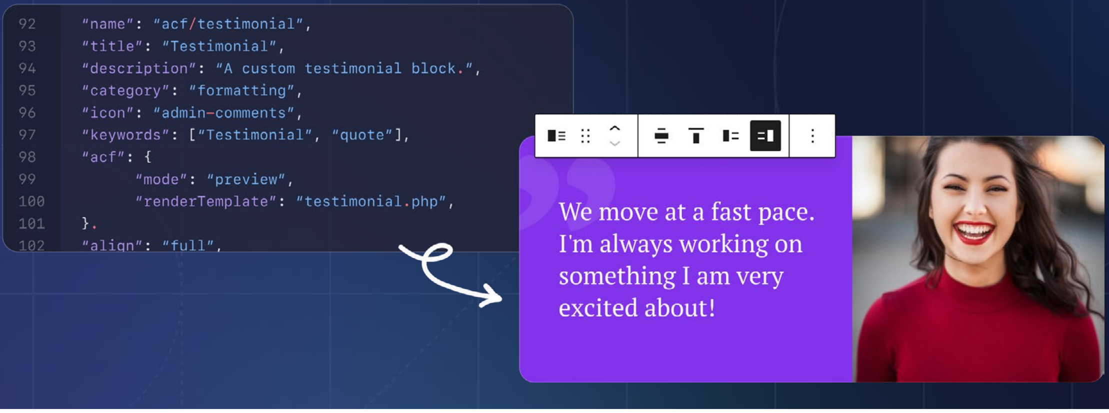
● Public
Makes a taxonomy visible on the frontend and in the admin dashboard.

○ Hierarchical
Hierarchical taxonomies can have descendants (like categories).

○ Advanced Configuration
I know what I'm doing, show me all the options.

Just like with custom post types, ACF offers you a wealth of options when you toggle on "Advanced Configuration". These settings allow for fine-grained control over the taxonomy's behavior, including labels, visibility, meta boxes, URLs, and REST API integration.

With just a few clicks, you can configure how terms are sorted, create default terms, define labels and descriptions, control visibility in the admin dashboard and frontend, and customize URL structures and query variables. Additionally, you can manage the taxonomy's presence in the REST API and define custom controllers.



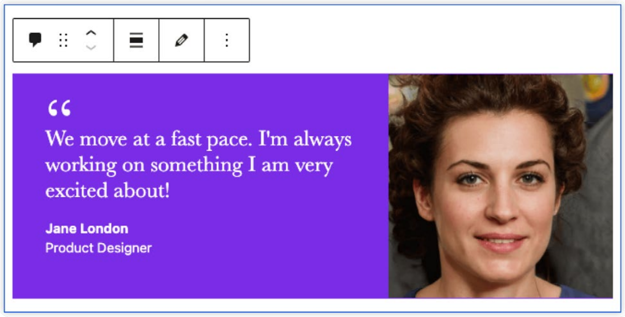
```
"name": "acf/testimonial",
"title": "Testimonial",
"description": "A custom testimonial block.",
"category": "formatting",
"icon": "admin-comments",
"keywords": ["Testimonial", "quote"],
"acf": {
      "mode": "preview",
      "renderTemplate": "testimonial.php",
},
"align": "full",
```

We move at a fast pace. I'm always working on something I am very excited about!

Level up with ACF PRO

ACF PRO is the premium version of Advanced Custom Fields, offering a range of advanced features that can significantly enhance the functionality of your WordPress sites with more complex and dynamic content structures. In this chapter, we'll explore the additional fields and features available in ACF PRO.

ACF Blocks: Introduces a new way of creating custom blocks for the WordPress block editor, with features like InnerBlocks, Variations, and Styles.



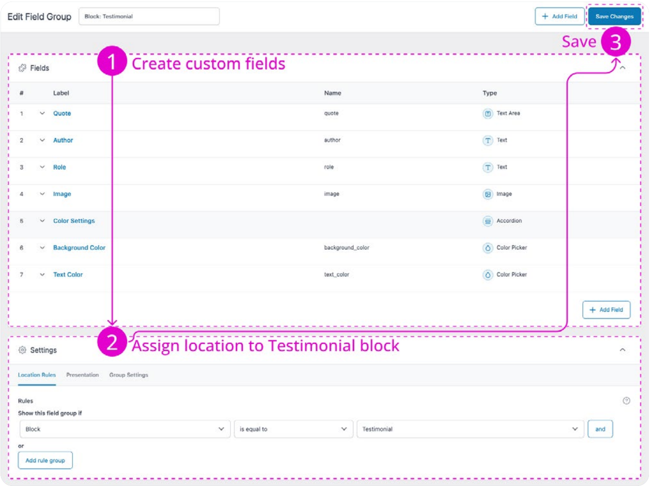
We move at a fast pace. I'm always working on something I am very excited about!

Jane London
Product Designer

ACF Blocks, such as the testimonial block shown here, appear in the WordPress block inserter the same way core blocks do.

ACF Blocks is a game-changer for WordPress developers, allowing you to create custom block types without needing extensive JavaScript or React knowledge. With ACF Blocks, you can build complex content structures and custom page templates using PHP templates, making it easier to manage and display dynamic content. This feature integrates seamlessly with ACF fields, giving you access to a wide range of field types and options for creating custom block content.

By using ACF Blocks, you can enhance the block editor experience, create flexible and customizable content structures, and streamline your development workflow. Whether you're building custom page templates, product showcases, or event listings, ACF Blocks provides the tools you need to create engaging and dynamic content. With its ease of use, flexibility, and customization options, ACF Blocks is an essential tool for any WordPress developer looking to take their content creation to the next level.



Edit Field Group: Block: Testimonial

Fields

#	Label	Name	Type
1	Quote	quote	Text Area
2	Author	author	Text
3	Role	role	Text
4	Image	image	Image
5	Color Settings		Accordion
6	Background Color	background_color	Color Picker
7	Text Color	text_color	Color Picker

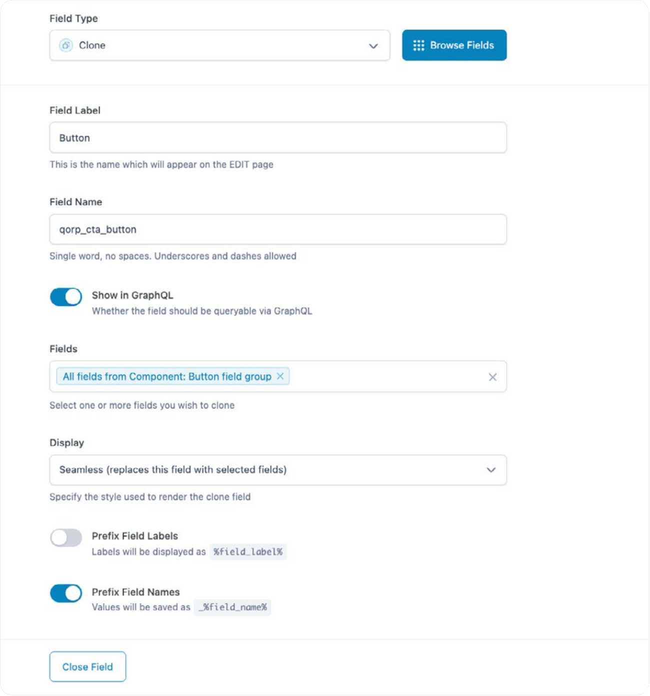
Settings

Rules
Show this field group if
Block | is equal to | Testimonial | and
or
Add rule group

The three steps required when creating and assigning field groups to ACF Blocks.

Clone field: Allows you to duplicate a field group and its contents, making it easy to create similar field groups with minimal effort.

The Clone field in ACF PRO is a valuable tool for streamlining your workflow, reducing data redundancy, and increasing the efficiency of your site development. The Clone field is particularly useful for creating reusable components like buttons, complex field structures, and custom post types. By breaking down complex structures into smaller, reusable components, you can easily manage and maintain large field groups. This feature also helps avoid data duplication and ensures that changes made to the original field group are reflected in all cloned instances.



Field Type: Clone

Field Label: Button
This is the name which will appear on the EDIT page

Field Name: qorp_cta_button
Single word, no spaces. Underscores and dashes allowed

Show in GraphQL: on
Whether the field should be queryable via GraphQL

Fields: All fields from Component: Button field group
Select one or more fields you wish to clone

Display: Seamless (replaces this field with selected fields)
Specify the style used to render the clone field

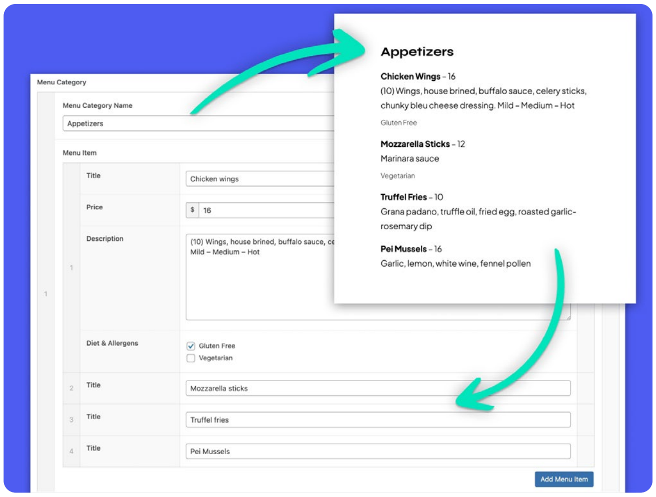
Prefix Field Labels: off
Labels will be displayed as %field_label%

Prefix Field Names: on
Values will be saved as _%field_name%

The Clone field excels at creating reusable components, allowing you to pull in existing field groups.

Repeater field: This enables you to create a set of fields that can be repeated as many times as needed, ideal for creating lists or collections of data.

The Repeater field serves as a wrapper for a theoretically unlimited number of subfields, giving you incredible versatility when building WordPress sites. Any of the 30 fields currently available can be used in a Repeater field.



Menu Category

Menu Category Name: Appetizers

Menu Item

	Field	Value
1	Title	Chicken wings
	Price	$ 16
	Description	(10) Wings, house brined, buffalo sauce, ce Mild – Medium – Hot
	Diet & Allergens	☑ Gluten Free ☐ Vegetarian
2	Title	Mozzarella sticks
3	Title	Truffel fries
4	Title	Pei Mussels

Appetizers

Chicken Wings – 16
(10) Wings, house brined, buffalo sauce, celery sticks, chunky bleu cheese dressing. Mild – Medium – Hot
Gluten Free

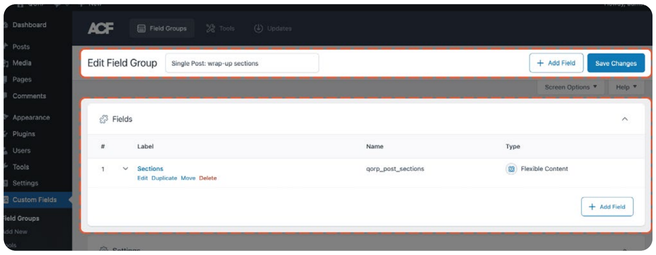
Mozzarella Sticks – 12
Marinara sauce
Vegetarian

Truffel Fries – 10
Grana padano, truffle oil, fried egg, roasted garlic-rosemary dip

Pei Mussels – 16
Garlic, lemon, white wine, fennel pollen

The only limits on the number of repeats are the ones defined by you in the field settings. However, there are performance implications on the edit screen (WP Admin) if a large number of Repeater rows are created. For Repeater fields with a very large number of rows that aren't edited often, we recommend enabling pagination on your Repeater field to curb any performance impact.

Flexible Content field: Provides a flexible way to create custom layouts and content structures, using a combination of fields and layouts.



Edit Field Group: Single Post: wrap-up sections

Fields

#	Label	Name	Type
1	Sections	qorp_post_sections	Flexible Content

Creating multiple layouts with the Flexible Content field.

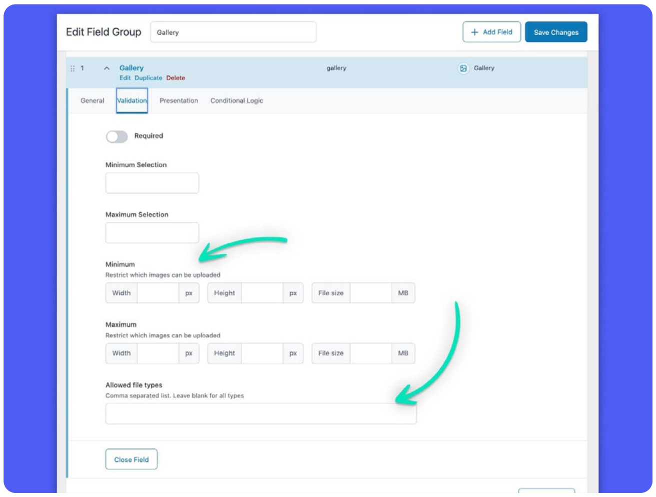
The ACF Flexible Content field is a powerful tool for creating dynamic, modular content structures. It acts as a blank canvas to which you can add an unlimited number of layouts with full control over the order. This field type is particularly useful for creating unique page content, building custom email templates, and managing complex data structures.

The Flexible Content field is highly versatile and can be used in a variety of scenarios. For example, you can create different layouts for different content types, such as project overviews and client feedback, and then populate these layouts dynamically using ACF and other tools like Bricks Builder. Additionally, you can apply styling and conditional logic to make your layouts more interactive and user-friendly.

Gallery field: Allows you to create a gallery of images, with options for customizing the display and behavior of the gallery.

The Gallery field allows you to create and manage image galleries, providing a user-friendly interface for uploading, organizing, and displaying images. The Gallery field supports a range of features, including image captions, ordering, and cropping, giving you full control over the display and layout of the uploaded images.

The Gallery field is particularly useful for creating visually appealing content structures like photo galleries, product showcases, and portfolios. It's also great for managing large collections of images, such as event photos or product catalogs. By using the Gallery field, you can create engaging and interactive image displays that enhance the user experience and showcase your visual content in a compelling way.

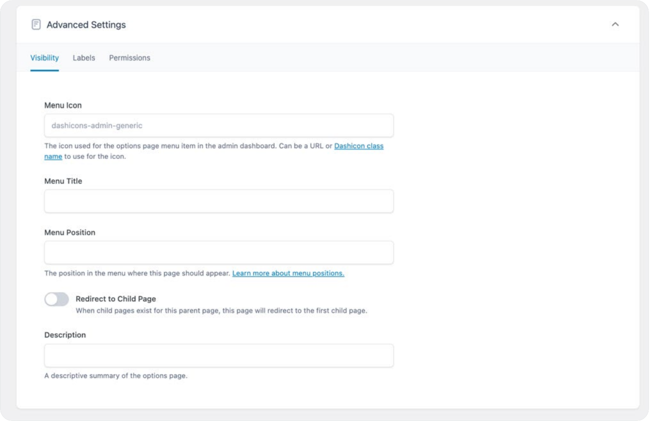


The Gallery field makes it simple for content editors to upload multiple images at once, but it also gives you control over which images can be uploaded, with settings to validate file types, image size, and number of images

Options Pages: This feature enables you to create custom options pages for your website, making it easy to manage global settings and options.

ACF PRO's Options Pages feature allows you to create custom options pages within the WordPress admin area, providing a centralized location for managing site-wide settings and configurations. This feature enables you to create custom fields and groups that can be used to store and manage data, such as site settings, social media links, or SEO metadata. Options Pages can be organized into tabs and sections, making it easy to categorize and access related settings.

The Options Pages feature is particularly useful for creating a unified settings area for your site, allowing administrators to easily manage and update site-wide configurations. It's also great for creating custom settings for plugins and themes, or for managing data that needs to be accessed across multiple pages or posts. By using Options Pages, you can streamline your site's administration and provide a more intuitive experience for users.



Menu Icon

dashicons-admin-generic

The icon used for the options page menu item in the admin dashboard. Can be a URL or Dashicon class name to use for the icon.

Menu Title

Menu Position

The position in the menu where this page should appear. Learn more about menu positions.

Redirect to Child Page

When child pages exist for this parent page, this page will redirect to the first child page.

Description

A descriptive summary of the options page.

The Advanced Settings in ACF PRO's Options Page feature allows you to control menu item icon, its position, and much more.

How ACF PRO improves your workflow

With a wide range of features and functionality, ACF PRO is an indispensable tool that can significantly enhance the workflow for both developers and content creators.

Whether you're building a complex site from scratch or managing an extensive catalog of content, ACF PRO can make your life easier in several key ways:

Increased Flexibility: ACF PRO's advanced fields and features offer unparalleled flexibility when creating custom content structures and layouts. This flexibility allows you to tailor WordPress sites precisely to your needs, ensuring you can easily adapt to any project requirements.

Improved Efficiency: Efficiency is at the heart of ACF PRO. By reducing the amount of code you need to write, ACF PRO streamlines your development process. Its intuitive interface and powerful features make it easier to manage complex content structures, saving you time and effort. This efficiency boost means you can focus more on creativity and less on the mundane aspects of coding, leading to faster project completion and higher productivity.

Enhanced User Experience: Creating a seamless user experience is crucial, and ACF PRO excels in this area. Its features allow you to develop more intuitive and user-friendly interfaces for managing content. This not only improves the end-user experience but also makes it easier for clients or content managers to update and maintain the website. The result is a smoother, more enjoyable interaction with the site, for both administrators and visitors.

Whether you're a developer looking to streamline your workflow or a content manager aiming to improve site usability, ACF PRO has the features you need to succeed.

Next steps

Congratulations on completing this comprehensive guide to getting started with ACF! By now, you should have a solid understanding of how to create and manage custom fields, model content, and leverage ACF's powerful features to enhance your WordPress development workflow.

Tips for planning and organizing custom fields, post types, and taxonomies

As you continue to work with ACF, it's essential to plan and organize your custom field data carefully. Here are a few tips to keep in mind:

- ✓ Start by identifying the types of content you need to create and the fields required to support that content.
- ✓ Use ACF's field group and field type system to organize your fields in a logical and structured way.
- ✓ Use ACF's built-in features, such as conditional logic and field dependencies, to create a more dynamic and user-friendly editing experience.
- ✓ Use ACF's import and export features to manage and version your field data across different environments.

Resources for further learning and support

ACF offers a wide range of resources to help you get the most out of its features and functionality. Here are some key resources to keep in mind:

- ACF's official documentation and FAQs
- ACF's community forums and support ticket system
- ACF's debug and troubleshooting guides
- ACF's blog and release notes, which provide insights into new features and improvements
- ACF’s tutorials, guides, and video walkthroughs

By leveraging these resources and continuing to explore ACF's features and functionality, you'll be well on your way to becoming an ACF expert and creating custom, data-driven content that engages and inspires your audience.

What’s next?

Now that you've completed this guide, it's time to take your ACF skills to the next level. Here are some next steps to consider:

- ✓ Experiment with ACF's more advanced features, such as its API and hooks system
- ✓ Create a custom ACF-powered project, such as a page-builder experience for clients or customized ACF Blocks
- ✓ Share your ACF knowledge with others, through tutorials, blog posts, or community forums

ACF is a powerful tool that requires practice and experimentation to master. By continuing to learn and explore its features, you'll be able to create custom, data-driven content that sets you apart from the competition.

WP engine™

WP Engine empowers companies and agencies of all sizes to build, power, manage, and optimize their WordPress websites and applications with confidence.

Serving 1.5 million customers across 150+ countries, the global technology company provides premium, enterprise-grade solutions, tools, and services, including specialized platforms for WordPress, industry-tailored eCommerce and agency solution suites, and developer-centric tools like Local, Advanced Custom Fields, and more. WP Engine's innovative technology and industry-leading expertise are why 8% of the web visits a WP Engine-powered site daily. Learn more at wpengine.com.

WP Engine is a proud member and supporter of the community of WordPress® users. The WordPress® trademarks are the intellectual property of the WordPress Foundation, and the Woo® and WooCommerce® trademarks are the intellectual property of WooCommerce, Inc. Uses of the WordPress®, Woo®, and WooCommerce® trademarks in this website are for identification purposes only and do not imply an endorsement by WordPress Foundation or WooCommerce, Inc. WP Engine is not endorsed or owned by, or affiliated with, the WordPress Foundation or WooCommerce, Inc.