File Upload

- Screenshots
- Settings
- Data
- Template usage
- Filters
- Binding events

import File from '../_parts/_file.md'; import Screenshots from '@site/src/components/Screenshots';

The file upload field displays an inline upload area that you can drag and drop or select files to upload. It doesn't open the Media Library popup for selecting existing files. You can upload new files only.

Screenshots

<Screenshots name="file-upload" col1={[['https://imgur.elightup.com/yGNNhOg.png', 'The file upload field interface'],]} />

Settings

Besides the common settings, this field has the following specific settings, the keys are for use with code:

Name	Key	Description
Max number of files	`max_file_uploads`	Max number of uploaded files. Optional.

Name	Key	Description
Force delete	`force_delete`	Whether or not delete the files from Media Library when deleting them from post meta. `true` or `false` (default). Optional. Note: it might affect other posts if you use the same file for multiple posts.
MIME types	`mime_type`	MIME type of files which we want to show in the Media Library. Note: this is a filter for items in the media popup, it doesn't restrict file types when uploading.
Show status	`max_status`	Display how many files uploaded/remaining. Applied only when "Max number of files" is defined. `true` (default) or `false`. Optional.
Max file size	`max_file_size`	Maximum file size that the user can upload, in bytes. Optionally supports b, kb, mb, gb, tb suffixes. e.g. "10mb" or "1gb".

This is a sample field settings array when creating this field with code:

```
[
    'id'               => 'file',
    'name'             => 'File upload',
    'type'             => 'file_upload',
    'force_delete'     => false,
    'max_file_uploads' => 2,
    'mime_type'        => 'application,audio,video',
    'max_status'       => false,
],
```

Data

This field saves multiple attachment IDs in the database. Each value (attachment ID) is stored in a single row in the database with the same meta key (similar to what `add_post_meta` does with the last parameter `false`).

Template usage

Filters

This field has some filters to change the texts displayed on the screen..

Filter	Default	Description
`rwmb_media_add_string`	+ Add Media	Add new file string
`rwmb_media_single_files_string`	file	Singular "file" string
`rwmb_media_multiple_files_string`	files	Plural "files" string
`rwmb_media_remove_string`	Remove	File remove string
`rwmb_media_edit_string`	Edit	File edit string
`rwmb_media_view_string`	View	File view string
`rwmb_media_select_string`	Select Files	Select files string
`rwmb_media_or_string`	or	The string "or" in "Drop files here to upload or Select Files"
`rwmb_media_upload_instructions_string`	Drop files here to upload	The upload instruction string

The code below changes the "+ Add Media" string:

```
add_filter( 'rwmb_media_add_string', function () {
    return '+ New File';
} )
```

Binding events

You can bind various events during uploading via the `data-uploader` of the input like this:

```
jQuery( function( $ ){
    setTimeout( function() {
        const myUploader =  $( 'input.rwmb-file_upload.rwmb-media' ).data( 'uploader' );

        myUploader.uploader.bind( 'FileUploaded', function( up, file, res ) {
            console.log( 'File Uploaded' );
```

```
        } );
    }, 1000 );
} );
```

:::warning

You need to use `setTimeout` to make sure your code runs after the uploader is initialized.

:::

Here is the list of events of the uploader. You can see more details about them in the official documentation of plupload, the library WordPress and Meta Box use for uploading files.

Event|Description ---|---|--- `Init`|Fires when the current uploader has been initialized. `PostInit`|Fires after the init event in case you need to perform actions there. `Refresh`|Fires when the silverlight/flash or other shim needs to move. `Browse`|Fires when browse_button is clicked and browse dialog shows. `BeforeUpload`|Fires just before a file is uploaded. Can be used to cancel the upload for the specified file by returning false from the handler. `UploadFile`|Fires when a file is to be uploaded by the runtime. `UploadProgress`|Fires while a file is being uploaded. Use this event to update the current file upload progress. `BeforeChunkUpload`|Fires just before a chunk is uploaded. This event enables you to override settings on the uploader instance before the chunk is uploaded. `ChunkUploaded`|Fires when a file chunk is uploaded. `FileUploaded`|Fires when a file is successfully uploaded. `UploadComplete`|Fires when all files in a queue are uploaded. `Error`|Fires when an error occurs. `Destroy`|Fires when destroy method is called.

□□□ #2
□ □□□ □□□ 2026-08-29 15:36:17 UTC
□ □□□ □□□ 2026-08-29 15:36:17 UTC