MB User Profile

- Registration form
- Login form
- Edit profile form
  - Default fields
  - User password
- Settings
- Email templates
- Hooks
  - rwmb_profile_redirect
  - rwmb_profile_before_process
  - rwmb_profile_after_process
  - rwmb_profile_before_form
  - rwmb_profile_after_form
  - rwmb_profile_before_display_confirmation
  - rwmb_profile_after_display_confirmation
  - rwmb_profile_before_submit_button
  - rwmb_profile_validate
  - rwmb_profile_insert_user_data
  - rwmb_profile_update_user_data
  - Default form fields
  - rwmb_profile_before_save_user
  - rwmb_profile_after_save_user
- Known issues
  - Upload files/images
  - Styling
  - Caching

MB User Profile helps you to add information to the user profile and edit these details on the front end. It also includes a register form, login form, and edit profile form that you can embed anywhere

using shortcodes.

user profile page

Note: this extension already includes **MB User Meta**.

Registration form

To insert a registration form to a page, select the **Registration Form** block from the block inserter. And you'll see the form preview like this:

Registration form block

Clicking on the block will reveal the form settings on the right. All settings are self-explained and have their description.

You can also use the registration shortcode to display the registration form:

```
[mb_user_profile_register]
```

Below is the list of the settings. The attribute is used for the shortcode.

Name	Attribute	Description
ID	`id`	Field group ID(s) created for users, separated by commas. Optional. Leave blank to show the default registration form.
Redirect URL	`redirect`	Redirect URL, to which users will be redirected after a successful registration.
Form ID	`form_id`	ID (HTML attribute) of the form.
Title	`label_title`	Label for the title of the form. Default empty.
Username field label	`label_username`	Label for the username input field.
Email field label	`label_email`	Label for the email input field.
Password field label	`label_password`	Label for the password input field.
Confirm password field label	`label_password2`	Label for the confirm password input field.
Submit button text	`label_submit`	Label for the submit button.

Name	Attribute	Description
Username field ID	`id_username`	ID (HTML attribute) of the username input field.
Email field ID	`id_email`	ID (HTML attribute) of the email input field.
Password field ID	`id_password`	ID (HTML attribute) of the password input field.
Confirm password field ID	`id_password2`	ID (HTML attribute) of the confirm password input field.
Submit button ID	`id_submit`	ID (HTML attribute) of the submit button.
Confirmation text	`confirmation`	Confirmation message if registration is successful.
Send confirmation email	`email_confirmation`	Send a confirmation email when registering (you need to set up an email SMTP to make this function work properly). If this param is `true`, the system will also check the confirmation status when the user login. All previous users are set and confirmed.
Password strength	`password_strength`	Set the required password strength. Available options: `strong`, `medium`, `weak`, `very-weak`, or `false` to disable password strength meter.
Use email for username	`email_as_username`	Use email for username. If this param is `true`, then the username field will disappear.
Always show	`show_if_user_can`	Always show the form if the current user has the proper capability. Should be a WordPress capability. Useful if admins want to register for other people.
Role for the new user	`role`	Role for the new user. If `append_role` is set to `true`, then the new role is appended, so users will have 2 roles: the default roles set by WordPress and this role. Default empty.
Append role	`append_role`	Whether to append the role to users instead of setting only one role for users.
reCaptcha key	`recaptcha_key`	Google reCaptcha site key (version 3). Optional.
reCaptcha secret	`recaptcha_secret`	Google reCaptcha secret key (version 3). Optional.

Login form

To insert a login form to a page, select the **Login Form** block from the block inserter. And you'll see the form preview like this:

Login form block

Clicking on the block will reveal the form settings on the right. All settings are self-explained and have their description.

You can also use the login shortcode to display the login form:

```
[mb_user_profile_login]
```

Below is the list of the settings. The attribute is used for the shortcode.

Name	Attribute	Description
Redirect URL	`redirect`	Redirect URL, to which users will be redirected after successful login.
Form ID	`form_id`	ID (HTML attribute) of the form.
Title	`label_title`	Label for the title of the form. Default empty.
Username field label	`label_username`	Label for the username input field.
Password field label	`label_password`	Label for the password input field.
Remember checkbox label	`label_remember`	Label for the remember checkbox field.
Lost password field label	`label_lost_password`	Label for the lost password link.
Submit button text	`label_submit`	Label for the submit button.
Username field ID	`id_username`	ID (HTML attribute) of the username input field.
Password field ID	`id_password`	ID (HTML attribute) of the password input field.
Remember checkbox ID	`id_remember`	ID (HTML attribute) of the remember checkbox field.
Submit button ID	`id_submit`	ID (HTML attribute) of the submit button.
Confirmation text	`confirmation`	Confirmation message if login is successful.
Default username value	`value_username`	Default value for the username field.

Name	Attribute	Description
Default remember	`value_remember`	Default value for the remember checkbox field - `true` or `false` (default).
reCaptcha key	`recaptcha_key`	Google reCaptcha site key (version 3). Optional.
reCaptcha secret	`recaptcha_secret`	Google reCaptcha secret key (version 3). Optional.

Edit profile form

To insert an edit profile form to a page, select the **Edit Profile Form** block from the block inserter. And you'll see the form preview like this:

Edit profile form block

Clicking on the block will reveal the form settings on the right. All settings are self-explained and have their description.

You can also use the edit profile shortcode to display the edit profile form:

```
[mb_user_profile_info id="field-group-id"]
```

Below is the list of the settings. The attribute is used for the shortcode.

Name	Attribute	Description
ID	`id`	Field group ID(s) created for users, separated by commas. All fields from field groups will be included in the profile form. Required.
User ID	`user_id`	User ID, whose info will be edited. If not specified, the current user ID is used.
Redirect URL	`redirect`	Redirect URL, to which users will be redirected after successful submission.
Form ID	`form_id`	ID (HTML attribute) of the form.
Title	`label_title`	Label for the title of the form. Default empty.
Password field label	`label_password`	Label for the password input field.

Name	Attribute	Description
Confirm password field label	`label_password2`	Label for the confirm password input field.
Submit button text	`label_submit`	Label for the submit button.
Password field ID	`id_password`	ID (HTML attribute) of the password input field.
Confirm password field ID	`id_password2`	ID (HTML attribute) of the confirm password input field.
Submit button ID	`id_submit`	ID (HTML attribute) of the submit button.
Confirmation text	`confirmation`	Confirmation message if the form submission is successful.
Password strength	`password_strength`	Set the required password strength. Available options: `strong`, `medium`, `weak`, `very-weak`, or `false` to disable password strength meter.
reCaptcha key	`recaptcha_key`	Google reCaptcha site key (version 3). Optional.
reCaptcha secret	`recaptcha_secret`	Google reCaptcha secret key (version 3). Optional.

Default fields

By default, the user profile form doesn't include any default user fields, such as first name, last name, or biography. To be able to edit these fields, please create a field group for users with those fields, keeping the same field IDs as WordPress's.

For example, the code below creates a meta box for editing the user's first name, last name, and biography:

```
add_filter( 'rwmb_meta_boxes', function( $meta_boxes ) {
    $meta_boxes[] = [
        'title'  => 'Default Fields',
        'id'     => 'default-fields',
        'type'   => 'user', // NOTICE THIS
        'fields' => [
            [
                // highlight-next-line
                'id'   => 'first_name',
                'name' => 'First Name',
```

```
                'type' => 'text',
            ],
            [
                // highlight-next-line
                'id'   => 'last_name',
                'name' => 'Last Name',
                'type' => 'text',
            ],
            [
                // highlight-next-line
                'id'   => 'description',
                'name' => 'Biography',
                'type' => 'textarea',
            ],
        ],
    ];
    return $meta_boxes;
} );
```

And use it in the user edit profile form with this shortcode:

```
[mb_user_profile_info id="default-fields"]
```

To make the plugin recognize the user fields, you need to set the correct ID for them:

Field	ID
User email	`user_email`
User nicename	`user_nicename`
User URL	`user_url`
Display name	`display_name`
First name	`first_name`
Last name	`last_name`
Registration date	`user_registered`
Biography	`description`
Rich editing	`rich_editing`
Syntax highlighting	`syntax_highlighting`
Admin color	`admin_color`
Comment shortcuts	`comment_shortcuts`

Field	ID
Show the admin bar on the front	`admin_bar_front`
User role	`role`

:::warning

These default fields do not work as sub-fields in a group.

:::

User password

To let users change their password, please use the field group ID `rwmb-user-info` in the shortcode below:

```
[mb_user_profile_info id="rwmb-user-info"]
```

Settings

MB User Profile has a settings page under **Meta Box > User Profile** that allows you to change some settings for the plugin.

User profile settings

- **Force password change**: force users to change their password after the first login.
- **Email confirmation success page**: select a page to show when users successfully confirm their email. If no page is selected, a default message will be displayed.
- **Email confirmation error page**: select a page to show when there are errors when users confirming their email. If no page is selected, a default message will be displayed.

Email templates

MB User Profile sends emails to users when:

- They reset their passwords
- They register and are requested to confirm by email

To overwrite the emails created by the plugin, please:

- Create a folder `mb-user-profile` in your theme (or your child theme)

- Copy the email templates from the MB User Profile's `templates` folder to the `mb-user-profile` folder that you've just created above
- Edit the new templates files

Now the emails sent by the plugin will use the templates in your theme.

Hooks

`rwmb_profile_redirect`

This filter allows you to change the URL of the redirect page after the form is submitted. It accepts 2 parameters: redirect URL and form config - the shortcode attributes.

```
$redirect = apply_filters( 'rwmb_profile_redirect', $redirect, $config );
```

You can use this filter to change the redirect URL after a user updated his profile.

```
add_filter( 'rwmb_profile_redirect', function( $redirect, $config ) {
    if ( 'my-meta-box' === $config['id'] ) {
        $redirect = 'https://domain.com/thank-you/';
    }
    return $redirect;
}, 10, 2 );
```

`rwmb_profile_before_process`

This action fires before the form is processed. It accepts one parameter `$config` - the form configuration, taken from the shortcode attributes.

`rwmb_profile_after_process`

This action fires after the form is processed, e.g. saved or updated. It accepts 2 parameters:

- `$config` - the form configuration, taken from the shortcode attributes.
- `$user_id` - the submitted user ID

You can use this action to do a custom redirect to your Thank you page or send an email notification.

```
add_action( 'rwmb_profile_after_process', function( $config, $user_id ) {
    if ( 'my-meta-box' === $config['id'] ) {
        wp_mail( 'admin@domain.com', 'New submission', 'A new user has been just submitted.'
);

        wp_safe_redirect( 'thank-you' );
        die;
    }
}, 10, 2 );
```

`rwmb_profile_before_form`

This action fires before form output. It accepts one parameter `$config` - the form configuration, taken from the shortcode attributes.

`rwmb_profile_after_form`

This action fires after form output. It accepts one parameter `$config` - the form configuration, taken from the shortcode attributes.

`rwmb_profile_before_display_confirmation`

This action fires before the confirmation message is displayed. It accepts one parameter `$config` - the form configuration, taken from the shortcode attributes.

`rwmb_profile_after_display_confirmation`

This action fires after the confirmation message is displayed. It accepts one parameter `$config` - the form configuration, taken from the shortcode attributes.

`rwmb_profile_before_submit_button`

This action fires before the submit button is displayed. It accepts one parameter `$config` - the form configuration, taken from the shortcode attributes.

`rwmb_profile_after_submit_button`

This action fires after the submit button is displayed. It accepts one parameter `$config` - the form configuration, taken from the shortcode attributes.

rwmb_profile_validate

This filter is used to check if the form is validated. You can use this filter to add a custom check for the data before it's processed.

If you return any value other than `true`, the form will not be processed and the error message will be displayed. So, you can return a string as an error message.

```
$is_valid = apply_filters( 'rwmb_profile_validate', $is_valid, $config );
```

Example:

```
add_filter( 'rwmb_profile_validate', function ($is_valid) {
	$email = $_POST['user_email'] ?? '';

	if ( $email === 'alex@example.com' ) {
		$is_valid = 'Email is already taken';
	}

	return $is_valid;
} );
```

rwmb_profile_insert_user_data

This filter is used to modify the submitted user data before it's passed to the `wp_insert_user` function to **create a new user**. It accepts 2 parameters: the array of user data and the shortcode parameters.

```
$data = apply_filters( 'rwmb_profile_insert_user_data', $data, $config );
```

Please note that this filter works only for default user fields such as user email or password. For changing custom field data, please use `rwmb_{$field_id}_value` filter.

rwmb_profile_update_user_data

This filter is used to modify the submitted user data before it's passed to the `wp_update_user` function to **update an existing user**. It accepts 2 parameters: the array of user data and the shortcode parameters.

```
$data = apply_filters( 'rwmb_profile_update_user_data', $data, $config );
```

Please note that this filter works only for default user fields such as user email or password. For changing custom field data, please use `rwmb_{$field_id}_value` filter.

Default form fields

To modify the default register, login, or edit profile form fields, please use the following filters:

```
$fields = apply_filters( 'rwmb_profile_register_fields', $fields );
$fields = apply_filters( 'rwmb_profile_login_fields', $fields );
$fields = apply_filters( 'rwmb_profile_info_fields', $fields );
```

`rwmb_profile_before_save_user`

This action fires before the user is created or updated.

```
do_action( 'rwmb_profile_before_save_user', $object );
```

The action accepts 1 parameter: the instance of the `MB_User_Profile_User` class, which handles the submission. It has the following public properties that you can use:

- `$user_id`: The submitted user ID
- `$config`: The configuration, taken from the shortcode attributes

`rwmb_profile_after_save_user`

This action fires after the user is created or updated. At that time, all the custom fields in the meta box are already saved into the user meta.

```
do_action( 'rwmb_profile_after_save_user', $object );
```

The action accepts 1 parameter: the instance of the `MB_User_Profile_User` class, which handles the submission. It has the following public properties that you can use:

- `$user_id`: The submitted user ID
- `$config`: The configuration, taken from the shortcode attributes

Known issues

Upload files/images

To be able to upload files or images (via fields `file_advanced`, `file_upload`, `image_advanced`, `image_upload`), users have to log in and proper capability upload_files to access the Media Library. If your users don't have that capability (like if they have the subscriber role), then the upload fields don't work. In that case, you can add the capability for that role as follows:

```
add_action( 'init', function () {
    $subscriber = get_role( 'subscriber' );
    $subscriber->add_cap( 'upload_files' );
} );
```

Another solution is using `file` or `image` fields. Both of them work similarly. They just don't have a nice UI, but they do the job very well.

Styling

The extension outputs the default fields' HTML with CSS comes from the Meta Box plugin. The style is basic and works in most cases. However, to make it match perfectly with your theme's style, you need to do some work with CSS.

Caching

As you might know, Meta Box uses nonces to prevent misuse or malicious requests. As the nonce fields are outputted directly in the HTML, they might be cached by caching plugins such as W3 Total Cache, WP Super Cache, etc. And thus, the verification of the nonce might not work properly and break the form submission. In this case, please do not cache the page where the form is embedded (caching plugins allow you to do that).

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