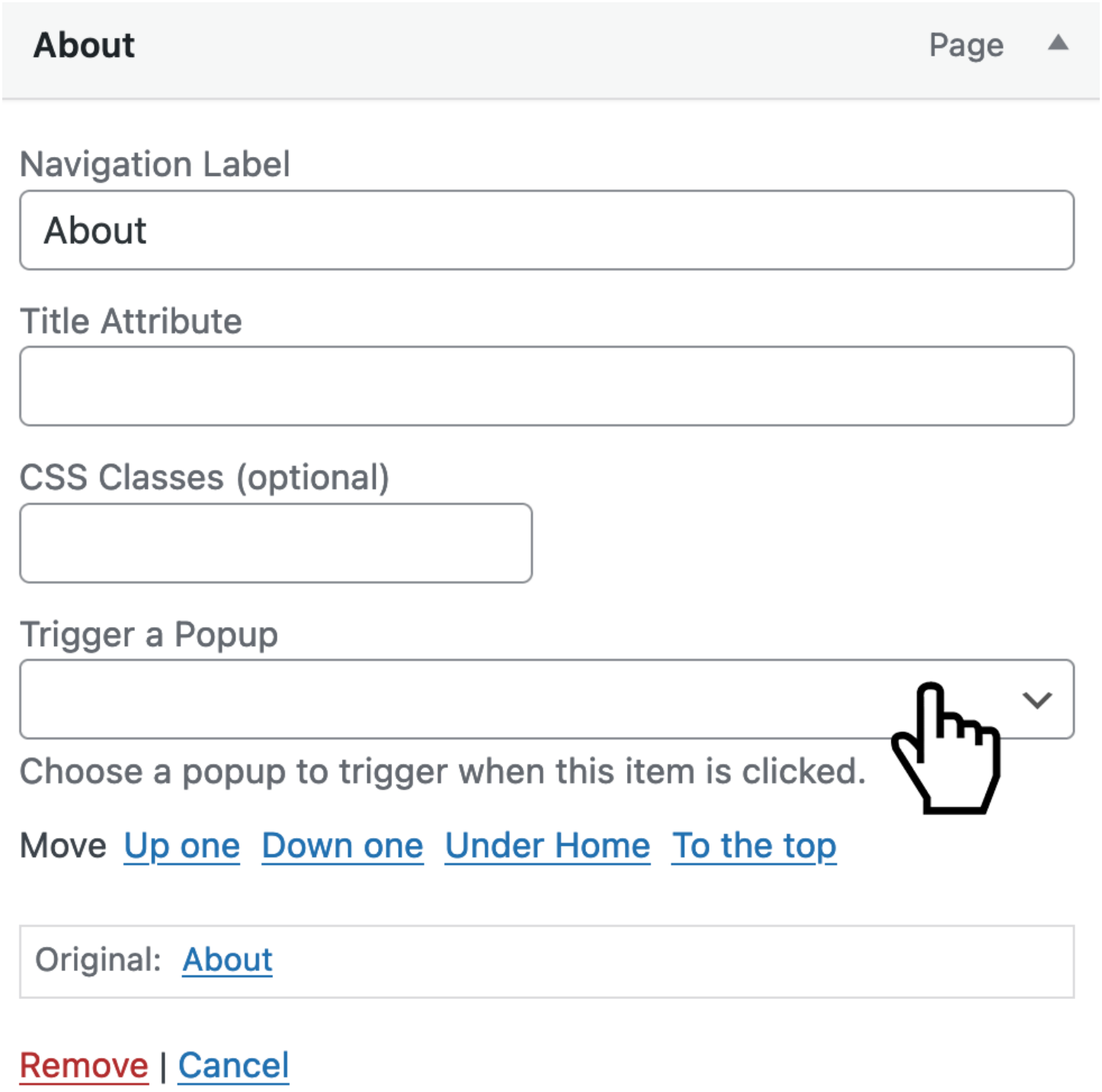
I Don’t See the Trigger a Popup Option in Appearance > Menus

Advanced FAQs · https://wppopupmaker.com/docs/advanced-faqs/i-dont-see-the-trigger-a-popup-option-in-appearance-menus/

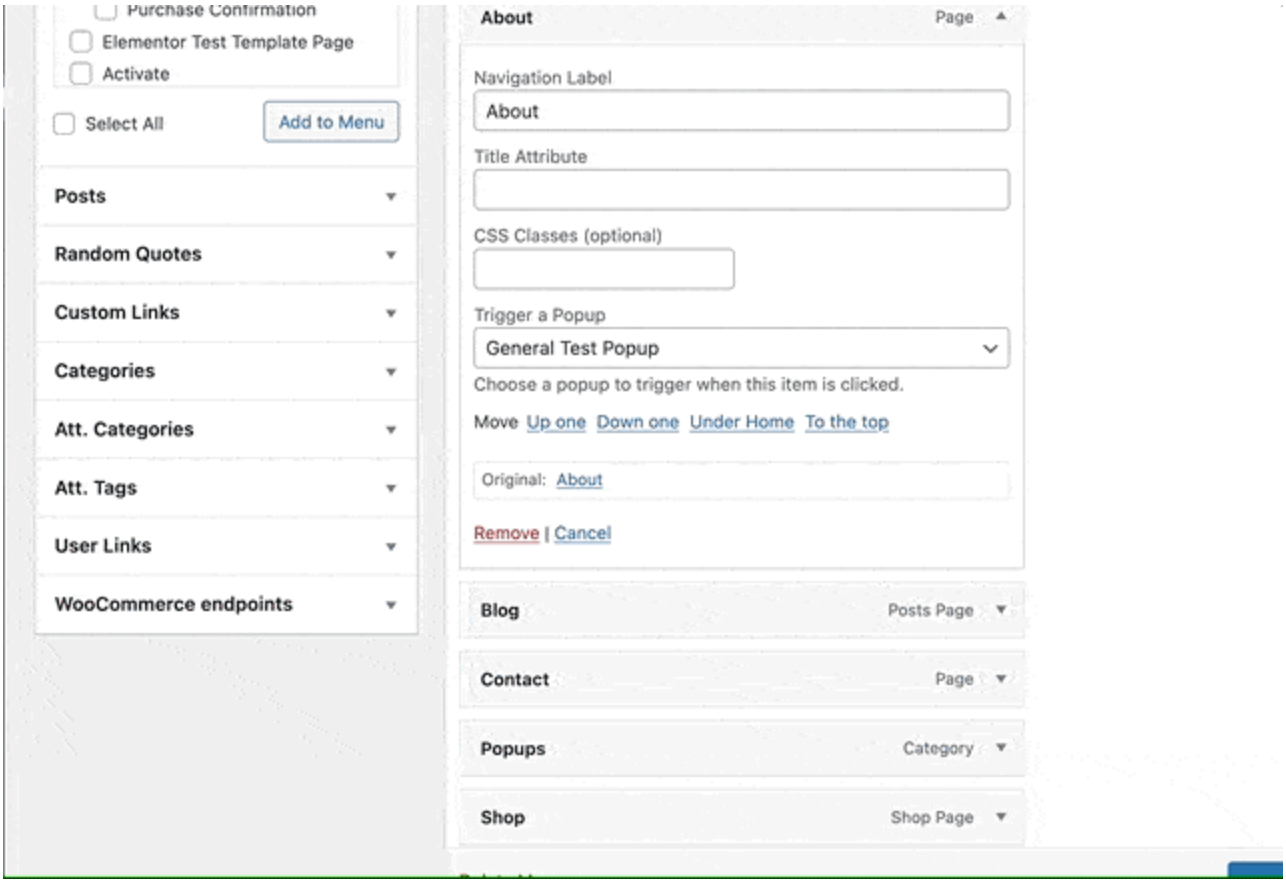
This guide applies to WordPress sites that still support access to **Appearance** > **Menus**.

Popup Maker adds a **Trigger a Popup** option in the WordPress menu editor (**Appearance** > **Menus**). That option lets you launch a popup when you click a menu item.

Here's what that option is *supposed* to look like.



Here's that menu trigger in action. The demo below shows us assigning a popup named **General Test Popup** to the menu item for the **About** page.



If you don't see the **Trigger a Popup** setting on your site, then either:

- your theme or
- one of your plugins

could be *accidentally* blocking the option from showing.

Don't worry. We can help work with your theme's or plugin's development team to get you up and running. Let's get started.

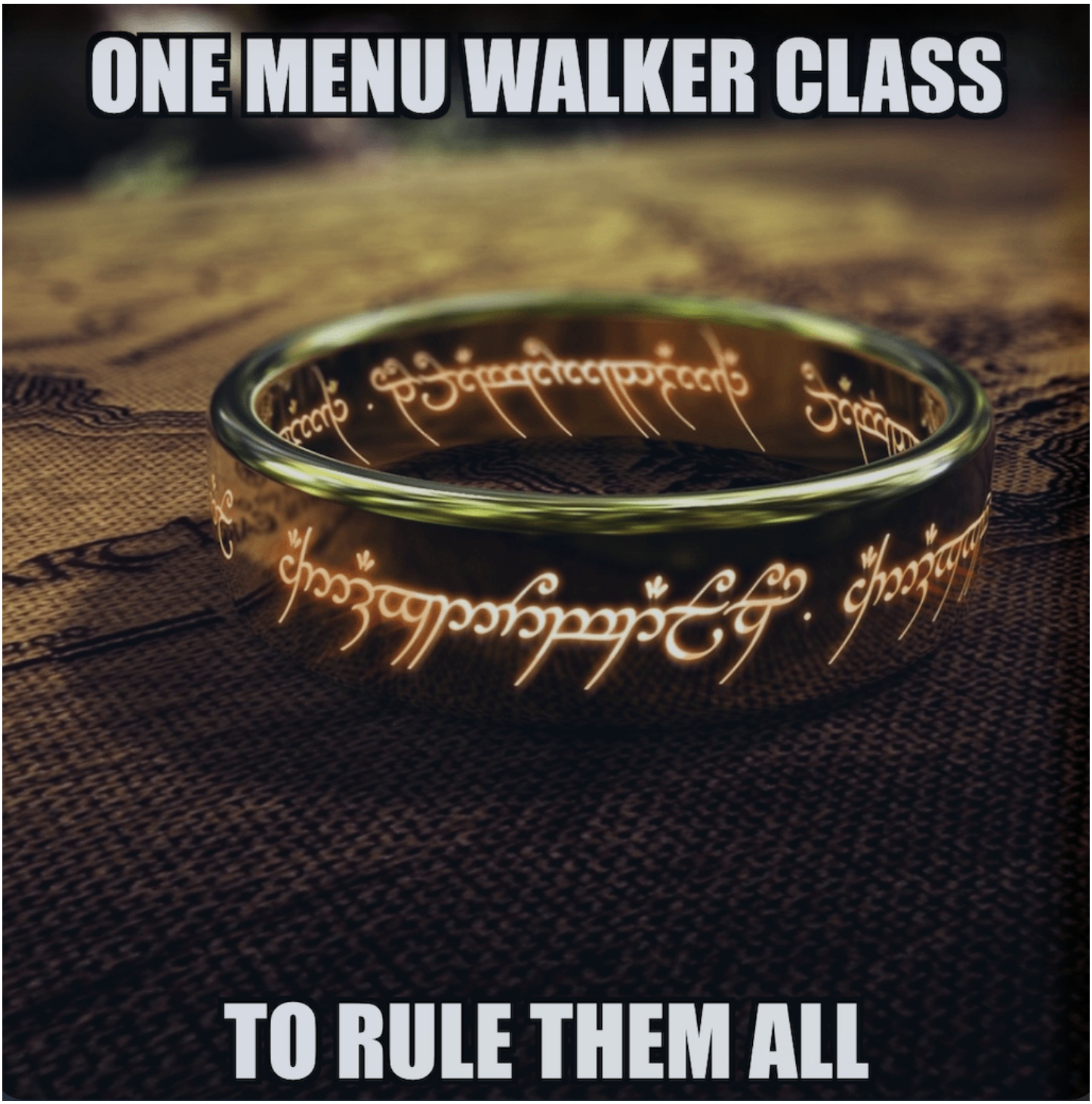
TL;DR

After a theme or plugin adds its custom menu settings it **must** use the wp_nav_menu_item_custom_fields action hook in its `Walker_Nav_Menu_Edit` class so all other custom fields display too.

How do themes or plugins unintentionally block Popup Maker's menu option?

Before WordPress 5.4 theme and plugin developers added custom settings to **Appearance** > **Menus** by replacing WordPress's core Walker_Nav_Menu_Edit PHP class.

But there's a *gotcha*. There can be only one Walker_Nav_menu_Edit class, and the last version of the class (either from WordPress, a theme, or a plugin) wins. I.e., that's the version you'll see working in **Appearance** > **Menus**. If the "winner" didn't account for other versions trying to show their menu options, then you'll only see the winner's menu options.



Example Scenario

Here's a scenario.

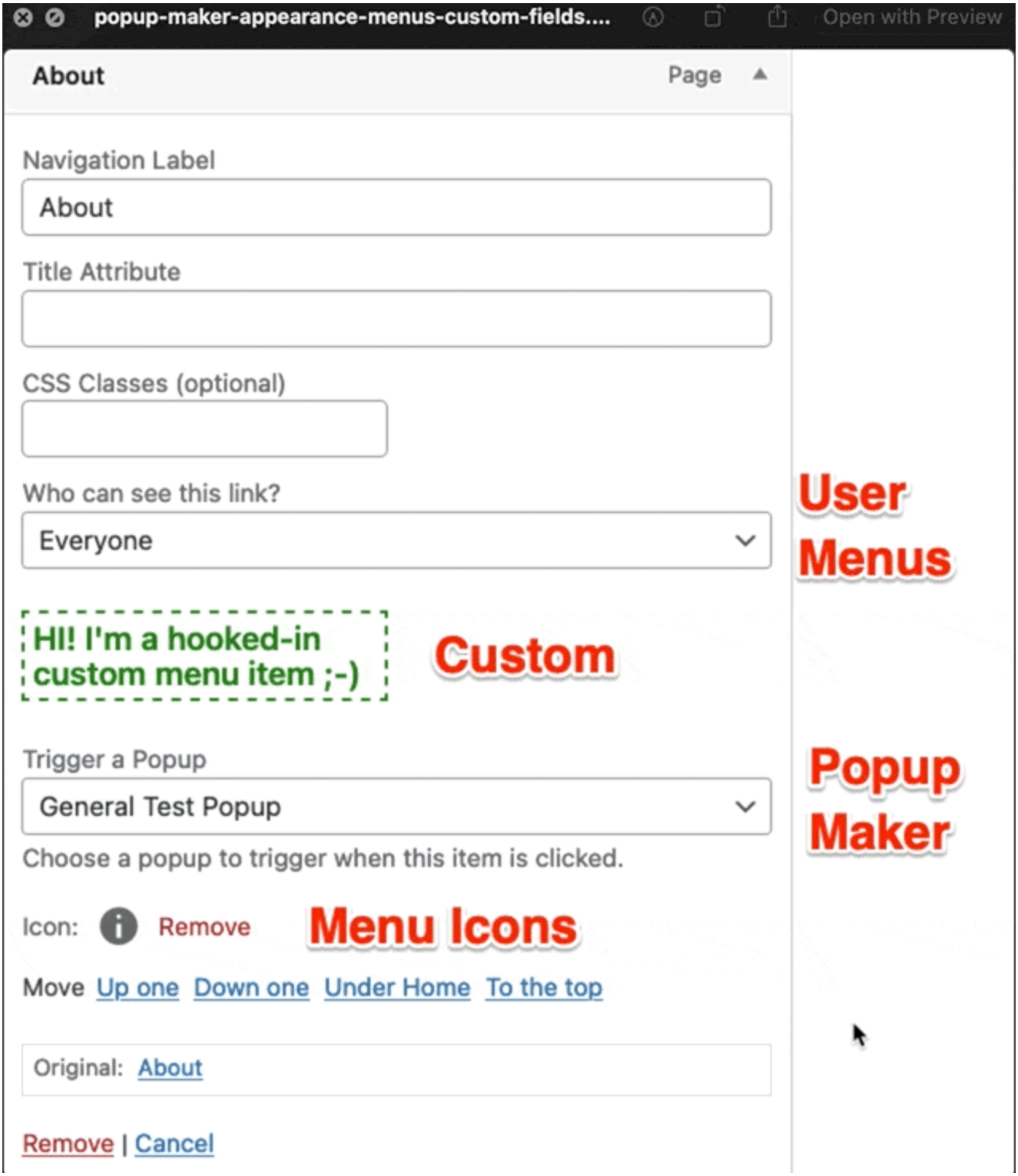
1. You install **Plugin X**.

2. **Plugin X** needs to add a menu setting called **SETTING X**.
3. **Plugin X** replaces WordPress's core `Walker_Nav_Menu_Edit` with its own version.
4. **Plugin X** doesn't realize your theme, and other plugins also add their custom menu settings.
5. When you edit a menu item under **Appearance** > **Menus**, you can see **SETTING X** just fine.
6. But now, the **Trigger a Popup** option you had before is gone.

Here's a before and after visual of our Plugin X scenario

Before Plugin X: We have custom fields from the User Menus, Popup Maker, and Menu Icons plugins and 1 custom field we wrote ourselves.

After Plugin X: We see only **SETTING X**. The other custom menu fields are gone.



A Quick Test

Change your site's theme to a default WordPress theme like Twenty Twenty-One. If you see the **Trigger a Popup** menu field now, then it's confirmed. Your theme needs to support custom menu fields from plugins!

If you still don't see the **Trigger a Popup** setting, you'll need to turn off your plugins 1-by-1 to figure out which plugin isn't playing nice.

Pro Tips

If you need to check for which plugin is causing the issue, try first turning off (1-by-1) the plugins that add menu fields.

What is a hook anyway? Find out what a WordPress hook is.

How to help your theme or plugin author fix the problem

Follow these instructions so your theme or plugin developer can add support for all custom menu fields.

1. Contact the developer of the theme or plugin that's not displaying Popup Maker's **Trigger a Popup** menu option. See our request email example below.
2. Let them know your **Trigger a Popup** menu option isn't displaying where it should be.
3. Ask them to use the wp_nav_menu_item_custom_fields action hook in their copy (or child class) of the Walker_Nav_Menu or Walker_Nav_Menu_Edit class.
4. Share this article (you're reading) with your developer, so they get all the details.

Example request email

Here's an email example you can use to reach out to your theme or plugin developer. Feel free to CC support [at] wppopupmaker.com.

> **Subject**: Please use the wp_nav_menu_item_custom_fields action hook
>
> Hi team!
>
> We want to use Popup Maker to display a popup when we click a menu item. But, Popup Maker's **Trigger a Popup** menu option disappears after we install your [theme | plugin].

> Can you please use the wp_nav_menu_item_custom_fields action hook in your copy (or child class) of the Walker_Nav_Menu or Walker_Nav_Menu_Edit class.
>
> For more details, check out this article that describes the issue and how to fix it.
>
> > I Don’t See the Trigger a Popup Option in Appearance > Menus
>
> Embedded media: https://wppopupmaker.com/docs/navigation-menu-editor/i-dont-see-the-trigger-a-popup-option-in-appearance-menus/embed/#?secret=V28QZak7ue#?secret=Ns688H92VO
>
> Let us know if you have any questions.
>
> Thanks so much for your support :-)

Code Examples

Here's a list of plugins we know add menu options correctly using the `wp_nav_menu_item_custom_fields` action hook.

- Advanced Custom Fields (ACF)
- Menu Icons
- Popup Maker
- User Menus
- WP Fusion Lite

You can look up and view the source code for each plugin in the WordPress plugin directory.

For a quick reference here are code examples of how Popup Maker and the Avada theme (no affiliation) use the `wp_nav_menu_item_custom_fields` action hook.

Popup Maker's child class of Walker_Nav_Menu_Edit

```
59      protected function get_custom_fields( $item, $depth, $args = [] ) {
60          ob_start();
61          $item_id = intval( $item->ID );
62          /**
63           * Get menu item custom fields from plugins/themes
64           *
65           * @param int $item_id post ID of menu
66           * @param object $item Menu item data object.
67           * @param int $depth Depth of menu item. Used for padding.
68           * @param array $args Menu item args.
69           *
70           * @return string Custom fields
71           */
72          do_action( 'wp_nav_menu_item_custom_fields', $item_id, $item, $depth, $args );
73
74          return ob_get_clean();
75      }
```

Visit the source code on GitHub.

Popup Maker's call to add its custom menu options

```
21      public static function init() {
22          add_filter( 'popmake_settings_misc', [ __CLASS__, 'settings' ] );
23
24          if ( PUM_Utils_Options::get( 'disabled_menu_editor', false ) ) {
25              return;
26          }
27
28          // Merge Menu Item Options
29          add_filter( 'wp_setup_nav_menu_item', [ __CLASS__, 'merge_item_data' ] );
30          // Admin Menu Editor
31          add_filter( 'wp_edit_nav_menu_walker', [ __CLASS__, 'nav_menu_walker' ], 999999999 );
32          // Admin Menu Editor Fields.
33          add_action( 'wp_nav_menu_item_custom_fields', [ __CLASS__, 'fields' ], 10, 4 );
34          add_action( 'wp_update_nav_menu_item', [ __CLASS__, 'save' ], 10, 2 );
35          add_filter( 'manage_nav-menus_columns', [ __CLASS__, 'nav_menu_columns' ], 11 );
36      }
```

Visit the source code on GitHub.

Avada's child class of Walker_Nav_Menu

```
                    the menu if the current theme supports it.', 'Avada' ); ?></span>
216                 </label>
217             </p>
218
219             <?php do_action( 'wp_nav_menu_item_custom_fields', $item_id, $item, $depth, $args, $id ); ?>
220
221             <fieldset class="field-move hide-if-no-js description description-wide">
222                 <span class="field-move-visual-label" aria-hidden="true"><?php esc_attr_e( 'Move',
                    'Avada' ); ?></span>
223                 <button type="button" class="button-link menus-move menus-move-up" data-dir="up"><?php
                    esc_attr_e( 'Up one', 'Avada' ); ?></button>
```

Avada's call to add its custom menu options

```
48      public function __construct() {
49
50          self::$theme_info = wp_get_theme();
51
52          add_action( 'admin_enqueue_scripts', [ $this, 'register_scripts' ] );
53          add_action( 'admin_enqueue_scripts', [ $this, 'register_stylesheets' ] );
54
55          do_action( 'fusion_init' );
56
57          self::$classes['menus'] = new Avada_Megamenu();
58
59          // Add the first level menu style dropdown to the menu fields.
60          add_action( 'wp_nav_menu_item_custom_fields', [ $this, 'add_menu_button_fields' ], 10, 5 );
61
62          // Add the mega menu custom fields to the menu fields.
63          if ( Avada()->settings->get( 'disable_megamenu' ) ) {
64              add_filter( 'avada_menu_options', [ $this, 'add_megamenu_fields' ], 20, 4 );
65          }
66
67          // Add the menu arrow highlights.
68          add_filter( 'avada_menu_arrow_hightlight', [ $this, 'add_menu_arrow_highlight' ], 10, 2 );
69
70          // Add special menu items meta box.
71          add_action( 'admin_head-nav-menus.php', [ $this, 'add_special_links_meta_box' ] );
72
73          // AutoComplete field ajax.
74          add_action( 'wp_ajax_avada_mega_menu_autocomplete', [ $this, 'get_autocomplete' ] );
75
76      }
```