Deprecated Action Hooks

- Use Cases
- How To Deprecate Action Hooks
- Updating Deprecated Action Hooks

Elementor offers the `Deprecation::do_deprecated_action()` method to handle the deprecation process for actions hooks inside Elementor code.

Use Cases

When considering naming conventions, the old naming conventions should work alongside the new ones.

An example of renaming occured when Elementor unified the registration process for several components in versions 3.5. Many action hooks across different managers were renamed.

How To Deprecate Action Hooks

We have an action hook called `elementor/old/action`, and want to rename it `elementor/new/action`.

For example, take the following code:

```
class Name {

	function init( $args ): void {
		do_action( 'elementor/old/action', $args );
	}

}
```

The new code, with the deprecation handler, will look like this:

```
use Elementor\Plugin;

class Name {

	function init( $args ): void {
		// Deprecation handler
		Plugin::$instance->modules_manager->get_modules( 'dev-tools' )->deprecation->do_deprecated_action(
			'elementor/old/action',
			[ $args ],
			'3.5.0',
			'elementor/new/action'
		);

		// Run the new hook
		do_action( 'elementor/new/action', $args );
	}

}
```

The new `elementor/new/action` hook will run as expected. In addition, the deprecation handler will run the old hook (for backwards compatibility) while sending some notifications to developers.

After eight major versions, the code will look like this:

```
class Name {

	function init( $args ): void {
		do_action( 'elementor/new/action', $args );
	}

}
```

Updating Deprecated Action Hooks

Addon developers who use the old hook should simply rename it, as follows:

```
function hook_callback() {
□/* ... */
}
- add_action( 'elementor/old/action', 'hook_callback' );
+ add_action( 'elementor/new/action', 'hook_callback' );
```

□□□ #2
□ □□□ □□□ 2026-03-26 17:08:44 UTC
□ □□□ □□□ 2026-03-26 17:08:44 UTC