Custom Emails: Developer Hooks & Filters Reference

Developers · https://wpfactory.com/docs/custom-emails-developer-hooks-filters/

Additional Custom Emails & Recipients for WooCommerce exposes filters and actions so you can change how emails are sent, alter their content, extend the shortcode set, and hook your own code into the send lifecycle, all without editing plugin files. Every hook below is called from the plugin's own code and uses the `wpfactory_wc_custom_emails_*` prefix.

Prefix change in v3.7.3

In version 3.7.3 the plugin renamed its prefixes from `alg` to `wpfactory`. This affects the main accessor function and most hooks:

Old (pre-3.7.3)	Current
`alg_wc_custom_emails()`	`wpfactory_wc_custom_emails()`
`alg_wc_ce_send_email()`	`wpfactory_wc_ce_send_email()`
`alg_wc_custom_emails_do_send`	`wpfactory_wc_custom_emails_do_send`
`alg_wc_custom_emails_do_send_order_email`	`wpfactory_wc_custom_emails_do_send_order_email`

The old filter names are still honored, but there is **no** back-compat alias for the `alg_wc_custom_emails()` and `alg_wc_ce_send_email()` functions. Update any code that calls them.

Send-gate filters

These decide whether an email actually goes out. They run in this order: `is_enabled`, then (for order emails) `do_send_order_email`, then `do_send`.

Filter	Passed value	Arguments	Effect

`wpfactory_wc_custom_emails_is_enabled`	`true`	`$enabled, $email, $object_id, $do_force_send`	Return `false` to stop this email early. `$do_force_send` is true for manual sends.
`wpfactory_wc_custom_emails_do_send_order_email`	`true`	`$do_send, $email, $order`	Return `false` to block an order email based on the order.
`wpfactory_wc_custom_emails_do_send`	`true`	`$do_send, $email`	Final gate, runs for every email type just before send.

```
// Never send a particular custom email on Sundays.
add_filter( 'wpfactory_wc_custom_emails_do_send', function ( $do_send, $email ) {
    if ( 'alg_wc_custom' === $email->id && 'Sun' === gmdate( 'D' ) ) {
        return false;
    }
    return $do_send;
}, 10, 2 );
```

Content filters

Filter	Filters	Arguments
`wpfactory_wc_custom_emails_subject`	Final subject	`$subject, $email, $order, $user, $product`
`wpfactory_wc_custom_emails_content`	Final body content	`$content, $email, $order, $user, $product`
`wpfactory_wc_custom_emails_from_name`	"From" name	`$from_name, $email`
`wpfactory_wc_custom_emails_from_address`	"From" address	`$from_email, $email`

Filter	Filters	Arguments
`wpfactory_wc_custom_emails_headers`	Email headers (Reply-to)	`$headers, $email`
`wpfactory_wc_custom_emails_exclude_recipients`	The exclude-recipients list	`$value, $email`
`wpfactory_wc_custom_emails_delay_start_time`	Base timestamp the delay is added to	`$time, $email, $object_id`
`wpfactory_wc_custom_emails_base_dir`	Attachments base directory	`$dir`

```
// Add a suffix to every custom email subject.
add_filter( 'wpfactory_wc_custom_emails_subject', function ( $subject, $email, $order, $user, $product ) {
    return $subject . ' [' . get_bloginfo( 'name' ) . ']';
}, 10, 5 );
```

Order-targeting filters

Filter	Purpose	Arguments
`wpfactory_wc_custom_emails_check_order_products`	Override the "require/exclude products" match result	`$matched, $order, $product_ids`
`wpfactory_wc_custom_emails_check_order_product_terms`	Override the category/tag match result	`$matched, $order, $term_ids, $taxonomy`
`wpfactory_wc_custom_emails_order_product_term_ids`	Change the term IDs read for an order item (for example to add custom taxonomies)	`$term_ids, $item, $order, $wanted_ids, $taxonomy`

Extensibility filters

Filter	Purpose
`wpfactory_wc_custom_emails_shortcode_prefix`	Prefix added to every shortcode tag (default empty), for example to namespace them.
`wpfactory_wc_custom_emails_shortcode_order_func`	Allowlist of `WC_Order` methods that `[order_func]` may call (default empty).
`wpfactory_wc_custom_emails_shortcode_product_func`	Allowlist of `WC_Product` methods that `[product_func]` may call (default empty).
`wpfactory_wc_custom_emails_stop_emails_list`	The list of default WooCommerce emails offered in the "Stop emails" option.
`wpfactory_wc_custom_emails_add`	The array of registered custom email instances (advanced).
`wpfactory_wc_custom_emails_class`	Maps an email number to its PHP class name.

Actions

Action	Fires
`wpfactory_wc_custom_emails_core_loaded`	When the core object is ready. Argument: the core instance.
`wpfactory_wc_custom_emails_before_add`	Just before the custom email classes are registered with WooCommerce.
`wpfactory_wc_custom_emails_settings_saved`	After the plugin's own settings tab is saved.
`wpfactory_wc_custom_emails_email_sent`	After an email is sent. Argument: the email instance.

Sending an email from code

Use the helper function to send a custom email for a given object (usually an order ID):

```
// Send custom email #1 for order 1234.
wpfactory_wc_ce_send_email( 1, 1234, 'triggered from my plugin' );
```

The first argument can be an email number (1 for the first email) or a class name. The third argument is an optional note recorded on the order.

Registering a custom shortcode

Read the current order, user, or product from the shortcodes object:



CUSTOM EMAILS

See every feature and the pricing on the Custom Emails plugin page.

View plugin →


```
add_shortcode( 'my_order_shipping_city', function ( $atts ) {
    if (
        function_exists( 'wpfactory_wc_custom_emails' ) &&
        ! empty( wpfactory_wc_custom_emails()->core->shortcodes->order )
    ) {
        return wpfactory_wc_custom_emails()->core->shortcodes->order->get_shipping_city();
    }
    return '';
} );
```

Need a hand?

Our team answers developer questions on the Custom Emails support forum.

Related articles

- Custom Email Triggers Reference
- Custom Email Preview Shows the Wrong or Old Content
- Custom Email Sends Too Early or at the Wrong Time
- Send a Custom Email from the Order Screen
- Custom Email Not Sending: Troubleshooting