Advanced Reporting: Developer Hooks & Filters Reference

Developers · https://wpfactory.com/docs/woocommerce-advanced-reporting-hooks-filters/

Developer Hooks & Filters Reference

WooCommerce Advanced Reporting and Statistics exposes a small set of actions and filters so you can extend the reports page, add your own overview widgets, and attach custom data to the orders report without editing plugin files. This page lists every public hook, verified against the plugin source.

All hooks use the `wpfactory_wc_ars_pro_` prefix. Register your callbacks from a small custom plugin or your theme's `functions.php`.

Actions

Action	Arguments	Fires when
`wpfactory_wc_ars_pro_output_settings`	None	When the plugin's admin page is rendered. This is the hook the plugin itself uses to draw the whole reports interface.
`wpfactory_wc_ars_pro_before_overview`	None	Just before the overview summary cards on the General report. Use it to output content above the totals.
`wpfactory_wc_ars_pro_after_overview`	None	Just after the overview summary cards. Use it to output content below the totals.

```
add_action( 'wpfactory_wc_ars_pro_after_overview', function () {
    echo '<p>Figures exclude wholesale orders.</p>';
} );
```

Filters

Filter	Filtered value	Extra arguments
`wpfactory_wc_ars_pro_overview_widgets`	The array of overview cards shown on the General report.	None
`wpfactory_wc_ars_pro_orders_custom_data`	An array of custom data attached to the orders report response (default empty array).	`$order_results` (the array of matching `WC_Order` objects)

Add an overview card

Each item in the `wpfactory_wc_ars_pro_overview_widgets` array is an associative array. The keys the plugin reads are `id`, `title`, `desc`, `icon` (a Font Awesome name without the `fa-` prefix), and optionally `div_class`, `div_style` and `span_desc`. The `id` becomes the CSS class the plugin's JavaScript targets when it fills the value in, so the built-in cards (for example `total`, `subtotal`, `grand_total`) get their numbers from the report calculation. A card with a new `id` will render but stays empty unless your own script populates `.your-id`.

```
add_filter( 'wpfactory_wc_ars_pro_overview_widgets', function ( $widgets ) {
    $widgets[] = array(
        'id'    => 'my_metric',
        'title' => 'MY METRIC',
        'desc'  => 'custom value',
        'icon'  => 'star',
    );
    return $widgets;
} );
```

Attach custom data to the orders report

The `wpfactory_wc_ars_pro_orders_custom_data` filter runs while the Orders AJAX response is built. It receives an empty array and the full list of matching orders, and whatever you return is added to the response under a `custom_data` key. Use it to pass extra aggregates to a custom front-end script.

```
add_filter( 'wpfactory_wc_ars_pro_orders_custom_data', function ( $data, $order_results ) {
    $data['order_count'] = count( $order_results );
    return $data;
}, 10, 2 );
```

HPOS compatibility



WOOCOMMERCE ADVANCED REPORTING AND STATISTICS

See every feature and the pricing on the WooCommerce Advanced Reporting and Statistics plugin page.

View plugin →

The plugin declares compatibility with WooCommerce High-Performance Order Storage (custom order tables). Its report queries run against either the legacy post-based storage or the HPOS tables, depending on what your store uses, so callbacks you register behave the same under both.

Requirements

- WooCommerce is required (declared with `Requires Plugins: woocommerce`).
- Minimum PHP version: 5.3.

Need a hand?

If you are building on top of Advanced Reporting & Statistics and need guidance, our team can help through the plugin support page.

Related articles

- WooCommerce Advanced Reporting and Statistics Settings Reference
- Getting Started with WooCommerce Advanced Reporting and Statistics
- How to Filter WooCommerce Reports by Country, Product, Customer and More
- WooCommerce Reporting Metrics Explained (Net, Gross, Total Sales)