

Actions, filters, triggers

Implementing deprecated filters

If you for some strange reason would like to implement filters that has been used in the old plugin, implementation should look like this. In this live example, you actually execute the deprecated filter before executing the up-to-date one. Doing this, you get the chance to inject old values on the way through the new filter.

```
$deprecatedStoreId = WordPress::applyFiltersDeprecated('set_storeid', null);
$storeId = (int)WordPress::applyFilters('setStoreId', $deprecatedStoreId);
```

Implemented and available filters

This plugin offers a bunch of internal filters and actions. Below is a list of them.

Filter name	Type	In	Return	Description						
rbwc_admin_dynamic_content	filter	\$content \$current_section	\$yourContent	Renders dynamic content for each section. If something specific, that is not included in the form fields, this could be used and will be appended in the of each section.						
rbwc_get_custom_form_fields	filter	\$currentArray	\$yourArray	If settings are dependent of other activated options in the configuration, this filter should be used. The plugin use this internally, to enable/disable the developer-options tab.						
rbwc_get_plugin_information	filter	\$content	\$yourContent	Content added to this section will be shown in the information/support, as a part of the "version view". If you for example need to show openssl releas other vital data for your platform, you should add it here. Example: <pre>['OpenSSL version' => defined('OPENSSL_VERSION_TEXT') ? OPENSSL_VERSION_TEXT : '']</pre> <table border="1"> <tr> <td>Plugin-version</td> <td>0.0.1.0</td> </tr> <tr> <td>Composer-version</td> <td>0.0.1.0</td> </tr> <tr> <td>PHP-Version</td> <td>7.2.31-1+ubuntu18.04.1+deb.sury.org+1</td> </tr> </table>	Plugin-version	0.0.1.0	Composer-version	0.0.1.0	PHP-Version	7.2.31-1+ubuntu18.04.1+deb.sury.org+1
Plugin-version	0.0.1.0									
Composer-version	0.0.1.0									
PHP-Version	7.2.31-1+ubuntu18.04.1+deb.sury.org+1									
rbwc_can_display_order_info_after_details	filter	boolean	boolean	rbwc_can_display_order_info_after_<WHERE> The can_display-trio is used to disable each payment info section in the admin view. The details section also has a separate action that triggers inste standard action from woocommerce. That action delivers all payment information required to recreate the "bluebox information" from v2.x.						
rbwc_show_order_details	action	\$orderData	is echo	This action is executed right after the rbwc_can_display_order_info_after_details -filter. By disabling the three can_display_order_info and execute hook with a full order representation, it is possible to create your own view right after the order details in the admin. This is pushed in here, in case the requires a different view from this plugin.						
rbwc_can_display_order_info_after_billing	filter	boolean	boolean	rbwc_can_display_order_info_after_<WHERE> The can_display-trio is used to disable each payment info section in the admin view.						
rbwc_can_display_order_info_after_shipping	filter	boolean	boolean	rbwc_can_display_order_info_after_<WHERE> The can_display-trio is used to disable each payment info section in the admin view.						
rbwc_get_payment_field_size	filter	\$size, \$fieldName	integer	If you believe you need a shorter or longer value for the default payment field lenght, that is set to 24, you can apply for that value here.						

rbwc_pref_article_number_sku	filter	boolean	boolean	If you have further requirements of how article numbers are fetched, you could return true for this, to make the plugin use Sku instead - if it exists. This is not configurable through the admin interface as it is handled as a filter instead. If it is really necessary to use SKU, use this filter.
rbwc_get_article_number	filter	\$currentArticleNumber, WC_Product	\$theArticleNumber	If you have further ways of fetching an article from a WC_Product object, this is where you do it.
rbwc_coupons_exempt	filter	boolean, WC_Coupon	boolean	For special occasions when the options should remain untouched but we still need to make exceptions for coupons.
rbwc_get_coupon_vat_pct	filter	integer	integer	Default value for vat percent in the coupon calculation method is 0. Coupons are normally, at least in Sweden, already applied with vat.
rbwc_get_address_disabled	filter	boolean	boolean	Disable getAddress form in checkout (where it is implementable).
rbwc_js_loaders_checkout	filter	array	array	List of javascripts (frameworks?) that should be loaded during RCO.
rbwc_get_shipping_name	filter	string	string	Generates own article number for shipping if the default "shipping" is not preferred, or if another translation is needed.
rbwc_get_shipping_description	filter	string	string	Generates a orderline description for the shipping, if the preferred translation for "Shipping" is not preferred.
rbwc_set_store_id	filter	string	string	Set a store id for the payment request.
rbwc_trigger_callback	filter	array	array	External code to handle test callbacks.
rbwc_get_address_field_controller	filter	array	array	getTransformedAddressResponse, ability to use other fields than the defaults, generated by the internal filter.
rbwc_get_address_button_text	filter	string	string	Alternative naming or translation for the getAddress request button. Default is "Get address".
resurs_trigger_test_callback	filter	string	string	External trigger for the test callback. 2.X-DEPRECATED
set_storeid	filter	string	string	Store id, old filter. 2.X-DEPRECATED
resurs_get_address_enabled	filter	boolean	boolean	Sets, in the old fashioned way, a state if getAddress should be enabled where it is normally disabled. 2.X-DEPRECATED
rbwc_customer_synchronize	trigger	<pre>\$('body').trigger('rbwc_customer_synchronize', { version: 2 })</pre>	-	Javascript frontend trigger. Used to synchronize customer billing/shipping fields. Event contains. depending on version. <div style="border: 1px solid #ccc; padding: 5px; margin: 5px 0;"> <p>When v1 is sending data</p> <pre>{ version: 1 }</pre> </div> <p>resursBankRcoDataContainer Version 1 sample data:</p>

```
{
  "rco_customer": {
    "paymentMethod": "INVOICE",
    "customerData": {
      "address": {
        "firstname": "Korv",
        "surname": "Alexandersson",
        "address": "Glassgatan 15",
        "addressExtra": "",
        "postal": "41655",
        "city": "Göteborg",
        "countryCode": "SE",
        "telephone": "0701122334",
        "email": "noreply@resurs.se"
      },
      "delivery": {},
      "ssn": "198001010001",
      "paymentMethod": "INVOICE"
    }
  },
  "rco_payment": "INVOICE"
}
```

When v2 is sending data

```
{
  version: 1
}
```

resursBankRcoDataContainer Version 2 sample data (when deliveryAddress has constant it looks similarly as billingAddress):

resursBankRcoDataContainer Version 2

```

{
  "rco_customer": {
    "type": "CustomerChangeEvent",
    "ts": 1632576948828,
    "source": "@rco-app",
    "phone": "0701122334",
    "email": "noreply@resurs.se",
    "billingAddress": {
      "firstName": "Vincent",
      "lastName": "Alexandersson",
      "addressRow1": "Glassgatan 15",
      "addressRow2": null,
      "postalCode": "41655",
      "city": "Göteborg"
    },
    "deliveryAddress": {
      "firstName": null,
      "lastName": null,
      "addressRow1": null,
      "addressRow2": null,
      "postalCode": null,
      "city": null
    }
  },
  "rco_payment": {
    "type": "PSP_DEBIT_CARD",
    "ts": 1632576948828,
    "source": "@rco-app",
    "method": "PSPCARD_DEBIT",
    "id": "PSPCARD_DEBIT",
    "fee": 0
  }
}

```

rbwc_pur
chase_rej
ect

trigger


```

${'body'
}.trigger
(
  'rbw
c_pu
rcha
se_r
ejec
t', {
  type
: '<
type
>'
}
);

```

-

Then ResursCheckout framework v1 or v2 rejects a payment, either because of a failure or a denied payment (v1 only), this trigger executes and can be picked up by integrated plugins to handle a rejected order.

rbwc_purchase_reject_customer_message	filter	string	string	<p>When Resurs Bank Checkout rejects a payment due to failure/deny, this filter adds an extra note to customer in a front-end message. The default look below:</p> <pre>\$failNote = sprintf(__('Order was rejected by Resurs Bank with status %s.', 'trbwc'), \$rejectType);</pre> <p>The above message is always injected in order notices and in the default customer message. After this message has been added to the order, further added:</p> <pre>\$failNote .= ' ' . WordPress::applyFilters('purchaseRejectCustomerMessage', __('Please contact customer service for more information.', 'trbwc'));</pre> <p>The above "Contact customer service"-message is added with a filter setup (which is this). If you want to change this message, you can use the mentioned filter to update the string with your own.</p>
rbwc_get_payment_method_icon	filter	\$url, \$paymentMethodData	string	<p>When logos are active for payment methods, you can customize the look of the icon by adding this filter. The output string should be a URL to your chosen logo. For each payment method. \$paymentMethodData contains details about type, specificType and id that is vital for method type detection.</p> <p>This also deprecates  RWC-33 - woocommerce_resurs_bank_' . \$type . '_checkout_icon (iconified method) <input type="button" value="DONE"/> .</p>
rbwc_get_specific_type_fields	filter	\$return (array), \$key	array	<p>Customizable customer form fields for checkouts. See this section.</p>
rbwc_get_minimum_annuity_price	filter	\$integer, \$customerCountry	integer	<p>When annuity factors are handled, the minimum payment price is 150 SEK/DKK/NOK or 15 EUR for FI. If you really need to change this value to something lower, this is the filter to use.</p>
rbwc_part_payment_string	filter	string, ['monthlyPrice' => \$monthlyPrice, 'monthlyDuration' => \$annuityDuration, 'paymentLimit' => \$minimumPaymentLimit]	string	<p>When displaying part payment information on product pages, the below string is set to be default. This filter executes an internal feature that makes it to show part payment information via a WordPress page. It also helps if you want to generate your own information without a page. The second variable filter is the the default values for duration and months.</p> <p>Part pay from %s per month.</p>
rbwc_method_minimum_limit	filter	int, \$paymentMethodInformation	int	<p>If you want to limit the minimum allowed payment amount for a specific payment method, this is the one to set the data with.</p> <p>For example, you require payment amounts to be higher than 1000, to turn on the INVOICE, but the payment method allows payments from 500.</p> <p><i>The minAmount of the payment method can never be exceeded.</i></p>
rbwc_method_maximum_limit	filter	int, \$paymentMethodInformation	int	<p>If you want to limit the maximum allowed payment amount for a specific payment method, this is the one to set the data with.</p> <p>For example, you require SWISH to not be higher than 10000 but the payment method allows payments up to 20000.</p> <p><i>The maxAmount of the payment method can never be exceeded.</i></p>