

Hooks in PrestaShop 1.5

Table of content

- [Hooks in PrestaShop 1.5](#)
 - [New & updated hooks since v1.4](#)
 - [New hooks since v1.4](#)
 - [Updated hooks since v1.4](#)
 - [Front-office and back-office](#)
 - [Existing front-office hooks](#)
 - [Existing back-office hooks](#)

Hooks in PrestaShop 1.5

Hooks in PrestaShop 1.5 have a new naming scheme, with a situation-specific prefix:

- **action**. These hooks are triggered by specific events that take place in PrestaShop.
- **display**. These hooks result in something being displayed, either in the front-end or the back-end.

This way, you will know right away what type of hook you are dealing with. Older hooks have been renamed to match this scheme.

New & updated hooks since v1.4

All hooks in PrestaShop have been updated for version 1.5, and several new ones were added.

New hooks since v1.4

Hook name	Hook description	What to expect in <code>\$params</code> ? (Pseudocode)
actionAdminControllerUpdate_optionsBefore		
actionAdminMetaSave	After saving configuration in AdminMeta.	N/A
actionAuthentication	After authentication.	N/A
actionBeforeAuthentication	Before authentication.	N/A
actionCartSave	After saving the Cart object.	N/A
actionCustomerAccountAdd	Called when a new customer successfully creates account.	<pre>array('_POST' => (array) \$_POST, 'newCustomer' => (object) Customer object);</pre>
actionHtaccessCreate	After .htaccess creation.	N/A
actionObjectCategoryDeleteAfter	Not invoked by the PrestaShop core itself - can be used after deleting a category to notify the Horizontal Top Menu.	N/A
actionObjectCategoryUpdateAfter	Not invoked by the PrestaShop core itself - can be used after updating a category to notify the Horizontal Top Menu.	N/A
actionObjectCmsDeleteAfter	Not invoked by the PrestaShop core itself - can be used after deleting a CMS page to notify the Horizontal Top Menu.	N/A

actionObjectCmsUpdateAfter	Not invoked by the PrestaShop core itself - can be used after deleting a CMS page to notify the Horizontal Top Menu.	N/A
actionObjectManufacturerDeleteAfter	Not invoked by the PrestaShop core itself - can be used after deleting a manufacturer to notify the Horizontal Top Menu.	N/A
actionObjectManufacturerUpdateAfter	Not invoked by the PrestaShop core itself - can be used after updating a manufacturer to notify the Horizontal Top Menu.	N/A
actionObjectProductDeleteAfter	Not invoked by the PrestaShop core itself - can be used after deleting a product to notify the Horizontal Top Menu.	N/A
actionObjectProductUpdateAfter	Not invoked by the PrestaShop core itself - can be used after updating a product to notify the Horizontal Top Menu.	N/A
actionObjectSupplierDeleteAfter	Not invoked by the PrestaShop core itself - can be used after deleting a supplier to notify the Horizontal Top Menu.	N/A
actionObjectSupplierUpdateAfter	Not invoked by the PrestaShop core itself - can be used after updating a supplier to notify the Horizontal Top Menu.	N/A
actionOrderDetail	To set the follow-up in Smarty when order detail is called.	<pre>array('carrier' => (object) Carrier object, 'order' => (object) Order object);</pre>
actionOrderReturn	Called after a new Order Return has been made.	<pre>array('orderReturn' => (object) OrderReturn object);</pre>
actionOrderSlipAdd	Called when the quantity of a product changes in an order. WARNING: only invoked when a product is actually removed from an order.	<pre>array('order' => Order object, 'productList' => array((int) product ID 1, (int) product ID 2, ..., (int) product ID n), 'qtyList' => array((int) quantity 1, (int) quantity 2, ..., (int) quantity n));</pre> <p>The order of IDs and quantities is important!</p>
actionOrderStatusPostUpdate	Called after the status of an order changes.	<pre>array('newOrderStatus' => (object) OrderState object, 'id_order' => (int) Order ID);</pre>
actionOrderStatusUpdate	Called before the status of an order changes.	<pre>array('newOrderStatus' => (object) OrderState object, 'id_order' => (int) Order ID);</pre>
actionPaymentCCAdd	Payment CC added	<pre>array('paymentCC' => (object) OrderPayment object);</pre>
actionPaymentConfirmation	Called after a payment has been validated	<pre>array('id_order' => (int) Order ID);</pre>

actionSearch	After the search in the store. Includes both instant and normal search.	array('expr' => (string) Search query, 'total' => (int) Amount of search results);
actionShopDataDuplication	After duplicating a shop.	array('old_id_shop' => (int) Old shop ID, 'new_id_shop' => (int) New shop ID);
actionTaxManager	Not used by PrestaShop core.	N/A
actionUpdateQuantity	After updating the quantity of a product.	array('id_product' => (int) Product ID, 'id_product_attribute' => (int) Product attribute ID, 'quantity' => (int) New product quantity);
actionValidateOrder	After an order has been validated. Doesn't necessarily have to be paid.	array('cart' => (object) Cart object, 'order' => (object) Order object, 'customer' => (object) Customer object, 'currency' => (object) Currency object, 'orderStatus' => (object) OrderState object);
actionWatermark	After a watermark has been added to an image.	array('id_image' => (int) Image ID, 'id_product' => (int) Product ID);
displayAdminCustomers	Can be used by modules when the tab AdminCustomers is displayed in the Back Office.	array('id_customer' = (int) Customer ID);
displayAdminOrder	Can be used by modules when the tab AdminOrder is displayed in the Back Office.	array('id_order' = (int) Order ID);
displayAdminStatsGraphEngine	Not used by PrestaShop core.	N/A
displayAdminStatsGridEngine	Not used by PrestaShop core.	N/A
displayAdminStatsModules	Used when the statistics page in the Back Office is shown.	N/A
displayBackOfficeFooter	Displayed right before the Back Office footer.	N/A
displayBackOfficeHeader	Displayed between the <head></head> tags on every Back Office page (when logged in).	N/A
displayBackOfficeHome	Displayed on the dashboard page of the Back Office.	N/A
displayBackOfficeTop	Shown above the actual content of a Back Office page	N/A

displayBeforeCarrier	This hook is displayed before the carrier list on the Front Office	<pre> array('carriers' => array(array('name' => (string) Name, 'img' => (string) Image URL, 'delay' => (string) Delay text, 'price' => (float) Total price with tax, 'price_tax_exc' => (float) Total price without tax, 'id_carrier' => (int) intified option delivery identifier, 'id_module' => (int) Module ID), 'checked' => (int) intified selected carriers, 'delivery_option_list' => array(array(0 => array(// First address '12,' => array(// First delivery option available for this address carrier_list => array(12 => array(// First carrier for this option 'instance' => Carrier Object, 'logo' => <url to the carrier's logo>, 'price_with_tax' => 12.4, // Example 'price_without_tax' => 12.4, // Example 'package_list' => array(1, // Example 3, // Example),), is_best_grade => true, // Does this option have the biggest grade (quick shipping) for this shipping address is_best_price => true, // Does this option have the lower price for this shipping address unique_carrier => true, // Does this option use a unique carrier total_price_with_tax => 12.5, total_price_without_tax => 12.5, position => 5, // Average of the carrier position),),), 'delivery_option' => array('<id_address>' => Delivery option, ...)); </pre> <p>NOTE: intified means an array of integers 'intified' by Cart::intifier</p>
displayBeforePayment	Redirect user to the module instead of displaying payment modules	<pre> array('module' => (string) Module link [e.g. /module /test/front]); </pre>
displayCarrierList	Display extra carriers in the carrier list.	<pre> array('address' => (object) Address object); </pre>
displayCustomerAccount	Display on page account of the customer.	N/A
displayCustomerAccountForm	Display some information on the form to create a customer account.	N/A
displayCustomerAccountFormTop	Display some information on the top of the form to create a customer account.	N/A
displayFooter	Add block in footer.	N/A

displayFooterProduct	Add new blocks under the product description.	N/A
displayHeader	A hook which allow you to do things in the header of every page.	N/A
displayHome	Display on content of home page.	N/A
displayInvoice	Add blocks to invoice (order).	N/A
displayLeftColumn	Display in left column.	N/A
displayLeftColumnProduct	Display in left column on product page (not to be confused with the general left column).	N/A
displayMyAccountBlock	Display extra information inside the "my account" block	N/A
displayMyAccountBlockFooter	Display extra information inside the "my account" block footer	N/A
displayOrderConfirmation	Called on order confirmation page	<pre>array('total_to_pay' => (float) Total amount with tax, 'currency' => (string) Currency sign, 'objOrder' => (object) Order object, 'currencyObj' => (object) Currency object);</pre>
displayOrderDetail	Displayed on order detail on front office	<pre>array('order' => (object) Order object);</pre>
displayPayment	Displayed on payment page	N/A
displayPaymentReturn	Displayed on payment return page	<pre>array('total_to_pay' => (float) Total amount with tax, 'currency' => (string) Currency sign, 'objOrder' => (object) Order object, 'currencyObj' => (object) Currency object);</pre>
displayPaymentTop	Top of payment page	N/A
displayPDFInvoice	Allow the display of extra information into the PDF invoice	N/A
displayProductButtons	Put new action buttons on product page	N/A
displayProductComparison	Extra Product Comparison	N/A
displayProductTab	Called on order product page tabs	N/A
displayProductTabContent	Called on order product page tabs	<pre>array('product' => (object) Product object);</pre>
displayRightColumn	Displayed in right column.	<pre>array('cart' => (object) Cart object);</pre> <p>Note that the Cart object can also be retrieved from the current Context.</p>
displayRightColumnProduct	Display in right column on product page (not to be confused with the general right column).	N/A

displayShoppingCart	Display some specific information	<pre>array('delivery' => \$delivery, 'delivery_state' => State::getNameById (\$delivery->id_state), 'invoice' => \$invoice, 'invoice_state' => State::getNameById(\$invoice- >id_state), 'formattedAddresses' => \$formatted_addresses, 'products' => array_values(\$products), 'gift_products' => \$gift_products, 'discounts' => \$cart_rules, 'is_virtual_cart' => (int)\$this- >isVirtualCart(), 'total_discounts' => \$total_discounts, 'total_discounts_tax_exc' => \$total_discounts_tax_exc, 'total_wrapping' => \$this->getOrderTotal(true, Cart::ONLY_WRAPPING), 'total_wrapping_tax_exc' => \$this->getOrderTotal (false, Cart::ONLY_WRAPPING), 'total_shipping' => \$total_shipping, 'total_shipping_tax_exc' => \$total_shipping_tax_exc, 'total_products_wt' => \$total_products_wt, 'total_products' => \$total_products, 'total_price' => \$base_total_tax_inc, 'total_tax' => \$total_tax, 'total_price_without_tax' => \$base_total_tax_exc, 'is_multi_address_delivery' => \$this- >isMultiAddressDelivery() ((int)Tools::getValue ('multi-shipping') == 1), 'free_ship' => \$total_shipping ? 0 : 1, 'carrier' => new Carrier(\$this->id_carrier, \$id_lang),);</pre> <p>Can be found in /classes/Cart.php</p>
displayShoppingCartFooter	Display some specific information on the shopping cart page	<pre>array('delivery' => \$delivery, 'delivery_state' => State::getNameById (\$delivery->id_state), 'invoice' => \$invoice, 'invoice_state' => State::getNameById(\$invoice- >id_state), 'formattedAddresses' => \$formatted_addresses, 'products' => array_values(\$products), 'gift_products' => \$gift_products, 'discounts' => \$cart_rules, 'is_virtual_cart' => (int)\$this- >isVirtualCart(), 'total_discounts' => \$total_discounts, 'total_discounts_tax_exc' => \$total_discounts_tax_exc, 'total_wrapping' => \$this->getOrderTotal(true, Cart::ONLY_WRAPPING), 'total_wrapping_tax_exc' => \$this->getOrderTotal (false, Cart::ONLY_WRAPPING), 'total_shipping' => \$total_shipping, 'total_shipping_tax_exc' => \$total_shipping_tax_exc, 'total_products_wt' => \$total_products_wt, 'total_products' => \$total_products, 'total_price' => \$base_total_tax_inc, 'total_tax' => \$total_tax, 'total_price_without_tax' => \$base_total_tax_exc, 'is_multi_address_delivery' => \$this- >isMultiAddressDelivery() ((int)Tools::getValue ('multi-shipping') == 1), 'free_ship' => \$total_shipping ? 0 : 1, 'carrier' => new Carrier(\$this->id_carrier, \$id_lang),);</pre> <p>Can be found in /classes/Cart.php</p>

Most of these hooks are the same as they were in version 1.4 of PrestaShop, but have been renamed with either an "action" or "display" prefix in order to be compatible with the new 1.5 naming scheme.

The following "updated hooks" lists were automatically generated by a script. Hook descriptions appear as they are in the PrestaShop database, and can sometimes be missing.

Old hook name	New hook name	Hook description
AdminStatsModules	displayAdminStatsModules	
GraphEngine	displayAdminStatsGraphEngine	
GridEngine	displayAdminStatsGridEngine	
PDFInvoice	displayPDFInvoice	Allow the display of extra information into the PDF invoice
addproduct	actionProductAdd	
adminCustomers	displayAdminCustomers	Launch modules when the tab AdminCustomers is displayed on back-office.
adminOrder	displayAdminOrder	Launch modules when the tab AdminOrder is displayed on back-office.
afterCreateHtaccess	actionHtaccessCreate	After <code>.htaccess</code> creation
afterDeleteAttribute	actionAttributeDelete	On deleting attribute feature value
afterDeleteAttributeGroup	actionAttributeGroupDelete	On deleting attribute group
afterDeleteFeature	actionFeatureDelete	On deleting attribute feature
afterDeleteFeatureValue	actionFeatureValueDelete	On deleting attribute feature value
afterSaveAdminMeta	actionAdminMetaSave	After save configuration in AdminMeta
afterSaveAttribute	actionAttributeSave	On saving attribute feature value
afterSaveAttributeGroup	actionAttributeGroupSave	On saving attribute group
afterSaveFeature	actionFeatureSave	On saving attribute feature
afterSaveFeatureValue	actionFeatureValueSave	On saving attribute feature value
afterSaveProduct	actionProductSave	On saving products
attributeForm	displayAttributeForm	Add fields to the form "attribute value"

attributeGroupForm	displayAttributeGroupForm	Add fields to the form "attribute group"
authentication	actionAuthentication	
backBeforePayment	displayBeforePayment	Redirect user to the module instead of displaying payment modules
backOfficeFooter	displayBackOfficeFooter	
backOfficeHeader	displayBackOfficeHeader	
backOfficeHome	displayBackOfficeHome	
backOfficeTop	displayBackOfficeTop	
beforeAuthentication	actionBeforeAuthentication	Before authentication
beforeCarrier	displayBeforeCarrier	This hook is display before the carrier list on Front office
cancelProduct	actionProductCancel	This hook is called when you cancel a product in an order
cart	actionCartSave	
categoryAddition	actionCategoryAdd	
categoryDeletion	actionCategoryDelete	
categoryUpdate	actionCategoryUpdate	
createAccount	actionCustomerAccountAdd	Called when a new customer successfully creates an account
createAccountForm	displayCustomerAccountForm	Display some information on the form to create a customer account
createAccountTop	displayCustomerAccountFormTop	
customerAccount	displayCustomerAccount	Display on page account of the customer
deleteProductAttribute	actionProductAttributeDelete	
deleteproduct	actionProductDelete	This hook is called when a product is deleted
extraCarrier	displayCarrierList	
extraLeft	displayLeftColumnProduct	
extraProductComparison	displayProductComparison	Extra Product Comparison
extraRight	displayRightColumnProduct	

featureForm	displayFeatureForm	Add fields to the form "feature"
featureValueForm	displayFeatureValueForm	Add fields to the form "feature value"
footer	displayFooter	Add block in footer
header	displayHeader	A hook which allow you to do things in the header of each pages
home	displayHome	
invoice	displayInvoice	Add blocks to invoice (order)
leftColumn	displayLeftColumn	
myAccountBlock	displayMyAccountBlock	Display extra information inside the "my account" block
newOrder	actionValidateOrder	
orderConfirmation	displayOrderConfirmation	Called on order confirmation page
orderDetail	actionOrderDetail	To set the follow-up in Smarty when order detail is called
orderDetailDisplayed	displayOrderDetail	Displayed on order detail on front office
orderReturn	actionOrderReturn	
orderSlip	actionOrderSlipAdd	Called when a quantity of one product change in an order.
payment	displayPayment	
paymentCCAdded	actionPaymentCCAdd	Payment CC added
paymentConfirm	actionPaymentConfirmation	
paymentReturn	displayPaymentReturn	
paymentTop	displayPaymentTop	Top of payment page
postProcessAttribute	actionAttributePostProcess	On post-process in admin feature value
postProcessAttributeGroup	displayAttributeGroupPostProcess	On post-process in admin attribute group
postProcessFeature	displayFeaturePostProcess	On post-process in admin feature
postProcessFeatureValue	displayFeatureValuePostProcess	On post-process in admin feature value
postUpdateOrderStatus	actionOrderStatusPostUpdate	
processCarrier	actionCarrierProcess	

productActions	displayProductButtons	Put new action buttons on product page
productListAssign	actionProductListOverride	Assign product list to a category
productOutOfStock	actionProductOutOfStock	Make action while product is out of stock
productTab	displayProductTab	Called on order product page tabs
productTabContent	displayProductTabContent	Called on order product page tabs
productfooter	displayFooterProduct	Add new blocks under the product description
rightColumn	displayRightColumn	
search	actionSearch	
shoppingCart	displayShoppingCartFooter	Display some specific information on the shopping cart page
shoppingCartExtra	displayShoppingCart	Display some specific information
taxManager	actionTaxManager	
top	displayTop	A hook which allow you to do things at the top of each pages.
updateCarrier	actionCarrierUpdate	This hook is called when a carrier is updated
updateOrderStatus	actionOrderStatusUpdate	Launch modules when the order's status changes.
updateProductAttribute	actionProductAttributeUpdate	
updateQuantity	actionUpdateQuantity	Quantity is updated only when the customer effectively place his order.
updateproduct	actionProductUpdate	
watermark	actionWatermark	

Front-office and back-office

These hooks are valid for both v1.4 and 1.5 of PrestaShop.

Existing front-office hooks

Home page and general site pages

Hook name	Description	What to expect in \$params? (Pseudocode)
displayHeader	Called within the HTML <head> tags. Ideal location for adding JavaScript and CSS files.	N/A
displayTop	Called in the page's header.	N/A
displayLeftColumn	Called when loading the left column.	N/A

displayRightColumn	Called when loading the right column.	<pre>array('cart' => (object) Cart object);</pre> <p>Note that the <code>Cart</code> object can also be retrieved from the current <code>Context</code>.</p>
displayFooter	Called in the page's footer.	N/A
displayHome	Called at the center of the homepage.	N/A

Product page

Hook name	Description	What to expect in <code>\$params</code> ? (Pseudocode)
displayLeftColumnProduct	Called right before the "Print" link, under the picture.	N/A
displayRightColumnProduct	Called right after the block for the "Add to Cart" button.	N/A
displayProductButtons	Called inside the block for the "Add to Cart" button, right after that button.	N/A
actionProductOutOfStock	Called inside the block for the "Add to Cart" button, right after the "Availability" information.	N/A
displayFooterProduct	Called right before the tabs.	N/A
displayProductTab	Called in tabs list, such as "More info", "Data sheet", "Accessories", etc.	N/A
displayProductTabContent	Called when a tab is clicked.	<pre>array('product' => (object) Product object);</pre>

Cart page

Hook name	Description	What to expect in <code>\$params</code> ? (Pseudocode)
actionCartSave	Called right after a cart creation or update.	N/A

displayShoppingCartFooter	Called right below the cart items table.	<pre>array('delivery' => \$delivery, 'delivery_state' => State::getNameById(\$delivery->id_state), 'invoice' => \$invoice, 'invoice_state' => State::getNameById(\$invoice->id_state), 'formattedAddresses' => \$formatted_addresses, 'products' => array_values(\$products), 'gift_products' => \$gift_products, 'discounts' => \$cart_rules, 'is_virtual_cart' => (int)\$this->isVirtualCart(), 'total_discounts' => \$total_discounts, 'total_discounts_tax_exc' => \$total_discounts_tax_exc, 'total_wrapping' => \$this->getOrderTotal(true, Cart::ONLY_WRAPPING), 'total_wrapping_tax_exc' => \$this->getOrderTotal(false, Cart::ONLY_WRAPPING), 'total_shipping' => \$total_shipping, 'total_shipping_tax_exc' => \$total_shipping_tax_exc, 'total_products_wt' => \$total_products_wt, 'total_products' => \$total_products, 'total_price' => \$base_total_tax_inc, 'total_tax' => \$total_tax, 'total_price_without_tax' => \$base_total_tax_exc, 'is_multi_address_delivery' => \$this->isMultiAddressDelivery() ((int)Tools::getValue('multi-shipping') == 1), 'free_ship' => \$total_shipping ? 0 : 1, 'carrier' => new Carrier(\$this->id_carrier, \$id_lang),);</pre> <p>Can be found in /classes/Cart.php</p>
displayShoppingCart	Called after the cart's table of items, right above the navigation buttons.	<pre>array('delivery' => \$delivery, 'delivery_state' => State::getNameById(\$delivery->id_state), 'invoice' => \$invoice, 'invoice_state' => State::getNameById(\$invoice->id_state), 'formattedAddresses' => \$formatted_addresses, 'products' => array_values(\$products), 'gift_products' => \$gift_products, 'discounts' => \$cart_rules, 'is_virtual_cart' => (int)\$this->isVirtualCart(), 'total_discounts' => \$total_discounts, 'total_discounts_tax_exc' => \$total_discounts_tax_exc, 'total_wrapping' => \$this->getOrderTotal(true, Cart::ONLY_WRAPPING), 'total_wrapping_tax_exc' => \$this->getOrderTotal(false, Cart::ONLY_WRAPPING), 'total_shipping' => \$total_shipping, 'total_shipping_tax_exc' => \$total_shipping_tax_exc, 'total_products_wt' => \$total_products_wt, 'total_products' => \$total_products, 'total_price' => \$base_total_tax_inc, 'total_tax' => \$total_tax, 'total_price_without_tax' => \$base_total_tax_exc, 'is_multi_address_delivery' => \$this->isMultiAddressDelivery() ((int)Tools::getValue('multi-shipping') == 1), 'free_ship' => \$total_shipping ? 0 : 1, 'carrier' => new Carrier(\$this->id_carrier, \$id_lang),);</pre> <p>Can be found in /classes/Cart.php</p>
displayCustomerAccountFormTop	Called within the client account creation form, right above the "Your personal information" block.	N/A

displayCustomerAccountForm	Called within the client account creation form, right before the "Register" button.	N/A
actionCustomerAccountAdd	Called right after the client account creation.	N/A
displayCustomerAccount	Called on the client account homepage, after the list of available links. Ideal location to add a link to this list.	N/A
displayMyAccountBlock	Called within the "My account" block, in the left column, below the list of available links. This is the ideal location to add a link to this list.	N/A
displayMyAccountBlockFooter	Displays extra information inside the "My account" block.	N/A
actionAuthentication	Called right after the client identification, only if the authentication is valid (e-mail address and password are both OK).	N/A
actionBeforeAuthentication	Called right before authentication.	N/A

Search page

Hook name	Description	What to expect in \$params? (Pseudocode)
actionSearch	Called after a search is performed. Ideal location to parse and/or handle the search query and results.	<pre>array('expr' => (string) Search query, 'total' => (int) Amount of search results);</pre>

Carrier choice page

Hook name	Description	What to expect in \$params? (Pseudocode)
-----------	-------------	--

displayBeforeCarrier	Displayed before the carrier list on front-office.	<pre> array('carriers' => array(array('name' => (string) Name, 'img' => (string) Image URL, 'delay' => (string) Delay text, 'price' => (float) Total price with tax, 'price_tax_exc' => (float) Total price without tax, 'id_carrier' => (int) intified option delivery identifier, 'id_module' => (int) Module ID), 'checked' => (int) intified selected carriers, 'delivery_option_list' => array(array(0 => array(// First address '12,' => array(// First delivery option available for this address carrier_list => array(12 => array(// First carrier for this option 'instance' => Carrier Object, 'logo' => <url to the carrier's logo>, 'price_with_tax' => 12.4, // Example 'price_without_tax' => 12.4, // Example 'package_list' => array(1, // Example 3, // Example),),), is_best_grade => true, // Does this option have the biggest grade (quick shipping) for this shipping address is_best_price => true, // Does this option have the lower price for this shipping address unique_carrier => true, // Does this option use a unique carrier total_price_with_tax => 12.5, total_price_without_tax => 12.5, position => 5, // Average of the carrier position),),), 'delivery_option' => array('<id_address>' => Delivery option, ...)); </pre> <p>NOTE: intified means an array of integers 'intified' by Cart::intifier</p>
displayCarrierList	Called after the list of available carriers, during the order process. Ideal location to add a carrier, as added by a module.	<pre> array('address' => (object) Address object); </pre>

Payment page

Hook name	Description	What to expect in \$params? (Pseudocode)
displayPaymentTop	Top of payment page.	N/A
displayPayment	Called when needing to build a list of the available payment solutions, during the order process. Ideal location to enable the choice of a payment module that you have developed.	N/A

displayPaymentReturn	Called when the user is sent back to the store after having paid on the 3rd-party website. Ideal location to display a confirmation message or to give some details on the payment.	<pre>array('total_to_pay' => (float) Total amount with tax, 'currency' => (string) Currency sign, 'objOrder' => (object) Order object, 'currencyObj' => (object) Currency object);</pre>
displayOrderConfirmation	A duplicate of paymentReturn.	<pre>array('total_to_pay' => (float) Total amount with tax, 'currency' => (string) Currency sign, 'objOrder' => (object) Order object, 'currencyObj' => (object) Currency object);</pre>
displayBeforePayment	Called when displaying the list of available payment solutions. Ideal location to redirect the user instead of displaying said list (i.e., 1-click PayPal checkout).	<pre>array('module' => (string) Module link [e.g. /module/test /front]);</pre>

Order page

Hook name	Description	What to expect in \$params? (Pseudocode)
actionOrderReturn	Called when the customer request to send his merchandise back to the store, and if now error occurs.	<pre>array('orderReturn' => (object) OrderReturn object);</pre>
displayPDFInvoice	Called when displaying the invoice in PDF format. Ideal location to display content within the invoice.	N/A

Mobile theme hooks

All these hooks are specific to PrestaShop's default mobile theme.

The four first ones are in PrestaShop's internal hooks database, so the user can attach an action /interface to these hooks using the Modules > Positions page.

All the other ones exist in the mobile theme TPL files, but are not in the internal hooks database. A developer can still use them to attach content to specific portion of the code.

Hook name	Description
displayMobileTopSiteMap	
displayMobileHeader	
displayMobileShoppingCartTop	
displayMobileAddToCartTop	
<i>displayMobileShoppingCartTop</i>	
displayMobileShoppingCartBottom	
displayMobileTopSiteMap	
displayMobileFooterChoice	
displayMobileShoppingCartTop	
displayMobileShoppingCartBottom	

displayMobileIndex	
mobileCustomerAccount	
displayMobileTop	
displayMobileAddToCartTop	
displayMobileShoppingCartTop	
displayMobileShoppingCartButton	
displayMobileShoppingCartBottom	
displayMobileShoppingCartTop	
displayMobileShoppingCartBottom	

Existing back-office hooks

General hooks

Hook name	Description	What to expect in \$params? (Pseudocode)
displayBackOfficeTop	Called within the header, above the tabs.	N/A
displayBackOfficeHeader	Called between the HEAD tags. Ideal location for adding JavaScript and CSS files.	N/A
displayBackOfficeFooter	Called within the page footer, above the "Power By PrestaShop" line.	N/A
displayBackOfficeHome	Called at the center of the homepage.	N/A

Orders and order details

Hook name	Description	What to expect in \$params? (Pseudocode)
actionValidateOrder	Called during the new order creation process, right after it has been created.	<pre>array('cart' => (object) Cart object, 'order' => (object) Order object, 'customer' => (object) Customer object, 'currency' => (object) Currency object, 'orderStatus' => (object) OrderState object);</pre>
actionPaymentConfirmation	Called when an order's status becomes "Payment accepted".	<pre>array('id_order' => (int) Order ID);</pre>
actionOrderStatusUpdate	Called when an order's status is changed, right before it is actually changed.	<pre>array('newOrderStatus' => (object) OrderState object, 'id_order' => (int) Order ID);</pre>
actionOrderStatusPostUpdate	Called when an order's status is changed, right after it is actually changed.	<pre>array('newOrderStatus' => (object) OrderState object, 'id_order' => (int) Order ID);</pre>
actionProductCancel	Called when an item is deleted from an order, right after the deletion.	<pre>array('order' => (object) Order object, 'id_order_detail' => (int) Order Detail ID);</pre>

displayInvoice	Called when the order's details are displayed, above the Client Information block.	N/A
displayAdminOrder	Called when the order's details are displayed, below the Client Information block.	array('id_order' = (int) Order ID);
actionOrderSlipAdd	Called during the creation of a credit note, right after it has been created.	array('order' => Order object, 'productList' => array((int) product ID 1, (int) product ID 2, ..., (int) product ID n), 'qtyList' => array((int) quantity 1, (int) quantity 2, ..., (int) quantity n)); The order of IDs and quantities is important!

Products

Hook name	Description	What to expect in \$params? (Pseudocode)
actionProductSave	Called when saving products.	array('id_product' => (int) Product ID);
actionUpdateQuantity	Called during an the validation of an order, the status of which being something other than "canceled" or "Payment error", for each of the order's items.	array('id_product' => (int) Product ID, 'id_product_attribute' => (int) Product attribute ID, 'quantity' => (int) New product quantity);
actionProductAttributeUpdate	Called when a product declination is updated, right after said update.	array('id_product_attribute' => (int) Product attribute ID);
actionProductAttributeDelete	Called when a product declination is deleted.	array('product' => (object) Product object);
actionWatermark	Called when an image is added to a product, right after said addition.	array('id_image' => (int) Image ID, 'id_product' => (int) Product ID);
displayAttributeForm	Add fields to the form "attribute value".	N/A
displayAttributeGroupForm	Add fields to the form "attribute group".	N/A
displayAttributeGroupPostProcess	Called when post-process in admin attribute group.	N/A
displayFeatureForm	Add fields to the form "feature".	N/A

displayFeaturePostProcess	Called when post-process in admin feature.	N/A
displayFeatureValueForm	Add fields to the form "feature value".	N/A
displayFeatureValuePostProcess	Called when post-process in admin feature value.	N/A

Statistics

Hook name	Description	What to expect in \$params? (Pseudocode)
displayAdminStatsGraphEngine	Called when a stats graph is displayed.	N/A
displayAdminStatsGridEngine	Called when the grid of stats is displayed.	N/A
displayAdminStatsModules	Called when the list of stats modules is displayed.	N/A

Clients

Hook name	Description	What to expect in \$params? (Pseudocode)
displayAdminCustomers	Called when a client's details are displayed, right after the list of the clients groups the current client belongs to.	N/A

Carriers

Hook name	Description	What to expect in \$params? (Pseudocode)
actionCarrierUpdate	Called during a carrier's update, right after said update.	<pre>array('id_carrier' => (int) Current Carrier ID, 'carrier' => (object) New Carrier object);</pre>