

VirtueMart Template Reference

This appendix provides a complete list of VirtueMart templates in the default theme. You will find lots of useful information for working with the templates which may help you decide which template to hack into and whether a certain function can be added to the template.

In the default theme, the templates are placed into seven subdirectories. This is a natural way of organizing the templates into template groups, with one subdirectory corresponding to one template group. In the reference, there is a brief introduction for each group, giving the basic structure of the group and some of its common properties. Then all the templates in that group will be listed with all details given.

Since the templates may not be completely independent of each other in the template detail, I try to provide more background information of that template before listing out all the available fields. The meaning of each subsection in the template is explained as follows:

- Purpose and usage: This simply states the purpose of the template and how it is used in the template system.
- Parent templates: If the template is used by another template, its parent templates are listed here.
- Child templates: If the template will use other templates, the so-called child templates will be listed here.

- **Related page/class files:** As explained in *Chapter 1, The VirtueMart Engine*, there are two major file types in VirtueMart that are related to frontend presentation. The class files located in the `classes` subdirectory are used to provide functions and communicate with the database. The page file located in the `html` subdirectory is used to control the actual presentation of the pages. A template may be used in a page file or in a class file (and sometimes invoked from Joomla! modules or another template file as well). The related page and class files are listed here. In case you want to know how the template is actually used, you can check on the page or class file.
- **Available fields:** VirtueMart will pass a number of variables to the template to help in its presentation. These variables are known as available fields. For the basic types of available field, you can check out *Chapter 2, The VirtueMart Default Theme*. Not all available fields are used in the `default` theme templates but you can find the complete list here.

Probably because creating a new template is quite costly in terms of programming resources, VirtueMart will usually reuse the same template object for all the templates on the same page. The `shop.browse` page, for example, uses close to ten different templates. This number does not include the repeated use of the core browse template for each single product in the product list. A consequence of this strategy is that available fields in a template can change with the order that templates are invoked. Because VirtueMart usually does not unset variables created for a previous template, it is difficult to say which fields are intended to pass to a specific template or just pass down because it is used in a previous template. Most of the time, fields inherited from previous usage may not be useful in a new template, but sometimes it may become handy when you are doing special customization. However, just be warned that those fields may not be available in a future version.

Global variables

Because Virtuemart will inject all global variables into the template, all of the following global variables become available fields that you can use in a template:

`$auth`

An array of user permission values. It contains the following elements:

- `show_price`: Whether to show prices to this user.
- `user_id`: The user identifier of integer value. 0 if not logged in.
- `username`: Login username. Demo if not logged in.
- `perms`: Array of permission. Can contain a combination of possible user roles.

- `first_name`: First name of user. Guest if not logged in.
- `last_name`: Last name of user. An empty string if not logged in.
- `shopper_group_id`: Shopper group identifier for the user. An integer value.
- `shopper_group_discount`: The discount percentage for the shopper group.
- `show_price_including_tax`: Whether to show price including tax for the user. 1 if true.
- `default_shopper_group`: Whether the user belongs to the default shopper group. 1 if true.
- `is_registered_customer`: Whether the user has registered for a VirtueMart account. 1 if true.

`$cart`

An array containing all the products placed in the shop basket.

`$CURRENCY`

An instance of `convertECB` object used for currency conversion.

`$CURRENCY_DISPLAY`

An instance of `CurrencyDisplay` object used to display the price and other monetary value in the currency settings as defined for the default vendor.

`$func`

The value of the `func` parameter in the URL request. It is the alias of the function that needs to be executed before passing the control to the requested page. When you click the **add-to-cart** button without using the Ajax update function, the `func` will be `addCart`. This will ask VirtueMart to execute the `addCart` function before passing the control to `shop.cart` to show the Shop Cart page.

`$Itemid`

The value of the `Itemid` parameter in the Request URL. Used to identify the menu used to request the current page.

`$keyword`

The value of the keyword request parameter.

`$limit`

The value of the number of items to display. Used only for Product Listing page together with `$limitstart`.

`$limitstart`

The position of the first item to display within the search result. Used only for Product Listing page together with `$limit`.

`$mainframe`

An instance of `JSite` object. Used to provide various basic functions for the site. Please refer to Joomla! 1.5 documentation for detail.

`$mm_action_url`

The value of the site URL, starts with `https://` for secure URL access and `http://` otherwise.

`$modulename`

The module name part of the page parameter in the URL request.

`$mosConfig_absolute_path`

The value of the root directory path of the Joomla! site.

`$mosConfig_lang`

The value of the default language for the Joomla! site.

`$mosConfig_live_site`

The value of the root URL of the Joomla! site, including the domain name.

`$my`

An instance of the `JUser` object. It represents the current user. Please refer to Joomla! 1.5 documentation for detail.

`$option`

The value of the `option` parameter in the Request URL. For VirtueMart pages should be `com_virtuemart`, but since templates can also be used by Joomla! module. This can contain other possible value.

`$page`

The value of the `page` parameter in the URL request. It is in the form `module.page`, where `module` is the module name and `page` is the page name.

`$pagename`

The page name part of the `page` parameter in the URL request.

`$perm`

An instance of the `ps_perm` object which provides functions to handle user permissions. It provides easy access to all the `ps_perm` functions.

`$ps_checkout`

An instance of the `ps_checkout` object. Contains the function for processing checkout.

`$ps_html`

An instance of the `ps_html` object used to provide VirtueMart HTML functions.

`$ps_payment_method`

An instance of the `ps_payment_method` object. Will contain value only for payment method.

`$ps_vendor_id`

The value of the `vendor_id`. The default value is 1.

`$PSHOP_SHIPPING_MODULES`

This contains the value of the constant `PSHOP_SHIPPING_MODULES` which is an array of the shipping Modules activated for the site.

`$sess`

An instance of the `ps_sess` object which provides functions to handle user session. It provides easy access to all `ps_sess` functions.

`$vars`

This is a variable created by VirtueMart for passing the content in the `$_REQUEST` variable for use in a module function. You can treat this as an equivalent to the `$_REQUEST` global variable.

`$vendor_image`, `$vendor_country_2_code`, `$vendor_country_3_code`,
`$vendor_state_name`, `$vendor_image_url`, `$vendor_name`, `$vendor_address`,
`$vendor_address_2`, `$vendor_city`, `$vendor_country`, `$vendor_mail`, `$vendor_store_name`, `$vendor_state`, `$vendor_zip`, `$vendor_phone`, `$vendor_currency`,
`$vendor_store_desc`, `$vendor_freeshipping`, `$vendor_currency_display_style`, `$vendor_date_format`, `$vendor_address_format`

These variables contain the corresponding values for the vendor as defined in the **Store/Edit Store** page in the VirtueMart backend.

`$VM_BROWSE_ORDERBY_FIELDS`

This contains the value of the constant `VM_BROWSE_ORDERBY_FIELDS` which is an array of allowed sort order fields for the Product List page.

`$vmInputFilter`

An instance of the `vmInputFilter` object which provides all sorts of function to sanitize request data. Please check the `classes/phpInputFilter/class.inputfilter.php` file for detail.

`$VM_LANG`

An instance of the `vmLanguage` object which provides functions to handle the language elements. This object is used so the template can print out any language element by using the `_()` function.

`$VM_MODULES_FORCE_HTTPS`

This contains the value of the constant `VM_MODULES_FORCE_HTTPS`, which is an array of VirtueMart modules that must be accessed using `https`.

`$vmLogger`

An instance of `vmLog_composite` object which provides the functions for logging VirtueMart messages. It is there so that the template can freely add information, warning, error or debugging messages to the log or show them on the browser.

1. Basket template group

Basket template group includes templates that are used in the display of shop basket. There are four core basket templates. They are named `basket_b2b.html.php`, `basket_b2c.html.php`, `ro_basket_b2b.html.php`, and `ro_basket_b2c.html.php`. The first two are used to produce the HTML for the editable shop basket. The last two are used to produce read-only shop basket where shop cart quantities cannot be changed.

1.1 `basket_b2b.html.php`

Purpose and usage

Used to display the shop basket content in the shop cart page and checkout pages.

Parent templates

None.

Child templates

common/couponField.tpl.php.

Related page/class files

shop.cart.php, basket.php, shop.index.php.

Available fields

`$coupon_display`

The text to be displayed in the coupon discount area. Most probably currency symbol + discount amount.

`$discount_after`

Whether there is a display for discount after tax. Can be true or false.

`$discount_before`

Whether there is a display for discount before tax. Can be true or false.

`$order_total_display`

The text to be displayed in the order total area. Most probably currency symbol + order total amount.

`$paypal_express_button`

The HTML code for the paypal express button.

`$product_rows`

An array of products in the shop basket. Each product consists of eight elements:

- `product_sku`: Value of `product_sku` as stored in database.
- `product_name`: Value of `product_name` as stored in database.
- `product_attributes`: Description text listing all the attributes the shopper selected for this product. An empty string if product does not have attribute.
- `product_price`: Text to be displayed in price. Price calculated with attribute and tax taken into consideration. Should be currency symbol + amount.

- `subtotal`: Text to be displayed as subtotal. Should be currency symbol + amount.
- `subtotal_with_tax`: Text to be displayed as subtotal + tax. Should be currency symbol + amount.
- `update_form`: HTML for the form to update the product including the quantity box.
- `delete_form`: HTML for the form to delete the product from basket.

`$shipping`

Whether shipping should be shown or not. Can be true or false.

`$shipping_display`

The text to be displayed in the shipping area. Most probably currency symbol + shipping amount.

`$show_tax`

Whether tax should be shown or not. Can be true or false.

`$subtotal_display`

The text to be displayed in the subtotal area. Most probably currency symbol + subtotal amount.

`$tax_display`

The text to be displayed in the tax area. Most probably currency symbol + tax amount.

1.2 basket_b2c.html.php

Same as `basket_b2b.html.php`.

1.3 ro_basket_b2b.html.php

Purpose and usage

Used to display the shop basket content in a read-only format. The quantity is not editable. Used in the final confirmation step during checkout.

Parent templates

None

Child templates

common/couponField.tpl.php.

Related page/class files

ro_basket.php, checkout.index.php.

Available fields

`$coupon_display`, `$discount_after`, `$discount_before`, `$order_total_display`, `$product_rows`, `$shipping`, `$shipping_display`, `$show_tax`, `$subtotal_display`, `$tax_display`

Same as `basket_b2b_html.php`.

`$discount_word`

The text to be displayed for the label of the payment discount. Can be **discount** or **additional fee**.

`$payment_discount_after`

Whether there is a display for payment discount after tax. Can be true or false.

`$payment_discount_before`

Whether there is a display for payment discount before tax. Can be true or false.

`$payment_discount_display`

The text to be displayed in the payment discount area. Most probably currency symbol + payment discount amount.

1.4 ro_basket_b2c.html.php

Same as `ro_basket_b2b.html.php`.

1.5 includes/paypal_express.tpl.php

Purpose and usage

Used to display the PayPal expression button and link.

Parent templates

None.

Child templates

None.

Related page/class files

basket.php.

Available fields

`$ppex_img_iso`

The ISO language identifier. Used to locate the PayPal button image.

2. Browse template group

The browse template group contains templates related to the display of the Product List page. The templates are used in several scenarios: for products in a category, products related to a manufacturer, product search result, and also the all product list. There are two subgroups within this template group. The templates in the root directory, also known as core browse templates, are for displaying individual products. The templates in the `includes` subdirectory are those for the display of the whole Product List page or an individual element within the core browse template.

2.1 browse_1.tpl.php

Purpose and usage

Used to format the display of one single product in the Product List page.

Parent templates

`browse_layouttable.tpl.php`, `browse_notables.tpl.php`.

Child templates

Use `price.tpl.php` and `includes/addtocart_form` indirectly.

Related page/class files

`shop.browse.php`.

Available fields

All available fields in `browse_layouttable.tpl.php` template, `browse_addtocart_form.tpl.php` template are also available in all core browse templates.

`$files`

The `files` element (which is also an array) of the result array from calling `ps_product_files::getFilesForProduct()`. Each element of the `$files` array contains the following elements (corresponding to the columns stored in the `vm_product_files` datatable):

- `file_id`: The file identifier or database key to retrieve the file data.
- `file_product_id`: The `product_id` that the file is related to.
- `file_name`: The actual filename of the file.
- `file_title`: The title of the file for display purposes.
- `file_description`: The description of the file.
- `file_extension`: The file extension.
- `file_mimetype`: The file type as classified in MIME.
- `file_url`: The URL for accessing the file.
- `file_published`: Whether the file is published. Should be 1 to be included in this array.
- `file_is_image`: Whether the file is an image file. Should be 0 to be included in this array.
- `file_image_height`: Not used for this type of file.
- `file_file_image_width`: Not used for this type of file.

- `file_image_thumb_height`: Not used for this type of file.
- `file_image_thumb_width`: Not used for this type of file.

`$images`

The `files` element (which is also an array) of the result array from calling `ps_product_files::getFilesForProduct()`. Each element of the `$images` array contains the following elements (all the fields are the same as `$files` above except as noted):

- `file_is_image`: Whether the file is an image file. Should be 1 to be included in this array.
- `file_image_height`: Height of full image in pixels.
- `file_file_image_width`: Width of full image in pixels.
- `file_image_thumb_height`: Height of thumb image in pixels.
- `file_image_thumb_width`: Width of thumb image in pixels.

`$form_addtocart`

The HTML for the add-to-cart form returned from invoking the template `addtocart_form.tpl.php`.

`$has_addtocart`

Whether the variable `$form_addtocart` has content. 1 for yes and 0 for no.

`$product_flypage`

The internal URL used to show the product details page.

`$product_thumb_image`

The processed value for the product thumb image location. Can be from an external site or internal site or the `no_image` graphic. This may be different from the value stored in the database table.

`$product_full_image`

The processed value for the product full image location. Can be from an external site or internal site or the `no_image` graphic. This may be different from the value stored in the database table.

`$full_image_width`

The width for the full image in pixels.

`$full_image_height`

The height for the full image in pixels.

`$product_name`

The value of `product_name` as recorded in the database. This is the name of the product.

`$product_s_desc`

The value of `product_s_desc` as recorded in the database. This is supposed to be the short description for the product but can be used for storing other information.

`$product_details`

The `<a>` tag that links to the flypage. Contains the text "Product Details" that points to the flypage.

`$product_rating`

The HTML for product rating as returned by invoking the `common/votes_allvotes.tpl.php` template. This will be an empty string for PDF template.

`$product_price`

The HTML for price as returned by invoking the `common/price.tpl.php` template. Includes original price, price after discount, and the saved value, all formatted with currency symbol and money format defined for the vendor.

`$product_price_raw`

The raw price of the product as returned by invoking the function `ps_product->get_adjusted_attribute_price()`.

`$product_sku`

The value of `product_sku` as recorded in the database. This is the human comprehensible product identifier.

`$product_weight`

The value of `product_weight` as recorded in the database. Supposed to be the weight of the product as an integer and used for calculating shipping. Can be used for storing other information if it is not used by shipping and virtual tax.

`$product_weight_uom`

The value of `product_weight_uom` as recorded in the database. This is the unit of measure (uom) for the weight.

`$product_length`

The value of `product_length` as recorded in the database. Supposed to be the length of the product as an integer, but can be used for storing other information.

`$product_width`

The value of `product_width` as recorded in the database. Supposed to be the width of the product as an integer, but can be used for storing other information.

`$product_height`

The value of `product_height` as recorded in the database. Supposed to be the height of the product as an integer, but can be used for storing other information.

`$product_lwh_uom`

The value of `product_lwh_uom` as recorded in the database. This is the unit of measure (uom) for length, width, and height (lwh), but can be used for storing other information.

`$product_available_date`

The value of `product_available_date` as recorded in the database formatted in date format defined in the language element `DATE_FORMAT_LC`.

`$product_availability`

The value of `product_availability` as recorded in the database. Supposed to be the text or image for showing the product availability. But can be used for storing other information.

`$cdate`

The value of `cdate` as recorded in the database formatted in date format defined in the language element `DATE_FORMAT_LC`. This is the date that the record was first created.

`$mdate`

The value of `mdate` as recorded in the database formatted in date format defined in the language element `DATE_FORMAT_LC`. This is the date that the record was last modified.

`$product_url`

The value of `product_url` as recorded in the database. Supposed to be the URL pointing to an external site for showing detail of the product.

2.2 browse_2.tpl.php

Purpose and usage

Used to format the display of one single product in the Product List page. Supposed to be used when there are two products in one row.

All other properties

Same as `browse_1.tpl.php`.

2.3 browse_3.tpl.php

Purpose and usage

Used to format the display of one single product in the Product List page. Supposed to be used when there are three products in one row.

All other properties

Same as `browse_1.tpl.php`.

2.4 browse_4.tpl.php

Purpose and usage

Used to format the display of one single product in the Product List page. Supposed to be used when there are four products in one row.

All other properties

Same as `browse_1.tpl.php`.

2.5 browse_5.tpl.php

Purpose and usage

Used to format the display of one single product in the Product List page. Supposed to be used when there are five products in one row.

All other properties

Same as `browse_1.tpl.php`.

2.6 browse_lite_pdf.tpl.php

Purpose and usage

Used to format the display of one single product in the Product List page. Supposed to be used when the Product List is displayed in a PDF file.

All other properties

Same as `browse_1.tpl.php`.

2.7 includes/addtocart_form.tpl.php

Purpose and usage

Used to format the add-to-cart form for an individual product on the Product List page.

Parent templates

All core browse templates.

Child templates

None.

Related page/class files

`shop.browse.php`.

Available fields

All available fields in header templates, `browse_orderby_form.tpl.php`, and `browse_pagenav.tpl.php` are also available in `addtocart_form.tpl.php` template. Refer to the general discussion under the *Available fields* section at the beginning of the appendix for detail.

`$product_id`

Product identifier of the product to display.

`$buttons_header`

This is the HTML as returned by invoking the template `common/buttons.tpl.php`. Supposed to show the buttons on the page header to provide PDF, print, and e-mail functions. The intention of this variable in the template is not clear. Probably, just happens to be there because of the template invoke order.

`$products_per_row`

The number of products per row in the Product Listing.

`$templatefile`

The core browse template that is used for displaying one single product. The intention of this variable in the template is not clear. Probably, just happens to be there because of the template invoke order.

`$i`

The index of the current display product is the Product Listing.

`$product_in_stock`

Whether the product is in stock. 1 for yes and 0 for no. If product is not in stock, the add to cart will not show if VirtueMart setting does not allow showing out of stock products.

`$ps_product_attribute`

An instance of the `ps_product_attribute` object used to access the function provided by the `ps_product_attribute` class. In the default theme, this is used for displaying the quantity box. (The add-to-cart form will not show in the Product List page if there are child products, advanced attributes or custom attributes.)

2.8 includes/browse_header_all.tpl.php

Purpose and usage

One of the browse header templates. Used to format the display of the header of Product List page without any request parameter.

Parent templates

`browse_layouttable.tpl.php`, `browse_notables.tpl.php`, `browse_listtable.top.php`.

Child templates

None.

Related page/class files

`shop.browse.php`.

Available fields

`$browsepage_lbl`

The heading for the Product List page: the value of the language element `PHPSHOP_BROWSE_LBL`.

2.9 includes/browse_header_category.tpl.php

Purpose and usage

One of the browse header templates. Used to format the display of the header of Product List page for a requested category.

Parent templates

`browse_layouttable.tpl.php`, `browse_notables.tpl.php`, `browse_listtable.top.php`.

Child templates

`categoryChildlist.tpl.php`.

Related page/class files

`shop.browse.php`.

Affected pages/objects

`shop.browse`.

Available fields

`$browsepage_lbl`

The heading for the Product List page: the category name for category product list.

`$categories`

An array of child categories of the displayed category. Each child category contains the following four elements:

- `category_name`: Value of the `category_name` field as stored in the `product_category` datatable for the child category.
- `category_id`: Value of the `category_id` field as stored in the `product_category` datatable for the child category.
- `category_thumb_image`: Value of the `category_thumb_image` field as stored in the `product_category` datatable for the child category. This is just the filename without directory path information.
- `number_of_products`: Number of products in the child category.

`$category_id`

The value of `category_id`. Same value as stored in `product_category` datatable.

`$category_name`

The value of `category_name`. Same value as stored in `product_category` datatable.

`$desc`

The description of the category. Its value is the retrieved directly from the product category database table.

`$navigation_childlist`

The HTML for the display of child category list returned from invoking the `common/categoryChildlist.tpl.php` template.

2.10 includes/browse_header_keyword.tpl.php

Purpose and usage

One of the browse header templates. Used to format the display of the header of Product List page during keyword search.

Parent templates

`browse_layouttable.tpl.php`, `browse_notables.tpl.php`, `browse_listtable.top.php`.

Child templates

None.

Related page/class files

`shop.browse.php`.

Available fields

`$browsepage_lbl`

The heading for the Product List page: the value of the language element `PHPSHOP_SEARCH_TITLE` + the search keyword input by user.

2.11 includes/browse_header_manufacturer.tpl.php

Purpose and usage

One of the browse header templates. Used to format the display of the header for manufacturer Product List page.

Parent templates

browse_layouttable.tpl.php, browse_notables.tpl.php, browse_listtable.top.php.

Child templates

None.

Related page/class files

shop.browse.php.

Available fields

`$browsepage_lbl`

The heading for the Product List page: the manufacturer name. It is the value of the field `mf_name` in the `manufacturer` datatable.

`$browsepage_lbltext`

The description for the manufacturer. It is the value of the field `mf_desc` in the `manufacturer` datatable.

2.12 includes/browse_layouttable.tpl.php

Purpose and usage

One of the templates for the Product List page. Used to lay out the products using a HTML table. The core browse template is used to show the individual product. This differs from the listable template which will display products in a table form without using the core browse template.

Parent templates

None.

Child templates

browse_header_all.tpl.php, browse_header_category.tpl.php, browse_header_keyword.tpl.php, browse_header_manufacturer.tpl.php, browse_1.tpl.php, browse_2.tpl.php, browse_3.tpl.php, browse_4.tpl.php, browse_5.tpl.php, browse_lite_pdf.tpl.php, browse_orderbyform.tpl.php, browse_pagenav.tpl.php, browse_searchparameter_form.tpl.php.

Related page/class files

shop.browse.php.

Available fields

Fields available in header templates, addtocart_form.tpl.php, browse_searchparameter_form.tpl.php, browse_orderbyform.tpl.php, and browse_pagenav.tpl.php are also available in the template. Refer to the general discussion under the *Available fields* section at the beginning of the appendix for details.

`$orderby_form`

The HTML of the orderby form as returned from invoking the template `browse_orderbyform.tpl.php`.

`$product_id`

The product ID of the last product in the list. This is not of any good use but is there when you need it.

`$buttons_header`

The HTML for the header buttons as returned from invoking the template `common/button.tpl.php`.

`$products_per_row`

Number of products per row.

`$templatefile`

The template file to use for displaying each individual product.

`$i`

The index of the last product + 1. Can be taken as the product count.

`$product_in_stock`

Stock of the last product in the list.

`$ps_product_attribute`

An instance of the `ps_product_attribute` object to provide functions of the `ps_product_attribute` class.

`$products`

An array of all products in the list. Each product is an array containing the variables in the core browse templates (other than the variables that are inherited from other templates). Please refer to `browse_1.tpl.php` for details.

`$browsepage_footer`

The page footer. Actually, this is just the page navigation links as returned from invoking the template `browse_pagenav.tpl.php`.

`$recent_products`

The HTML of the products visited by the shopper.

`$ps_product`

An instance of the `ps_product` object to provide functions of the `ps_product` class.

2.13 includes/browse_listtable.tpl.php

Purpose and usage

One of the templates for the Product List page. Used to format the display of a Product List page in table form instead of a list form. Each product will display as a table row.

Parent templates

None.

Child templates

browse_header_all.tpl.php, browse_header_category.tpl.php, browse_header_keyword.tpl.php, browse_header_manufacturer.tpl.php, browse_orderbyform.tpl.php, browse_pagenav.tpl.php, browse_searchparameter_form.tpl.php.

All other properties

Same as includes/browse_layouttable.tpl.php.

2.14 includes/browse_notables.tpl.php

Purpose and usage

One of the templates for the Product List page. Used to format the display of a Product List page without using an HTML table.

All other properties

Same as includes/browse_layouttable.tpl.php.

2.15 includes/browse_orderbyfields.tpl.php

Purpose and usage

Used to generate the HTML for the order by drop-down used to control the sorting order. This template is used by the `browse_orderbyform.tpl.php`.

Parent templates

`browse_orderbyform.tpl.php`.

Child templates

None.

Related page/class files

`shop.browse.php`.

Available fields

Fields available in `browse_orderbyfields.tpl.php` are the same as that in the `browse_orderbyform.tpl.php` template.

2.16 includes/browse_orderbyform.tpl.php

Purpose and usage

Used to generate the HTML for the order by form used to control the sorting order as well as the top page navigation links, if it is enabled.

Parent templates

`browse_layouttable.tpl.php`, `browse_notables.tpl.php`, `browse_listtable.tpl.php`.

Child templates

`browse_orderbyfield.tpl.php`.

Related page/class files

`shop.browse.php`.

Available fields

Fields available in header templates, `browse_searchparameter_form.tpl.php` are also available in the `browse_orderbyform.tpl.php` template. Refer to the general discussion under the *Available fields* section at the beginning of the appendix for details.

`$show_limitbox`

Whether to show the limitbox. 1 for yes and 0 for no.

`$show_top_navigation`

Whether to show the page navigation links at the top.

`$pagenav`

An instance of the `vmPageNav` object which can provide all functions of the `vmPageNav` class to output the HTML needed for the page links.

`$orderby`

The value of the sorting order selected. This can be one of the following values:

- `product_list`: Order according to the product added to the category. Since it is uncontrollable except through direct database access, this pretty much means the sorting order will be random.
- `product_name`: Sort by alphabetical order of product name.
- `product_price`: Sort by product price.
- `product_sku`: Sort by alphabetical order of product sku.
- `product_cdate`: Sort by creation date.

`$icon`

The filename for the sorting direction (ascending or descending).

`$selected`

An array specifying the state of the ascending and descending order in case a drop-down is used for the sorting direction selection. The array contains two unnamed element, either (`"selected=\"selected\"", ""`) or (`"" , "selected=\"selected\""`). This has to be used together with `$asc_desc` and `$icon`.

`$asc_desc`

An array specifying the two possible states of sorting direction in case a drop-down is used for the sorting direction selection. The array contains two unnamed elements, either (`"DESC", "ASC"`) or (`"ASC", "DESC"`). This has to be used together with `$selected` and `$icon`.

`category_id`

The value of `category_id` as returned in the request parameters (either POST or GET).

`manufacturer_id`

The value of `manufacturer_id` as returned in the request parameters (either POST or GET).

`keyword1`

The value of `keyword1` as returned in the request parameters (either POST or GET).

`keyword2`

The value of `keyword2` as returned in the request parameters (either POST or GET). (`keyword2` is used only for advanced searches.)

`search_string`

The value of the parameter string to be used in the URL for successive page access. Built up from all the request parameters.

2.17 includes/browse_pagenav.tpl.php

Purpose and usage

Used to generate the HTML for the page navigation links. Supposed to contain the list of possible pages, first, previous, next, last page, and a drop-down for selecting the page. In the default theme, this template is used for the bottom page navigation links only.

Parent templates

browse_layouttable.tpl.php, browse_notables.tpl.php, browse_listtable.tpl.php.

Child templates

None.

Related page/class files

shop.browse.php.

Available fields

`$pagenav`

An instance of the `vmPageNav` object which can provide all functions of the `vmPageNav` class to output the HTML needed for the page links.

2.18 includes/ browse_searchparameter_form.tpl.php

Purpose and usage

Used to generate the HTML for the additional search parameter form for products that belong to product types. This form is available only when the request contains a `product_type_id` parameter.

Parent templates

`browse_layouttable.tpl.php`, `browse_notables.tpl.php`, `browse_listtable.tpl.php`.

Child templates

None.

Related page/class files

`shop.browse.php`, `shop.parameter_search_form.php`.

Available fields

`$ps_product_type`

An instance of the `ps_product_type` object which can provide all functions of the `ps_product_type` class to output the HTML needed for product type.

`$product_type_id`

The `product_type_id` of the product type.

3. Checkout template group

3.1 checkout_bar.tpl.php

Purpose and usage

Used to display the checkout bar throughout the checkout pages.

Parent templates

None.

Child templates

None.

Related page/class files

checkout.index.php, ps_checkout.php.

Available fields

`$step_count`

The total number of steps in the checkout process.

`$steps_to_do`

An array of the steps that needs to be done. Each step array contains the following values:

- `step_name`: An internal string identifying the step. See `$steps` below for possible values.
- `step_msg`: The step to be displayed in the bar. Used as a label for the step.
- `step_order`: The order of the step.

`$steps`

An array of all possible steps in the checkout. Each step is an array containing only one string element as an identifier for the step. Can be one of the following values:

- `CHECK_OUT_GET_SHIPPING_ADDR`
- `CHECK_OUT_GET_SHIPPING_METHOD`
- `CHECK_OUT_GET_PAYMENT_METHOD`
- `CHECK_OUT_GET_FINAL_CONFIRMATION`

`$highlighted_step`

The step to be highlighted. Can be any value from 1 to `$step_count`.

`$ship_to_info_id`

The identifier for the user address to ship to.

`$shipping_rate_id`

The identifier for the shipping rate selected by user. Can be empty.

3.2 customer_info.tpl.php

Purpose and usage

Used to display detailed information of the customer for checking.

Parent templates

`get_shipping_address.tpl.php`.

Child templates

None.

Related page/class files

`checkout.index.php`, `ps_checkout.php`.

Available fields

`$db`

An instance of `ps_DB` object which contains a database function as well as the data returned from the database. The database query of this object returns user info data from the database.

3.3 get_final_confirmation.tpl.php

Purpose and usage

Used to generate the final confirmation page during the checkout.

Parent templates

None.

Child templates

None.

Related page/class files

checkout.index.php, ps_checkout.php.

Available fields

`$zone_qty`

A variable for use in `zone_shipping` shipping method. Supposed to be the total number of products to ship. Its role in the template is not clear.

`$ship_to_info_id`

The identifier for the shipping address chosen by shopper.

`$shipping_rate_id`

The identifier for the shipping method chosen by shopper.

`$current_stage`

The stage number in the checkout.

`$payment_method_id`

The identifier of the payment method chosen by shopper.

`$weight_total`

The total weight of the orders.

`$basket_html`

The HTML as generated by the `ro_basket` templates.

3.4 get_payment_method.tpl.php

Purpose and usage

Used to display the payment method page for checkout.

Parent templates

None.

Child templates

`list_payment_method.tpl.php`.

Related page/class files

`checkout.index.php`, `ps_checkout.php`.

Available fields

`$zone_qty`

A variable for use in `zone_shipping` shipping method. Supposed to be the total number of products to ship. Its role in the template is not clear.

`$ship_to_info_id`

The identifier for the shipping address chosen by shopper.

`$shipping_rate_id`

The identifier for the shipping method chosen by shopper.

`$current_stage`

The stage number in the checkout.

`$payment_method_id`

The identifier of the payment method chosen by shopper.

`$weight_total`

The total weight of the orders.

`$basket_html`

The HTML as generated by the `ro_basket` templates.

3.5 `get_payment_method_paypal_ex.tpl.php`

Purpose and usage

Used to display the payment method page for checkout when PayPal extension is used.

All other properties

same as `get_payment_method.tpl.php`.

3.6 `get_shipping_address.tpl.php`

Purpose and usage

Used to display the get shipping address page.

Parent templates

None.

Child templates

`list_shipto_addresses.tpl.php`, `customer_info.tpl.php`.

Related page/class files

`checkout.index.php`, `ps_checkout.php`.

Available fields

`$zone_qty`

A variable for use in `zone_shipping` shipping method. Supposed to be the total number of products to ship. Its role in the template is not clear.

`$ship_to_info_id`

The identifier for the shipping address chosen by shopper.

`$shipping_rate_id`

The identifier for the shipping method chosen by shopper.

`$current_stage`

The stage number in the checkout.

`$payment_method_id`

The identifier of the payment method chosen by shopper.

`$weight_total`

The total weight of the orders.

`$basket_html`

The HTML as generated by the basket templates.

3.7 `get_shipping_method.tpl.php`

Purpose and usage

Used to display the get shipping method page.

Parent templates

None.

Child templates

`list_shipping_method.tpl.php`.

Related page/class files

`checkout.index.php`, `ps_checkout.php`.

Available fields

`$zone_qty`

A variable for use in `zone_shipping` shipping method. Supposed to be the total number of products to ship. Its role in the template is not clear.

`$ship_to_info_id`

The identifier for the shipping address chosen by shopper.

`$shipping_rate_id`

The identifier for the shipping method chosen by shopper.

`$current_stage`

The stage number in the checkout.

`$payment_method_id`

The identifier of the payment method chosen by shopper.

`$weight_total`

The total weight of the orders.

`$basket_html`

The HTML as generated by the `ro_basket` templates.

3.8 `list_payment_methods.tpl.php`

Purpose and usage

Used to list out all the payment methods available.

Parent templates

`get_payment_method.tpl.php` (indirectly through `ps_checkout` class).

Child templates

None.

Related page/class files

`checkout.index.php`, `ps_checkout.php`.

Available fields

`$db_nocc`

An instance of `ps_DB` object. Contains a list of payment methods that are not related to credit cards.

`$db_cc`

An instance of `ps_DB` object. Contains a list of payment methods that are related to credit cards.

`$nocc_payments`

Whether there are payment methods not related to credit cards. Can be true or false.

`$payment_method_id`

The identifier of the payment method selected by shopper. Can be empty.

`$first_payment_method_id`

The identifier for the first payment method. Normally the first non-credit-card payment method. If there is no non-credit-card payment method, this will be the first credit-card payment method.

`$count`

Total number of payment methods to be listed.

`$cc_payments`

Whether there are payment methods related to credit cards. Can be true or false.

`$ps_creditcard`

An instance of the `ps_creditcard` object used to provide all functions of the `ps_creditcard` class.

3.9 list_shipping_methods.tpl.php

Purpose and usage

Used to display a list of shipping methods available. This page is basically a container that does nothing but call all the shipping methods to list out their rates.

Parent templates

`get_shipping_method.tpl.php` (indirectly through `ps_checkout` class).

Child templates

None.

Related page/class files

checkout . index . php, ps_checkout . php.

Available fields

`$vars`

An array containing data for checking the shipping rate. The elements are:

- `ship_to_info_id`: The identifier for the shipping address of order.
- `shipping_rate_id`: `$shipping_method_id`.
- `weight`: The total weight of the order products.
- `zone_qty`: Variable containing the total number of items to be used by `zone_shipping` shipping method. Its use in template is not clear.
- `PSHOP_SHIPPING_MODULES`: An array of shipping modules enabled in the VirtueMart settings.

3.10 list_shipto_addresses.tpl.php

Purpose and usage

Used to display a list of shipping addresses.

Parent templates

`get_shipping_address . tpl . php` (indirectly through `ps_checkout` class).

Child templates

None.

Related page/class files

checkout . index . php, ps_checkout . php.

Available fields

`$db`

An instance of `ps_DB` object. Contains a list of shipping address user info returned from the database.

`$user_id`

The identifier for the logged in user.

`$name`

The name attribute of the input element for the ship to address. It should be `ship_to_info_id`.

`$value`

The value attribute of the input element for the ship to address. It should be the value of `$ship_to_info_id`.

`$bt_user_info_id`

The value of the billing address user info. If the shipping address and billing address are the same, this should be the same as `$value`.

3.11 login_registration.tpl.php

Purpose and usage

Used to display the login and registration page.

Parent templates

None.

Child templates

None.

Related page/class files

`checkout.index.php`, `ps_checkout.php`.

Available fields

`$zone_qty`

A variable for use in `zone_shipping` shipping method. Supposed to be the total number of products to ship. Its role in the template is not clear.

`$ship_to_info_id`

The identifier for the shipping address chosen by shopper. Normally should be empty in login and registration page.

`$shipping_rate_id`

The identifier for the shipping method chosen by shopper. Normally should be empty in login and registration page.

`$current_stage`

The stage number in the checkout.

`$payment_method_id`

The identifier of the payment method chosen by shopper. Normally should be empty in login and registration page.

`$weight_total`

The total weight of the orders.

4. Common template group

4.1 `availability.tpl.php`

Purpose and usage

Used to display the availability information on the Product List and Product Details page.

Parent templates

All core browse templates and all flypage templates.

Child templates

None.

Related page/class files

shop.browse.php, shop.product_details.php.

Available fields

`$product_id`

The identifier for the product.

`$product_available_date`

The date that product will be available, formatted in date format as specified.

`$product_availability`

The text or image to indicate product availability.

`$product_in_stock`

The total in stock.

4.2 buttons.tpl.php

Purpose and usage

Used to display the buttons for print, PDF, and e-mail.

Parent templates

Used by all core browse templates and flypage templates.

Child templates

None.

Related page/class files

shop.browse.php, shop.product_details.php, ps_product.php.

Available fields

`$option`

The value of the option parameter in request URL. Normally is `com_virtuemart` but may be other values if the template is invoked from a Joomla! module.

`$category_id`

The value for the `category_id`.

`$product_id`

The value for the `product_id`.

4.3 categoryChildlist.tpl.php

Purpose and usage

Used to generate the HTML for child categories.

Parent templates

shopIndex.tpl.php, all core browse template except PDF.

Child templates

None.

Related page files

shop.browse.php, shop.index.php.

Available fields

`$browsepage_lbl`

The heading for the page. Usually contains the name of the category.

`$desc`

The description for the category. This is the same as stored in the database and can contain HTML.

`$categories`

An array of child categories. Each child category is an array containing the following elements:

- `category_name`: Name of the category.
- `category_id`: Identifier for the category.
- `category_thumb_image`: Filename of the thumb image. No path info included.
- `number_of_products`: Number of products in the category.

4.4 couponField.tpl.php

Purpose and usage

Used to display the shop basket content in the shop cart page and checkout pages.

Parent templates

All basket templates.

Child templates

None.

Related page/class files

`shop.cart.php`, `basket.php`, `checkout.index.php`.

Available fields

This template does not need any specific fields, but all available fields in shop basket templates (such as `basket/basket_b2c.html.php`) are also available in this template.

4.5 featuredProducts.tpl.php

Purpose and usage

Used to display the list of featured products as an addendum element in the Product List and Product Details pages. Note that Joomla! featured product module does not use this template.

Parent templates

None.

Child templates

None.

Related page/class files

shop.browse.php, shop.product_details.php, shop.index.php, ps_product.php.

Available fields

`$featured_products`

An array of featured products. Each featured product contains the following elements:

- `product_sku`: The value as stored in `vm_product` database.
- `product_name`: The value as stored in `vm_product` database.
- `product_price`: The value as stored in `vm_product` database.
- `product_s_desc`: The value as stored in `vm_product` database.
- `product_url`: The value as stored in `vm_product` database.
- `product_thumb`: The value as stored in `vm_product` database. This is just the file name without the path information.
- `product_full_image`: The value as stored in `vm_product` database. This is just the file name without the path information.
- `product_id`: The value as stored in `vm_product` database.
- `flypage`: The flypage template used for the product.

- `form_addtocart`: The HTML form returned from invoking the template `browse/includes/addtocart_form.tpl.php`.

4.6 login_form.tpl.php

Purpose and usage

Used to display the login form. Usually used in first page of checkout.

Parent templates

None.

Child templates

None.

Related page files

`checkout.index.php`.

Available fields

`$action`

The URL to post the form data for authenticating the user. The URL for Joomla 1.5 normal authentication page is `index.php?option=com_user&task=login`.

`$return_url`

The URL to return to after login. Usually `checkout.index` page but can be set to an other page. The text is `base_64` encoded to make sure it can be retrieved by the `com_user` component.

`$validate`

The validation code to make sure the login is from proper website access.

4.7 minicart.tpl.php

Purpose and usage

Used to display an abridged version of the shop basket. Intended for Joomla! modules.

Parent templates

None.

Child templates

None.

Related page files

shop.basket_short.php.

Available fields

`$minicart`

An array of products in the shop basket. Each product consists of five elements:

- `url`: The URL of the flypage of the product.
- `product_name`: Value of `product_name` as stored in database.
- `attributes`: Description text listing all the attributes shopper selected for this product. An empty string if product does not have attribute.
- `price`: HTML for the display of the total price, currency, and quantity included.
- `quantity`: Quantity of product.

`$empty_cart`

Whether the cart is empty or not. Can be true or false.

`$delete_cart`

HTML for the `<a>` tag that links to `shop.cart_reset` page. This will remove all products from the shop basket.

`$vmMinicart`

Whether to use a summary version of the cart where only the number of products and total cost will be displayed. Can be true or false.

`$total_products`

The text for displaying the total number of products in the shop basket.

`$total_price`

The text for displaying the total cost of products in the shop basket.

`$show_cart`

HTML for the `<a>` tag that links to the `shop.cart` page.

`$saved_cart`

HTML for the `<a>` tag that links to the `shop.saved_cart` page. This will restore the cart that was saved previously.

4.8 `moreImages.tpl.php`

The `moreImages.tpl.php` is an empty file in the default theme. It is probably just a legacy file that is no longer used. The more images code currently resides in the `theme.php` file.

4.9 `pathway.tpl.php`

Purpose and usage

Used to generate the HTML for displaying the pathway.

Parent templates

All pages template group templates, all flypage templates, and all core browse templates.

Child templates

None.

Related page/class files

All frontend VirtueMart pages in the `account`, `checkout`, and `shop` modules.

Available fields

`$pathway`

An array of pathway items. Each pathway item is an array containing the following elements:

- `name`: Name of the category or subcategory.
- `link`: URL that links to the category or subcategory. Can be empty, if it is the last element that links to the current page.

`$product_name`

Name of the product for display.

`$next_product`

An array of product links pointing to the next product in the same category. Can be empty if there is no next product in the sequence. It contains the following elements:

- `product_id`: Product identifier of the next product.
- `product_name`: The name of the next product.

`$next_product_url`

The URL to access the Product Details page of the next product.

`$previous_product`

An array of product links pointing to the previous product in the same category. Can be empty, if there is no previous product in the sequence. It contains the following elements:

- `product_id`: Product identifier of the next product.
- `product_name`: The name of the next product.

`$previous_product_url`

The URL to access the Product Details page of the previous product.

`$return_link`

The URL to return to.

4.10 price.tpl.php

Purpose and usage

Used to generate the HTML for displaying the price.

Parent templates

None.

Child templates

None.

Related page/class files

ps_product.php, shop.browse.php, shop.product_details.php.

Available fields

`$product_id`

The product identifier.

`$product_name`

The product name as recorded in the database.

`$price_info`

An array containing the price info for the product. The elements are:

- `product_price`: The value of the price in decimal.
- `product_currency`: The value of the currency code.
- `product_base_price`: The value of the base price excluding attributes.
- `product_has_multiple_prices`: Whether product has multiple pricing structure that depends on quantity. 1 for yes and 0 for no.
- `product_price_id`: The `price_id` (an integer) identifying the price in the database.
- `item`: Whether there is a price. 1 for yes and 0 for no.

`$base_price_info`

An array containing the base price info for the product (disregarding the attribute). The elements are the same as `$price_info`.

`$discount_info`

An array containing the discount info for the product with the user group discount taken into account.

- `amount`: The value for the discount in decimal.
- `is_percent`: Whether the discount is a percentage or an absolute value.
- `create_time`: Time that the discount is created. This is an integer representing the Unix time stamp. `$text_including_tax`

The text that tells how much tax is included.

`$undiscounted_price`

The price before discount.

`$base_price`

The price before discount with attribute taken into account.

`$price_table`

The HTML for the price table for various quantities.

4.11 product_type.tpl.php

Purpose and usage

Used to generate the HTML for the product type details.

Parent templates

None.

Child templates

None.

Related page/class files

`ps_product_type.php`.

Available fields

`$product_types`

An array of product types. Each product type is an array containing the following elements:

- `product_type_count_params`: The number of parameters for this product type.
- `parameters`: An array of product type parameters. Each parameter contains the following elements:
 - `parameter_label`: Label for the parameter
 - `parameter_description`: Description of the parameter
 - `tooltip`: HTML for the tooltip containing the parameter description
 - `parameter_value`: Value of the parameter
 - `parameter_unit`: Unit of the parameter

4.12 productsnapshot.tpl.php

Purpose and usage

Used to display abridged details of a product. Usually for a Joomla! module or addendum information on a VirtueMart page.

Parent templates

None.

Child templates

Use `price.tpl.php` indirectly.

Related page/class files

`ps_product.php`.

Available fields

`$product_id`

Product identifier. This represents the same value as the `product_id` in the product data table.

`$product_name`

This represents the same value as the `product_name` in the product data table.

`$show_product_name`

Whether to show the product name.

`$product_link`

URL that links to the flypage of the product.

`$product_thumb_image`

Path for the thumbnail image of the product.

`$price`

This is the HTML for the price of the product. HTML will depend on the `price.tpl.php` template located in the common template group.

`$addtocart_link`

URL that will add the product to the shop basket. This link will invoke the add-to-cart function and then go to the shop cart page.

4.13 recent.tpl.php

Purpose and usage

Used to display a list of products that shopper has visited within the current session.

Parent templates

All flypage templates and core browse templates.

Child templates

None.

Related page/class files

shop.browse.php, shop.product_details.php.

Available fields

`$recent_products`

An array of products that have been visited. Each product is an array containing the following elements:

- `product_s_desc`: Short description of the product
- `product_url`: URL that links to the Product Details page of the product
- `category_url`: URL that links to the Product List page of the category
- `product_name`: Name of the product
- `category_name`: Name of category
- `product_thumb_image`: Filename of the product thumb image or URL that links to an external site

4.14 reviews.tpl.php

Purpose and usage

Used to display a list of reviews for the product.

Parent templates

All flypage templates.

Child templates

None.

Related page/class files

shop.product_details.php, ps_reviews.php.

Available fields

`$num_rows`

The number of reviews to be displayed.

`$reviews`

An array of review arrays. Each review array contains the following elements:

- `userid`: User ID of the user providing the review.
- `username`: The username of the user providing the review.
- `name`: The name of the user providing the review.
- `time`: The time of the review.
- `user_rating`: The rating of the product provided by the user.
- `comment`: The text of the comment by the user.

`$showall`

Whether to show all the reviews in one page. Otherwise, only the last five reviews are returned. 1 for yes and 0 for no. The value is read from the request URL parameter `showall`.

4.15 reviewform.tpl.php

Purpose and usage

Used to display the review form to collect reviews from shoppers. The review form will show only when the shopper is logged in.

Parent templates

All flypage templates.

Child templates

None.

Related page/class files

`shop.product_details.php`, `ps_reviews.php`.

Available fields

`$product_id`

The product identifier of the product to be reviewed.

`$alreadycommented`

Whether the shopper has commented before. 1 for yes and 0 for no.

`$review_comment`

The text for explaining the review, including the maximum and minimum length. This is the text from parsing the language element `PHPSHOP_REVIEW_COMMENT`.

4.16 relatedProducts.tpl.php

Purpose and usage

Used to display a list of related products on the Product Details page.

Parent templates

All flypage templates.

Child templates

None.

Related page/class files

`shop.product_details.php`.

Available fields

`$ps_product`

An instance of the `ps_product` class. Used to provide all sort of functions available in the `ps_product` class.

`$products`

An instance of the `ps_DB` class. It carries the database query property to retrieve the list of related products. The list of related products can be accessed through the functions `next_record()` and `f().next_record()`. `next_record()` will increment the object pointer to the next available related product and `f()` will return the field value by passing in a field list.

4.17 shopIndex.tpl.php

Purpose and usage

Used to display the home page of the store.

Parent templates

None.

Child templates

`categoryChildlist.tpl.php`, `recent.tpl.php`.

Related page/class files

`shop.index.php`.

Available fields

`$categories`

The HTML as returned by invoking the template `categoryList.tpl.php` for showing the top categories of the store.

`$ps_product`

An instance of the `ps_product` object. Used to provide all sorts of function of the `ps_product` class.

`$recent_products`

The HTML as returned by invoking the template `recent.tpl.php` for showing the recent product visited by the shopper in this session.

4.18 voteform.tpl.php

Purpose and usage

Used to display the vote form for the product.

Parent templates

All flypage templates.

Child templates

None.

Related page/class files

shop.product_details.php, ps_reviews.php.

Available fields

`$product_id`

The product identifier of the product to vote for.

4.19 votes_allvotes.tpl.php

Purpose and usage

Used to display the ratings for the product.

Parent templates

All flypage templates and core browse templates.

Child templates

None.

Related page/class files

shop.browse.php, shop.product_details.php.

Available fields

`$allvotes`

The total number of votes for the product. It is the same value as the field `allvotes` stored in the `vm_product_reviews` database.

`$rating`

The average rating for the product. It is the same value as the field `rating` stored in the `vm_product_reviews` database.

`$product_id`

The product identifier for the product of the vote and rating.

5. order_emails template group

5.1 confirmation_email.tpl.php

Purpose and usage

Used to prepare the HTML e-mail to be sent to the shopper on confirmation of the order.

Parent templates

None.

Child templates

None.

Related page/class files

`ps_checkout.php`.

Available fields

`$is_email_to_shopper`

Whether this e-mail is sent to the shopper or administrator. 1 for yes and 0 for no.

`$db`

An instance of `ps_DB` object which contains database function as well as the data returned from the database. The database query of this object returns data for all order info related to this order.

`$dboi`

An instance of `ps_DB` object which contains database function as well as the data returned from the database. The database query of this object returns data for all order items related to this order.

`$dbbt`

An instance of `ps_DB` object which contains database function as well as the data returned from the database. The database query of this object returns data for the bill-to user info related to this order.

`$dbst`

An instance of `ps_DB` object which contains database function as well as the data returned from the database. The database query of this object returns data for the ship-to user info related to this order.

`$ps_product`

An instance of `ps_product` object that provides all functions of the `ps_product` class.

`$shippingfields`

See `$fields` in `account.billing.tpl.php` template in the page template group.

`$registrationfields`

See `$fields` in `account.billing.tpl.php` template in the page template group.

`$order_id`

The identifier of the order that links to the `vm_order` datatable.

`$order_discount`

The HTML for the payment discount amount.

`$order_discount_lbl`

The label of the payment discount line.

`$order_discount_plusminus`

The sign of the payment discount amount. Plus means a discount. Minus means additional fee.

`$coupon_discount`

The HTML for the coupon discount amount.

`$coupon_discount_lbl`

The label for the coupon discount line.

`$coupon_discount_plusminus`

The sign of the coupon discount amount. Plus means a discount. Minus means additional fee.

`$order_date`

The formatted date of the order.

`$order_status`

The status of the order.

`$legal_info_title`

The title for the condition of sales section.

`$legal_info_html`

The HTML for the condition of sales section.

`$order_link`

The link to see the order on the website.

`$payment_info_lbl`

Label for the payment information section.

`$payment_info_details`

HTML for the payment information.

`$shipping_info_lbl`

\$label for the shipping information section.

`$shipping_info_details`

HTML for the shipping information section.

`$from_email`

The e-mail address to put in the from address of the e-mail.

`$customer_note`

The customer note the shopper typed in on the final confirmation page.

`$order_header_msg`

A message to be placed on the header of the order.

`$order_subtotal`

The HTML for the order subtotal.

`$order_shipping`

The HTML for the shipping cost.

`$order_tax`

The HTML for the tax of the order.

`$order_total`

The HTML for the order total.

5.2 enquiry_email.tpl.php

Purpose and usage

Used to prepare the HTML e-mail to be sent to the shopper and vendor when he/she posts a question to the vendor.

Parent templates

None.

Child templates

None

Related page/class files

ps_communication.php, shop_ask.php.

Available fields

`$vendorname`

Name of vendor.

`$subject`

Subject of the e-mail.

`$contact_name`

Name of shopper.

`$contact_email`

E-mail of shopper.

`$product_name`

Name of the product.

`$product_s_description`

Short description of the product.

`$product_url`

The link to the flypage of the product.

`$product_sku`

The sku of the product.

`$product_thumb`

Image reference of the thumbnail image. This works only in the vendor e-mail as the image is not embedded in the shopper e-mail.

6. Pages template group

6.1 account.billing.tpl.php

Purpose and usage

Used to display the billing address edit page.

Parent templates

None.

Child templates

None.

Related page/class files

`account.billing.php`.

Available fields

`$pathway`

An array of pathway items. Each pathway item is an array containing the following elements:

- `name`: Name of the category or subcategory.
- `link`: URL that links to the category or subcategory. Can be empty, if it is the last element which links to the current page.

`$vmPathway`

The HTML of the pathway generated by invoking the template `common/pathway.tpl.php`.

`$fields`

An array of user fields designated for registration/billing address. Each user field is an object containing the following properties (For details of the user fields, you can check in the VirtueMart backend, click **Admin/Manage User Fields** on the left menu):

- `fieldid`: Identifier of the field.
- `name`: Name of the field, for example, e-mail.
- `title`: The label of the field, it can be a language element or a literal string.
- `description`: Description of the field.
- `type`: Field type, please check the VirtueMart backend for possible field types.
- `maxlength`: Maximum length of the field.
- `size`: Size of the field for display.
- `required`: Whether the field is a required field. 1 for yes and 0 for no.
- `ordering`: The order of the display sequence.
- `cols`: The columns attribute for textarea.
- `rows`: The rows attribute for textarea.
- `value`: Possible values of the field.
- `default`: Default value of the field.
- `published`: Whether the field is published (enabled). 1 for yes and 0 for no.
- `registration`: Whether the field is used in the account management form. 1 for yes and 0 for no.
- `shipping`: Whether the field is used in the account management form. 1 for yes and 0 for no.
- `account`: Whether the field is used in the account management form. 1 for yes and 0 for no.
- `readonly`: Whether the field is read-only. 1 for yes and 0 for no.
- `calculated`: Whether the field is a calculated field. 1 for yes and 0 for no. The property does not seem to be actually used.
- `sys`: Whether the field is used for the system. 1 for yes and 0 for no. A system field name cannot be changed.

- `vendor_id`: The identifier for the vendor. This property does not seem to be in use yet.
- `params`: Additional parameters. This property does not seem to be in use yet.

`$db`

An instance of `ps_DB` object that contains database function as well as the data returned from the database. The database query of this object returns the billing user info of the logged in user.

`$next_page`

The URL of the page to go to when the user is finished with the page (either save or cancel). This is needed because the page may be referred by the `account.index` page or `checkout.index` page (get shipping address).

`$missing`

The list of required fields that do not have a value entered. This field is used only when the form needs to be redisplayed because of validation errors.

`$skip_fields`

An array of field names that will be skipped in the display. For silent registration, for example, we don't need the username and password fields.

6.2 `account.index.tpl.php`

Purpose and usage

Used to display the **Account Management** home page.

Parent templates

None.

Child templates

None.

Related page/class files

`account.index.php`.

Available fields

`$ps_order`

An instance of the `ps_order` object that allows the template to use all functions of the `ps_order` class.

6.3 account.order_details.tpl.php

Purpose and usage

Used to display the invoice details in the frontend restricted area (**Account Management**).

Parent templates

None.

Child templates

`common/pathway.tpl.php`.

Related page/class files

`account.order_details.php`.

Available fields

`$print`

Whether to show the print button on the page. 1 for yes and 0 for no. Actually, the print button is always there. The only difference is whether the template will need to handle the print button.

`$order_id`

The identifier for locating the order from the database.

`$pathway`, `$vmPathway`

See `account.billing.tpl.php` template.

`$db`

An instance of `ps_DB` object which contains database function as well as the data returned from the database. The database query of this object returns major data of the order.

`$dbbt`

An instance of `ps_DB` object which contains database function as well as the data returned from the database. The database query of this object returns data of the all user info related to this order. There may be two user info. The first one is the bill-to user. The second one (which may not exist if the bill-to user is same as ship-to user) is the ship-to user.

`dbpm`

An instance of `ps_DB` object which contains database function as well as the data returned from the database. The database query of this object returns data of the order payment.

`$user`

The first record of the `$dbbt` object which represents the bill-to user info. This is not actually used in the default theme template.

`$registrationfields`

See `$fields` in `account.billing.tpl.php` template.

`$shippingfields`

See `$fields` in `account.billing.tpl.php` template.

`$time_offset`

The time offset for the site as recorded in the Joomla! configuration. Used to make sure the invoice date is correctly calculated in the local time of the store.

6.4 account.orders.tpl.php

Purpose and usage

Used to display a list of orders for the logged in user. Can be a page on its own or part of the **Account Management** home page (`account.index` page).

Parent templates

`account.index.tpl.php`.

Child templates

None.

Related page/class files

`account.index.php`, `account.orders.php`.

Available fields

`$ps_order`

An instance of `ps_order` object used to provide all functions of the `ps_order` class.

6.5 `account.shipping.tpl.php`

Purpose and usage

Used to display a list of orders for the logged in user.

Parent templates

None.

Child templates

`common/pathway.tpl.php`.

Related page/class files

`account.shipping.php`.

Available fields

`$pathway`, `$vmPathway`

See `account.billing.tpl.php` template.

`$db`

An instance of `ps_DB` object which contains database function as well as the data returned from the database. The database query of this object returns all shipping user info for the logged in user.

6.6 `account.shipto.tpl.php`

Purpose and usage

Used to display the Shipping Information edit form for creating new shipping address or update existing shipping address.

Parent templates

None.

Child templates

`common/pathway.tpl.php`.

Related page/class files

`account.shipto.php`.

Available fields

`$pathway`, `$vmPathway`

See `account.billing.tpl.php` template.

`$next_page`

The URL of the page to go to when the user is done with the page (either save or cancel). This is needed because the page may be referred by the `account.shipping` page or `checkout.index` page (getting shipping address).

`fields`

See `$fields` in `account.billing.tpl.php` template.

`$missing`

The list of required fields which do not have value entered. This field is used only when the form needs to be re-displayed because of validation errors.

`$db`

An instance of `ps_DB` object which contains database function as well as the data returned from the database. The database query of this object returns all shipping user info for the logged in user.

`$user_info_id`

The identifier for the user info to be edited. It is empty for a new shipping address.

6.7 checkout.thankyou.tpl.php

Purpose and usage

Used to display thank you page after the order is confirmed and recorded in the database.

Parent templates

None.

Child templates

None.

Related page/class files

`checkout.thankyou.php`.

Available fields

`$order_id`

The identifier for the confirmed order.

`$ps_product`

An instance of the `ps_product` object used to provide all functions of the `ps_product` class.

`$user`

An instance of `JUser` object used to access data of the user.

`$dbbt`

An instance of `ps_DB` object which contains database function as well as the data returned from the database. The database query of this object returns the billing user info for the order.

`$db`

An instance of `ps_DB` object which contains database function as well as the data returned from the database. The database query of this object returns data of the confirmed order.

6.8 shop.ask.tpl.php

Purpose and usage

Used to display the Ask Seller page and also after the e-mail has been sent to the vendor.

Parent templates

None.

Child templates

None.

Related page/class files

`shop.ask.php`, `ps_communication.php`.

Available fields

`$product_id`

The identifier for the product to ask questions about.

`$product_sku`

The product sku of the product.

`$category_id`

The category ID of the product.

`$product_link`

The URL of the Product Details page for the product.

`$set`

The value of `set` parameter as passed in through the request. Indicates where the e-mail has been sent. 1 for yes and 0 for no.

`$name`

Name of the shopper if he/she is logged in.

`$email`

E-mail address of the shopper if he/she is logged in.

`$flypage`

The value of the flypage as passed back in the request parameter.

`$subject`

The subject of the e-mail to send.

`$db_product`

An instance of `ps_DB` object which contains database function as well as the data returned from the database. The database query of this object returns data for the product in question.

6.9 shop.cart.tpl.php

Purpose and usage

Used to display the Shop Cart page.

Parent templates

None.

Child templates

The basket templates, `common/couponField.tpl.php`.

Related page/class files

`shop.cart.php`, `basket.php`.

Available fields

`$show_basket`

Whether to show the shop basket. 1 for yes and 0 for no.

`$continue_link`

The URL to link to when the shopper clicks the **Continue Shopping** link.

`$category_id`

The `category_id` of the product added.

`$product_id`

The `product_id` of the product added.

`$manufacturer_id`

The `manufacturer_id` of the product added.

6.10 shop.infopage.tpl.php

Purpose and usage

Used to display vendor information.

Parent templates

None.

Child templates

None.

Related page/class files

shop.infopage.php.

Available fields

`$v_name`

Vendor name.

`$v_address_1`

Vendor address line 1.

`$v_address_2`

Vendor address line 2.

`$v_zip`

Vendor zip.

`$v_city`

Vendor city.

`$v_title`

Vendor title.

`$v_first_name`

First name of vendor contact.

`$v_last_name`

Last name of vendor contact.

`$v_fax`

Vendor fax.

`$v_email`

Vendor e-mail address.

`$v_logo`

Filename of the vendor logo.

`$v_category`

Vendor category.

`$db`

An instance of `ps_DB` object which contains database function as well as the data returned from the database. The database query of this object returns data for the vendor.

6.11 shop.manufacturer.tpl.php

Purpose and usage

Used to display the Manufacturer page.

Parent templates

None.

Child templates

None.

Related page/class files

`shop.manufacturer_page.php`.

Available fields

`$mf_name`

Name of the manufacturer.

`$mf_email`

E-mail address of the manufacturer.

`$mf_desc`

Description of the manufacturer.

`$mf_url`

The URL of the manufacturer.

`$db`

An instance of `ps_DB` object which contains database function as well as the data returned from the database. The database query of this object returns data of the manufacturer.

6.12 shop.savedcart.tpl.php

Purpose and usage

Used to display the Saved Cart page.

Parent templates

None.

Child templates

None.

Related page/class files

`shop.savedcart.php`.

Available fields

`$replaceSaved`

The form used to replace the saved cart with the current cart.

`$mergeSaved`

The HTML form used to merge the current cart with the saved cart.

`$deleteSaved`

The HTML form used to delete the saved cart.

`$show_basket`

Whether to show the shop basket. 1 for yes and 0 for no.

`$continue_link`

The link to go back to the previous page.

`$category_id`

The category ID of the product recently added.

`$product_id`

The product ID of the product recently added.

`$manufacturer_id`

The manufacturer ID of the product recently added.

6.13 shop.waiting_list.tpl.php

Purpose and usage

Used to display the Out of Stock page to allow the shopper to sign up for notification when the product is back in stock.

Parent templates

None.

Child templates

None.

Related page/class files

shop.waiting_list.php, zw_waiting_list.php.

Available fields

`$product_id`

The product ID of the product to be put on the waiting list.

6.14 shop.waiting_thanks.tpl.php

Purpose and usage

Used to display the thank you page after the shopper is placed on the waiting list for an out of stock product.

Parent templates

None.

Child templates

None.

Related page/class files

shop.waiting_thanks.php, zw_waiting_list.php.

Available fields

`$product_id`

The product ID of the product the shopper is waiting for.

`$ok`

The return value of the add-to-waiting-list (`waitinglistadd`) function. True if successful, otherwise False.

7.product_details template group

The `product_details` template group contains templates related to the display of Product Details page. The templates in the root directory, also known as flypage templates. They control the display of everything but will invoke a lot of child templates for displaying the details. The templates in the `includes` subdirectory are supplementary templates for displaying individual elements in the Product Details page.

7.1 flypage.tpl.php

Purpose and usage

One of the flypage templates that control the display of the Product Details page.

Parent templates

None.

Child templates

The flypage template contains many subtemplates.

Related page/class files

`shop.product_details.php`.

Available fields

`$pathway`

See `account.billing.tpl.php` template in the page template group.

`$product_name`

The name of the current product.

`$next_product`

An array of the product containing the following two elements:

- `product_id`: Product ID of the next product
- `product_name`: Name of the next product

`$next_product_url`

The URL that points to the Product Details page of the next product.

`$previous_product`

An array of the product containing the following two elements:

- `product_id`: Product ID of the previous product

- `product_name`: Name of the previous product

`$previous_product_url`

The URL that points to the Product Details page of the previous product.

`$return_link`

The link to the parent product. This has a value only for child products and is an empty string for other products.

`category_id`

Category ID of the product.

`$product_id`

Product ID of the product.

`$buttons_header`

The HTML for the header buttons as returned by invoking the `common/button.tpl.php` template.

`$productArray`

This is an array of all the properties of the product stored in the database `vm_product` table. The standard elements are `product_id`, `vendor_id`, `product_parent_id`, `product_sku`, `product_s_desc`, `product_desc`, `product_thumb_image`, `product_full_image`, `product_publish`, `product_weight`, `product_weight_uom`, `product_length`, `product_width`, `product_height`, `product_lwh_uom`, `product_url`, `product_in_stock`, `product_available_date`, `product_availability`, `product_special`, `product_discount_id`, `ship_code_id`, `cdate`, `mdate`, `product_name`, `product_sales`, `attribute`, `custom_attribute`, `product_tax_id`, `product_unit`, `product_packaging`, `child_options`, `quantity_options`, `child_option_ids`, and `product_order_levels`. These fields are available as separate variables. The array is passed to the template for completeness. If you have customized the `vm_product` datatable, new fields will also be available in this variable.

`$vendor_id`

The vendor of the product.

`$product_parent_id`

The parent product of the product.

`$product_sku`

sku of the product.

`$product_s_desc`

Short description of the product.

`$product_desc`

Full description of the product.

`$product_thumb_image`

Thumb image filename or URL.

`$product_full_image`

Full image filename or URL.

`$product_publish`

Whether the product is published. 1 for yes and 0 for no.

`$product_weight`

Weight of the product.

`$product_weight_uom`

Unit of measure for the weight.

`$product_length`

Length of the product.

`$product_width`

Width of the product.

`$product_height`

Height of the product.

`$product_lwh_uom`

Unit of measure for the length, width, and height.

`$product_url`

External URL of the product, if any.

`$product_in_stock`

The total in stock.

`$product_available_date`

Date that the product is available. This is an integer representing the Unix time stamp.

`$product_availability`

The text or image location for specifying the product availability

`$product_special`

Whether the product is a featured product. 1 for yes and 0 for no.

`$product_discount_id`

The discount ID. Link to `vm_product_discount` datatable.

`$ship_code_id`

Its use in VirtueMart is not clear.

`$cdate`

Date the product was created.

`$mdate`

Date the product was last modified.

`$product_sales`

Total sales of the product.

`$attribute`

Advanced attribute of the product.

`$custom_attribute`

Custom attribute of the product.

`$product_tax_id`

The tax ID, linked to the `vm_tax_rate` datatable.

`$product_unit`

Unit for the product.

`$product_packaging`

How the product is packaged.

`$child_options`

A string specifying the display options for child products. All the options are separated by commas. The options are:

- `display_use_parent`
- `product_list`
- `display_headers`
- `product_list_child`
- `product_list_type`
- `display_desc`
- `desc_width`
- `attrib_width`
- `child_class_sfx`

You can refer to the product edit form, the **Display Options** tab, to see the exact meaning of each.

`$quantity_options`

A string specifying the display options for the quantity box. All the options are separated by commas. The options are:

- `quantity_box`
- `quantity_start`
- `quantity_end`
- `quantity_step`

You can refer to the product edit form, the **Display Options** tab, to see the exact meaning of each.

`$child_option_ids`

The use of this field in VirtueMart is not clear.

`$product_order_levels`

The stock level when product should be replenished.

`$product_image`

The HTML for showing the product image.

`$more_images`

The HTML for displaying more images by calling the theme function `vmMoreImagesLink()`.

`$files, $images`

See `browse_1.tpl.php` template in browse template group.

`$file_list`

The HTML for the list of files associated with the product in a series of links.

`$edit_link`

The link to edit the product detail. Available only for administrator.

`$manufacturer_link`

The link to the manufacturer.

`$product_price`

The HTML for showing the price of product as returned by invoking the template `common/price.tpl.php`.

`$product_price_lbl`

The label for the price language element.

`$product_price_raw`

The array of price attributes in its raw form. It contains the following elements:

- `product_price`: Selling price after attribute calculation
- `product_currency`: Currency of price
- `product_base_price`: Base price without attribute
- `product_has_multiple_prices`: Whether the product has quantity price
- `product_price_id`: Price ID that links to the `vm_product_price` datatable.

`$product_description`

The product description after applying Joomla! content plugins if enabled in the config.

`$manufacturer_id`

The manufacturer ID of the product.

`$flypage`

The flypage template to be used. This is the value specified in the request URL.

`$ps_product_attribute`

An instance of the `ps_product_attribute` object used to provide functions of the `ps_product_attribute` class.

`$addtocart`

The HTML for the add-to-cart form as returned by invoking the `addtocart_form.tpl.php` template.

`$navigation_pathway`

The HTML for the pathway as returned by invoking the `common/pathway.tpl.php` template.

`$navigation_childlist`

The HTML for the category child list as returned by invoking the `common/categoryChildlist.tpl.php` template.

`$product_reviews`

The HTML for the reviews as returned by invoking the `common/reviews.tpl.php` template.

`$product_reviewform`

The HTML for the review form as returned by invoking the `common/review_form.tpl.php` template. This will show only if the shopper is logged in. Otherwise, this will be an empty string.

`$product_availability_data`

An array of availability data. It contains the following elements:

- `product_id`: Product identifier
- `product_available_date`: Product available date in Unix time stamp format (integer)
- `product_availability`: The text for product availability or the file location of the image
- `product_in_stock`: The number of stock as recorded in database

`$related_products`

The HTML showing related products as returned by invoking the `common/relatedProducts.tpl.php` template.

`$vendor_link`

The link to the vendor page (`page=shop.infopage`).

`$product_type`

The HTML for the product type data as returned by calling the function `ps_product_type::list_product_type()`. This function adds all the HTML returned from the specific product type template together, (each product type can have its own template) and then combines the resulting HTML with that returned by invoking the `common/product_type.tpl.php` template.

`$ask_seller_href`

The URL that links to the ask seller page.

`$ask_seller_text`

The text showing the **Ask a Question** link.

`$ask_seller`

The HTML for the **Ask a Question** link.

`$recent_products`

The HTML for the recent products visited by the shopper as returned by invoking the `common/recent.tpl.php` template.

7.2 flypage-ask.tpl.php

Same as `flypage.tpl.php`.

7.3 flypage_images.tpl.php

Same as `flypage.tpl.php`.

7.4 flypage_lite_pdf.tpl.php

Same as `flypage.tpl.php`.

7.5 flypage_new.tpl.php

Same as `flypage.tpl.php`.

7.6 garden_flypage.tpl.php

Same as `flypage.tpl.php`.

7.7 includes/addtocart_advanced_attribute.tpl.php

Purpose and usage

Used to generate the HTML for displaying the advanced attribute.

Parent templates

`addtocart_drop.tpl.php`, `addtocart_list_multi.tpl.php`, `addtocart_list_single.tpl.php`.

Child templates

None.

Related page/class files

shop.product_details.php.

Available fields

`$attributes`

An array of attribute arrays. Each attribute array contains the following elements:

- `product_id`: The product identifier.
- `title`: The label for the field to be shown to the shopper.
- `titlevar`: The value of the field name to be passed back to server.
- `options_list`: An array of possible values for the attribute. Each option is an array with two elements:
 - `base_var`: The value of the option to be passed back to the server
 - `base_value`: The value of the display of the option

7.8 includes/addtocart_custom_attribute.tpl.php

Purpose and usage

Used to generate the HTML for displaying the custom attribute.

Parent templates

addtocart_drop.tpl.php, addtocart_list_multi.tpl.php, addtocart_list_single.tpl.php.

Child templates

None.

Related page/class files

shop.product_details.php.

Available fields

`$attributes`

An array of attribute arrays. Each attribute array contains the following elements:

- `product_id`: The product identifier
- `title`: The label for the field to be shown to the shopper
- `titlevar`: The value of the field name to be passed back to server

7.9 includes/addtocart_drop.tpl.php

Purpose and usage

Used to display child products as a drop-down.

Parent templates

`addtocart_form.tpl.php`.

Child templates

`addtocart_advanced_attribute.tpl.php`, `addtocart_custom_attribute.tpl.php`.

Related page/class files

`shop.product_details.php`.

Available fields

`$cls_suffix`

The CSS class suffix to use to allow individual customization per product.

`$product_id`

Product ID of the product.

`$advanced_attribute`

The HTML of the custom attribute as returned by invoking the `addtocart_advanced_attribute.tpl.php` template.

`$custom_attribute`

The HTML of the custom attribute as returned by invoking the `addtocart_custom_attribute.tpl.php` template.

`$drop_down`

The HTML for the drop-down box of child products.

7.10 includes/addtocart_form.tpl.php

Purpose and usage

Used to generate the HTML for the add-to-cart form.

Parent templates

All flypage templates.

Child templates

`addtocart_drop.tpl.php`, `addtocart_list_multi.tpl.php`, `addtocart_list_single.tpl.php`, `quantity_box_general.tpl.php`, `quantity_box_radio.tpl.php`.

Related page/class files

`shop.product_details.php`.

Available fields

Most of the fields available in flypage templates are also available in the `addtocart_form.tpl.php` template.

7.11 includes/addtocart_list_multi.tpl.php

Purpose and usage

Used to display child products as a table with one add-to-cart button for each product.

Parent templates

All flypage templates.

Child templates

None.

Related page/class files

`shop.product_details.php`.

Available fields

`$cls_suffix`

The CSS class suffix for the display to allow individual product customization.

`$product_id`

The product ID of the product.

`$display_header`

Whether to display table heading. 1 for yes and 0 for no. The column heading in `$headings` will be used.

`$display_product_type`

Whether to display product type detail. 1 for yes and 0 for no.

`$product_price`

Price of the product.

`$headings`

An array of the column headings.

`$desc_width`

Width for the description column.

`$attrib_width`

Width for all other columns.

`$productArray`

An array of product properties with the following elements for the parent product: `product_id`, `product_name`, `product_parent_id`, `product_sku`, `product_in_stock`, `product_full_image`, and `product_thumb_image`. The elements are also available as independent variables.

`$product_name`

Name of product.

`$product_parent_id`

Product ID of parent product.

`$product_sku`

sku of the product.

`$product_in_stock`

The total in stock.

`$product_full_image`

The full image filename or external URL.

`$product_thumb_image`

The thumb image filename or external URL.

`$child_link`

Whether to show the link to the child product in the list. True for yes and False for no.

`$products`

An array of child products. Each child product is an array with the following elements:

- `bgcolor`: CSS class for the row.
- `product_id`: ID of the child product.
- `category_id`: Category ID of the child product. (Actually, child products do not have a category; it must belong to the same category as its parent. We are not sure why this field is included.)
- `Itemid`: The order of the child product in the list.

- `parent_id`: Product ID of parent is repeated here.
- `flypage`: The flypage template for the product.
- `product_image`: HTML for displaying the product thumb image including view full size image link.
- `product_title`: The name of the child product.
- `attrib_value`: An array of the attributes of the child product.
- `quantity_box`: Type of quantity box.
- `price`: List price of parent product.
- `actual_price`: Selling price of child product.
- `product_type`: HTML of product type as returned by invoking the `common/product_type.tpl.php` template.
- `product_in_stock`: Product in stock.
- `advanced_attribute`: HTML of advanced attribute as returned by invoking the `addtocart_advanced_attribute.tpl.php` template.
- `custom_attribute`: HTML of custom attribute as returned by invoking the `addtocart_custom_attribute.tpl.php` template.

`$child_count`

The number of child products.

7.12 `includes/addtocart_list_single.tpl.php`

Purpose and usage

Used to display child products as a table with one add-to-cart button for all products.

All other properties

Same as `addtocart_list_multi.tpl.php` template.

7.13 `includes/quantity_box_general.tpl.php`

Purpose and usage

Used to generate the HTML for displaying the quantity box.

Parent templates

`addtocart_form.tpl.php`.

Child templates

None.

Related page/class files

`shop.product_details.php`, `shop.browse.php`, `ps_product_attribute.php`.

Available fields

`$prod_id`

Product ID of the product.

`$quantity`

The value for the quantity (selected previously by the shopper or preset in the code).

`$display_type`

The type of quantity box to use.

`$child`

Whether the product is a child product. 1 for yes and 0 for no.

`$quantity_options`

An array specifying the properties of the quantity options. The elements are:

- `quantity_start`: Minimum acceptable value for the quantity. 0 means no restriction.
- `quantity_end`: Maximum acceptable value for the quantity. 0 means no restriction.
- `quantity_step`: Each step for changing quantity.
- `quantity_box`: The type of quantity box. Can be:
 - None: Default if type is not specified.
 - Hide: Do not show quantity box. Set quantity to 1.

- Drop: Use a drop-down box for the shopper to select. Useful when the quantity step > 1.
- Check: Use checkbox to allow the shopper to select the product. The quantity will automatically set to 0 to 1.
- Radio: Use radio buttons to list out all possible options.

7.14 includes/quantity_box_radio.tpl.php

Purpose and usage

Used to generate the HTML for displaying quantity box for products with child products showing as a radio button group.

All other properties

Same as `quantity_box_general.tpl.php`.

