
commerce

Release 1.0.0

Sebastian Fischer

2015-07-12 11:01

CONTENTS

1	Introduction	3
1.1	What does this extension do?	3
1.2	Current situation	3
1.3	Sponsors	3
2	Installation	5
3	User Guide	7
3.1	TS-Template	7
3.2	Pages and Plugins	7
4	Webmodule Commerce	13
4.1	Webmodul Commerce	13
5	Configuration	23
5.1	Configuration	23
5.2	Realurl fixed post vars	37
5.3	Realurl post var sets	38
5.4	Tutorial	40
6	Development	43
6.1	Payment API	43
6.2	Hooks	43
7	Commerce Coding Guidelines	65
7.1	CGL	65
7.2	Deprecated log	65
7.3	Versions	65
7.4	Maintainance of the code	65
8	Deprecation	67
8.1	Deprecated sind 4.x	67
8.2	Removed methods since 4.x	67
8.3	Removed hooks since 4.x	73
9	Breaking Changes	83
9.1	Remove of iframe rendering the category tree	83
10	Support	85
11	Known problems	87
12	Changelog	89

Extension Key commerce

Language en

Keywords forEditors, forAdmins, forBeginners, forIntermediates

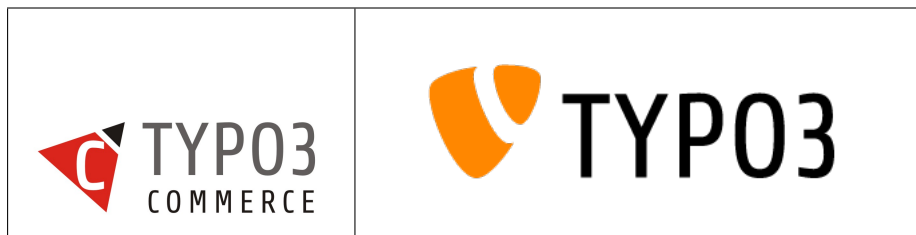
Copyright 2015

Author Sebastian Fischer

Email typo3@evoweb.de

License Open Content [License](#)

Rendered 2015-07-12 11:01



The content of this document is related to TYPO3,

a GNU/GPL CMS/Framework available from www.typo3.org

INTRODUCTION

1.1 What does this extension do?

This extension sets up a powerful shop being able to meet the current requirements for modern e-commerce-software.

1.2 Current situation

This extension is still in Beta-Mode, so slight changes are possible. Please report bugs at <http://bugs.typo3.org>. The mailing list is available at <http://lists.typo3.org/cgi-bin/mailman/listinfo/typo3-project-commerce>.

1.3 Sponsors

The development of this extension was supported by the following companies: <http://www.marketing-factory.de>, e-netconsulting KG, [n@work](#) Internet Informationssysteme GmbH. Parts of this documentation are provided by the Open Source Press Verlag GmbH. The text is from the book Der TYPO3 Webshop by Andrea Herzog-Kienast and Franz Holzinger. Thanks to the sponsors.

INSTALLATION

A quick introduction can be found in the doc-directory **commerce_quickinstall.txt**.

Install the extension with the key commerce and all other required extensions in the Extension Manager. Before starting the configuration of Commerce, some pages of the shop need to be set up and some other preparations are required too.

PHP version 5.3.x and TYPO3 version 6.2 are needed here.

The following extensions need to be installed and running:

- Commerce Extension Key: commerce
- Address list, Extension Key: tt_address
- Dynamic Flexforms, Extension Key: dynaflex
- Money code Library, Extension Key: moneylib
- Static Info Tables , Extension Key: static_info_tables
- Optional: Frontend User Registration, Extension Key: sr_feuser_register

Additionally, a Login Extension is required/ recommended.

You may install **felogin** (system extension with 4.2) and define a redirect after logout. This is important, because otherwise a user would get an error message if he wants to log out from a product page, as the page or the content will no longer be available after logout. This would cause an error message or display of an empty page. If you define a redirect after Logout, this problem will not arise.


Basically, there are two modes for the installation of Commerce. The Simple Mode implies an easier installation of the shop, whereas the standard mode provides all functions.


Simple Mode is a light installation of the shop. In **Simple Mode** there shall only be one article and price per product.

In order to do so, you have to activate the Simple Mode in the Extension Manager first. Check the corresponding box.

Information

Extension Manager

Extension:  Commerce (commerce)

ACTIVE STATUS:
The extension is installed (loaded and running)!
Click here to remove the extension: 

CONFIGURATION:
(Notice: You may need to clear the cache after configuration of the extension. This is required if the extension adds TypoScript depending on these settings.)

Enable features

Invoice BACKEND LOGIN required [invoiceBackendOnly]
If set, then the Invoice plugin will check for a backend login. If false, everyone can create an invoice with the correct link. So WATCH OUT FOR THIS ONE! It should probably be enabled!
☒

Simple Mode for products [simpleMode]
Alpha, Alpha !Use this if you only have one article per product
☒

Automatically generate prices while... [genprices]
Automatically generate prices while when changing one price according to the tax (0 - Do nothing, 1 recalculate not existing prices, 2 calculate from net, 3 calculate from gross)
 (Integer)

Page-ID for Invoice [invoicePageID]
Defines which Page displays the Invoice.
 (Integer)

Invoice Page Type [invoicePageType]
Defines which pagetype the Invoice should generate. Maybe a PDF-Generator exists and you want to use it, you have to configure the pageType of the PDF-output.
 (Integer)

Figure 2.1: Illustration 1: Activation of simple mode in extensionmanager

USER GUIDE

3.1 TS-Template

Create a new TS-Template for Commerce in your Template-SysFolder. Name it **+commerce**, assign the static template COMMERCE via **Include static (from extension)** and save the record. Include this template as usual in the TS-Template of your root page.

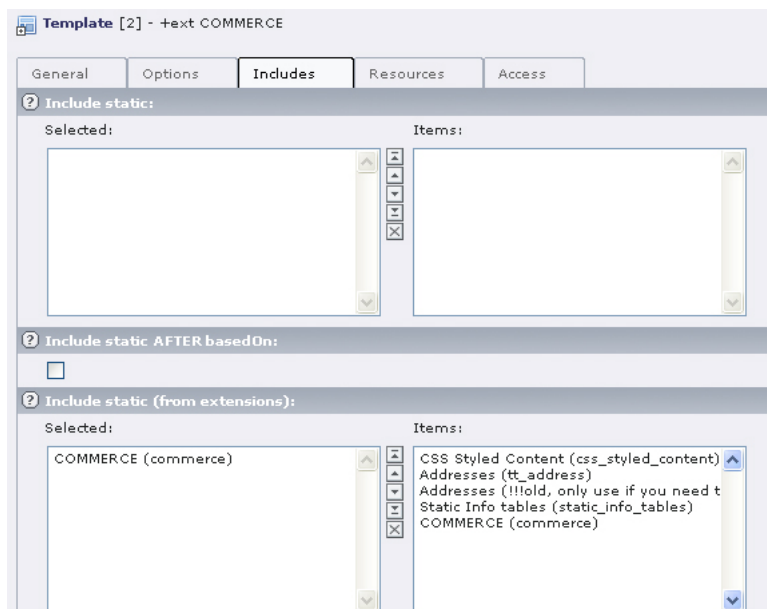


Figure 3.1: Illustration 2: Assign static template

3.2 Pages and Plugins

Now you have to create the pages required for your shop.

3.2.1 List view

„Shop“ will be the main page of your shop, where products and articles are listed. In contrary to other pages which will be added later, this page shall be visible in the menu. Assign the plugin to this page. Select „General plugin“ from the list and **Commerce: Product List** in the form. At the moment we will not consider the plugin-options of the flexforms. You can also ignore the starting point, as it does not need to be defined in the Commerce-shop. Then save the record.

The screenshot shows a web application interface for configuring a plugin. At the top, there are three tabs: 'General', 'Plugin', and 'Access', with 'Plugin' being the active tab. Below the tabs, there's a section labeled 'Plugin:' with a dropdown menu set to 'Commerce: Product Listing'. Underneath, the 'Plugin Options:' section is visible, containing two sub-tabs: 'Product data' (active) and 'Templates'. In the 'Product data' section, there are several fields: 'Display type' is a dropdown set to 'Teaser view'; 'Number topproducts' is a text input field; 'Product:' is a dropdown menu; and 'Category selection' is a complex area with a 'Selected:' list containing 'products' and an 'Items:' list containing 'Category'. At the bottom, there's a 'Startingpoint:' section with a text input field and a 'Page' icon.

Figure 3.2: Illustration 3: Plugin list view

If you want to have a list view, which only displays a distinct product with its related articles (e.g. one type of trousers in different colors) you have to select the desired product in the field **detail product (for single view)** and save the record. This is, of course, not yet possible, as the products and articles do not exist. On this page, you will only see the selected product with the corresponding articles.

3.2.2 Single view

You reach the single view of a product in the frontend by clicking on the product title. It is not necessary to create a separate page for the single view, because it will always be shown on the same page as the list view. A text which you assign to the parent element -instead of the single articles – will be displayed on click on the title of the parent element (which is the product). This corresponds to the single view of the product.

3.2.3 Shopping cart

Create a page for the shopping cart and assign the plugin (general plugin). In the form you have to select **Commerce: shopping cart** from the list of extensions and save the record.

3.2.4 Orders

For the final order you need another shopping cart page. Select the **General Plugin** and **Commerce: Checkout** from the extensions in the form. Then save the record.

3.2.5 Address management

Create a page for the management of addresses: **General Plugin**, in the form **Commerce: Address management**.

3.2.6 Invoice

Create a page for the invoices. You already know how to do this: **General Plugin** and **Commerce: Invoice**.

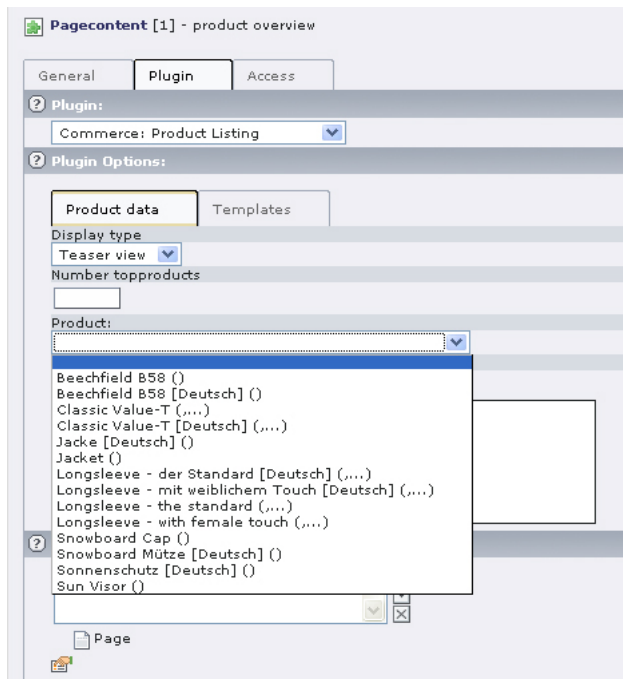


Figure 3.3: Illustration 4: Single product

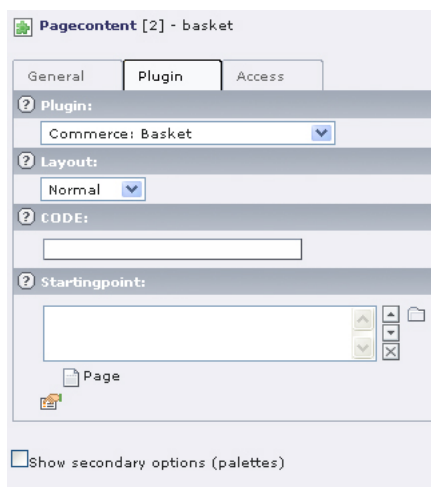


Figure 3.4: Illustration 5: Plugin shopping cart

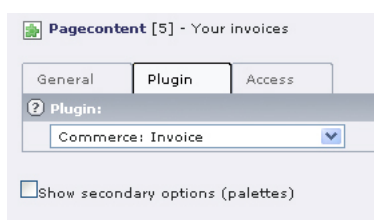


Figure 3.5: Illustration 6: Plugin Invoice

3.2.7 Page and Content „empty shopping cart“ and „empty order“

Create a text page which will be visible in the frontend if the shopping cart is empty or if an order does not yet exist. Create the text for these pages (e.g. „Your shopping cart is empty“ or „Please select an article“) and save the record.

3.2.8 SysFolder

Create two sysFolders, one for user addresses and one for customer addresses. The addresses will be managed by the extension **tt_address** in the folder **addresses**. You need two different folders, as the frontend user registration used by Commerce only knows one address per user, whereas Commerce has a separate address for invoice and delivery. Several addresses are thus possible for every user.

Management of users will be done in the sysFolder Users. The address has to be entered in the checkout at the latest. If the user cancels the order at a later stage their data will not be saved in the system. If the order is completed, user data and orders will be saved in this folder.

With the confirmation of the order the customer will receive their user name and a password they can use for future purchases in the shop.



Figure 3.6: Illustration 7: Page content for an empty shopping cart

3.2.9 Assign PIDs

If all necessary pages for the shop are created, switch to the Extension Manager and assign the PID of the **Invoice** page (Page-ID for Invoice) and the list view Preview Page Type. Finally click **Update** and open the TS-Template +commerce.

Assign the other IDs via the Constant-Editor and make other settings, if required. In order to do so, select the category **PLUGIN.COMMERCE: GENERAL**.

Check the box for every entry in the area **Links and Targets** and click the Update-Button. Now enter the PIDs of the pages.

- Address PID: SysFolder of customer addresses
- USER PID: SysFolder of front end users
- Edit Address PID: address management page
- Basket PID: shopping cart page
- Override PID: list view
- Checkout PID: orders page
- Empty basket PID: text page for empty shopping cart

- Empty Checkout PID: text page for empty order

These settings should be enough for the start. We will use the default values for all other constants.

WEBMODULE COMMERCE

4.1 Webmodul Commerce

Now – or after having cleared the cache – you see the category **Commerce** in the webmodule with its subcategories Category, Orders, Systemdata and Statistics.

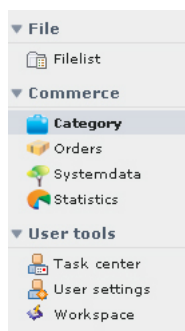


Figure 4.1: Illustration 8: Webmodule Commerce

4.1.1 Categories

Commerce manages products and the corresponding articles in so-called **categories**. There are main categories and subcategories for the management of products and articles.

Go to the web module **Commerce** and click on **category** in order to create the first category. There is an entry *system* in the list, where you have to select *create new record* → **category** and assign a name in the back end form, in our example *Clothing*. It has the UID **2** in our sample configuration, which is preset in the constants already. In case this category should not be your main category, assign the UID of the desired main category in the constants.

Now create a subcategory, ie. within **clothing** another category **t-shirts** and if required another subcategory, e.g. **t-shirts for children**. You can also click on every category with the left mouse button in order to insert additional subcategories using **new subcategory**.

It is not possible to move products to other categories using copy/paste. If you would like to move a product into another category, you have to open the product and assign another category using the form. You can see the corresponding form and a view of the category tree in illustration 9.

4.1.2 Products

You can enter the single products via the category-module, e.g. in **t-shirt** via *create new record* → *products*. Assign a title and a short text for information, add a picture and save the record. All other information such as teaser text, teaser image (for the preview) etc. can be entered using the „**general**“ tab. In the beginning, we will not consider

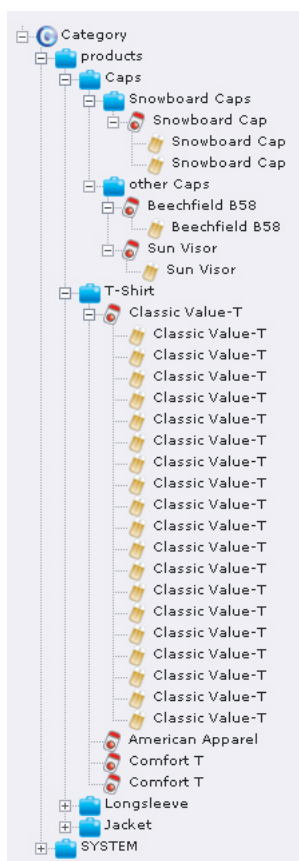


Figure 4.2: Illustration 9: Category tree

the other tabs. Unfortunately it is not possible to save an alternative text for an image in Commerce, which would be required for an accessible shop for people with disabilities.

In the categories' page tree you will now find your product in the category hierarchy, i.e. here in the category clothing the sub-category **t-shirt** with the product **t-shirt**. Within this subcategory there is another subcategory called **t-shirts for children**.

Products are parent elements of all specific versions of a product, i.e. the articles that are really ordered. An article is defined within the category t-shirt by weight of fabric, size, color etc. But before creating an article please have a short look at the subjects payment and delivery, as the corresponding methods are also defined in categories.

4.1.3 Related pages

You can assign a link on any page of the website to a product, making the page a related page. On the General tab in the product record you can select the page and assign it to the product.

4.1.4 Related products

In Commerce you can assign related products to another product. These „related products“ will be displayed in the single view of the product and can also be ordered there. To make this link visible in the frontend, the marker `###RELATED_PRODUCTS###` has to be set in the template `pi1/category_product.tpl`.

Select the desired product in the backend and enter the related products in the „General“ tab.

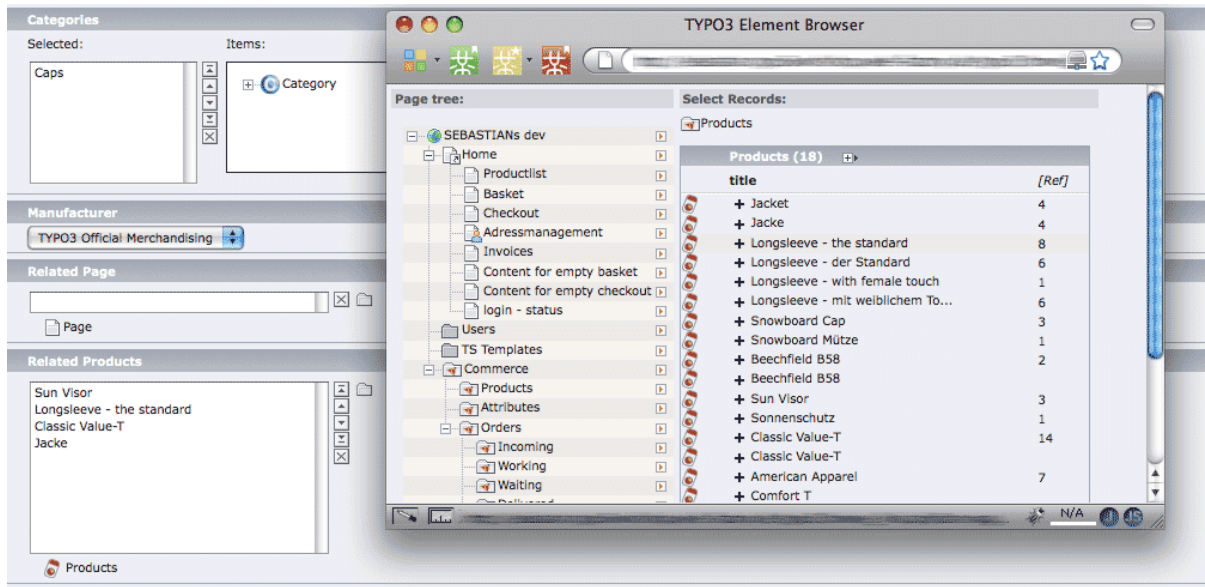


Figure 4.3: Illustration 10: Related products

4.1.5 Payment and delivery

The allowed methods of payment are defined in the **Payment** section. In Commerce the backend form for the payment methods is comparable to the one for the articles. Some methods are pre-set, but can be changed and enlarged. Select for example the payment method **cash on delivery** and change the title in the **general tab** to **Nachnahme**. Move to the price tab and enter the price for **cash on delivery**.

In order to insert a new method of payment, you have to create another „article“. Select the „product“ **PAYMENT** form the list shown in the illustration and the **tab „create article“** in the form that will be displayed. Select the **tab creatable articles**, check the box **create an empty record** and save it. In your category tree there will be a new payment method. You can edit the corresponding record and set up the new payment method. The same procedure is valid for the delivery methods.

4.1.6 Master data

Leave the categories for some other preparations and have a look at master data. You will have to complete the product informations there. Go to the web module **Commerce**, select the entry **Master Data**, which gives access to other entries.

An important feature in Commerce are the attributes. An **attribute** characterizes a product („size“, „color“, „fabric“) and corresponds to a certain **value** („XL“, „red“, „cotton“). The different attributes lead to the articles that will later on be ordered by the customers, e.g. a red cotton t-shirt in size XL. In order to test this, you need to define some attributes first and assign values to them.

Via **new attribute** you can create an attribute (eg. weight), give it a title and a unit (eg. gramme). You can also add an image as symbol for the weight.

Check the box for **„has list of values“**, because as soon as you save the attribute, you can assign values to it. Click the tab **assigned values** and create the corresponding values.

In the example, the values **100**, **200** and **300** were assigned as weight/quality of fabric for shirts. In the backend form you can define that the unit gramme shall be displayed using integer values. After this you can define a second attribute **color** and assign possible values.

An attribute can have an icon, too.

For products you can additionally save information on the **manufacturer**, e.g. *manufacturer of the t-shirt*.

The same applies to **suppliers**. In the sample shop a **t-shirt- supplier** was created.

4.1.7 Products and articles

With these steps, all preparations for the creation of the single articles are done, as the articles will be specified using the attributes and addresses we just defined.

Now go back to the categories.

Select the parent product **t-shirt** in the category tree and there the tab **select attribute**. In the field **selection** you can assign the attribute objects **weight** and **color** and save the record.

Create an article

Switch to the tab **create article** and go there to the section creatable articles, which opens a list with checkboxes. If you do not want to select one of the attributes, you can **create an empty record**, that is one for every article you want to sell for a product. These articles will be assigned attributes from a list box in the tab **existing articles**. If you **select all articles**, all articles that result from the combination of the attribute values will be created. If a t-shirt is available in three colors and three different types of fabric, you will automatically get 9 different articles. If you do not want to offer all combinations, as the white t-shirt is only available with a weight of 100 g, check the corresponding combination of values. After saving the record your article will be displayed beneath the product.

Go back to the parent product **t-shirt** and select the tab **select attributes**.

In the field **Should** you have to set all attributes being only relevant for one single article. It is up to the editor to enter the desired data. In our example, the attribute **washability** was first added without a list of values. If you edit the attributes in the article – not in the product! – you will see a field beneath the heading washability, where you can enter the temperature for washing. In this case, the attribute is only valid for this single article. If you change your mind, you can create a list of values at a later stage. In this case, the values you entered manually, will be automatically integrated, if you check the box for **has list of values**. Of course you can also add other values.

Enter all attributes in the field **Can** that can apply. For example the t-shirt can have an appliqué on the front and on the back side. This attribute has a list of values with the values front and back, and when you edit the attributes in the article, you can assign one of the values.

All attributes with global values belong to the field **product**, ie. those being valid for all articles of the product. An example would be **material**, in case all articles are made of the same material. Create an attribute, assign it via **product** in the product record and save the record. You will then see another tab **edit attributes**, where you can assign the attribute. You can only select one global value for a product and all corresponding articles. In the frontend you will see the single view of the product which is not visible for every article. In our sample shop all t-shirts are made of „100% cotton“.

In the field **filter attributes** you can set those attributes, which will later on be used for the search or other filters. As these filter attributes are very complex and have to be assigned for every shop on an individual basis, every shop owner has to implement those filter attributes by themselves. Currently they are still implemented as an extension and require some programming work. You can set up complex AND or OR connections with individual differences for every shop.

In the article you also have to enter the corresponding tax rate.

In case the tax rate is the same for all articles or for most articles, you can define it via TypoScript, too:

```
mod.commerce.category.defaultTaxValue = 19
```

If the manufacturer is the same for all articles of the product, it will be assigned via the product (not the article). The supplier can depend on the article and the type of delivery and will be defined by one of both categories.

In the **prices** tab of an article you will find several input methods. First, assign a price of EUR 10.- and set **1** for **tier price from** and **tier price to**. If you enter 10 for **tier price to**, the price will be valid for orders up to 10 items of the article.

If you would like to define a different price for 11 – 20 items, please check **add new price** first and save the record. After this you can define another tier price of EUR 8.- for example using the values 11 respectively 20.

Using the wizard you can also create tier prices automatically. You have to enter the the initial quantity in „tier price initial quantity“. In „tier price quantity“ you can fix how many tier prices are to be created. In „tier prices steps“ you will have to enter the extra charge for every price.

Tier prices can also be fixed for a distinct time by defining the start- and stop-date. You can even fix group prices using the field **access**. If you want to offer tier prices for user groups, you have to create a new price or new tier prices for any of those groups. Keep in mind that this can soon become confusing and thus prone to errors. If a user is assigned to several user groups – which should be avoided – he gets the best price.

4.1.8 Shopping Cart

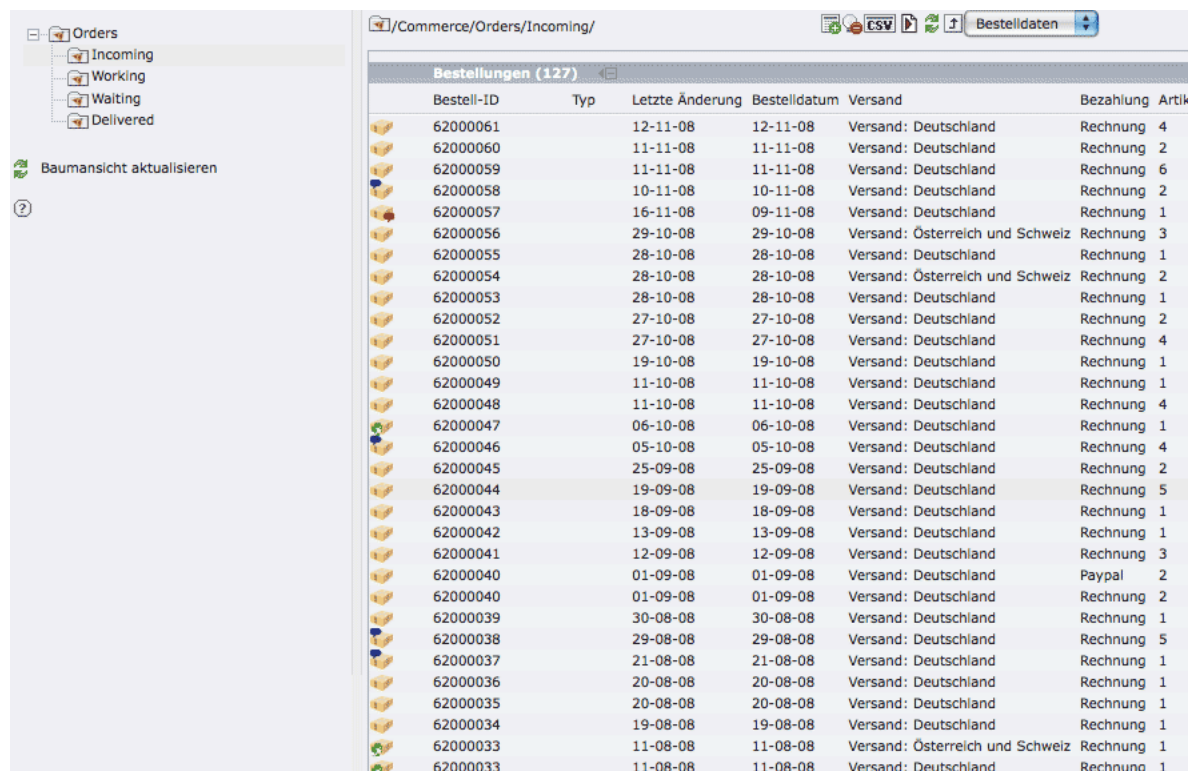
You can start the first test orders from the list view by putting articles in your cart. The shopping cart view you will then see provides you with an overview with prices at the right. The default order is: net total for products, gross total for products, net shipping cost, gross shipping cost, net handling fees, gross handling fees (in case you included those additional costs in your payment methods.)

4.1.9 Orders

Switch in the category **Commerce** to orders. In Incoming you can find all test orders, i.e. all orders that have not yet been processed by the shop administrator. By clicking on the icon in front of the record, you will see the entire order record.

If you click on the email address of a customer, the view will be split up in customer and order data. Using the combobox in the upper right you can switch between those two views.

In order to move the order into another folder, eg. from Incoming to Working, you have to check the box at the end of the order and select the corresponding folder or select the order by clicking on the icon in front of the order (right mouse button → edit). In doing so you can see the status of the order (tab customer date), **delivery address** and **article**, until the goods are delivered and the transaction is closed.



Bestell-ID	Typ	Letzte Änderung	Bestelldatum	Versand	Bezahlung	Artikel
62000061		12-11-08	12-11-08	Versand: Deutschland	Rechnung	4
62000060		11-11-08	11-11-08	Versand: Deutschland	Rechnung	2
62000059		11-11-08	11-11-08	Versand: Deutschland	Rechnung	6
62000058		10-11-08	10-11-08	Versand: Deutschland	Rechnung	2
62000057		16-11-08	09-11-08	Versand: Deutschland	Rechnung	1
62000056		29-10-08	29-10-08	Versand: Österreich und Schweiz	Rechnung	3
62000055		28-10-08	28-10-08	Versand: Deutschland	Rechnung	1
62000054		28-10-08	28-10-08	Versand: Österreich und Schweiz	Rechnung	2
62000053		28-10-08	28-10-08	Versand: Deutschland	Rechnung	1
62000052		27-10-08	27-10-08	Versand: Deutschland	Rechnung	2
62000051		27-10-08	27-10-08	Versand: Deutschland	Rechnung	4
62000050		19-10-08	19-10-08	Versand: Deutschland	Rechnung	1
62000049		11-10-08	11-10-08	Versand: Deutschland	Rechnung	1
62000048		11-10-08	11-10-08	Versand: Deutschland	Rechnung	4
62000047		06-10-08	06-10-08	Versand: Deutschland	Rechnung	1
62000046		05-10-08	05-10-08	Versand: Deutschland	Rechnung	4
62000045		25-09-08	25-09-08	Versand: Deutschland	Rechnung	2
62000044		19-09-08	19-09-08	Versand: Deutschland	Rechnung	5
62000043		18-09-08	18-09-08	Versand: Deutschland	Rechnung	1
62000042		13-09-08	13-09-08	Versand: Deutschland	Rechnung	1
62000041		12-09-08	12-09-08	Versand: Deutschland	Rechnung	3
62000040		01-09-08	01-09-08	Versand: Deutschland	Paypal	2
62000040		01-09-08	01-09-08	Versand: Deutschland	Rechnung	2
62000039		30-08-08	30-08-08	Versand: Deutschland	Rechnung	1
62000038		29-08-08	29-08-08	Versand: Deutschland	Rechnung	5
62000037		21-08-08	21-08-08	Versand: Deutschland	Rechnung	1
62000036		20-08-08	20-08-08	Versand: Deutschland	Rechnung	1
62000035		20-08-08	20-08-08	Versand: Deutschland	Rechnung	1
62000034		19-08-08	19-08-08	Versand: Deutschland	Rechnung	1
62000033		11-08-08	11-08-08	Versand: Österreich und Schweiz	Rechnung	1
62000033		11-08-08	11-08-08	Versand: Deutschland	Rechnung	1

Figure 4.4: Illustration 11: Order list

There is an icon in front of every order. If the order contains a comment from the shop, the icon will get a red flag. If there is a comment from the customer, there will be a blue icon. And if the order shall be sent to an alternative adress, there will be a green flag.

Moving orders into another folder

Orders can be moved into other folders, as soon as the status of the order has changed, e.g. from „incoming“ to „working“.

Deleted and cancelled orders should immediately be moved into a cancellation-folder, which will not be included in the aggregation (comp. Statistics). Otherwise the amounts will go into your „real“ orders and distort your statistics.

In order to exclude orders from aggregation, you have to enter the UUIDs in the constant editor in „excludeStatistic-Folders“.

Creating a new folder for cancelled orders

Create this folder as sysFolder in the „orders“-area. In the field includes extension you can assign **Commerce system folder** (see illustration).

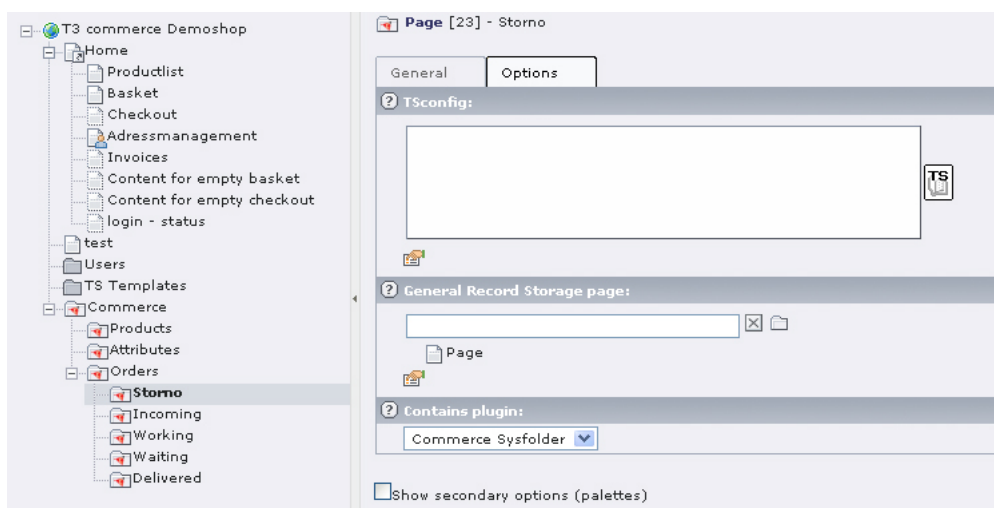


Figure 4.5: Illustration 12: creating a folder for cancellations

In the web module menu {Commercesm orders} you can now move your cancelled orders via the combobox into this folder.

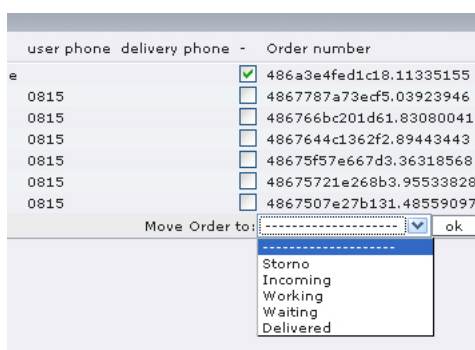


Figure 4.6: Illustration 13: Move order in cancellation folder

Export orders

If you would like to export all orders of a folder, including all available fields, then export the file via the csv-symbol and re-include it into the desired program. You can also write a short php-script and integrate all data directly into your invoicing system.

Types and states of order

A type of order classifies the order. It can be done manually. The same applies to the users' state. You could for example define the order types B2C and B2B and assign one of them to any order.

You can also assign a good or a bad payment history to customers.

Types and states of orders will be set using the web module list in the folder **orders** (create new record).

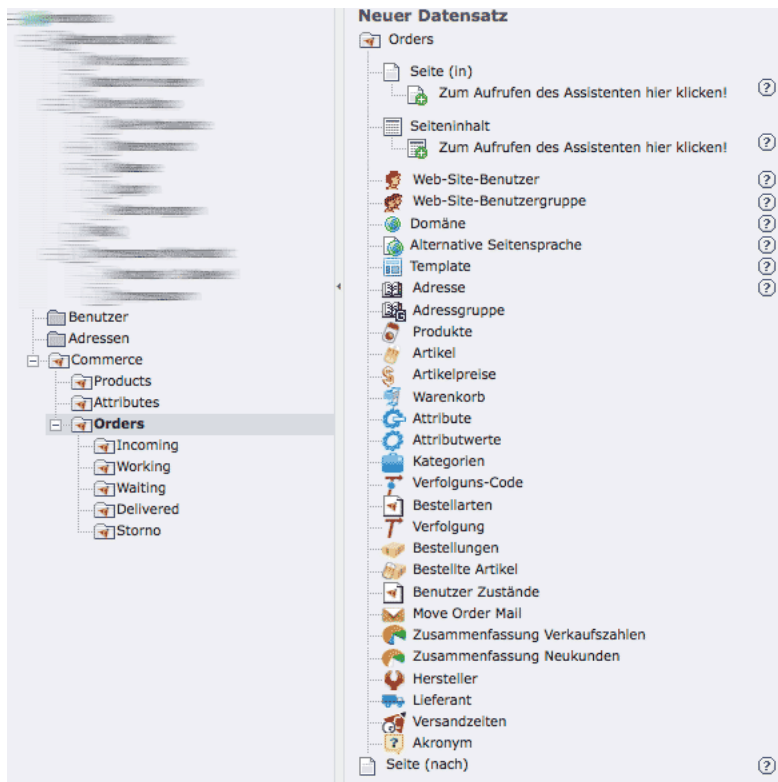


Figure 4.7: Illustration 14: Create new record: type and status of orders

As soon as you have created a type of order, e.g. B2C, you can select it using the combobox type of order and assign it.

A user state will be set in the folder customersysFolder and assigned to the user, using the corresponding listbox on the „advanced“ tab.

4.1.10 Statistics

Stay in the web module Commerce, but go to statistics, in order to view the statistics for your orders. In this context, the term **aggregation** is important, i.e. the total of values in a table. In this case it is the total of orders. The statistics will be saved in so-called aggregation-tables. You have to start the complete aggregation once.

There might be a great amount of data for capture and processing. It is thus possible that the time needed for calculation exceeds the maximum allowed script runtime (which is set to 30 seconds by default). If you get an error

Bestellungen [72] - 62000056

Basis Kundendaten Lieferadresse Artikel

Bestell-ID
62000056

Bestelldatum
29-10-2008

Bestellart
B2C

Währung ISO-Code
Euro

Bestellungsstatus
Incoming

Bezahlungsart
Rechnung

Figure 4.8: Illustration 15: Select type of order



Figure 4.9: Illustration 16: Aggregation

message such as „...Maximum execution time ...exceeded“, you have to increase the maximum script runtime in the php.ini-file by entering a higher value for max_execution_time (for example 300 for 5 minutes).

You can also run the generation process as CLI-script, for example as regular cron job:

```
/pfad_zur_installation/typo3/cli_dispatch.phpsh commerce statistics incrementalAggregation
/pfad_zur_installation/typo3/cli_dispatch.phpsh commerce statistics completeAggregation
```

It will capture and evaluate all orders. After this you only need to activate the surplus aggregation, in order to get changes in the database. This will take much less time then a complete evaluation.

With the selection statistics in the list box you will get an overview, which does not include the orders of the current day, though. Statistics for single days, months or hours will also be possible.

STATISTIC:			
7 - 2008			
Days	turnover	amount	orders
1	0.00	10	1
2	0.00	0	0
3	0.00	0	0
4	0.00	0	0
5	0.00	0	0
6	0.00	0	0
7	0.00	0	0
8	0.00	0	0
9	0.00	0	0
10	0.00	0	0

Figure 4.10: Illustration 17: Daily statistics

If you would like to print or further process statistics, then simply copy the entire table und paste it into a spreadsheet.

CONFIGURATION

5.1 Configuration

5.1.1 Move Order Mail

Another interesting feature of Commerce are the so-called **Move Order Mails**, automatic emails, which you can send internally and to customers as soon as the status of an order has changed and the order was moved to another folder. In this way you can inform the customer, as soon as you received their payment or sent out the goods.

You can also inform your staff in the shipping department that the goods should be prepared for shipping.

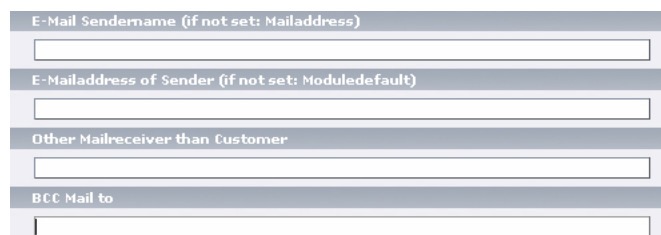
Go to list view of orders in the web module, click on the order Delivered for which you would like to enter the change message and enter a new record via *Move Order Mail*.

Assign a title for this mail and define first when it has to be sent (i.e. when it is moved into this folder). Select for example Mail *when move in* and assign a HTML-Template for this message via *Reference to Mailtemplate*. You will find the template for this example under *res/auslieferung.html*.

Other templates for Move Order Mails are available in the same folder under *paketfreigabe.html* and *speditionsfreigabe.html*. You can of course include adaptations and templates of your own.

Unless you change further settings, this message will be sent to the customer as soon as their order changes to the status **delivered**.

If you want to send the mail to additional recipients, e.g. one to the customer, a second one to the shop owner and a third to the scheduler, you can use the BCC-field. For every recipient and for every Move Order Mail you can use an individual template with individual text.



The image shows a web form for configuring a Move Order Mail. It consists of four sections, each with a label and a text input field:

- E-Mail Sendname (if not set: Mailaddress)**: Input field.
- E-Mailaddress of Sender (if not set: Moduledefault)**: Input field.
- Other Mailreceiver than Customer**: Input field.
- BCC Mail to**: Input field.

Figure 5.1: Illustration 18: Move Order Mail - email recipient

HTML-Template

If you want to use an individual HTML Shop template instead of the default settings, you will find it in the folders **typo3conf/ext/commerce/pi1** to **../pi6**. It is not possible to copy these files as usual to your template-folder in fileadmin using the extension manager. But you may download them via FTP from the webserver and move them afterwards. For processing you have to change the file types from **.tpl** to **.tmpl** or **.html**. You can also add the file type as allowed changes via the install tool.

Constants

In the constants editor you will find the following settings - in our test shop the PIDs mentioned in the beginning of the chapter and one of the templates are assigned:

```
plugin.tx_commerce_lib.addressPid = 18
plugin.tx_commerce_lib.userPid = 17
plugin.tx_commerce_lib.basketPid = 11
plugin.tx_commerce_lib.checkoutPid = 12
plugin.tx_commerce_lib.emptyBasketPid = 15
plugin.tx_commerce_lib.emptyCheckoutPid = 16
plugin.tx_commerce_lib.editAddressPid = 13
plugin.tx_commerce_lib.overridePid = 2
plugin.tx_commerce_pi1.templateFile = fileadmin/templates/commerce/category_product.html
```

With this configuration Commerce already offers an operational shop. Further information on the configuration of Commerce can be found in the forums and in the mailing lists. A short installation guide comes with the extension. For the sample shop the implementation of this very comprehensive shop system ends here.

Please keep in mind that Commerce has far more to offer than the basic installation which was described here.

5.1.2 Hooks

The list of all available hooks can be found in the directory /docs in the file hooks.html. We will now show you some examples for the usage of a hook.

In order to introduce new product fields (FIELD1 and FIELD2) you can proceed as follows:

First of all, create a new extension using the kickstarter. Enlarge the table in the kickstarter where you want to add fields. For product fields you have to use the table tx_commerce_products. Create a new file within the new extension, where you add a class with the following functions:

```
function postInit(&$model) {
    $model->add_fields_to_fieldlist (array('FIELD1', 'FIELD2'));
}

function additionalMarker($markerArray, $singleViewPlugin) {
    $markerArray['###FIELD1###'] = $singleViewPlugin->product->getField('FIELD1');
    $markerArray['###FIELD2###'] = $singleViewPlugin->product->getField('FIELD2');
    return $markerArray;
}
```

Include the following code in localconf.php in the root folder of this extension, in order to include the function:

```
require_once (t3lib_extMgm::extPath($_EXTKEY) . 'PFAD/ZU_IHRER_HOOK_KLASSE.php');
$GLOBALS['TYPO3_CONF_VARS']['EXTCONF']['commerce/lib/class.tx_commerce_piibase.php']['singleview']
$GLOBALS['TYPO3_CONF_VARS']['EXTCONF']['commerce/lib/class.tx_commerce_product.php']['postinit'] [
```

You can now use the marker `###FIELD1###` and `###FIELD2###` in the template of the product view (pi1/category_product.tpl). The fields have to be entered in the extension configuration in the field „fields for products“ to be considered.

5.1.3 Reference

Plugin Commerce: General

Constants

The following table contains the overall attributes from the global settings. All features can be set using the constant editor.

fields for products... [coProducts]
that should be displayed in simple mode

hidden[starttime|endtime|fe_group],title,subtitle[navtitle],description,images,categorie

fields for articles... [coArticles]
that should be displayed in simple mode

ordernumber,eancode,prices

Define the Charset for the csv File... [BECSVCharset]
Define the Charset for the csv Files in the order module

iso-8859-1

Field mapping feuser-tt_address [feuser_address_mapping]
Specify list of feuser <-> address field mappings. Use this format: fe_usersfield,tt_addressfield

company,company;name,surname;title,title;address,address;zip,zip;city,city;country,co

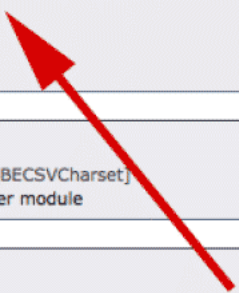


Figure 5.2: Illustration 19: Fields for products

Property	Data type	Description	De- fault
addressPid	Integer	PID of the SysFolders, where addresses are stored	0
userPid	Integer	PID of the SysFolder, where new users are stored - tt_address	0
editAddressPid	Integer	PID of the page, where address data can be processed	0
basketPid	Integer	PID of the shopping cart page	0
emptyBasketPid	Integer	PID of the page, where the content element is located, which will be displayed when the shopping cart is empty; will be called via the shopping cart page.	0
emptyCheckoutPid	Integer	PID of the page, where the content element is located, which will be displayed if the checkout is not possible; will be called via the shopping cart page (empty order)	0
checkoutPid	Integer	PID of the page of the shopping cart checkout()	0
overridePid	Integer	PID of the page with the list view	0
paymentArticleId	Integer	article UID of the default method of payment	1
payProdId	Integer	product UID of the default method of payment	1
delProdId	Integer	UID of the default method of delivery	2
catUid	Integer	default category UID in the frontend	2
currency	String	ISO-code for the currency (from static_info_tables)	EUR
regularArticleTypes	String	regular type of article (article = 1, payment = 2, delivery = 3)	1
UserGroup	Integer	UID of user group for new customers	1
UseRootline-Information-ToUrl	Integer	Adds new parameters to the URL in order to speed up presentation of the navigation	empty
priceFrom-Net	Boolean	Consider net price for tax calculation	empty
useStockHandling	Boolean	Stock management	empty
currencyId	Integer	UID of the ISO codes for currencies from (static_info_tables)	49
5.1. Configuration			25
showErrors	Boolean	Error message in frontend	1
displayProductIfOne-Product	Boolean	If a category has one product only, the product list view will be skipped and the detail page will immediately be opened.	1

[tx_commerce_lib.constants]

Plugin Commerce: Display

Property	Data type	Description	Default
templateFile	String	Path to the HTML template	EXT:commerce/pi1/category_product.tpl
attributeLine-breakChars	String	Line break	

[tx_commerce_pi1]

Plugin Commerce: Basket

Property	Data type	Description	Default
templateFile	String	Path to the HTML template of the shopping cart	EXT:commerce/pi2/shoppingcarttpl
basketStoragePid	Integer	Basket storage pid	0

[tx_commerce_pi2]

Plugin Commerce: Checkout

Property	Data type	Description	Default
TemplateFile	String	Path to the HTML template for administrator – mail upon new orders	EXT:commerce/pi3/template_adminmail.tpl
AdminMail-TemplateFile	String	Path to the HTML template for the checkout (finish order)	EXT:commerce/pi3/template_checkout.tpl
UserMailTemplateFile	String	Path to the HTML template for customer information after reception of order	EXT:commerce/pi3/template_usermail.tpl

[tx_commerce_pi3]

Plugin Commerce: Address Management

Property	Data type	Description	Default
template-File	String	Path to HTML Template of address manager	EXT:commerce/pi4/template_addresses.
minAddress-Count	Integer	Defines the minimum number of addresses, a customer must have. The customer can only delete addresses, if the minimum value is exceeded.	1

[tx_commerce_pi4]

Reference lib.tx_commerce.articles

Property	Data type	Description	De-fault
stdWrap	stdWrap	Wrap around single article	
defaultField	stdWrap	Standard Wrap around every field which is not defined in the fields selection	
DefaultQuantity	Integer	Standard amount of articles. Replaces tx_commerce_pi1.defaultArticleAmount	0
fields	COA	The field-objekt -> see Reference fieldObject	
addToBasketLink	TypoLink	Typolink feature which allows putting articles directly in the shopping cart	

Reference fieldObject

A FieldObject is a field list, which can be defined as IMGTEXT, IMAGE, IMG_RESOURCE, STDWRAP. Every database field in a table can be defined as such an object.

Example

```
fields {
    // every column in this table could be wrapped with the
    // stdWrap, IMGTEXT, IMAGE

    title = stdWrap
    title {
        wrap = |
    }
    description_extra = stdWrap
    description_extra {
        parseFunc < lib.parseFunc_RTE
    }
    images = IMAGE
    images {
        defaultImgConf {
            file.import.current = 1
        }
        file.maxW = 150
    }
}
```

For the field types IMGTEXT, IMAGE, IMG_RESOURCE the Property imgPath can be defined. If this is not defined in TS, the defaultpath uploads/tx_commerce will be used.

Property	Data type	Description	Default
imgPath	String	Path to the image files; has to end with a / (Slash)	If not defined: uploads/tx_commerce

Reference pi1 (Listing)

Property	Data type	Description	Default
defaultArticleAmount	Integer	Standard number of articles	
maxRecords	Integer	Maximum amount of records shown per page	
templateMarker	String	Configuration of single template-marker	
allArticles	Boolean	Defines whether all articles or a selectbox will be displayed for a selection of articles. For select boxes the value has to be set to 0. JavaScript is needed for selectboxes	1
hideProductsInList	Boolean	Hide all products in the category list (show categories only)	
displayTopProducts	Boolean	Display of top products	
checkCategoryTree	Boolean	If activated, the subcategories submitted with GET-parameter will be checked against the main categories, which are also submitted. If the subcategory is not below this main category, the main category will be shown.	
hideEmptyProdAttr	Boolean	Hide empty product attributes	
hideEmptyShalAttr	Boolean	Hide empty „Shall“-attributes	
hideEmptyCanAttr	Boolean	Hide empty „Can“-attributes	
showHiddenValues	Boolean	Show hidden values	
singleView	COA		
.renderProductNameAsPageTitle	Integer	0. don't render product title as page title 1. render always product title as page title 2. only render product title as page title if product was not set by plugin settings (select product) and if product was not set by TS	
.categories	COA	TS Setup for display in a single category	
listView	COA	configuration of product-list view	
listViewTop	COA	configuration of top-product list view	
general_stdWrap	stdWrap	overall stdWrap	
categoryListView	COA	configuration of category list	
attributeLinebreakChars	String	String for separation of attribute values in select boxes	
5.1. Configuration			29
mutipleAttributeValueWrap	stdWrap	stdWrap around every attribute value for several selected attributes	

[tx_commerce_pi1]

Reference pi2 (Basket)

Property	Data type	Description	De- fault
templateFile	String	Template file	
delProdId	Integer	UID of the default delivery method (delivery product ID)	
payProdId	Integer	UID of default – payment method (payment product ID)	
regularArticle-Types	String	Regular type of article (article = 1, payment =2 , delivery = 3)	
listPid	Integer	PID of the list view	
checkoutPid	Integer	PID of checkout	
basketPid	Integer	PID of basket	
basketStoragePid	Integer	PID of basket storage	
defaultPaymentArticleId	Integer	ID of article of default payment method	
templateMarker	String	configuration of template marker	
fields	COA	the fieldObject -> see reference fieldObject	
payment.allowedArticles	String	Comma separated list of allowed methods of payment (UIDs). This selection can be overwritten by the hook makePayment	false
delivery.allowedArticles	String	Comma separated list of allowed methods of delivery (UIDs). This selection can be overwritten by the Hook makeDelivery	false

Reference pi3 (Checkout)

Property	Data type	Description	De- fault
templateFile	String	HTML Template for the order page	
addressMgmPid	Integer	The PID for the page which will be used for the address management.	
regularArticleTypes	String	Regular article type, see constants	
usermail	-> mail	Configuration of confirmation mail, which the customer receives	
adminmail	-> mail	Configuration of confirmation mail, which the administrator receives	
currency	String	ISO-code of currency	
billing	-> form	Forms configuration for billing adress	
.deliveryAd- dress.delivery_radio	std- Wrap	Wrap for the radio button for the selection of a delivery address	
.deliveryAd- dress.nodelivery_radio	std- Wrap	Wrap for the radio button in case no billing address equals a delivery address	
.deliveryAd- dress.delivery_label	std- Wrap	Wrap for the label for the selection of a delivery address	
.deliveryAd- dress.nodelivery_label	std- Wrap	Wrap for the label, of no billing adress equals a delivery address	
delivery	-> form	The forms configuration for the delivery address	
createNewUsers	Boolean	If set, a new user will be created during checkout, if the customer has not been logged in.	
userPID	Integer	PID of SysFolder with the frontend users	
userGroup	Integer	The UID of the user group which will be assigned to new users, in case registration was finished regularly.	
termsdefaultchecked	Boolean	Terms and conditions selected by default?	
paymentIsDeliv- eryAdressDefault	Boolean	Default setting, whether invoice and delivery address are the same.	
deliveryAdressIsSep- arateDefault	Boolean	Default setting, whether there is a separate delivery address	
addressPid	Integer	PID of the sysFolder with tt_address data	
cantMakeCheckout	Boolean	disallows checkout	

[plugin.tx_commerce_pi3]

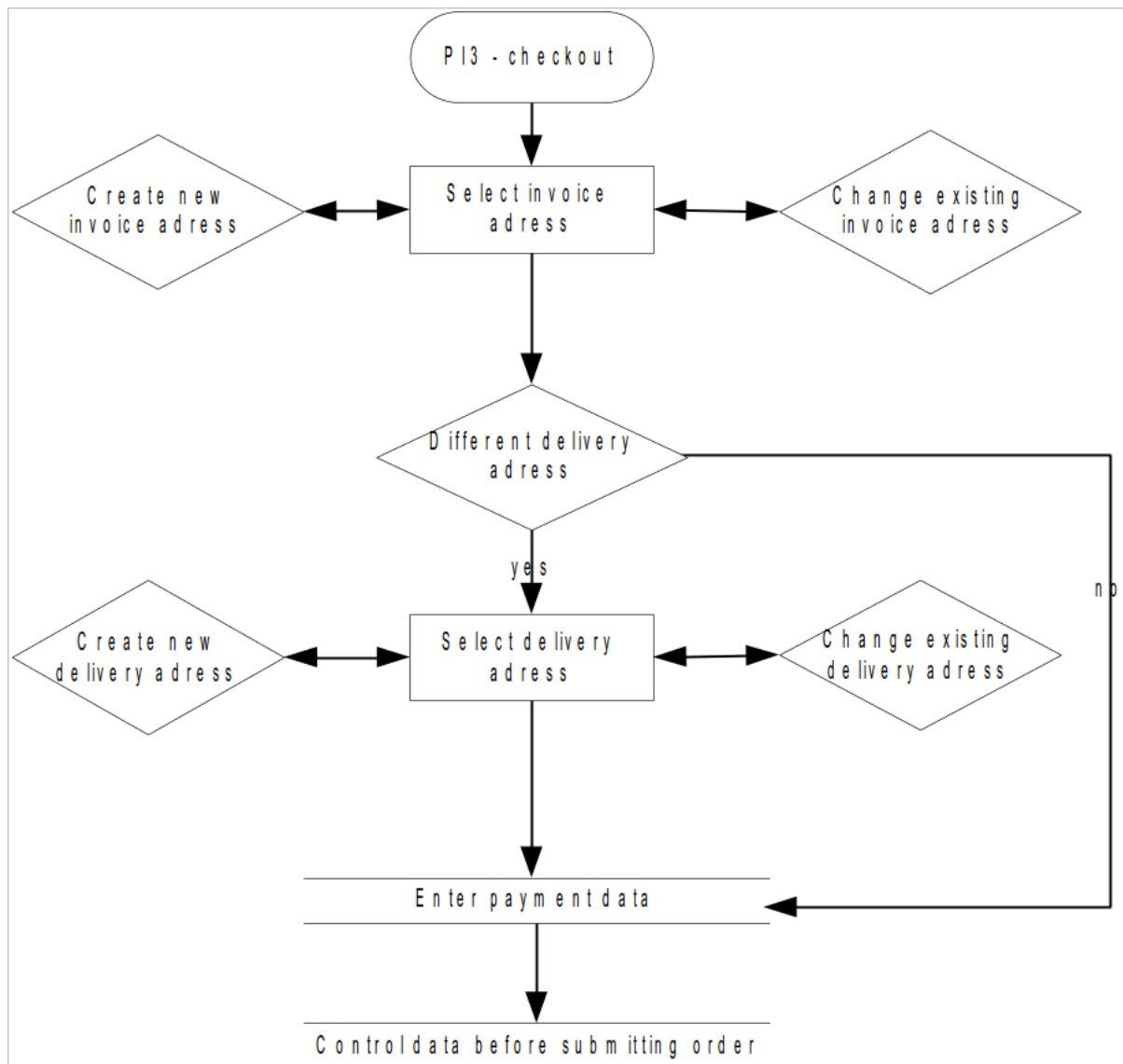


Figure 5.3: Illustration 16: Checkout process

Reference pi4 (Address management)

Property	Data type	Description	Default
templateFile	String	The HTML template for the address management plugin	
minAddressCount	Integer	Minimal amount of addresses which have to be allocated to a single user. Is used to check whether an address can be deleted.	1
selectAddressTypes	String	Comma separated list of address types which can be processed by a plugin.	1,2
editAddressPid	Integer	PID of the page where address data can be processed	
addressPid	Integer	PID of the sysFolder with tt_address data	
addressHeaderWrap	stdWrap	Wrapper for the heading of the list of addresses	
editLinkWrap	stdWrap	Wrapper for the link for processing of addresses	
deleteLinkWrap	stdWrap	Wrapper for the link to delete an address	
newLinkWrap	stdWrap	Wrapper for the link to enter an address	
yesLinkWrap	stdWrap	Wrapper for the confirmation button to delete an address	
noLinkWrap	stdWrap	Wrapper for the abort button for deletion of an address	
sysMessageWrap	stdWrap	Wrapper for all messages coming from the system, e.g. error messages	
hideEmptyFields	Boolean	If set, empty fields will be removed from the list view; does not concern the processing view.	0
emptyFieldSign	String	If hideEmptyField is not set: Marker, which will be shown instead of a field value.	
formFields	-> form	Defines the form fields, which are used for processing of addresses	
mandatorySign	String	Sign for mandatory fields	•

[plugin.tx_commerce_pi4]

Reference pi6 (Invoices)

Property	Data type	Description	Default
decode	Boolean	Set to 1, if you want to create invoices as pdf-file	0
currency	String	ISO-code of currencies	
showCurrencySign	Boolean	If 1, the currency symbol will be printed in the invoice	
invoice-header	Image resource	If a screen resource is given, the image will be included in the standard HTML template of the invoice (letter head).	
shopname	stdWrap	Name of the shop	
shopdetails	COA	Additional information concerning the shop, e.g. the address	
intro	stdWrap	Introductory text which will be displayed in the beginning of the default template (welcome to the shop...)	
thankyou	stdWrap	Final phrase in the end of the default templates (Thank you very much for shopping with us...)	

[plugin.tx_commerce_pi6]

Reference navigation.special

Property	Data type	Description	Default
category	Integer	CategoryUID: Starting category for the navigation	{ \$plugin.tx_commerce_lib.catUid }
show-Products	Boolean	If products should be shown in the navigation	0
overridePID	Integer	PID for generating Links in the Navigation	{ \$plugin.tx_commerce_lib.overridePid }
entryLevel	Integer	Entry Level for the navigation	0
maxLevel	Integer	Maximum Level to be generated in the Navigation. Since a complete Array of all levels (regardless the Menu Configuration) will be rendered, this could use much memory	PHP_INT_MAX
noAct	Boolean	If activated the navigation will only generate menu elements in the stat NO. If you don't need any active elements in the navigation, this could speedup the generation process	0
hideEmptyCategories	Boolean	Hides empty Categories	{ \$plugin.tx_commerce_lib.hideEmptyCategories }
additionalFields	String	Additional Fields which could be added to the Menu Navigation	
expandAll	Boolean	Expand all. This must be set in special not in MENU object	

[lib.tx_commerce.navigation.special]

-> mail

Property	Data type	Description	Default
showCurrency	Boolean	If set, the currency will be shown in the confirmation mail.	0
charset	String	Charset of the mail	utf-8
encoding	String	Encoding of the mail	8bit
from	String	The sender address of the shop	shop@example.com
from_name	String	name of the sender	Commerce ShopMaster
cc	String	Additional CC-addresses, comma-separated	
bcc	String	Additional BCC-addresses, comma-separated	
mailto	String	Administrator's email	administrator@example.com
templateFile	String	HTML-template of the mail	

-> form

The Commerce forms are not static. They will be rendered dynamically depending on the configuration. The options will be set via TypoScript.

Overall settings:

Property	Data type	Description	Default
addressType	Integer	Address type to be processed in this form	
userConnection	String	Defines the field in the source table (e.g. tt_address) makes the connection to the frontend user.	tx_commerce_fe_user_id
sourceTable	String	Table, where addresses are stored	tt_address
sourceLimiter	-> sourceLimiter	Mandatory fields	
sourceFields	-> sourceFields	Fields to be displayed	


Two of the features are of secondary importance:

-> sourceLimiter

Property	Data type	Description	Default
field	String	Default field	tx_commerce_address_type_id
value	String	Default value	

-> sourceFields

Die Sourcefields definieren für jedes einzelne Feld einer Tabelle wie es verarbeitet und gerendert wird. Der Schlüssel ist immer der Name der Quelltable. Jedes Feld hat spezielle Eigenschaften.

Property	Data type	Description	Default
type	String	Defines rendering of the field in the frontend. Possible values: <ul style="list-style-type: none"> • single (simple input field) • select (Select-box) 	input
mandatory	Boolean	Mandatory field	0
eval	String	Validation of single fields. The comma-separated list of methods: <ul style="list-style-type: none"> • email (entered value must be a valid email-address) • username (checks, whether this username already exists) • string (checks, whether the entered value is a string) • int (checks, whether the entered value is an integer) • min (minimum value. Has to be separated by the key-word with an underscore, e.g. min_10) • max (maximum, see min) • alpha (the entered value must be alpha-numeric) 	
table	String	Defines the table, the values come from	
select	String	Where-Clause for the SELECT-query	1
label	String	The table-field which shall be assigned as field name	
value	String	The field that is used as value for the option elements.	
default	String	Default value, these values will be set as default in the form	
readonly	Boolean	If set, the field cannot be changed in the frontend	
Maxlength	Integer	Maximum field-length of an input field; no validation by server!	
36		inspiring people to share.	 TYP03
orderby	String	Field name by which the results are sorted.	

PageTSconfig

Property	Data type	Description	De- fault
cate- gory.defaultTaxValue	Integer	Default tax rate	
orders.delProdUid	Integer	Limitation of display to orders with the given product.	
orders.payProdUid	Integer	Limitation of display to orders with the given payment method.	
cate- gory.defaultTaxValue	Integer	Default tax rate, if not set for the article	

[mod.commerce]

5.2 Realurl fixed post vars

The following configuration enables urls like <http://www.example.com/de/somepath/CategoryName/ProductName.html>
This prevents an extra .../product/...

ext_localconf.php:

```
$TYPO3_CONF_VARS['EXTCONF']['realurl'] = array(
    '_DEFAULT' => array(
        'init' => array(
            'enableCHashCache' => 1,
            'appendMissingSlash' => 'ifNotFile',
            'enableUrlDecodeCache' => 1,
            'enableUrlEncodeCache' => 1,
        ),
        'fileName' => array(
            'defaultToHTMLsuffixOnPrev' => 1,
            'index' => array(
                'index.html' => array(
                    'keyValues' => array(
                    ),
                ),
            ),
        ),
    ),
    'preVars' => array(
        array(
            'GETvar' => 'L',
            'valueMap' => array(
                'de' => '0',
                'en-gb' => '1',
            ),
            'valueDefault' => 'de',
        ),
    ),
    'fixedPostVars' => array(
        'txcommerceDetailConfiguration' => array(
            array(
                'GETvar' => 'tx_commerce_pil[catUid]',
                'lookUpTable' => array(
                    'table' => 'tx_commerce_categories',
                    'id_field' => 'uid',
                    'alias_field' => 'title',
                    'maxLength' => 120,
                    'useUniqueCache' => 1,
                    'addWhereClause' => ' AND NOT deleted',
                    'enable404forInvalidAlias' => 1,
                ),
            ),
        ),
    ),
);
```

```

        'autoUpdate' => 1,
        'expireDays' => 5,
        'useUniqueCache_conf' => array(
            'spaceCharacter' => '-'
        )
    ),
    array(
        'GETvar' => 'tx_commerce_pil[showUid]',
        'lookUpTable' => array(
            'table' => 'tx_commerce_products',
            'id_field' => 'uid',
            'alias_field' => 'title',
            'maxLength' => 120,
            'useUniqueCache' => 1,
            'addWhereClause' => ' AND NOT deleted',
            'enable404forInvalidAlias' => 1,
            'autoUpdate' => 1,
            'expireDays' => 5,
            'useUniqueCache_conf' => array(
                'spaceCharacter' => '-'
            )
        ),
    ),
    // uid of the detail commerce page on live
    '62' => 'txcommerceDetailConfiguration',
    '217' => 'txcommerceDetailConfiguration',
),
'postVarSets' => array (
    '_DEFAULT' => array(
        'plaintext' => array(
            'type' => 'single',
            'keyValues' => array(
                'type' => 99
            )
        ),
    ),
),
'pagePath' => array (
    'type' => 'user',
    'userFunc' => 'EXT:realurl/class.tx_realurl_advanced.php:&tx_realurl_adva
    'spaceCharacter' => '_',
    'languageGetVar' => 'L',
    'expireDays' => 3,
    'rootpage_id' => 1,
    'segTitleFieldList' => 'tx_realurl_pathsegment,alias,nav_title,title',
),
),
);

```

5.3 Realurl post var sets

The following configuration enables urls like <http://www.example.com/de/somepath/product/CategoryName/ProductName.html>. This adds an extra .../product/... but adds also the benefit of being able to add additional shop pages without the need to add more configuration. So a editor is able to add pages without the need to have a developer configure more realurl settings.

ext_localconf.php:

```

$TYPO3_CONF_VARS['EXTCONF']['realurl'] = array(
    '_DEFAULT' => array(
        'init' => array(
            'enableCHashCache' => 1,
            'appendMissingSlash' => 'ifNotFile',
            'enableUrlDecodeCache' => 1,
            'enableUrlEncodeCache' => 1,
        ),
        'fileName' => array(
            'defaultToHTMLSuffixOnPrev' => 1,
            'index' => array(
                'index.html' => array(
                    'keyValues' => array(
                    ),
                ),
            ),
        ),
    ),
    'preVars' => array(
        array(
            'GETvar' => 'L',
            'valueMap' => array(
                'de' => '0',
                'en-gb' => '1',
            ),
            'valueDefault' => 'de',
        ),
    ),
    'fixedPostVars' => array(
    ),
    'postVarSets' => array(
        '_DEFAULT' => array(
            // commerce configuration
            'product' => array(
                array(
                    'GETvar' => 'tx_commerce_pil[catUid]',
                    'lookUpTable' => array(
                        'table' => 'tx_commerce_categories',
                        'id_field' => 'uid',
                        'alias_field' => 'title',
                        'maxLength' => 120,
                        'useUniqueCache' => 1,
                        'addWhereClause' => ' AND NOT deleted',
                        'enable404forInvalidAlias' => 1,
                        'autoUpdate' => 1,
                        'expireDays' => 5,
                        'useUniqueCache_conf' => array(
                            'spaceCharacter' => '-'
                        )
                    )
                ),
                array(
                    'GETvar' => 'tx_commerce_pil[showUid]',
                    'lookUpTable' => array(
                        'table' => 'tx_commerce_products',
                        'id_field' => 'uid',
                        'alias_field' => 'title',
                        'maxLength' => 120,
                        'useUniqueCache' => 1,
                        'addWhereClause' => ' AND NOT deleted',
                        'enable404forInvalidAlias' => 1,
                        'autoUpdate' => 1,
                    )
                )
            )
        )
    )
)

```

```
        'expireDays' => 5,
        'useUniqueCache_conf' => array(
            'spaceCharacter' => '-'
        )
    ),
),
    'plaintext' => array(
        'type' => 'single',
        'keyValues' => array(
            'type' => 99
        )
    ),
),
),
    'pagePath' => array (
        'type' => 'user',
        'userFunc' => 'EXT:realurl/class.tx_realurl_advanced.php:&tx_realurl_adva
        'spaceCharacter' => '_',
        'languageGetVar' => 'L',
        'expireDays' => 3,
        'rootpage_id' => 1,
        'segTitleFieldList' => 'tx_realurl_pathsegment,alias,nav_title,title',
    ),
),
);
```

5.4 Tutorial

5.4.1 Category-navigation

Use the following TypoScript-Code in order to get a category navigation:

```
lib.tx_commerce.navigation = HMENU
lib.tx_commerce.navigation {
    special = userfunction
    special.userFunc = user_tx_commerce_catmenu_pub->init
    special {
        // category for first level
        category = {$plugin.tx_commerce_lib.catUid}
        // show products
        showProducts = 0

        // comma-separated list of fields, which shall additionally be processed in the menu
        additionalFields = teaser, teaserimages
        // PID fuer die Anzeige der Seite
        overridePid = {$plugin.tx_commerce_lib.overridePid}
        // EntryLevel
        entryLevel = 0

        // list of categories, where the manufacturer will be shown with the category in the menu
        // menu
        displayMenuForCat <

        // add parameter depth an dpath to URL for a even faster rendering of the navigation
        useRootlineInformationToUrl = {$plugin.tx_commerce_lib.useRootlineInformationToUrl}
        // menu level
```

```
        1 = TMENU
        2 = TMENU
        3 = TMENU
    }
    1 = TMENU
    1.itemArrayProcFunc = user_tx_commerce_catmenu_pub->clear
    2 = TMENU
    2.itemArrayProcFunc = user_tx_commerce_catmenu_pub->clear
    3 = TMENU
    3.itemArrayProcFunc = user_tx_commerce_catmenu_pub->clear
}
```


DEVELOPMENT

6.1 Payment API

Related to issue: <http://forge.typo3.org/issues/4980>

The ‘new’ payment API gives programmers a basic API to select and create different payment methods under different circumstances.

The basic structure is like this: - There are multiple payment types (eg. credit card and prepayment). - A specific payment type can have multiple criteria to determine if this payment type is available in the specific environment (for example if prepayment is available in the customers country). If all criteria return true, the payment type is available and can be selected by the customer. - A payment type can have multiple payment provider, there can be for example multiple credit card provider. The first available provider will be chosen. - Like a payment type, a provider can have multiple criteria, if all criteria return true for a specific provider, the provider is selected as payment provider of the chosen payment type.

See the default `$GLOBALS['TYPO3_CONF_VARS']['EXTCONF']['COMMERCE_EXTKEY']['SYSPRODUCTS']['PAYMENT']` in `ext_localconf.php` for example configuration and settings.

6.2 Hooks

This document describes all the hooks in the typo3 extension commerce (version 0.8.17) as of 01/04/2007. The hooks are sorted by the classes and functions they can be found in. Each hook has a type which is either Single or Multiple. Single hook classes are written directly to the `$GLOBALS['TYPO3_CONF_VARS']['EXTCONF']['commerce/{namespacepart/classname}']['$hookname']` variable as there may never be more than one hook-object. For hooks with the type “multiple” `$GLOBALS['TYPO3_CONF_VARS']['EXTCONF']['commerce/{namespacepart/classname}']['$hookname']` is an array which can contain multiple hook objects. Each of them is called in the order they are found in the array by foreach. Some hook-methods can return data which is used by the calling function. Others may return some values but they are ignored. For those I used the return type void as known from the C programming language.

6.2.1 Single object hooks

```
$GLOBALS['TYPO3_CONF_VARS']['EXTCONF']['commerce/' . $classname][$hookname]
```

Controller/BaseController

Hookname	Call	Parameter / Result	Type
formatAttributeValue	formatAttributeValue	\$matrixKey \$attributeUid \$attributeValue \$result \$parentController Result	Integer Integer String String \CommerceTeam\Commerce\Controller\ String

Controller/BasketController

Hookname	Call	Parameter / Result	Type
generateBasket	makeDelivery	\$parentController \$basket \$result Result	\CommerceTeam\Commerce\Controller\ \CommerceTeam\Commerce\Domain\Mo String
	makePayment	\$parentController \$basket \$result Result	String \CommerceTeam\Commerce\Controller\ \CommerceTeam\Commerce\Domain\Mo String String
alternativePrefixId	singleDisplayPrefixId	Result	String

Domain/Model/Article

Hookname	Call	Parameter / Result	Type
getActualPriceforScaleUid	getActualPriceforScaleUid	\$count \$parentModel Result	Integer Integer
getSpecialPrice	specialPrice	\$specialPrice \$priceUids Result	Array Array void
calculateDeliveryCosts	calculateDeliveryCostNet calculateDeliveryCostGross	\$deliveryCostNet \$parentModel Result \$deliveryCostGross \$parentModel Result	Integer void Integer void

Domain/Repository/ArticleRepository

Hookname	Call	Parameter / Result	Type
getPrices	priceOrder additionalPriceWhere	\$orderField Result \$parentRepository \$uid Result	String String Integer String

Domain/Repository/CategoryRepository

Hookname	Call	Parameter / Result	Type
getChildCategories	categoryOrder	\$orderField \$parentRepository Result	String \\CommerceTeam\\Commerce\\Domain\\Re
	categoryQueryPostHook	\$data \$parentRepository Result	String Array \\CommerceTeam\\Commerce\\Domain\\Re Array
getChildProducts	productOrder	\$orderField \$parentRepository Result	String \\CommerceTeam\\Commerce\\Domain\\Re
	productQueryPreHook	\$queryArray \$parentRepository Result	String Array \\CommerceTeam\\Commerce\\Domain\\Re
	productQueryPostHook	\$data \$parentRepository Result	Array Array \\CommerceTeam\\Commerce\\Domain\\Re Array

Domain/Repository/ProductRepository

Hookname	Call	Parameter / Result	Type
getArticles	articleOrder	\$orderField Result	String String
	additionalWhere	\$where Result	String String

Hook/OrdermailHooks

Hookname	Call	Parameter / Result	Type
generateMail	processMarker	\$markerArray \$parentHook Result	Array \\CommerceTeam\\Commerce\\Hook\\Orde Array

Utility/ArticleCreatorUtility

Hookname	Call	Parameter / Result	Type
createArticle	preinsert	\$articleData Result	Array void

ViewHelpers/Navigation

Hookname	Call	Parameter / Result	Type
makeArrayPostRender	sortingOrder	\$sorting \$uidRoot \$mainTable \$tableMm \$mDepth \$path \$parentViewHelper Result	String Integer String String Integer Integer \CommerceTeam\Commerce\ViewHelper String
makeSubChildArrayPostRender	sortingOrder	\$sorting \$categoryUid \$mainTable \$mmTable \$mDepth \$path \$parentViewHelper	String Integer String String Integer Integer \CommerceTeam\Commerce\ViewHelper

6.2.2 Multiple object hooks

```
$GLOBALS['TYPO3_CONF_VARS']['EXTCONF']['commerce'][$classname][$hookname][]
```


Controller/AddressesController

Hookname	Call	Parameter / Result	Type
getListing	processAddressMarker processListingMarker	\$itemMarkerArray \$address \$piArray \$parentController Result \$baseMarkerArray \$linkMarkerArray \$addressItems \$addressType \$piArray \$parentController Result	Array Array Array \CommerceTeam\Commerce\Controller\ Array Array Array String Integer Array \CommerceTeam\Commerce\Controller\ void
getAddressForm	processAddressfieldsMarkerArray processAddressFormMarker	\$fieldsMarkerArray \$fieldTemplate \$addressData \$action \$addressUid \$config \$parentController Result \$baseMarkerArray \$action \$addressUid \$addressData \$config \$parentController Result	Array String Array String Integer Array \CommerceTeam\Commerce\Controller\ Array Array String Integer Array Array \CommerceTeam\Commerce\Controller\ void
deleteAddress	deleteAddress	\$addressid \$parentController Result	Integer \CommerceTeam\Commerce\Controller\ String
getInputField	postGetInputField	\$content \$fieldName \$fieldConfig \$fieldValue \$parentController Result	String String Array Mixed \CommerceTeam\Commerce\Controller\ String
6.2. Hooks checkAddressForm	validationMethod_*	\$parentController \$fieldName	49 \CommerceTeam\Commerce\Controller\ String

Controller/BaseController

Hookname	Call	Parameter / Result	Type
addAdditionalLocallang	loadAdditionalLocallang	\$parentController Result	\CommerceTeam\Commerce\Controller\ void
makeListView	additionalMarker	\$markerArray \$parentController Result	Array \CommerceTeam\Commerce\Controller\ Array
getArticleMarker	additionalMarkerArticle	\$markerArray \$article \$parentController Result	Array \CommerceTeam\Commerce\Domain\M \CommerceTeam\Commerce\Controller\ Array
makeBasketView	postBasketView	\$content \$articleTypes \$lineTemplate \$template \$basket \$parentController Result	String Array String String \CommerceTeam\Commerce\Domain\M \CommerceTeam\Commerce\Controller\ Array
makeBasketInformation	processMarkerTaxInformation processMarkerBasketInformation	\$taxRateTemplate \$basket \$parentController Result \$markerArray \$basket \$parentController Result	String \CommerceTeam\Commerce\Domain\M \CommerceTeam\Commerce\Controller\ String Array \CommerceTeam\Commerce\Domain\M \CommerceTeam\Commerce\Controller\ Array
makeLineView	processMarkerLineView	\$markerArray \$basketItem \$parentController Result	Array \CommerceTeam\Commerce\Domain\M \CommerceTeam\Commerce\Controller\ Array
6.2. Hooks			Array 51
renderValue	postRenderValue	\$output	String

Controller/BasketController

Hookname	Call	Parameter / Result	Type
main	postInit additionalMarker	\$parentController Result \$markerArray \$parentController Result	\CommerceTeam\Commerce\Controllerv String Boolean Array \CommerceTeam\Commerce\Controllerv Array
handleDeleteBasket	postdelBasket	\$basket \$parentController Result	\CommerceTeam\Commerce\Domain\M \CommerceTeam\Commerce\Controllerv void
handleAddArticle	preartAddUid postDeleteArtUidSingle preartAddUidSingle postartAddUidSingle postartAddUid	\$basket \$parentController Result \$articleUid \$articleAddValues \$soldCountValue \$basket \$parentController Result \$articleUid \$articleAddValues \$product \$article \$basket \$parentController Result \$articleUid \$articleAddValues \$product \$article \$basket \$parentController Result \$basket \$parentController Result	\CommerceTeam\Commerce\Domain\M \CommerceTeam\Commerce\Controllerv void Integer Array Integer \CommerceTeam\Commerce\Domain\M \CommerceTeam\Commerce\Controllerv void Integer Array \CommerceTeam\Commerce\Domain\M \CommerceTeam\Commerce\Domain\M \CommerceTeam\Commerce\Domain\M \CommerceTeam\Commerce\Controllerv void Integer Array \CommerceTeam\Commerce\Domain\M \CommerceTeam\Commerce\Domain\M \CommerceTeam\Commerce\Domain\M \CommerceTeam\Commerce\Controllerv void
6.2. Hooks			53 \CommerceTeam\Commerce\Domain\M \CommerceTeam\Commerce\Controllerv void

Controller/CheckoutController

Hookname	Call	Parameter / Result	Type
init	checkoutSteps	\$checkoutSteps \$parentController Result	Array \CommerceTeam\Commerce\Controller\ void
main	processData preSwitch {*} postSwitch postRender	\$parentController Result \$currentStep \$parentController Result \$parentController Result \$currentStep \$content \$parentController Result \$currentStep \$content \$parentController Result	 void Array void void String String String String String String String
getBillingAddress	processMarker	\$markerArray \$parentController Result	Array \CommerceTeam\Commerce\Controller\ Array
getDeliveryAddress	processMarker	\$markerArray \$parentController Result	Array \CommerceTeam\Commerce\Controller\ Array
handlePayment	alternativePaymentStep processMarker	\$payment \$parentController Result \$markerArray \$parentController Result	 \CommerceTeam\Commerce\Payment\Pa \CommerceTeam\Commerce\Controller\ String Array Array
6.2. Hooks	processMarker	\$markerArray \$parentController Result	55 Array \CommerceTeam\Commerce\Controller\

Controller/InvoiceController

Hookname	Call	Parameter / Result	Type
main	additionalMarker	\$markerArray \$subpartArray \$parentController Result	Array Array \CommerceTeam\Commerce\Controller\ Array

Controller/ListController

Hookname	Call	Parameter / Result	Type
init	preInit	\$parentController Result	\CommerceTeam\Commerce\Controller\ void
	postInit	\$parentController Result	\CommerceTeam\Commerce\Controller\ void
renderSingleView	preRenderSingleView	\$product \$category \$subpartName \$subpartNameNostock \$parentController Result	\CommerceTeam\Commerce\Domain\M \CommerceTeam\Commerce\Domain\M String String
	additionalMarker	\$markerArray \$parentController Result	\CommerceTeam\Commerce\Controller\ String Array \CommerceTeam\Commerce\Controller\ Array
makeArticleView	additionalMarker	\$markerArray \$parentController \$article Result	Array \CommerceTeam\Commerce\Controller\ \CommerceTeam\Commerce\Domain\M
	additionalAttributeMarker	\$markerArrayItem \$parentController \$uid Result	Array Array \CommerceTeam\Commerce\Controller\ Integer Array Array \CommerceTeam\Commerce\Domain\M \CommerceTeam\Commerce\Controller\ Array
	additionalMarkerMakeArticleView	\$product \$parentController Result	

Domain/Model/Article

Hookname	Call	Parameter / Result	Type
init	postinit	\$parentModel Result	\CommerceTeam\Commerce\Domain\M void

Domain/Model/ArticlePrice

Hookname	Call	Parameter / Result	Type
init	postinit	\$parentModel Result	\CommerceTeam\Commerce\Domain\M void
getPriceNet	postpricenet	\$parentModel Result	\CommerceTeam\Commerce\Domain\M void
getPriceGross	postpricegross	\$parentModel Result	\CommerceTeam\Commerce\Domain\M void

Domain/Model/Attribute

Hookname	Call	Parameter / Result	Type
init	postinit	\$parentModel Result	\CommerceTeam\Commerce\Domain\M void

Domain/Model/AttributeValue

Hookname	Call	Parameter / Result	Type
init	postinit	\$parentModel Result	\CommerceTeam\Commerce\Domain\M void

Domain/Model/Basket

Hookname	Call	Parameter / Result	Type
loadDataFromDatabase	loadDataFromDatabase	\$returnData \$parentModel Result	Array \\CommerceTeam\\Commerce\\Domain\\Model\\Basket\\BasketModel.cs void
loadPersistentDataFromDatabase	loadPersistentDataFromDatabase	\$returnData \$parentModel Result	Array \\CommerceTeam\\Commerce\\Domain\\Model\\Basket\\BasketModel.cs void
storeDataToDatabase	storeDataToDatabase	\$oneItem \$insertData Result	\\CommerceTeam\\Commerce\\Domain\\Model\\Basket\\BasketModel.cs Array Array

Domain/Model/Category

Hookname	Call	Parameter / Result	Type
init	postinit	\$parentModel Result	\\CommerceTeam\\Commerce\\Domain\\Model\\Category\\CategoryModel.cs void

Domain/Model/Product

Hookname	Call	Parameter / Result	Type
init	postinit	\$parentModel Result	\\CommerceTeam\\Commerce\\Domain\\Model\\Product\\ProductModel.cs void

Hook/DatamapHooks

Hookname	Call	Parameter / Result	Type
preProcessOrder	moveOrdersPreMoveOrder	\$order \$incomingFieldArray Result	Array Array void
	moveOrdersPostMoveOrder	\$order \$incomingFieldArray Result	Array Array void

Hook/OrdermailHooks

Hookname	Call	Parameter / Result	Type
ordermoveSendMail	postOrdermoveSendMail	\$mailconf \$orderdata \$template Result	Array Array String void

Tree/Leaf/Data

Hookname	Call	Parameter / Result	Type
loadRecords	addExtendedFields	\$itemTable \$extendedFields Result	String String String

Tree/Leaf/MasterData

Hookname	Call	Parameter / Result	Type
getRecordsByMountpoints	getRecordsByMountpointsPreLoadRecords	\$positions \$parentLeaf Result	Array \CommerceTeam\Commerce\Tree\Leaf\M
	getRecordsByMountpointsPostProcessRecords	\$records \$parentLeaf Result	void Array \CommerceTeam\Commerce\Tree\Leaf\M void

Utility/BackendUtility

Hookname	Call	Parameter / Result	Type
copyProduct	beforeCopy	\$uid \$uidLast \$overrideArray Result	Integer Integer Array void
	afterCopy	\$newUid \$uid \$overrideArray Result	Integer Integer Array void
copyCategory	beforeCopy	\$uid \$uidLast \$overrideArray Result	Integer Integer Array void
	afterCopy	\$newUid \$uid \$overrideArray Result	Integer Integer Array void
overwriteProduct	beforeOverwrite	\$uidFrom \$uidTo \$datamap Result	Integer Integer Array void
	beforeCopy	\$uidFrom \$uidTo \$datamap \$tce Result	Integer Integer Array void

Utility/DataHandlerUtility

Hookname	Call	Parameter / Result	Type
showCopyWizard	beforeFormClose	\$uidClip	Integer
		\$uidTarget	Integer
		\$command	String
	beforeFormClose	\$userIgnoreClose	Boolean
		Result	String
		\$uidClip	Integer
	beforeTransform	\$uidTarget	Integer
		\$command	String
		\$userIgnoreClose	Boolean
commitCommand	beforeCommit	\$uidClip	Integer
		\$uidTarget	Integer
		\$command	String
	afterCommit	Result	void
		\$uidClip	Integer
		\$uidTarget	Integer
		\$command	String
		Result	void

Utility/GeneralUtility

Hookname	Call	Parameter / Result	Type
generateSessionKey	postGenerateSessionKey	\$key	String
		Result	String
sendMail	preProcessMail	\$mailconf	Array
		\$additionalData	Array
		Result	void
	ownMailRendering	\$mailconf	Array
		\$additionalData	Array
		\$hooks	Array
	postProcessMail	Result	Boolean
		\$message	String
		\$mailconf	Array
		\$additionalData	Array
		Result	String

ViewHelpers/TreelibTceforms

Hookname	Call	Parameter / Result	Type
processItemArrayForBrowseableTreeDefault	processItemArrayForBrowseableTreeDefault	 \$value \$table \$uid Result	 String String Integer String

COMMERCE CODING GUIDELINES

7.1 CGL

The TYPO3 Commerce defines the latest Version of the TYPO3 Core CGL to be used for TYPO3 Commerce. All new code must be created by the CGL, patches should also be fix CGL issues.

7.2 Deprecated log

TYPO3 Commerce will use the TYPO3 deprecated log for old methods inside commerce. Deprecated methods will be deleted from the code not earlier than one year after adding to the log.

7.3 Versions

TYPO3 Commerce will use the “strict version” scheme. Bugfixes and security issues will be added at least to the current TRUNK version and the latest old stable version. Features will only be added to the TRUNK version. All versions will have a tag inside the SVN.

7.4 Maintainance of the code

7.4.1 Write Access

All members of the TYPO3 Commerce project have write access to the SVN

7.4.2 TYPO3 Core Mailinglist

A project has been set up at forge <https://forge.typo3.org/projects/extension-commerce/>. All patches should be posted in addition with RFC or RFI to the slack channel <https://typo3.slack.com/messages/commerce/>, everyone is allowed to post to this list!

7.4.3 Approval

All patches must have at least a “+1 on testing” for approval to commerce. Additional “+1 on reading” are welcome! A commit to the SVN will be made by a commerce team member.

7.4.4 FYI96

Small bugfixes, additional hooks could be announced by a FYI96. If no protest on the channel will be posted this issue can be committed to the code.

7.4.5 Nobrainer

Nobrainer, comments in the code, CGL changes can be made as direct commit to the code without any posting

7.4.6 Twitter

The official hash-tag for twitter-posts ist #t3commerce

DEPRECATION

8.1 Deprecatcd sind 4.x

The tx_graytree_foldername field should not be used anymore. Instead the field tx_commerce_foldername was integrated. Until June 2016 an update helps to migrate the values.

Developers are adviced to not use this field anymore.

8.2 Removed methods since 4.x

Class	Method	Replacement
CommerceTeamCommerceControllerBaseController	addTypoLinkToTS	addTypoLinkToTypoScript
	makeControl	Method got removed from the api. Method was not used in pibase context
	makeproductAttributList	Method got removed from the api
	makeArticleAttributList	Method got removed from the api
	makeSingle-View	Method got removed from the api

Class	Method	Replacement
CommerceTeamCommerceDomainModelAbstractEntity	getMark-erArray	please use CommerceTeamCommerceControllerBaseController->renderElement in combination with CommerceTeamCommerceDomainModelAbstractEntity::returnAssocArray instead
	get_uid	getUid
	get_LOCALIZED_UID	getLocalizedUid
	get_lang	getLang
	re- turn_assoc_array	returnAssocArray
	add_field_to_fieldlist	addFieldToFieldlist
	add_fields_to_fieldlist	addFieldsToFieldlist
	is_valid_uid	isValidUid
	get_attributes	getAttributes
	load_data	loadData

Class	Method	Replacement
CommerceTeamCommerceDomainModelArticle	get_title	getTitle
	get_subtitle	getSubtitle
	get_classname	getClassname
	get_description_extra	getDescriptionExtra
	get_article_price_uid	getPriceUid
	get_price_gross	getPriceGross
	get_price_net	getPriceNet
	getArticlePriceUid	getPriceUid
	getPossiblePriceUids	getPriceUids
	get_tax	getTax
	get_ordernumber	getOrdernumber
	get_parent_product	getParentProduct
	get_article_attributes	getArticleAttributes
	get_article_type_uid	getArticleTypeUid
	load_prices	loadPrices

Class	Method	Replacement
CommerceTeamCommerceDomainModelArticlePrice	get_price_net	getPriceNet
	get_price_gross	getPriceGross

Class	Method	Replacement
CommerceTeamCommerceDomainModelAttribute	get_all_values	getAllValues
	get_values	getValues
	get_value	getValue
	get_title	getTitle
	get_attributes	getAttributes
	get_unit	getUnit

Class	Method	Replacement
CommerceTeamCommerceDomainModelAttribute-Value	getshowicon	Method got removed from the api
	get_attributes	getAttributes
	get_value	getValue

Class	Method	Replacement
CommerceTeamCommerceDomainModelBasicBasket	recalculate_sums	recalculateSums
	delete_all_articles	deleteAllArticles
	delete_article	deleteArticle
	change_quantity	changeQuantity
	get_articles_by_article_type_uid_as_uidlist	getArticlesByArticleType-UidAsUidlist
	get_assoc_arrays	getAssocArrays
	add_article	addArticle
	isReadOnly	getReadOnly
	getIsReadOnly	getReadOnly
	get_gross_sum	getSumGross
	get_net_sum	getSumNet

Class	Method	Replacement
CommerceTeamCommerceDomainModelBasket	loadPersistantDataFrom-Database	loadPersistentDataFrom-Database
	load_data_from_database	loadDataFromDatabase
	store_data_to_database	storeDataToDatabase
	store_data	storeData
	set_session_id	setSessionId
	get_session_id	getSessionId

Class	Method	Replacement
CommerceTeamCommerceDomainModelBasketItem	recalculate_item_sums	recalculateItemSums
	calculate_gross_sum	calculateGrossSum
	calculate_net_sum	calculateNetSum
	change_quantity	changeQuantity
	get_article_uid	getArticleUid
	get_tax	getTax
	get_price_gross	getPriceGross
	get_price_uid	getPriceUid
	get_item_sum_net	getItemSumNet
	get_item_sum_gross	getItemSumGross
	get_item_sum_tax	getItemSumTax
	get_article_assoc_array	getArticleAssocArray
	get_product_assoc_array	getProductAssocArray
	get_array_of_assoc_array	getArrayOfAssocArray
	get_price_net	getPriceNet
	get_article_article_type_uid	getArticleTypeUid
	get_quantity	getQuantity

Class	Method	Replacement
CommerceTeamCommerceDomainModelCategory	isPSet	isPermissionSet
	get_rec_child_categories_uidlist	getChildCategoriesUidlist
	get_categorie_rootline_uidlist	getParentCategoriesUidlist
	ProductsBelowCategory	hasProductsInSubCategories
	get_l18n_categories	getL18nCategories
	getCategoryTSconfig	getTyposcriptConfig
	numOfChildCategories	getChildCategoriesCount
	get_subproducts	getChildProducts
	get_child_products	getChildProducts
	has_subproducts	hasProducts
	has_subcategories	hasSubcategories
	getSubcategories	getChildCategories
	get_child_categories	getChildCategories
	get_category_path	getCategoryPath
	get_keywords	getKeywords
	get_parent_category	getParentCategory
	getAllProducts	getProducts
	load_perms	loadPermissions
	get_navtitle	getNavtitle
	get_description	getDescription
	get_images	getImages
	get_teaser	getTeaser
	get_subtitle	getSubtitle
	get_title	getTitle
	load_data	loadData

Class	Method	Replacement
_CommerceTeamCommerceDomainModel-Product	getRelevantArticles	Method got removed from the api
	get_selectattribute_matrix	getSelectAttributeMatrix
	get_Articles_by_Attribute	getArticlesByAttribute
	get_Articles_by_AttributeArray	getArticlesByAttributeArray
	compareBySorting	Method got removed from the api
	get_l18n_products	getL18nProducts
	getNumberOfArticles	getArticlesCount
	get_teaser	getTeaser
	get_description	getDescription
	get_subtitle	getSubtitle
	get_t3ver_oid	getT3verOid
	get_pid	getPid
	getMasterparentCategorie	getMasterparentCategory
	get_title	getTitle
	get_masterparent_categorie	getMasterparentCategory
	get_parent_categories	getParentCategories
	get_images	getImages
	set_leng_description	Please use typoscript instead
	get_attribute_matrix	getAttributeMatrix
	get_attribute_matrix	getAttributeMatrix
	get_product_attribute_matrix	getAttributeMatrix
	get_product_attribute_matrix	getAttributeMatrix
	getArticles	getArticleUids
	load_articles	loadArticles

Class	Method	Replacement
CommerceTeamCommerceDomainRepositoryArticleRepository	get_parent_product_uid	getParentProductUid
	get_prices	getPrices
	get_attributes	getAttributes

Class	Method	Replacement
CommerceTeamCommerceDomainRepositoryAttributeRepository	get_attribute_value_uids	getAttributeValueUids

Class	Method	Replacement
CommerceTeamCommerceDomainRepositoryCategoryRepository	getUid	Method got removed from the api
	getLangUid	Method got removed from the api
	get_parent_category	getParentCategory
	get_parent_categories	getParentCategories
	get_l18n_categories	getL18nCategories
	get_child_categories	getChildCategories
	get_child_products	getChildProducts

Class	Method	Replacement
CommerceTeamCommerceDomainRepositoryFolderRepository	getFolderPidList	Method got removed from the api

Class	Method	Replacement
CommerceTeamCommerceDomainRepositoryProductRepository	get_articles	getArticles
	get_attributes	getAttributes
	get_l18n_products	getL18nProducts
	get_related_product_uids	getRelatedProductUids
	get_parent_category	getMasterParentCategory
	get_parent_categorie	getMasterParentCategory
	get_parent_categories	getParentCategories

Class	Method	Replacement
CommerceTeamCommerceDomainRepositoryRepository	get_attributes	getAttributes

Class	Method	Replacement
CommerceTeamCommerceHookPi4Hooks	deleteAddress	beforeDeleteAddress
	beforeAddressSave	Method got removed from the api
	beforeAddressEdit	Method got removed from the api
	no- tify_addressObserver	afterAddressSave or afterAddressEdit
	checkAddressDelete	Method got removed from the api

Class	Method	Replacement
CommerceTeamCommerceHookTcehooksHandlerHooks	notify_feuserObserver	notifyFeuserObserver
	no- tify_addressObserver	notifyAddressObserver

Class	Method	Replacement
CommerceTeamCommercePaymentPaymentAbstract	getPObj	getParentObject

Class	Method	Replacement
CommerceTeamCommerceUtilityArticleCreatorUtility	createNewPriceCB	Method got removed from the api
	createNewScalePricesCount	Method got removed from the api
	createNewScalePricesSteps	Method got removed from the api
	createNewScalePricesStartAmount	Method got removed from the api
	deletePriceButton	Method got removed from the api

Class	Method	Replacement
CommerceTeamCommerceUtilityBackendUtility	savePriceFlexformWithArticle	Method got removed from the api
	fix_articles_price	updatePriceXMLFromDatabase
	fix_product_atributte	updateXML
	fix_category_atributte	updateXML
	isPSet	isPermissionSet

Class	Method	Replacement
CommerceTeamCommerceUtilityFolderUtility	init_folders	initFolders

Class	Method	Replacement
CommerceTeamCommerceUtilityGeneralUtility	formatPrice	getAttributes

Class	Method	Replacement
CommerceTeamCommerceViewHelpersNavigation	getManuAsCat	getManufacturerAsCategory
	getCategoryRootlineforTS	getCategoryRootlineforTypoScript
	getActiveCats	Method got removed from the api
	CommerceRootline	renderRootline

Class	Method	Replacement
CommerceTeamCommerceViewHelpersOrderEditFunc	article_order_id	articleOrderId
	sum_price_gross_format	sumPriceGrossFormat
	order_articles	orderArticles
	order_status	orderStatus
	invoice_adress	invoiceAddress
	delivery_adress	deliveryAddress
	adress	address
	fe_user_orders	feUserOrders

8.3 Removed hooks since 4.x

Class	Method	Hook	Replacement
CommerceTeamCommerceControllerAddressesController	getList-ing	['commerce/pi4/class.tx_commerce_pi4.php'] [hook:commerce2]	['commerce/Controller/AddressesController'] ['getListing']
		['commerce/Classes/Controller/AddressesController.php'] ['getListing']	['commerce/Controller/AddressesController'] ['getListing']
	getAddress-Form	['commerce/pi4/class.tx_commerce_pi4.php'] [hook:commerce2]	['commerce/Controller/AddressesController'] ['getAddressForm']
		['commerce/Classes/Controller/AddressesController.php'] ['getAddressFormItem']	['commerce/Controller/AddressesController'] ['getAddressForm']
	deleteAddress	['commerce/pi4/class.tx_commerce_pi4.php'] [hook:commerce2]	['commerce/Controller/AddressesController'] ['deleteAddress']
		['commerce/Classes/Controller/AddressesController.php'] ['deleteAddress']	['commerce/Controller/AddressesController'] ['deleteAddress']
	checkAddress-Form	['commerce/pi4/class.tx_commerce_pi4.php'] [hook:commerce2]	['commerce/Controller/AddressesController'] ['checkAddressForm']
		['commerce/Classes/Controller/AddressesController.php'] ['checkAddressForm']	['commerce/Controller/AddressesController'] ['checkAddressForm']
	saveAddress-Data	['commerce/pi4/class.tx_commerce_pi4.php'] [hook:commerce2]	['commerce/Controller/AddressesController'] ['saveAddress']
		['commerce/Classes/Controller/AddressesController.php'] ['saveAddress']	['commerce/Controller/AddressesController'] ['saveAddress']
	getAddresses	['commerce/pi4/class.tx_commerce_pi4.php'] [hook:commerce2]	['commerce/Controller/AddressesController'] ['getAddresses']
		['commerce/Classes/Controller/AddressesController.php'] ['getAddresses']	['commerce/Controller/AddressesController'] ['getAddresses']
	get-Input-Field	['commerce/pi2/class.tx_commerce_pi2.php'] [hook:commerce2]	['commerce/Controller/AddressesController'] ['getInputField']

Class	Method	Hook	Replacement
CommerceTeamCommerceControllerBaseController	addAdditionalLocallang	['commerce/lib/class.tx_commerce_pibase.php']['locallang']	['commerce/Controller/BaseController']['addAdditionalLocallang']
		['commerce/Classes/Controller/BaseController.php']['locallang']	['commerce/Controller/BaseController']['addAdditionalLocallang']
	makeListView	['commerce/Classes/Controller/ListView.php']['preRenderListView']	Removed because no methods of this hook were used
		['commerce/lib/class.tx_commerce_pibase.php']['listview']	['commerce/Controller/BaseController']['makeListView']
		['commerce/Classes/Controller/BaseController.php']['listView']	['commerce/Controller/BaseController']['makeListView']
	getArticleMarker	['commerce/lib/class.tx_commerce_pibase.php']['articleMarker']	['commerce/Controller/BaseController']['getArticleMarker']
		['commerce/Classes/Controller/BaseController.php']['articleMarker']	['commerce/Controller/BaseController']['getArticleMarker']
	makeBasketView	['commerce/lib/class.tx_commerce_pibase.php']['makeBasketView']	['commerce/Controller/BaseController']['makeBasketView']
		['commerce/Classes/Controller/BaseController.php']['makeBasketView']	['commerce/Controller/BaseController']['makeBasketView']
	makeBasketInformation	['commerce/lib/class.tx_commerce_pibase.php']['makeBasketInformation']	['commerce/Controller/BaseController']['makeBasketInformation']
		['commerce/Classes/Controller/BaseController.php']['makeBasketInformation']	['commerce/Controller/BaseController']['makeBasketInformation']
	makeLineView	['commerce/lib/class.tx_commerce_pibase.php']['makeLineView']	['commerce/Controller/BaseController']['makeLineView']
		['commerce/Classes/Controller/BaseController.php']['makeLineView']	['commerce/Controller/BaseController']['makeLineView']
	renderValue	['commerce/lib/class.tx_commerce_pibase.php']['renderValue']	['commerce/Controller/BaseController']['renderValue']
		['commerce/Classes/Controller/BaseController.php']['renderValue']	['commerce/Controller/BaseController']['renderValue']
	renderElement	['commerce/lib/class.tx_commerce_pibase.php']['renderElement']	['commerce/Controller/BaseController']['renderElement']
		['commerce/Classes/Controller/BaseController.php']['renderElement']	['commerce/Controller/BaseController']['renderElement']
	formatAttributeValue	['commerce/lib/class.tx_commerce_pibase.php']['formatAttributeValue']	['commerce/Controller/BaseController']['formatAttributeValue']
		['commerce/Classes/Controller/BaseController.php']['formatAttributeValue']	['commerce/Controller/BaseController']['formatAttributeValue']
	renderProductsForList	['commerce/lib/class.tx_commerce_pibase.php']['renderProductsForList']	['commerce/Controller/BaseController']['renderProductsForList']
		['commerce/Classes/Controller/BaseController.php']['renderProductsForList']	['commerce/Controller/BaseController']['renderProductsForList']

Class	Method	Hook	Replacement
CommerceTeam-CommerceControllerBasketController	main	['commerce/pi2/class.tx_commerce_pi2.php'] ['main']	['commerce/Controller/BasketController.php'] ['main']
		['commerce/Classes/Controller/BasketController.php'] ['main']	['commerce/Controller/BasketController.php'] ['main']
	handleDeleteBasket	['commerce/pi2/class.tx_commerce_pi2.php'] ['postDeleteBasket']	['commerce/Controller/BasketController.php'] ['postDeleteBasket']
		['commerce/Classes/Controller/BasketController.php'] ['postDeleteBasket']	['commerce/Controller/BasketController.php'] ['postDeleteBasket']
	handleAddArticle	['commerce/pi2/class.tx_commerce_pi2.php'] ['handleAddArticle']	['commerce/Controller/BasketController.php'] ['handleAddArticle']
		['commerce/Classes/Controller/BasketController.php'] ['addArticleUid']	['commerce/Controller/BasketController.php'] ['handleAddArticle']
	handlePaymentArticle	['commerce/pi2/class.tx_commerce_pi2.php'] ['postPaymentArticle']	['commerce/Controller/BasketController.php'] ['postPaymentArticle']
		['commerce/Classes/Controller/BasketController.php'] ['postPaymentArticle']	['commerce/Controller/BasketController.php'] ['handlePaymentArticle']
	handleDeliveryArticle	['commerce/pi2/class.tx_commerce_pi2.php'] ['postDeliveryArticle']	['commerce/Controller/BasketController.php'] ['postDeliveryArticle']
		['commerce/Classes/Controller/BasketController.php'] ['postDeliveryArticle']	['commerce/Controller/BasketController.php'] ['handleDeliveryArticle']
	getQuickView	['commerce/pi2/class.tx_commerce_pi2.php'] ['getQuickView']	['commerce/Controller/BasketController.php'] ['getQuickView']
		['commerce/Classes/Controller/BasketController.php'] ['getQuickView']	['commerce/Controller/BasketController.php'] ['getQuickView']
	generateBasket	['commerce/pi2/class.tx_commerce_pi2.php'] ['generateBasket']	['commerce/Controller/BasketController.php'] ['generateBasket']
		['commerce/Classes/Controller/BasketController.php'] ['generateBasket']	['commerce/Controller/BasketController.php'] ['generateBasket']
		['commerce/pi2/class.tx_commerce_pi2.php'] ['generateBasketMarker']	['commerce/Controller/BasketController.php'] ['generateBasketMarker']
		['commerce/Classes/Controller/BasketController.php'] ['generateBasketMarker']	['commerce/Controller/BasketController.php'] ['generateBasketMarker']
	makeDelivery	['commerce/pi2/class.tx_commerce_pi2.php'] ['makeDeliveryArticles']	['commerce/Controller/BasketController.php'] ['makeDeliveryArticles']
		['commerce/Classes/Controller/BasketController.php'] ['deliveryArticles']	['commerce/Controller/BasketController.php'] ['makeDelivery']
	makePayment	['commerce/pi2/class.tx_commerce_pi2.php'] ['makePaymentArticles']	['commerce/Controller/BasketController.php'] ['makePaymentArticles']
		['commerce/Classes/Controller/BasketController.php'] ['paymentArticles']	['commerce/Controller/BasketController.php'] ['makePayment']

Class	Method	Hook
CommerceTeamCommerceControllerCheckoutController	validateAddress	['commerce/pi3/class.tx_commerce_pi3.ph
		['commerce/Classes/Controller/CheckoutC
		['commerce/pi3/class.tx_commerce_pi3.ph
		['commerce/Classes/Controller/CheckoutC
	getInstanceOfTceMain	['commerce/pi3/class.tx_commerce_pi3.ph
		['commerce/Classes/Controller/CheckoutC
	saveOrder	['commerce/pi3/class.tx_commerce_pi3.ph
		['commerce/Classes/Controller/CheckoutC
	generateMail	['commerce/pi3/class.tx_commerce_pi3.ph
		['commerce/Classes/Controller/CheckoutC
	init	['commerce/pi3/class.tx_commerce_pi3.ph
		['commerce/Classes/Controller/CheckoutC
	main	['commerce/pi3/class.tx_commerce_pi3.ph
		['commerce/Classes/Controller/CheckoutC
	getBillingAddress	['commerce/pi3/class.tx_commerce_pi3.ph
		['commerce/Classes/Controller/CheckoutC
	getDeliveryAddress	['commerce/pi3/class.tx_commerce_pi3.ph
		['commerce/Classes/Controller/CheckoutC
	handlePayment	['commerce/pi3/class.tx_commerce_pi3.ph
		['commerce/Classes/Controller/CheckoutC
	getListing	['commerce/pi3/class.tx_commerce_pi3.ph
		['commerce/Classes/Controller/CheckoutC
	finishIt	['commerce/pi3/class.tx_commerce_pi3.ph
		['commerce/Classes/Controller/CheckoutC
	getOrderId	['commerce/pi3/class.tx_commerce_pi3.ph
		['commerce/Classes/Controller/CheckoutC
	getBasketSum	['commerce/pi3/class.tx_commerce_pi3.ph
		['commerce/Classes/Controller/CheckoutC
	getInputForm	['commerce/pi3/class.tx_commerce_pi3.ph
		['commerce/Classes/Controller/CheckoutC
	handleAddress	['commerce/pi3/class.tx_commerce_pi3.ph
		['commerce/Classes/Controller/CheckoutC
	canMakeCheckout	['commerce/pi3/class.tx_commerce_pi3.ph
		['commerce/Classes/Controller/CheckoutC
	sendUserMail	['commerce/pi3/class.tx_commerce_pi3.ph
		['commerce/Classes/Controller/CheckoutC
	sendAdminMail	['commerce/pi3/class.tx_commerce_pi3.ph
		['commerce/Classes/Controller/CheckoutC

Class	Method	Hook	Replacement
CommerceTeamCommerce-ControllerInvoiceController	main	['commerce/pi6/class.tx_commerce_pi6.php'] ['invoiceController'] ['main']	['commerce/Controller/InvoiceController'] ['main']
		['commerce/Classes/Controller/InvoiceController.php'] ['invoice']	['commerce/Controller/InvoiceController'] ['main']

Class	Method	Hook	Replacement
CommerceTeamCommerceControllerListController	init	['commerce/pi1/class.tx_commerce_pi1.php'] ['commerce/pi1/class.tx_commerce_pi1.php'] ['commerce/pi1/class.tx_commerce_pi1.php']	['commerce/Controller/ListController'] ['commerce/Controller/ListController'] ['commerce/Controller/ListController']
		['commerce/Classes/Controller/ListController.php'] ['init']	['commerce/Controller/ListController'] ['init']
		['commerce/pi1/class.tx_commerce_pi1.php'] ['commerce/pi1/class.tx_commerce_pi1.php'] ['commerce/pi1/class.tx_commerce_pi1.php']	['commerce/Controller/ListController'] ['commerce/Controller/ListController'] ['commerce/Controller/ListController']
		['commerce/Classes/Controller/ListController.php'] ['postInit']	['commerce/Controller/ListController'] ['init']
	renderSingleView	['commerce/lib/class.tx_commerce_pibase.php'] ['singleView']	['commerce/Controller/ListController'] ['renderSingleView']
		['commerce/Classes/Controller/ListController.php'] ['renderSingleView']	['commerce/Controller/ListController'] ['renderSingleView']
	makeArticleView	['commerce/lib/class.tx_commerce_pibase.php'] ['articleView']	['commerce/Controller/ListController'] ['makeArticleView']
		['commerce/Classes/Controller/ListController.php'] ['articleView']	['commerce/Controller/ListController'] ['makeArticleView']

Class	Method	Hook	Replacement
CommerceTeamCommerceDomainModelArticle	init	['commerce/lib/class.tx_commerce_article.php'] ['postinit']	['commerce/Domain/Model/Article'] ['init']
		['commerce/Classes/Domain/Model/Article.php'] ['postinit']	['commerce/Domain/Model/Article'] ['init']
	getActualPriceforScaleUid	['commerce/lib/class.tx_commerce_article.php'] ['getActualPriceforScaleUid']	['commerce/Domain/Model/Article'] ['getActualPriceforScaleUid']
		['commerce/Classes/Domain/Model/Article.php'] ['getActualPriceforScaleUid']	['commerce/Domain/Model/Article'] ['getActualPriceforScaleUid']
	getSpecialPrice	['commerce/lib/class.tx_commerce_article.php'] ['specialPrice']	['commerce/Domain/Model/Article'] ['getSpecialPrice']
		['commerce/Classes/Domain/Model/Article.php'] ['specialPrice']	['commerce/Domain/Model/Article'] ['getSpecialPrice']
	calculateDeliveryCosts	['commerce/lib/class.tx_commerce_article.php'] ['calculateDeliveryCosts']	['commerce/Domain/Model/Article'] ['calculateDeliveryCosts']
		['commerce/Classes/Domain/Model/Article.php'] ['calculateDeliveryCosts']	['commerce/Domain/Model/Article'] ['calculateDeliveryCosts']

Class	Method	Hook	Replacement
CommerceTeamCommerceDomainModelArticlePrice	init	['commerce/lib/-class.tx_commerce_article_price.php'] ['postinit']	['commerce/Domain/Model/ArticlePrice.php'] ['init']
		['commerce/Classes/Domain/Model/ArticlePrice.php'] ['postinit']	['commerce/Domain/Model/ArticlePrice.php'] ['init']
	get-PriceNet	['commerce/lib/-class.tx_commerce_article_price.php'] ['postinit']	['commerce/Domain/Model/ArticlePrice.php'] ['getPriceNet']
		['commerce/Classes/Domain/Model/ArticlePrice.php'] ['postPriceNet']	['commerce/Domain/Model/ArticlePrice.php'] ['getPriceNet']
	get-PriceGross	['commerce/lib/-class.tx_commerce_article_price.php'] ['postinit']	['commerce/Domain/Model/ArticlePrice.php'] ['getPriceGross']
		['commerce/Classes/Domain/Model/ArticlePrice.php'] ['postPriceGross']	['commerce/Domain/Model/ArticlePrice.php'] ['getPriceGross']

Class	Method	Hook	Replacement
CommerceTeamCommerceDomainModelAttribute	init	['commerce/lib/-class.tx_commerce_attribute.php'] ['postinit']	['commerce/Domain/Model/Attribute.php'] ['init']
		['commerce/Classes/Domain/Model/Attribute.php'] ['postinit']	['commerce/Domain/Model/Attribute.php'] ['init']

Class	Method	Hook	Replacement
CommerceTeamCommerceDomainModelAttributeValue	init	['commerce/lib/-class.tx_commerce_attribute_value.php'] ['postinit']	['commerce/Domain/Model/AttributeValue.php'] ['init']
		['commerce/Classes/Domain/Model/AttributeValue.php'] ['postinit']	['commerce/Domain/Model/AttributeValue.php'] ['init']

Class	Method	Hook	Replacement
CommerceTeamCommerceDomainModelBasket	load-DataFrom-Database	['commerce/lib/-class.tx_commerce_basket.php'] ['loadDataFromDatabase']	['commerce/Domain/Model/Basket.php'] ['loadDataFromDatabase']
		['commerce/Classes/Domain/Model/Basket.php'] ['loadDataFromDatabase']	['commerce/Domain/Model/Basket.php'] ['loadDataFromDatabase']
	storeDataTo-Database	['commerce/lib/-class.tx_commerce_basket.php'] ['storeDataToDatabase']	['commerce/Domain/Model/Basket.php'] ['storeDataToDatabase']
		['commerce/Classes/Domain/Model/Basket.php'] ['storeDataToDatabase']	['commerce/Domain/Model/Basket.php'] ['storeDataToDatabase']
	loadPersistent-DataFrom-Database	['commerce/Classes/Domain/Model/Basket.php'] ['loadPersistentDataFromDatabase']	['commerce/Domain/Model/Basket.php'] ['loadPersistentDataFromDatabase']

Class	Method	Hook	Replacement
CommerceTeamCommerceDomainModelCategory	init	['commerce/lib/-class.tx_commerce_category.php'] ['postinit']	['commerce/Domain/Model/Category.php'] ['init']
		['commerce/Classes/Domain/Model/Category.php'] ['postinit']	['commerce/Domain/Model/Category.php'] ['init']

Class	Method	Hook	Replacement
CommerceTeamCommerce-DomainModelProduct	init	['commerce/lib/-class.tx_commerce_product.php']['postinit']	['commerce/Domain/Model/Product']['init']
		['commerce/Classes/Domain/Model/Product.php']['postinit']	['commerce/Domain/Model/Product']['init']

Class	Method	Hook	Replacement
CommerceTeamCommerce-DomainRepositoryArticleRepository	getPrices	['commerce/lib/-class.tx_commerce_article.php']['priceOrder']	['commerce/Domain/Repository/ArticleRepository']['getPrices']
		['commerce/Classes/Domain/Repository/ArticleRepository.php']['priceOrder']	['commerce/Domain/Repository/ArticleRepository']['getPrices']
		['commerce/lib/-class.tx_commerce_article.php']['additionalPriceWhere']	['commerce/Domain/Repository/ArticleRepository']['getPrices']
		['commerce/Classes/Domain/Repository/ArticleRepository.php']['additionalPriceWhere']	['commerce/Domain/Repository/ArticleRepository']['getPrices']

Class	Method	Hook	Replacement
CommerceTeamCommerceDomainRepositoryCategoryRepository	getChildCategories	['commerce/lib/-class.tx_commerce_db_category.php']['categoryQueryPreHook']	['commerce/Domain/Repository/CategoryRepository']['getChildCategories']
		['commerce/Classes/Domain/Repository/CategoryRepository.php']['categoryOrder']	['commerce/Domain/Repository/CategoryRepository']['getChildCategories']
		['commerce/lib/-class.tx_commerce_db_category.php']['categoryQueryPostHook']	['commerce/Domain/Repository/CategoryRepository']['getChildCategories']
		['commerce/Classes/Domain/Repository/CategoryRepository.php']['categoryQueryPostHook']	['commerce/Domain/Repository/CategoryRepository']['getChildCategories']
	getChildProducts	['commerce/lib/-class.tx_commerce_db_category.php']['productQueryPreHook']	['commerce/Domain/Repository/CategoryRepository']['getChildProducts']
		['commerce/Classes/Domain/Repository/CategoryRepository.php']['productOrder']	['commerce/Domain/Repository/CategoryRepository']['getChildProducts']
		['commerce/lib/-class.tx_commerce_db_category.php']['productQueryPostHook']	['commerce/Domain/Repository/CategoryRepository']['getChildProducts']
		['commerce/Classes/Domain/Repository/CategoryRepository.php']['productQueryPostHook']	['commerce/Domain/Repository/CategoryRepository']['getChildProducts']
		['commerce/lib/-class.tx_commerce_db_category.php']['productQueryPreHook']	['commerce/Domain/Repository/CategoryRepository']['getChildProducts']
		['commerce/Classes/Domain/Repository/CategoryRepository.php']['productQueryPostHook']	['commerce/Domain/Repository/CategoryRepository']['getChildProducts']

Class	Method	Hook	Replacement
CommerceTeamCommerceDomainRepositoryProductRepository	getArticles	['commerce/lib/-class.tx_commerce_product.php']['articleOrder']	['commerce/Domain/Repository/ProductRepository']['getArticles']
		['commerce/Classes/Domain/Repository/ProductRepository.php']['articleOrder']	['commerce/Domain/Repository/ProductRepository']['getArticles']
		['commerce/lib/-class.tx_commerce_product.php']['additionalWhere']	['commerce/Domain/Repository/ProductRepository']['getArticles']
		['commerce/lib/-class.tx_commerce_product.php']['additionalWhere']	['commerce/Domain/Repository/ProductRepository']['getArticles']
		['commerce/Classes/Domain/Repository/ProductRepository.php']['additionalWhere']	['commerce/Domain/Repository/ProductRepository']['getArticles']

Class	Method	Hook	Replacement
CommerceTeamCommerceHookDataMapHooks	preProcessOrder	['commerce/Classes/Hook/-class.tx_commerce_dmhooks.php']['moveOrders']	['commerce/Hook/-DataMapHooks']['preProcessOrder']
		['commerce/Classes/Hook/-DataMapHooks.php']['moveOrders']	['commerce/Hook/-DataMapHooks']['preProcessOrder'] -method to call changed from moveOrders_preMoveOrder to moveOrdersPostMoveOrder

Class	Method	Hook	Replacement
CommerceTeamCommerceHookOrdermailHooks	ordermoveSendMail	['commerce/Classes/Hook/-class.tx_commerce_ordermailhooks.php']['ordermoveSendMail']	['commerce/Hook/OrdermailHooks']['ordermoveSendMail']
		['commerce/Classes/Hook/OrdermailHooks.php']['ordermoveSendMail']	['commerce/Hook/OrdermailHooks']['ordermoveSendMail']
	generateMail	['commerce_ordermails/mod1/class.tx_commerce_ordermail.php']['generateMail']	['commerce/Hook/OrdermailHooks']['generateMail']
		['commerce/Classes/Hook/OrdermailHooks.php']['generateMail']	['commerce/Hook/OrdermailHooks']['generateMail']

Class	Method	Hook	Replacement
CommerceTeamCommerceTreeLeafData	loadRecords	['commerce/tree/-class.leafData.php']['loadRecords']	['commerce/Tree/LeafData']['loadRecords']
		['commerce/Classes/Tree/LeafData.php']['loadRecords']	['commerce/Tree/LeafData']['loadRecords']

Class	Method	Hook	Replacement
CommerceTeamCommerceUtilityArticleCreatorUtility	createArticle	['commerce/-class.tx_commerce_articlecreator.php']['createArticlePreInsert']	['commerce/Utility/ArticleCreatorUtility']['createArticle']
		['commerce/Classes/Utility/ArticleCreatorUtility.php']['createArticlePreInsert']	['commerce/Utility/ArticleCreatorUtility']['createArticle']

Class	Method	Hook	Replacement
CommerceTeamCommerceUtilityBackendUtility	copyProduct	['commerce/lib/-class.tx_commerce_belib.php']['copyProductClass']	['commerce/Utility/BackendUtility']['copyProduct']
		['commerce/Classes/Utility/BackendUtility.php']['copyProduct']	['commerce/Utility/BackendUtility']['copyProduct']
	copyCategory	['commerce/lib/-class.tx_commerce_belib.php']['copyCategoryClass']	['commerce/Utility/BackendUtility']['copyCategory']
		['commerce/Classes/Utility/BackendUtility.php']['copyCategory']	['commerce/Utility/BackendUtility']['copyCategory']
	overwriteProduct	['commerce/lib/-class.tx_commerce_belib.php']['overwriteProductClass']	['commerce/Utility/BackendUtility']['overwriteProduct']
		['commerce/Classes/Utility/BackendUtility.php']['overwriteProduct']	['commerce/Utility/BackendUtility']['overwriteProduct']

Class	Method	Hook	Replacement
CommerceTeamCommerceUtilityDataHandlerUtility	showCopyWizard	['commerce/-mod_cce/class.tx_commerce_cce_db.php']['showCopyWizardClass']	['commerce/Utility/DataHandlerUtility']['showCopyWizard']
		['commerce/Classes/Utility/DataHandlerUtility.php']['showCopyWizard']	['commerce/Utility/DataHandlerUtility']['showCopyWizard']
	commitCommand	['commerce/-mod_cce/class.tx_commerce_cce_db.php']['commitCommandClass']	['commerce/Utility/DataHandlerUtility']['commitCommand']
		['commerce/Classes/Utility/DataHandlerUtility.php']['commitCommand']	['commerce/Utility/DataHandlerUtility']['commitCommand']

Class	Method	Hook	Replacement
CommerceTeamCommerceUtilityFolderUtility	initFolders	['commerce/-class.tx_commerce_create_folder.php']['postcreateFolder']	Method got removed from the api
		['commerce/-class.tx_commerce_create_folder.php']['precreateFolder']	Method got removed from the api
		['commerce/-class.tx_commerce_create_folder.php']['postcreateFolder']	Method got removed from the api
		['commerce/-class.tx_commerce_create_folder.php']['postcreateFolder']	Method got removed from the api

Class	Method	Hook	Replacement
CommerceTeamCommerceUtilityGeneralUtility	generateSessionKey	['commerce/lib/-class.tx_commerce_div.php']['generateSessionKey']	['commerce/Utility/GeneralUtility']['generateSessionKey']
		['commerce/Classes/Utility/GeneralUtility.php']['generateSessionKey']	['commerce/Utility/GeneralUtility']['generateSessionKey']
	sendMail	['commerce/lib/-class.tx_commerce_div.php']['sendMail']	['commerce/Utility/GeneralUtility']['sendMail']
		['commerce/Classes/Utility/GeneralUtility.php']['sendMail']	['commerce/Utility/GeneralUtility']['sendMail']

Class	Method	Hook	Replacement
CommerceTeamCommerceViewHelpersNavigation	makeArrayPostRender	['commerce/lib/class.tx_commerce_db_navigation.php']	['commerce/ViewHelpers/Navigation']
		['commerce/Classes/ViewHelpers/Navigation.php']	['commerce/ViewHelpers/Navigation']
	makeSubChildArrayPostRender	['commerce/lib/class.tx_commerce_db_navigation.php']	['commerce/ViewHelpers/Navigation']
		['commerce/Classes/ViewHelpers/Navigation.php']	['commerce/ViewHelpers/Navigation']

Class	Method	Hook	Replacement
CommerceTeamCommerceViewHelpersTreelibTceforms	processItemArrayForBrowseableTreeDefault	['commerce/treelib/class.tx_commerce_treelib_tceforms.php']	['commerce/ViewHelpers/TreelibTceforms']
		['commerce/Classes/ViewHelpers/TreelibTceforms.php']	['commerce/ViewHelpers/TreelibTceforms']

Class	Method	Hook	Replacement
CommerceTeamCommerceTreeLeafMasterData	getRecordsByMountpoints	\$GLOBALS['TYPO3_CONF_VARS']['SC_OPTIONS']['typo3/CONF/Extbase/Classes/TreeLeafMasterData.php']	\$GLOBALS['TYPO3_CONF_VARS']['SC_OPTIONS']['typo3/CONF/Extbase/Classes/TreeLeafMasterData.php']

BREAKING CHANGES

9.1 Remove of iframe rendering the category tree

As the div rendering mode is stable the iframe rendering gets removed. Its not needed anymore and is discouraged due to reduce the entry points in total.

Removed file `Classes/ViewHelpers/IframeTreeBrowser.php`

Removed properties `CommerceTeamCommerceViewHelpersTreelibTceforms::iframeContentRendering`

Removed methods `CommerceTeamCommerceViewHelpersTreelibTceforms::setIframeContentRendering`
`CommerceTeamCommerceViewHelpersTreelibTceforms::isIframeContentRendering` `CommerceTeamCommerceViewHelpersTreelibTceforms::isIframeRendering` `CommerceTeamCommerceViewHelpersTreelibTceforms::getTreeContent` `CommerceTeamCommerceViewHelpersTreelibTceforms::setIframeTreeBrowserScript`
`CommerceTeamCommerceViewHelpersTreelibTceforms::renderIframe` `CommerceTeamCommerceViewHelpersTreelibTceforms::getIframeParameter`

SUPPORT

You will get support concerning this extension in the newsgroup: <news://news.typo3.org/typo3.projects.commerce> .

KNOWN PROBLEMS

Related Products: Currently, only 3 products will be shown, even though it is possible to assign a product to as many other products as desired. Additionally, sorting is only done by entry date of the products.

The category tree in product listing plugin does not work if TYPO3 is installed in a subdirectory and not in webroot. You can select categories but you cannot not expand and fold the tree here. As a workaround you can expand the tree in the Commerce>Categories module before opening the plugin. See <http://forge.typo3.org/issues/9120>

CHANGELOG