

EXT: Shop System

Extension Key: **tt_products**

Copyright 2000-2005, Franz Holzinger, <kontakt@fholzinger.com>

This document is published under the Open Content License
available from <http://www.opencontent.org/opl.shtml>

The content of this document is related to TYPO3
- a GNU/GPL CMS/Framework available from www.typo3.com

Table of Contents

EXT: Shop System	1	Automatic creation of frontend users and address fields.....	7
Introduction	1	Product properties:.....	7
What does it do?.....	1	Configuration	9
Screenshot.....	2	FAQ.....	9
Users manual	3	Files.....	9
Note:.....	3	Description.....	9
Mini-Basket.....	3	Reference.....	10
Calculation script:.....	3	class.tx_ttproducts.php properties.....	10
Discount:.....	3	Payment and shipping configuration.....	16
Search link for the products of the last X days.....	3	Pricecalc, discountprice and creditpoints configuration.....	18
Offers and highlights.....	4	Inherited from zk_products 1.3.2	19
AGB General trading conditions.....	4	payment_DIBS.php.....	20
MEMO page	4	Tutorial	21
Adminstration	5	Known problems	24
Installation.....	5	To-Do list	24
Handling of Categories.....	5	Changelog	24
Important.....	5		
Template markers.....	5		

Введение


Возможности экстеншена?




Typos3 shop extension обеспечивает следующие возможности:

- Отображения списка продуктов со множеством картинок и деталями
- Продуктовую корзину
- Страницу оплаты
- Слежение за статусами заказов
- Автоматическое создание счета и списка доставки
- Различная такса для ордера, отгрузки и оплаты
- Управление ценными бумагами (basic stock management)
- Отправка CSV администратору магазина для каждого ордера (2 формата файлов на выбор)
- E-Mail-Attachments для письма-подтверждения (для примера AGB в Германии = General trading conditions)
- Возможность выбрать варианты продукта (цвет, размер, gradings и сопутствующих товаров по умолчанию)

- Вынуждать клиента соглашаться с General trading conditions (AGB) при помощи checkbox
- Предложения, подсветка и новые добавленные продукты
- Специальная подготовка, вес и груз (может использоваться для вычисления оплаты доставки)
- Способность ограничивать методы оплаты определенными пользовательскими группами
- Автоматическое создание frontend пользователя при первом заказе
- Вспоминание продуктов через мемо, когда пользователь залогинивается снова
- Дисконтные проценты для разных пользователей
- Несколько методов вычисления цены с уступкой для торговых посредников
- Отображение заказов: заказ может быть показан на основе fe-user basis CODE=ORDERS
- Система Creditpoint: клиенты могут накапливать пункты кредита при каждом заказе. Накопленные пункты дадут им скидку на новых заказах или же определенные продукты могут быть "куплены" в счет этих пунктов.
- Оправдательная система (Voucher system): если новый пользователь указывает, что он регистрируется при содействии другого уже зарегистрированного пользователя, то последний получает определенные пункты кредита в качестве бонуса. Новый же клиент может получить скидку на первый заказ.
- Подарочные сертификаты (Gift certificate): Пользователь может купить сертификат и послать его своему другу в качестве подарка. После регистрации друга с помощью этого сертификата он получает на свой счет кредит.

Screenshot

 [NÆSTE SIDE](#)

Produkt:	Angiv antal:
Canon Ixus II  Ultra kompakt APS kamera i titanium. Fuldautomatisk og 2x zoom. Vejl. Pris : 2295,- Web-pris: 1.995,00 Kr. (ex. moms: 1.596,00)	Antal: <input type="text"/> <input type="button" value="Læg i kurv"/>
Canon Ixus X-1  APS kamera til extreme forhold. Kan dykkes med ned til 5 meter. Leveres med film, batteri, rem og dansk manual. Vejl. 1.995,- Web-pris: 1.490,00 Kr. (ex. moms: 1.192,00)	Antal: <input type="text"/> <input type="button" value="Læg i kurv"/>
Canon Ixus Z-65  APS kamera med 2,5x zoom, Tænd, sigt og skyd. Leveres med film, rem, taske og dansk manual. Vejl. 2.295,- Web-pris: 1.750,00 Kr. (ex. moms: 1.400,00)	Antal: <input type="text"/> <input type="button" value="Læg i kurv"/>
Canon Ixus Z-70  APS kamera med 3x zoom, udført i metal. Leveres med film, batteri, rem og dansk manual. Vejl. 3.295,- Web-pris: 2.450,00 Kr. (ex. moms: 1.960,00)	Antal: <input type="text"/> <input type="button" value="Læg i kurv"/>

Пользовательская документация

Замечание:

Вы должны заполнить полдуктовое поле 'In Stock (pcs)' ("В запасе") среди свойств продукта, иначе не произойдет отображение продуктов в списке во frontend в случае равенства этого поля 0. Это было изменено к старой версии. Вы будете хранить цены с включенной таксой в базе данных. Это поможет вам создать цены такими, какими будет видеть их пользователь. Эти цены будут иметь вид вроде "89.99" и "не 89.72".

Перед вводом цен на продукты вы должны решить, будете ли вы вводить цены с- или без таксы (смотрите TAXincluded). Все подсчеты и конфигурирование цены будут осуществляться с использованием сохраненных в таблице tt_products цен.

Вы можете использовать шаблоны (template) с множеством колонок (displayBasketColumns=1 или выше) даже если вы имеете одну колонку в списке (list table).

Используйте в качестве примера шаблон из папки tt_products/template, когда вы в первый раз пытаетесь создать магазин. Вы должны изменить id страницы в линке (после 'id=...') на id страницы корзины вашего сайта.

Когда вы хотите пользоваться множеством цветов, размеров или gradings, вы должны изменить ####FIELD_NAME### на ####FIELD_NAME_BASKET### в BASKET_TEMPLATE

Мини-Корзина (Mini-Basket):

Кроме "normal" корзины, в которой показывается законченный список выбранный продуктов и некоторый опции заказа, имеется также и мини-корзина ("mini-basket"), в которой показывается только число продуктов в корзине и общая цена.

Для использования мини-корзины, вставьте новый контент-элемент "Plugin: Products" и напишите в поле CODE-field: OVERVIEW.

Используйте следующие метки в tt_products шаблоне для создания и определения этой мини-корзины:

Code Listing:

```
<!-- ###BASKET_OVERVIEW_TEMPLATE### begin -->
<div class="shop_minibasket">
  <div class="shop_minibasket_image"><!--###LINK_BASKET###-->###IMAGE_BASKET###<!--
###LINK_BASKET###--></div>
  <p>###NUMBER_GOODSTOTAL### Products (EUR ###PRICE_GOODSTOTAL_TAX###.-)</p>
<!-- ###BASKET_OVERVIEW_TEMPLATE### end -->

<!-- ###BASKET_OVERVIEW_EMPTY### begin -->
<p>No products in basket.</p>
<!-- ###BASKET_OVERVIEW_EMPTY### end -->
</div>
<!-- ###BASKET_OVERVIEW_TEMPLATE### end -->
```

Скрипт подсчета (Calculation script):

Если вы хотите написать и использовать ваш собственный скрипт подсчета цены, вы должны всегда использовать только priceTax переменную. Переменная priceNoTax была удалена в August 2008, потому что она всегда может быть подсчитана из переменной priceTax и константы TAXpercentage.

Вы можете использовать глобальные PHP переменные \$this->calculatedArray и \$this->itemArray для самостоятельного вычисления цены.

Скидки (Discount):

Имеется несколько методов получения скидок.

1. Использование поля "discount" для frontend пользователя и введение процента скидки.
2. Использование поля "price2" для специальных feuser групп или других вещей. Эти условия задаются в TypoScript с помощью IF-условия.
3. Использование подсчета величины скидки (discountprice), когда скидка зависит от полного числа продуктов, которые имеют специальную цену. Это возможно для frontend группы пользователей 'discountGroupName'.

Поисковая ссылка для продуктов за последние X дней

Это практически страница "What is new?". Создайте простую ссылку на страницу SEARCH и добавьте "newitemdays" как параметр.

<http://www.mytypo3shop.com/?id=8&newitemdays=10>

Поисковый результат будет содержать все продукты за последние 7 дней.

Вы должны использовать plugin code LISTNEWITEMS. Вы должны также установить константу "newItemDays".

Предложения и подсветка (Offers and highlights)

Вы можете использовать plugin code LISTOFFERS для продуктов, помеченных как предложения ('offer') и код LISTHIGHLIGHTS для новых продуктов.

Генеральное соглашение сторон (AGB General trading conditions)

Вы имеете возможность использовать генеральное торговое соглашение перед тем, как начать реализовывать заказ и продолжить процесс. Страница может быть определена с "PIDagb". AGBs будет отображена на этой странице. Вы можете указать параметр "target" для ссылки при помощи "agbtarget".

Документ (например, PDF файл) на сервере, содержащий AGB, может быть установлен как "AGBattachment". Так же пользователем может быть указан почтовый ящик, на который будет отправлено приаттаченное к заказу соглашение. Администратор магазина не будет получать такой файл. (Он может получить CSV файл с данными заказа вместо этого, если хочет).

Страница памяти (MEMO page)

Вы можете добавлять продукты на страницу "памяти", когда пользователь заходит вновь. Эта страница памяти будет сохраняться в fe_user таблице.

Просто создайте новую страницу с Shop plugin как элементом контента и введите код "MEMO". Вы должны установить "starting point" также, как и для SEARCH страницы.

Администрирование

Если вы хотите сохранить совместимость в будущем, вы должны установить или оставить пустой

```
$TYPO3_CONF_VARS['EXTCONF']['tt_products']['pageAsCategory'] = 0
```

Для 95% совместимости с zk_products расширением вы должны установить эту переменную в 1. Но лучше не делайте этого, если возможно.

Активируйте сейчас flexforms в backend. Поле кода будет заменено графическим пользовательским интерфейсом. Однако, вы должны ввести заново все значения полей CODE.

```
$TYPO3_CONF_VARS['EXTCONF']['tt_products']['useFlexforms'] = 1
```

Инсталляция

Инсталлируйте экстеншен с помощью менеджера расширений. Если вы уже используете старую версию tt_products, которая инсталлирована глобально (путь typo3/ext/), рекомендуется инсталлировать новое расширение локально "local" (папка typo3conf/ext/) без переписывания старой версии. Делая так, вы легко сможете переключиться на прежнюю версию.

Деинсталируйте экстеншен из Extension Manager. Затем закачайте версию из Online Repository при помощи Extension Manager.

Затем используйте Extension Manager's "Available extensions to install" и добавьте Shop system снова.

Этот extension работает совместно со следующими расширениями: feuser_admin, sr_static_info, sr_feuser_register, rehtmlarea, mkl_currxrate, conf_userts и feuserextrafields.

Обработка категорий (Handling of Categories)

Имеется много способов создать магазин. Обычный путь состоит в создании sysfolders для продуктов и списка и простых сайтов, где продукты отображаются single. Если вы хотите создать большой магазин TYPO3, это становится немного проблематичным.

Если вы хотите создать большой магазин с сотнями артиклов, вам нужно будет скорее всего использовать коммерческое расширение вместо tt_products.

Пример tt_products/ext_localconf.php:

```
$TYPO3_CONF_VARS['EXTCONF']['tt_products']['pageAsCategory'] = 0;
```

Важно

Текущая версия полностью совместима вниз с tt_products 1.2.7. Но необходимо сделать некоторую адаптацию с шаблонным файлом. В `###BASKET_TEMPLATE###` вы должны создать строку `<input type="hidden" name="mode_update" value="1">` перед вашей версией `<input type="Submit" name="products_update" value="update basket">`

Это совместимо с zk_products 1.3.2 Но может быть использовано совместно..

Файлы шаблонов (Template files)

В ваших файлах шаблонов для магазина вам необходимо сделать следующие установки для работы:

```
<input type="hidden" name="mode_update" value="1">
```

```
<input type="Submit" name="products_update" value="update basket">
```

Впишите эти строки в ваши FORM-attributes в BASKET_TEMPLATE.

Метки шаблонов

Следующие метки шаблонов используются в tpl-файле. Вам необходимо добавлять `###` перед и после них. Имеется больше меток, поэтому для полной картины смотрите в пример файла шаблона.

area markers:

Marker:	Description:	Area:
BASKET_TEMPLATE		top
BASKET_INFO_TEMPLATE	Это обычно используется для того, чтобы позволить пользователю ввести адресную информацию отдельно от реальной корзины. Точно такая же фишка, как и BASKET_TEMPLATE.	top

Marker:	Description:	Area:
BASKET_ORDERCONFIRMATION_TEMPLATE	Последняя страница после оформления заказа. Не будет использовано, если установлена PIDthanks смотрите: BASKET_ORDERTHANKS_TEMPLATE	
BASKET_ORDERCONFIRMATION_NOSAVE_TEMPLATE	Эти HTML данные будут добавлены на экран после того, как заказ сохранится в базе данных.	
BASKET_ORDERTHANKS_TEMPLATE	Используется для отображения страницы благодарности, если PIDthanks установлена смотрите BASKET_ORDERCONFIRMATION_TEMPLATE	
BASKET_OVERVIEW_EMPTY	Сообщение 'Ваша корзина пуста' для OVERVIEW	
BASKET_OVERVIEW_TEMPLATE		
BASKET_PAYMENT_TEMPLATE		
BASKET_REQUIRED_INFO_MISSING		
BASKET_TEMPLATE_EMPTY	message 'your basket is empty'	
BASKET_TEMPLATE_NOT_LOGGED_IN	error message that the user has not logged in	
BASKET_TEMPLATE_INVALID_GIFT_UNIQUE_ID	error message that a wrong unique id for a gift product has been entered	
BILL_TEMPLATE	how your bill file will look like	
DELIVERY_TEMPLATE	for the file of your delivery sheet	
EMAIL_PLAINTEXT_TEMPLATE	the email notification text	
EMAIL_NEWUSER_TEMPLATE	email after creation of new frontend use	
ITEM_LIST_TEMPLATE	the listing of products on the starting LIST page	
ITEM_LIST_GIFTS_TEMPLATE	listing of the gift products	
ITEM_SEARCH		
ITEM_SEARCH_EMPTY		
ITEM_SINGLE_DISPLAY		
ITEM_SINGLE_DISPLAY_GIFT	single display used when this is a gift product	
ITEM_SINGLE_DISPLAY_RECORDINSERT	see displayCurrentRecord: render the \$cObj->data	
MEMO_TEMPLATE		
MEMO_NOT_LOGGED_IN	error message that use has not been logged in for MEMO	
ORDERS_LIST_TEMPLATE	display of the order list	
TRACKING_EMAIL_GIFT_NOTIFY_TEMPLATE	notification to the gift recipient in the order tracking	
TRACKING_ENTER_NUMBER		
TRACKING_WRONG_NUMBER		

single markers

Marker:	Type:	Description:	Area:
BROWSE_LINKS		for browsing the display list over several pages	
DELIVERYCOSTS	value	sum of delivery costs and payment costs	

Marker:	Type:	Description:	Area:
DELIVERY_... NAME, ADDRESS, TELEPHONE, FAX, EMAIL, COMPANY, CITY, ZIP, STATE, COUNTRY	value	address information fields have to be connected with a prefix PERSON_... the customer of the order DELIVERY_... the recipient of the order	BASKET_INFO_TEMPLATE
EXTERNAL_COBJECT	value	extra preprocessing Cobject	
FIELD_NAME	value in form	the name of a field in a form	
FIELD_NAME_BASKET	value in form	the basket data in encrypted format	
GC1, GC2, GC3	value	global colors	all
GW1B, GW2B, GW1E, GW2E	wrap	global wraps	all
LINK_BASKET	wrap	link to the basket page	basket
LINK_DATASHEET		link to the datasheet file in uploads/tx_tproducts/datasheet	
ORDER_STATUS_TIME, ORDER_STATUS, ORDER_STATUS_INFO, ORDER_STATUS_COM MENT	value	order values	TRACKING_DISPLAY_INFO
PERSON_...	value	See DELIVERY_...	
PID_TRACKING	value	the tracking pid	
STATUS_CODE_60	wrap	used to allow the disappearance of the text with status code by the shop	TRACKING_DISPLAY_INFO
STATUS_OPTIONS	value	Select menu of state options	ADMIN_CONTROL inside TRACKING_DISPLAY_INFO

Автоматическое создание frontend пользователя и поля адреса

Возможно создание frontend пользователя автоматически после каждого заказа. Вы должны установить "createUsers" в "1", ввести PID sysfolder как PIDuserFolder в поле Setup. Затем установите memberOfGroup в ID вашей frontend user group.

Пользователи получают письмо с сформированным аккаунтом после первого заказа. Это письмо содержит имя аккаунта и автоматически сформированный пароль.

Если вы не хотите использовать простое адресное поле, а хотите использовать поле адреса, название улицы и номера дома, и, в дополнение, почтовый индекс, город и страну, тогда вы должны установить это все в вашем файле шаблона. Если вы установите "useStaticInfoCountry=1", тогда небольшое поле таблицы static_info_country в fe_users будет использовано вместо страны. Это используется для создания выпадающего списка стран.

Свойства продукта:

Цвет, Размер и Сортировка

Ввод вариантов продуктов вы можете осуществить разделением через точку с запятой. Так для рубашки с различными цветами и размером введите "red;green;blue" в первом варианте (variant1) и "S;X;XL;XXL" в variant2. Цвет (Variant 1) и Размер (Variant 2): Введите здесь значения с названием и значения, разделенные точкой с запятой ";" как на следующем примере:

M;L;XL;XXL

Изменение шаблона для поддержки этой фичи:

```

<!-- ###display_variant1### -->
###GW2B### Color: ###GW2E### <SELECT style="font-size: 10px"
name="###FIELD_COLOR_NAME###" rows="1">###PRODUCT_COLOR###</SELECT><br>
<!-- ###display_variant1### -->
<!-- ###display_variant2### -->
###GW2B### Size: ###GW2E### <SELECT style="font-size: 10px"
name="###FIELD_SIZE_NAME###" rows="1">###PRODUCT_SIZE###</SELECT><br>
<!-- ###display_variant2### -->

```

Если вы не создаете цвет или размер выбираемыми, вы должны установить selectColor или selectSize в 0 и использовать только `###PRODUCT_COLOR###` и `###PRODUCT_SIZE###` метки.

В шаблоне корзины вы должны установить `###FIELD_NAME_BASKET###` вместо прежней `###FIELD_NAME###`. И вы должны установить useArticles=1 и сделать записи в tt_products_articles таблице.

Вес, объем и специальная подготовка

Каждый продукт может иметь вес (kg) и объем (Yes/No). Общий вес вычисляется и может быть использован в подсчете цены доставки.

Если продукт маркируется как объемный ('bulkily'), тогда предупреждающее сообщение показывается ('bulkilyWarning' в setup). Установкой 'bulkilyAddition' в setup вы можете добавить цену доставки для этого объемного продукта.

Специальная подготовка не имеет своей функции. Только метка (как с BulkilyWarning) будет написана. Вы можете установить здесь линк для страницы с почтовой формой.

Пример:

```
specialPreparation = special preparation is possible! <A  
href="?id=999&tt_products=###PRODUCT_ID###">Order here</A>.
```

Управление запасом продуктов (Basic stock management)

Поле inStock может быть использовано администратором запасов. Если "in stock" установле в "0", то продукт не будет виден для пользователя.

После каждого реализованного заказа число продуктов будет уменьшать на "складе" ("in stock").

Если новый продукт создан, то его количество будет установлено в 1 по умолчанию. Если вы установите 'AlwaysInStock=1', тогда этот продукт будет всегда виден и доступен пользователю. Иначе число продуктов будет уменьшаться с каждым заказом.

Вы можете определить размерность количества товара сохранением значения в "inStockPieces" как, например, "кусочки".

Несколько налоговых ставок

Каждый продукт имеет свой собственный налог. Но для каждого продукта вы должны ввести его самостоятельно.

Конфигурация

FAQ

- Configuration questions - ToDo

Files

File:	Description:
class.tx_ttproducts.php	Main class used to display the product list or the shopping basket. Call it from a USER cObject with 'userFunc = user_products->main_products'
products_comp_calcScript.inc	Example 'calculationScript'
products_template.tpl products_template_htmlmail.tpl	Example templates in English. '.._htmlmail.tpl' is a HTML-wrap for the HTML-emails sent.
products_template_dk.tpl	Example template in Danish
example_template_bill_de.tpl	bananaguard.de template with examples for bill and delivery sheet in German
products_template_se.tpl	Example template in Swedish
product_detail.tpl product_proefpakketten.tpl shop-a-box.tpl producten.tpl	Example templates for creditpoints and voucher system in Dutch
'handleScripts' for interfacing with external payment gateways	
payment_DIBS.php	Script for interfacing with DIBS (Danish Internet Payment System) in Denmark. You can reach them at http://www.architrade.com/uk/ .
payment_DIBS_template.tpl	Template file for DIBS payment.

[tsref:(cObject).TEST]

Описание

Встроенная покупательская корзина и отображение продуктов работают в пределах TYPO3. Имеется простой интерфейс, который позволяет вам писать свои собственные включения с существующими платежными системами.

Currently there's an implementation with DIBS in Denmark, found at www.architrade.dk .

Reference

class.tx_ttproducts.php properties

Property:	Data type:	Description:	Default:																																						
templateFile	resource	Файл шаблона. Смотрите пример в 'media/scripts/products_template.tpl'																																							
pid_list	string /stdWrap	Pids откуда извлекаются категории, продукты и тп. По дефолту – текущая страница. Допускается множество pid's через запятую!																																							
code	string /stdWrap	<p>Код (CODE), определяющий, что скрипт должен делать. В backend этот пункт заполняется через flexforms вместо старого способа – через поле CODE(capital letters).</p> <p><u>Products:</u></p> <table border="1"> <tr> <td>list (LIST)</td> <td>список продуктов</td> </tr> <tr> <td>list gifts (LISTGIFTS)</td> <td>список подарков</td> </tr> <tr> <td>list highlights (LISTHIGHLIGHTS)</td> <td>список “подствеченных” продуктов</td> </tr> <tr> <td>list offers (LISTOFFERS)</td> <td>список продуктов – спец предложений</td> </tr> <tr> <td>list new items (LISTNEWITEMS)</td> <td>список новых продуктов</td> </tr> <tr> <td>single view (SINGLE)</td> <td>single версия продукта (LIST code может быть использован также) или GET/POST var 'tt_products' может быть установлена.</td> </tr> <tr> <td>search (SEARCH)</td> <td>Отображение search диалога для прод-в</td> </tr> </table> <p><u>Currency:</u></p> <table border="1"> <tr> <td>Selector - CURRENCY</td> <td>Выбор валюты</td> </tr> </table> <p><u>Basket:</u></p> <table border="1"> <tr> <td>content (BASKET)</td> <td>Отображение покупательской корзины. Код 'BASKET' работает, как основа, но пользуется специфическими кодами INFO, PAYMENT и FINALIZE, вы можете соединить функции вместо отображения множества страниц</td> </tr> <tr> <td>overview (OVERVIEW)</td> <td>mini-basket, содержит число и общую стоимость продуктов</td> </tr> <tr> <td>input customer data (INFO)</td> <td>ввод адреса</td> </tr> <tr> <td>input payment (PAYMENT)</td> <td>последнее – проверка и оплата</td> </tr> <tr> <td>finalize order (FINALIZE)</td> <td>Окончание оформление и отправка письма</td> </tr> </table> <p><u>Orders:</u></p> <table border="1"> <tr> <td>tracking (TRACKING)</td> <td>Слежение за статусом заказа, счета и доставки</td> </tr> <tr> <td>bill (BILL)</td> <td>создание файла, содержащего счет</td> </tr> <tr> <td>delivery (DELIVERY)</td> <td>создание файла, содержащего список доставки</td> </tr> <tr> <td>list (ORDERS)</td> <td>список заказов</td> </tr> </table> <p><u>General:</u></p> <table border="1"> <tr> <td>memo (MEMO)</td> <td>memo продуктов frontend пользователей</td> </tr> <tr> <td>help (HELP)</td> <td>как пользоваться tt_products</td> </tr> </table> <p>По возможности не используйте CODE-поля.</p>	list (LIST)	список продуктов	list gifts (LISTGIFTS)	список подарков	list highlights (LISTHIGHLIGHTS)	список “подствеченных” продуктов	list offers (LISTOFFERS)	список продуктов – спец предложений	list new items (LISTNEWITEMS)	список новых продуктов	single view (SINGLE)	single версия продукта (LIST code может быть использован также) или GET/POST var 'tt_products' может быть установлена.	search (SEARCH)	Отображение search диалога для прод-в	Selector - CURRENCY	Выбор валюты	content (BASKET)	Отображение покупательской корзины. Код 'BASKET' работает, как основа, но пользуется специфическими кодами INFO, PAYMENT и FINALIZE, вы можете соединить функции вместо отображения множества страниц	overview (OVERVIEW)	mini-basket, содержит число и общую стоимость продуктов	input customer data (INFO)	ввод адреса	input payment (PAYMENT)	последнее – проверка и оплата	finalize order (FINALIZE)	Окончание оформление и отправка письма	tracking (TRACKING)	Слежение за статусом заказа, счета и доставки	bill (BILL)	создание файла, содержащего счет	delivery (DELIVERY)	создание файла, содержащего список доставки	list (ORDERS)	список заказов	memo (MEMO)	memo продуктов frontend пользователей	help (HELP)	как пользоваться tt_products	HELP
list (LIST)	список продуктов																																								
list gifts (LISTGIFTS)	список подарков																																								
list highlights (LISTHIGHLIGHTS)	список “подствеченных” продуктов																																								
list offers (LISTOFFERS)	список продуктов – спец предложений																																								
list new items (LISTNEWITEMS)	список новых продуктов																																								
single view (SINGLE)	single версия продукта (LIST code может быть использован также) или GET/POST var 'tt_products' может быть установлена.																																								
search (SEARCH)	Отображение search диалога для прод-в																																								
Selector - CURRENCY	Выбор валюты																																								
content (BASKET)	Отображение покупательской корзины. Код 'BASKET' работает, как основа, но пользуется специфическими кодами INFO, PAYMENT и FINALIZE, вы можете соединить функции вместо отображения множества страниц																																								
overview (OVERVIEW)	mini-basket, содержит число и общую стоимость продуктов																																								
input customer data (INFO)	ввод адреса																																								
input payment (PAYMENT)	последнее – проверка и оплата																																								
finalize order (FINALIZE)	Окончание оформление и отправка письма																																								
tracking (TRACKING)	Слежение за статусом заказа, счета и доставки																																								
bill (BILL)	создание файла, содержащего счет																																								
delivery (DELIVERY)	создание файла, содержащего список доставки																																								
list (ORDERS)	список заказов																																								
memo (MEMO)	memo продуктов frontend пользователей																																								
help (HELP)	как пользоваться tt_products																																								
defaultCode	string	Код по умолчанию (смотрите выше), если значение пустое. По умолчанию, значение не установлено и появляется экрано помощи. Вы можете ничего не устанавливать здесь. Example: plugin.tt_products.defaultCode = HELP																																							
defaultProductID	int+	По умолчанию номер uid продукта для простого отображения используется, когда ссылка на скрипт не содержит 'tt_products' параметр. Установите это значение по-умолчанию, когда вы получаете ошибку: “GET/POST var 'tt_products' was missing.”																																							
recursive	int /stdWrap	Номер количества рекурсивного поиска уровней для выбора tt_products в список.																																							

Property:	Data type:	Description:	Default:
domain	string	Url магазина. Если не установлено, будет определено автоматически. Будут переписаны <code>###DOMAIN###</code> маркеры.	
altMainMarkers	(array of strings)	<p>Позволяет определить альтернативные subpart-маркеры для различных вариантов главного шаблона в shopping basket system. Приведен список subparts, которые вы можете изменять:</p> <p>Properties: TRACKING_WRONG_NUMBER TRACKING_ENTER_NUMBER BASKET_REQUIRED_INFO_MISSING BASKET_TEMP ITEM_SINGLE_DISPLAY_RECORDINSERT ITEM_SINGLE_DISPLAY ITEM_SEARCH ITEM_LIST_TEMPLATE ITEM_LIST_GIFTS_TEMPLATE ITEM_SEARCH_EMPTY BASKET_TEMPLATE BASKET_INFO_TEMPLATE BASKET_PAYMENT_TEMPLATE BASKET_ORDERCONFIRMATION_TEMPLATE EMAIL_PLAINTEXT_TEMPLATE BILL_TEMPLATE DELIVERY_TEMPLATE</p> <pre>/+ stdWrap</pre> <p>Example: altMainMarkers.BASKET_TEMPLATE = BASKET_DESIGN2 altMainMarkers.BASKET_TEMPLATE.wrap = ### ###</p> <p>This example changes the main subpart marker for the regular basket display from the default <code>###BASKET_TEMPLATE###</code> to the custom supplied design <code>###BASKET_DESIGN2###</code> (found in the same template HTML-file)</p>	
stdSearchFieldExt	list of fields	Дополнительные поля для поиска По умолчанию, список включает title,note,itemnumber. Любые поля, которые вы определите, будут включены по умолчанию.	
limit	int+	Максимальное количество отображаемых продуктов на одной странице.	50
limitImage	int+	Максимальное количество отображаемых изображений для одного продукта в списке.	1
separateImage	boolean	Обычно, все изображения отображаются вместе. С separateImage=on вы можете использовать <code>###PRODUCT_IMAGE3###</code> для каждого номера изображения отдельно.	
image	IMAGE cObject	Конфигурирование изображения в single display	
listImage	IMAGE cObject	Конфигурирование изображения в list display	
basketImage	IMAGE cObject	Конфигурирование изображения в basket display	
datasheetIcon	IMAGE cObject	Изображение иконки для спецификации. Переписывает <code>###ICON_DATASHEET###</code>	
basketPic	string	URL ссылк для изображения корзины	
noImageAvailable	resource	Изображение публикуется, в случае, если ни одного изображения не приаттчено к продукту. Изображение обрабатывается через IMAGE cObject which is active in the actual display of that image. That is, one of the above IMAGE cObjects.	
displayListCatHeader	boolean	Отображение Category Header in list Если установлено, the category headers (page titles) will automatically be displayed in the product lists. This is not always convenient because you might have chosen a header-title for the "list" content element.	1
displayBasketCatHeader	boolean	Display Category Header in basket. If this option is set, the category headers (page titles) will automatically be displayed in the basket lists.	0

Property:	Data type:	Description:	Default:
displayBasketColumns	int+	<p>Число столбцов в LIST, SEARCH списках в таблице. Вы можете адаптировать ваш шаблон, используя специальные маркеры. ITEM_SINGLE_PRE_HTML и ITEM_SINGLE_POST_HTML должны быть вокруг ваших <TD>-</TD> тэгов, чтобы таблица была создана корректно.</p> <p>Example:</p> <pre><!-- ###ITEM_SINGLE### begin--> ###ITEM_SINGLE_PRE_HTML### <TD height="150" bgcolor="###GC1###" valign="bottom" align="center"> <!--###LINK_ITEM###--> ###GW1B######PRODUCT_TITLE######GW1E### <!--###LINK_ITEM###-->
 ###PRODUCT_IMAGE###
 ###GW1B### item count: ###GW1E### <INPUT size="3" maxlength="4" type="text" name="###FIELD_NAME###" value="###FIELD_QTY###"> <p> </TD> ###ITEM_SINGLE_POST_HTML### <!-- ###ITEM_SINGLE### end --></pre>	
CSSRowEven	string	Cascading Stylesheets (CSS) для строк в products listing.	
CSSRowUneven	string	Тоже, что и выше, для uneven rows.	
CSSListDefault	string	CSS for default entries in a list view.	
CSSListCurrent	string	CSS для текущего выделенного продукта в list view.	
NoSingleViewOnList	boolean	Обычно вы получаете ссылку на полное описание продукта на странице отображения LIST code. Если вы хотите использовать для полного описания другую страницу с SINGLE code – вы должны установить эту переменную в 1.	
itemMarkerArrayFunc	function-name	<p>Каждый раз, когда продукт отображается в корзине, в списке одиночных описаний (list of single), вызывается метод getItemMarkerArray() в user_products. Эта функция заполняет и возвращает массив, так называемый markerArray(), с key/values для замены в шаблоне.</p> <p>Если вы введете правильное значение функции здесь (смотрите datatype 'function-name' для подробностей!), этот массив будет передан в эту функцию как второй параметр. Первым параметром будут идти свойства TypeScript для itemMarkerArrayFunc.</p> <p>Parent PHP-Object reference: .parentObj свойство является жестко запрограммированной ссылкой на объект user_products (PHP).</p> <p>Example: (если эта функция или класс включены!)</p> <pre>itemMarkerArrayFunc = user_addFieldsMarkerArr itemMarkerArrayFunc.simpleOption = 1</pre>	
PIDitemDisplay	int+/Array of integers	<p>PID для single отображения.</p> <p>Если вы хотите использовать определенную страницу для отображения деталей продукта, введите PID (page-uid) здесь. Если вы установите тип = sql, вы можете установить разные условия здесь, используя несколько строк. В строках могут содержаться условия AND. Pid для первого выполненного условия будет возвращен.</p> <pre>PIDitemDisplay { 10.type = sql 10.where = color=red 10.pid = 142 20.type = sql 20.where = color=white 20.pid = 143 }</pre> <p>Если вы установите тип = pid, тогда pid записи будет использоваться.</p> <pre>PIDitemDisplay { 10.type = pid }</pre>	

Property:	Data type:	Description:	Default:
PIDsearch	int+	PID страницы поиска. Если вы хотите, чтобы все продукты можно было найти на специальной странице, введите PID тут! ЗАМЕЧАНИЕ: Если вы установите этот PID, все поисковые запросы будут (должны) быть обработаны с элементом контента с кодом (CODE) "search" на этой странице.	
PIDbasket	int+	PID для страницы корзины. Если вы хотите изменять количество продуктов где-либо – идите на специальную страницу (например. the shopping basket page), введите PID здесь.	
PIDstoreRoot	int+	PID магазина (store root). Этот PID rootPage магазина. Если не установлено – магазин будет оперировать всеми страницами из корня сайта. Но если сайт имеет много страниц – может пострадать производительность.	
PID_sys_products_orders	int+	PID для записей sys_products_orders. По умолчанию, они будут получать pid страницы оплаты (окончания).	
PIDGiftsTable	int+	PID для tt_products_gifts таблицы. Подарочные ордера будут храниться здесь.	
PIDinfo	int+	PID для страницы заполнения имени и адреса.	
PIDfinalize	int+	PID для страницы подтверждения, где пользователь будет соглашаться с правилами перед оплатой.	
PIDthanks	int+	PID для страницы благодарности. BASKET_ORDERTHANKS_TEMPLATE будет использовано.	
PIDtracking	int+	PID для слежения за заказом (order tracking).	
paymentActivity	string	Когда оплачивается, используемый скрипт оплаты будет активизирован. Возможные значения: payment, finalize	finalize
advanceOrderNumber WithInteger	int+ , int+	Если это значение установлено, тогда каждый раз в момент создания нового ордера счетчик заказов увеличивается на случайное число между [first-integer] и [second integer]. Example: 1,10 (This will increment the counter randomly between 1 and 10) 5,5 (This will increment the counter with 5 each time)	
alwaysAdvanceOrderNumber	boolean	Если установлено тогда порядковый номер будет всегда увеличиваться, и пустые порядковые номера не будут многократно использованы. Вы должны установить это, если используете сценарий оплаты, чтобы заплатить через систему оплаты, которая не принимает двойные порядковые номера.	
parseFunc	->parseFunc	Описание продукта обрабатывается с использованием этих свойств. Так, например, если вы хотите позволить создать таблицу HTML-тэгами в "Note field" - вы должны установить parseFunc.allowTags или использовать denyTags. Для совместимости RTE с HTML вы должны установить следующие в корень шаблона страниц. Example: keepNonMatchedTags = 1 RTE.default.proc.preserveTables = 1 Example: parseFunc.allowTags = table, tr, td, b, i, u, a, img, br, div, center, pre, font, hr, sub, sup, p, strong, em, li, ul, ol, blockquote, strike, span, h1, h2, h3, h4, h5, h6 parseFunc.denyTags = *	styles.content.parseFunc
categoryHeader	cObject	Генерирование заголовка категории. Example: categoryHeader = TEXT categoryHeader.current = 1	
TAXpercentage	double	Налог на продажу (НДС) TAX/VAT в процентах. Double значение (!) (подразумевается, "используйте . как разделитель") Example: # Danish sales TAX is 25%: TAXpercentage = 25.00	

Property:	Data type:	Description:	Default:
TAXincluded	boolean	Установите параметр, если налог (TAX) <i>включен</i> в цены в базе данных! (... и, конечно: очистите параметр, если налог не включен в цены в базе данных и должен быть добавлен при отображении продукта) Вся обработка заказа проанализирует этот флаг и вычислит соответствующую цену.	1
priceDec	int+	Десятичные числа цены (Price decimals)	
priceDecPoint	string	Десятичный разделитель цены (Price decimal point)	
priceThousandPoint	string	Разделитель тысяч (Price Thousand point) Введите разделитель тысяч, если такие числа будут на сайте.	
currencySymbol	string	Обозначение денежной единицы. Используется в администрации магазина. Example: EUR DKK USD \$	
lockLoginUserInfo	boolean	Если установлено и пользователь авторизовался, адрес этого пользователя (fe_user) жестко связан для шагов оплаты этого пользователя.	
loginUserInfoAddress	boolean	Если lockLoginUserInfo установлено, то осуществляется переключение на заполнение поля адреса из адреса, страны, индекса и города fe_user.	
requiredInfoFields	<i>list of string</i>	Список полей, которые необходимы для заполнения адресной информации. Этот пример демонстрирует все варианты. Example: requiredInfoFields = name, address, telephone, fax, email, company, city, zip, state, country	
orderBy	string	Список полей, по которым список продуктов будет упорядочиваться. Для работы с экстеншеном Shopsort: pid,category,sorting. Первым параметром сортировки всегда будет pid.	pid,category,title
orderByCategoryTitle	boolean	Если вы хотите, чтобы список выводился по названию (title) из tt_products_cat. Это переопределяет orderBy.	
orderByItemNumberSg	boolean	Если single item отображается – то произойдет сортировка по ItemNumber вместо uid; используется для ###LINK_NEXT_SINGLE### и ###LINK_PREV_SINGLE###	
orderNumberPrefix	string	Префикс для числа заказов. Max 10 символов. Если эта строка начинается с '%', тогда остальные символы интерпретируются как PHP date format.	
orderEmail_from	string	От данного email address рассылаются письма для подтверждения пользователям.	
orderEmail_fromName	string	От данного имени рассылаются письма на подтверждение пользователем.	
orderEmail_to	<i>list of email-addresses</i>	Запятая разделяет список получателей письма заказа. Адреса администрации и супервизоров магазина должны быть здесь!	
orderEmail_htmlmail	boolean / string	Если установлено, письмо подтверждения отправляется как HTML. Если orderEmail_htmlmail.removeImagesWithPrefix установлено, тогда изображения и их HTML тэги не посылаются в письме.	
email_notify_default	boolean	Если email-уведомление по адресу электронной почты доставки пользователя разрешено по умолчанию (он может поменять этосам в модуле слежения позже)	

Property:	Data type:	Description:	Default:
statusCodes	Array of integers	<p>Коды статусов, используемые в модуле слежения за заказом. Числа выше 100 удаляют заказ из листа слежения. Число ноль – это статус незавершенного заказа (и не-завершенные заказы со временем могут быть расценены как мусор...).</p> <p>Числа 50-59 является доступным для выбора клиента. Число 1 зарезервировано для выбора, когда заказ переходит от состояния 0 к состоянию подтверждения - 1. Число 1 не может быть выбрано админом.</p> <p>Это все будет записано в метку ###STATUS_OPTIONS###.</p> <p>Example:</p> <pre>statusCodes { 1 = Order submitted by user 2 = Order is received and accepted by store 10 = Shop is awaiting goods from third-party 11 = Shop is awaiting customer payment 12 = Shop is awaiting material from customer 13 = Order has been payed 20 = Goods shipped to customer 21 = Gift certificates shipped to customer 30 = Other message from store 50 = Customer request for cancelling 51 = Message from customer to shop 60 = Send gift certificate message to receiver 100 = Order shipped and closed 101 = Order closed 200 = Order cancelled }</pre>	
update_code	string	'password' используется администратором магазина для перехода в систему слежения во front end. Парольное поле формы появляется, если BE_USER авторизовался, но этот пароль все же необходим.	password
statusDate_stdWrap	->stdWrap	stdWrap для status date	
		Example: %e-%m-%y %H:%M	
orderDate_stdWrap	->stdWrap	stdWrap for the order date	
		Example: %e-%m-%y	
displayCurrentRecord	boolean	Если установлено, определенные параметры манипулируются для того, чтобы позволить скрипту отрендерить одиночный продукт - \$cObj->data. Если данный параметр установлен, subpart маркеры ###ITEM_SINGLE_DISPLAY_RECORDINSERT### будут использованы вместо regular subpart ###ITEM_SINGLE_DISPLAY### если будут найдены.	
externalProcessing	cObject	Это cObject может быть использован для вызова функции, которая манипулирует корзиной. Эти манипуляции могут быть основаны на внешней системе заказов. Вывод включен в вершину заказа (HTML) на странице-корзине. Этот cObject выполняет каждый раз main_products метод класса user_products в productsLib и он вызывается перед любым главным процессом. Смотрите класс для деталей.	
externalProcessing_final	cObject	cObject заключительный шаблон подтверждения заказа	
externalFinalizing	cObject	Этот cObject может вызывать функцию, которая очищает установки во внешней системе заказов. Это сестринская функция описанной выше функции и они так или иначе должны быть использованы в связке. Эта функция вызывается немедленно после вызова finalize-function. Например, эта функция могла бы быть полезна для очищения любой внешней корзины, облегченную (facilitated) by the .externalProcessing cObject Note: Вывод НЕ включен где-либо.	
wrap1	-> stdWrap	Глобальная Wrap 1. Это будет соединено в маркерах ###GW1B### and ###GW1E### . Не изменяйте вводимое значение, а только вращайте чем-либо.	
		Example: wrap1.wrap = 	
wrap2	-> stdWrap	Глобальная Wrap 2 (смотрите выше) маркер ###GW2B### и ###GW2E###	
selectColor	boolean	Если true цвет продукта выбирается из ниспадающего списка.	1

Property:	Data type:	Description:	Default:
selectSize	boolean	Если true размер продукта выбирается в select box.	1
selectGradings	boolean	Если true gradings продукта выбирается в select box.	1
selectAccessory	boolean	Если true принадлежности к продукту выбирается в check box.	0
color1	string /stdWrap	Значение ###GC1### marker (Global color 1)	
color2	string /stdWrap	Значение ###GC2### marker (Global color 2)	
color3	string /stdWrap	Значение ###GC3### marker (Global color 3)	
accessoryText	string	Текст принадлежности в ###PRICE_ACCESSORY_TEXT###	
useArticles	boolean	Если вы хотите использовать продукты в различных вариантах – вы можете установить это в 1. Запас в таблице articles будет использоваться вместо таблицы products.	
priceTagObj	IMAGE	Картинка для отображения цены.	
usePriceTag	boolean	Если true, priceTagObj будет использовано. Если false, текст будет отображаться.	0
useStaticInfoCountry	boolean	Если true короткое fe_user static_info_country поле будет использовано для страны вместо поля страны.	0
whereGift	string	Это SQL условие “where” определяет, какие продукты будут рассматриваться, как подарки.	
payment / shipping	(see below)	Конфигурация оплаты и методов доставки, их значения и стоимость, и дополнительные скрипты подсчета цены и систем оплаты Смотрите описание ниже!	

[tsref:(script).productsLib]

Конфигурация оплаты и доставки

Оплата и доставка очень подобны в конфигурации и поэтому разделяют одни и те же свойства со специальными замечками, если какое-либо свойство только для одного типа. Конфигурация оплаты и доставки – это всего несколько вопросов для выбора на странице корзины. Это – выбор одного из нескольких способов транспортировки и один из многих способов оплаты. Поэтому вы можете выбрать либо radio-button представление, либо selector box.

Число отобранных методов оплаты и доставки отражено в определенных местах html-шаблона и вы можете также использовать специальные PHP скрипты оплаты. Все это возможно.

Property:	Data type:	Description:	Default:
radio	boolean	Если выбрано – вы получаете radio button layout. Иначе - selector-box.	0
template	string	(Radio layout only) Если .radio true, это строка в 'template' для radio items. По умолчанию (в одну строку!): <pre><noBr>###IMAGE### <input type="radio" name="recs [tt_products]['. \$key. ']' onClick="submit()" value="###VALUE###"###CHECKED###> ###TITLE###</noBr>
</pre>	
wrap	string	(Select layout only) Если .radio - false, эта строка wraps <option> тэг в <select>-тэге! Default is (in one line): <pre>'<select name="recs[tt_products]['. \$key. ']' onChange="submit()" "> </select>'</pre>	
TAXpercentage	double	Налог НДС TAX/VAT в процентах. Double значение (!) (значение, "use . as decimal point") Это заменяет priceNoTax. Это может отличаться от глобального значения налога с тем же именем. Example: <pre># Danish payment TAX is 25%: payment.TAXpercentage = 25.00</pre>	
Array of integers		Конфигурация, смотрите ниже Examples: <pre>TAXpercentage = 12 10.title = Credit card 10.image.file = media/logos/dankort.gif 10.priceTax = 10.percentOfGoodstotal = 0 10.calculationScript = media/scripts/products_comp_calcScript.inc 30.title = By mail 30.image.file = media/logos/postdanmark.gif 30.priceTax = 40</pre>	
Конфигурация payment / shipping пунктов			
title	string	Название, например, "Master card" or "Ground mail"	
image	IMAGE cObject	Logo image для пункта меню	
priceTax	double or array of integers	Цена пункта, включая налог НДС (VAT) (заметьте, вы должны вычислить размер VAT здесь самостоятельно!) Вы используете целые для опеределения минимального числа разделов, для который цены правильна. 6 items и более будут стоить 5.8 в данном примере. Example: <pre>30.priceTax.type = count 30.priceTax.1 = 4 30.priceTax.6 = 5.8</pre>	
priceNoTax	double	Устарело (1 st July 2008) Price of item, excluding VAT Notice: you have to calculate the VAT amount here by yourself!	use TAXpercentage instead
priceFactWeight	double	Доставка ТОЛЬКО: Цена вычисляется исходя из веса всех продуктов. Будет добавлена в priceTax. Вес умножается с этим фактором, чтобы увеличить цену доставки.	

Property:	Data type:	Description:	Default:
priceNoReseller	int+	Цена для посредника. Может быть только 2 варианта в настоящее время. Price2 будет использована только в случае, если ее значение больше 0. Example: [usergroup = 1] priceNoReseller = 2 [global] ...	
getDiscountPrice	boolean	Если установлено – сниженная цена будет использована даже для нормальных клиентов, не входящих, как frontend пользователи, в discount группу.	
percentOfGoodstotal	double	Цена продукта, вычисленная из процента общей стоимости перед оплатой/доставкой.	
percentOfTotalShipping	double	payment ONLY: Если установлена, стоимость оплаты вычисляется в проценте от общей суммы налога на продукт, включая размер налога на стоимость доставки.	
discountGroupName	string	Название группы для discount'a	
calculationScript	resource	PHP скрипт, который включается в “чистую” функцию и должна быть написана для манипулирования количеством во внутреннем массиве. Этот скрипт может использоваться для вычисления специальной суммы оплаты относительно объекта оплаты/доставки. Как пример реализации, смотрите media/script/products_comp_calcScript.inc, который показывает, как поднять окончательную сумму с 5.75% с тем, чтобы скомпенсировать сумму, необходимую для оплаты организациям кредитной карточки. Свойства вычисляемого скрипта передаются в функцию как \$conf массив.	
handleScript	resource	PHP скрипт, который включается в “чистый” метод, вызываемый из products_basket() в user_products классе, когда оформление заказа окончено. Эта функция должна позаботиться о отображении шаблона в течение процесса оплаты так же хорошо, как о завешении заказа впоследствии. Смотрите pi/payment_DIBS.php для примера. Файл HTML-шаблона следует. Свойства обрабатывающего скрипта передают в функцию \$conf массив. Содержимое переменной \$content возвращается как контент.	
handleURL	string	Если установлена, эта handleURL вызывается вместо THANKS-url (by PIDthanks) в заказе, что позволяет handleScript процессу предоставлять информацию, если оплата происходит по кредитной карте или иначе.	
handleTarget	string	Альтернативный target для формы.	
excludePayment	list of integers	shipping ONLY: Это список ключей методов оплаты (их номера), которые не доступны данному способу доставки. Например, если люди выбирают товары в магазине, и вы не хотите, чтобы они платили онлайн или передавали деньги из рук в руки, а только заказывали товары – вы можете исключить эти ненужные методы оплаты из списка. Example: ... 40.title = Pick up in store 40.excludePayment = 10,40 }	
show	boolean	Если установлено – товары показываются в списке.	1
showLimit	double	Если установлено, тогда этот товар будет показан только в случае если осталось не менее данного количества товаров в корзине. 0 ... всегда показывает этот элемент	0
type	string	Значение числа: count ... число продуктов weight ... вес в граммах price ... общая стоимость продуктов	

Property:	Data type:	Description:	Default:
WherePIDMinPrice	<i>list of integers</i>	<p>shipping ONLY: Устанавливает минимальную стоимость доставки, если есть товар в корзине из данной системной папки с данным PID. Где 155 - это PID и 7.5 минимальная цена за доставку, когда, по крайней мере, один продукт, в корзине из данной странице с данным PID.</p> <p>Example:</p> <pre> plugin.tt_products.shipping { 10.title = Parcel 10.priceTax.type = weight 10.priceTax.WherePIDMinPrice.155 = 7.5 10.priceTax.1 = 1.5 10.priceTax.500 = 2.5 10.priceTax.1000 = 3.5 } </pre>	

Pricecalc, discountprice and creditpoints configuration

Pricecalc дает вам возможность вычислять стоимость продуктов, используя таблицу вычислений. Скидка (discount price) будет использована для всех пользователей, которые состоят в группе с названием, установленным в discountGroupName.

Property:	Data type:	Description:	Default:
prod	two-edged list of integers	<p>The left edge of integers correspond to lines belonging together, the meaning of the right edge depends on the settings for each line.</p> <p>Pricecalc: Специальная цена для продуктов. Где 1 цена продукта 4.99, 2 продукт будет стоить 8.99. Со скидкой это будет форма для продукта, с pricecalc это цена для всех продуктов вместе, где 1 стоит 4.99 в системной папке продуктов. Discountprice "перегружает" pricecalc, если возможно, потому что это будет более дешевым тогда. Цена, подсчитанная здесь, будет перезаписана, если price2 используется.</p> <p>Example:</p> <pre>pricecalc { 10.type = count 10.field = price 10.where = 10.prod.1 = 4.99 10.prod.2 = 8.99 10.prod.5 = 19.99 20.type = count 20.field = price 20.where = 20.prod.1 = 6.99 20.prod.2 = 13.98 20.prod.5 = 29.99 }</pre>	0
		<p>Discountprice: Здесь простая цена продуктов вычисляется в зависимости от количества продуктов, если type=count. Дополнительные установки говорят о том, подсчитываются ли все продукты вместе даже с различных строк (lines)..</p> <p>Example:</p> <pre>discountprice { additive = 1 10.type = count 10.field = price 10.where = 10.prod.1 = 4.99 10.prod.100 = 2.49 10.prod.1050 = 2.39 20.type = count 20.field = price 20.where = 20.prod.1 = 6.99 20.prod.100 = 2.59 20.prod.1050 = 2.49 }</pre>	
		<p>Creditpoints: Это говорит вам как много creditpoints будет даваться покупателю если куплены продукты в магазине. Правильные значения – это проценты от цены заказанных продуктов, если type=price.</p> <p>Example:</p> <pre>creditpoints { pricefactor = 0.5 10.where = 10.type = price 10.prod.1 = 0.02 10.prod.101 = 0.04 10.prod.501 = 0.06 }</pre>	
additive	double	Только правильные discount price. Если установлено – все продукты с любым из этих discount prices подсчитываются вместе which discount price will apply. Если не установлено, продукты с собственной ценой подсчитываются.	
type	string	Meaning of the right edge integer which usually gets calculated: count ... the products count (pricecalc and discountprice only) price ... the price field is used (creditpoints only)	

Property:	Data type:	Description:	Default:
factor	double	Подсчитывается, как много денег кто-либо будет получать за свои кредитпоинц. 2 creditpoins дают 1 Euro или другая валюта по выбору. Example: <pre>creditpoints { pricefactor = 0.5 }</pre>	

[tsref:(script).productsLib.payment/(script).productsLib.shipping]

Наследование из from zk_products 1.3.2

Property:	Data type:	Description:	Default:
max_note_length	int+	В list view: Примечание будет ограничено после max_note_length символов, и будут добавлены в конец точки... Это может быть опасно, если вы используете html в замечаниях по продукту, because a tag could stay open and the following items look ugly	100
specialPreparation	string	Этот текст будет вставлен в ###PRODUCT_SPECIAL_PREP### маркеры.	
createUsers	Boolean	Если вы установите это и PIDuserFolder, для каждого покупателя, который не залогинился и его email address как пользователя, новый frontend user будет создан. Email с именем пользователя (его e-mail address) и password будут посланы ему (смотрите ###EMAIL_NEWUSER_TEMPLATE###)	0
PIDuserFolder	int+	Sysfolder, где новые пользователи будут сохраняться	116
memberOfGroup	int+	ID frontend user group, новые пользователи будут в этой группе	1
PIDmemo	int+	ID memo page	
PIDagb	int+	PID страницы со стандартным соглашением о торговле("AGB" в Германии) Только если этот id установлен, AGB проверка будет активна.	0
Agbtarget	Small	Target для PIDagb link	_blank
AGBAttachment	Resource	Если этот файл существует, он будет приаттачен к письму, подтверждающему заказ. Администратор магазина не получит agb attachment, но CSV, если вы разрешите это.	Fileadmin/agb.pdf
GiftAttachment	Resource	Если этот файл создан, он будет приаттачен к письму заказанного подарка	
generateCSV	Boolean	Генерирование CSV для каждого заказа. Этот файл будет сохранен на сервере и послан администратору магазина по почте (с подтверждением заказа)	0
CSVdestination	../csvorders/	Директория на сервере, куда CSV файлы сохраняются. Эта директория не должна быть доступна по http, потому что иначе каждый сможет прочитать пользовательские заказы. Соответственно, не размещайте директории в папке fileadmin.	../csvorders/
CSVfields	string	Выберите, какие данные будут сохраняться в ваш CSV. По умолчанию, это все важные поля кроме примечаний	(Long, long list)
CSVinOneLine	boolean	Если вы установите это в 1, вся дополнительная информация (shipping, payment, address data, delivery note) будет применено к первому заказанному продукту. Это хорошо, если вы хотите импортировать файл куда-либо. Если вы оставите это в 0, файл будет форматирован лучшим образом в программах, наподобие Excel.	0
AlwaysInStock	boolean	Если установлено, запас не будет уменьшаться, если что-либо будет куплено.	1
showNotInStock	boolean	Если установлено, продукт будет показываться, даже если его нет в запасе.	
notInStockMessage	string	Смотрите выше.Это будет показываться, если нет продуктов в запасе и showNotInStock установлен.	Not in stock
inStockPieces	string	Это размерность для inStock.	pieces
newItemDays	int+	В LISTNEWITEMS, недавно добавленные продукты за последние n дней будут показаны	7
bulkilyWarning	string	Текст для ###BULKILY_WARNING### больших продуктов.	
bulkilyAddition	int+	Фактор, который умножается с продуктом, который является большим	
bulkilyFeeTax	int+	Такса за доставку большеразмерных продуктов.	

Property:	Data type:	Description:	Default:
New settings for payment/shipping configuration			
visibleForGroupID	int+	Payment ONLY: Этот метод оплаты сейчас доступен, если пользователь залогинился и является членом данной frontend user group	
AddRequiredInfoFields	string	Payment ONLY: Дополнительные обязательный поля в INFO странице, если данный метод оплаты выбран. Используется для оплаты кредитными картами (в основном).	

payment_DIBS.php

Properties of the handleScript for DIBS interfacing (<http://www.architrade.com/uk/>):

Property:	Data type:	Description:	Default:
templateFile	resource	Файл шаблона для использования DIBS Вы должны положить следующую строку в form в tt_products шаблоне перед вызовом DIBS скрипта: <pre><input type="hidden" name="products_cmd" value="cardno"></pre>	
soloe	boolean	Если установлено, скрипт используется под-шаблон с маркером ###DIBS_SOLOE_TEMPLATE### вместо дефолтного ###DIBS_CARDNO_TEMPLATE###	
direct	boolean	Если установлено, скрипт использует под-шаблон с маркером ###DIBS_DIRECT_TEMPLATE### вместо дефолтного ###DIBS_CARDNO_TEMPLATE###	
merchant	boolean	Торговый (Merchant) id	
currency	int+	Номер валюты, ISO4217 format	
relayURL	string	Url магазина, где безопасный сервер обрабатывает корзину.	
test	boolean	Если установлено, test-field установлено в формах.	
cardType	string	Тип карт, Example values: DK = Dankort V-DK = Visa-Dankort MC(DK) = Mastercard/Eurocard issued in Danmark VISA = Visakort issued abroad MC = Mastercard/Eurocard issued abroad DIN(DK) = Diners Club, Denmark DIN = Diners Club, international	
account	string	DIBS account feature	
addOrderInfo	boolean	Если установлено, инфо о заказе добавляется в форму. DIBS может собирать эту инфу и отображать в процессе оплаты.	
k1 k2	string	DIBS key values	

[tsref:(script).productsLib.paymentDIBS]

Tutorial

Example of a configuration from Inter-Photo A/S (www.inter-photo.dk):

```
xxxxx.xxxx {
  10.title = Dankort, VISA-Dankort
  10.image.file = media/logos/dankort.gif
  10.image.params = align=absmiddle vspace=2
  10.handleScript = media/scripts/payment_DIBS.php
  10.handleScript {
    merchant = xxxxxx
    test = 0
    k1 = xxxxxx
    k2 = xxxxxx
    currency = 208
    addOrderInfo = 1
    account =
    cardType = DK,V-DK
    relayURL = http://www.inter-photo.dk/index.php?id=204
  }
  10.handleURL = index.php?204
  10.handleTarget = _top

  20 < .10
  20.title = Unibank e-betaling
  20.image.file = media/logos/soloe.gif
  20.handleScript.soloe=1
}
```

Example in product view make product title the page's title

Если вы хотите устанавливать названия продуктов в некоторые маркеры, например, для использования в `automake_template` или подобном, попробуйте этот код:

Code Listing:

```
# first set pagetitle to the page's title
temp.pagetitle = TEXT
temp.pagetitle.field = title

# now overwrite with the product's title in case it's non-empty
[globalVar = GP:tt_products > 0]
temp.pagetitle = COA
temp.pagetitle {
  10 = RECORDS
  10 {
    source.data = GPvar:tt_products
    tables = tt_products
    conf.tt_products = TEXT
    conf.tt_products {
      field = title
    }
  }
}
// For single record display, cache has to be disabled.
config.no_cache = 1
[global]

temp.mainTemplate = TEMPLATE
temp.mainTemplate {
  # Feeding the content from the Auto-parser to the TEMPLATE cObject:
  template =< plugin.tx_automaketemplate_pil
  # Select only the content between the <body>-tags
  workOnSubpart = DOCUMENT_BODY
  subparts.title < temp.pagetitle
  ....
}
```

Example templates

Здесь представлена часть шаблона, где новый пользователь регистрируется автоматически как front end user.

```
<BR><BR><BR><BR><HR><BR><BR><BR>
<h3>EMAIL_NEWUSER_TEMPLATE</h3>
<em>
Subpart used as template for the account-creation-emails. First line is
used as subject for the mail.<br>
</em><br>
<pre>
<!-- ###EMAIL_NEWUSER_TEMPLATE### begin
```

```

Subpart used as template for the account-creation-emails First line is
used as subject for the mail.
-->
New user account created
Dear ###PERSON_NAME###,
you have made an order at http://...../ for the first time.
To make orders in the future more easy a user account has been created.
Your account data:
user name: ###USERNAME###
password: ###PASSWORD###
Regards,

the Shopmaster
<!-- ###EMAIL_NEWUSER_TEMPLATE### end -->
</pre>

```

Example bananaGuard

(смотрите файл `example_template_bill_de.tmpl` delivered from <http://bananaguard.de>)

Этот пример включает вычисление специальной цены и автоматическое создание счета и списка товаров для доставки. Вы должны инсталлировать extensions `feuser_admin` и `conf_users`, если хотите создать подобный магазин. Переместите `example_template_bill_de.tmpl` и `fileadmin/tmpl_files/products_eur_.tmp`. Для использования этим шаблоном вы должны найти `'?id='` в вашем тексте и заменить на следующие PIDs вашей системы.

The pages and plugins for the page tree are:

- 'BananaGuard' with main template of your choice and

Constants:

```

plugin.tx_srfuserregister_pi1.email = info@shopms.de
plugin.tx_srfuserregister_pi1.confirmPID = 83 ... You have to use your confirm PID.
styles.content.loginform.pid = 108 ... Your Users sysfolder PID.

```

-- 'Home' which is a shortcut to BananaGuard

-- 'Preise und Versand' with special information about the prices and shipping

-- 'BananaShop' which has its own template especially for usage with the shopping system

Constants:

```

plugin.tt_products.file.templateFile = typo3/ext/tt_products/pi/products_eur_.tmpl
plugin.tt_products.TAXpercentage = 16
plugin.tt_products.priceDecPoint = .
plugin.tt_products.maxW_list = 80
plugin.tt_products.outputfolder = fileadmin/data

```

```

plugin.tt_products.color2 = #003399
plugin.tt_products.color1 = #FFFFFF
plugin.tt_products.wrap2 = <font face="verdana" size=2 color=white> | </font>
content.tableCellColor = #003399

```

Setup:

```

plugin.tt_products.code.field = select_key
plugin.tt_products.AlwaysInStock = 1

```

```

plugin.tt_products.statusCodes.1 = Bestellungseingang
plugin.tt_products.statusCodes.11 = Der bananaSHOP wartet auf Ihren Zahlungseingang
plugin.tt_products.statusCodes.20 = Ihre Ware wird versendet
plugin.tt_products.statusCodes.101 = Bestellung abgeschlossen
plugin.tt_products.statusCodes.200 = Bestellung storniert

```

```

plugin.tt_products.orderEmail_from = info@bananaguard.de
plugin.tt_products.orderEmail_fromName = bananaGUARD.de
plugin.tt_products.orderEmail_to = info@bananaguard.de
orderNumberPrefix = order2005_

```

```

plugin.tt_products.discountGroupName = Team
plugin.tt_products.lockLoginUserInfo=true

```

```

plugin.tt_products.displayBasketColumns = 3
plugin.tt_products.outputFolder = {$plugin.tt_products.outputfolder}

```

```

plugin.tt_products {
    payment >

```

```

payment {
    radio = 1
    TAXpercentage = 16
    10.title = Vorkasse
    20.title = PayPal
    20.percentOfTotalShipping = 0.04
    30.title = Nachnahme
    30.discountDeactive = 1

```



```

30.priceTax.1 = 4
30.priceTax.6 = 5.8
30.showLimit = 99
}

shipping {
  radio = 1
  TAXpercentage = 16
  10.title = Deutschland
  10.image.file =
  10.priceTax.type = count
  10.priceTax.1 = 2.5
  10.priceTax.6 = 4.8
  10.priceTax.50 = 10
  10.priceTax.100 = 25
  10.priceTax.120 = 30
  10.priceTax.300 = 150
  10.percentOfGoodstotal = 0

  20.title = Europa (nicht Deutschland)
  20.image.file =
  20.priceTax.type = count
  20.priceTax.1 = 8.8
  20.priceTax.6 = 11.8
  20.priceTax.50 = 11.8
  20.priceTax.100 = 25
  20.priceTax.120 = 30
  20.priceTax.300 = 150
  20.percentOfGoodstotal = 0

  30.title = Selbstabholung
  30.image.file =
  30.priceTax.1 = 0
  30.percentOfGoodstotal = 0
}

pricecalc {
  type = count
  field = price
  10.prod.1 = 4.99
  10.prod.2 = 8.99
  10.prod.5 = 19.99
  20.prod.1 = 6.99
  20.prod.2 = 13.98
  20.prod.5 = 29.99
}

discountprice {
  type = count
  field = price
  additive = 1
  10.prod.1 = 4.99
  10.prod.100 = 2.89
  10.prod.1050 = 2.77
  20.prod.1 = 6.99
  20.prod.100 = 3.00
  20.prod.1050 = 2.89
}

}

plugin.tt_products.basketImage.imageLinkWrap.height = 800
plugin.tt_products.listImage.file.maxW = 150
plugin.tt_products.shipping.40 >
plugin.tt_products.PIDagb =

plugin.tt_products.createUsers = 1
plugin.tt_products.orderEmail_htmlmail = 1

```

Plugin:

Products, CODE: LIST

The next pages are:

--- 'Warenkorb'

Plugin:

Products, CODE: BASKET, At the BORDER a login content type with send to page 'Warenkorb'

--- 'Kasse'

Plugin:

Products, CODE: PAYMENT, FINALIZE, INFO

- 'Bestellungsstatus'
 - Plugin:
 - Products, CODE: TRACKING
- 'Rechnung'
 - Plugin:
 - Products, CODE: BILL
- 'Lieferung'
 - Plugin:
 - Products, CODE: DELIVERY
- 'Articles' Sysfolder
 - Plugin:
 - Products, CODE: LIST
- 'Kontakt/Bestellung' contains a form, text and login
- 'BananaINSIDER' only visible after frontend login, contains a login plugin at the right border
 - Access of the visibility settings is the group 'Team'
- 'My profile'
 - Plugin:
 - Frontend User Registration
- 'INSIDER Infos'
- 'bananaBOARD'
 - Plugin:
 - Board, Tree, CODE: FORUM, POSTFORM
- 'Users' Sysfolder of the group 'Team'

Known problems

- Please look at the website http://wiki.typo3.org/index.php/Ext_tt_products .
 - Get the latest development version at http://fholzinger.com/development/tt_products-2.2.6.<Subnr>.zip .
- <Subnr> ... Guess the latest subnumber of the latest tt_products version.

To-Do list

- Code improvements to use the same classes as the commerce extension.

Changelog

- 26.03.2005 bring in of zk_products from Klaus Zierer (zk_products), multiple column listing, bill , delivery and special price calculations by Franz Holzinger
- 06.05.2005 second price and price for additional accessory from Jens Schmietendorf, example template from <http://bananaguard.de> , zk_products 1.3.2 from Klaus Zierer, VAT by Franz Holzinger
- 23.06.2005 products_mail.inc deleted, example template by Franz Holzinger
- 26.07.2005 Display orders, creditpoint and voucher system by Bert Hiddink
- 03.08.2005 Notes from the zk_products forum, PIDtracking
- 11.09.2005 Gift certificates
- 14.09.2005 Flexforms instead of CODEs