

# Kentico CMS 6.0 E-commerce Guide



# Table of Contents

<b>E-commerce Guide</b>	<b>7</b>
Kentico CMS E-commerce module.....	7
<b>Getting Started</b>	<b>9</b>
Making your first purchase.....	9
Adding new products.....	14
Adding product options.....	18
Customizing product categories.....	23
Adding product to multiple categories.....	27
Displaying featured products.....	30
Sorting and paging products.....	32
Changing company details.....	36
<b>Customizing the product fields</b>	<b>43</b>
Defining a new product type.....	43
Customizing product design.....	51
Adding images to product gallery.....	53
Adding product custom fields.....	56
<b>Customizing web site design</b>	<b>60</b>
Changing the header and menu (master page).....	60
Changing colors using CSS styles.....	69
<b>Integration with your existing Kentico CMS website</b>	<b>73</b>
Overview .....	73
Web parts.....	74
My account .....	74
Random products .....	77
Shopping cart .....	81
Shopping cart preview .....	84
Similar products by sale .....	86
Top N products by sale .....	87
Top N newest products .....	89
Wishlist .....	91
Displaying and resizing images.....	93
Overview .....	93
Displaying images .....	93
Storing images .....	96
Resizing images .....	96

<b>Tools.....</b>	<b>99</b>
Overview .....	99
Forms .....	100
Blogs .....	101
Events .....	101
File import .....	102
Forums .....	102
Geo mapping .....	103
Image Gallery .....	104
Messaging .....	105
Newsletters .....	106
Polls .....	107
Reporting .....	107
RSS Feeds .....	108
Staging .....	108
User Contributions .....	109
Web Analytics .....	109
Web Farm synchronization .....	110
Adding items to the shopping cart.....	111
Adding items to the wish list.....	114
Displaying product price.....	115
Getting product URL.....	116
Searching.....	116
 <b>Configuration settings .....</b>	 <b>119</b>
Configuration overview.....	119
Orders.....	120
Customers.....	121
Products.....	123
Product options.....	126
Manufacturers.....	127
Suppliers.....	127
Discount coupons.....	127
Discount levels.....	127
Reports.....	128
Configuration.....	128
Store settings .....	128
Departments .....	129
Shipping options .....	130
Payment methods .....	132
Tax classes .....	134
Currencies .....	134
Exchange rates .....	135
Order status .....	135
Public status .....	135
Internal status .....	136
Invoice .....	136

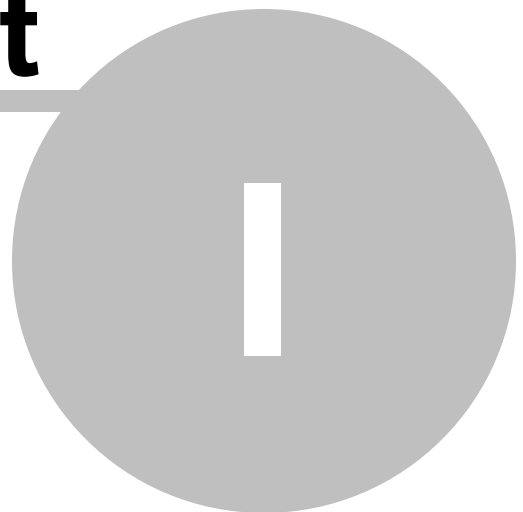
Enabling the e-commerce module.....	137
Countries and states.....	139
E-commerce and multi-site configuration.....	140
Mapping document fields to product properties.....	141
Site Manager settings.....	141
Web.config settings.....	144
Localization settings.....	145
Customizing invoice and e-mail templates.....	147
Using product inventory.....	152
Discounts overview.....	157
Deleting old shopping carts.....	162
<b>Site separation</b> .....	<b>165</b>
Overview.....	165
Basic configuration scenarios.....	165
Enabling the use of global objects.....	166
Objects with both site and global option.....	167
Objects with only site or global option.....	169
Objects with special cases.....	170
Security.....	170
<b>Product types</b> .....	<b>172</b>
Overview.....	172
Product.....	172
Membership.....	172
E-product.....	175
Donation.....	178
Bundle.....	180
Product with the text product option.....	182
<b>Purchase process and payment gateways</b> .....	<b>187</b>
Purchase process overview.....	187
Customizing the purchase process.....	188
Developing custom dialogs for the checkout process.....	190
Payment gateways.....	196
Authorize.NET configuration.....	197
PayPal configuration.....	199
Developing custom payment gateways.....	201
Payment results.....	207
Customer credit.....	209



<b>Security</b>	<b>212</b>
Overview.....	212
Permissions.....	212
Department administrators.....	214
UI personalization.....	214
E-commerce sample roles.....	221
<b>Developing custom providers</b>	<b>226</b>
Overview.....	226
<b>E-commerce internals and API</b>	<b>228</b>
Overview.....	228
Database tables.....	228
API classes.....	237
API examples.....	243
Overview .....	243
<b>Figure A: Tax calculation process</b>	<b>245</b>
<b>Figure B: Discount calculation</b>	<b>247</b>
<b>Figure C: Shopping cart price calculation</b>	<b>249</b>
<b>Figure D: Shopping cart retrieval</b>	<b>251</b>

# Part

---



**E-commerce Guide**

---

# 1 E-commerce Guide

## 1.1 Kentico CMS E-commerce module

The Kentico CMS E-commerce module provides you with a range of out-of-box solutions for your E-business. The task of developing and managing the e-business website, once painfully slow and complex, has been diminished to little more than few clicks.

For customers, the built-in E-commerce module offers a possibility of making purchase on-line through the integrated shopping cart, checking orders status, subscribing to a newsletter etc.

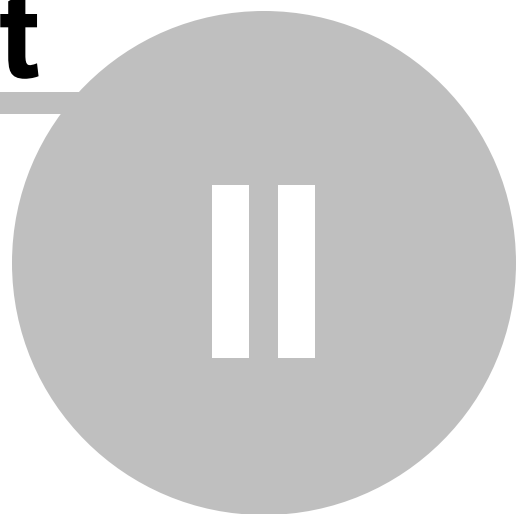
For site owners, the module offers tools for managing orders, shipping and payment options, products or manufactures lists and much more.

It supports:

- Product management
- Product options management
- Custom product types
- Product image gallery
- Product inventory
- Order management
- Customer management
- Multiple currencies
- Configurable tax calculation based on country and state
- Discount coupons
- Discount levels
- Volume discounts
- Wish list
- Custom providers for alternative shipping and tax calculations
- Custom checkout process
- Built-in payment processors, such as PayPal, Authorize.NET
- Custom payment processors
- Reports and statistics

# Part

---



## Getting Started

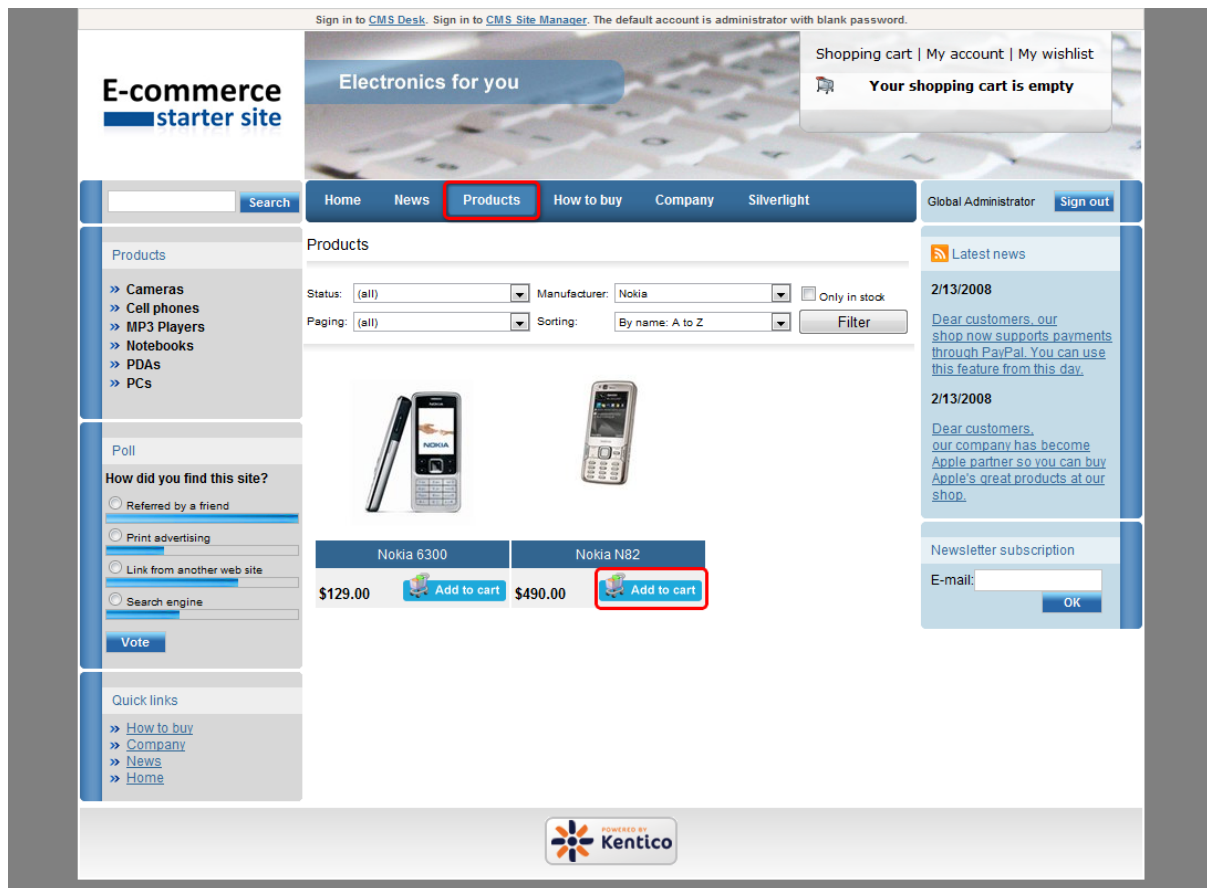
---

## 2 Getting Started

### 2.1 Making your first purchase

For start, you will learn how to buy a product on your website. This experience will help you in revealing any possible pitfalls which might be awaiting your customers on your website.

1. Open your favorite internet browser and go to your website. Go to the **Products** tab and click the **Add to cart** button below **Nokia N82**.



2. Now specify your order. Choose **Black** as **Color** and enter **4** into the text box next the **Add to cart** button. You are about to add 4 pieces of Nokia N82 in black color to your shopping cart. Click **Add to cart**.

## Nokia N82



### Parameters:

Width: 112  
 Height: 50  
 Weight(g): 114  
 Display type: TFT 16M colors  
 Display resolution: 240 x 320  
 Bluetooth: yes  
 IrDA: no  
 GPRS: yes  
 EDGE: yes  
 HSCSD: yes  
 3G: yes  
 Wi-Fi: yes  
 Java: yes  
 Built-in camera: yes  
 MP3 player: yes

Our price: \$563.50  
 Without tax: \$490.00

### Nokia N82 - Color

- ☐ Silver (+ \$0.00)  
☒ Black (+ \$10.00)

Total price (without tax): \$500.00



3. You have been redirected to the shopping cart. In the **Step 1**, click the first **Price detail** button.

Step 1 of 6 - Add some products to the shopping cart

Shopping cart

Currency: U.S. Dollar

Remove	Product name	Units	Unit price	Unit discount	Tax	Subtotal
	<a href="#">Nokia N82</a>	4	490.00	29.40	276.36	118.76
	- Black	4	10.00	0.00	6.00	46.00

If you have a coupon code, please enter it here:

Total shipping: \$0.00  
 Total price: **\$2164.76**

4. Now you can see detailed description of the price of your order including the volume discount and taxes. Close the window.

Product price detail - Windows Internet Explorer

### Product price detail

**Nokia N82**

Unit price without tax	\$490.00
------------------------	----------

Discounts	Per unit
- VIP customers	10% \$49.00
- Volume discount for minimal amount of 3 units	6% \$26.46
Total discount	\$75.46
Price after discount	\$414.54

Taxes
Total tax
Price with tax

Total
Units
Total price

Close

5. Now click the **Check out** button at the bottom.

Step 1 of 6 - Add some products to the shopping cart

### Shopping cart

Currency: U.S. Dollar

Remove	Product name	Units	Unit price	Unit discount	Tax	Subtotal
	Nokia N82	4	490.00	29.40	276.36	2118.76
	- Black	4	10.00	0.00	6.00	46.00

If you have a coupon code, please enter it here:

Total shipping: \$0.00  
Total price: **\$2164.76**

Empty

Continue shopping


6. In the **Step 2**, choose **Create a new account**. Now enter information for your new account:

- **First name:** David
- **Last name:** Simons

- **E-mail(user name):** DavidSimons@example.com

Click **Next**.

Step 2 of 6 - Registration check



**User registration**

☐ Sign in using your existing account

☒ Create a new account

First name: David

Last name: Simons

E-mail (user name): DavidSimons@example.com

Company account: ☐

Password: ●●●●

Confirm password: ●●●●

☐ Continue as anonymous customer

Back Next


7. In the **Step 3**, enter the new billing address:

- **Name (Company or personal):** Development Soft
- **Address Line:** 1020 Blueberry Ln.
- **City:** Tucson
- **ZIP:** 85754

Please note that you can choose the shipping or company address different from the billing address by checking one of the check boxes at the bottom. For now, just click **Next**, though.



Step 3 of 6 - Select billing and shipping address



**Billing address**

Billing address: (new) ▼

Name (company or personal): Development Soft

Address lines: 1020 Blueberry Ln.

City: Tuscon

ZIP: 85754

Country: USA ▼

Alaska ▼

Phone number:


**Shipping address**

☐ My shipping address is different from the billing address.

Back Next

8. In the **Step 4**, you can choose the shipping and payment method for your delivery. Click **Next**.

Step 4 of 6 - Select payment and shipping methods



**Shipping and payment methods**


Shipping: DHL ▼

Payment: Cash on delivery ▼

Back Next

9. In the **Step 5**, make sure that you all the order information are correct. If everything looks right, click **Order now**.

Step 5 of 6 - Order preview



**Order preview**

**Billing address**  
Development Soft  
1020 Blueberry Ln.  
Tuscon  
85754  
USA, Alaska

**Shipping address**  
Development Soft  
1020 Blueberry Ln.  
Tuscon  
85754  
USA, Alaska

**Payment method:** Cash on delivery      **Shipping option:** DHL

Product name	Units	Unit price	Unit discount	Tax	Subtotal
Nokia N82	4	490.00	29.40	276.36	2118.76
- Black	4	10.00	0.00	6.00	46.00

Shipping: \$0.00  
**Total price: \$2164.76**

Tax name	Tax summary
Sales tax	282.36

**Order note:**

[Back](#) [Order now](#)

If everything went well, you can see the confirmation text.

## Thank you for your order

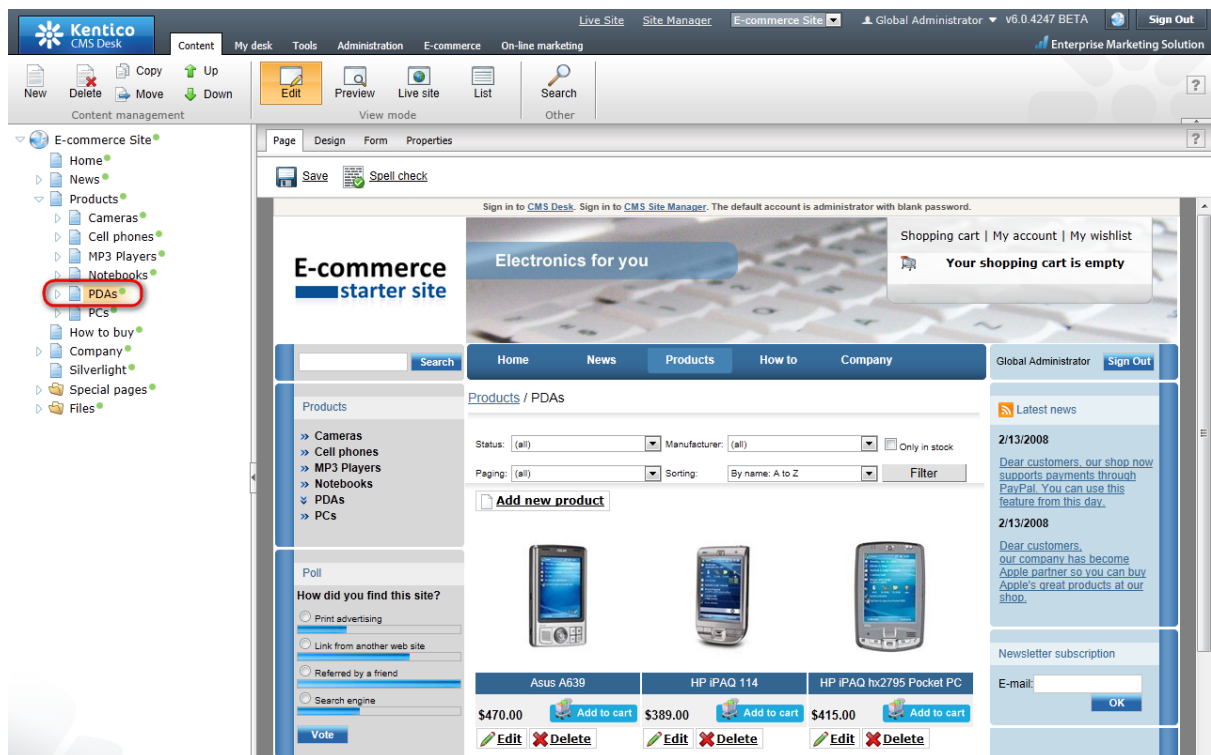
We have received your order and we will process it within 24 hours.

*This is a sample page showing the confirmation text. You can configure your payment methods in **Tools -> E-commerce -> Configuration -> Payment methods** so that the user is redirected to some third-party on-line payment gateway, such as Authorize.NET, PayPal and others.*

Congratulations, you've just made your first order.

## 2.2 Adding new products

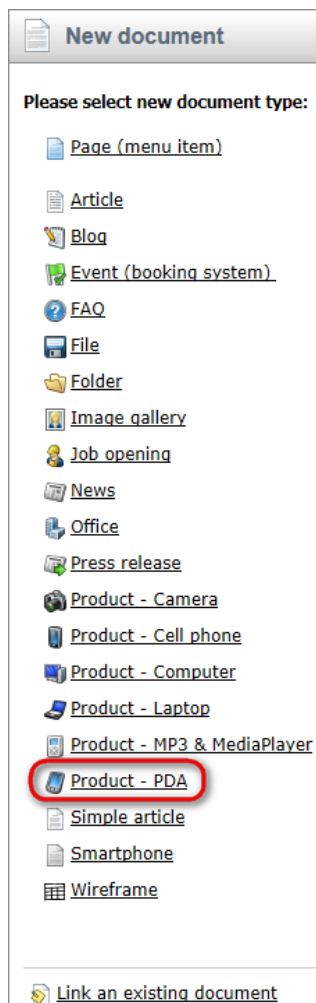
1. Sign in to **CMS Desk**.
2. Go to **CMS Desk -> Products -> PDAs**.



3. Click **New** at the document action toolbar.







4. Choose the **Product - PDA** document type.



5. Adding new product is two step process. Firstly, you enter attributes for the new document. Secondly, you enter attributes for the new product. For explanation about the difference between these two please see the box at the end of the [Products](#) chapter. Now enter attributes for the new document:

- **Name:** HTC Touch Cruise
- **Battery:** Lithium ion
- **Display type:** TFT LCD
- **Resolution:** 240x320
- **RAM(MB):** 128
- **Processor(MHZ):** 400
- **Weight(g):** 200
- **Operating system:** Microsoft Windows Mobile 6

Now enter attributes for product itself. Check the **Create a new product** checkbox and enter **900** as **Price**. Click  **Save** at the top to save your new product.

 **Save**  **Save and create another**  **Spell check**

Name:

Battery:

Display type:


Resolution:


RAM (MB):

Processor (MHz):

Weight(g):


Operating system:


Publish from:   **Now**

Publish to:   **Now**

☒ Create a new product

Price:  (USD)

Image: Upload:   **Browse...**

Description: 

Department:  ▼

The new camera has been added to your product list.



No image

**Parameters:**

Battery:	Lithium ion
Display type:	TFT LCD
Resolution:	240×320
RAM (MB):	128
Processor (MHz):	400
Weight(g):	200
Operating system:	Microsoft Windows Mobile 6

**Our price: \$900.00**  
Without tax: \$900.00



### Creating product from existing document

This chapter describes how to create a new product while creating new document. In **Kentico CMS**, you can also choose to create product from an already existing document, though.

All you have to do is to select a given document in the content tree and switch to the **Product** tab. Then check **Mark document as product**. By choosing **Select an existing product**, you can use an existing product as a template for a new product. By choosing **Create a new product**, you can create a new one from scratch.

## 2.3 Adding product options

In Kentico CMS, product options can be defined for each product. With specified product option for the given product, you can offer your customers greater variability in choosing the right product and boost your sales as well. In this chapter, you will learn how to set up a new product option for the product which hasn't had any product option specified so far.

1. Go to **CMS Desk -> Tools -> E-commerce -> Product options**.

The screenshot shows the Kentico CMS Desk interface. The top navigation bar includes 'Live Site', 'Site Manager', 'E-commerce Site', and 'Global Administrator'. The main menu has 'E-commerce' highlighted, with 'Product options' also highlighted. Below the menu, the 'Product option categories' section is visible, showing a list of categories with their selection types, enabled status, and global status.

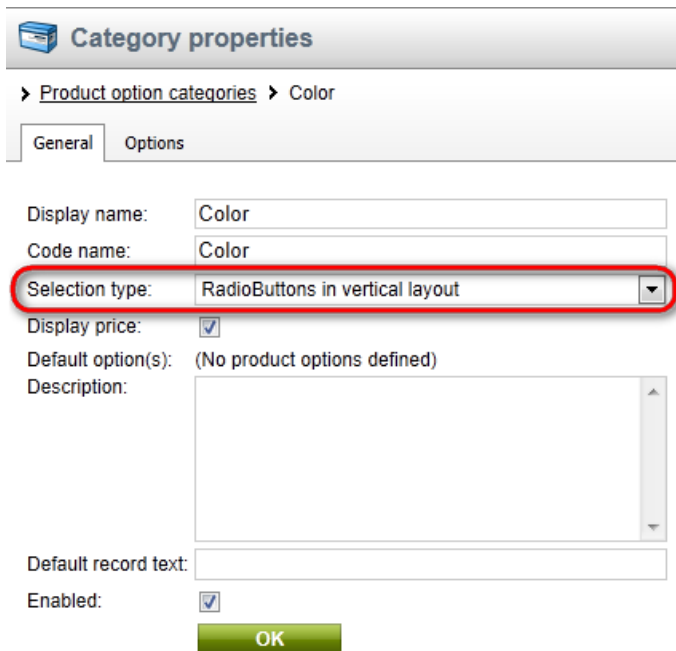
Actions	Name	Selection type	Enabled	Is global
	Cell phones accessories	Checkboxes - horizontal	Yes	No
	Creative Zen Stone - Color	Dropdown list	Yes	No
	iPod Shuffle - Colors	Radiobuttons - vertical	Yes	No
	Nokia 6300 - Color	Dropdown list	Yes	No
	Nokia N82 - Color	Radiobuttons - vertical	Yes	No
	Notebooks - AMD CPU	Dropdown list	Yes	No
	Notebooks - DDR2 modules	Dropdown list	Yes	No
	Notebooks - HDD	Dropdown list	Yes	No
	Notebooks - Intel CPU	Dropdown list	Yes	No
	Operating systems	Dropdown list	Yes	No
	PCs - Cases	Dropdown list	Yes	No

2. Click **New option category**.

3. Enter *Color* into the **Display name** and **Code name** text boxes and click **OK**.

The screenshot shows the 'New category' form. It includes fields for 'Display name' and 'Code name', both containing the text 'Color'. Below these fields is a 'Type' section with two radio buttons: 'Category with selectable options' (selected) and 'Category with one text option only'. At the bottom of the form is a green 'OK' button.

4. Select **RadioButtons in vertical layout** from the **Selection type** drop-down menu.



**Category properties**

Product option categories > Color

General Options

Display name: Color

Code name: Color

Selection type: RadioButtons in vertical layout

Display price: ☒

Default option(s): (No product options defined)

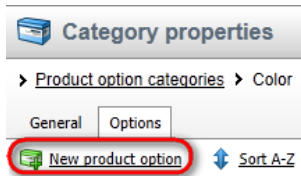
Description:

Default record text:

Enabled: ☒

OK

5. Switch to the **Options** tab and click **New product option**.



**Category properties**

Product option categories > Color

General Options

New product option Sort A-Z

6. Enter *Black* into the **Product name** text box, *0* into the **Price** text box and choose **General** from the **Department** drop-down menu. Then click **OK** at the bottom.



**Category properties**

> [Product option categories](#) > Color

General Options

> [Product options](#) > New product option

Save Save and create another

**General info**

Product name: Black

Product number:

Price: 0 (USD)

Department: General

Manufacturer: (none)

Supplier: (none)

Image: Upload: Browse...

Description:

Product type: Product

6. Click the **Options** tab and then click **New product option** again.
7. Now enter *Silver* as **Product name**, *10* as **Price** and choose **General** as **Department**. Then click **OK**.
8. Click the **Options** tab and **New product option** again.
9. Enter *Green* as **Product name**, *-10* as **Price** and choose **General** as **Department**. Then click **OK**.
10. Now switch to **Products** tab and click the **Edit** button next to the **Apple MacBook Air** product name.

The screenshot shows the Kentico CMS E-commerce interface. The top navigation bar includes 'Live Site', 'Site Manager', 'E-commerce Site', and 'Global Administrator'. The main menu has tabs for 'Content', 'My desk', 'Tools', 'Administration', 'E-commerce', and 'On-line marketing'. Under 'E-commerce', there are sub-tabs for 'Orders', 'Customers', 'Products', 'Discounts', 'Manufacturers', 'Suppliers', and 'Configuration'. The 'Products' sub-tab is active, showing a list of products. The 'New product' button is highlighted. Below the product list, there is a 'Display advanced filter' section.

Actions	Product name	Product number	Product price	Available items	Public status	Internal status	Enabled	Is global
	Acer Aspire 3105WLMi		\$490.00	0			Yes	No
	Acer configurator		\$899.00				Yes	No
	Acer Extensa 7620G		\$99.00	0			Yes	No
	Apple iPhone		\$499.00	0	Featured product		Yes	No
	Apple MacBook Air		\$2094.00	2			Yes	No
	Asus A639		\$470.00	1			Yes	No
	Asus F3U AP059C		\$999.00	-1			Yes	No
	Canon Digital Rebel XTi		\$597.00	0	Featured product		Yes	No
	Canon PowerShot A720 IS		\$195.00	95			Yes	No

11. Switch to the **Options** tab and click the **Add categories** button. In the pop-up window, select **Color** product option category and click **OK**.

The screenshot shows the 'Product properties' window for the 'Apple MacBook Air' product. The 'Options' tab is selected. The 'Add categories' button is highlighted. The 'Color' category is selected in the list below.

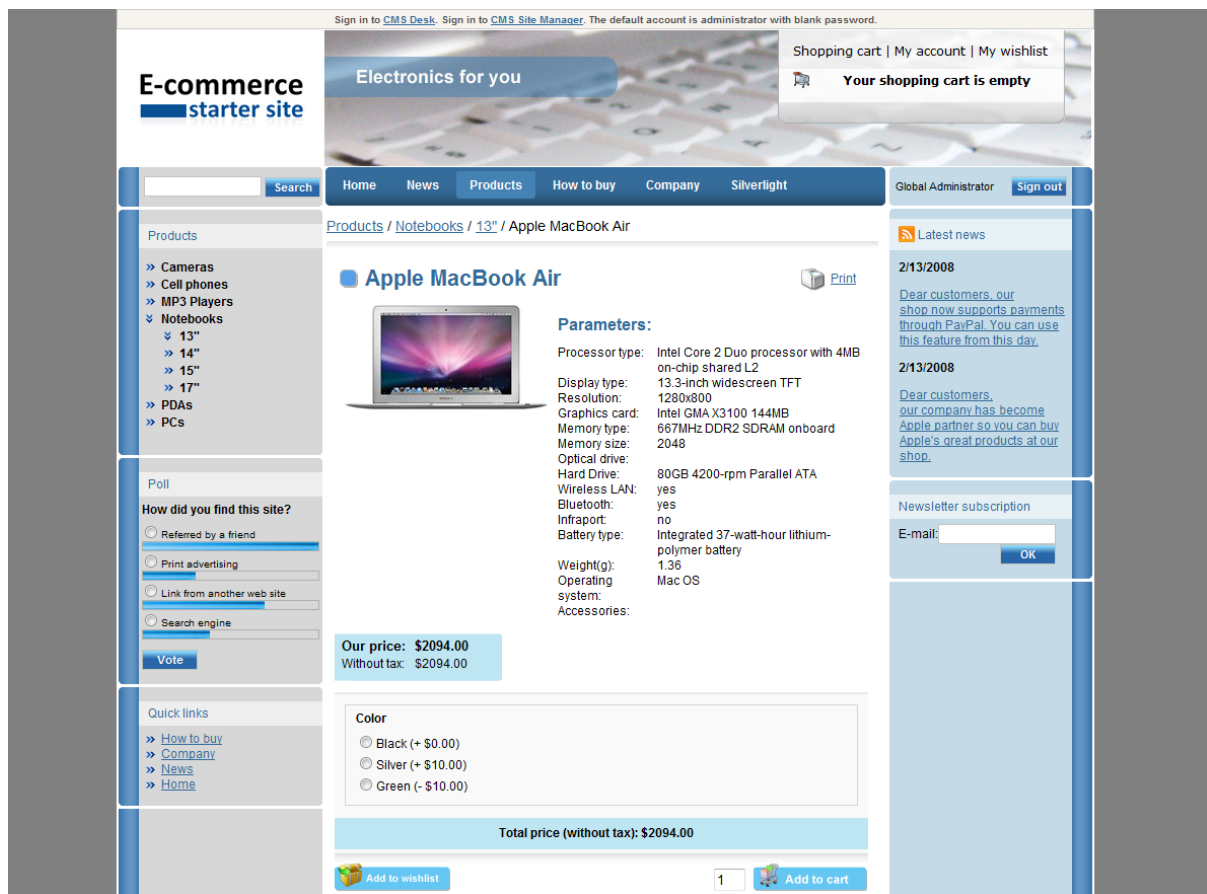
The changes were saved.

The following categories with product options are assigned to the product:

Category name
Color

Remove selected Add categories

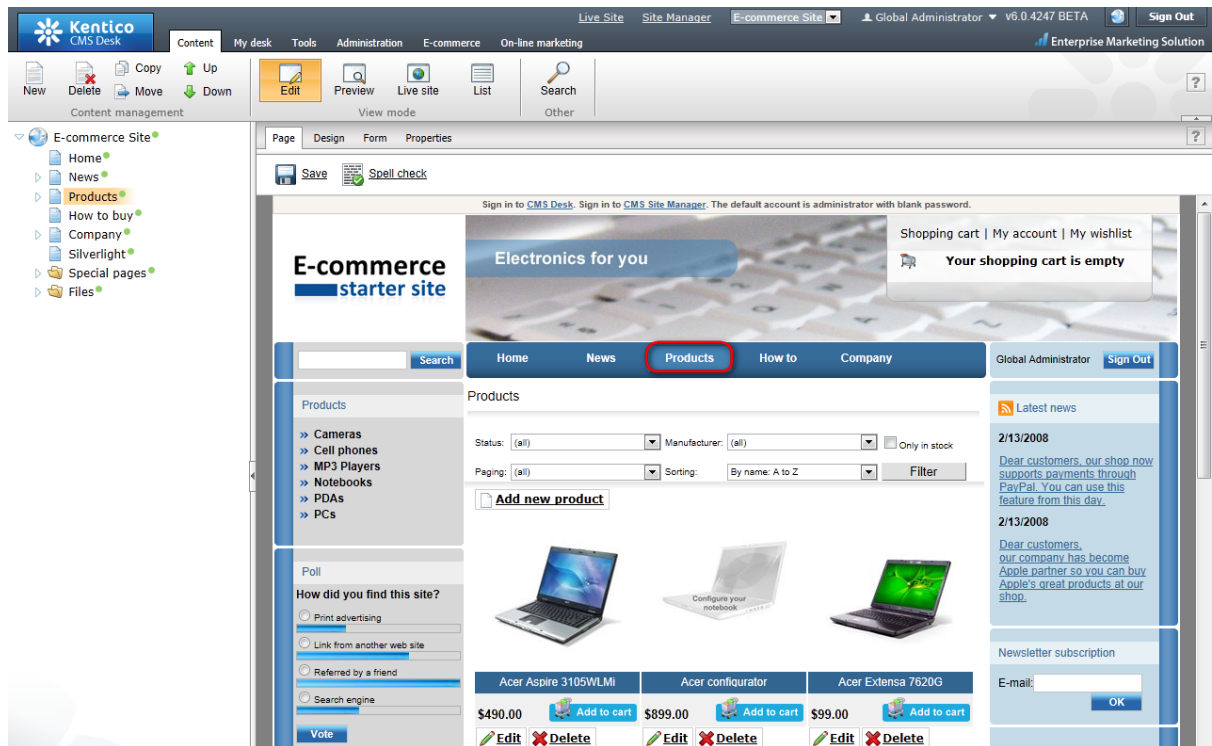
The new product option has been added to the **Apple MacBook Air** product.



For information how to display product options in product detail please refer to the [Adding items to the shopping cart](#) chapter.

## 2.4 Customizing product categories

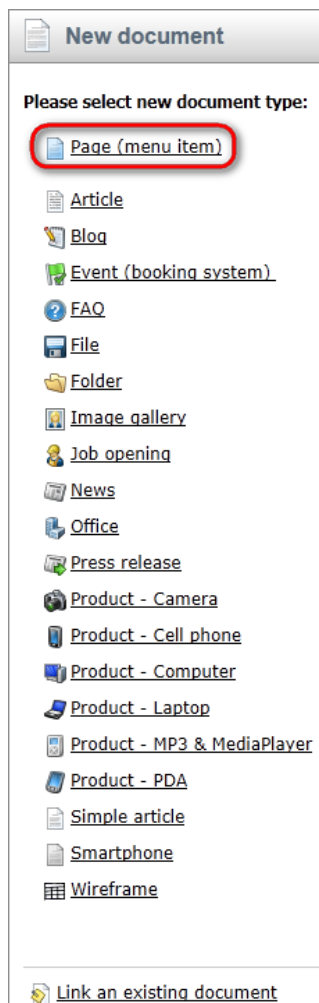
1. Go to **CMS Desk** -> **Content** and choose **Products** in the content tree.




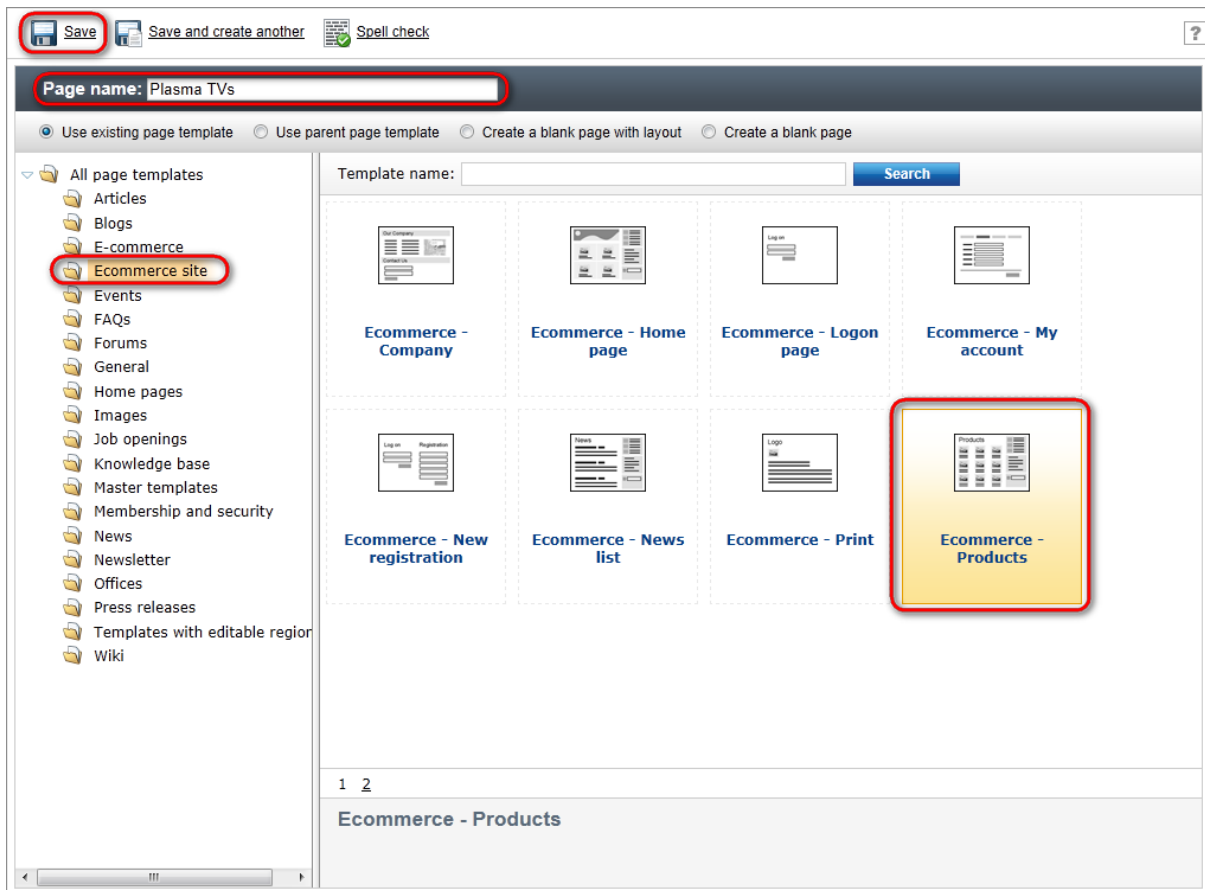
2. Click  **New** at the document action toolbar.



3. Choose the  **Page(menu item)** document type.



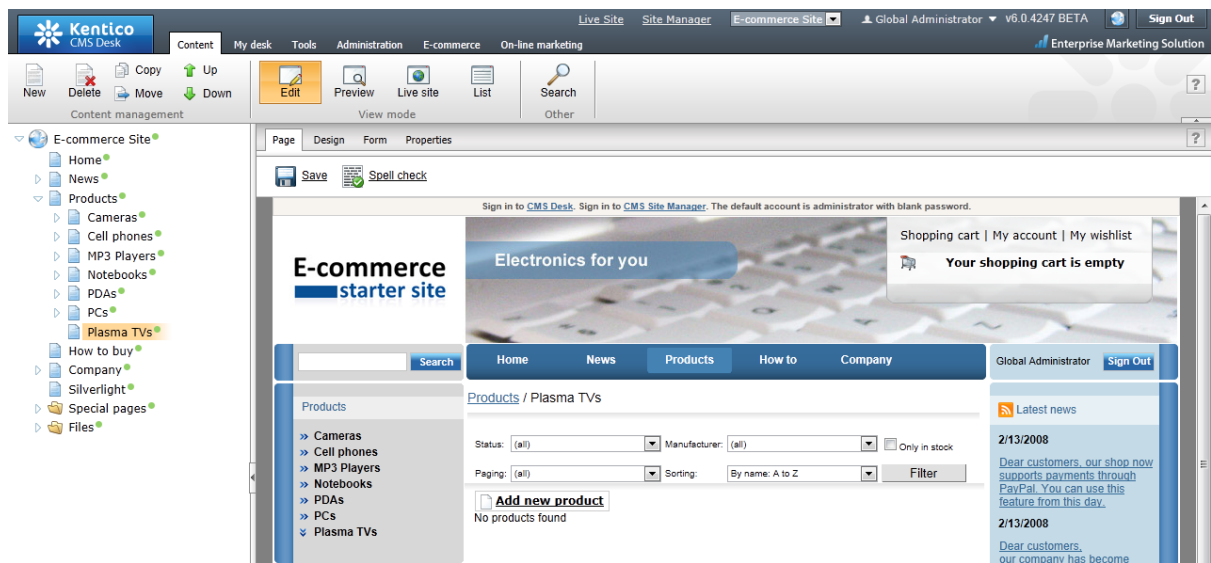
4. Enter *Plasma TVs* as the **Page name**, make sure the **Use existing page template** radio button is selected and choose **E-commerce site -> Ecommerce - Products** template. Click  **Save**.



5. Your new page has been appended to the **Products** page. Now use the **Down** button at the document action toolbar to move it to the bottom of the section.



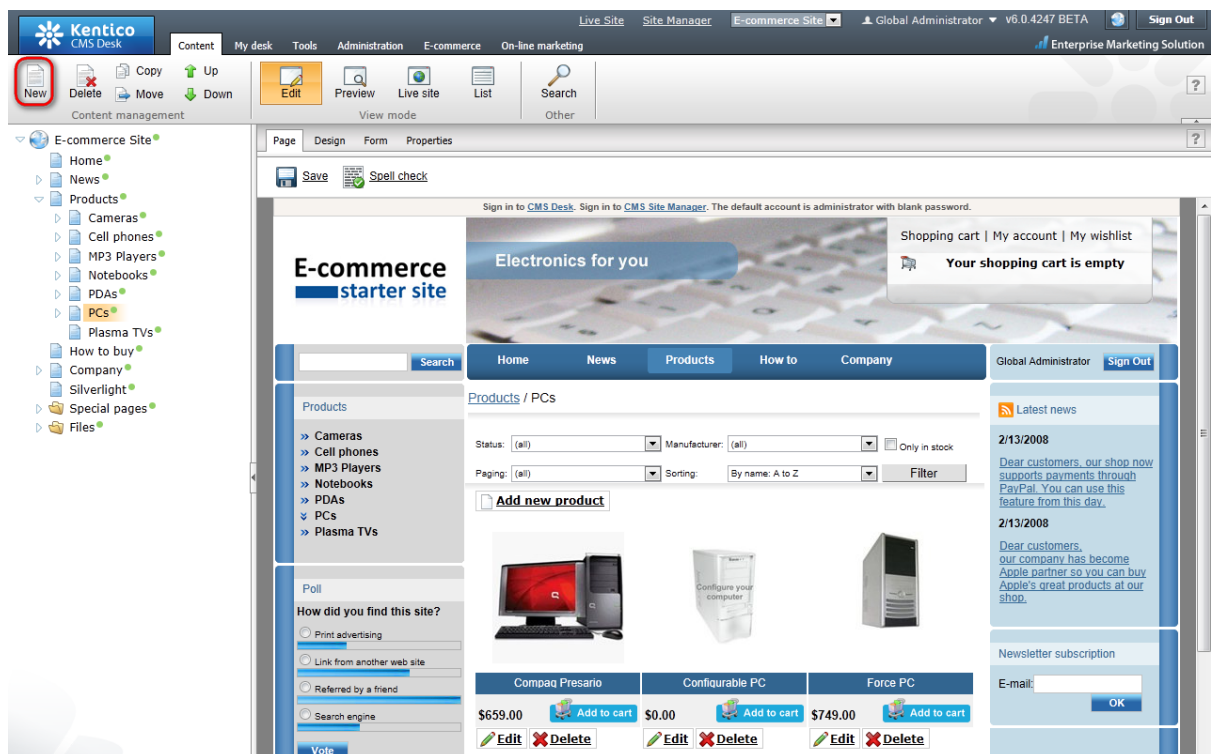
That's how you add a new category to your **Products** page.



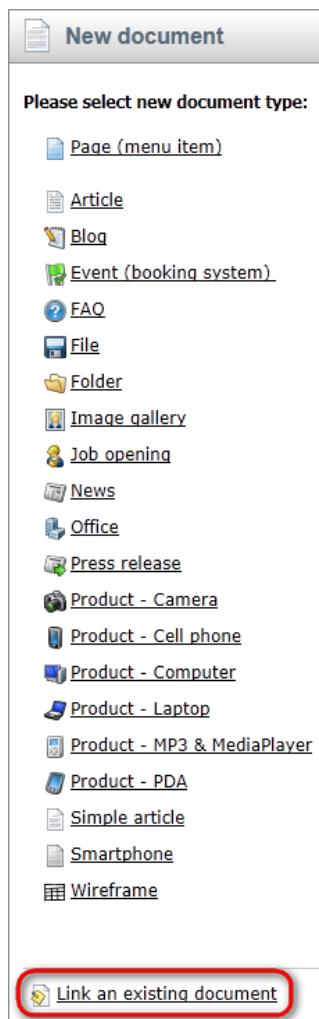
## 2.5 Adding product to multiple categories

You might need to add some product into more than one category. This chapter will help you to learn how to add a product to multiple categories.

1. Go to **CMS Desk -> Content -> Products -> PCs** and click  **New** at the document action toolbar.

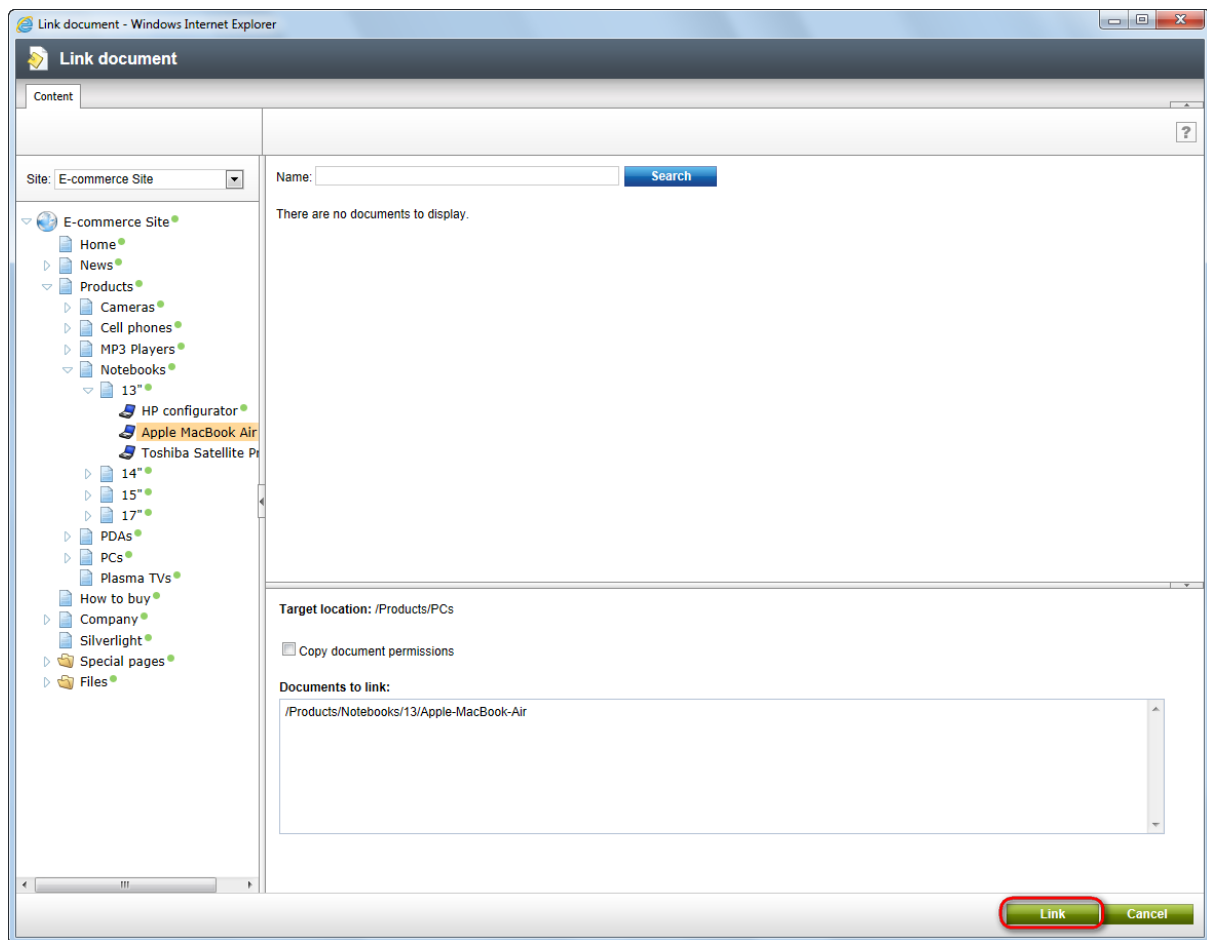


2. Click  **Link an existing document** at the bottom.

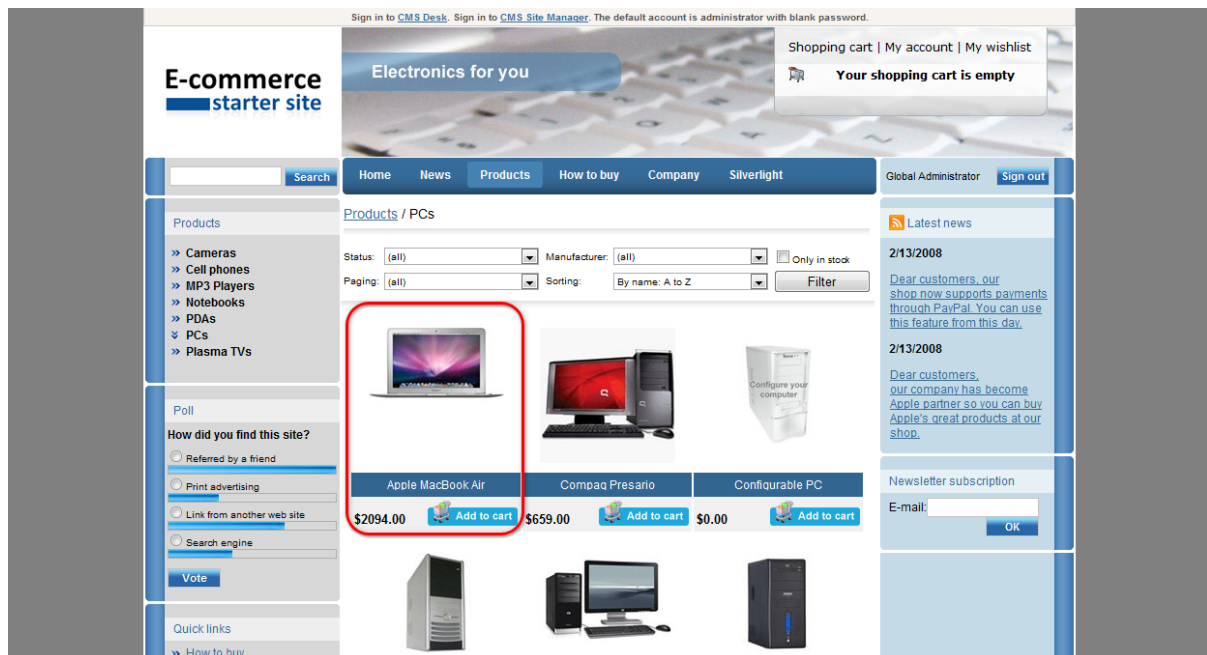


3. Now select **Apple MacBook Air** at **E-commerce Site -> Products -> Notebooks -> 13"** and click **Link**.





Apple MacBook Air has been added to the PCs category.



**Please note:**

- While copying a document, the same product (with the same SKU) that has been linked to the original document is linked to the new document as well.
- While creating link to an existing document, a product (SKU) is not directly related to the given link. The newly created link is an "image" of the existing document, so that the product is linked to the original document only.

## 2.6 Displaying featured products

1. Go to **CMS Desk -> Tools -> E-commerce** and choose **Products** tab.

Site: E-commerce Site

New product

Product name:

Product type: (all)

















Department: (all my departments)

Show


Display advanced filter

Actions	Product name	Product number	Product price	Available items	Public status	Internal status	Enabled	Is global
	Acer Aspire 3105WLMi		\$490.00	0			Yes	No
	Acer configurator		\$899.00				Yes	No
	Acer Extensa 7620G		\$99.00	0			Yes	No
	Apple iPhone		\$499.00	0	Featured product		Yes	No
	Apple MacBook Air		\$2094.00	2			Yes	No
	Asus A639		\$470.00	1			Yes	No
	Asus F3U AP059C		\$999.00	-1			Yes	No
	Canon Digital Rebel XTi		\$597.00	0	Featured product		Yes	No
	Canon PowerShot A720 IS		\$195.00	95			Yes	No
	Compaq Presario		\$659.00	-1			Yes	No
	Configurable PC		\$0.00	0			Yes	No
	Creative ZEN		\$150.00	-5	Featured product		Yes	No
	Creative Zen Stone		\$50.00	0			Yes	No

2. Click the **Edit** () button next to the product which you want to display as a featured product.


Actions	Product name
 	Acer Aspire 3105WLMi
 	Acer configurator
 	Acer Extensa 7620G
 	Apple iPhone
 	Apple MacBook Air
 	Asus A639
 	Asus F3U AP059C
 	Canon Digital Rebel XTi

3. From the **Public status** drop-down menu, choose **Featured products** and click **OK** at the bottom.

 **Product properties**

Products > Apple iPhone

General Tax classes Volume discounts Options Documents

 Save

**General info**

Product name: Apple iPhone

Product number:





Price: 499.00 (USD)

Department: Phones





Manufacturer: Apple

Supplier: (none)

Image:

Actions	Update	File name	Size
 		 iphone_hero.jpg	7.4 kB

Description:

**B** *I* |    

iPhone is a revolutionary new mobile phone that allows you to make a call by simply tapping a name or number in your address book, a favorites list, or a call log. It also automatically syncs all your contacts from a Windows PC, Mac, or Internet service. And it lets you select and listen to voicemail messages in whatever order you want — just like email.

Product type: Product

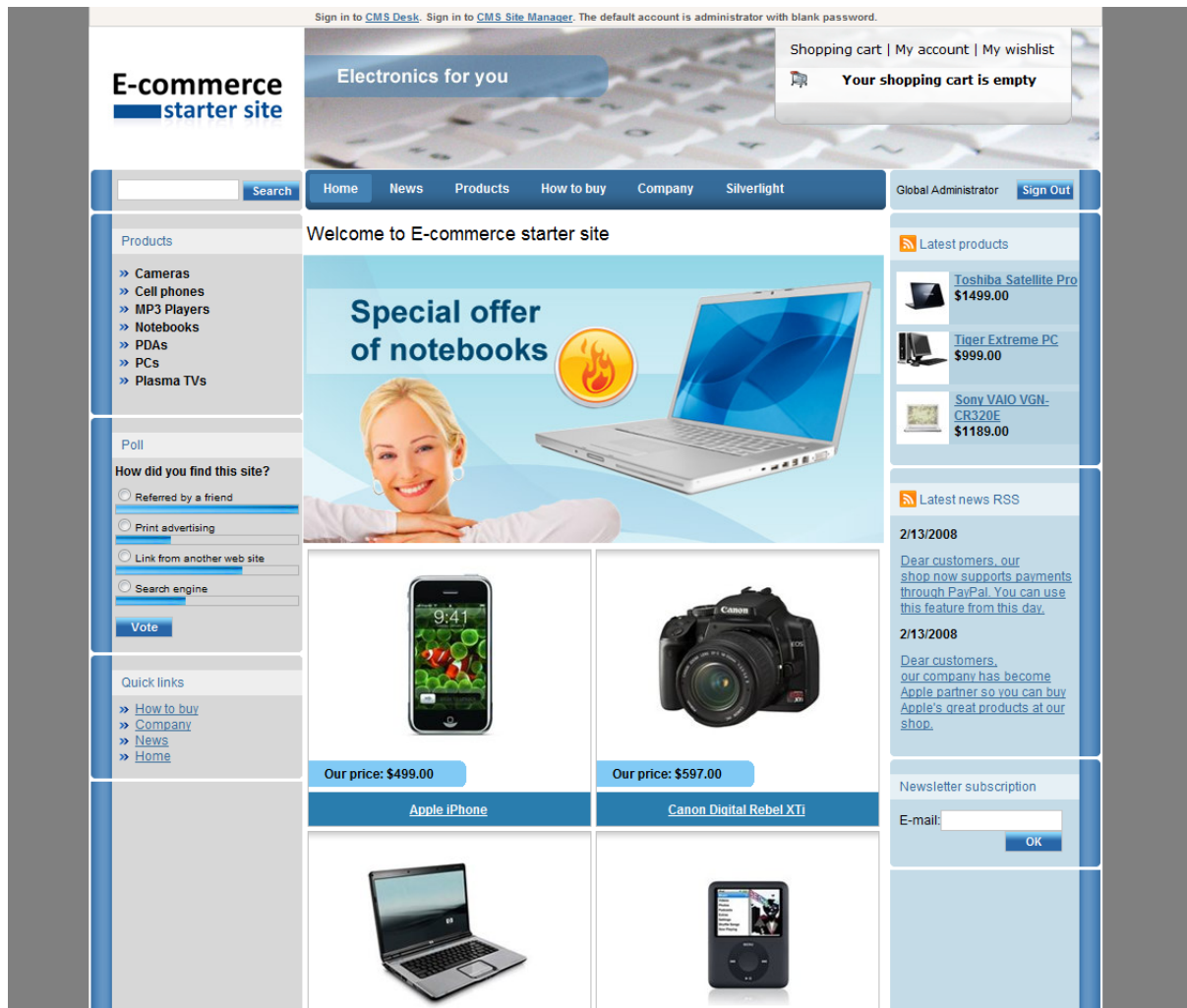
**Status**

Enabled: ☒

Public status: Featured product

Internal status: (none)

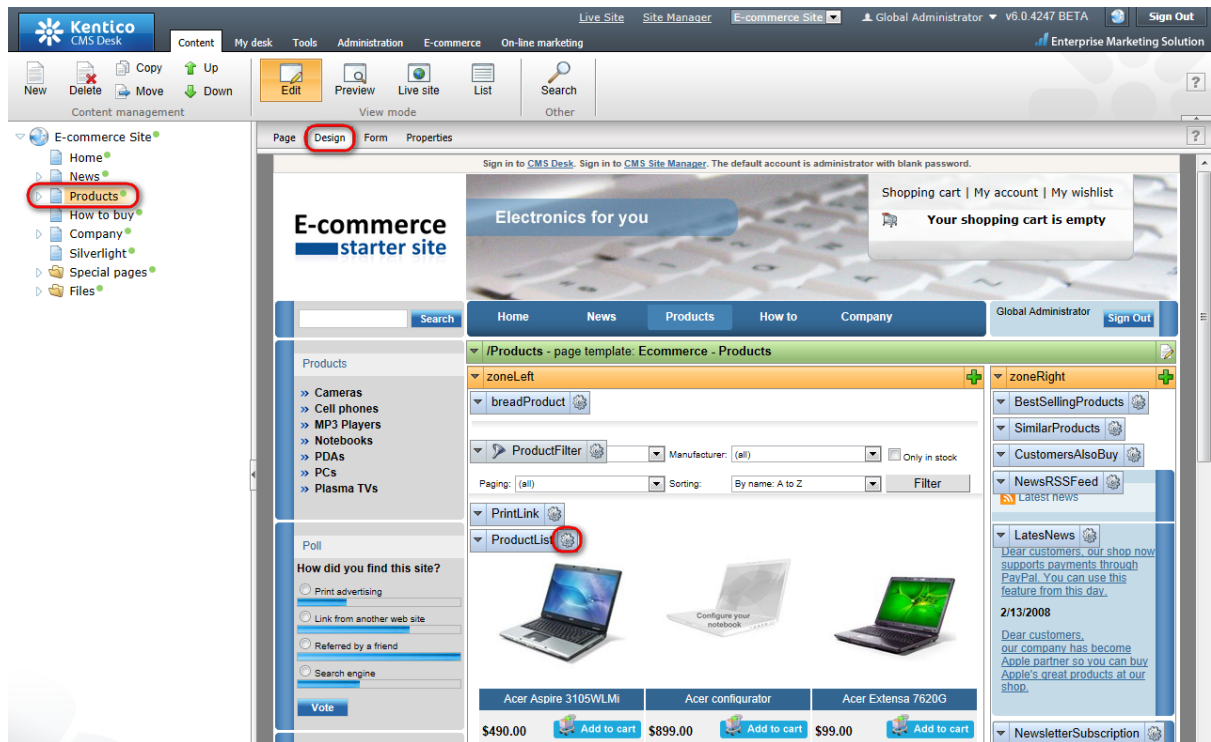
Your product has been added to the featured products.



On home page, by default there are displayed 4 products randomly chosen from all products with public status set to **Show as featured product**. Should you want set different filters for displaying products on you home page or should you need more information about the **Random products** web part, please refer to the [Random products](#) chapter.

## 2.7 Sorting and paging products

1. Go to **CMS Desk** -> **Products** and switch to the **Design** tab. Click the **Configure** button of the **Product list** web part.



2. In the webpage dialog, find the **Content filter** section and enter *SKUPrice ASC* into the **ORDER BY expression** text box, then find the **Paging** section and enter *16* into the **Page size** checkbox. Now click **OK**.

Web part properties (Datalist) - Windows Internet Explorer

**Web part properties (Datalist)** [Documentation](#)

General Layout

Combine with default culture: ☐ Yes ☐ No ☒ Use site settings

Culture code:

Maximum nesting level:

**ORDER BY expression:**

Select only published: ☒

Select top N documents:

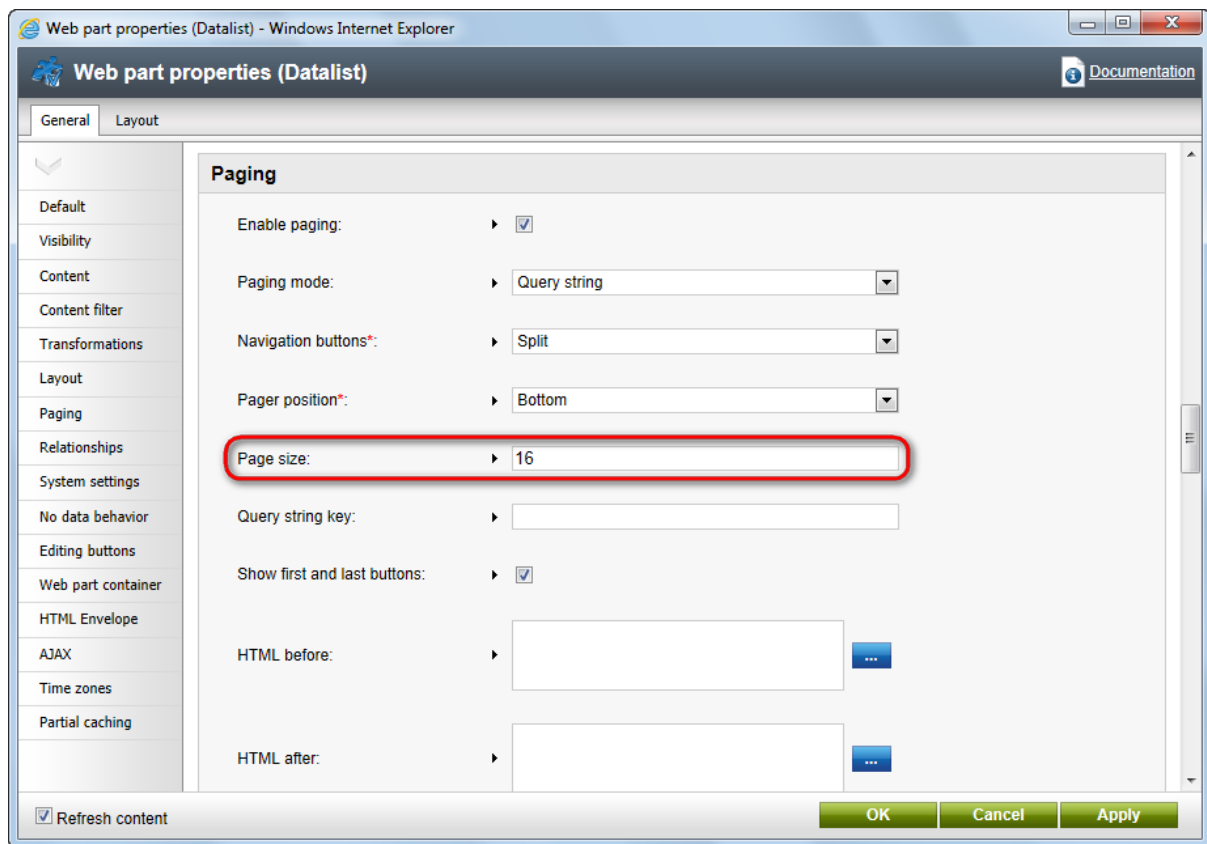
Site name:

WHERE condition:

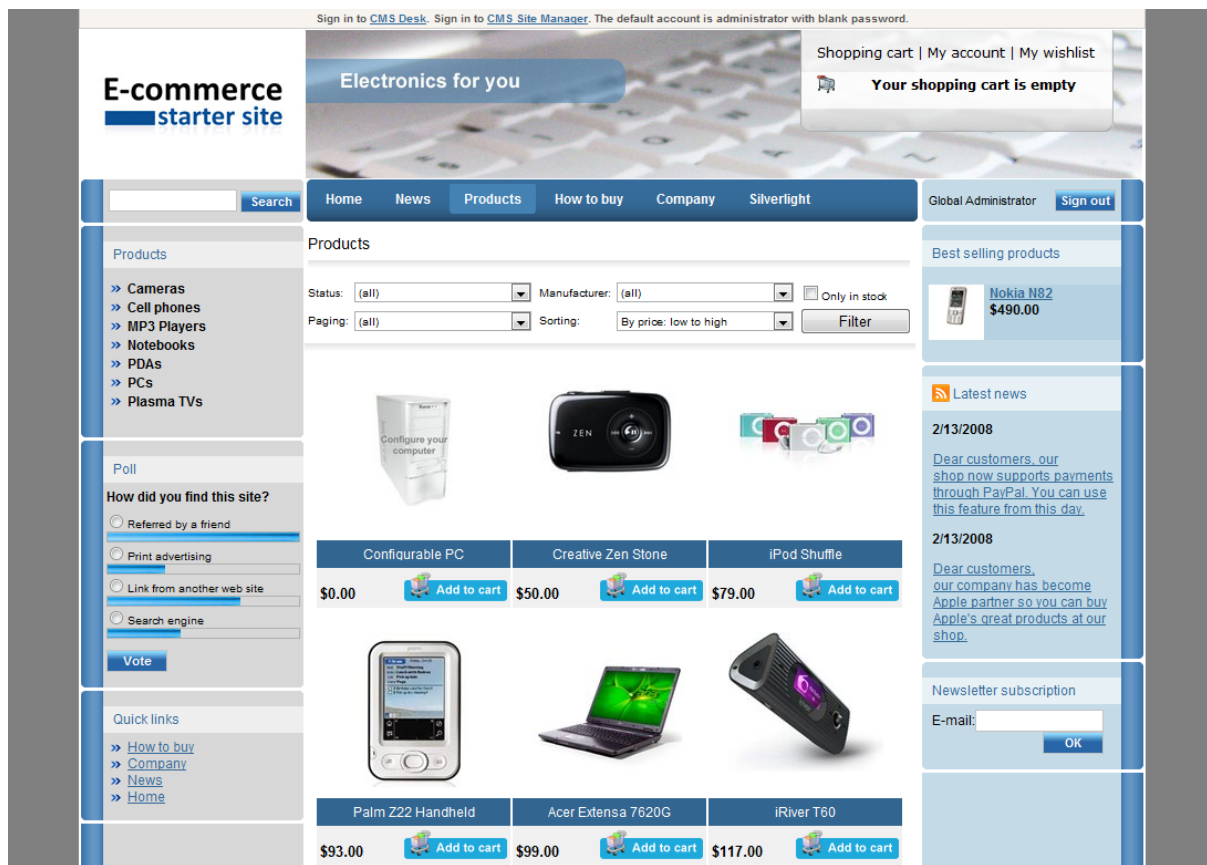
Columns:

Filter out duplicate documents: ☐

☒ Refresh content



3. Switch to the **Preview** viewing mode. You can see that the products are sorted from the cheapest to the most expensive and there are 16 of them.



### Sorting products

You may decide to sort products in a different way.

Should you want to sort products from the most expensive to the cheapest, you would enter *SKUPrice DESC* into the **ORDER BY expression** text box.

You can choose to sort products alphabetically as well. For sorting from A to Z, enter *NodeName ASC*. For sorting from Z to A, enter *NodeName DESC* into the **ORDER BY expression** text box.

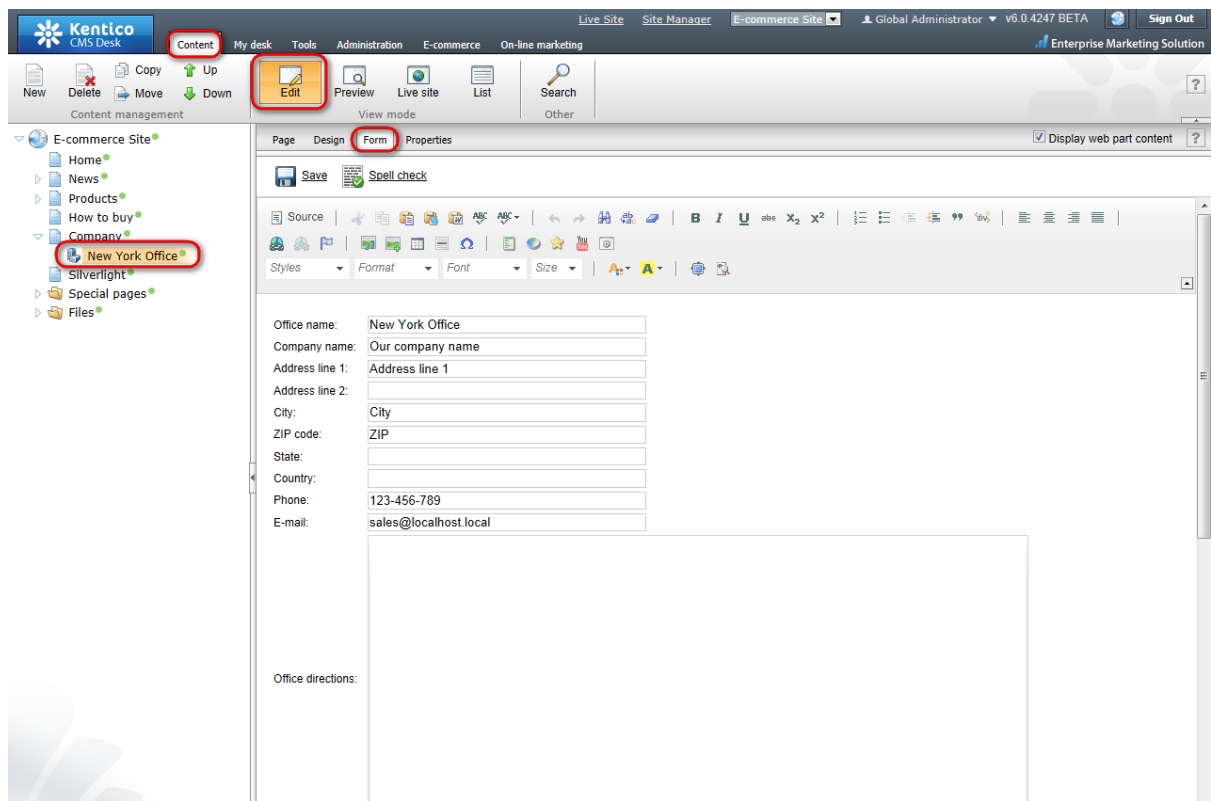
## 2.8 Changing company details

In this chapter, you will find out how to change your company details and add a new office.

### Changing company details

1. Go to **CMS Desk -> Content -> E-commerce Site -> Company -> New York Office** and switch to the **Form** tab.

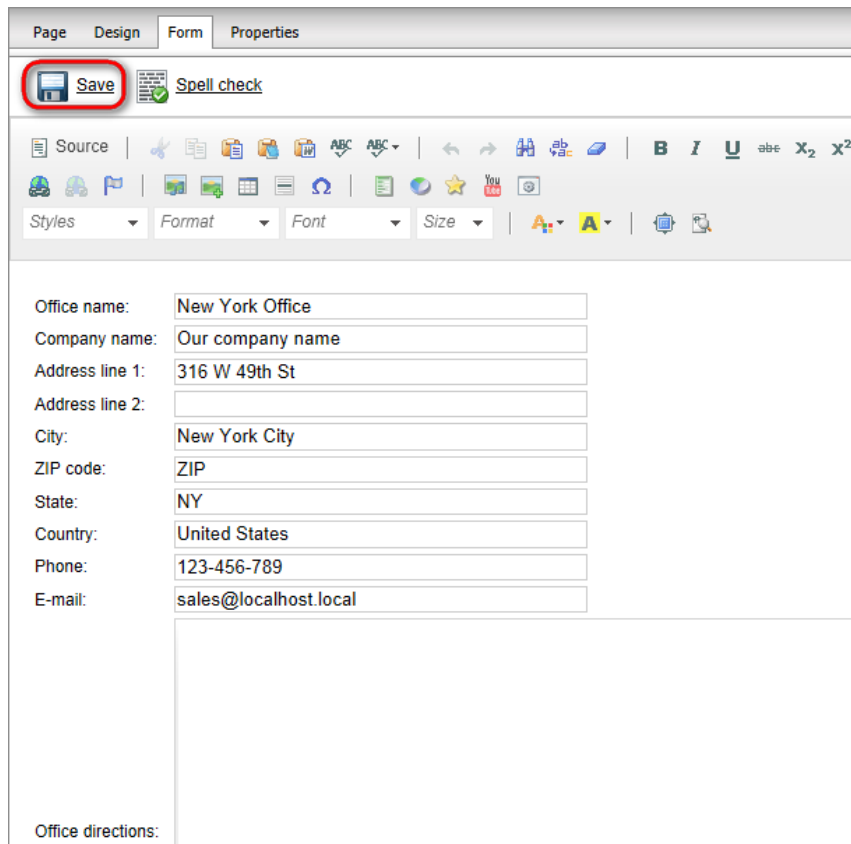





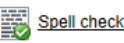
2. Enter your new company details:















- **Address line 1:** 316 W 49th St
- **City:** New York City
- **ZIP code:** 10019
- **State:** NY
- **Country:** United States


















































































































































































































































































































































































































































Then click **Save**.

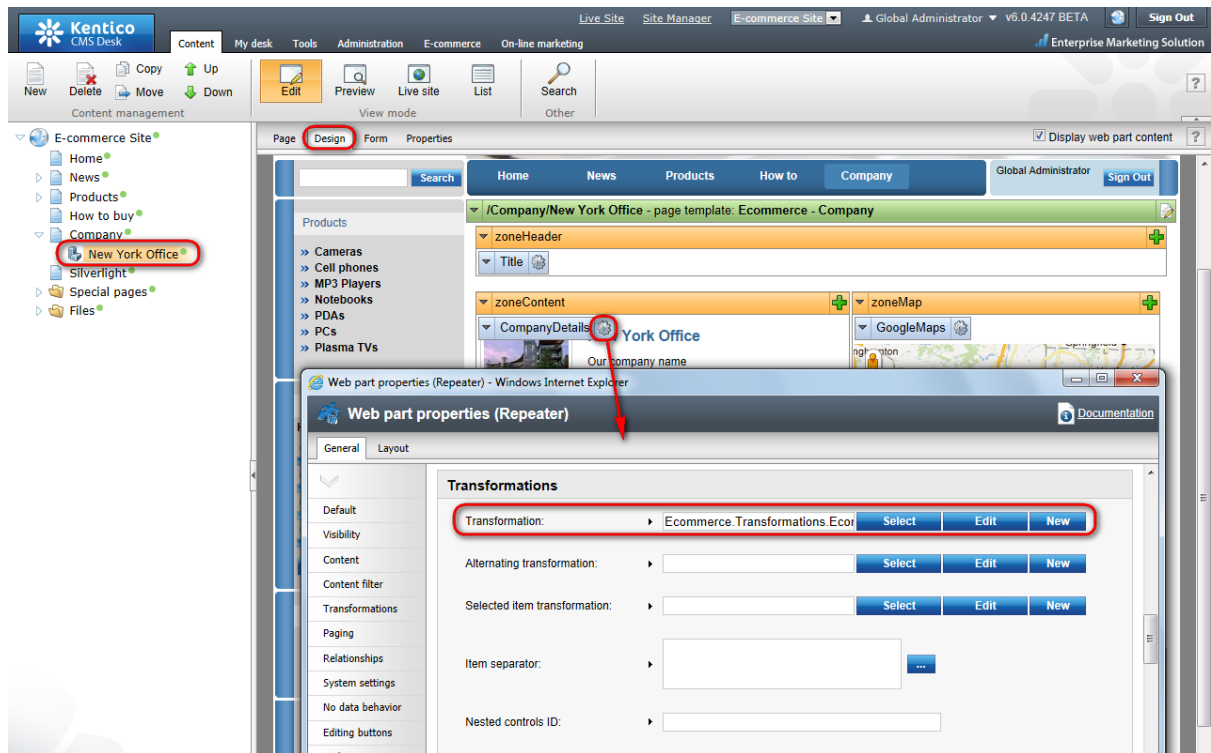


Page Design **Form** Properties

 Save  Spell check

Source |       |     | **B** *I* U    



                                                            

                                                            

                                                            

                                                            

                                                            

                                                            

                                                            

     





## Creating new office

1. Go to **CMS Desk -> Content -> E-commerce Site -> Company** and click the **New** button from the document action toolbar.
2. Choose **Office** as the new document type.
3. Now enter details for your new office:
  - **Office name:** San Francisco Office
  - **Address line 1:** 23 New Montgomery St
  - **City:** San Francisco
  - **ZIP code:** 94105
  - **State:** CA
  - **Country:** United States










Then scroll down a little bit and click **Browse** to select an image as **Office photo**. Then click **Save**.







 Save









 Save and create another








 Spell check





Source

**B** *I* U        

Styles Format Font Size    

Office name: San Francisco Office

Company name:

Address line 1: 23 New Montgomery St

Address line 2:

City: San Francisco

ZIP code: 94105

State: CA

Country: United States

Phone:





E-mail:

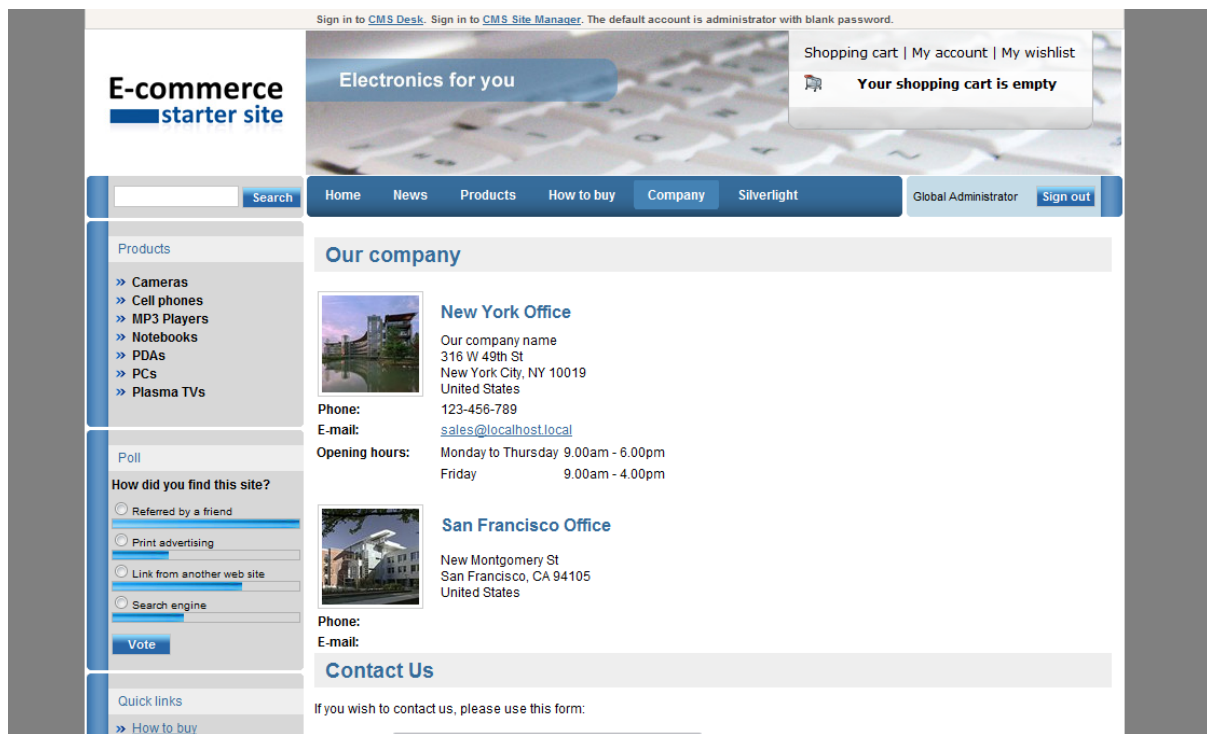
Office directions:

Latitude:

Longitude:

Office photo:

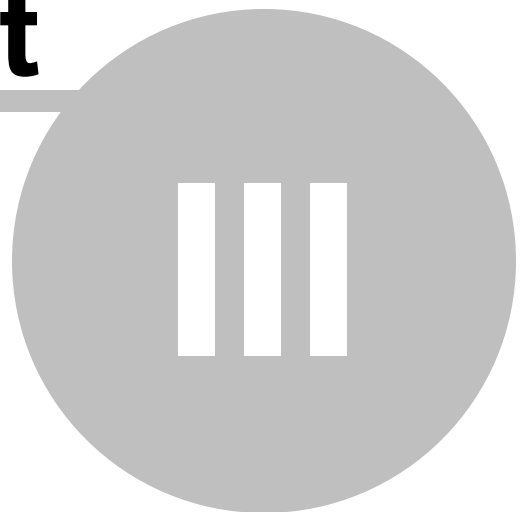
Actions	Update	Name	Size
 		 SToffice.png	293 kB



**Please note:** In the **Form** tab, you can use the **Office description** text box to enter additional information about your office (e.g. opening hours). Should you feel this text box is not convenient for you (for instance you want to edit displayed text in the **Page** tab), you can always leave Office description blank and add the Text -> Editable text web part on the page instead.

# Part

---



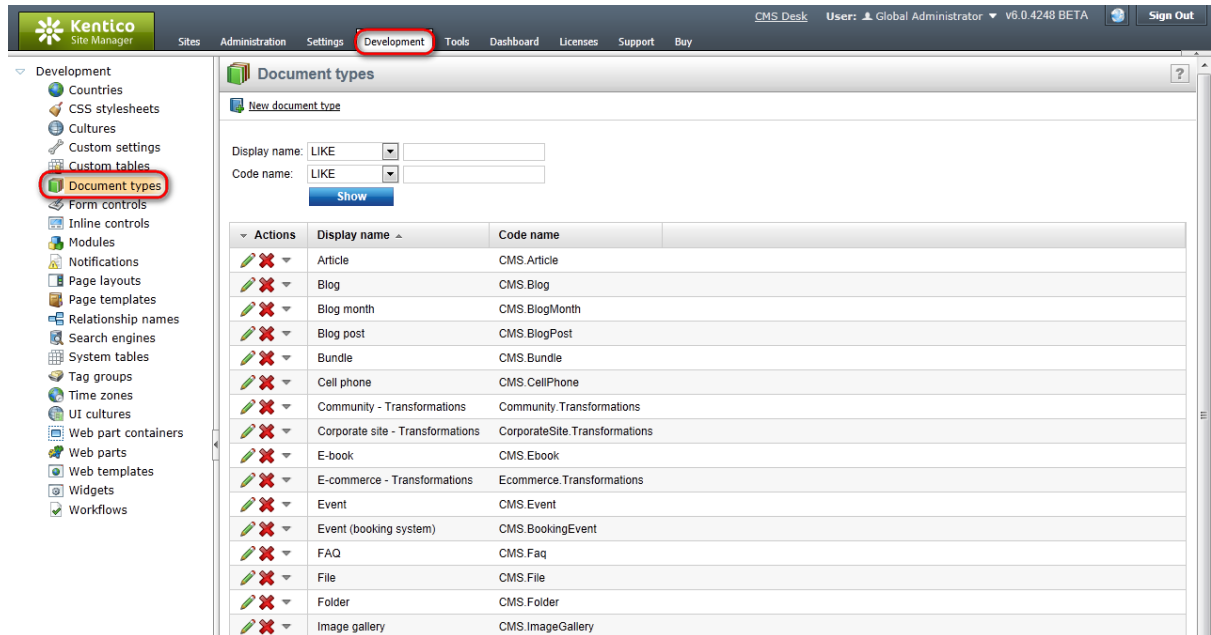
## Customizing the product fields

---

## 3 Customizing the product fields

### 3.1 Defining a new product type

1. Go to Site Manager -> Development -> Document Types.



2. Click the **New Document Type** link.

3. You will be redirected to the **New document type** wizard.

In the **Step 1**, enter:

- **Document type display name:** Plasma TV
- **Document type code name:** CustomProduct (**namespace**), television (**document type**)

Click **Next**.

**Step 1** | **General**  
Please enter document type display name (for users) and code name (it will be used in your code when necessary).

Document type display name:

Document type code name:    
namespace document type

[Next >](#)

4. In the **Step 2**, leave the default values and click **Next** again.

**Step 2** | **Data type**  
Please choose document data type. If you choose a document type with custom attributes you will also need to supply names of the new database table and its primary key.

☒ The document type has custom fields

Table name:

Primary key name:

Inherits fields from document type:

☐ The document type is only a container without custom fields

[Next >](#)

5. In the **Step 3**, you are asked to define fields for the new document type. Click **New Attribute (+)** and enter:

- **Column name:** TVName
- **Attribute type:** Text
- **Attribute size:** 100
- **Field caption:** Product name
- **Form control type:** Input
- **Form control:** Text box









Click  **Save field**.

**Step 3**


**Fields**  
Please define custom attributes of the document type and their appearance in the editing form. You can define attributes, such as product number, product weight, press release text, etc.

TelevisionID\*

TVname\*



**Quick links:**  
[Database](#)  
[Field appearance](#)  
[Editing control settings](#)  
[Validation](#)  
[CSS styles](#)

 **Save field**

The changes were saved.

**Database**  
Column name: TVname  
Attribute type: Text  
Attribute size: 100  
Allow empty value: ☐  
Default value:

☒ Display attribute in the editing form

**Field appearance**  
Field caption: Product name  
Form control type: Input  
Form control: Text box  
Field description:

Next >

6. Click **New Attribute**  again. Enter:

- **Column name:** TVScreenType
- **Attribute type:** Text
- **Attribute size:** 100
- **Field caption:** Screen type
- **Form control type:** Input
- **Form control:** Text box

Click  **Save field**.

### Step 3

#### Fields

Please define custom attributes of the document type and their appearance in the editing form. You can define attributes, such as product number, product weight, press release text, etc.

TelevisionID\*

TVname\*

TVScreenType\*

↑

↓

+

+

+

×

Save field

The changes were saved.

Database

Column name:

TVScreenType

Attribute type:

Text

Attribute size:

100

Allow empty value:

☐

Default value:

☒ Display attribute in the editing form

Field appearance

Field caption:

Screen type

Form control type:

Input

Form control:

Text box

Field description:

Quick links:

[Database](#)
[Field appearance](#)
[Editing control settings](#)
[Validation](#)
[CSS styles](#)

Next >

7. Now click **New Attribute** (+) again and enter information for the last field:

- **Column name:** TVScreenSize
- **Attribute type:** Integer number
- **Field caption:** Screen size in inches
- **Form control type:** Input
- **Form control:** Text box

Click  **Save field** and then **Next**.

**Step 3**

**Fields**  
Please define custom attributes of the document type and their appearance in the editing form. You can define attributes, such as product number, product weight, press release text, etc.

TelevisionID\*

TVname\*

TVScreenType\*

TVScreenSize\*

↑

↓

+

+

+

+

×

Quick links:

[Database](#)

[Field appearance](#)

[Editing control settings](#)

[Validation](#)

[CSS styles](#)

Save field

The changes were saved.

Database

Column name: TVScreenSize

Attribute type: Integer number

Attribute size:

Allow empty value: ☐

Default value:

☒ Display attribute in the editing form

Field appearance

Field caption: Screen size in inches

Form control type: Input

Form control: Text box

Field description:

Next >

8. In the **Step 4**, choose *TVName* in the **Document name source** drop-down list. Click **Next**.

**Step 4**

**Additional settings**  
Please choose the source field that will be used as a document name. You can choose either one of the custom fields or you can choose to use document name as a separate field.

Document name source: TVName

Next >

9. In the **Step 5**, click the **Add document types** button and select the **Page (menu item)** document type. Click **Next**.

**Step 5** | **Parent types**  
Please select document types under which this document template can be placed.

<input type="checkbox"/>	Document type name
<input type="checkbox"/>	Page (menu item) (CMS.Menuitem)

[Remove selected](#) [Add document types](#) ▼

[Next >](#)

10. Now make sure that you have assigned the document type to the Ecommerce site and click **Next**.

**Step 6** | **Sites**  
Please select sites where this document type can be used:

<input type="checkbox"/>	Site name
<input type="checkbox"/>	E-commerce Site

[Remove selected](#) [Add sites](#) ▼

[Next >](#)

11. In the **Step 7**, you are asked to specify how documents of this type will be indexed and displayed in the search results. For more information on these settings, please refer to the [Settings for particular object types](#) topic of Kentico CMS Developer's Guide. Make your choice and click **Next**.

**Step 7**

**Search options**  
Please set search fields for Smart search module.

Title field:

DocumentName

Content field:

DocumentContent

Image field:

(none)

Date field:

DocumentCreatedWhen

Set automatically

Field name	Content	Searchable	Tokenized	Custom search name
TelevisionID	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
TVname	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
TVScreenType	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
TVScreenSize	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	

Next >

12. The wizard has finished the configuration of the new document type. It has automatically created not only the database table, but also the SQL queries for SELECT, INSERT, UPDATE, DELETE operations and a default transformation.

Click **Finish**.

**Step 8**

**The wizard has finished**

The setup has finished the following steps:

- > The new document type was created.
- > The new editing form was created.
- > The document types were added among allowed child types of the new document type.
- > The sites were selected where this document type can be used.
- > The default queries were created.
- > The default ASCX transformations were created.
- > The default permission names were created.
- > The default icon was created.
- > Document smart search specification was created.

Finish

12. You will be redirected to the General tab of the product type editing interface. Enable the **Document is product type** option and click **OK**.

**Document type properties**

> Document types > Plasma TV



General Fields Form Transformations Queries Child types Sites Alternative forms Search fields Documents

Document type display name: Plasma TV

Document type code name: CustomProduct namespace television document type

Table name: CustomProduct\_television

Inherits fields from document type: (none)

Document type icons:  Upload ...  Upload ...

New page:

Editing page:

Editing form:

Preview page:

List page:

Use publish from/publish to: ☒

Show template selection: ☐

Default page template:  Select Clear

Behaves as Page (menu item) type: ☐

**Document is product type: ☒**

OK

13. Now go to **CMS Desk -> Products -> Plasma TV** and click **New** at the document action toolbar. You'll see that you can choose **Plasma TV** as the new document type.



### Adding attributes

This chapter explains how to add an attribute to specific document type. (e.g. product-camera, product - cell phones).

For adding attribute common to all products, please refer to the [Adding product custom fields](#) chapter.

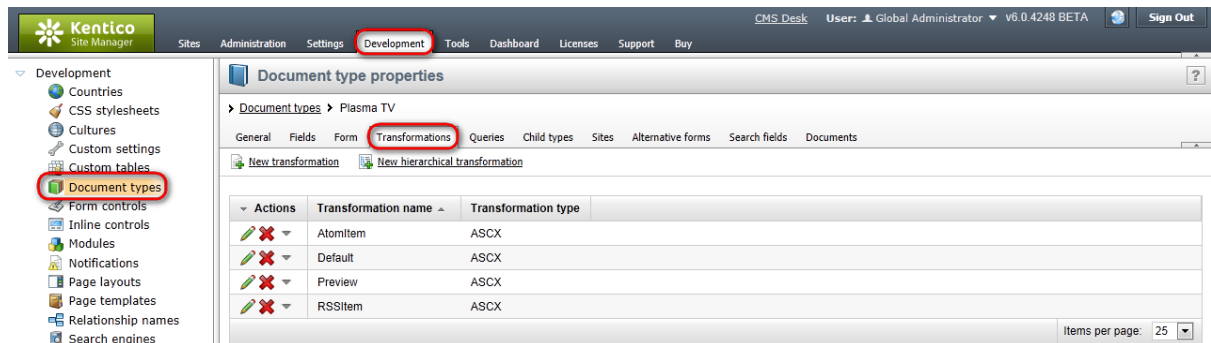
### Changing document type icon

The default icon is created with the new document type. Should you want to replace this icon with your own, go to **<your web project folder> \App\_Themes\Default\Images\CMSDesk\Icons\** and find the file named **<namespace>\_<your new document type>.gif** (in our example, this is going to be **CustomProduct\_television.gif**). All you have to do is replace this file with the image file of your new icon. Please note that the new file has to be named the same as the one being replaced.

## 3.2 Customizing product design

### Editing existing transformation

1. Go to **CMS Site Manager -> Development -> Document types**, choose the **Plasma TV** document type and switch to the **Transformation** tab.



2. Click the **Edit transformation** button next to the **Default** transformation name.

Actions	Transformation name	Transformation type
	AtomItem	ASCX
	Default	ASCX
	Preview	ASCX
	RSSItem	ASCX

3. Delete the content of the **Code** text box and insert the code from the gray box below. This code will change the transformation to display only the name of a TV and its screen type. Click **Save** at the top.

```
<table>
  <tr>
    <td>Product name:</td>
    <td><%# Eval( "TVName" ) %></td>
  </tr>
  <tr>
    <td>Screen type:</td>
    <td><%# Eval( "TVScreenType" ) %></td>
  </tr>
</table>
```

**Document type properties**

> Document types > Plasma TV

General Fields Form Transformations Queries Child types Sites Alternative forms Search fields Documents

> Transformations > Default

General Theme

**Save** Check out to file

You can check out the transformation to the c:\inetpub\wwwroot\KenticoCMS4248.10360Beta\CMSTransformations\custom\television\default.ascx file to edit it externally.

Transformation name: Default

Transformation type: ASCX

Code: Default **Generate default transformation**

```
<%@ Control Language="C#" AutoEventWireup="true" Inherits="CMS.Controls.CMSAbstractTransformation" %>
<%@ Register TagPrefix="cc1" Namespace="CMS.Controls" Assembly="CMS.Controls" %>

<table>
<tr>
<td>Product name:</td>
<td><%= Eval("TVName") %></td>
</tr>
<tr>
<td>Screen type:</td>
<td><%= Eval("TVScreenType") %></td>
</tr>
</table>
```

1 / 13

Add CSS styles

[Click here to view list of transformations examples.](#)

That's how you change default transformation for your product so that the product name and screen type are displayed.

Home News **Products** How to buy Company Silverlight

[Products](#) / [Plasma TVs](#) / LG plasma TV 42PC51

Product name: LG plasma TV 42PC51  
Screen type: Plasma

**Please note:** You can modify the design of the **product list** web part by changing the **CMS.Root.EcommerceProductsList** transformation in the manner similar to the given example.

### Adding namespaces

You can add additional namespaces into the **web.config** file in the following location:

```
<system.web>
  <pages validateRequest="false">
    <namespaces>
      <add namespace="CMS.CMSHelper" />
```



```
<add namespace="CMS.GlobalHelper" />
</namespaces>
</pages>
</system.web>
```

In case you add namespace into the **web.config** file, you don't have to specify it when calling its objects in transformations. Therefore, instead of calling **CMS.GlobalHelper.ResHelper.GetString("MyCustomString")**, you could call just **ResHelper.GetString("MyCustomString")**.



### Print page

Kentico CMS allows you to add a link button to your web page that will create print version of the given document. Please refer to the **Print page** chapter in **Kentico CMS Developer's Guide** for more information.

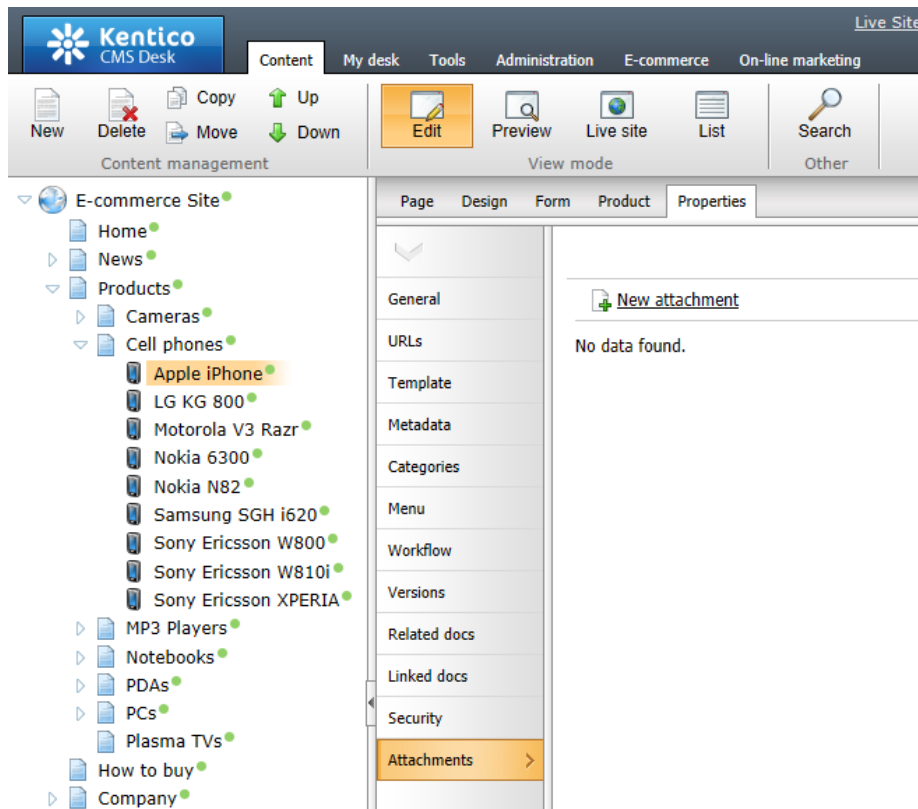
## 3.3 Adding images to product gallery

Now you will learn how to add images to product gallery.

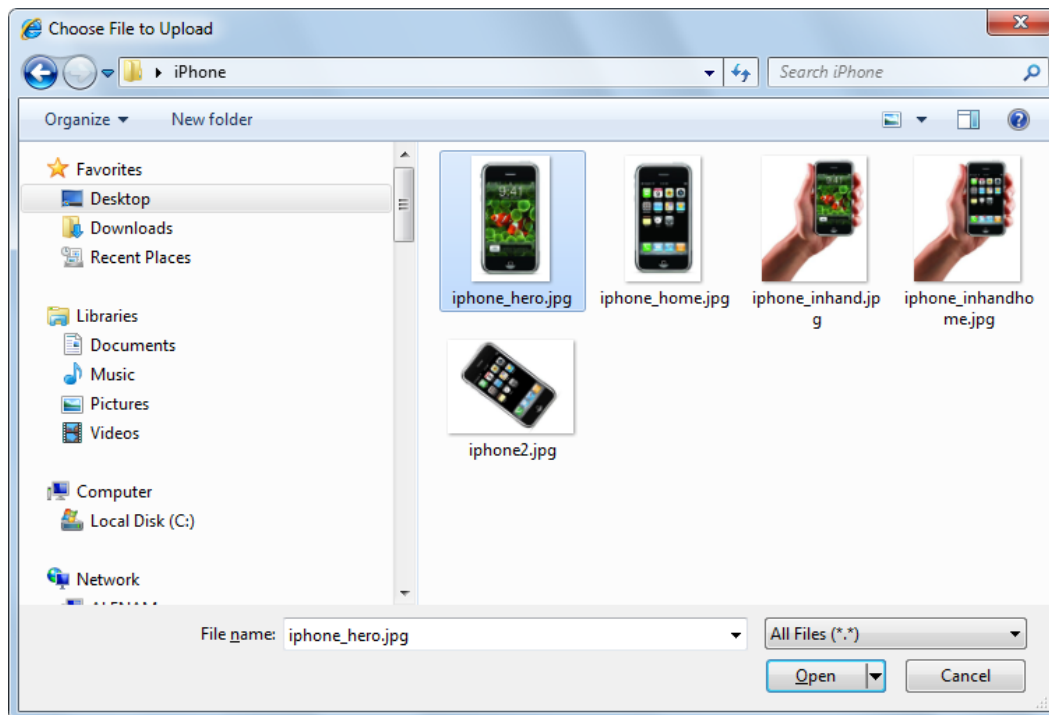
The images in the product gallery are displayed using the **Attachments image gallery** web part. Therefore, it is essential that you have placed this web part into your product page template.



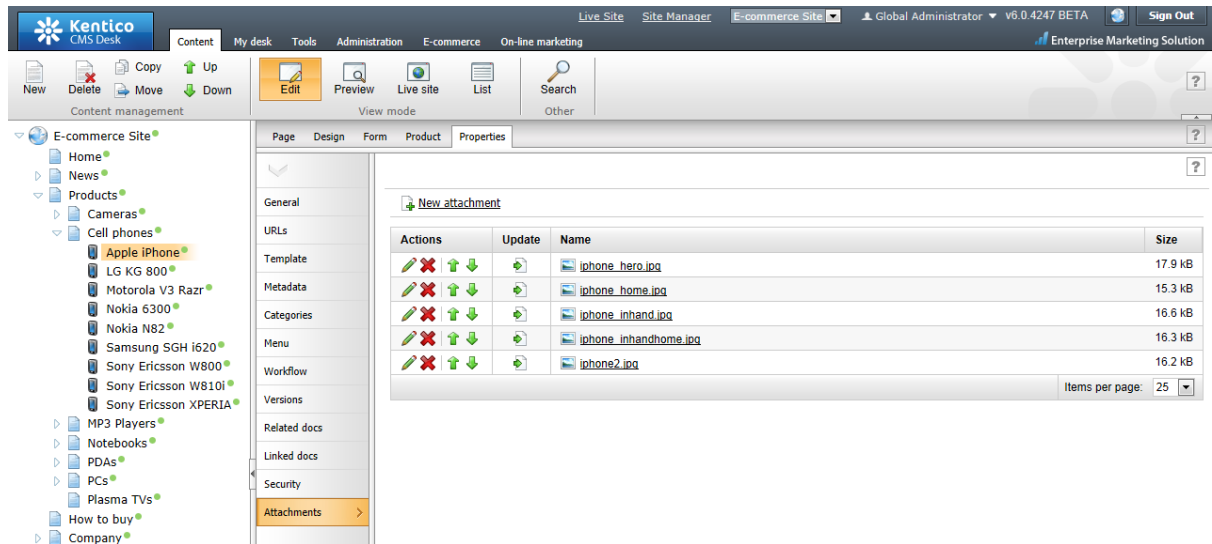
The images of a product are stored as attachments to the document. To add images to the product, go to **CMS Desk -> Content -> Edit -> Properties -> Attachments** tab and select the product you want to add the images to.



By clicking the **New attachment** link, a dialog for uploading files appears where you can add images. Just select the desired image and click **Open**. Repeat the procedure for all images you want to attach.

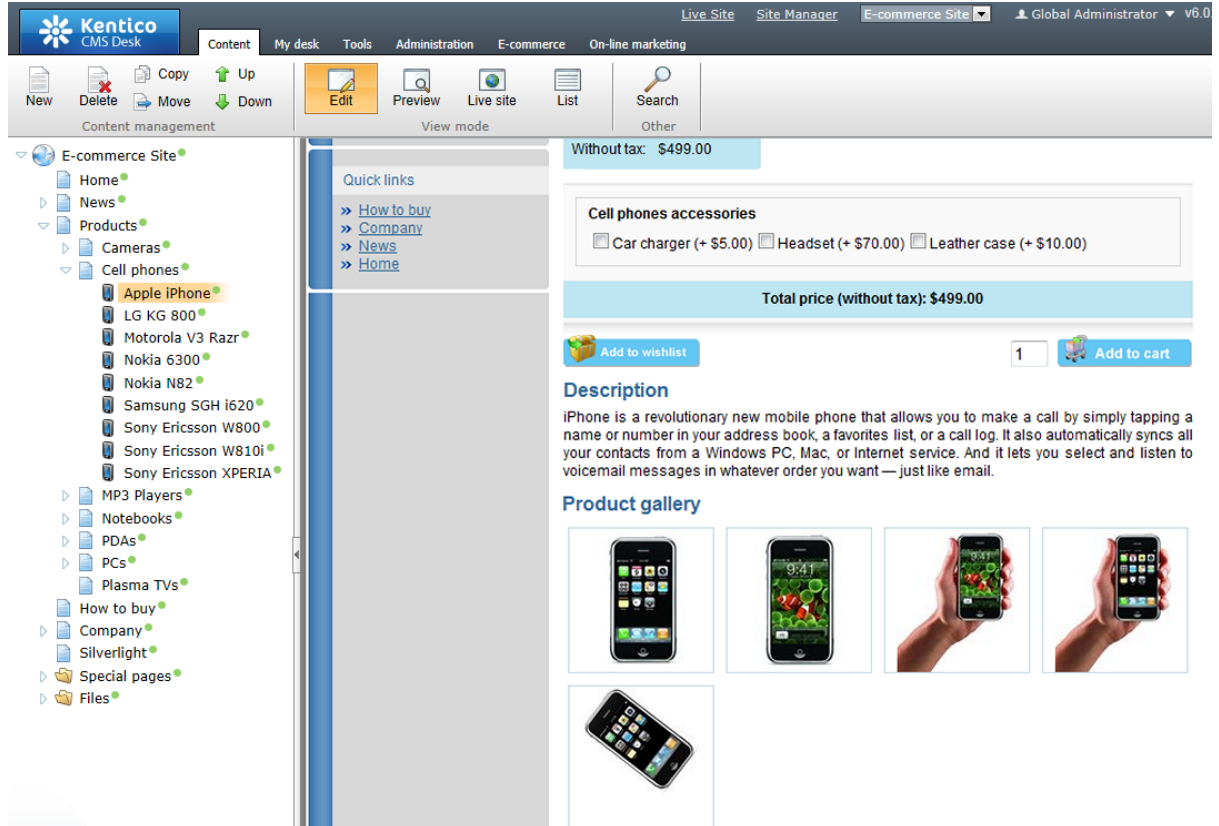


The images are stored as attachments of the product.



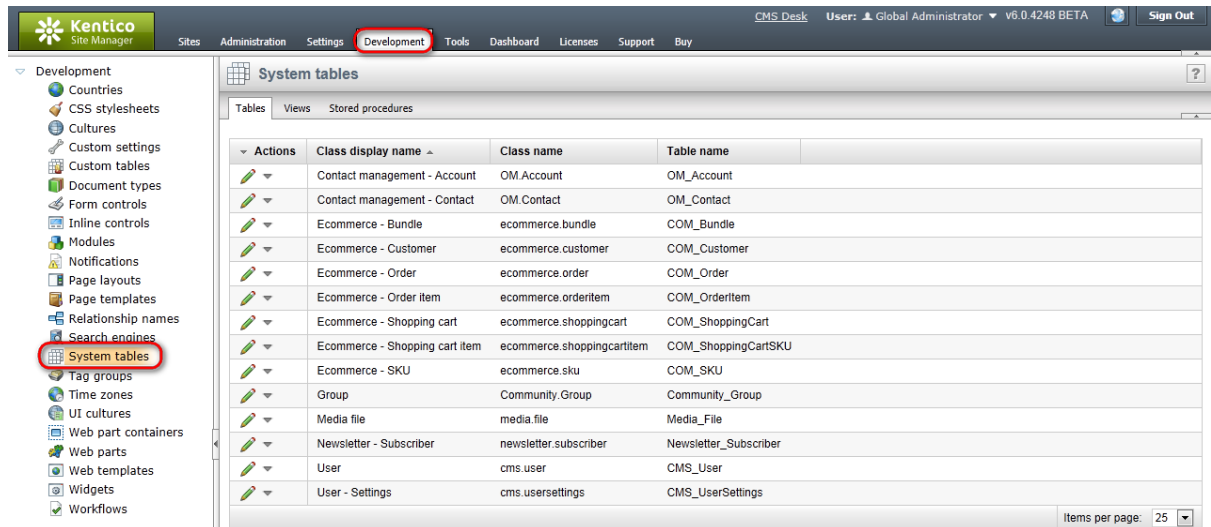
You can arrange the images in the attachment using the **Up** (↑) and **Down** (↓) icons. Images can be edited via the **Edit** (✎) icon, or deleted by clicking the **Delete** (✖) icon.

When the images are attached to the product, you can display them on the live site within the **Attachments image gallery** web part.



## 3.4 Adding product custom fields

1. Go to **CMS Site Manager -> Development -> System tables**.



2. Click **Edit** next to the **Ecommerce - SKU**.

Actions	Class display name	Class name	Table name
	Contact management - Account	OM.Account	OM_Account
	Contact management - Contact	OM.Contact	OM_Contact
	Ecommerce - Bundle	ecommerce.bundle	COM_Bundle
	Ecommerce - Customer	ecommerce.customer	COM_Customer
	Ecommerce - Order	ecommerce.order	COM_Order
	Ecommerce - Order item	ecommerce.orderitem	COM_OrderItem
	Ecommerce - Shopping cart	ecommerce.shoppingcart	COM_ShoppingCart
	Ecommerce - Shopping cart item	ecommerce.shoppingcartitem	COM_ShoppingCartSKU
	Ecommerce - SKU	ecommerce.sku	COM_SKU
	Group	Community.Group	Community_Group
	Media file	media.file	Media_File
	Newsletter - Subscriber	newsletter.subscriber	Newsletter_Subscriber
	User	cms.user	CMS_User
	User - Settings	cms.usersettings	CMS_UserSettings

3. Click **New Attribute** (+) on the **Fields** tab.

4. Enter *SKUColor* into the **Column name** property. Choose **Text** as the **Attribute type**, enter *100* as the **Attribute size** and *Product color* as the **Field Caption**. Choose *Text box* as the used **Form control**. Click **Save field**.

**System tables**

Tables Views Stored procedures

System tables > Ecommerce - SKU

Fields Queries Alternative forms Search fields

**SKUColor**

SKUNumber  
SKUName\*  
SKUDescription  
SKUPrice\*  
SKUEnabled\*  
SKUDepartmentID  
SKUManufacturerID  
SKUInternalStatusID  
SKUPublicStatusID  
SKUSupplierID  
SKUAvailableInDays  
SKUGUID\*  
SKUImagePath  
SKUWeight  
SKUWidth  
SKUDepth  
SKUHeight  
SKUAvailableItems  
SKUSellOnlyAvailable  
SKUCustomData  
SKUOptionCategoryID  
SKUOrder  
SKULastModified\*  
SKUCreated

**Quick links:**  
[Database](#)  
[Field appearance](#)  
[Editing control settings](#)  
[Validation](#)  
[CSS styles](#)

**Save field**

The changes were saved.

**Database**

Column name: SKUColor  
Attribute type: Text  
Attribute size: 100  
Allow empty value: ☒  
Default value:

☒ Display attribute in the editing form

**Field appearance**

Field caption: Product color  
Form control type: Input  
Form control: Text box  
Field description:

Has depending fields: ☐  
Depends on another field: ☐




5. Now switch to **CMS Desk** and go to **Tools -> E-commerce -> Products**. Click **Edit** next to **Acer Aspire 3105WLMi**.

**Products**

[New product](#)

Product name:   
Product type: (all)  
Department: (all my departments)  
**Show**

Display advanced filter

Actions	Product name	Product number	Product price
	Acer Aspire 3105WLMi		\$490.00
	Acer configurator		\$899.00
	Acer Extensa 7620G		\$99.00

6. Switch to the **Custom fields** tab and enter *Blue* into the **Product color** text box. Then click **OK**.

The screenshot shows the Kentico CMS Desk interface. The top navigation bar includes 'Content', 'My desk', 'Tools', 'Administration', 'E-commerce', and 'On-line marketing'. The 'E-commerce' section is active, showing 'Orders', 'Customers', 'Products', and 'Discounts'. The 'Product properties' section for 'Acer Aspire 3105WLMi' is displayed. The 'Custom fields' tab is selected, and the 'Product color' field is set to 'Blue'. An 'OK' button is visible below the field.

That's how you create a custom field for all products and how you set its value for specific product.

## Product custom fields in transformations

For displaying values of the given custom field in transformation, enter *Eval("<custom field name>")*. For instance, for the example given in this chapter, you would enter *Eval("SKUColor")* (please see the [Customizing product design](#) chapter for more details).

In versions prior to 4.0, you had to update the following database views manually with the name of the new custom field in order for the *Eval("<custom field name>")* to work correctly. Now it is done **automatically** so that you **don't need to modify the database** at all.

1. View\_COM\_SKU
2. View\_CMS\_Tree\_Joined
3. View\_CMS\_Tree\_Joined\_Versions

## Product custom fields in code

In code, you can get and set the given value in the following way:

[C#]

```
using System;
using CMS.Ecommerce;

// How to set value of the custom field
SKUInfo product = new SKUInfo();
product.SetValue("SKUColor", "green");

// How to get value of the custom field
string color = Convert.ToString(product.GetValue("SKUColor"));
```

# Part

---



IV

**Customizing web site design**

---

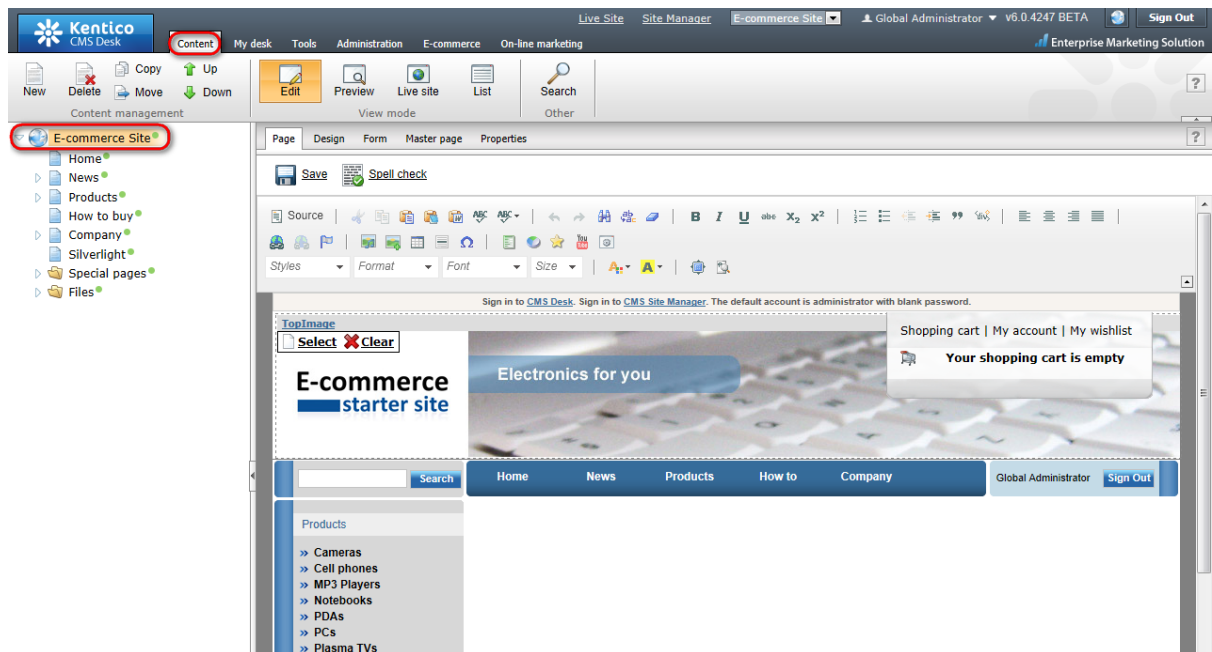
## **4 Customizing web site design**

### **4.1 Changing the header and menu (master page)**

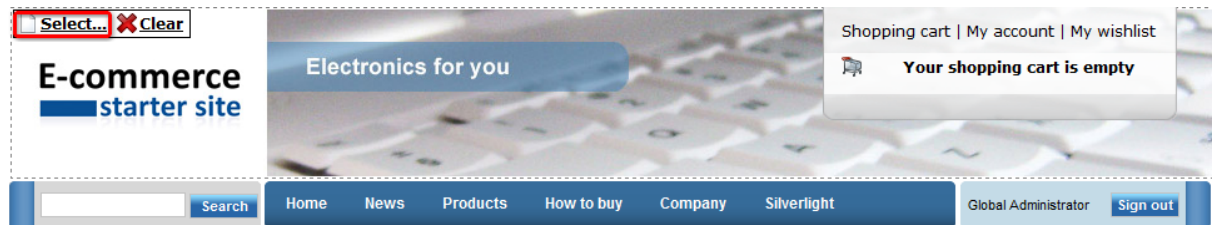
Changing a logo in the header



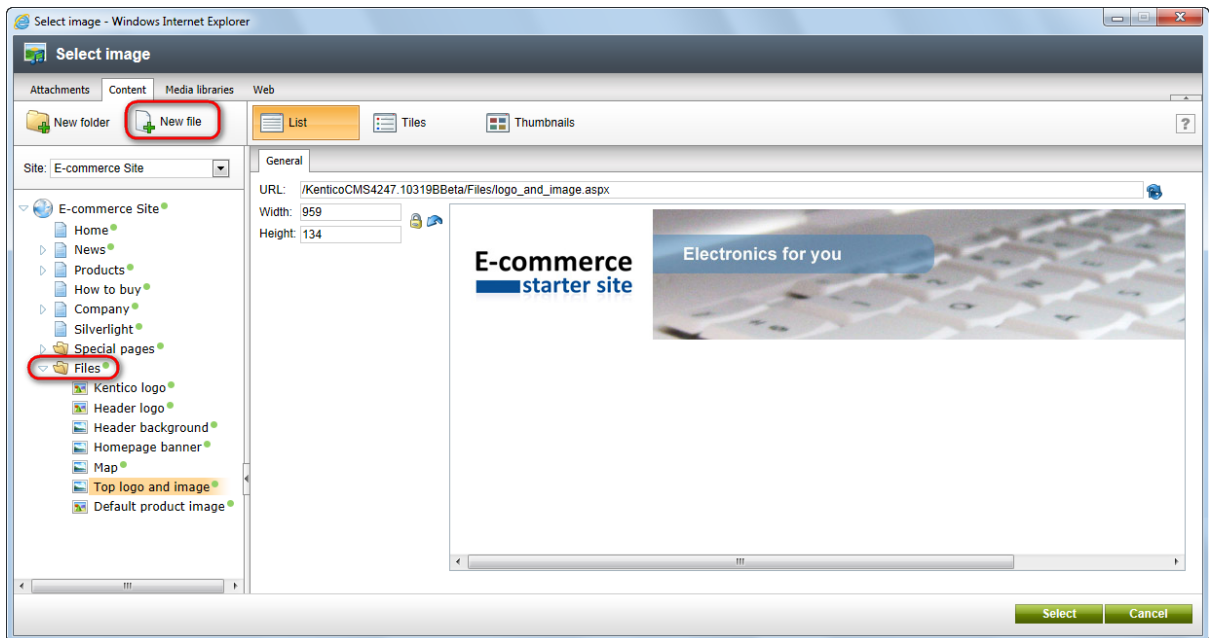
1. Go to **CMS Desk -> Content -> Ecommerce site.**



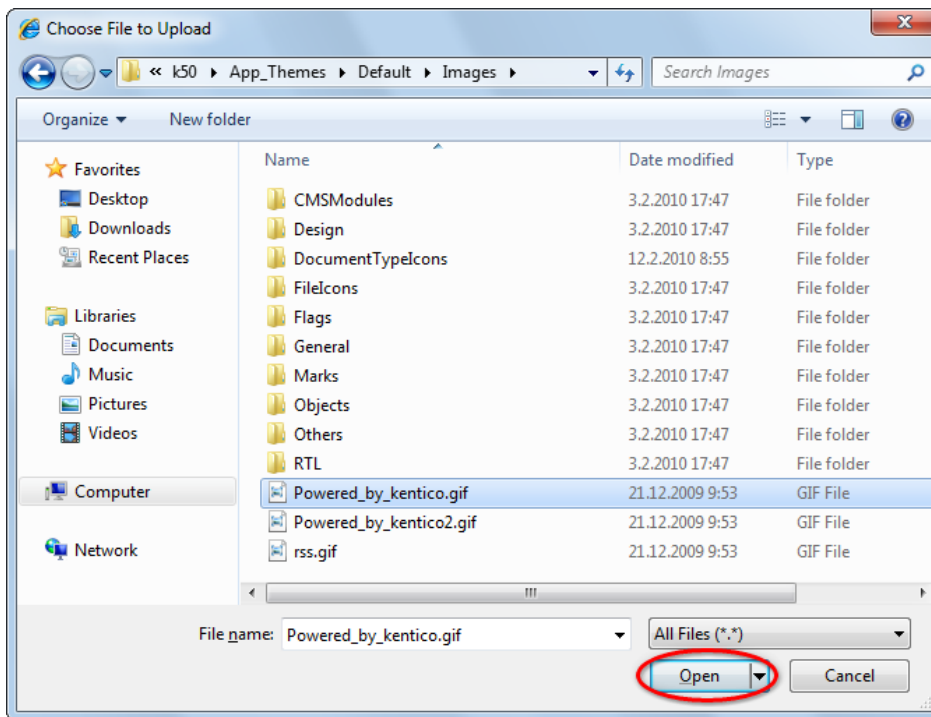
2. Click **Select ...** above the logo.



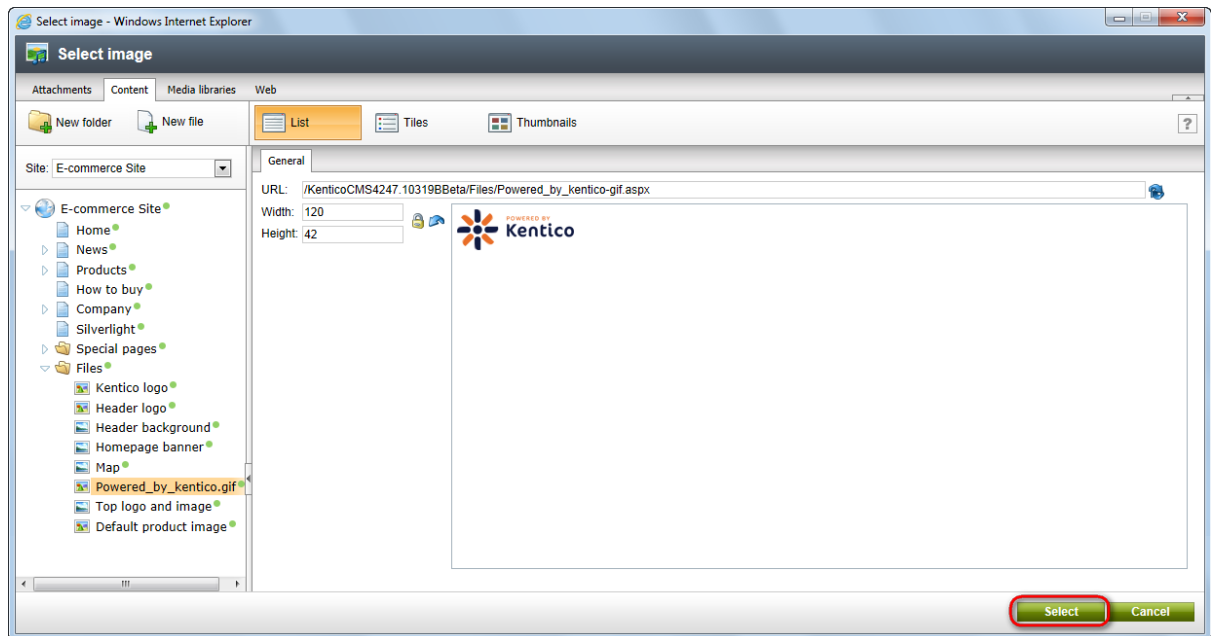
3. In the web page dialog, select **Files** in the content tree and click the **New file** button at the top-left.



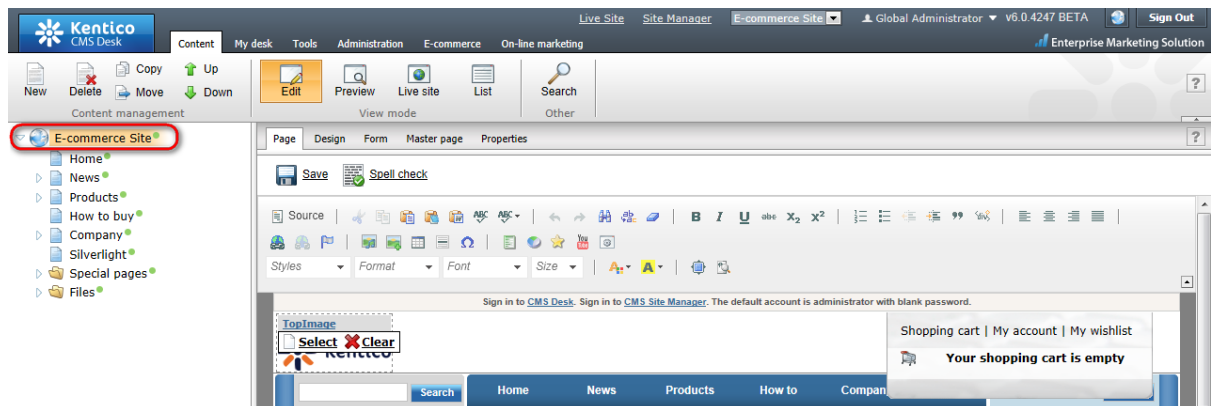
4. Find the image you want to upload and click **Open**.



5. Click **Select**.



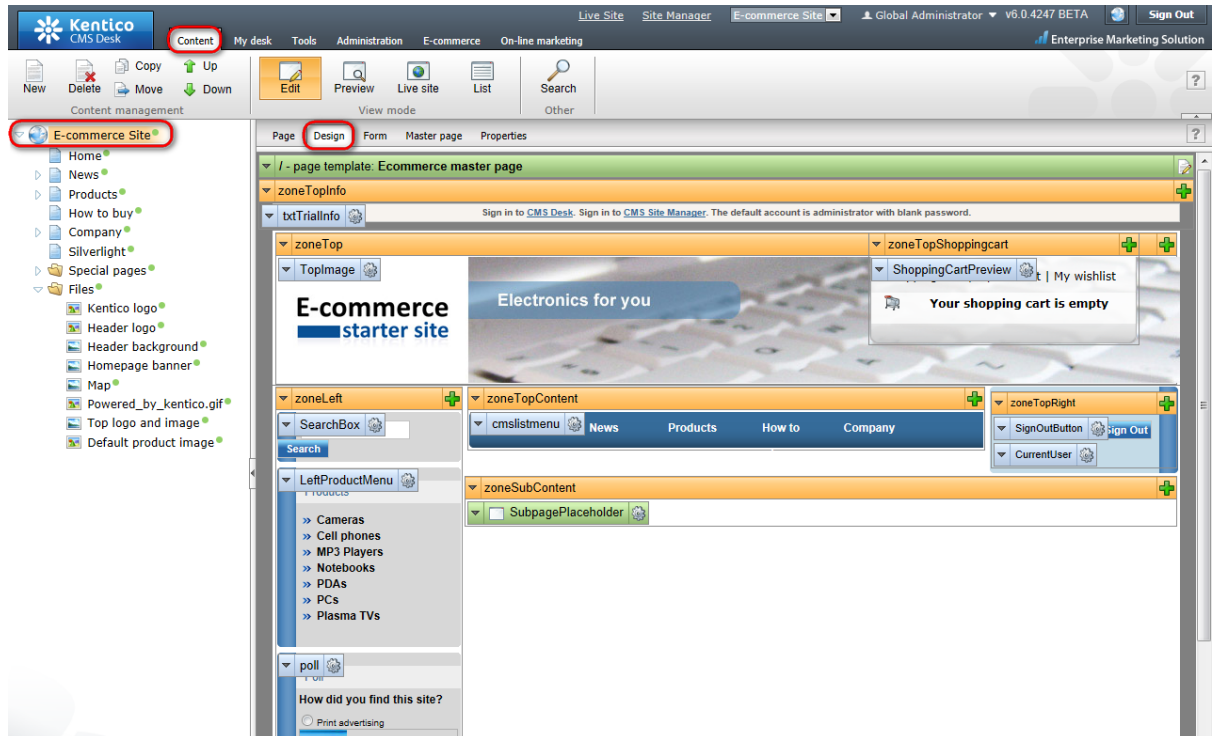
6. Now click **Save**.



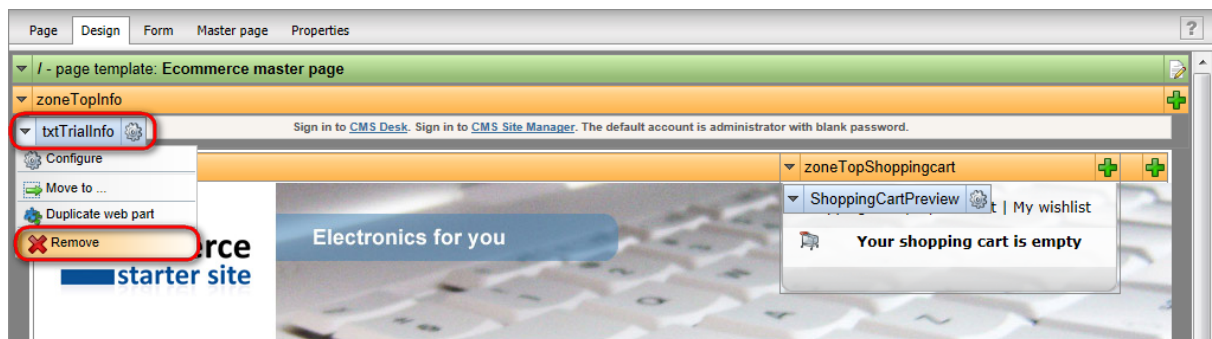
You've just publish a new logo on your website.

## Removing the log-on bar

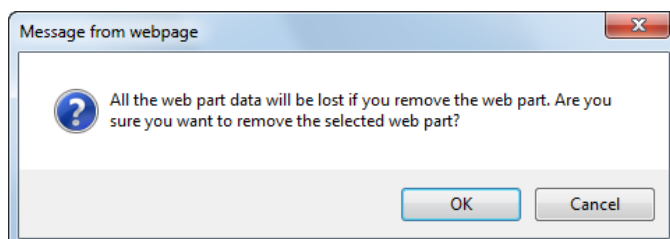
1. Go to **CMS Desk** -> **Content** -> **Ecommerce site** and switch to the **Design** tab.



2. Right-clicked the **txtTrialInfo** webpart at the top-left and choose **Remove**.



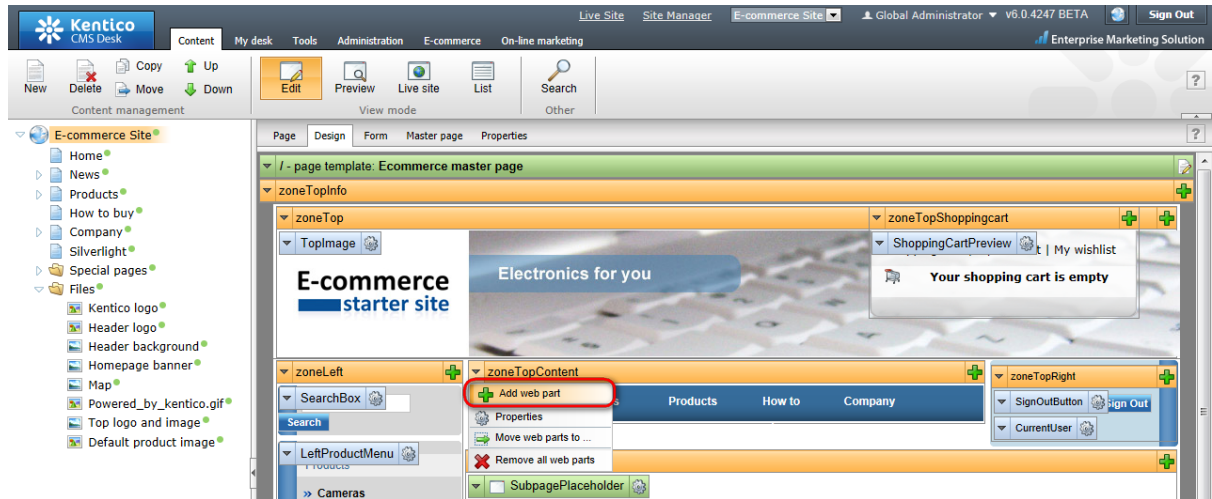
3. Click **OK** to remove the bar from your website.



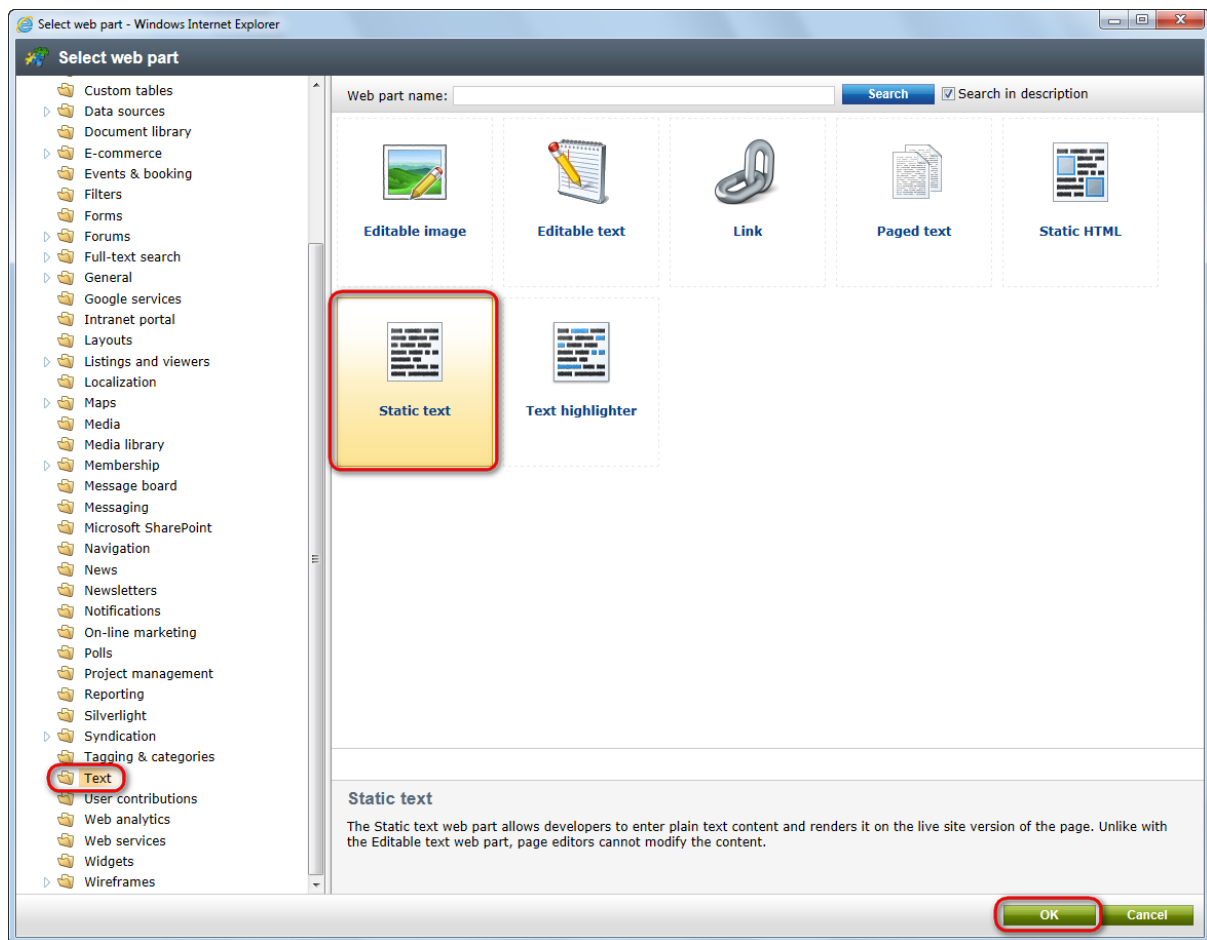
That's how you remove the signing-in bar from your website.

## Changing content of the menu

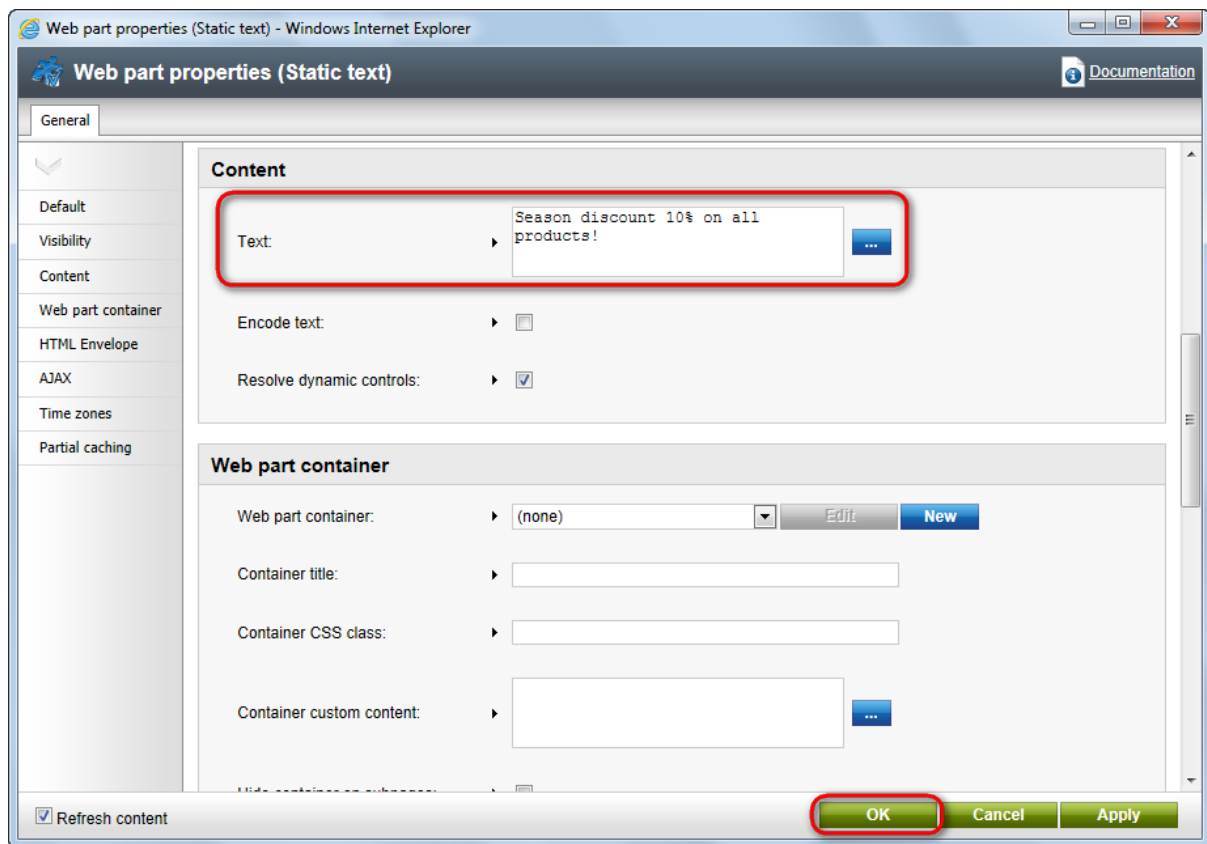
1. Go to **CMS Desk -> Content -> Ecommerce site** and switch to the **Design** tab.
2. Click the **Add web part** button from the drop-down menu in **zoneTopContent**.



3. In the web page dialog, choose **Text -> Static text** and click **OK**.



4. Enter *Season discount 10% on all products!* as **Text** and click **OK**.



5. Now switch to the **Live site** viewing mode. You can see the new text below the main menu.



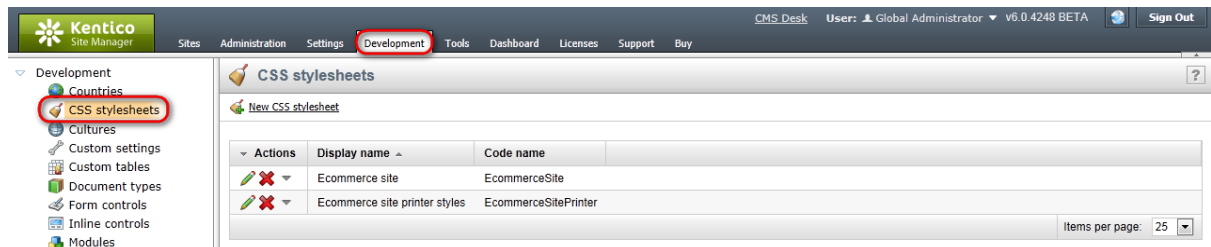


### More information





Should you need more information about customizing your website design, please refer to **Kentico CMS Developer's Guide**.

## 4.2 Changing colors using CSS styles

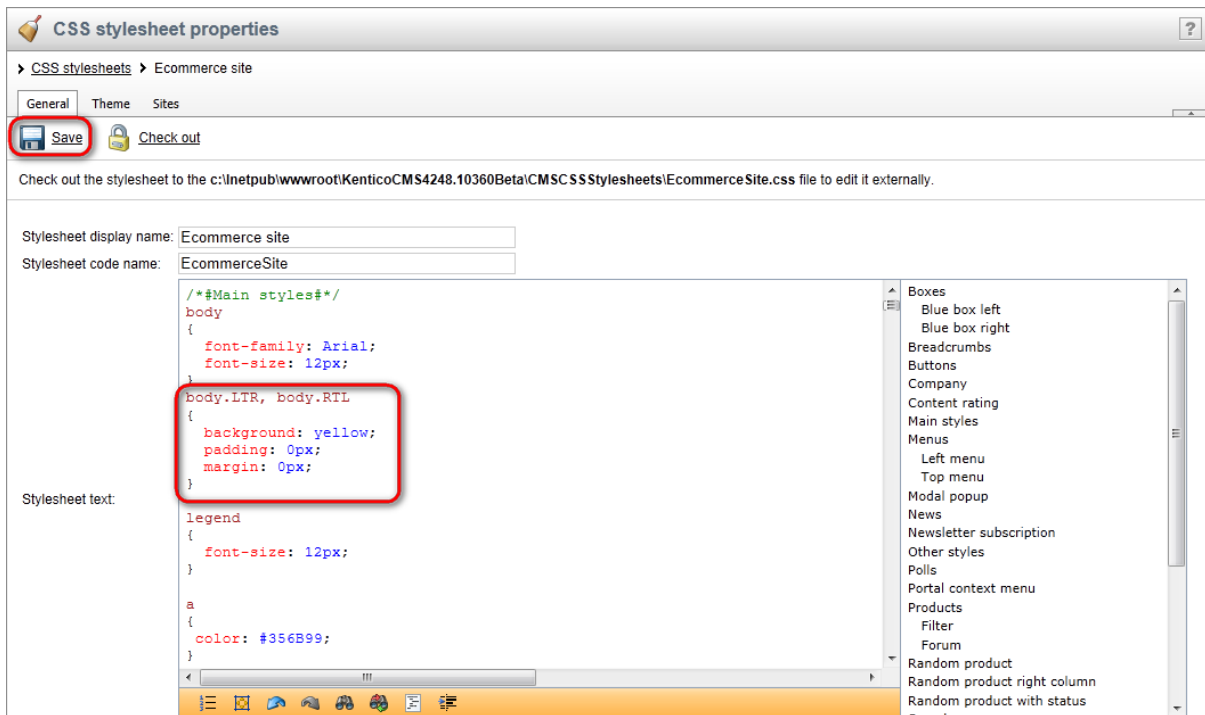
1. Go to **CMS Site Manager -> Development -> CSS stylesheets**.



2. Click the **Edit** button next to the **Ecommerce site**.

Actions	Display name ^	Code name
 	Ecommerce site	EcommerceSite
 	Ecommerce site printer styles	EcommerceSitePrinter



3. In the **Stylesheet text** text box change background color of **body.LTR**, **body.RTL** from gray to yellow. Click **Save** at the top.



**CSS stylesheet properties**

> CSS stylesheets > Ecommerce site

General Theme Sites

 **Save**  Check out

Check out the stylesheet to the c:\inetpub\wwwroot\KenticoCMS4248.10360Beta\CMSCSSStylesheets\EcommerceSite.css file to edit it externally.

Stylesheet display name: Ecommerce site

Stylesheet code name: EcommerceSite

Stylesheet text:

```

/*Main styles*/
body
{
    font-family: Arial;
    font-size: 12px;
}

body.LTR, body.RTL
{
    background: yellow;
    padding: 0px;
    margin: 0px;
}

legend
{
    font-size: 12px;
}

a
{
    color: #356B99;
}

```

Boxes

- Blue box left
- Blue box right

Breadcrumbs

Buttons

Company

Content rating

Main styles

Menus

- Left menu
- Top menu

Modal popup

News

Newsletter subscription

Other styles

Polls

Portal context menu

Products

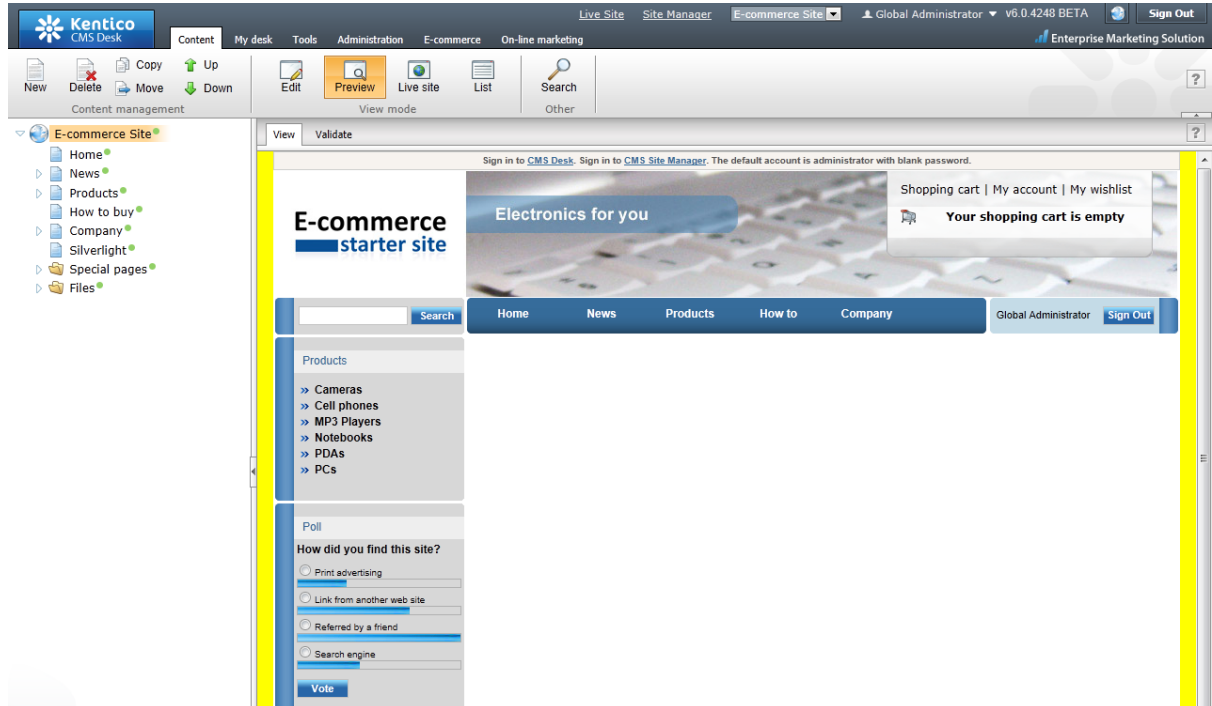
- Filter
- Forum

Random product

Random product right column

Random product with status

4. Go to **CMS Desk** and switch to the **Preview** mode. You can see that the background of the E-commerce website is yellow.



# Part

---



V

**Integration with your existing Kentico CMS website**

## 5 Integration with your existing Kentico CMS website

### 5.1 Overview

The e-commerce module provides several web parts that allow you to integrate it into your website:

- **Shopping cart** - displays the shopping cart content and ensures the check-out process. For more info please see the [Shopping cart](#) chapter.
- **Shopping cart preview** - displays the links to the shopping cart, to my account and to my wishlist and the total value of the shopping cart content. For more info please see the [Shopping cart preview](#) chapter.
- **My account** - displays the details of the current user, such as personal settings, addresses, orders, invoices and allows user to change the password. For more info please see the [My account](#) chapter.
- **Wishlist** - displays the wishlist of the currently logged-on user. For more info please see the [Wishlist](#) chapter.
- **Similar products by sale** - displays products which were purchased together with the current product by previous customers. For more info please see the [Similar products by sale](#) chapter.
- **Random products** - displays random N products that correspond to the criteria specify in the content filter. For more info please see the [Random products](#) chapter.
- **Product datalist** - display products based on their e-commerce product (SKU) properties instead of displaying the standard CMS documents.
- **Top N products by sales** - displays N best-selling products. For more info please see the [Top N products by sales](#) chapter.
- **Top N newest products** - displays N newest products. For more info please see the [Top N newest products](#) chapter.

These web parts can be used also as independent user controls placed on the ASPX pages.

### How to organize the pages

A typical content tree structure of an e-commerce-enabled website looks like this:

- **Home** (page)
- **Products** (page)
  - **Category 1** (page; contains Repeater or DataList web parts displaying products with links to shopping cart)
    - **Product 1** (product specification document; it's marked as a product)
    - **Product 2** (product specification document; it's marked as a product)
  - **Category 2** (page; contains Repeater or DataList web parts displaying products with links to shopping cart)
    - **Sub category 1** (page; contains Repeater or DataList web parts displaying products with links to shopping cart)
      - **Product 3** (product specification document; it's marked as a product)
      - **Product 4** (product specification document; it's marked as a product)
- **News** (page)
- **Special pages**
  - **Shopping Cart** (page; it contains the Shopping cart web part and can be configured to be hidden in the navigation and site map)
  - **My Account** (page; it contains the My account web part and can be configured to be hidden in the navigation and site map)

## 5.2 Web parts

### 5.2.1 My account

The **My account** web part displays various information about the current user's account, e.g. user's personal settings, addresses, orders, memberships etc., and allows its settings to be managed. This web part is only visible for authenticated users.

The screenshot shows the 'My Account' web part in the Kentico CMS 6.0 E-commerce Starter Site. The page has a header with the site logo 'E-commerce starter site' and a navigation menu. The 'My Account' section is highlighted with a red border. It contains a sub-menu with 'Details', 'Addresses', 'Orders', 'Credit', 'Change password', and 'Subscriptions'. The 'Details' tab is active, showing a form for user information. The form includes fields for First name (Global), Last name (Administrator), Company account (checkbox), E-mail (administrator@localhost.local), Phone, Fax, Preferred currency (U.S. Dollar), Preferred shipping option (none), Preferred payment method (none), and Country/state (none). An 'OK' button is at the bottom of the form. The left sidebar contains a 'Products' list, a 'Poll' section, and 'Quick links'. The right sidebar shows a shopping cart status and a wishlist link.

Sign in to [CMS Desk](#). Sign in to [CMS Site Manager](#). The default account is administrator with blank password.

Shopping cart **My account** My wishlist  
Your shopping cart is empty

**E-commerce starter site**

Electronics for you

Home News Products How to buy Company Silverlight Global Administrator Sign out

**My Account**

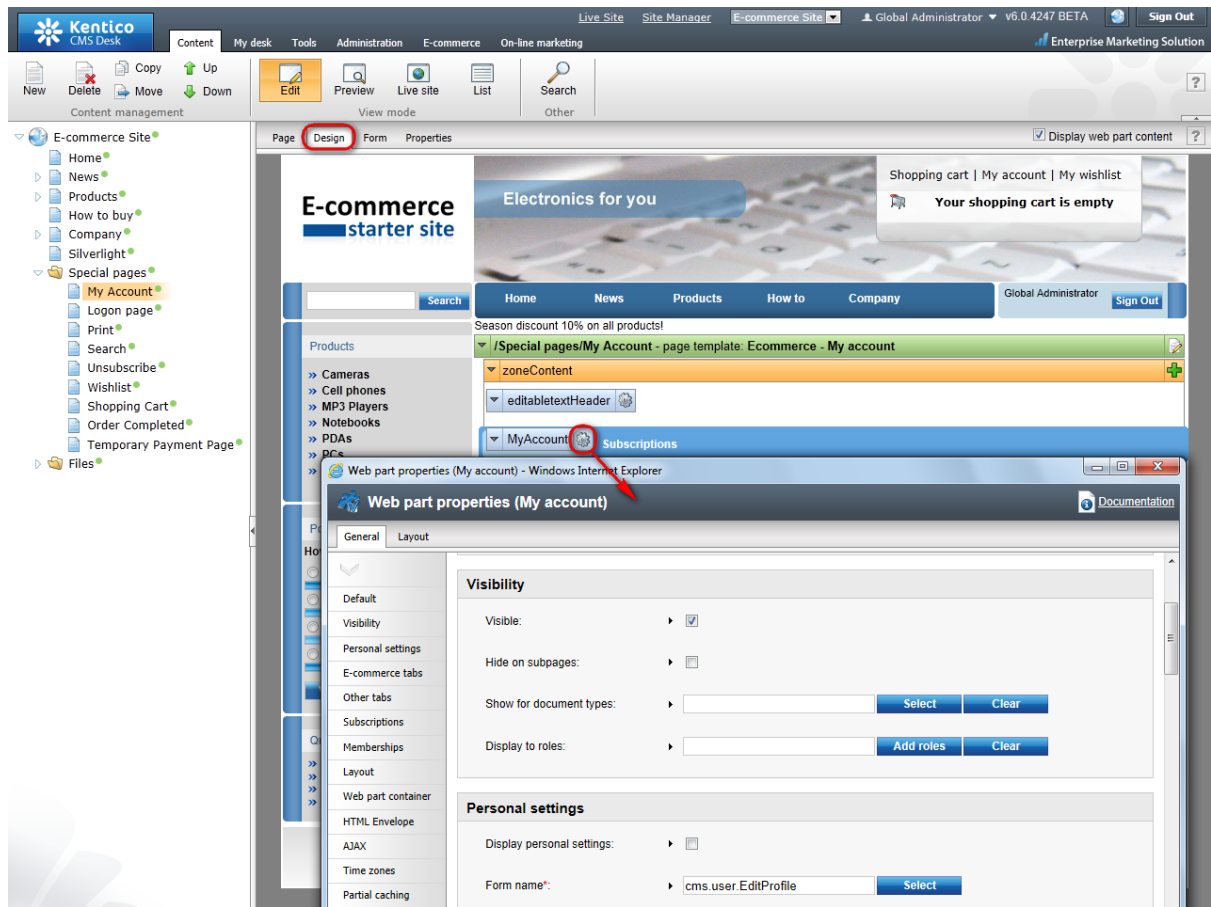
Details Addresses Orders Credit Change password Subscriptions

First name: Global  
Last name: Administrator  
Company account: ☐  
E-mail: administrator@localhost.local  
Phone:   
Fax:   
Preferred currency: U.S. Dollar  
Preferred shipping option: (none)  
Preferred payment method: (none)  
Country/state: (none)

OK

Powered by Kentico

You can modify the functionality of **My account** by setting the following properties in the **Web Part properties** dialog:



## Personal settings

Display personal settings	Indicates if a tab where users can edit their personal settings should be displayed.
Form name	Can be used to select an alternative form that will be used to view and edit the user's personal settings. Alternative forms are specified in format <class name>.<form code name>.
Allow user to edit field visibility	Indicates if a tab where users can change their password should be displayed. This tab will always be hidden for users that are authenticated against an external source (those that are marked with the IsExternal flag).
Allow empty password	Indicates if users should be allowed to enter an empty password for their account.

## Ecommerce

Display customer details	Indicates if a tab where users can edit their customer details should be displayed. This only applies to users who are registered as customers in the E-commerce module.
--------------------------	--

Display my addresses	Indicates if a tab where customers can manage their addresses should be displayed.
Display my orders	Indicates if a tab where customers can manage their orders should be displayed.
Show order tracking number	Indicates if the tracking number should be visible in the order list.
Display my credit	Indicates if a tab where customers can manage their credit events should be displayed.

### Other tabs

Display my subscriptions	Indicates if a tab where users can manage their subscriptions should be displayed.
Display my notifications	Indicates if a tab where users can view their notification messages should be displayed.
Display my messages	Indicates if a tab where users can view their personal messages (sent using the website's internal messaging system) should be displayed.
Display my friends	Indicates if a tab where users can manage their friends should be displayed.
Display my memberships	Indicates if the Memberships tab should be displayed to users. Here, users can view which memberships are currently assigned to them and their remaining duration.
Display my categories	Indicates if a tab where users can manage their personal categories should be displayed.

### Memberships

Memberships page URL	If a path is entered, the web part will generate a link to a page displaying paid membership products, which can be added to the shopping cart. The page thus allows users to <a href="#">buy new memberships or renew existing ones</a> .
----------------------	--

#### Please note

To have the **Buy membership button** enabled on a page using the My account web part, the **Memberships/Memberships page URL property** must be defined.

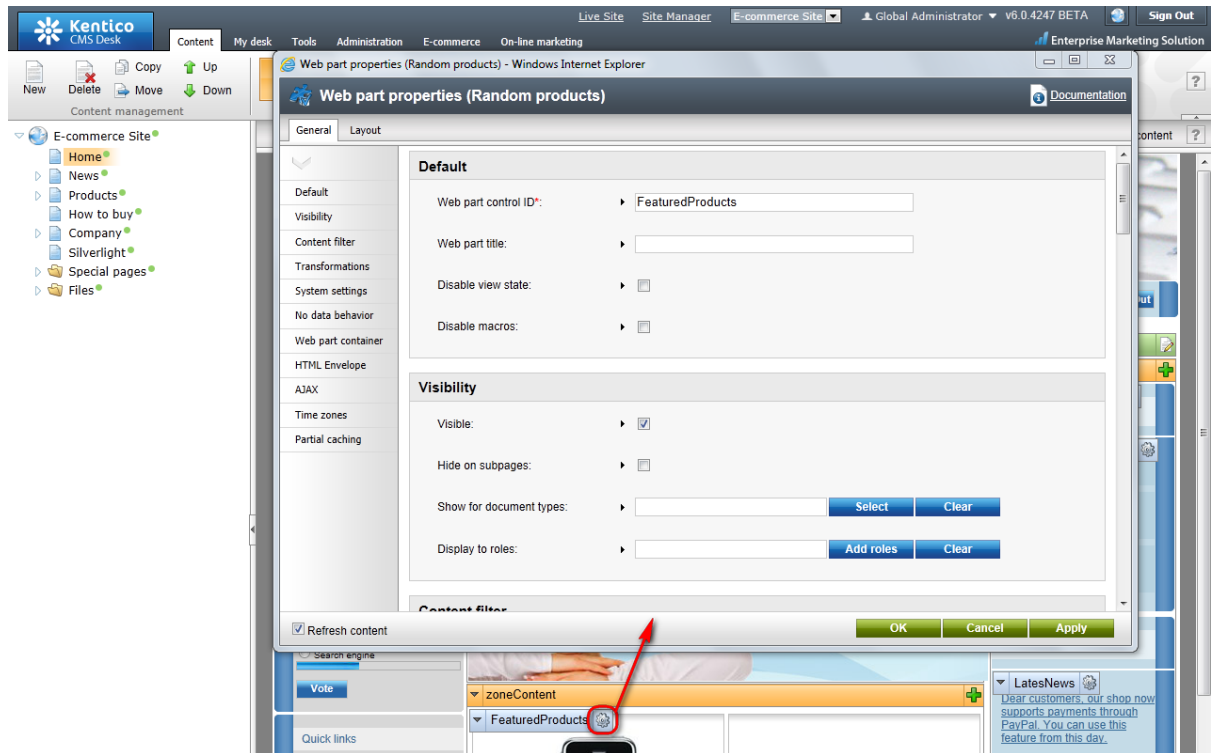
For more details about the **My account** web part please refer to the web part documentation, available from the **Web part properties (My account)** dialog.



## 5.2.2 Random products

The **Random products** web part displays products that correspond to the criteria specified in the content filter.

You can modify the functionality of **Random products** by setting properties in the **Web Part properties** dialog as you can see on the following image:



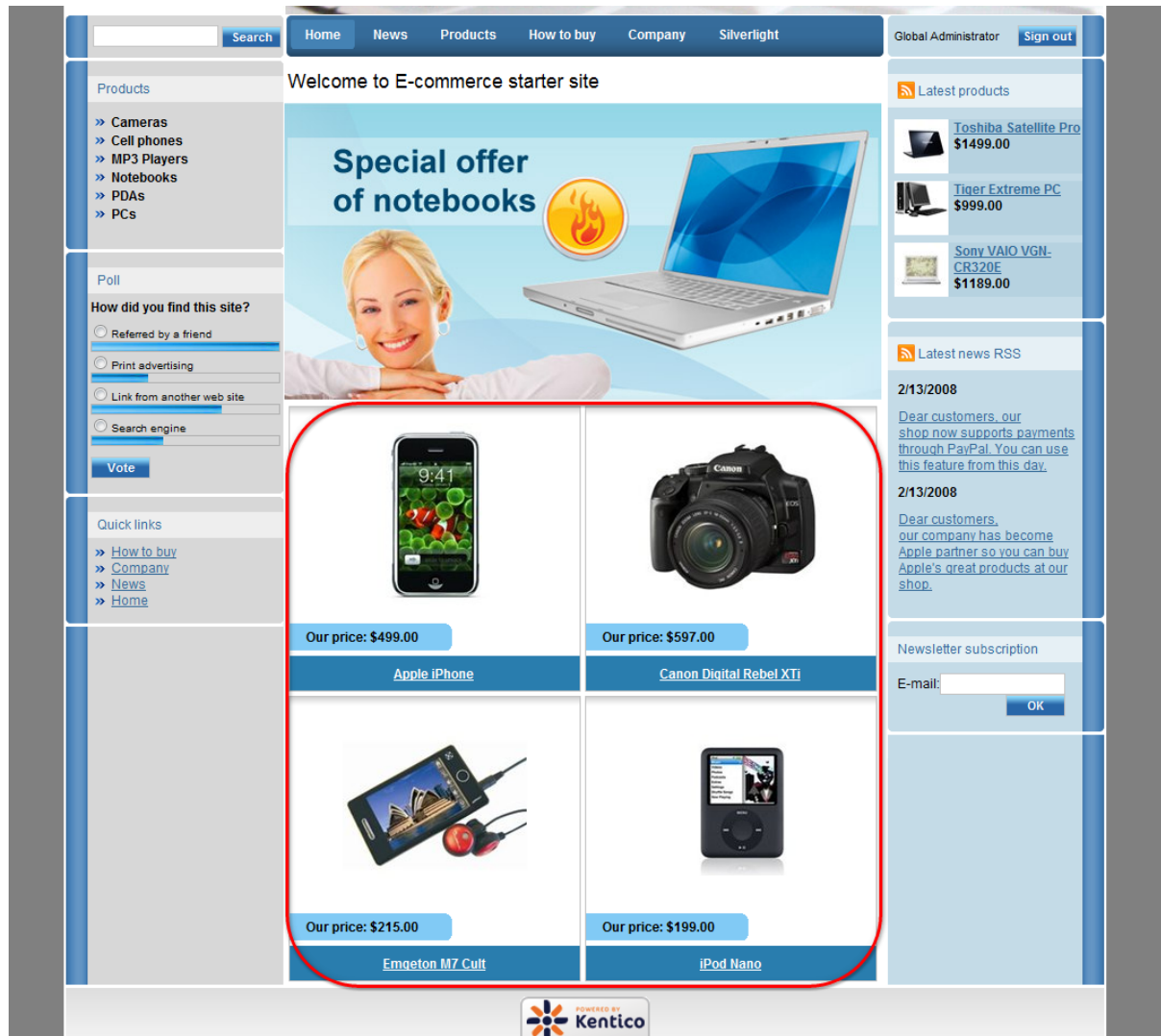
### Content filter main properties

<b>Product public status</b>	Public status of products which should be displayed.
<b>Product internal status</b>	Internal status of products which should be displayed.
<b>Product department</b>	Department of products which should be displayed.
<b>Only random N products</b>	Indicates how many random products should be displayed. Don't put any value, if you want display all products.
<b>Top N</b>	The first N products selected according to the chosen criteria.
<b>Path</b>	Path of the documents to be displayed.
<b>Document Types</b>	Indicates from which document types should be products chosen.
<b>WHERE Condition</b>	WHERE part of the SELECT query.
<b>ORDER BY expression</b>	ORDER BY part of the SELECT query.

The **Random products** web part can be used in many different ways. All you have to do is to set the

content filter accordingly to get rights products displayed.

## 1. Featured products on Home page



On the **Home** page, **Random products** is set to display the maximum of 4 random products that have the public status set to **Show as featured product**:

### Only random N products: 4

The maximum of 4 random products is displayed.

### Product public status: Show as featured product

Only products with the public status set to **Show as featured product** are displayed.

### Transformation name: ecommerce.transformations.EcommerceSite\_FeaturedProducts

The ecommerce.transformations.EcommerceSite\_FeaturedProducts transformation used in the list view mode.

For description of how to display the given product as the featured product, please refer to the [Displaying](#)

[featured products](#) chapter.

## 2. Similar products at the product detail



At the product detail, the given web part is set to display the maximum of 4 random products which have price similar to the currently chosen product and which are of the same type.

**Show for document types:** CMSProduct.Camera;CMSProduct.CellPhone;CMSProduct.Computer;CMSProduct.Laptop;CMSProduct.MediaPlayer;CMSProduct.PDA

**Random product** is displayed only in the given document types, therefore, at product detail. If a user switch to the product list, the given web part is hidden. If you add a new product type and you want it to be displayed with the **Random product** web part, you need to add your new product type into the **Show for document types** text box.

### Only random N products: 4

The web part displays 4 random products at most, according to the filter conditions.

**WHERE condition:** (SKUPrice > {%SKUPrice%} \* 0.6) AND (SKUPrice < {%SKUPrice%} \* 1.4) AND (NodeSKUID <> {%NodeSKUID%})

Ensures that only products with the price similar to the currently chosen product are displayed.

**Document types:** {%ClassName%}

Ensures that only the products of the same type as the currently chosen product are displayed.

For more details about the context macros please refer to the **Development -> Macro expressions** chapter in **Kentico CMS Developer's Guide**.

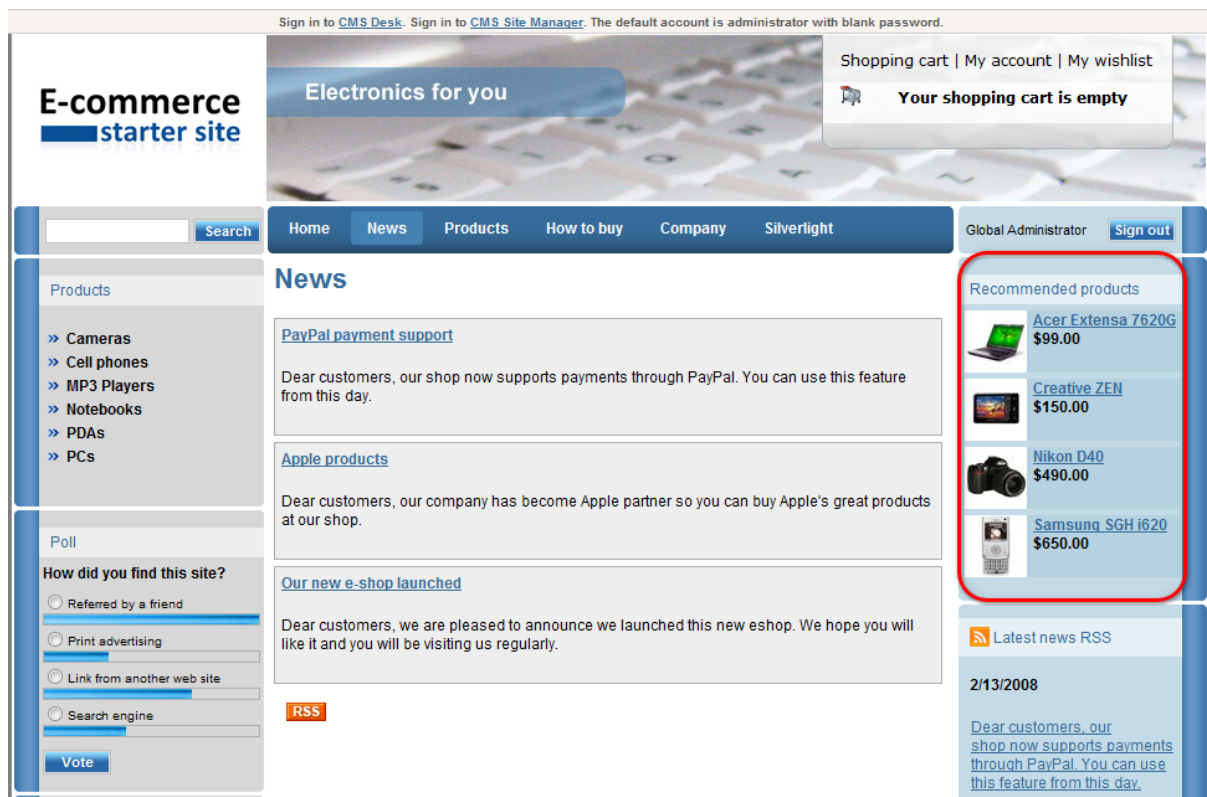
**Transformation name:** Ecommerce.Transformations.EcommerceSite\_RandomProducts

The Ecommerce.Transformations.EcommerceSite\_RandomProducts transformation used in the list view mode.

For instance, if the currently chosen product is type of **cmsproduct.cellphone** with **SKUPrice** set to 100, the **Random product** web part displays the maximum of 4 products, type of **cmsproduct.cellphone** with **SKUPrice** between 60 and 140.

### 3. Recommended products on News page and Home page

On the **News** page, the given web part is set to display 4 random products, regardless of their type.



In this case, the **Only random N products** property is set to **4** and the **Transformation name** property is set to **toecommerce.transformations.Ecommerce\_RandomProducts**.

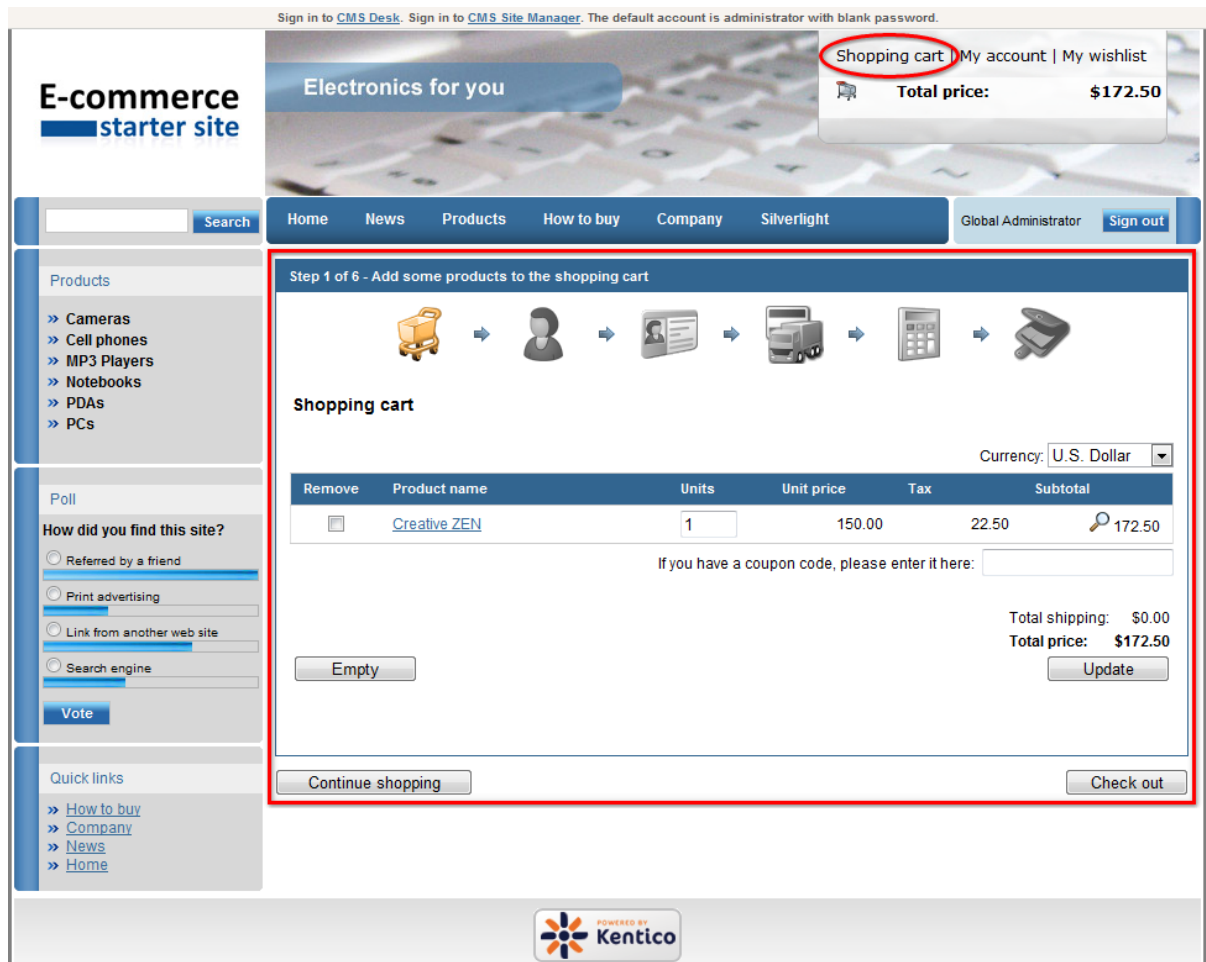
On the **Home** page, the given web part is set to display 1 random product, regardless of its type.



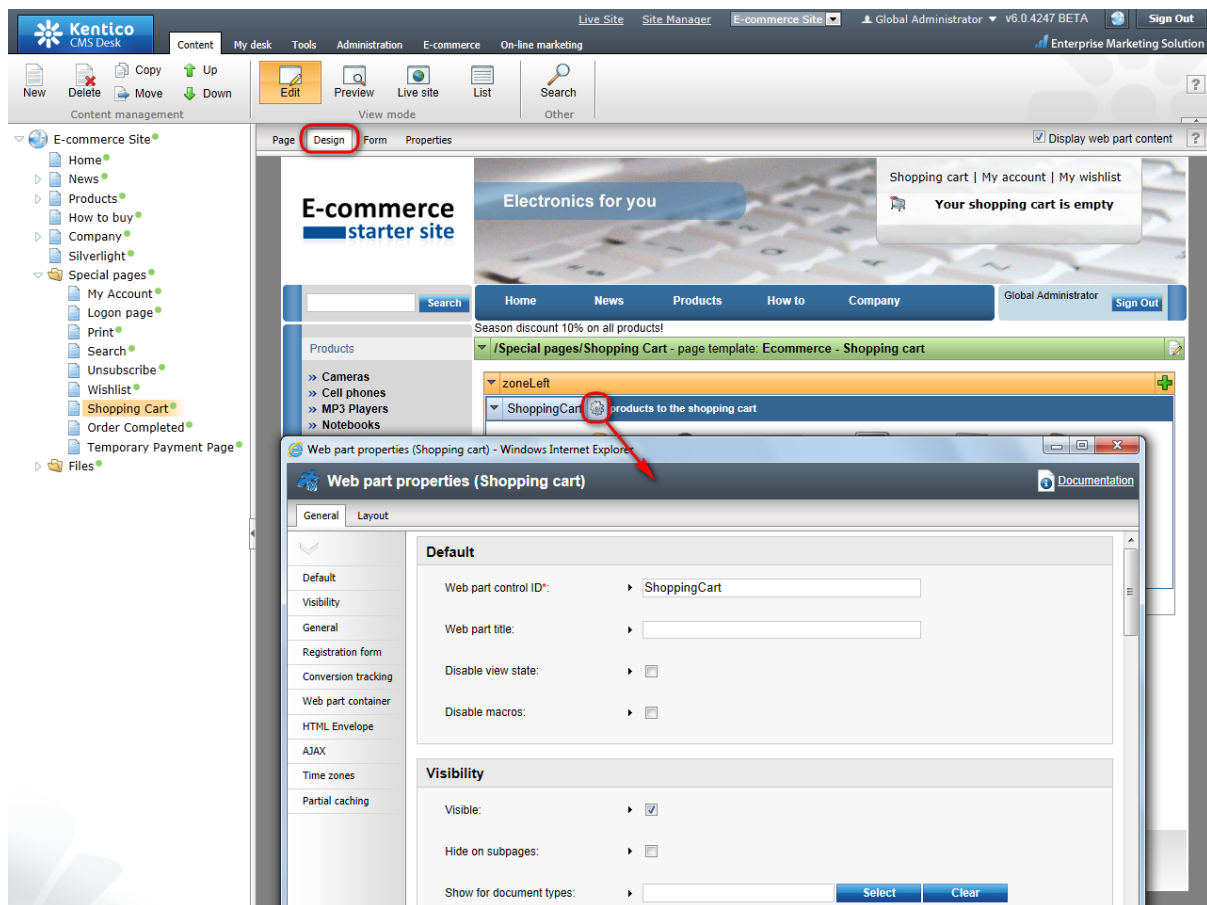
In this case, the **Only random N products** property is set to **1**. The **Transformation name** property is set to **toecommerce.transformations.Ecommerce\_RandomProduct**.

### 5.2.3 Shopping cart

The **Shopping cart** web part displays the shopping cart content and ensures the check-out process.



You can modify the functionality of **Shopping cart** by setting following properties in the **Web Part properties** dialog:



## General properties

<b>Default URL after purchase</b>	Default page where the user should be redirected if no URL is specified for the given payment option.
<b>Allow forgotten password retrieval</b>	Indicates if the forgotten password can be retrieved in the <b>Shopping Cart sign-in</b> dialog.
<b>Display step images</b>	Indicates if the images should be displayed during the order process
<b>Image step separator</b>	The separator displayed between shopping progress images.
<b>Enable product price detail</b>	Indicates if the link to the price detail page should be displayed.
<b>Required fields mark</b>	HTML code for the required fields mark, e.g. asterisk (*)

## Registration form properties

<b>Assign user to role</b>	If you enter some role to this field, the user will be automatically assigned to it after registration.
----------------------------	---



**Notify administrator about new registrations to e-mail address**

Enter administrator's e-mail address if you want to send registration notification message.

## Conversion tracking properties

**Registration conversion name**

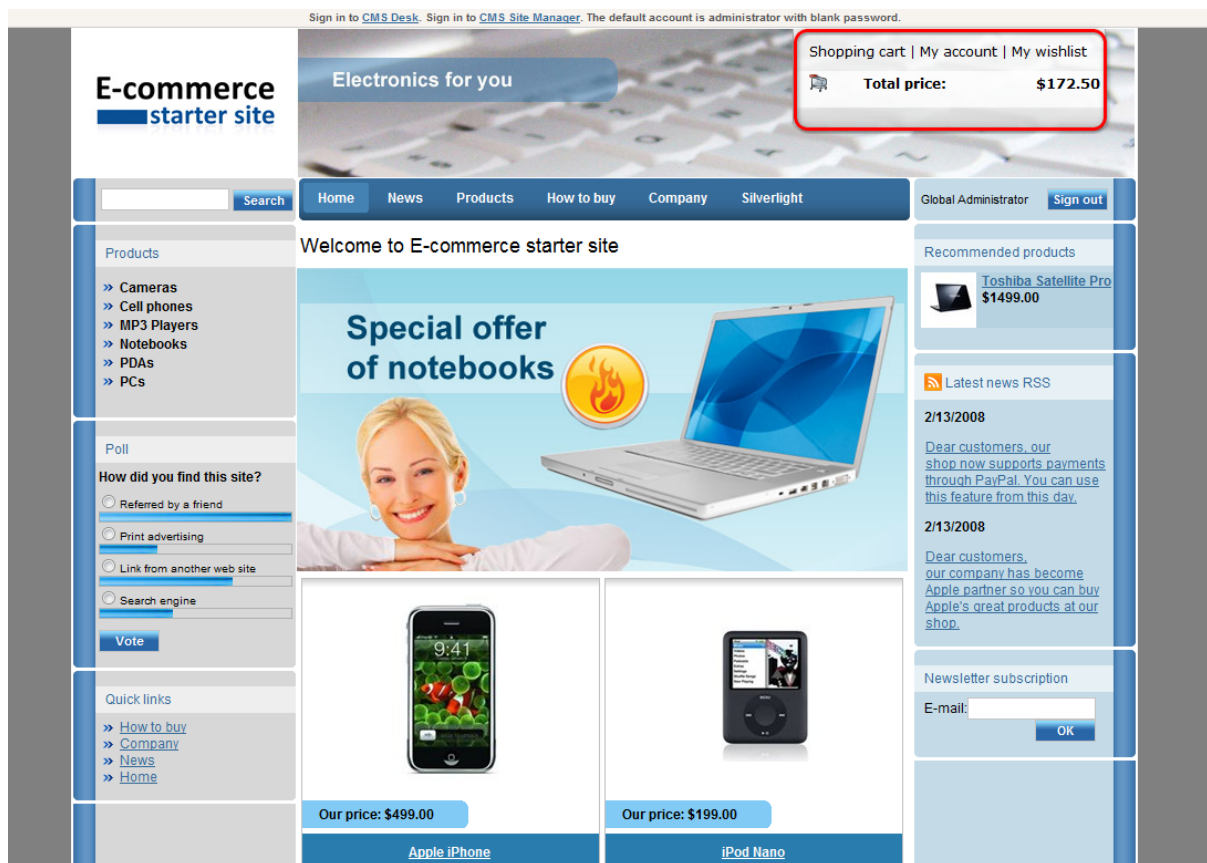
Name of the registration conversion used in web analytics.

**Order conversion name**

Name of the order conversion used in web analytics.

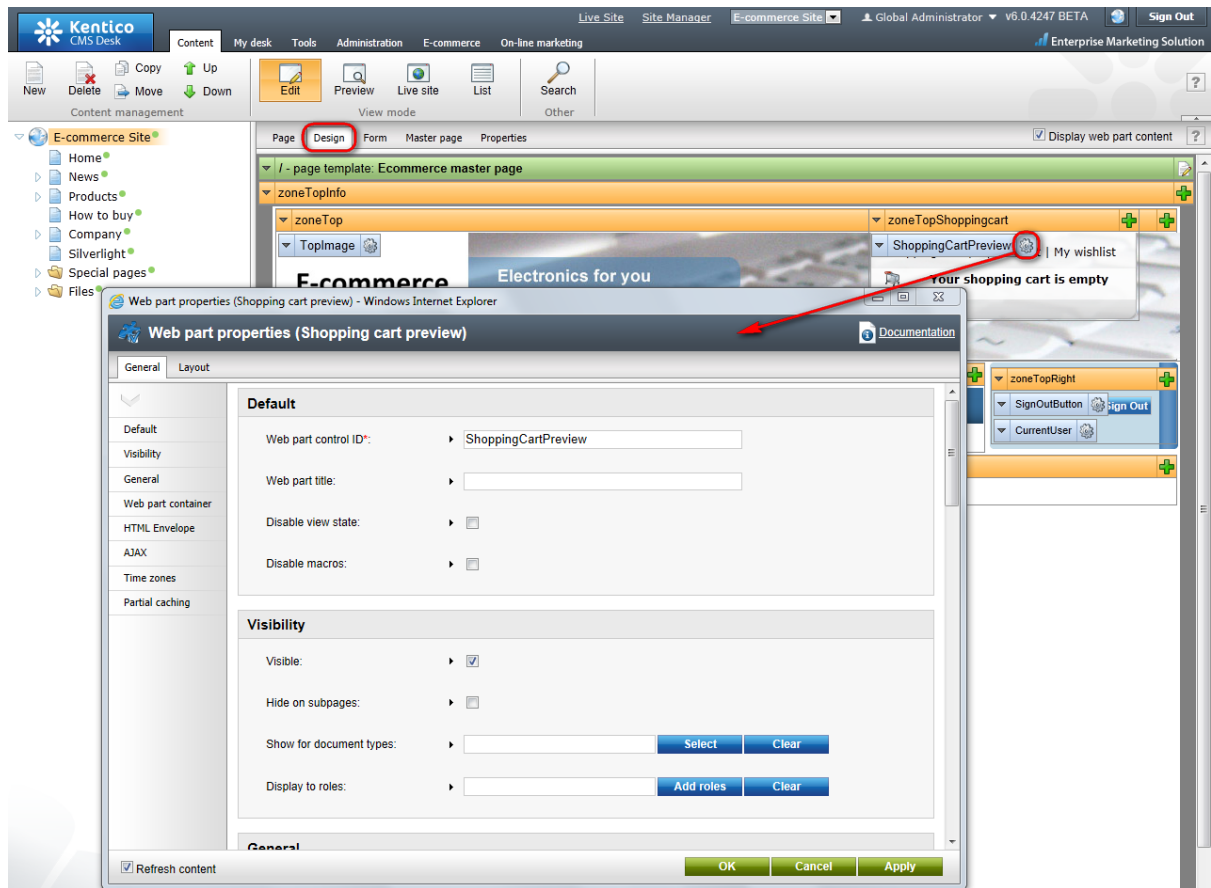
### 5.2.4 Shopping cart preview

The **Shopping cart preview** web part displays the links to the shopping cart, to my account and to my wishlist and the total value of the shopping cart content.



You can modify the functionality of **Shopping cart preview** by setting following properties in the **Web Part properties** dialog:





## General properties

<b>Shopping cart link URL</b>	URL of the page with <b>Shopping cart</b> web part. If not set, the default URL from the <b>Site Manager -&gt; Manager -&gt; Settings -&gt; E-commerce -&gt; Shopping cart URL</b> settings is used.
<b>Shopping cart link text</b>	Text of the link to the <b>Shopping cart</b> page.
<b>Wishlist link URL</b>	URL of the page with <b>Wishlist</b> web part. If not set, the default URL from the <b>Site Manager -&gt; Settings -&gt; E-commerce -&gt; Wishlist URL</b> settings is used.
<b>Wishlist link text</b>	The link text for the <b>Wishlist</b> URL.
<b>My Account link URL</b>	URL of the page with <b>My Account</b> web part. If not set, the default URL from the <b>Site Manager -&gt; Settings -&gt; E-commerce -&gt; My account URL</b> settings is used.
<b>MyAccount link text</b>	Text of the link to the <b>My Account</b> page.
<b>Total price text</b>	Text displayed next to the total price.
<b>Show total price row</b>	Indicates if total price row should be displayed.

## 5.2.5 Similar products by sale

The **Similar products by sale** web part displays N products which other customers buy with the chosen product most commonly. By default, four products are displayed at the most.

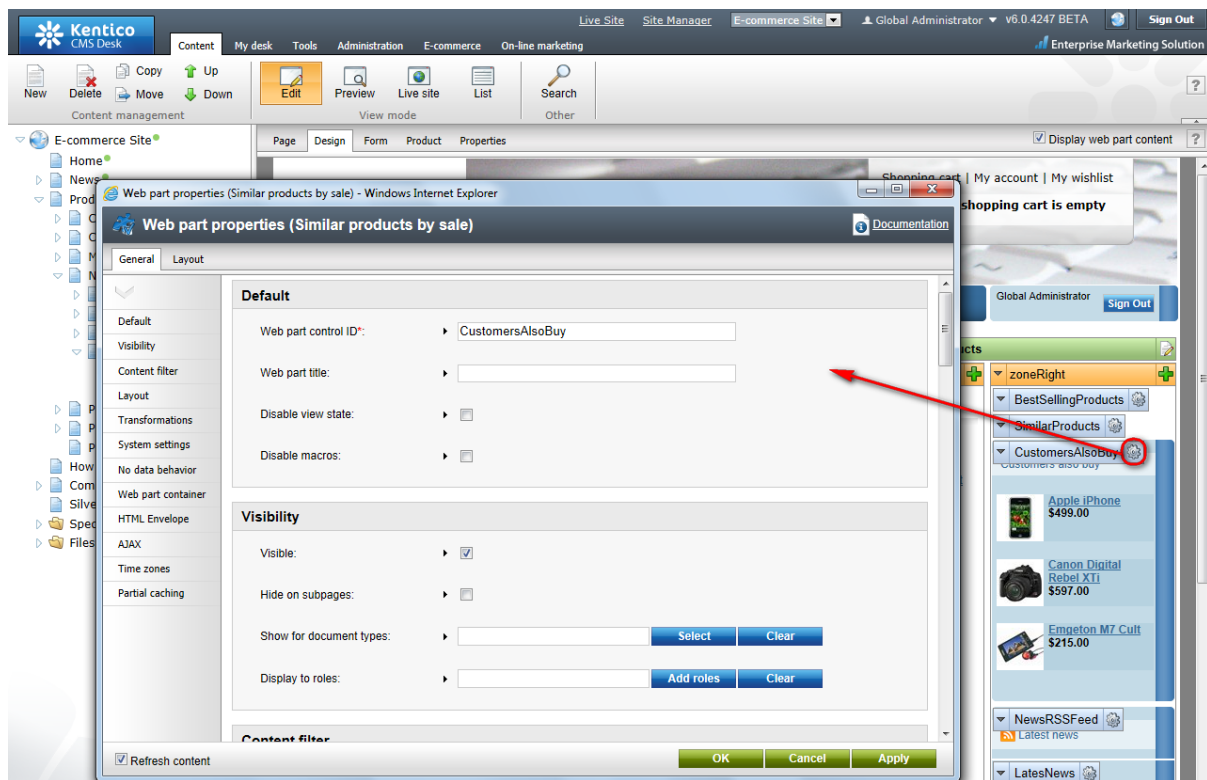
For instance, if a customer has bought product A together with products B and C, some other customer who is viewing the details of product A is being displayed also with products B and C. This is because products B and C were purchased together with product A by the first customer.

The screenshot shows an E-commerce starter site with the following elements:

- Header:** "E-commerce starter site" logo, "Electronics for you" banner, and a shopping cart icon with the message "Your shopping cart is empty".
- Navigation:** Home, News, Products, How to buy, Company, Silverlight.
- Left Sidebar:**
  - Search bar
  - Products list: Cameras, Cell phones, MP3 Players, Notebooks, PDAs, PCs
  - Poll: "How did you find this site?" with options: Referred by a friend, Print advertising, Link from another web site, Search engine.
  - Quick links: How to buy, Company, News, Home
- Main Content Area:**
  - Breadcrumb: Products / Cell phones / Samsung SGH i620
  - Product title: Samsung SGH i620
  - Product image: Samsung SGH i620
  - Parameters:
 

Width:	94.9
Height:	59.3
Weight(g):	126
Display type:	TFT 65M colors
Display resolution:	320 x 240
Bluetooth:	yes
IrDA:	no
GPRS:	yes
EDGE:	yes
HSCSD:	no
3G:	yes
Wi-Fi:	no
Java:	yes
Built-in camera:	yes
MP3 player:	yes
  - Price: Our price: \$650.00, Without tax: \$650.00
  - Color selection: Samsung SGH i620 - Color, Red (+ \$0.00)
  - Total price (without tax): \$650.00
- Right Sidebar:**
  - Similar products:
    - Nikon Coolpix S51 \$1950.00
    - Nikon D40 \$490.00
    - Olympus Stylus FE-230 \$159.00
    - Pentax K10D \$804.00
  - Customers also buy (highlighted with a red box):
    - Creative ZEN \$150.00
    - Nikon Coolpix S51 \$1950.00
  - Latest news RSS: 2/13/2008

You can modify the functionality of **Similar products by sale** by setting following properties in the **Web Part properties** dialog:



## Content filter main properties

<b>Path</b>	Path of the documents to be displayed.
<b>Document types</b>	Types of documents that should be displayed, separated with a semicolon (;).
<b>Select top N products</b>	Selects only top N products. If blank, all items are selected..
<b>WHERE condition</b>	WHERE part of the SELECT query.
<b>ORDER BY expression</b>	ORDER BY part of the SELECT query.

## Transformations main properties

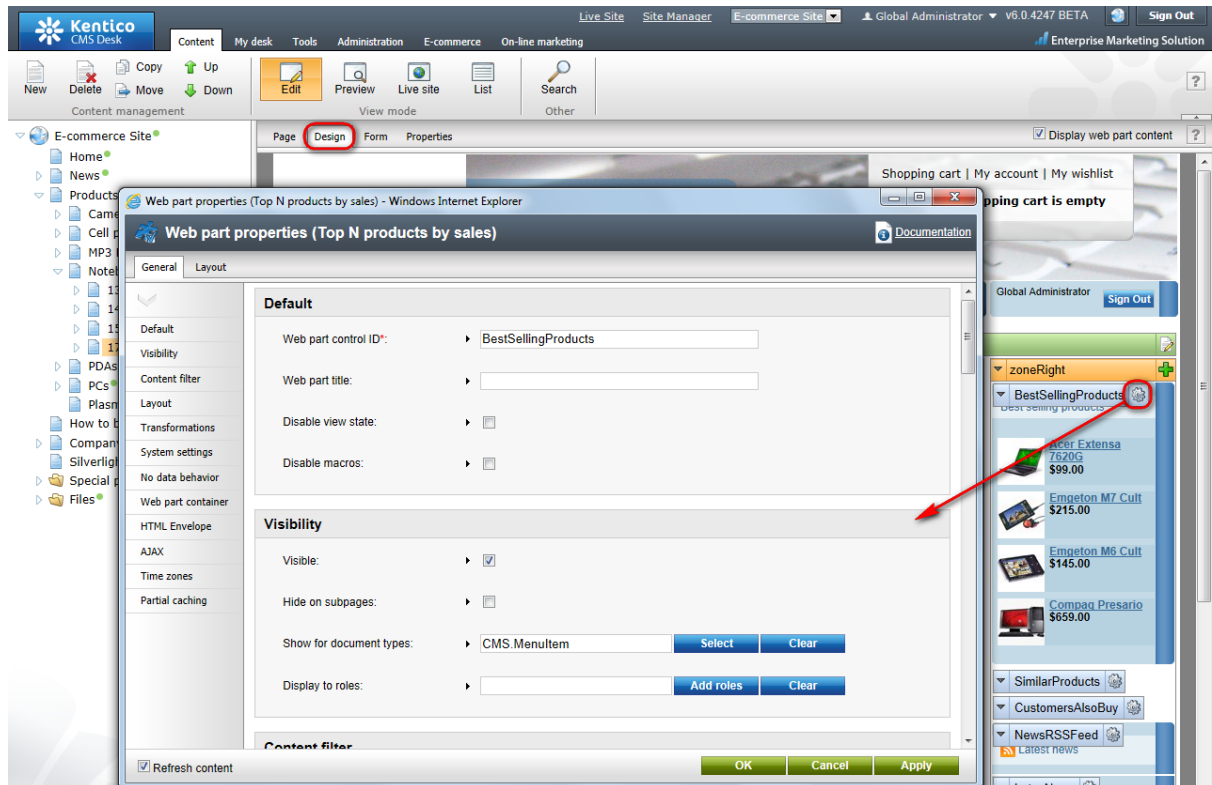
<b>Transformation name</b>	Transformation used in the list view mode.
----------------------------	--

### 5.2.6 Top N products by sale

The **Top N products by sale** web part displays the top N best-selling products. The best-selling products are chosen according to the frequency of their occurrence in customers' orders, not according to the total volume of sales.

For instance, if a customer buys 10 items of the product A and 2 items of the product B and a second customer buys 4 items of the product B, the product B is evaluated as the best-selling.

You can modify the functionality of **Top N products by sale** by setting following properties in the **Web Part properties** dialog:

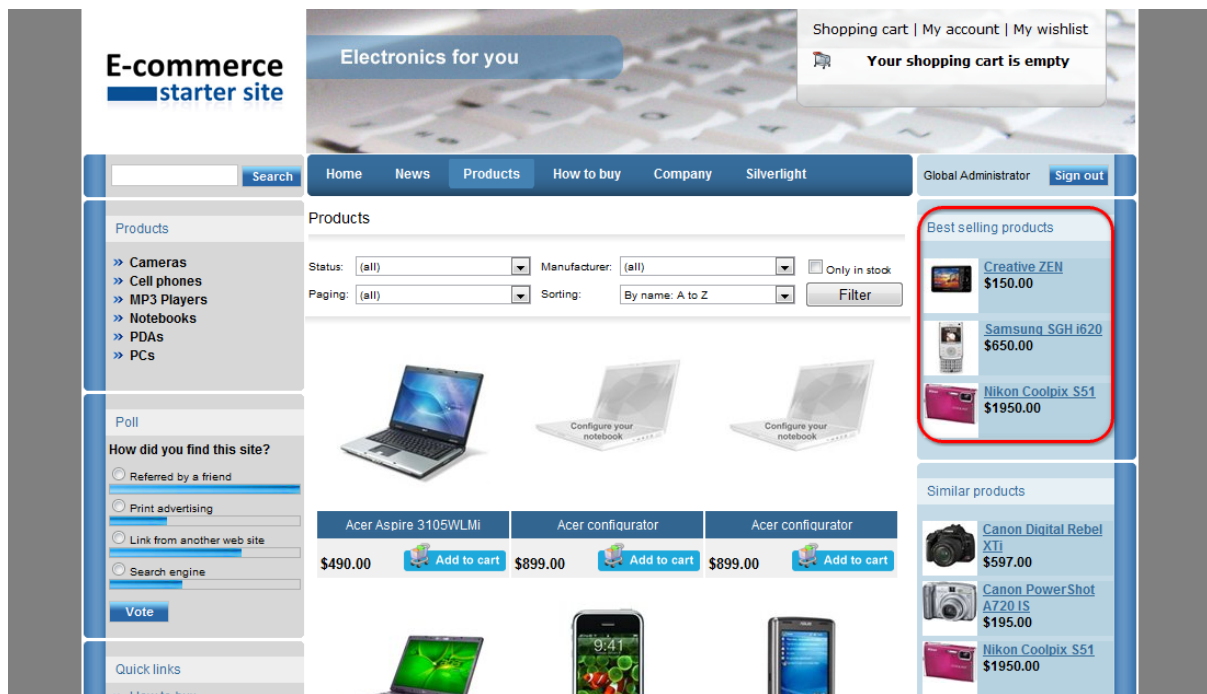


## Content filter main properties

<b>Path</b>	Path of the documents to be displayed.
<b>Document types</b>	Types of documents that should be displayed, separated with a semicolon (;).
<b>Select top N products</b>	Indicates how many best-selling products should be displayed.
<b>WHERE condition</b>	WHERE part of the SELECT query.
<b>ORDER BY expression</b>	ORDER BY part of the SELECT query.

## Transformations main properties

<b>Transformation name</b>	Transformation used in the list view mode.
----------------------------	--



On the sample E-commerce site at the **Products** page, the **Top N products by sale** web part is configured to display 4 products at most from the currently chosen products category. Its properties are set to the following values:

**Show for document types:** CMS.Menuitem

The given web part is displayed only in **CMS.Menuitem** documents. On the sample **E-commerce site**, the given web part is therefore displayed only in the list of products of a chosen category and is hidden in the product detail.

**Path:** ./%

Only best-selling products from the currently selected category are displayed. For more information about the path expressions please refer to the **Appendix A - Path expressions** chapter in **Kentico CMS Developer's Guide**.

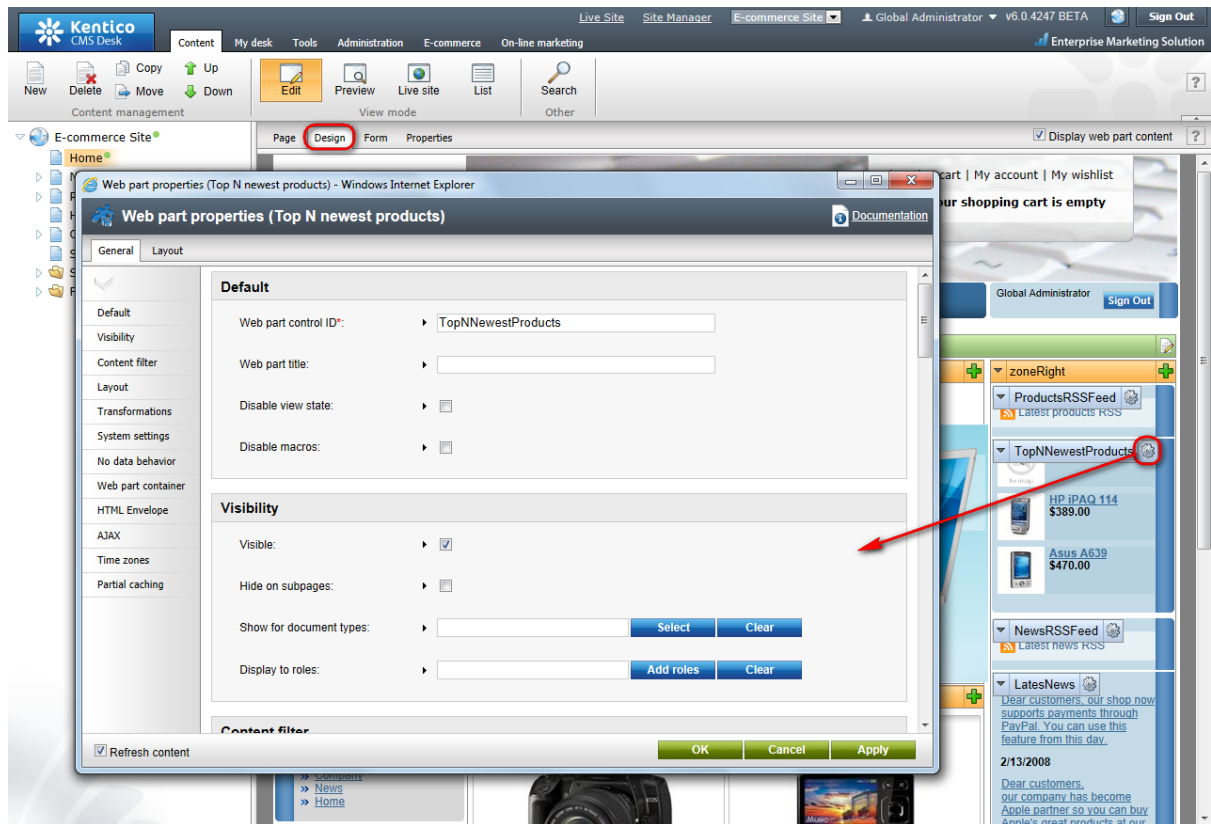
**Select top N products:** 4

The maximum of 4 best-selling products is displayed.

## 5.2.7 Top N newest products

The **Top N newest products** web part displays the top N newest products, i.e. products that have been added recently.

You can modify the functionality of the **Top N newest products** web part by setting the following properties in the **Web Part properties** dialog:



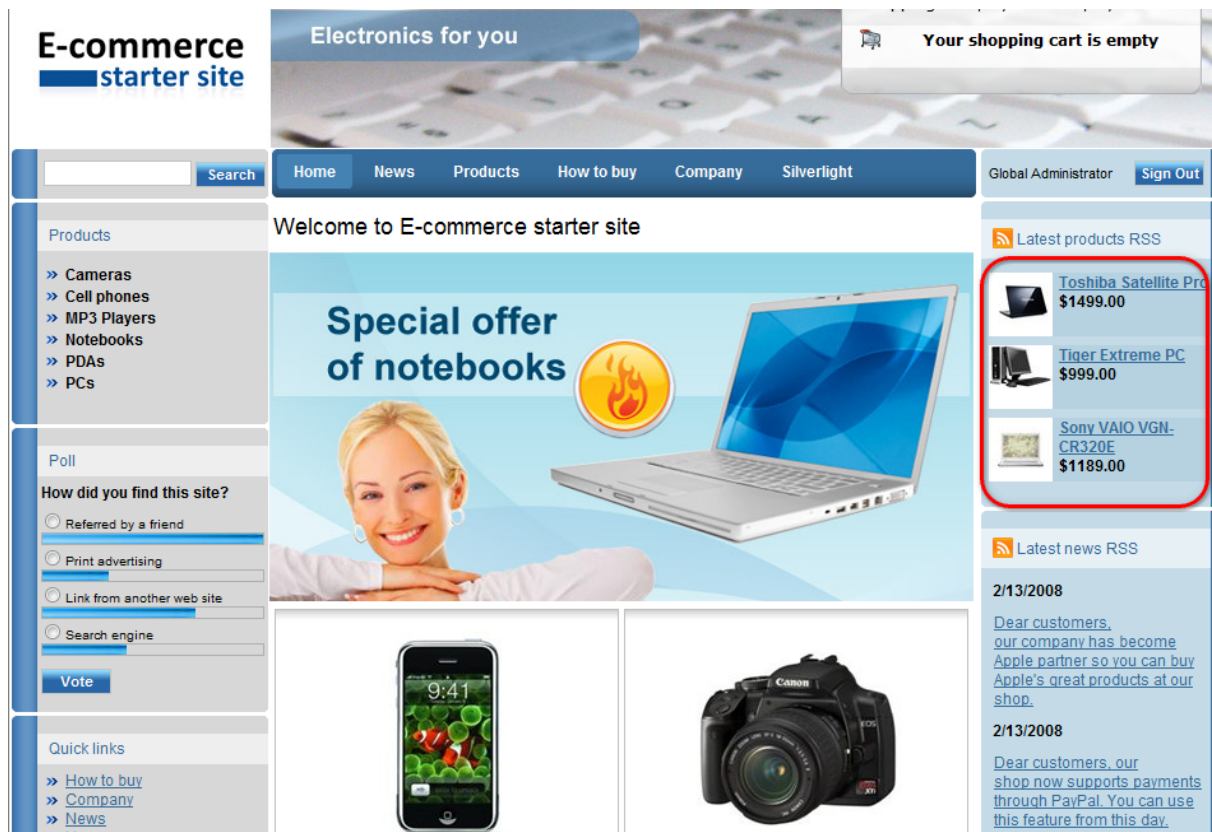
## Content filter main properties

<b>Path</b>	Path of the documents to be displayed.
<b>Document types</b>	Types of documents that should be displayed, separated by a semicolon (;).
<b>Select top N products</b>	Indicates how many newest products should be displayed.
<b>WHERE condition</b>	WHERE part of the SELECT query.
<b>ORDER BY expression</b>	ORDER BY part of the SELECT query.

## Transformations main properties

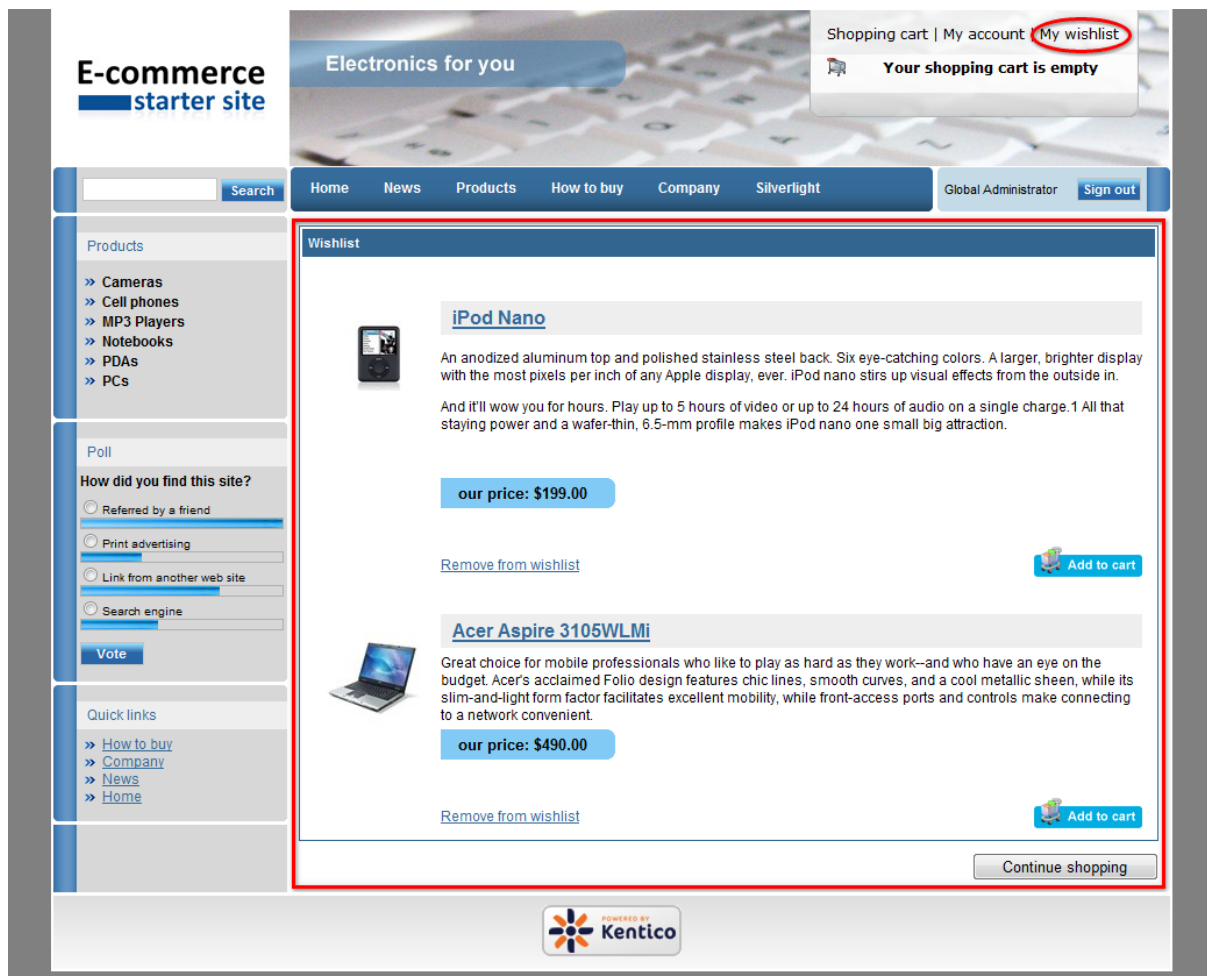
<b>Transformation name</b>	Transformation used in the list view mode.
----------------------------	--

On the **Home** page of the sample E-commerce site, the **Top N newest products** web part is configured to display 3 newest products by default.



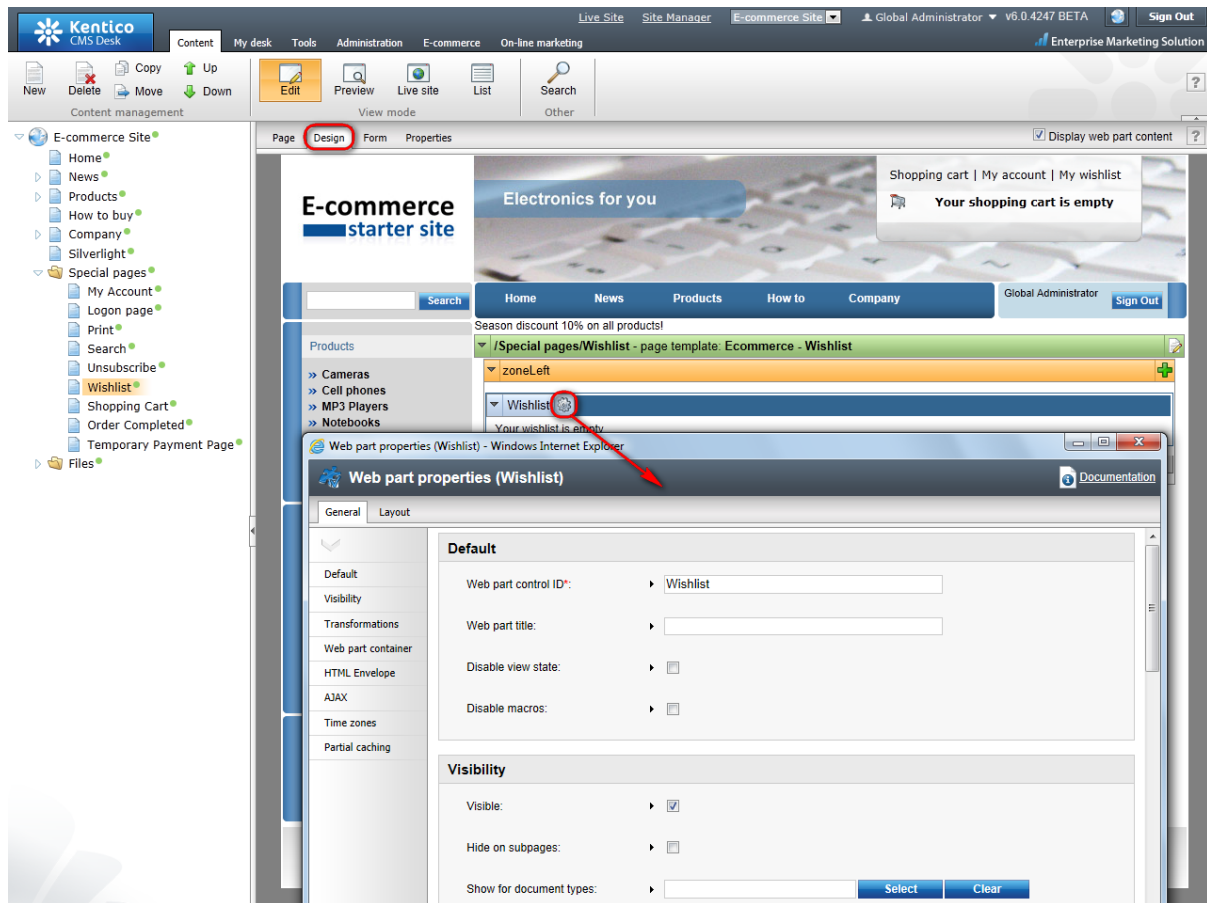
## 5.2.8 Wishlist

The **Wishlist** web part displays the wishlist of the currently logged-on user.



You can modify the functionality of **My wishlist** by setting following properties in the **Web Part properties** dialog:





## Main properties

Transformation name	Transformation used in the list view mode.
---------------------	--

## 5.3 Displaying and resizing images

### 5.3.1 Overview

If you want to use a picture multiple times on your website, each time in different sizes, it is sufficient to upload it just once. KenticoCMS is able to resize it for you. Please consider that the size of a picture can only be decreased, not increased, though. For this reason, it is important that you upload your pictures in the maximum size you want to use on your website.

### 5.3.2 Displaying images

#### 1. How to display image in transformation

To get an image in the given size, you need to insert one of the following methods into the transformation:

### a) Getting image by its attachment GUID

**GetImage**(object attachmentGuidColumn, object maxSideSize, object width, object height, object alt)

**GetImage**(object attachmentGuidColumn)

**GetImage**(object attachmentGuidColumn, int maxSideSize)

**GetImage**(object attachmentGuidColumn, int width, int height)

### b) Getting image by its URL

**GetImageByUrl**(object imageUrl, object maxSideSize, object width, object height, object alt)

**GetImageByUrl**(object imageUrl)

**GetImageByUrl**(object imageUrl, int maxSideSize)

**GetImageByUrl**(object imageUrl, int width, int height)

These methods use following parameters:

**attachmentGuidColumn** - attachment GUID

**imageUrl** - image url

**maxSideSize** - required image max side size

**width** - required image width

**height** - required image height

**alt** - image alternate text

All the methods generate HTML code for inserting an image according to given parameters.

## Examples of displaying image in transformation by its URL

### a) Using Live URL

The screenshot displays the Kentico CMS 6.0 E-commerce Guide interface. The top navigation bar includes links for 'Live Site', 'Site Manager', 'E-commerce Site', and 'Global Administrator'. The left sidebar shows a tree view of the 'E-commerce Site' structure, with 'Files' expanded to show 'Header logo'. The main content area displays the 'Properties' tab for the 'Header logo' attachment. The 'Live URL' property is highlighted with a red circle and contains the value: `/KenticoCMS4247.10319BBeta/Files/e-commerce_site_logo.aspx`.

```
<%#GetImageByUrl("~/Files/ecommerce_site_logo.aspx")%>
```

b) Using the field value of the given document type that represents the image URL (e.g. SKUImagePath):

### Examples of displaying image in transformation by its attachment GUID

a) Using the field value of the given document type that represents the attachment GUID

In the following example, we have defined the new document type **Employee**. This type has two attributes: **Employee name** specified in the **EmployeeName** column and **Employee photo** specified in the **EmployeePhoto** column.

The screenshot shows the 'Form' tab of the Kentico CMS interface for an 'Employee' document type. It includes a 'Save' button and a 'Spell check' button. The form contains the following fields: 'EmployeeID' with the value '1', 'Employee name' with the value 'David Simons', and 'Employee photo' with the value 'david\_simons.gif'. There is an 'Upload' button and a 'Browse...' button next to the photo field. At the bottom, there are 'Publish from' and 'Publish to' fields, each with a 'Now' button.

To display the employee photo, you need to insert the **GetImage** method with the following syntax into the transformation.

```
<%#GetImage(Eval("EmployeePhoto"))%>
```

## 2. How to display product image in transformation

To get a product image in the given size, you have to insert the **GetProductImage** method with the following syntax into the transformation:

```
GetProductImage(object imageUrl, object alt)
GetProductImage(object imageUrl, object width, object height, object alt)
GetProductImage(object imageUrl, object maxSideSize, object alt)
```

An example of displaying an image without resizing:

```
<%#EcommerceFunctions.GetProductImage(Eval("SKUImagePath"), Eval(
"SKUName"))%>
```

The **GetProductImage** method generates HTML code that will insert an image into your page. If the

product image is not set, the method displays the default product image (see [Sitemanager settings](#) for more details) in the given size.



Please see the examples in the [Resizing images](#) chapter for more details.



### Storing images

For the best performance while loading images, please ensure images are enabled to be stored in the file system. For more details, please refer to the [Storing images](#) chapter.

## 5.3.3 Storing images

To improve the performance while loading images, please check **Store Files in File System** at **CMS Site Manager -> Files**.

Checking this option enables images to be stored in the file systems. Thus, anytime an image of the given size is needed, it can be loaded directly from the file system and doesn't have to be resized again from the original picture.

To be able to store images in the file system, the modify permissions have to be set for the whole web project (See the **Disk permissions problems** chapter in **Kentico CMS Developer's Guide** for more details).

All metafiles and attachments are stored in the folder specified at **CMS Site Manager -> Settings -> System -> Files -> Files folder**. If the **Files folder** is not set, the following happens:

- The object metafiles that are not assigned to the specific website are stored in **<web project folder>/CMSFiles**.
- The object metafiles that are assigned to the specific website are stored in **<web project folder>/<site name>/metafiles**.
- Attachments (they are always assigned to the specific website) are stored in **<web project folder>/<site name>/files**.

## 5.3.4 Resizing images

This chapter explains how to resize product images. However, all images in general can be resized in the following manner with usage of the methods for displaying images (**GetImage()** and **GetImageByUrl()**) with appropriate parameters.

You can set a size of the original image (width: 300px, height: 150px) in the following ways:



## 1. Setting Maxsize value

This sets the longer side to the value specified as **Maxsize**. The other side is calculated automatically so that the aspect ratio remains the same with the original image. In the following image, **Maxsize** is set to 250px.

The **GetProductImage** method for resizing an image to the Maxsize value of 250px:

```
EcommerceFunctions.GetProductImage(Eval("SKUImagePath"), 250, Eval("SKUName"))
```



## 2. Setting Height value

This sets the height of an image. The width is calculated automatically so that the aspect ratio remains the same with the original image. In the following image, **Height** is set to 100px.

```
EcommerceFunctions.GetProductImage(Eval("SKUImagePath"), 0, 100, Eval("SKUName"))
```



### 3. Setting Width value

This sets the width of an image. The height is calculated automatically so that the aspect ratio remains the same with the original image. In the following image, **Width** is set to 150px.

```
EcommerceFunctions.GetProductImage(Eval("SKUImagePath"), 150, 0, Eval("SKUName"))
```



### 4. Setting Width and Height values

The given values are set on condition they are not greater than the original size of an image. Please note that the aspect ration may not be maintained. In the following image, **Width** is set to 100px and **Height** is set to 150px.

```
EcommerceFunctions.GetProductImage(Eval("SKUImagePath"), 100, 150, Eval("SKUName"))
```



In the following image, **Width** is set to 300px and **Height** is set to 50px.

```
EcommerceFunctions.GetProductImage(Eval("SKUImagePath"), 300, 50, Eval("SKUName"))
```



If at least one value is greater than the original size of an image, the original size is set. In the following image, **Width** is set to 600px and **Height** is set to 300px.

```
EcommerceFunctions.GetProductImage(Eval("SKUImagePath"), 600, 300, Eval("SKUName"))
```



## 5.4 Tools

### 5.4.1 Overview

Kentico CMS includes a wide range of modules which can be used to build your e-commerce solution.

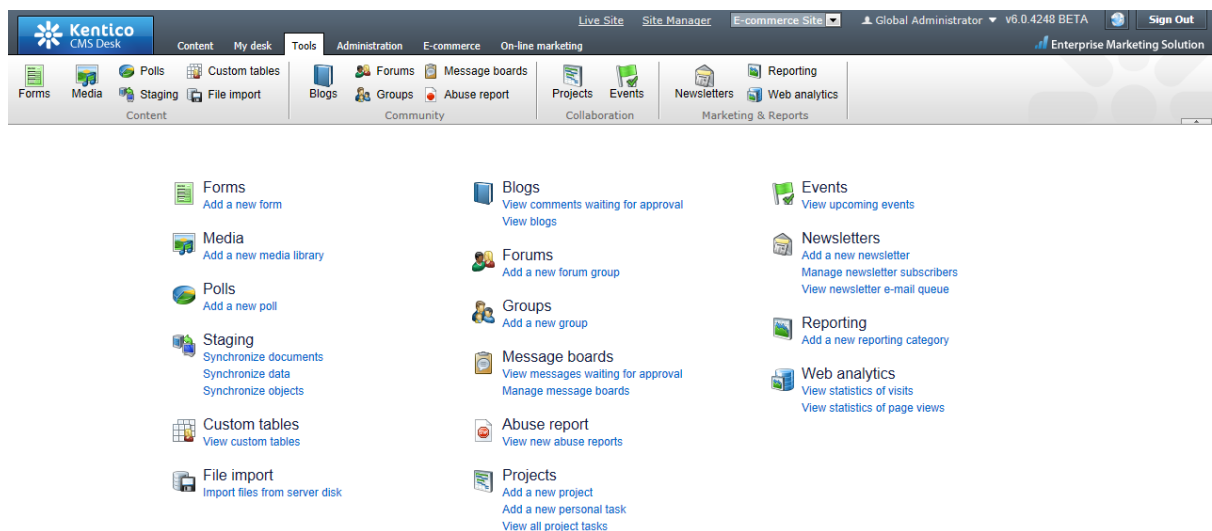
The included modules are:

- Blogs
- Events
- File import
- Forms
- Forums
- GeoMapping
- Image gallery
- Messaging
- Newsletters
- Polls
- Reporting

- RSS feeds
- Staging
- User contributions
- Web analytics
- Web farm synchronization

Please note that you are not limited to the pre-defined modules. You can easily create your own modules with the desired functionality. Please see the **Custom modules** chapter in **Kentico CMS Developer's Guide** for more details.

Please note that not all versions of Kentico CMS support all modules. For more information about modules which are included in the specific version of Kentico CMS please see the feature matrix at Kentico website (<http://kentico.com/cms-asp-net-features/Feature-Matrix.aspx>).



## 5.4.2 Forms

The Forms module enables you to create and publish simple on-line forms without writing a single line of code. For instance, you can manage communication with your customers more effectively using the **Contact us** form.

You can edit form properties at **CMS Desk -> Tools -> Forms -> <edit Form>**.



The screenshot shows the 'Form Properties' window in the Kentico CMS Desk. The 'General' tab is selected, showing the following details for the 'Contact Eshop' form:

- Form display name: Contact Eshop
- Form code name: ContactEshop
- Table name: Form\_ContactEshop

Under 'After the form is submitted:', the 'Display text' option is selected with the text 'Thank you for your message. We will contact you'. Other options include 'Redirect to URL:', 'Clear form', and 'Continue editing'.

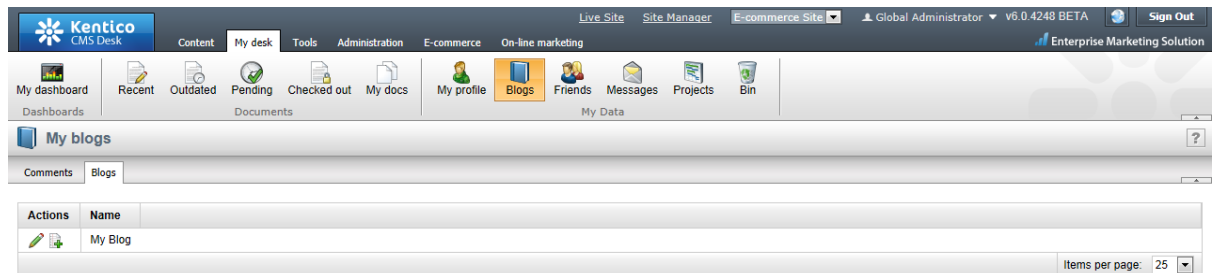
The 'Submit button text' is 'Send message' and the 'Submit button image' is an 'OK' button.

For more details and information about forms please refer to the **Modules -> Forms** chapter in **Kentico CMS Developer's Guide**.

### 5.4.3 Blogs

The Blogs module allows you to publish a personal or company blog. You can publish multiple blogs on the same site and there can be multiple editors for each blog.

You can edit blogs at **CMS Desk -> My Desk -> Blogs**.

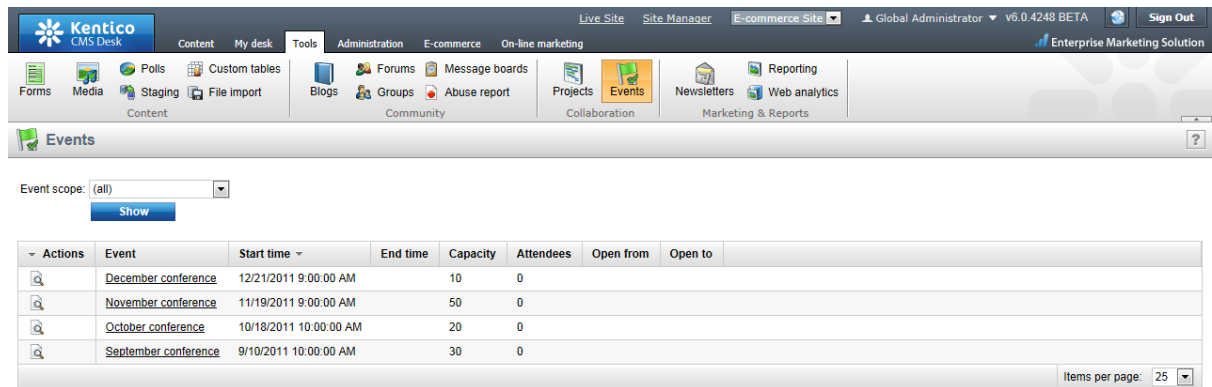


For more details and information about blogs please refer to the **Module Blogs** chapter in **Kentico CMS Developer's guide**.

### 5.4.4 Events

The Events module enables you to create an event and manage its attendees. After attendees are added, you can send e-mails to them with the important information about the event. The bookin system can be used both for on-line and off-line meetings.

You can edit events and add attendees at **CMS Desk -> Tools -> Events -> <view event>**.

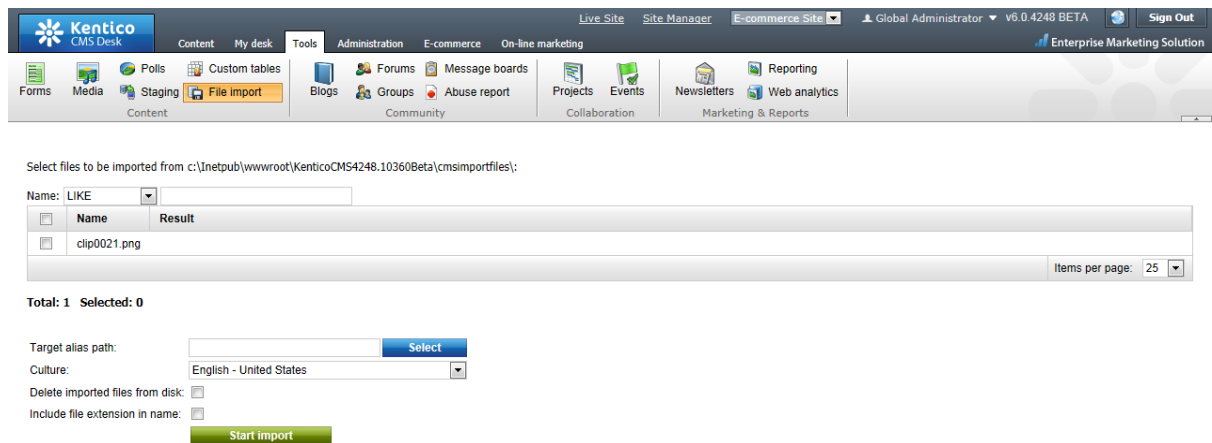


For more details and information about the Events module please refer to the **Modules -> Events** chapter in **Kentico CMS Developer's Guide**.

### 5.4.5 File import

The **File import** module allows you to import many files including their folder structure from the disk to Kentico CMS content repository, so that you do not have to upload the one-by-one using the user interface.

You can configure the path to your import directory at **CMS Desk -> Tools -> File import**.

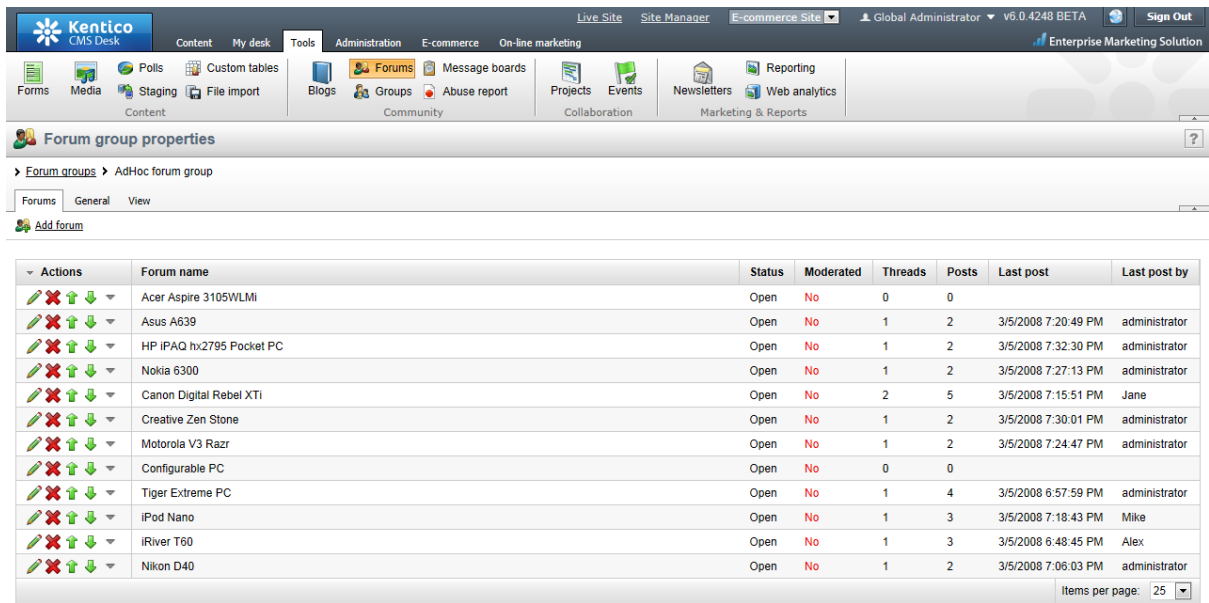


For more details and information about **File import** please refer to the **Module File import** chapter in **Kentico CMS Developer's guide**.

### 5.4.6 Forums

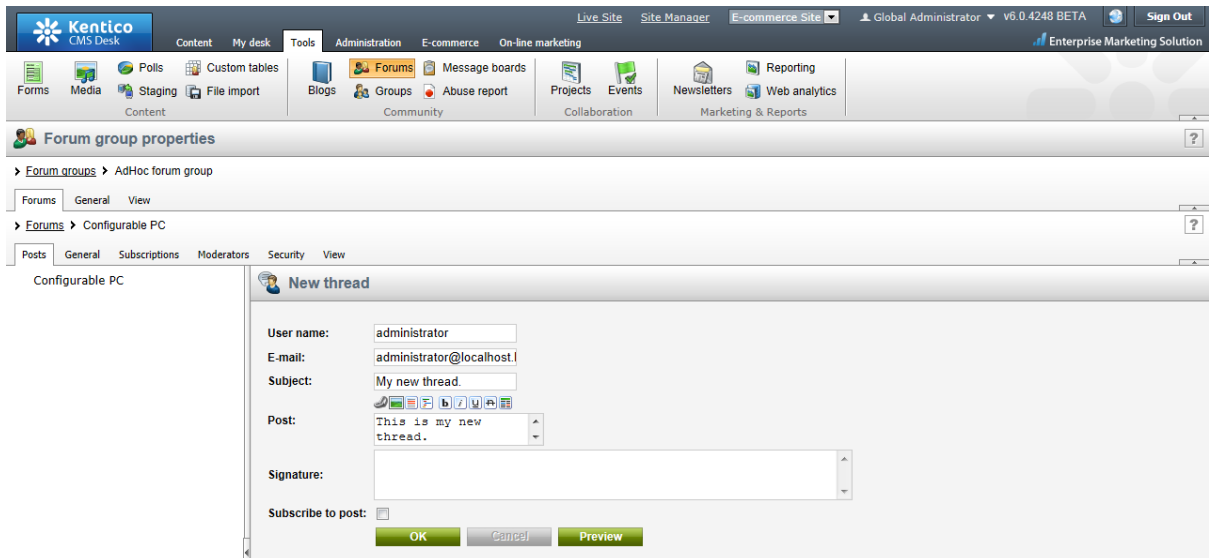
The Forums module allows your customers to add comments and feedback to each product.

You can edit forums on your website at **CMSDesk -> Tools -> Forums -> <edit forum group>**.



Actions	Forum name	Status	Moderated	Threads	Posts	Last post	Last post by
	Acer Aspire 3105WLMi	Open	No	0	0		
	Asus A639	Open	No	1	2	3/5/2008 7:20:49 PM	administrator
	HP IPAQ hx2795 Pocket PC	Open	No	1	2	3/5/2008 7:32:30 PM	administrator
	Nokia 6300	Open	No	1	2	3/5/2008 7:27:13 PM	administrator
	Canon Digital Rebel XTi	Open	No	2	5	3/5/2008 7:15:51 PM	Jane
	Creative Zen Stone	Open	No	1	2	3/5/2008 7:30:01 PM	administrator
	Motorola V3 Razr	Open	No	1	2	3/5/2008 7:24:47 PM	administrator
	Configurable PC	Open	No	0	0		
	Tiger Extreme PC	Open	No	1	4	3/5/2008 6:57:59 PM	administrator
	iPod Nano	Open	No	1	3	3/5/2008 7:18:43 PM	Mike
	iRiver T60	Open	No	1	3	3/5/2008 6:48:45 PM	Alex
	Nikon D40	Open	No	1	2	3/5/2008 7:06:03 PM	administrator

After choosing a forum group, you can **Edit** (), **Delete** (), **Move up** () and **Move down** () individual forums.



**New thread**

User name:

E-mail:

Subject:

Post:

Signature:

Subscribe to post: ☐

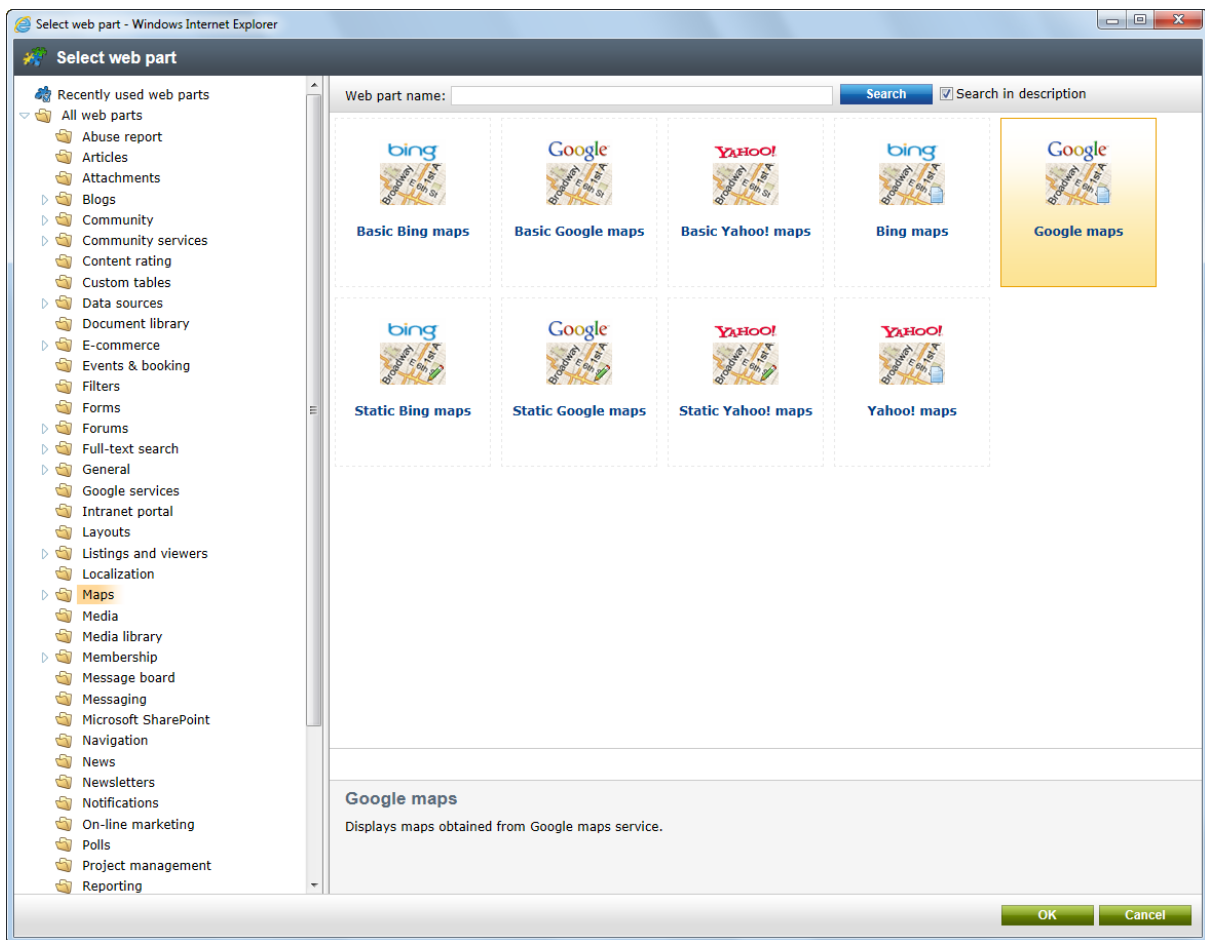
After you choose a forum to edit you can add, edit and delete threads and posts at of the given forum.

For more information please refer to the **Module Forums** chapter in **Kentico CMS Developer's guide**.

## 5.4.7 Geo mapping

The Geo mapping module uses Google Maps to display content on the map. You can use this module to display your offices, your store, your partners or real estates you offer on a map.

For the Geo mapping module implementation, use the **Maps -> Google Maps** web part.

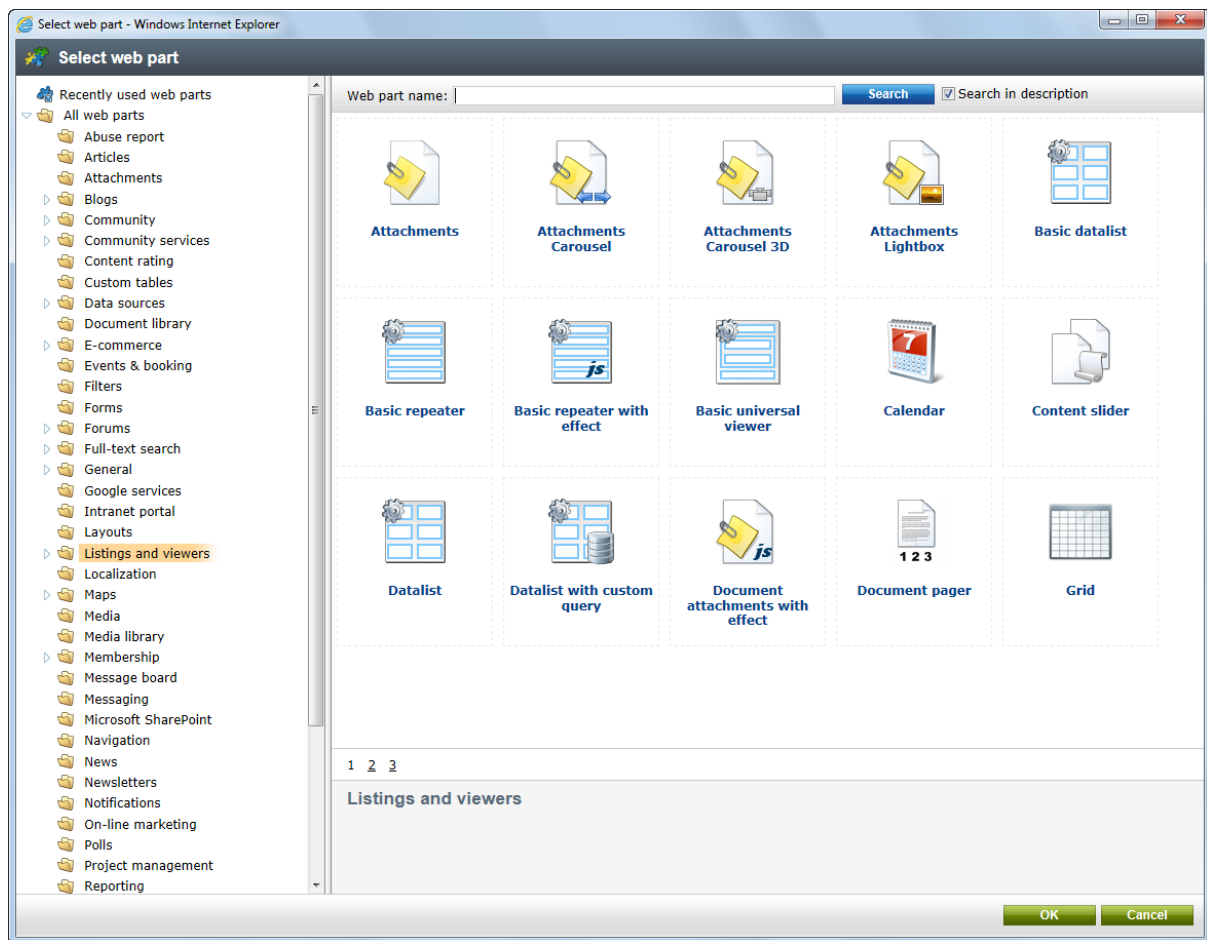


For more details and information about the Geo mapping module, please refer to the Geo mapping chapter in Kentico CMS Developer's Guide.

## 5.4.8 Image Gallery

Using the **Image Gallery** module, you can create easily image gallery pages. The module encompasses three page templates and three web parts suitable for creating image galleries.

For the implementations of **Image Gallery**, use the **Content slider**, **Image gallery** and **Lightbox** web parts.

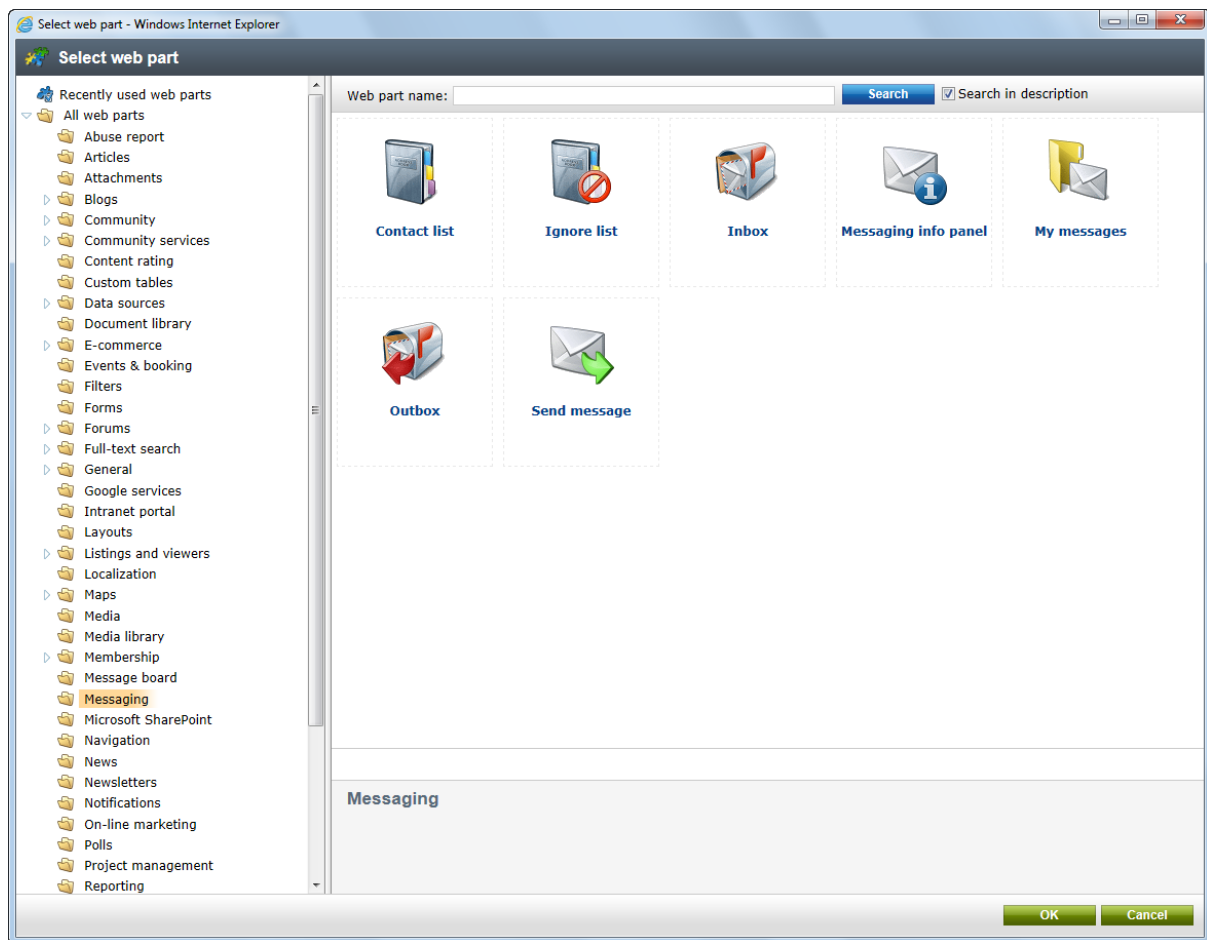


For more details and information about **Image Gallery** please refer to the **Module Image Gallery** chapter in **Kentico CMS Developer's guide**.

### 5.4.9 Messaging

The **Messaging** module allows you to send, receive and manage messages. Its purpose is to provide an internal way of communication with other users of the website.

For the **Messaging** module implementation, use the web parts from the **Messaging** folder in the **Web part selection** dialog.

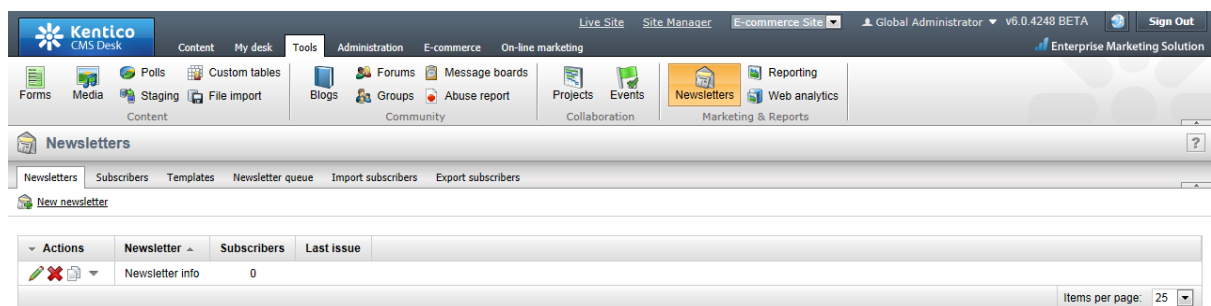


For more details and information about **Messaging** please refer to the **Module Messaging** chapter in **Kentico CMS Developer's guide**.

### 5.4.10 Newsletters

The Newsletter module allows you to author and mail out e-mail newsletters. Though newsletters, you can inform your customers about new products, current offers and discounts and much more. The newsletter can be static or dynamic.

You can edit newsletters at **CMS Desk -> Tools -> Newsletters**.



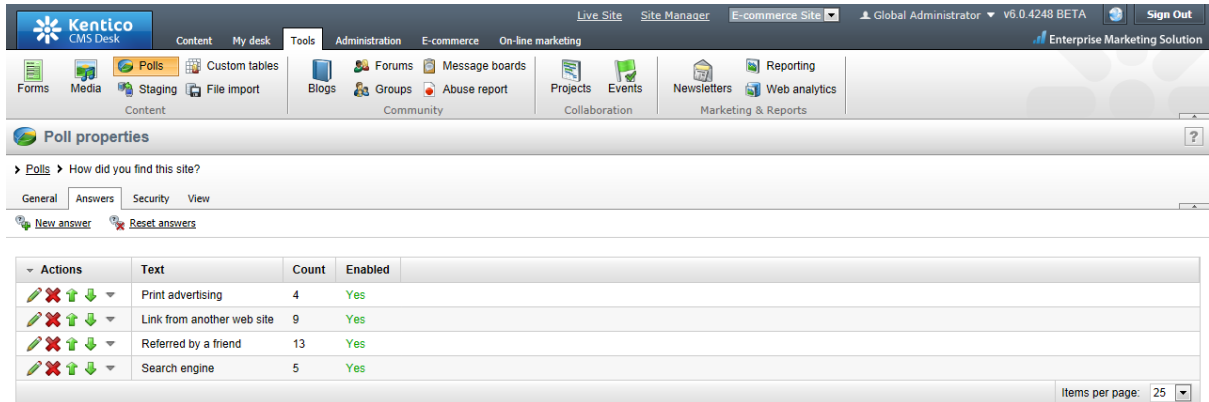
For more details and information about newsletters please refer to the **Module Newsletters** chapter in

Kentico CMS Developer's guide.

### 5.4.11 Polls

The Polls module enables you to create and publish polls with single or multiple answer so that you are able to get vital information about how are your customers and what they want.

You can edit polls at **CMS Desk -> Polls -> <edit poll>**.



After you choose a poll to edit, you can set the poll question at the **General** tab and the poll answers at the **Answers** tab.

For more details and information about polls please refer to the **Module Polls** chapter in **Kentico CMS Developer's guide**.

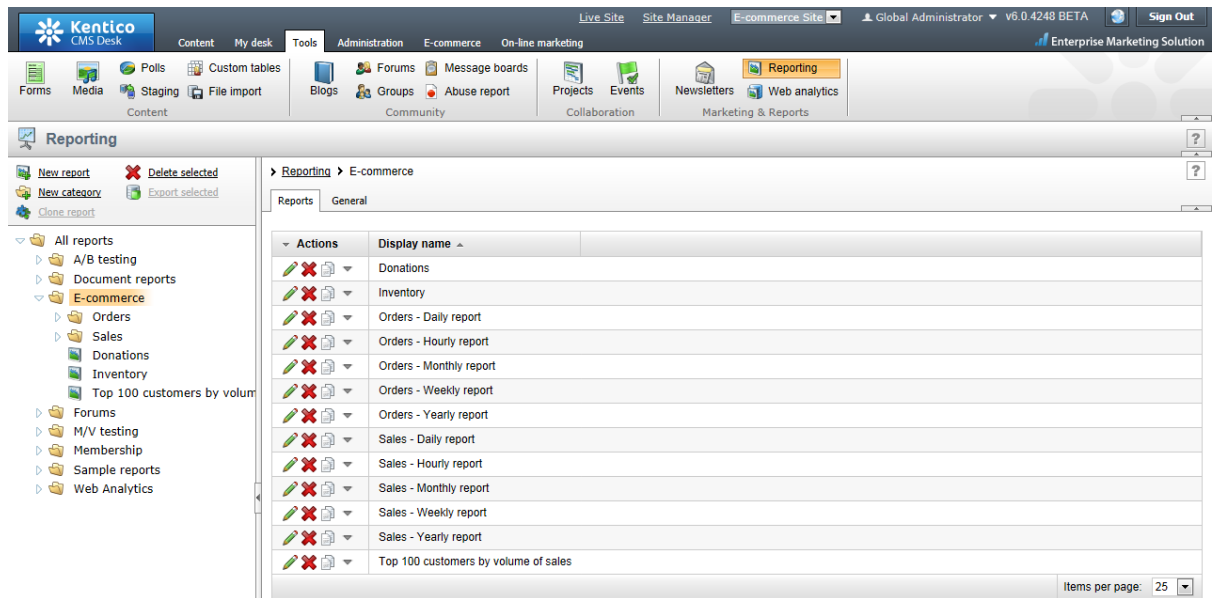
### 5.4.12 Reporting

The Reporting module enables you to create internal reports to watch activity in the Kentico CMS system and on the website. Kentico CMS comes with several pre-defined e-commerce reports that show statistics related to your on-line business.

The following pre-defined e-commerce reports are available:

- Inventory
- Number of orders per day
- Number of orders per month
- Number of orders per year
- Sales per day
- Sales per month
- Sales per year
- Top 100 customers by volume sales

You can edit reports at **CMS Desk -> Tools -> Reporting**.



After choosing a report category, you can **Edit** (✎) or **Delete** (✖) individual reports.

For more details and information about how to set report properties please refer to the **Module Reporting** chapter in **Kentico CMS Developer's guide**.

### 5.4.13 RSS Feeds

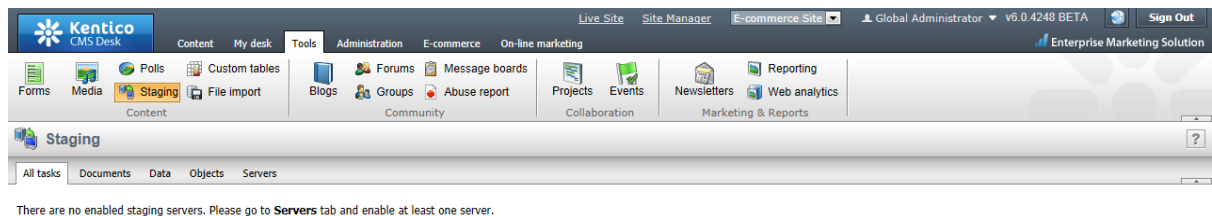
Kentico CMS allows you to publish content using RSS, Atom or XML feeds. This functionality is provided by a set of web parts belonging to the **Syndication** module.

For more details and information about feeds and how to create them for a different types of documents and objects, please refer to the [Modules -> Syndication \(RSS, Atom, XML\)](#) chapter of **Kentico CMS Developer's Guide**.

### 5.4.14 Staging

The **Staging** module allows you to store separately the content in development, staging (editing) and production (live) environment. It allows you to easily transfer document and object changes to another server or make a complete synchronization of documents and objects from one server to another.

You can manage content staging at **CMS Desk -> Tools -> Staging**.



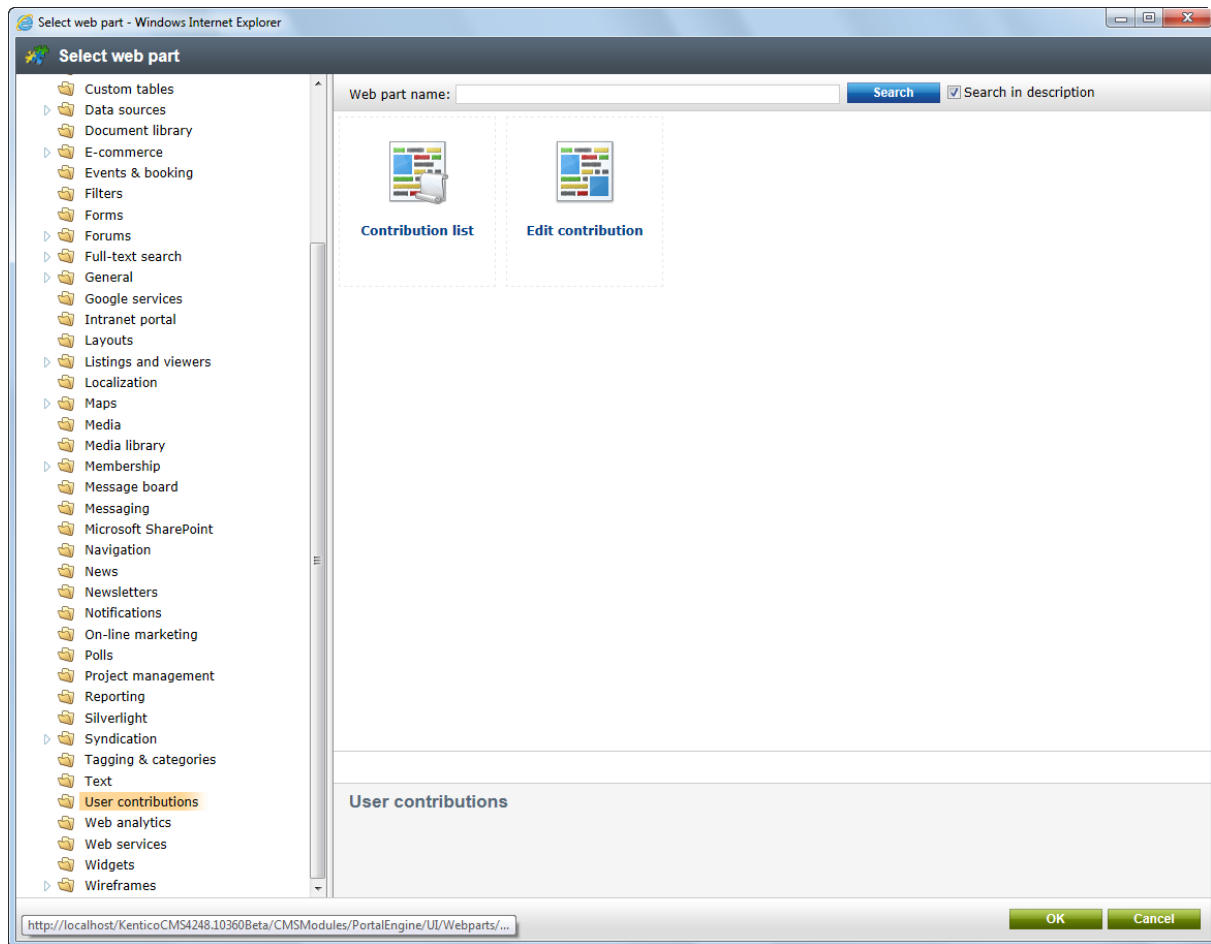
For more details and information about the Staging module please refer to the **Modules -> Staging** chapter in **Kentico CMS Developer's Guide**.



### 5.4.15 User Contributions

The **User Contributions** module allows you to create content editing interface for site members. Using this interface, the website visitors are able to create, edit and delete content, even if they are not editors and cannot access Kentico CMS Desk.

For the **User Contributions** module implementations, use the **Contribution list** and **Edit contribution** web part.

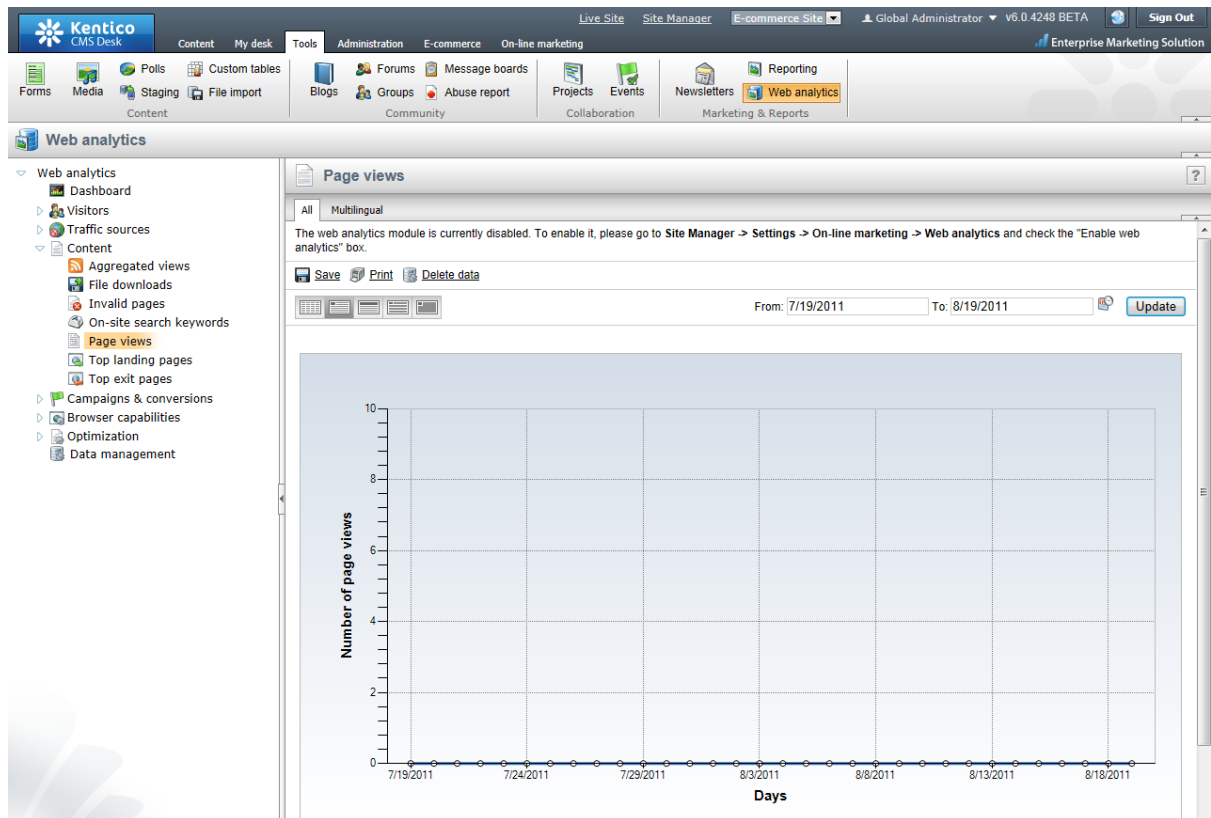


For more details and information about **User Contributions** please refer to the **Module User Contributions (Wiki)** chapter in **Kentico CMS Developer's guide**.

### 5.4.16 Web Analytics

The Web Analytics module enables you to track and analyze website visits, page views, file downloads and other metrics of the website.

You can modify view the Web Analytics reports at **CMS Desk -> Tools -> Web Analytics**.

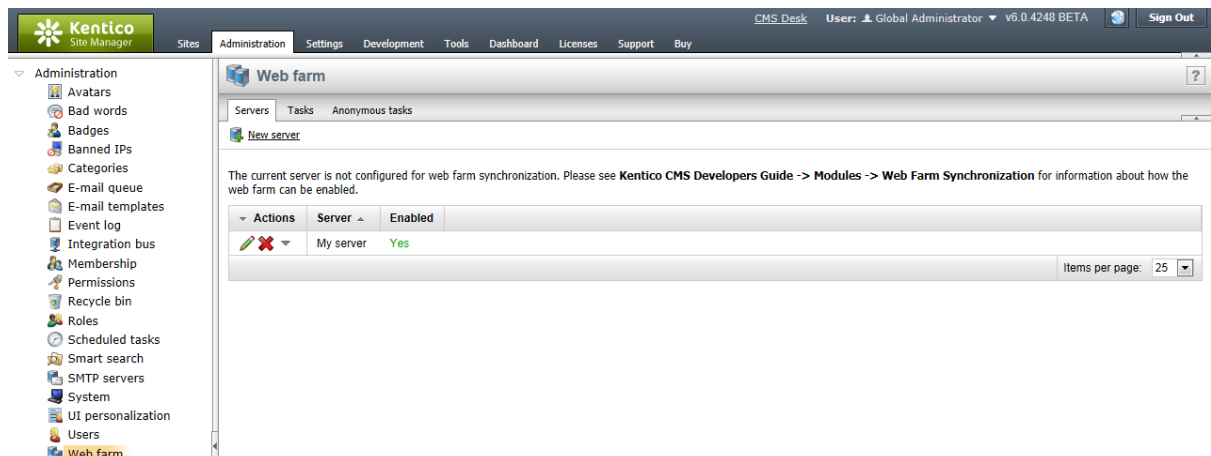


For more details and information about **Web Analytics** please refer to the **Module Web Analytics** chapter in **Kentico CMS Developer's guide**.

### 5.4.17 Web Farm synchronization

Kentico CMS offers native web farm support. The web farm support provides the synchronization of the changes made to the site settings on one of the servers to all the other servers and provides synchronization of the files uploaded to the site to all other servers.

You can manage web farm servers at **CMS Site Manager -> Administration -> Web farm**.



For more details and information about **Web Farm synchronization** please refer to the **Module Web Farm synchronization** chapter in **Kentico CMS Developer's guide**.

## 5.5 Adding items to the shopping cart

The items can be added to the shopping cart by calling the page with the **Shopping cart** web part with following URL parameters:

- **productId** - the ID of the product stored in the NodeSKUID database field (in the CMS\_Tree table) or SKUID (in the COM\_SKU table)
- **quantity** - the number of product units to be added to the shopping cart
- **options** - the string of the selected product option IDs separated by commas

It means your link will typically look like this:

```
<a href="/shoppingcart.aspx?productId=10&quantity=1&options=12,24">Add to cart</a>
```

This link has been generated by the following **ShoppingCartItemSelector** control. All you have to do is adding the control to transformation and setting its properties accordingly.

### 1. Product detail

```
<%@ Register Src="/CMSModules/Ecommerce/Controls/ProductOptions/ShoppingCartItemSelector.ascx" TagName="CartItemSelector" TagPrefix="ucl" %>

<ucl:CartItemSelector id="cartItemSelector" runat="server" SKUID='<%# ValidationHelper.GetInteger(Eval("SKUID"), 0) %>' SKUEnabled='<%# ValidationHelper.GetBoolean(Eval("SKUEnabled"), false) %>' AddToCartImageButton="AddToCart.gif" ShowUnitsTextBox="true" ShowProductOptions="true" AddToWishlistImageButton="AddToWishlist.gif" />
```

The control has the **ShowUnitsTextBox** and **ShowProductOptions** properties set to **true**. Therefore, the product options and units text box are visible on the website. Please note that **AddToWishlistImageButton** properties is set as well. Thus, the **Add to wishlist** button is displayed as well.

Please note: the `<%@ Register ...>` part needs to be placed at the beginning of the whole transformation code; otherwise, the transformation will not be functional

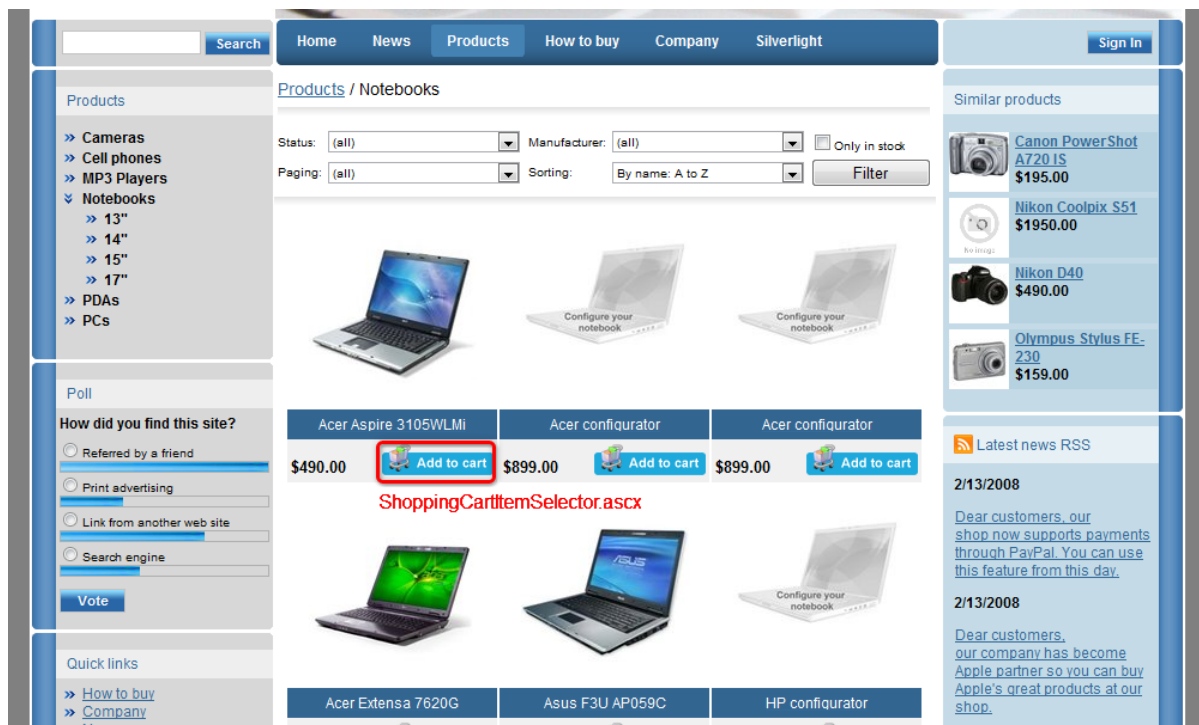
The screenshot shows the 'Acer configurator' product page. The sidebar on the left contains navigation links for Products (Cameras, Cell phones, MP3 Players, Notebooks, PDAs, PCs), a Poll, and Quick links. The main content area displays the product name, a list of parameters (Processor type, Display type, Resolution, Graphics card, Memory type, Memory size, Optical drive, Hard Drive, Wireless LAN, Bluetooth, Infraport, Battery type, Weight(g), Operating system, Accessories), and a price section. A red box highlights the 'ShoppingCartItemSelector.ascx' control, which shows product options for Intel CPU, HDD, and DDR2 modules, along with a total price and an 'Add to cart' button. The right sidebar contains a 'Latest news' section and a 'Newsletter subscription' form.

## 2. Product list

```
<%@ Register Src="~/CMSModules/Ecommerce/Controls/ProductOptions/
ShoppingCartItemSelector.ascx" TagName="CartItemSelector" TagPrefix="uc1" %>

<uc1:CartItemSelector id="cartItemSelector" runat="server" SKUID='<%#
ValidationHelper.GetInteger(Eval("SKUID"), 0) %>' SKUEEnabled='<%#
ValidationHelper.GetBoolean(Eval("SKUEEnabled"), false) %>' AddToCartImageButton
="AddToCart.gif" ShowUnitsTextBox="false" ShowProductOptions="false" />
```

This control has **ShowUnitsTextBox** and **ShowProductOption** properties set to **false**. Therefore, the product options and units text box are not visible on the website.



### Complete list of shopping cart item selector properties:

<b>SKUID</b>	Product ID
<b>SKUEnabled</b>	Indicates whether the current product is enabled. If set to <b>true</b> , the button/link for adding product to the shopping cart is rendered, otherwise it is not rendered.
<b>AddToCartImageButton</b>	File name of the image which is used as a source for image button to add product to the shopping cart, default image folder is '~/App_Themes/<stylesheet name>/Images/ShoppingCart/'.
<b>AddToCartLinkText</b>	String (simpletext or localizable string) of the link to add product to the shopping cart.
<b>ShowUnitsTextBox</b>	Indicates if text box for entering number of units to add to the shopping cart should be displayed. If it is hidden, number of units is equal to DefaultQuantity.
<b>ShowProductOptions</b>	Indicates if product options of the current product should be displayed. If <b>ShowProductOption</b> is set to <b>false</b> and a customer clicks <b>Add to shopping cart</b> for a products with specified product options, they are redirected to product detail where they are required to choose product option before the product can be added to the shopping cart.
<b>ShowDonationProperties</b>	Indicates if donation properties (text box for entering donation amount and check box for marking donation as private) are displayed for donation products.
<b>DefaultQuantity</b>	Default quantity when adding product to the shopping cart, it is set to 1 by default.

<b>ShowTotalPrice</b>	If true, total price will be shown in the bottom section of the product options list.
<b>ShowPriceIncludingTax</b>	If true, the total price enabled by the <b>ShowTotalPrice</b> property and product option prices will be displayed with tax added. If false, tax will not be included in the total price.

## Displaying total product price

If you enable the **ShowTotalPrice** property of the shopping cart item selector, a total price of the product including prices of product options will be displayed at the bottom section of the product options list, as you can see in the screenshot below. The price can either be displayed with or without tax, based on the value of the **ShowPriceIncludingTax** property of the control. Displayed prices of the product options are affected by the value of this property too.

Our price: \$899.00  
Without tax: \$899.00

**Notebooks - AMD CPU**

AMD X2 1800 Mhz (+ \$150.00)

**Notebooks - HDD**

Samsung 80GB SATA (+ \$80.00)

**Notebooks - DDR2 modules**

512MB DDR2 (+ \$35.00)

Our world model was built specifically to investigate five major trends of global concern – accelerating industrialization, rapid population growth, widespread malnutrition, depletion of nonrenewable resources, and a deteriorating environment.

**Total price (without tax): \$1164.00**



### Shopping cart URL

If you use **ShoppingCartItemSelector**, the shopping cart URL is taken from the **Site Manager -> Settings -> E-commerce -> Shopping cart URL** value.

## 5.6 Adding items to the wish list

The items can be added to the wish list by calling the page with the **Wishlist** web part with following URL parameter:

- **productId** - the ID of the product stored in the NodeSKUID database field (in the CMS\_Tree table) or SKUID (in the COM\_SKU table)

It means your link will typically look like this:

```
<a href="~/wishlist.aspx?productId=10">Add to wish list</a>
```

In the transformations, you can use the following method to display the link to the wish list:

```
<%# EcommerceFunctions.GetAddToWishListLink(Eval("NodeSKUID")) %>
```

You can use the **ShoppingCartItemSelector** control to display the **Add to wishlist** button as well. All you have to do is to set one of the following properties.

<b>AddToWishlistImageButton</b>	File name of the image which is used as a source for image button to add product to the wish list, default image folder is '~/App_Themes/<stylesheet name>/Images/ShoppingCart/'.
<b>AddToWishlistLinkText</b>	String (simpleset text or localizable string) of the link to add product to the wish list.



#### Wish list URL

If you use **ShoppingCartItemSelector** or functions like `GetAddToWishListLink`, the wish list URL is taken from the **Site Manager -> Settings -> E-commerce -> Wish list URL** value.

## 5.7 Displaying product price

### Displaying product price on the website

The price is stored in the `COM_SKU.SKUPrice` field. The following method allows you to display the price **in your transformation** formatted according to the formatting string of the currently selected currency specified at **CMS Desk -> E-commerce -> Configuration -> Currencies -> <edit currency> -> Currency formatting string**:

```
EcommerceFunctions.GetSKUFormattedPrice(bool discounts, bool taxes)
```

- **bool discounts** - indicates if price is displayed with the applied customer's discount level.
- **bool taxes** - indicates if price is displayed with the applied taxes. Please note: computation of all product taxes is time-consuming. Therefore, the result is stored in the cache as specified at **CMS Site Manager -> Settings -> website -> Cache content (minutes)**.

All product prices are displayed in currency chosen according to the following priorities:

1. the currency of the shopping cart
2. the preferred currency of a customer
3. the default currency

Therefore, if the currency of the shopping cart is not set yet (the shopping cart hasn't been created so

far), all product prices are displayed in the preferred currency of the given customer. If the customer doesn't specify the preferred currency, all product prices are displayed in the default currency.

## 5.8 Getting product URL

In some cases, you may need to get the URL of the document to which some product is assigned. Knowing product URL can be useful when you work with specific web parts such as the **Product datalist** which is using the e-commerce product (SKU) properties to display products instead of the standard CMS documents.

For this, you need to insert the **GetProductUrl** method with the following syntax into your transformation:

```
<%# EcommerceFunctions.GetProductUrl(Eval("SKUID")) %>
```

As a result, you will get the product URL in the `/CMSPages/GetProduct.aspx?productId=125` format.

Alternatively, you can provide the method with values of the **SKUGUID** and **SKUName** columns:

```
<%# EcommerceFunctions.GetProductUrl(Eval("SKUGUID"), Eval("SKUName")) %>
```

This syntax returns the URL in the `~/getproduct/<skuguid>/<safe_skuname>.<extension>` format.

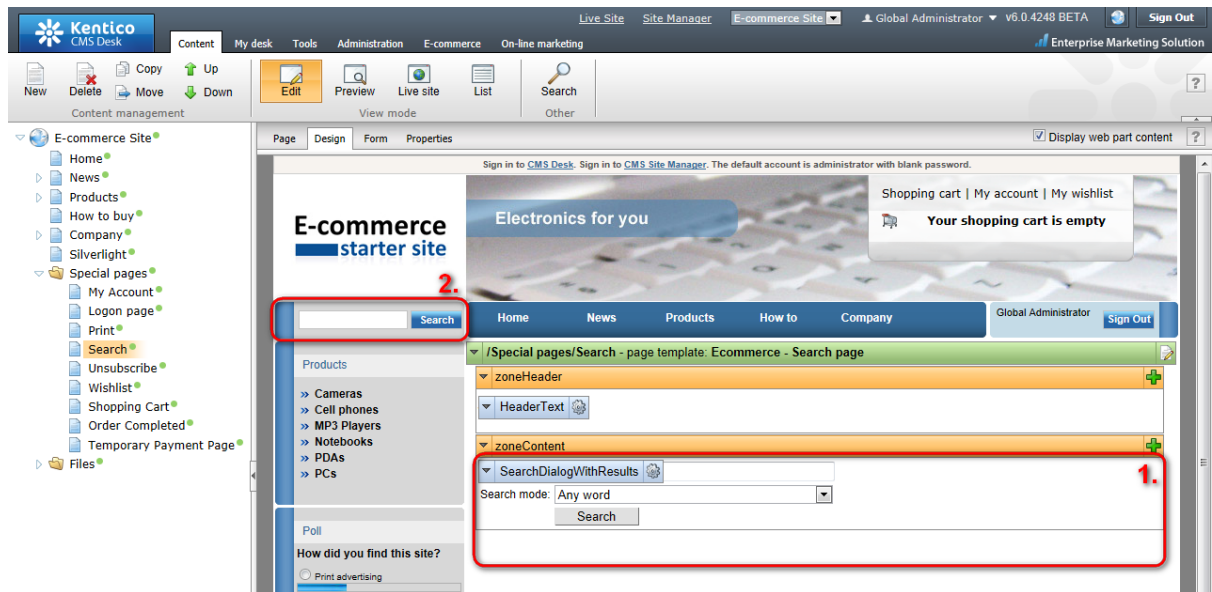
## 5.9 Searching

Kentico CMS comes with the built-in **Smart search** module that enables you to search for the given expression in the content of your website.

The search functionality is ensured by the **Special pages -> Search** page, where the **Smart search dialog with results** web part (1. in the screenshot) is placed. This web part allows searching and displays the search results.

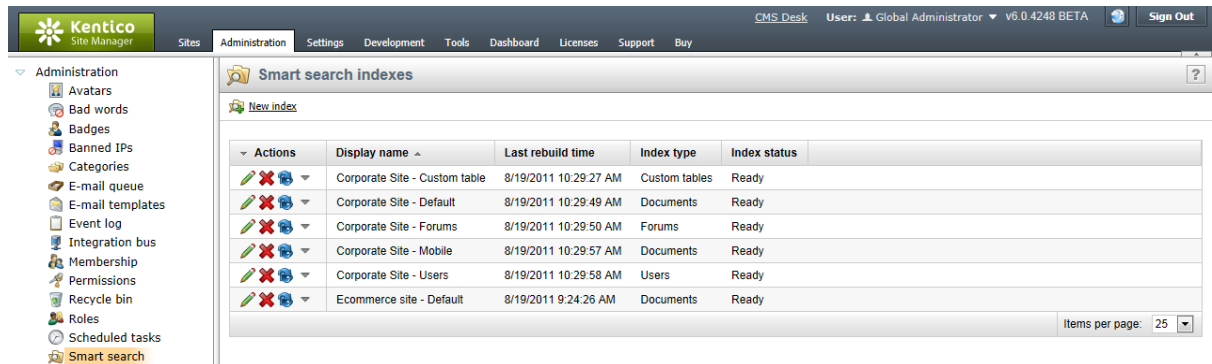
The **Smart search box** web part (2. in the screenshot) on the master page is connected to the **Smart search dialog with results** web part by means of the **Search results page URL** property. When a user searches using the **Smart search box**, they are redirected to the **Special pages -> Search** page where the **Smart search dialog with results** web part displays the results.





The Smart search module provides index-based search. You need to specify which index should be searched using the **Indexes** property of the **Smart search dialog with results** web part.

Indexes can be created and managed in **Site Manager -> Administration -> Smart search**. For detailed information about Kentico CMS Smart search module, please refer to [Kentico CMS Developer's Guide -> Modules -> Smart search](#).



# Part

---



VI

**Configuration settings**

---

## 6 Configuration settings

### 6.1 Configuration overview

When you start using the **E-commerce module** it's recommended that you configure it in the following order:

1. Enable the **E-commerce module** at **Site Manager -> Settings** if it's not enabled yet.  
More details: [Enabling the e-commerce module](#)
2. Configure **access rights** to the **E-commerce module** interface.  
More details: [Security](#)
3. Set up **countries and states**.  
More details: [Countries and states](#)
4. Configure **store settings**.  
More details: [Store settings](#)
5. Set up **departments** and assign **store administrators** to their **departments**.  
More details: [Departments](#)
6. Configure **currencies**. If you plan to use multiple currencies, specify the **exchange rates**.  
More details: [Currencies](#), [Exchange rates](#)
7. Set up **shipping options**.  
More details: [Shipping options](#)
8. Set up **payment methods**.  
More details: [Payment methods](#)
9. Configure **tax classes**.  
More details: [Tax classes](#)
10. Set up order processing **workflow**.  
More details: [Order status](#)
11. Configure **public and internal product status**.  
More details: [Public status](#), [Internal status](#)
12. Modify the **invoice/receipt** design.  
More details: [Invoice](#)
13. Create documents with product details at **CMS Desk -> Content** and mark the documents as **products**. Enter the product details.  
More details: [Products](#)
14. Make a testing order for each **payment option**.

Site separation in the E-commerce module, i.e. how to allow and use site-specific objects and global objects shared across all your sites, is described in detail in the [Site separation](#) chapter.

## Possible configurations

The **E-commerce module** allows you to manage products in two basic modes:

- **standard mode** - you manage the products in **Kentico CMS**. Each product is represented by a combination of document that contains product information displayed to the visitor and a product record that contains standard product information, such as product price, sizes, taxes, etc.
- **custom product provider** - you manage the products in an external system and publish them inside Kentico CMS-based website. You can learn more on custom providers in chapter [Developing custom providers](#).

## 6.2 Orders

The Orders can be managed at **Kentico CMS Desk -> Tools -> E-commerce -> Orders** of the e-commerce module.

### General properties

- **Order ID** - unique identifier of the order
- **Date** - date and time when the order was placed
- **Invoice number** - the assigned invoice number (it can be changed on **Invoice** tab)
- **Status** - order processing status
- **Customer** - customer who made this order
- **Company address** - address of the company which made the order
- **Order note** - notes added by customer or by store managers


### Shipping

- **Shipping option** - chosen shipping option
- **Shipping address** - chosen shipping address
- **Tracking number** - tracking number of the order

### Billing

- **Payment method** - chosen payment method
- **Currency** - currency used in the invoice
- **Billing address** - chosen billing address
- **Payment result** - payment results as described in the [Payment results](#) chapter

### Items

Here you can view the list of individual order items and modify the order. You can also **Edit**  properties of the order items if the given order **is not marked as paid**. However, editing is enabled only if the *web.config* file in your current web project folder contains the **CMSEnableOrderItemEditing** key with its value set to *TRUE*. Please refer to the [Web.config settings](#) chapter for more information.

### Invoice

Here you can re-generate or print the invoice and change the invoice number manually.

## History

Here you can see the history of order processing.

## 6.3 Customers

You can manage customers at **Kentico CMS Desk -> Tools -> E-commerce -> Customers**. The customer profile can be created either manually in this administration interface or it can be created automatically when the customer registers during the purchase process.

You can create new customers here. While creating a new customer, you can choose to provide them with **User name** and **Password** straight away and create the user account for them.

### General customer properties

- **First name** - customer's first name
- **Last name** - customer's last name
- **Company** - company name (optional)
- **Organization ID** - registration ID of the customer's company
- **Tax registration ID** - tax registration ID of the customer's company
- **E-mail** - customer's e-mail address
- **Phone** - customer's phone number
- **Fax** - customer's fax number
- **Preferred currency** - the currency in which the prices are displayed when the customer signs in
- **Preferred payment method** - customer's preferred payment option (automatically set based on their last purchase)
- **Preferred shipping option** - customer's preferred shipping option (automatically set based on their last purchase)
- **Country/state** - customer's main country and state
- **Enabled** - if you uncheck the box, the customer profile will not be listed in the customer list, while still being stored in the database for your records and to keep customer's purchase history.
- **Discount level** - discount level applied to the customer's orders

### Addresses

Each customer can have several addresses for billing and shipping stored in their profile. Every address has the following properties:

- **Personal or company name** - name line in the address
- **Address lines** - address line 1 and address line 2
- **City** - city
- **ZIP code** - zip/postal code
- **Country** - country
- **Phone number** - phone number
- **Enabled** - indicates if the address should be offered to the client. If you set it to disabled, it will no longer be displayed, but it will be kept in the database for your records and to keep customer's purchase history.
- **Use as shipping address** - indicates if the address should be offered to customer as a shipping address
- **Use as billing address** - indicates if the address should be offered to customer as a billing address
- **Use as company address** - indicates if the address should be offered to customer as a company

address

## Orders

- the list of orders made by this customer

## Credit

The customer credit history.

The customer credit is stored in the default currency. If a payment is made in a currency different from default, the payment amount is converted to the default currency and the customer credit is decreased by this amount - a new credit event is created.

For information about enabling payments with the customer credit please refer to the [Customer credit](#) chapter.

The displayed total value of the customer credit is sum of all credit events.

## Newsletters

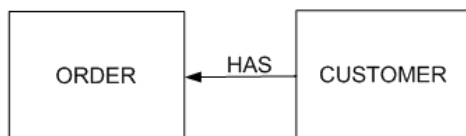
Here you can select newsletters to which a customer is subscribed.

For more information about newsletters please refer to the **Newsletter module overview** chapter in **Kentico CMS Developer's Guide**.

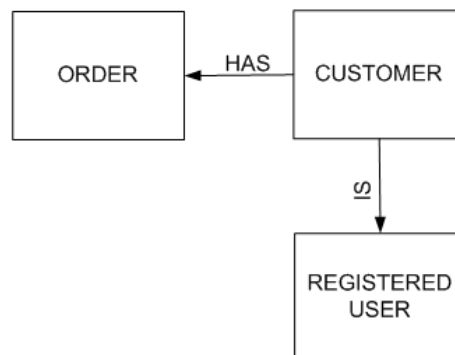
## Login

Here you can manage user accounts that users use to sign in to your e-commerce site (if they are registered users, not only anonymous site visitors)

A. Customer-order relation if an order is made by an anonymous customer.

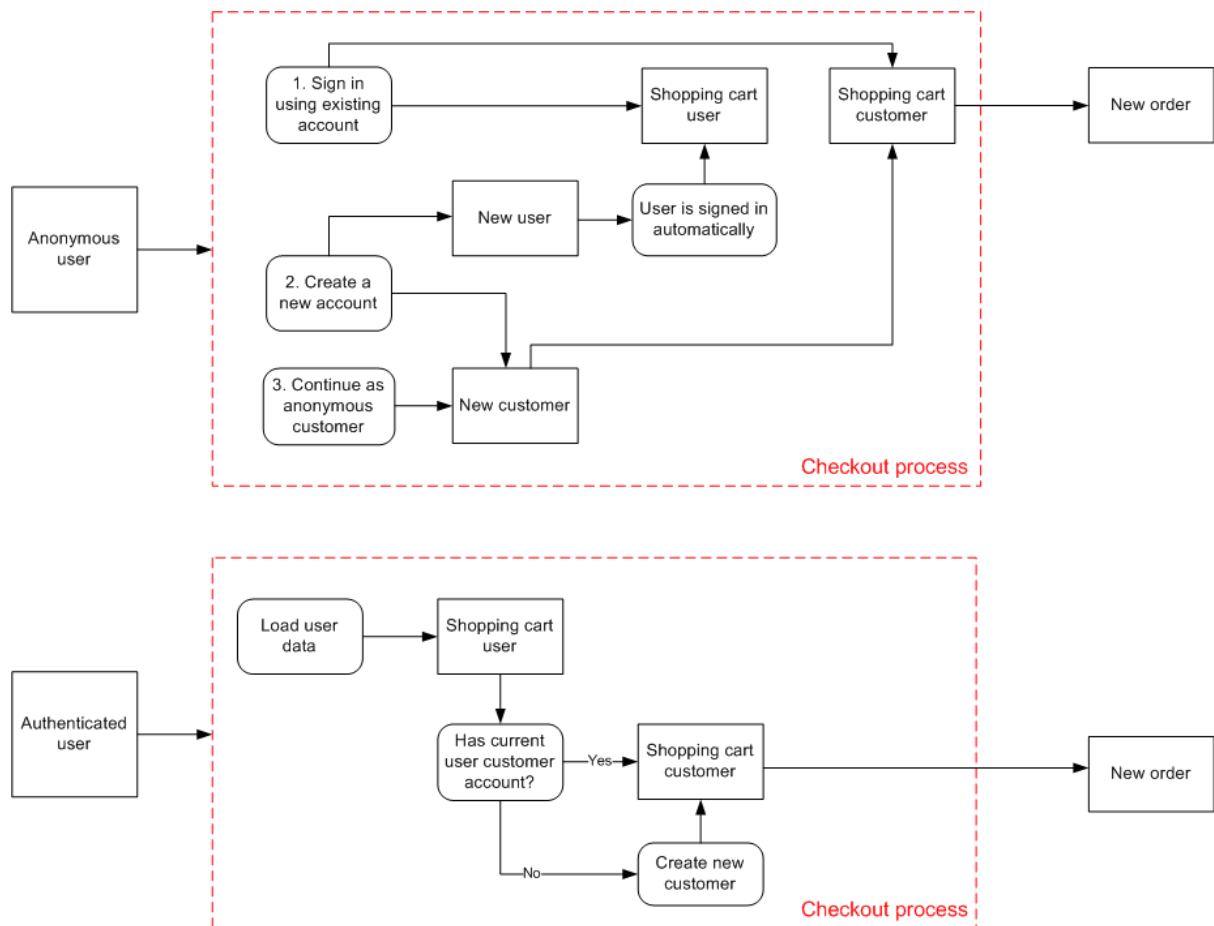


B. Customer-order-user relation if an order is made by a registered user.



You can create and edit users at **CMS Desk -> Administration -> Users**.

The following diagram shows the checkout process for both an anonymous and authenticated user in greater detail.



## 6.4 Products

Kentico CMS allows you to integrate the e-commerce data with documents. You can mark any document as a product under one of the available [product types](#) and define its product properties.

There are two ways how you can enter the products:



1. Go to **CMS Desk -> Content -> Product**, check the **Mark document as product** box and choose **Create a new product**.
2. Go to **CMS Desk -> Tools -> E-commerce -> Products** and enter all your product details. Then go to **CMS Desk -> Content -> Product**, check the **Mark document as product** and select an existing product.

### General tab

Every product has the following properties:

#### General info:

- **Product name** - product name displayed on the website and in the shopping cart.
- **Product type** - defines the type of product. You can choose from five [different types](#), specifically the general type (i.e. product) and four specialized types derived from this general type (i.e. membership, e-product, donation and bundle).

- **Product number** - product number (serial number or SKU number) for your records.
- **Description** - product description used for special product listings.
- **Price** - product price in the main currency.
- **Department** - the department that is responsible for this product.
- **Manufacturer** - the manufacturer of the product.
- **Supplier** - the supplier of the product.
- **Image** - URL of the product image. When a file is uploaded in the field, two icons are displayed next to it: 1. **Edit** () - if the file is an image, clicking the icon opens it in the built-in [image editor](#). If the file is not an image (which would not make sense as no product image could be displayed), the [metadata editor](#) is opened after clicking the icon. 2. **Delete** () - removes the file from the field.

### Status:

- **Enabled** - indicates if product can be added to the shopping cart.
- **Public status** - status of the product displayed to site visitors - e.g. Available immediately.
- **Internal status** - status of the product displayed only to the site owners - e.g. Products going off-sale. Invisible to the site visitors.

### Shipping:

- **Needs shipping** - indicates if the product needs shipping.
- **Package weight** - weight in your chosen unit.
- **Package height** - height in your chosen unit.
- **Package width** - width in your chosen unit.
- **Package depth** - depth in your chosen unit.

### Inventory:

- **Sell only if items are available** - indicates if customers can purchase only quantity that is in stock (checked box).
- **Available items** - number of items available in stock.
- **Availability (days)** - number of days required for processing the order (e.g. the order may require 1 day so that you get the product from the distributor).
- **Max items in one order** - defines the maximum number of items in one order.

### Custom fields tab

Here you can see list of the custom fields available for all products. You can specify the custom fields at **CMS Site Manager -> Development -> System tables -> Ecommerce - SKU**.

Please note that the **Custom fields tab** is visible only if there are some custom fields defined.

### Tax classes tab

You can specify **Tax classes** for every product - you can choose any number of taxes that apply to the given product.

### Volume discounts

On the **Volume discounts** tab in the **Product properties** dialog, you can define discount that applies to



the ordered product when the customer purchases specified amount of the product.

For every volume discount line, you need to enter:

- **minimum amount** - number of ordered items from which the discount applies
- **discount value** - relative or absolute discount for the given minimum amount

When the customer updates the number of ordered products, the discount is automatically calculated and the customer is displayed with the final price. For more details please refer to the [Discounts overview](#) chapter.

## Options tab

Here you can specify option categories (e.g. size of clothes) - you can choose any number of categories that assign to the given product. Customers are asked to choose their product configuration before the product is added to the shopping cart.

For further information please refer to the [Product options](#) chapter.



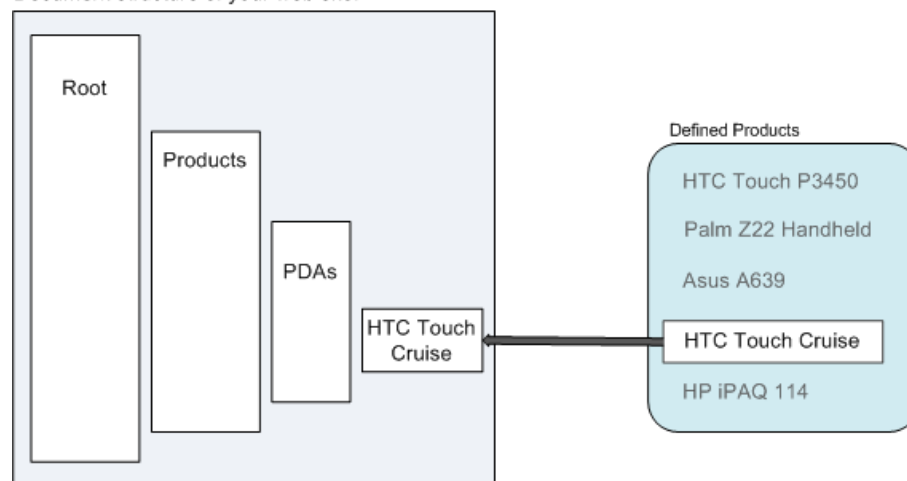
### Assigning product to document

In Kentico CMS, products are defined separately from the documents. When you want to display a product on your website, you firstly choose a document that will act as a container for the product and then you assign the given product to the chosen document.

While creating document, you can also choose to create document and product at the same time and product will be assign to the given document automatically.

The fact that products are defined separately enables them to be independent of the document structure of a specific website. This allows you to use the same product on multiple sites.

Document structure of your web site.



**Please note:** Should you want your products to be created automatically or to get its product field value directly from the document field value, please refer to the [Mapping document fields to product properties](#) chapter.

## 6.5 Product options

You can manage product options at **Kentico CMS Desk -> Tools -> E-commerce -> Product options**.

Here you can either create a new option category by clicking the  **New category** link or **Edit** () and **Delete** () the existing one. When creating a new category, from the **Type** selector choose either **Category with selectable options** to create a product option category containing selectable options or choose **Category with one text option only** to create a category containing a text product option.

### General

Every option category with selectable options has the following properties:

**Category name** - category name displayed to the customer

**Selection type** - type of selection, a customer use to choose the product option

**Display price** - if enabled, product option prices will be displayed next to product options

**Default option(s)** product options which are selected by default before a customer chooses their own one(s)

**Description** - the description of the category as it is visible to customers

**Default record text** - the caption of the default option, which is added to the product options collection

**Enabled** - indicates if the category is enabled

Every option category with a text product option has the following properties:

**Display name** - the name of the category displayed to the customer.

**Code name** - the name of the category displayed in the code.

**Text type** - defines the type of text; either a text box or text area can be selected.






**Display price** - if enabled, product option prices will be displayed next to product options.

**Default text** - the default text of the text product option.

**Description** - the description of the category as it is visible to the customers.

**Enabled** - indicates if the category is enabled.

### Options

Here you can either add a new product option to the current option category by clicking the  **New product option** link or you can **Edit** () or **Delete** () existing options or change their succession () () within the current option category. Please note that a text product option cannot be deleted and you cannot add more options to the given category.

**Product options** have the same properties as **Products** with the exception of the volume discounts which cannot be assigned (as well as discount coupon) to a product option.

On contrary with the product price, the product option price can be negative. The product option price is the amount that is either added to or subtracted from the given product price. The product option is sold as an addition to the given product, not as its replacement. If certain discount level applies to the given customer, this discount level applies for the chosen product option as well.

## 6.6 Manufacturers

The **Manufacturers** dialog at **Kentico CMS Desk -> Tools -> E-commerce -> Manufactures** allows you to manage the list of manufacturers who make the products you sell. This information is optional, so you do not need to supply it. You can use it for your internal records or you can display it on your website (e.g. you can display a link to the manufacturer's website where clients can find more details).

Every manufacturer has the following properties:

- **Manufacturer display name** - manufacturer name
- **Homepage** - URL of the internet home page of the manufacturer
- **Enabled** - indicates if the given manufacturer can be used for newly created products

## 6.7 Suppliers

You can define your suppliers at **Kentico CMS Desk -> Tools -> E-commerce -> Suppliers**. This information is typically used only for your records. This information is optional and you do not need to enter it.

Every supplier has the following properties:

- **Supplier display name** - name of the supplier
- **Supplier e-mail** - supplier e-mail contact
- **Supplier phone** - supplier phone contact
- **Supplier fax** - supplier fax contact
- **Enabled** - indicates if the supplier is active and can be used for newly added products

## 6.8 Discount coupons

You can define discount coupons at **Kentico CMS Desk -> Tools -> E-commerce -> Discount coupons**. The discount coupon is represented by a special code that is used by the customer on the Shopping Cart page.

Every discount coupon has the following properties:

- **Discount coupon name** - friendly name of the discount displayed to customers and store managers
- **Discount coupon code** - the coupon code that will be used by the customer during the purchase
- **Discount value** - you can choose between absolute and relative discount and you can enter the discount value in the box below
- **Valid from, Valid to** - time period when the discount can be applied

On the **Products** tab, you can choose which products the discount should be applied to or not applied to. Please note that you can assign a discount coupon only to the given product, not to its product options. **Discount coupons** cannot be defined for **Product options**.

For more details please refer to the [Discounts overview](#) chapter.

## 6.9 Discount levels

You can define discount levels that apply to all products at **Kentico CMS Desk -> Tools -> E-commerce -> Discount levels**. Every customer can be assigned with a discount level. For example, you can assign all Gold Partners with 30% discount on all products they purchase on your website.

The discount level can be defined on the **Discount levels** tab. A discount level has the following properties:

- **Display name** - the discount level name displayed to the users
- **Code name** - the discount level name used in the code
- **Value** - percentage value of the discount
- **Valid from/to** - time period when the discount level is applied. If you leave both values empty, the discount is time-unlimited.
- **Enabled** - indicates if the discount level should be used

On the **Departments** tab, you can choose the departments on which the discount level can be applied.

Once you define the discount levels, you need to assign them to the customers in the **Customer properties** dialog, in the **Discount level** drop-down list.

For more details please refer to the [Discounts overview](#) chapter.

## 6.10 Reports

At **Kentico CMS Desk -> Tools -> E-commerce -> Reports** you can display, save and print following e-commerce statistics:

- **Numbers of orders per year** - number of orders made on the website in one year
- **Numbers of orders per month** - number of orders made on the website in one month
- **Number of orders per day** - number of orders made on the website in one day
- **Sales per year** - sales in one year in your default currency
- **Sales per month** - sales in one month in your default currency
- **Sales per day** - sales in one day in your default currency
- **Inventory** - the inventory list of all products defined in the **Products** tab
- **Top 100 customers by volume of sales**

For more information, please refer to the **Module Reporting** chapter in **Kentico CMS Developer's guide**.

## 6.11 Configuration

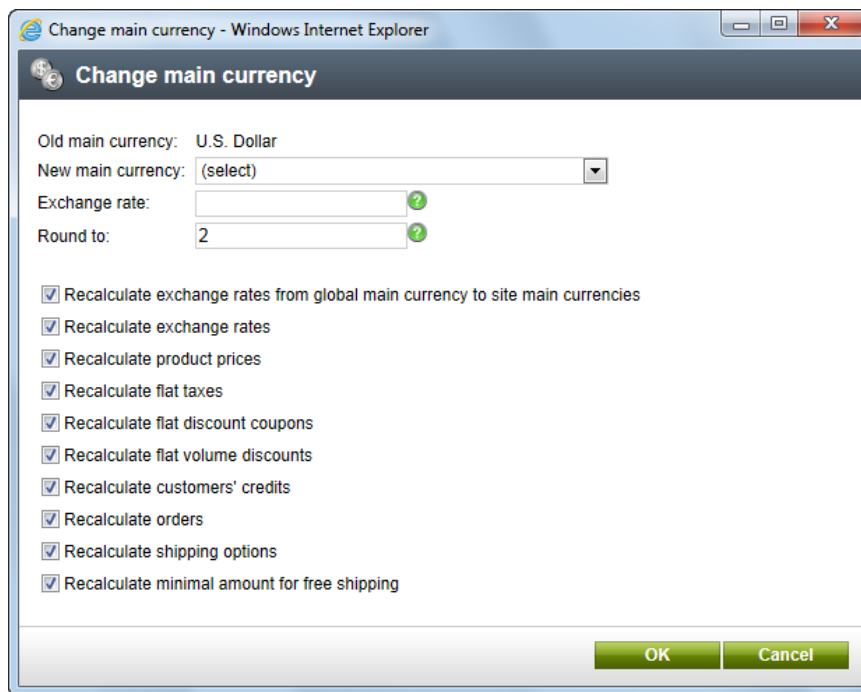
### 6.11.1 Store settings

Go to **Kentico CMS Desk -> E-commerce -> Configuration -> Store Settings** and set the following values:

#### General tab

#### Currencies

- **Main currency** - the default currency used as base for calculation of exchange rates. After clicking the **Change** button a new dialog appears enabling you to select items which should be recalculated according to exchange rate to the new main currency.








The description of other settings located on the **General** tab can be found in the [Sitemanager settings](#) topic.

## E-mails tab

The description of the settings located on the **E-mails** tab can be found in the [Sitemanager settings](#) topic.

## Checkout process tab

Here you can either create a new step in the checkout process by clicking the  **New step** link or you can **Edit** () or **Delete** () the existing steps or change their succession () (). You can also generate a default process using the **Generate default process** link or you can generate the process from global settings using the **Generate from global process** link. However, in both these cases, your current process will be lost.

For more information about purchase process please refer to the [Customizing the purchase process](#) chapter.

## Global objects tab

On this tab you can enable the use of E-commerce module [global objects](#) for the current site. More details on how to perform this task can be found in the [Enabling the use of global objects](#) topic.

### 6.11.2 Departments

Go to **Kentico CMS Desk -> Tools -> E-commerce -> Configuration -> Departments**. Department is used for organization of your product management efforts - you can specify users who can manage

products in the given department. It means you can have a different product manager for e.g. books and for electronics departments.

Please note that the **departments are not the same as product categories** displayed to site visitors.

You can create either only single department and place all products in it or you can create separate departments if you have several product managers.

Every department has these properties:

- **Department display name** - friendly name displayed to users
- **Department code name** - code name used by developers in the code
- **Default tax class for new products** - the tax class that is automatically assigned to the products created in the given department

In the **Users** tab, you can specify the users who can manage products in these departments.

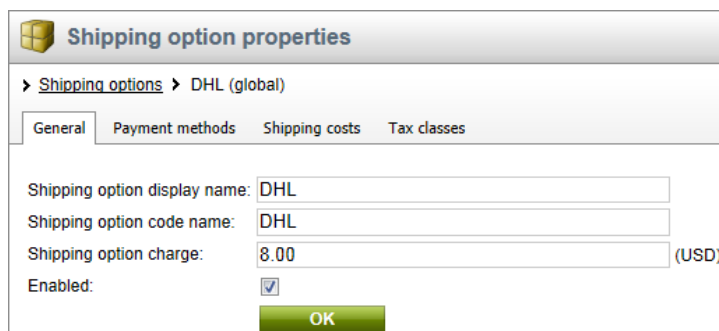
### 6.11.3 Shipping options

You can manage shipping options at **CMS Desk -> Tools -> E-commerce -> Configuration -> Shipping options**.

#### General tab

Each shipping option has the following properties on the **General** tab:


- **Shipping option display name** - friendly name displayed to customers and store managers
- **Shipping option name** - code name used by developers
- **Shipping option charge** - amount charged for this shipping option (in the main currency)
- **Enabled** - indicates if shipping option is offered to customers



The screenshot shows a web-based configuration window titled "Shipping option properties". It has a breadcrumb trail: "Shipping options > DHL (global)". Below the title bar are four tabs: "General", "Payment methods", "Shipping costs", and "Tax classes". The "General" tab is active. It contains four input fields: "Shipping option display name:" with the value "DHL", "Shipping option code name:" with the value "DHL", "Shipping option charge:" with the value "8.00" and a "(USD)" label to the right, and "Enabled:" with a checked checkbox. At the bottom right is a green "OK" button.

#### Payment methods tab

On the **Payment methods** tab, you can choose which payment methods will be available for the given shipping option.

 **Shipping option properties**

> [Shipping options](#) > DHL (global)

General | **Payment methods** | Shipping costs | Tax classes

The changes were saved.

The shipping option enables customers to choose one of the following payment methods:

<input type="checkbox"/>	Payment name
<input type="checkbox"/>	Cash on delivery
<input type="checkbox"/>	Credit card
<input type="checkbox"/>	Customer credit
<input type="checkbox"/>	PayPal


Remove selected
 Add payments

## Shipping costs tab

On the **Shipping costs** tab, you can define shipping costs for the given shipping option. There are two types of shipping: [flat shipping](#) and [shipping based on weight](#).


### Flat shipping

No shipping cost is defined. The same cost is charged for all shipping for the given shipping option, as defined in the **Shipping option charge** property on the **General** tab.

 **Shipping option properties**

> [Shipping options](#) > DHL (global)


General | **Payment methods** | **Shipping costs** | Tax classes

 [New shipping cost](#)

There are no shipping costs defined.


### Shipping based on weight





Shipping cost is defined and the cost is charged for the particular weight range. Please note that the cost of the lowest weight range is defined in the **Shipping option charge** property on the **General** tab.




 **Shipping option properties**

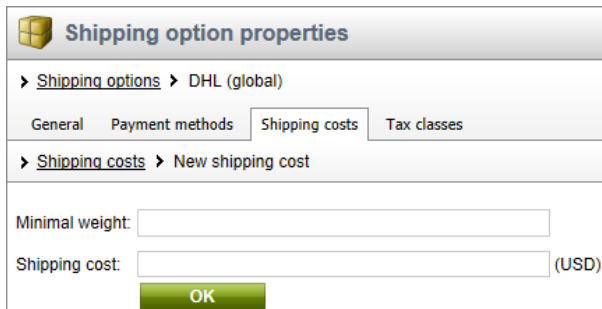
> [Shipping options](#) > DHL (global)

General | **Payment methods** | **Shipping costs** | Tax classes

 [New shipping cost](#)

Actions	Minimal weight	Shipping cost
	0	\$8.00
 	5	\$15.00
 	10	\$20.00

You can add a new shipping cost by clicking the  **New shipping cost** link and you can also **Edit** () and **Delete** () the existing shipping costs.



**Shipping option properties**

> [Shipping options](#) > DHL (global)

General   Payment methods   **Shipping costs**   Tax classes

> [Shipping costs](#) > New shipping cost

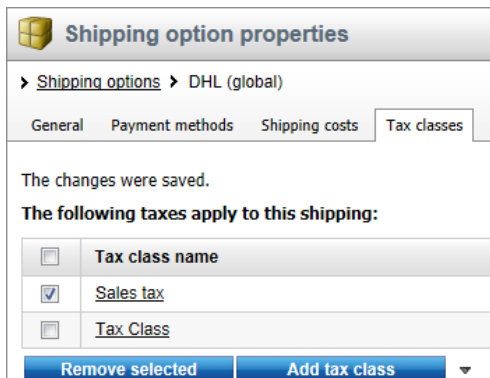
Minimal weight:

Shipping cost:  (USD)

**OK**

## Tax classes tab

On the **Tax classes** tab, you can choose which tax classes will be applied to the given shipping option. Tax classes can be edited at **CMS Desk -> Tools -> E-commerce -> Configuration -> Tax classes**.



**Shipping option properties**

> [Shipping options](#) > DHL (global)

General   Payment methods   Shipping costs   **Tax classes**

The changes were saved.

**The following taxes apply to this shipping:**

<input type="checkbox"/>	Tax class name
<input checked="" type="checkbox"/>	<a href="#">Sales tax</a>
<input type="checkbox"/>	<a href="#">Tax Class</a>

**Remove selected**   **Add tax class** ▼

### 6.11.4 Payment methods

You can manage payment methods at **Kentico CMS Desk -> Tools -> E-commerce -> Configuration -> Payment methods**. Every payment method has the following properties:

- **Payment method display name** – Friendly name (localizable string or simple text) displayed to the users.
- **Payment method code name** – Unique identifier which must be unique within all of the site payment methods.
- **Enabled** – Indicates if payment method could be used for payment.
- **Allow if no shipping is supplied** - Indicates if the payment method is allowed if no shipping is supplied. If checked for a given payment method, the method is offered in case shipping is not available during the purchase process (e.g. if no product with the Needs shipping flag is found in the shopping cart).
- **Payment gateway URL** - External payment gateway url or relative path to your document controlled by the Kentico CMS which represents your custom payment processor. It can be parameterized with data macros to evaluate the data fields of the order and related objects.

The following example shows how you can insert finished order ID and customer billing address state code to the payment url, however, you can get any value from the related objects. See **Kentico CMS Database Reference** for detailed column listing of the objects:



```
http://www.SomePaymentGateway.com?orderid={%Order.OrderId%}&state={%BillingAddress.State.StateCode%}
```

{%Order.OrderID%}	Displays the value of specified order data column ( <b>COM_Order</b> )
{%ShoppingCart.ShoppingCartID%}	Displays the value of specified shopping cart data column ( <b>COM_ShoppingCart</b> )
{%OrderStatus.StatusID%}	Displays the value of specified order status data column ( <b>COM_OrderStatus</b> )
{%BillingAddress.AddressID%}	Displays the value of specified billing address data column ( <b>COM_Address</b> )
{%BillingAddress.Country.CountryID%}	Displays the value of specified billing address country data column ( <b>CMS_Country</b> )
{%BillingAddress.State.StateID%}	Displays the value of specified billing address state data column ( <b>CMS_State</b> )
{%ShippingAddress.AddressID%}	Displays the value of specified shipping address data column ( <b>COM_Address</b> )
{%ShippingAddress.Country.CountryID%}	Displays the value of specified shipping address country data column ( <b>CMS_Country</b> )
{%ShippingAddress.State.StateID%}	Displays the value of specified shipping address state data column ( <b>CMS_State</b> )
{%CompanyAddress.AddressID%}	Displays the value of specified company address data column ( <b>COM_Address</b> )
{%CompanyAddress.Country.CountryID%}	Displays the value of specified company address country data column ( <b>CMS_Country</b> )
{%CompanyAddress.State.StateID%}	Displays the value of specified company address state data column ( <b>CMS_State</b> )
{%ShippingOption.ShippingOptionID%}	Displays the value of specified shipping option data column ( <b>COM_ShippingOption</b> )
{%PaymentOption.PaymentOptionID%}	Displays the value of specified payment option data column ( <b>COM_PaymentOption</b> )
{%Currency.CurrencyID%}	Displays the value of specified currency data column ( <b>COM_Currency</b> )
{%Customer.CustomerID%}	Displays the value of specified customer data column ( <b>COM_Customer</b> )
{%DiscountCoupon.DiscountCouponID%}	Displays the value of specified discount coupon data column ( <b>COM_DiscountCoupon</b> )

- **Payment gateway assembly name** – Name of a library which includes your payment gateway class inside.
- **Payment gateway class name** – Name of a class which represents your custom payment gateway processor.
- **Order status when payment succeeds** – Order status which is set to the existing order when its

payment succeeds.

- **Order status when payment fails** – Order status which is set to the existing order when its payment fails.

**Please note:** the offered payment methods depend on the chosen [shipping option](#). You need to specify this relationship when editing required shipping option. If there is no enabled shipping option all payment methods are offered.

### 6.11.5 Tax classes

You can manage tax classes at **Kentico CMS Desk -> Tools -> E-commerce -> Configuration -> Tax classes**.

**Tax classes** represent a customizable system that allows you to configure taxes paid for goods. You can specify different tax classes with different rates for each country. Then you apply these taxes to particular products.

Every tax class has the following properties:

- **Tax class display name** - friendly name displayed to users
- **Tax class code name** - code name used by developers in the code
- **Zero tax if Tax ID is supplied** - indicates if the tax should be calculated as zero (0) if the customer enters the Tax ID. The Tax ID is only displayed during the purchase process if the **Configuration -> Store Settings -> Show Tax registration ID field** box is checked.

On the **Countries** and **States** tab, you can set the value for particular countries (the country is recognized based on the billing address). The tax value can be either percentage (by default) or it can be a flat fee. If both country and state tax are specified, only the state tax is used.

### 6.11.6 Currencies

The store may offer goods for prices in several currencies. The available currencies can be managed at **Kentico CMS Desk -> Tools -> E-commerce -> Configuration -> Currencies**.

#### How to configure currencies

1. Enter all currencies you will use and delete or disable (uncheck the box **Enabled**) the currencies that you do not want to use. You will need to enter the following properties:

- **Currency display name** - the friendly name displayed to site visitors
- **Currency code name** - the code name used by developers
- **Currency code** - the official code of the currency used in exchange rates
- **Currency formatting string** - the format used to display amounts in the given currency - use {0} expression to insert the value into the formatting text
- **Significant digits** - the number of digits in the price that will be used in the total amount. The value will be rounded if the actual number of decimal digits is higher
- **Enabled** - indicates if the currency should be displayed in the currency drop-down list

2. If applicable, set the new currency as main currency at **Configuration -> General**.

3. If you're using multiple currencies, please specify exchange rates at **Configuration -> Exchange rates**.

### 6.11.7 Exchange rates

You can configure the exchange rates between the main currency and the alternative currencies at **Kentico CMS Desk -> Tools -> E-commerce -> Configuration -> Exchange rates**. The exchange rates are organized into **Exchange tables** that specify the complete exchange rate table for a given time period. You can define a new exchange table e.g. for every day.

Every exchange table has the following properties:

- **Table display name** - friendly name of the exchange table - e.g. "January 15, 2007 - January 20, 2007"
- **Valid from, Valid to** - time period the exchange table applies to
- **Exchange rates** - here you can enter "how much you have to pay for a single unit of the given currency"

### 6.11.8 Order status

You can configure various order states at **Kentico CMS Desk -> Tools -> E-commerce -> Configuration -> Order status**. This can be e.g. New -> In progress -> Closed or Canceled. You can also specify their flow using the up/down arrows.

Every status has the following properties:

- **Order status display name** - friendly name displayed to store managers.
- **Order status code name** - code name used by developers.
- **Order status color** - the color used to highlight the orders in given status in the Orders dialog.
- **Send notification** - indicates whether e-mail notifications are sent to the customer and administrator when the status of the existing order is set to the given status. The E-commerce order status notification to customer e-mail template is used when sending notifications to customers. The E-commerce order status notification to administrator e-mail template is used when sending notifications to administrators. This setting is independent of the Send order notification setting in CMS Desk -> Tools -> E-commerce -> Configuration -> Store settings -> Emails. Furthermore, the Send e-commerce e-mails from and Send e-commerce e-mails to text boxes in CMSDesk -> Tools -> E-commerce -> Configuration -> Store settings -> Emails need to have their values specified so that you can receive and send notifications.
- **Enabled** - indicates whether this status should be used for new orders.
- **Mark order as paid** - indicates if the system should automatically mark the order as paid when it receives the given status.



#### Please note

Editing orders marked as paid is limited. Specifically, you cannot change any value which could influence your order price calculation.  
However, orders not marked as paid can be edited without limitation.

### 6.11.9 Public status

You can configure various states of the product displayed on the public site at **Kentico CMS Desk -> Tools -> E-commerce -> Configuration -> Public status**. This can be e.g. "Available", "Accepting pre-orders" or "Not available".

Every status has the following states:

- **Public status display name** - friendly name displayed to store managers
- **Public status code name** - code name used by developers
- **Enabled** - indicates if this status should be used for new orders

This can be e.g. "Available", "Accepting pre-orders" or "Not available".

### 6.11.10 Internal status

You can configure various states of the product used by store managers to differentiate products at **Kentico CMS Desk -> Tools -> E-commerce -> Configuration -> Internal status**. This can be e.g. "Featured product", "New!" or "Sale". You can then use this information on your website to display the products e.g. in the "New products" box on your home page.

Every status has the following states:

- **Internal status display name** - friendly name displayed to store managers
- **Internal status code name** - code name used by developers
- **Enabled** - indicates if this status should be used for new orders

This can be e.g. "Featured product", "New!" or "Sale". You can then use this information on your website to display the products e.g. in the **Featured products** list on your home page.

### 6.11.11 Invoice

You can customize the invoice (or receipt) design at **Kentico CMS Desk -> Tools -> E-commerce -> Configuration -> Invoice** dialog. You can edit the HTML code representing your invoice here and use special expressions to insert dynamic parts of the invoice.

For example, you can use the **##BILLINGADDRESS##** expression to insert the customer's billing address into the invoice.

You can find a complete list of available dynamic expressions at the bottom of the dialog.

**Please note:** In Kentico CMS offers you the capability of splitting an invoice into several pages for printing. Should you experience difficulties with the printed design of an invoice in Internet Explorer, try to print it using the Firefox web browser instead.

Invoice		Company logo		
Invoice number:	2	Order date:	4/6/2008	
Supplier	Customer			
Company address	Development Soft 1020 Blueberry Ln. Tucson 85754 USA, Arizona			
Payment option				
Cash on delivery				
Product name	Units	Price/unit	Tax	Subtotal
Configurable PC	1	0.00	0.00	0.00
- Windows Vista Home Premium	1	216.00	0.00	216.00
- CD RW Samsung Black SATA	1	42.00	0.00	42.00
- INTEL Core 2 Quad Q6600 2.40GHz	1	220.00	0.00	220.00
- nVIDIA GeForce 6600GTS 512MB	1	440.00	0.00	440.00
- SAMSUNG 500GB SATA	1	139.00	0.00	139.00
- DDR2 1024MB 667MHz	1	30.00	0.00	30.00
- ACER 22"	1	460.00	0.00	460.00
- AUDIGY SE bulk	1	30.00	0.00	30.00
- Repro 5.1	1	60.00	0.00	60.00
- Printer	1	90.00	0.00	90.00
- Scanner	1	69.00	0.00	69.00
Configurable PC	1	0.00	0.00	0.00
- Windows XP Home	1	191.00	0.00	191.00
- CD RW Samsung Black SATA	1	42.00	0.00	42.00
- INTEL Core 2 Quad Q6600 2.40GHz	1	220.00	0.00	220.00
- nVIDIA GeForce 6600GTS 512MB	1	440.00	0.00	440.00
- SAMSUNG 500GB SATA	1	139.00	0.00	139.00
- DDR2 1024MB 667MHz	1	30.00	0.00	30.00
- ACER 22"	1	460.00	0.00	460.00
- AUDIGY SE bulk	1	30.00	0.00	30.00
- Repro 5.1	1	60.00	0.00	60.00
- Printer	1	90.00	0.00	90.00
- Scanner	1	69.00	0.00	69.00
Configurable PC	2	0.00	0.00	0.00
- Windows Vista Business	2	270.00	0.00	540.00
- CD RW Samsung Black SATA	2	42.00	0.00	84.00
- INTEL Core 2 Quad Q6600 2.40GHz	2	220.00	0.00	440.00
- nVIDIA GeForce 6600GTS 512MB	2	440.00	0.00	880.00
- SAMSUNG 500GB SATA	2	139.00	0.00	278.00
- DDR2 1024MB 667MHz	2	30.00	0.00	60.00
- ACER 22"	2	460.00	0.00	920.00
- AUDIGY SE bulk	2	30.00	0.00	60.00
- Repro 5.1	2	60.00	0.00	120.00

Product name	Units	Price/unit	Tax	Subtotal
- Printer	2	55.00	0.00	110.00
- Scanner	2	55.00	0.00	110.00
Configurable PC	1	0.00	0.00	0.00
- Red Hat Linux	1	55.00	0.00	55.00
- CD RW Samsung Black SATA	1	42.00	0.00	42.00
- INTEL Core 2 Quad Q6600 2.40GHz	1	220.00	0.00	220.00
- nVIDIA GeForce 6600GTS 512MB	1	440.00	0.00	440.00
- SAMSUNG 500GB SATA	1	139.00	0.00	139.00
- DDR2 1024MB 667MHz	1	30.00	0.00	30.00
- ACER 22"	1	460.00	0.00	460.00
- AUDIGY SE bulk	1	30.00	0.00	30.00
- Repro 5.1	1	60.00	0.00	60.00
- Printer	1	55.00	0.00	55.00
- Scanner	1	55.00	0.00	55.00
Total shipping:				\$ 8.00
Total price:				\$ 9197.00
Tax summary:				

## 6.12 Enabling the e-commerce module

You may need to enable the e-commerce product before you start using it. Go to **Kentico CMS Site Manager -> Settings -> E-commerce**, select your website in the drop-down list and check the box "Product" tab enabled. Click **Save**.

It will ensure that documents can be marked as products in the **Kentico CMS Desk -> Content** section.

The screenshot displays the Kentico CMS Desk interface. The top navigation bar includes 'Live Site', 'Site Manager', and 'E-commerce site'. The main menu shows 'Content', 'My desk', 'Tools', 'Administration', 'E-commerce', and 'On-line marketing'. The 'Content' section is active, showing a toolbar with 'New', 'Delete', 'Copy', 'Move', 'Up', 'Down', 'Edit', 'Preview', 'Live site', 'List', and 'Search'. The left sidebar shows the 'E-commerce site' structure with folders like 'Home', 'News', 'Products', 'How to buy', 'Company', 'Silverlight', 'Special pages', and 'Files'. The 'Products' folder is selected. The main content area shows a form for creating a new product with fields for Name, Resolution(Mpx), Memory, Battery, Sensor type, Light Sensitivity, Camera flash, Lens, Digital zoom, White balance, Video support, Weight(g), Width(mm), Height(mm), and Depth(mm). There are also 'Publish from' and 'Publish to' buttons. A red circle highlights the 'Create a new product' button at the bottom.

**Kentico CMS Desk** Live Site Site Manager E-commerce site

Content My desk Tools Administration E-commerce On-line marketing

New Delete Copy Move Up Down Edit Preview Live site List Search

Content management View mode Other

E-commerce site

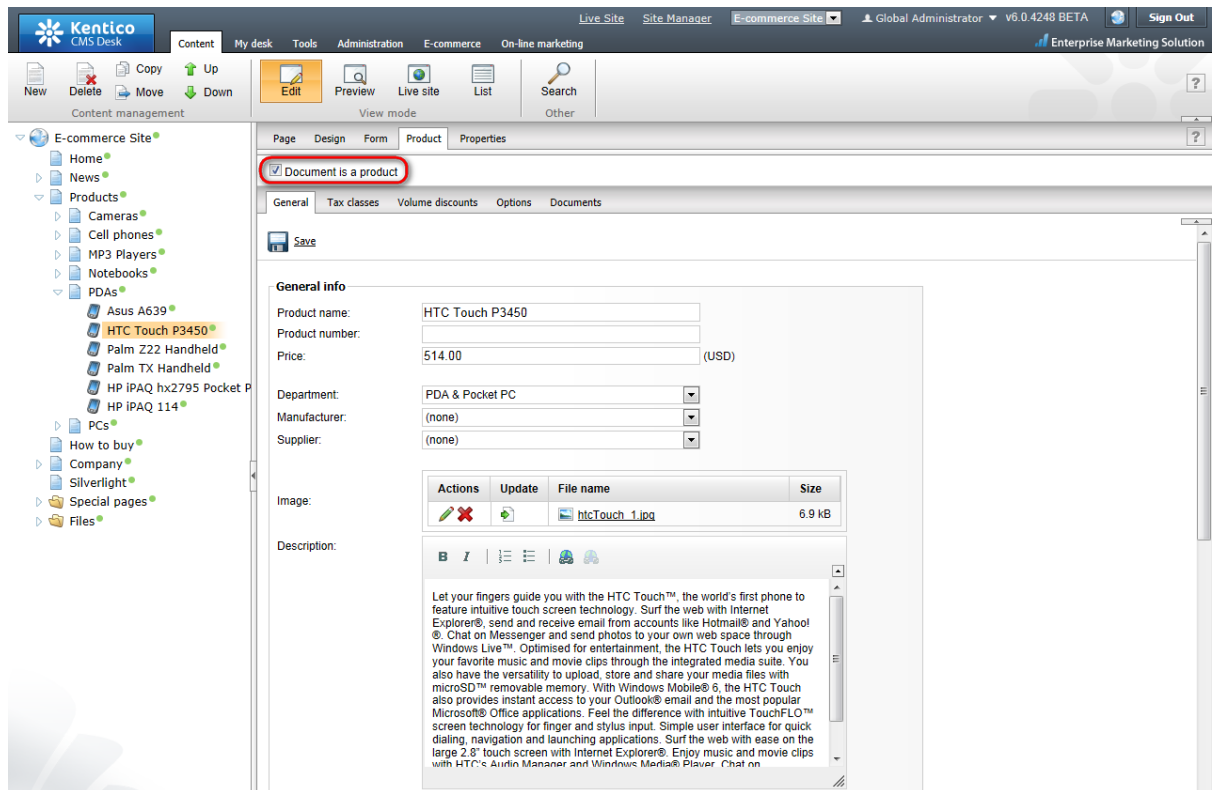
Home News Products How to buy Company Silverlight Special pages Files

Save Save and create another Spell check

Name: Resolution(Mpx): Memory: Battery: Sensor type: Light Sensitivity: Camera flash: Lens: Digital zoom: White balance: Video support: Weight(g): Width(mm): Height(mm): Depth(mm):

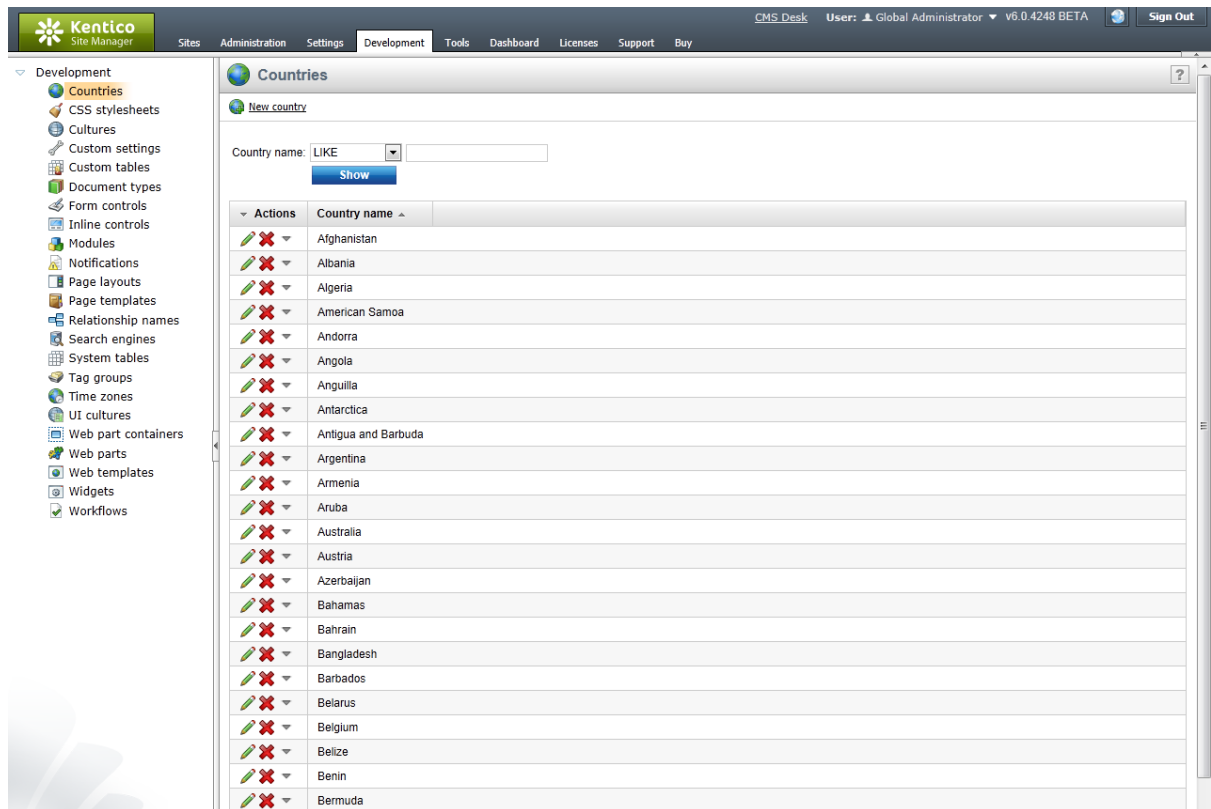
Publish from: Now Publish to: Now

☐ Create a new product



## 6.13 Countries and states

Various dialogs through-out the e-commerce module and the check-out process display a list of countries and states. You can manage the list of available countries and states at **Site Manager -> Development -> Countries**. These settings need to be pre-defined by some of the global administrators.



## 6.14 E-commerce and multi-site configuration

When you run multiple sites from a single Kentico CMS interface, it's important to consider that some e-commerce settings are shared by all websites. Thus, it's recommended that you use the multi-site support only for a single company that manages multiple e-commerce stores with shared settings.

The following settings are **shared** among websites:

- **orders**
- **customers**
- **products**
- **product options**
- **manufacturers**
- **suppliers**
- **discounts**
- **departments**
- **tax classes**
- **currencies**
- **exchange rates**
- **order status**
- **public status**
- **internal status**

The following settings are **site-specific**:

- **store settings**



- shipping options
- payment methods
- invoice template

## 6.15 Mapping document fields to product properties

You can map document fields into the product fields. Then, when the document is modified, the values are automatically updated in its product properties.

Please note that you cannot map product fields to document fields. The mapping doesn't work the other way round.

You can map the fields in the **Site Manager -> Development -> Document Types -> <select document type> -> E-commerce** dialog.

You can also choose to automatically mark the document as a product when a new document of the given type is created. In this case, you need to check the box **Automatically create product when a new document is created** and choose the **default department** to which the product will be assigned. In this case, you need to map at least the **Product name** and **Product price** source fields in the drop-down lists above.

The screenshot shows the 'Document type properties' dialog in the Kentico Site Manager. The left sidebar lists various development tools, with 'Document types' selected. The main panel shows the 'E-commerce' tab for the 'Product - Camera' document type. It contains several dropdown menus for mapping document fields to product properties: 'Product name source' (CameraName), 'Product image source' (none), 'Package weight source' (none), 'Package height source' (none), 'Package width source' (none), 'Package depth source' (none), 'Product price source' (CameraWidth), and 'Product description source' (none). There is a checkbox for 'Automatically create product when a new document is created' which is checked. Below this is a 'Default department for new product' dropdown set to 'Cameras', with 'Select' and 'Clear' buttons. At the bottom is an 'OK' button.

## 6.16 Site Manager settings

### E-commerce

You can set the following e-commerce module properties in **Site Manager -> Settings -> E-commerce** :

### Taxes

- **Default country** - sets the default country, choose your country or country where you sell the most. All taxes are applied based on their values in the default country unless the customer specifies their country or state during the checkout process.
- **Apply taxes based on** - indicates whether the taxes are applied based on shipping or billing address. Taxes of orders without a shipping address will be calculated based on a billing address, regardless of this setting.

## Accounts

- **Allow anonymous customers** - indicates if customers need to register on your site so that they can make the purchase.
- **Use an extra company address** - indicates if the option of providing company address is available in the check out process.
- **Require company account information** - indicates if it is compulsory to provide company account information in the check out process.
- **Show Tax registration ID field** - if checked, customer details will contain the tax registration ID field (e.g. VAT registration ID). The field will be displayed also in **My account** on the live site and during the check out process.
- **Show organization ID field** - if checked, customer details will contain the organization ID field. The field will be displayed also in **My account** on the live site and during the check out process.

## Shipping

- **Weight formatting string** - the format used to display product weight. Use {0} expression to insert the weight into the formatting string.
- **Minimal amount for free shipping** - the minimum order value when the shipping is not charged.

## Pages

- **Default product image URL** - default product image URL (virtual path). This image is used when product image is not specified.
- **My account URL** - My account page URL (virtual path).
- **Wishlist URL** - Wishlist page URL (virtual path).
- **Shopping cart URL** - Shopping cart page URL (virtual path).
- **Redirect to shopping cart** - if checked, the user is redirected to the shopping cart content page when the **Add to shopping cart** button is clicked. Otherwise, the user stays on the same page and the product is added to the shopping cart at the background.

## User interface

- **"Product" tab enabled** - determines whether the Product tab should be displayed while editing document in **CMSDesk -> Content**. It allows content editor to mark editing document as product. Please see the [Enabling the e-commerce module](#) chapter for more details.

## E-mails

- **Send e-commerce e-mails from** - an e-mail address from which the e-commerce notification e-mails are sent.
- **Send e-commerce e-mails to** - an e-mail address (e.g. merchant's e-mail address) to which the e-commerce notification e-mails are sent.
- **Send order notification** - indicates if e-mail notifications are sent after an order is finished and saved. The e-mail template **E-commerce order notification to customer** is used when sending a notification to a customer. The e-mail template **E-commerce order notification to administrator** is used when sending a notification to the administrator.
- **Send payment notification** - indicates if e-mail notifications are sent after the payment is completed. This is done automatically by the system on condition that the user paid using a payment gateway or if the order changed its status to one with the **Mark order as paid** property enabled

(available in **E-commerce -> Configuration -> Order status -> Edit (✎) order status**). Besides, a store administrator can manually complete the payment by enabling the **Order is paid** property (available in **CMS Desk -> E-commerce -> Orders -> Edit (✎) order** on the **Billing** tab). Please note that the e-mail template **E-commerce - Order payment notification** to customer is used when sending a notification to a customer whereas the e-mail template **E-commerce - Order payment notification** to administrator is used when sending a notification to an administrator.

- **Send e-products reminder (days)** - the number of days before e-product expiration when a notification is to be sent to the user.
- **Use customer's culture for e-mails** - indicates if e-mails to customer will be sent in the shopping cart culture.

## Conversion tracking

- **Registration conversion name** - the name of the conversion that will be logged when a user successfully registers on the website through the checkout process.
- **Registration conversion value** - sets a number that will be recorded along with the registration conversion when it is logged. The values are cumulative, i.e. when a conversion hit is logged, the specified value is added to the total sum previously recorded for this conversion. You may insert a macro expression into this field to dynamically retrieve a value from the current site context.
- **Order conversion name** - the name of the conversion that will be logged when a user completes an order.
- **Order conversion value** - sets a number that will be recorded along with the order conversion when it is logged. The values are cumulative, i.e. when a conversion hit is logged, the specified value is added to the total sum previously recorded for this conversion. You may insert a macro expression into this field to dynamically retrieve a value from the current site context.
- **Add to shopping cart conversion name** - the name of the conversion that will be logged when a user adds a product to the shopping cart.
- **Add to shopping cart conversion value** - sets a number that will be recorded along with the specified conversion when a product is added to the shopping cart. The values are cumulative, i.e. when a conversion hit is logged, the specified value is added to the total sum previously recorded for this conversion. You may insert a macro expression into this field to dynamically retrieve a value from the current site context.

## Payment Gateway - Authorize.NET

At **Site Manager -> Settings -> E-commerce -> Payment Gateway - Authorize.NET** you can set the following properties:

- **API Login**
- **Test mode**
- **Transaction key**

For more information please see the [Authorize.NET configuration](#) chapter.

## Payment Gateway - PayPal

At **Site Manager -> Settings -> E-commerce -> Payment Gateway - PayPal** you can set the following properties:

- **Business**
- **Cancel Return Url**
- **Notify Url**

- Return Url

For more information please see the [PayPal configuration](#) chapter.

## Global objects

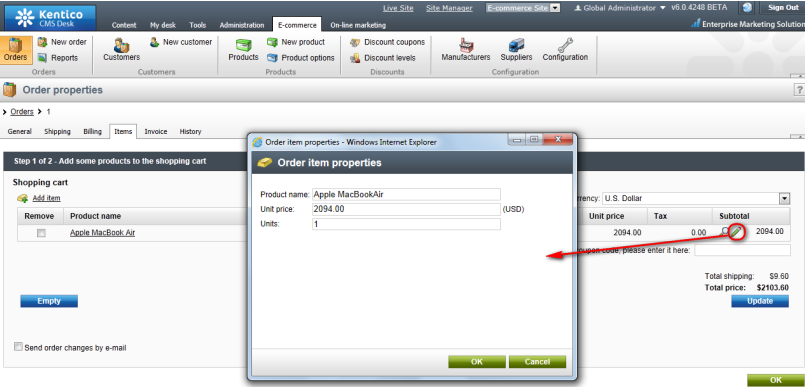
To learn how to enable and configure E-commerce module [global objects](#), please refer to the [Enabling the use of global objects](#) topic.

## 6.17 Web.config settings

In the **web.config** file, you can set appropriate keys in the following location to modify functionality of the e-commerce module:

```
<configuration>
  <appSettings>
    <add key="CMSEnableOrderItemEditing" value="true" />
    .
    .
    .
  </appSettings>
</configuration>
```

The following keys can be set:

<b>CMSUseCurrentSKUData</b>	<p>If set to <b>false</b> (default value), the name and price of the existing order item are used while editing order items.</p> <p>If set to <b>true</b>, the current product name and product price of a product are used while editing order items.</p>
<b>CMSEnableOrderItemEditing</b>	<p>If set to <b>true</b> and the order is not marked as paid, it is possible to <b>Edit</b> (✎) order item price or order item name. Please note that you can make these changes only if the <b>CMSUseCurrentSKUData</b> key is set to <b>false</b> at the same time.</p>  <p>It is set to <b>false</b> by default.</p>

<b>CMSUseMetaFileForProductImage</b>	<p>If set to <b>true</b> (default value), user is asked to choose a product image from the file system on their computer. The product image is uploaded to server and saved as a metafile of the given product (product image document of type <b>cms.file</b> is not created). The path to the given metafile is saved as a product image path (SKUImagePath).</p> <p>If set to <b>false</b>, user is asked to choose a document of type <b>cms.file</b> to become a product image. The path to the selected document is saved as a product image path (SKUImagePath).</p>
<b>CMSShoppingCartExpirationPeriod</b>	Specify the number of days after which a shopping cart is considered to be old and is erased with the <b>Deleting old shopping carts</b> task.

## Custom web.config settings

You can add your own settings (key and its value) into the **web.config** file. To work with these settings, use methods of the **CMS.SettingsProvider.SettingsKeyProvider** class.

The following example shows how to get the value (type of double) of the key named **MyCustomKey**.

[C#]

```
double value = CMS.SettingsProvider.SettingsKeyProvider.GetDoubleValue(
    "MyCustomKey" );
```

## 6.18 Localization settings

**KenticoCMS** supports localization of its features through localization expressions.

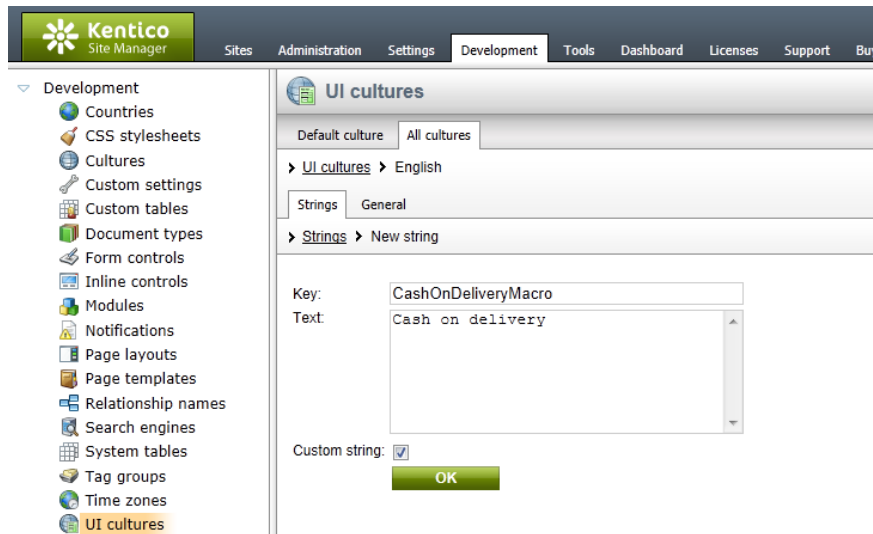
In the e-commerce module, the following features can be localized meaning that you can set their value either as plain text or as the macro expression.

- Product display name
- Product option display name
- Product option category display name
- Payment method display name
- Shipping option display name
- Currency display name
- Tax class display name
- Discount coupon display name
- Discount level display name
- Order status display name
- Public status display name
- Country display name
- State display name

Following example should suggest you how to supply a localized value into the document type transformation code or into your own aspx page.

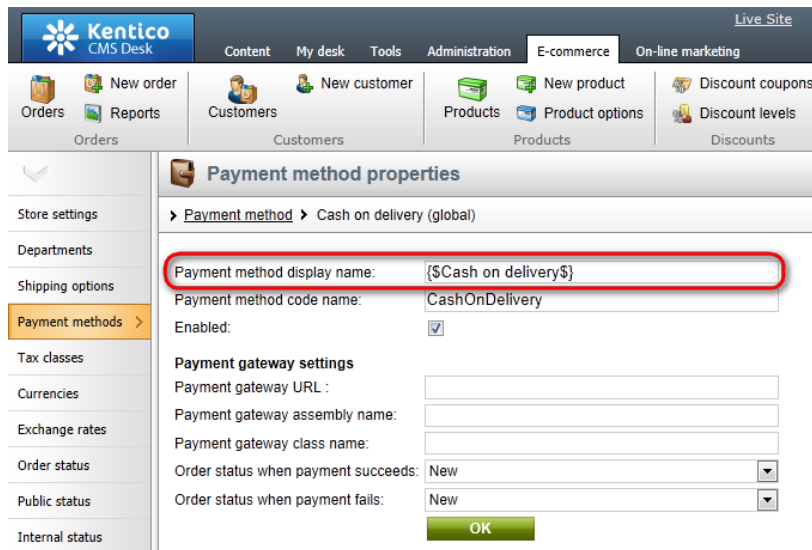
## 1. Creating localized expressions

At **CMS Site Manager** -> **Development** -> **UI cultures** you can create a new string for the given cultures.



## 2. Displaying localized expressions

For instance, if you have defined new string with the **Key** attribute *CashOnDeliveryMacro* and **Text** attribute *Cash on delivery* for the English culture and *Comptant à la livraison* for the french culture, using localization method **CMS.GlobalHelper.ResHelper.LocalizeString("{key}")** will result in displaying **Cash on delivery** to the English-speaking person and **Comptant à la livraison** to the French-speaking person.



For further information on the localization macros please refer to the **Localization Expressions** and **Development -> Macro expressions** chapters in **Kentico CMS Developer's Guide**.

## 6.19 Customizing invoice and e-mail templates

For your invoice and e-mail templates, you can use a whole range of pre-defined and data macros as you can find them at **CMSDesk -> Tools -> Ecommerce -> Configuration -> Invoice**.

### Pre-defined macros

Should you want to replace the current pre-defined macros, you need to enable the usage of the custom e-commerce provider and replace the appropriate method from CustomOrderInfoProvider.

Macro	Description	Method
##BILLINGADDRESS##	Displays customer address	<code>string GetAddress(int addressId)</code>
##COMPANYADDRESS##	Displays company address	<code>string GetAddress(int addressId)</code>
##SHIPPINGADDRESS##	Displays shipping address	<code>string GetAddress(int addressId)</code>
##PAYMENTOPTION##	Displays payment method	<code>string GetPaymentOption(int paymentId)</code>
##SHIPPINGOPTION##	Displays shipping option	<code>string GetShippingOption(int shippingId)</code>
##INVOICENUMBER##	Displays invoice number	<code>string GetInvoiceNumber(object order)</code>
##ORDERDATE##	Displays order date	<code>string GetOrderDate(object order)</code>
##ORDERNOTE##	Displays order note	<code>string GetOrderNote(object order)</code>
##PRODUCTLIST##	Displays list of ordered products	<code>string GetProductList(DataTable dt, object currency, bool renderDiscount)</code>
##TOTALSHIPPING##	Displays total shipping for order	<code>string GetTotalShipping(double value, object currency)</code>
##TOTALPRICE##	Displays total price for order	<code>string GetTotalPrice(double value, object currency)</code>
##TAXRECAPITULATION##	Displays list of all taxes used in order	<code>string GetTaxRecapitulation(DataTable dt, object currency)</code>
##TAXREGISTRATIONID##	Displays customer tax registration ID	<code>string GetTaxRegistrationID(object customerObj)</code>
##ORGANIZATIONID##	Displays customer organization ID	<code>string GetOrganizationID(object customerObj)</code>

### Data macros

You can use data macros to evaluate the data fields of the order and related objects. See 'Kentico CMS Database Reference' for detailed column listing of the objects.

Following example displays code of the customer billing address state, however, you can get any value from the related objects: `{%BillingAddress.State.StateCode%}`

Macro	Description
-------	-------------

{%Order.OrderID%}	Displays the value of specified order data column ( <b>COM_Order</b> )
{%ShoppingCart.ShoppingCartID%}	Displays the value of specified shopping cart data column ( <b>COM_ShoppingCart</b> )
{%OrderStatus.StatusID%}	Displays the value of specified order status data column ( <b>COM_OrderStatus</b> )
{%BillingAddress.AddressID%}	Displays the value of specified billing address data column ( <b>COM_Address</b> )
{%BillingAddress.Country.CountryID%}	Displays the value of specified billing address country data column ( <b>CMS_Country</b> )
{%BillingAddress.State.StateID%}	Displays the value of specified billing address state data column ( <b>CMS_State</b> )
{%ShippingAddress.AddressID%}	Displays the value of specified shipping address data column ( <b>COM_Address</b> )
{%ShippingAddress.Country.CountryID%}	Displays the value of specified shipping address country data column ( <b>CMS_Country</b> )
{%ShippingAddress.State.StateID%}	Displays the value of specified shipping address state data column ( <b>CMS_State</b> )
{%CompanyAddress.AddressID%}	Displays the value of specified company address data column ( <b>COM_Address</b> )
{%CompanyAddress.Country.CountryID%}	Displays the value of specified company address country data column ( <b>CMS_Country</b> )
{%CompanyAddress.State.StateID%}	Displays the value of specified company address state data column ( <b>CMS_State</b> )
{%ShippingOption.ShippingOptionID%}	Displays the value of specified shipping option data column ( <b>COM_ShippingOption</b> )
{%PaymentOption.PaymentOptionID%}	Displays the value of specified payment option data column ( <b>COM_PaymentOption</b> )
{%Currency.CurrencyID%}	Displays the value of specified currency data column ( <b>COM_Currency</b> )
{%Customer.CustomerID%}	Displays the value of specified customer data column ( <b>COM_Customer</b> )
{%DiscountCoupon.DiscountCouponID%}	Displays the value of specified discount coupon data column ( <b>COM_DiscountCoupon</b> )

## Custom macros

Following example shows how you can display customer total credit and current time in invoice or e-mails, however you can define your own custom macros.

Please refer to the **Development -> Macro expressions** chapter in **Kentico CMS Developer's Guide** for more details.

**[C#]**



```
using System;
using CMS.GlobalHelper;

public static string ResolveCustomMacro(MacroResolver sender, string expression,
out bool match)
{
    match = false;
    string result = expression;

    //
    // Custom macro examples
    //
    switch (expression.ToLower())
    {
        // Display customer total credit, macro in template: {#totalcredit#}
        case "totalcredit":

            match = true;

            // Get shopping cart object from resolver
            CMS.Ecommerce.ShoppingCartInfo cartObj = sender.SourceObject as
CMS.Ecommerce.ShoppingCartInfo;
            if ((cartObj != null) && (cartObj.ShoppingCartCustomerID > 0))
            {
                // Get customer total credit in default currency
                double totalCredit =
CMS.Ecommerce.CreditEventInfoProvider.GetCustomerTotalCredit(cartObj.
ShoppingCartCustomerID);

                CMS.Ecommerce.CurrencyInfo currentCurrency = null;
                double currentRate = 1.0;

                // Use shopping cart currency and its exchange rate
                if (cartObj.CurrencyInfoObj != null)
                {
                    currentCurrency = cartObj.CurrencyInfoObj;
                    currentRate = cartObj.ExchangeRate;

                    // Apply exchange rate
                    totalCredit =
CMS.Ecommerce.ExchangeTableInfoProvider.ApplyExchangeRate(totalCredit,
currentRate);

                }
                // Use default currency
                else
                {
                    currentCurrency = CMS.Ecommerce.CurrencyInfoProvider.
GetMainCurrency();
                }

                // Round value
                totalCredit =
CMS.Ecommerce.CurrencyInfoProvider.RoundTo(totalCredit, currentCurrency);

                // Return formatted value
                result =
```

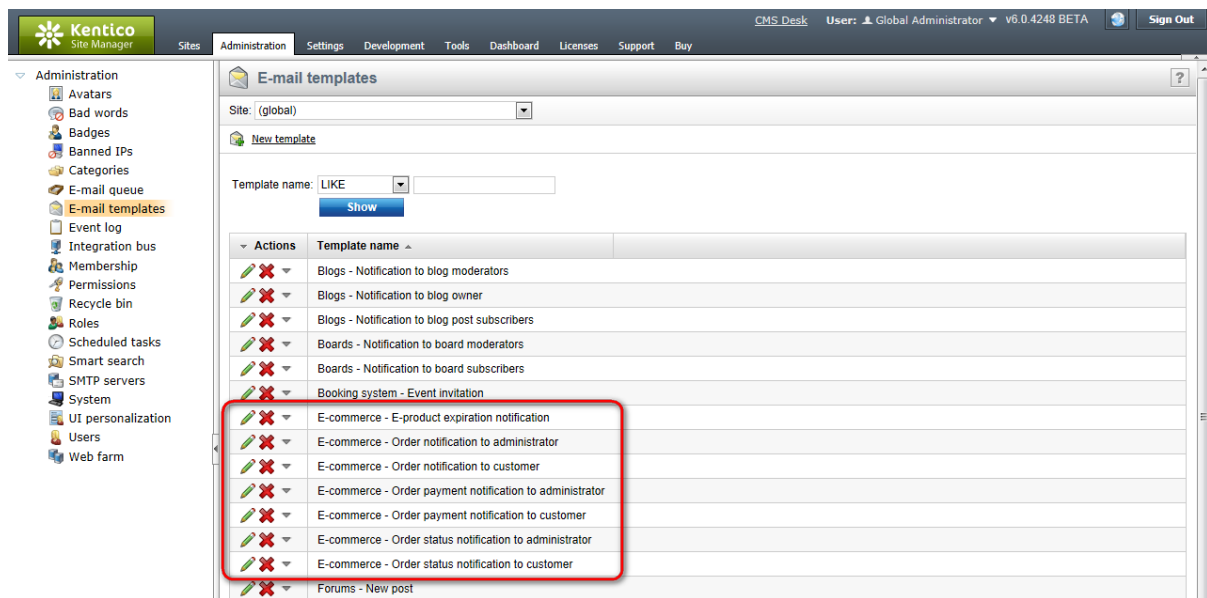
```
CMS.Ecommerce.CurrencyInfoProvider.GetFormattedPrice(totalCredit, currentCurrency);

        }
        break;

        // Display current date and time, macro in template: {#currenttime#}
        case "currenttime":
            result = DateTime.Now.ToString();
            break;
    }
    return result;
}
```

## E-mail templates

You can manage e-mail templates at **CMS Site Manager -> Administration -> E-mail templates**.



## Customizing e-mail subject

Beside e-mail templates, you can use the very same macros in subject of an e-mail. All you have to do is change the appropriate localization strings as specified in the table below. You can change the given localization strings at **<your web project folder>/CMSResources/CMS.<given UI culture code >.resx** (for default culture it's **CMS.resx**)

E-mail template	E-mail subject localization string
order notification to administrator	ordernotification.administratorssubject
order notification to customer	ordernotification.customerssubject
order payment notification to administrator	orderpaymentnotification.administratorssubject
order payment notification to customer	orderpaymentnotification.customerssubject
order status notification to administrator	orderstatusnotification.administratorssubject
order status notification to customer	orderstatusnotification.customerssubject

Following example shows how to customize subject of the order notifications which are sent to administrator:

1. Go to **<your web project folder>/CMSResources/** and open the **CMS.resx** file.
2. Find the **ordernotification.administratorssubject** key and change its value to **Order with ID {% Order.OrderID%} - {%OrderStatus.StatusDisplayName%}** so that the given part of code look in the following way.

```
<data name="ordernotification.administratorssubject" xml:space="preserve">
  <value>
```

```

    Order with ID {%Order.OrderID%} - {%OrderStatus.StatusDisplayName%}
</value>
<comment>IsNotCustom</comment>
</data>

```

This will include information about order ID and order status in the subject of an e-mail send to the administrator.

## Default settings of the e-mail subjects

When new order is made or items of the existing one are changed:

E-mail subject string	Default value
ordernotification.administratorssubject	Order: {%Order.OrderInvoiceNumber%}
ordernotification.customerssubject	Your order: {%Order.OrderInvoiceNumber%}

When payment is received:

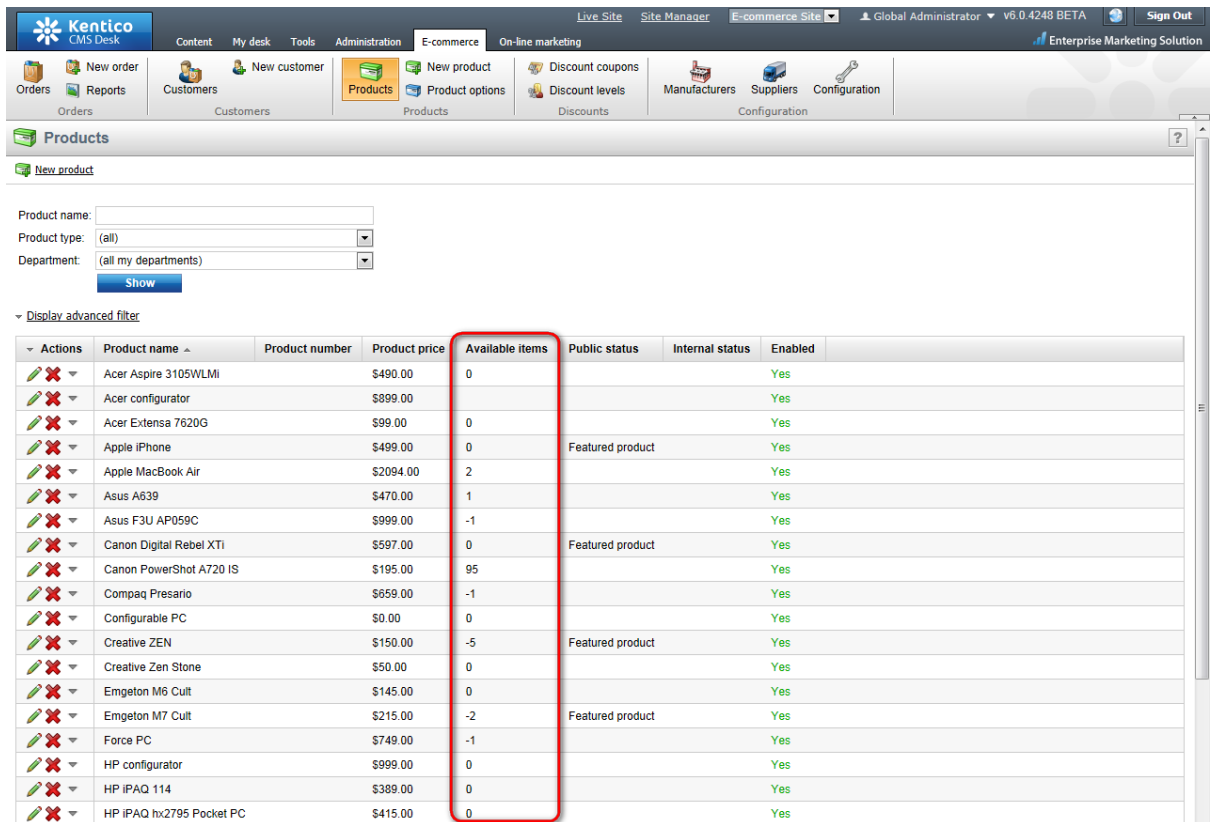
E-mail subject string	Default value
orderpaymentnotification.administratorssubject	Order {%Order.OrderInvoiceNumber%}: Payment received
orderpaymentnotification.customerssubject	Your order {%Order.OrderInvoiceNumber%}: Payment received

When status of the existing order is changed:

E-mail subject string	Default value
orderstatusnotification.administratorssubject	Order {%Order.OrderInvoiceNumber%}: {%OrderStatus.StatusDisplayName%}
orderstatusnotification.customerssubject	Your order {%Order.OrderInvoiceNumber%}: {%OrderStatus.StatusDisplayName%}

## 6.20 Using product inventory

Kentico CMS allows you to manage your product inventory at **CMS Desk -> Tools -> E-commerce -> Products**. Here, you can quickly check if a product is available and how many items do you have in stock.



Product name:

Product type: (all)

Department: (all my departments)

[Show](#)

▼ Display advanced filter

Actions	Product name	Product number	Product price	Available items	Public status	Internal status	Enabled
	Acer Aspire 3105WLMi		\$490.00	0			Yes
	Acer configurator		\$899.00				Yes
	Acer Extensa 7620G		\$99.00	0			Yes
	Apple iPhone		\$499.00	0	Featured product		Yes
	Apple MacBook Air		\$2094.00	2			Yes
	Asus A639		\$470.00	1			Yes
	Asus F3U AP059C		\$999.00	-1			Yes
	Canon Digital Rebel XTi		\$597.00	0	Featured product		Yes
	Canon PowerShot A720 IS		\$195.00	95			Yes
	Compaq Presario		\$659.00	-1			Yes
	Configurable PC		\$0.00	0			Yes
	Creative ZEN		\$150.00	-5	Featured product		Yes
	Creative Zen Stone		\$50.00	0			Yes
	Emgeton M6 Cult		\$145.00	0			Yes
	Emgeton M7 Cult		\$215.00	-2	Featured product		Yes
	Force PC		\$749.00	-1			Yes
	HP configurator		\$999.00	0			Yes
	HP IPAQ 114		\$389.00	0			Yes
	HP IPAQ hx2795 Pocket PC		\$415.00	0			Yes

There are two ways you can handle products in your inventory:

- 1.) Sell a product only if they are some available items in stock at **CMS Desk -> Tools -> E-commerce -> Products**.
- 2.) Always sell a product no matter if there are any available items in stock.

You can distinguish between these two by checking the **Sell only if items are available** checkbox either at **CMS Desk -> Tools -> E-commerce -> Products -> <edit given product> -> General** or on the **Product** tab of the product chosen in the content tree.

The screenshot shows the Kentico CMS 6.0 E-commerce interface. The top navigation bar includes links for Content, My desk, Tools, Administration, E-commerce, and On-line marketing. The main menu on the left lists various e-commerce functions like Orders, Reports, Customers, New customer, Products, New product, Product options, Discount coupons, Discount levels, Manufacturers, Suppliers, and Configuration. The central area displays the 'Product properties' form for 'Apple MacBook Air'. The 'General' tab is selected, showing fields for Product type (Product), Status (Enabled, Public status: (none), Internal status: (none)), and Inventory (Sell only if items available: ☐, Available items: 2, Availability (days): 0, Max items in one order: ).

## 1. Sell product only if there are available items in stock

For enabling this option, you have to enter the number of available items into the **Available items** text box and check the **Sell only if items are available** checkbox. An order is accepted only if the number of ordered items is no greater than the number of available items.







This close-up shows the 'Inventory' section of the product properties form. The 'Sell only if items available' checkbox is checked. The 'Available items' field contains the value '10'. The 'Availability (days)' field contains '0', and the 'Max items in one order' field is empty.

The sufficiency of the available items is checked every time a user does one of the following:

- Add product to the **shopping cart**
- Click the **Update** button in the shopping cart
- Click the **Checkout** button in the shopping cart
- Click the **Order now** button in the Order Preview



If you want to buy a product with product options, the chosen product and product options have to be enabled and there has to be sufficient number of available items of the given product with the given product option. Otherwise the error message appears.

Step 1 of 6 - Add some products to the shopping cart

 →  →  →  →  → 

### Shopping cart

Currency: U.S. Dollar

Remove	Product name	Units	Unit price	Tax	Subtotal
	<a href="#">Asus A639</a> <small>The product is currently not available in required quantity. The quantity was set to maximum available quantity, which is 1.</small>	<input type="text" value="1"/>	470.00	0.00	 470.00

If you have a coupon code, please enter it here:

Total shipping: \$0.00  
Total price: **\$470.00**

Please note that you can check the **Sell only if items are available** checkbox for product options as well. You can set check the **Sell only if items are available** checkbox and enter the number of available items for product options at **CMS Desk -> Tools -> E-commerce -> Products Options -> <edit option category> -> Options -> <edit product options>**.

Kentico CMS Desk

Content My desk Tools Administration **E-commerce** On-line marketing

Live Site Site Manager E-commerce Site Global Administrator v6.0.4248 BETA Sign Out

Enterprise Marketing Solution

Orders New order Reports Customers New customer Products New product **Product options** Discount coupons Discount levels Manufacturers Suppliers Configuration

### Category properties

> Product option categories > Cell phones accessories

General Options

> Product options > Car charger

General Tax classes

Product type: Product

**Status**

Enabled: ☒

Public status: (none)

Internal status: (none)

**Inventory**

Sell only if items available: ☒

Available items:

Availability (days):

Max items in one order:

**Shipping**

Needs shipping: ☒

If the checkbox is checked, there has to be the sufficient quantity of the given product option in stock so that an order is accepted. Otherwise the error message appears.

Step 1 of 6 - Add some products to the shopping cart

**Shopping cart**

Currency: U.S. Dollar

Remove	Product name	Units	Unit price	Tax	Subtotal
<input type="checkbox"/>	<a href="#">Motorola V3 Razr</a>	<input type="text" value="1"/>	130.00	0.00	<input type="text" value="130.00"/>
	- Car charger	<input type="text" value="1"/>	5.00	0.00	<input type="text" value="5.00"/>
	The product is currently not available in required quantity. The maximum available quantity is 0.				

If you have a coupon code, please enter it here:

Total shipping: \$0.00  
Total price: \$135.00

Please note that the very same check is made anytime a user edit the already existing order. When an item is removed from the order, it is added back to stocked items.

## 2. Always sell product

When the **Sell only if items available** checkbox is unchecked, an order is processed disregarding the number of available items of the given product or product option. Therefore, a customer can order more items that it is readily available in stock. Should that happen, the number of available items will be negative. Therefore, you'll know exactly how many items do you need to order to meet demands.

**Please note:** You can check the number of available items in stock at **CMSDesk -> Tools -> Ecommerce -> Reports -> Inventory** as well.



SKU name	Product number	Department name	Available items
512MB DDR2		Hardware	-19
Abit AW9D - Intel 975X		Hardware	2
ACER 22"		Hardware	-1
Acer Aspire 3105WLMi		Hardware	0
Acer configurator		Hardware	-
Acer Extensa 7620G		Hardware	0
ActionTech Gladiator CK-18B blue		Hardware	4
AMD ATHLON 64 LE-1620		Hardware	0
AMD ATHLON 64 X2 4200+		Hardware	0
AMD ATHLON 64 X2 4800+		Hardware	0
AMD ATHLON 64 X2 5000+		Cameras	0
AMD ATHLON 64 X2 5600+		Hardware	0
AMD ATHLON 64 X2 6400+		Cameras	0
AMD Sempron 1800Mhz		Hardware	0
AMD Sempron 64 LE-1200		Hardware	-1
AMD X2 1800 Mhz		Hardware	-1
AMD X2 2000 Mhz		Hardware	0
AOC 19"		Hardware	0
AOC 20"		Hardware	-1
Apple iPhone		Phones	0
Apple MacBook Air		Hardware	2

## 6.21 Discounts overview

In the Kentico CMS E-commerce module, you can define three different types of discounts for your customers:

- **Discount coupons** - discount from the price of the specified products. For more details see the [Discount coupons](#) chapter.
- **Discount levels** - discount that is assigned to the customer for all their purchases. For more details see the [Discount levels](#) chapter.
- **Volume discount** - discount assigned to the product that applies when the customer purchases specified amount of the given product. For more details see the [Products](#) chapter.

In the following examples, we purchase five units of **Apple MacBook Air** with a total price of \$10470 without any discount. Each example demonstrates where you can specify the given type of discount and how it affects the total price of the purchase.

Step 1 of 6 - Add some products to the shopping cart

### Shopping cart

Currency: U.S. Dollar

Remove	Product name	Units	Unit price	Tax	Subtotal
<input type="checkbox"/>	<a href="#">Apple MacBook Air</a>	5	2094.00	0.00	10470.00

If you have a coupon code, please enter it here:

Total shipping: \$0.00  
Total price: **\$10470.00**

## Discount coupons

You can create a new discount coupon or edit existing ones at **CMS Desk -> Tools -> E-commerce -> Discount coupons**.

Kentico CMS Desk

Content My desk Tools Administration **E-commerce** On-line marketing

Live Site Site Manager E-commerce Site Global Administrator v6.0.4248 BETA Sign Out

Enterprise Marketing Solution

Orders New order Reports Customers New customer Products New product Product options Discount coupons Discount levels Manufacturers Suppliers Configuration

**New discount coupon**

> Discount coupons > New discount coupon

Discount coupon name:

Discount coupon code:

☒ Absolute discount ☐ Relative discount


Discount value:  (USD)

Valid from:

Valid to:



While making purchase, a customer is required to enter a coupon code into the given text box to make use of it. The given discount is subtracted from the product price.

Step 1 of 6 - Add some products to the shopping cart



**Shopping cart**

Currency: U.S. Dollar

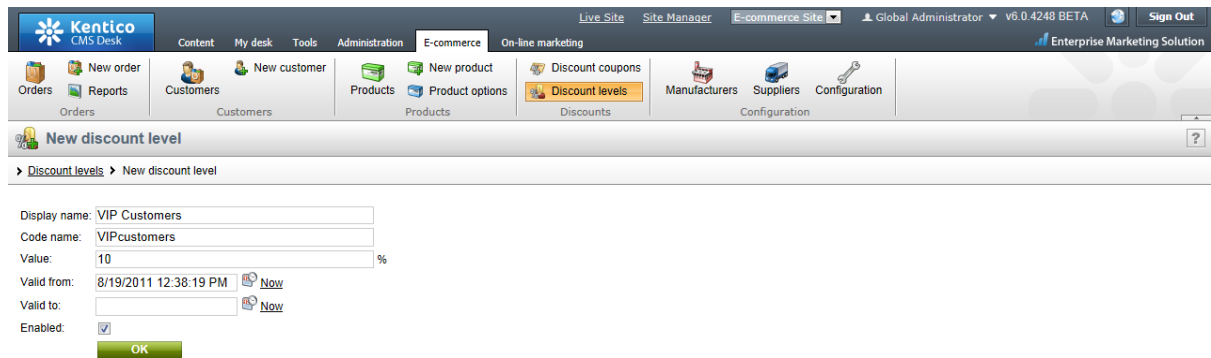
Remove	Product name	Units	Unit price	Unit discount	Tax	Subtotal
	<a href="#">Apple MacBook Air</a>	<input type="text" value="5"/>	2094.00	<b>100.00</b>	0.00	 9970.00

If you have a coupon code, please enter it here:

Total shipping: \$0.00  
Total price: **\$9970.00**

## Discount levels

You can set up a new discount level or edit existing ones at **CMS Desk -> Tools -> E-commerce -> Discount levels**.



**Kentico CMS Desk** | Live Site | Site Manager | E-commerce Site | Global Administrator | v6.0.4248 BETA | Sign Out

Content | My desk | Tools | Administration | **E-commerce** | On-line marketing

Orders | Reports | Customers | New customer | Products | New product | Discount coupons | Manufacturers | Suppliers | Configuration

**New discount level**

Discount levels > New discount level

Display name:   
Code name:   
Value:  %  
Valid from:    
Valid to:    
Enabled: ☒

After creating a new discount level, you have to assign it to those users who you want to take advantage of it.

**Kentico CMS Desk** | Content | My desk | Tools | Administration | **E-commerce** | On-line marketing | Live Site | Site Manager | E-commerce Site | Global Administrator | v6.0.4248 BETA | Sign Out

Orders | New order | Reports | Customers | New customer | Products | New product | Product options | Discount coupons | Discount levels | Manufacturers | Suppliers | Configuration

**Customer properties** | Customers | Simons David

General | Addresses | Orders | Credit | Newsletters

The changes were saved.

**General info**

First name: David  
 Last name: Simons  
 Company:  
 Organization ID:  
 Tax registration ID:  
 Country/state: USA  
 (none)

**Contacts**

E-mail: DavidSimons@example.com  
 Phone:  
 Fax:

**Registered user info**

**Login**

User name: DavidS [Edit](#)  
 Enabled: ☒

**Preferred settings on this site**

Preferred currency: U.S. Dollar  
 Preferred payment method: (none)  
 Preferred shipping option: (none)







**Discounts**

Discount level on this site: **VIP Customers**

[OK](#)


Before making a purchase, a user has to sign in to their account so that the assigned discount level is automatically take into account while calculating the unit and total price.

**Step 1 of 6 - Add some products to the shopping cart**

 →  →  →  →  → 

**Shopping cart**

Currency: U.S. Dollar

Remove	Product name	Units	Unit price	Unit discount	Tax	Subtotal
	<a href="#">Apple MacBook Air</a>	5	2094.00	<b>209.40</b>	0.00	9423.00

If you have a coupon code, please enter it here:

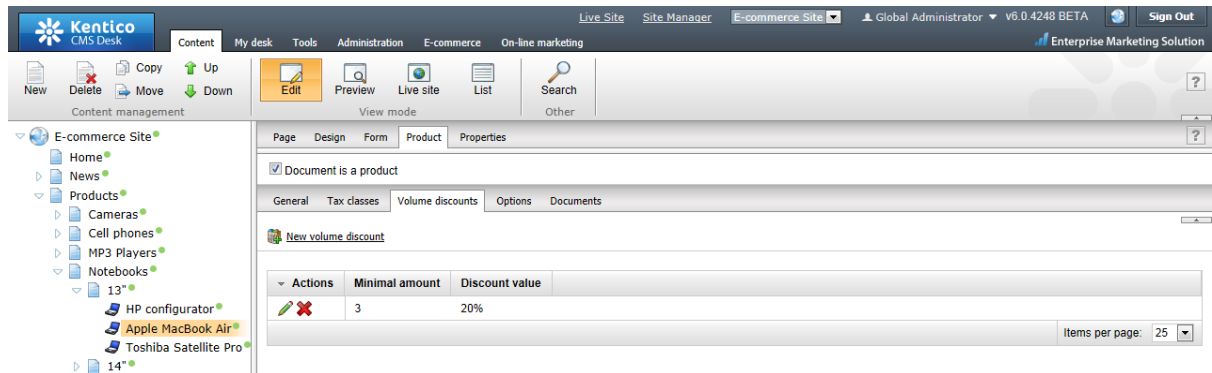
Total shipping: \$0.00  
**Total price: \$9423.00**

[Empty](#) [Update](#)

[Continue shopping](#) [Check out](#)

## Volume discounts

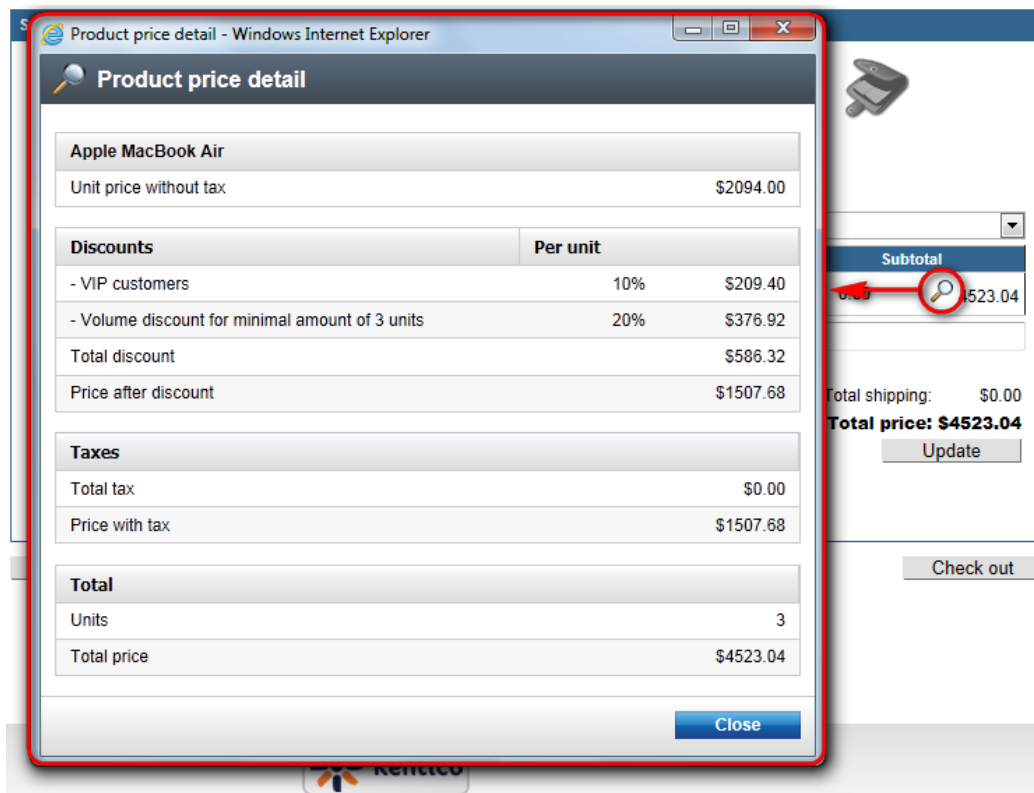
You can define volume discounts at **CMS Desk -> Content -> <choose product> -> Product -> Volume discounts**.



On condition a user adds to the **shopping cart** sufficient amount of units of the product with the assigned volume discount, the given discount level is automatically taken into account and its subtracted from the unit price.



You can combine all the discount types. While combining the discount coupon and the discount level, though, only higher discount applies.



For more details about how are discounts calculated see [Figure B: Discount calculation](#).

## 6.22 Deleting old shopping carts

The deletion of old shopping carts can be done automatically with the pre-defined task.

By default, all the shopping carts older than 30 days are considered old and therefore are erased when running the **Deleting old shopping carts** task. You can change the period of time after which the shopping cart expires by modifying the **CMSShoppingCartExpirationPeriod** key in the **web.config** file. See the [Web.config settings](#) chapter for more details.

This task erases not only old shopping carts, but anonymous customers with no order as well. This anonymous customer with no order appears when an anonymous user adds an item into the shopping cart. They are then required to enter their name and e-mail and they become a customer. If this anonymous user, for whatever reason, doesn't get through all steps of the checkout process, they remain in the system as anonymous customers with no order.

You can schedule the deletion of old shopping carts task by setting the **Scheduler** at **CMS Site Manager -> Administration -> Scheduled tasks**.

The screenshot shows the Kentico Site Manager Administration interface. The left sidebar lists various administration tasks, with 'Scheduled tasks' highlighted. The main area displays the 'Task properties' window for a task named 'Delete old shopping carts'. The task is configured with the following details:

- Task display name:** Delete old shopping carts
- Task name:** Ecommerce.DeleteOldShoppingCarts
- Task assembly name:** CMS.Ecommerce
- Task class name:** CMS.Ecommerce.ShoppingCartCleaner
- Period:** Day
- Start time:** 8/19/2011 12:57:57 PM (with a 'Now' button)
- Every:** 1 Day
- Task interval:** Days (Monday, Tuesday, Wednesday, Thursday, Friday, Saturday, Sunday are all checked)
- Task data:** (Empty text area)
- Task enabled:** ☒
- Delete task after last run:** ☐
- Run task in separate thread:** ☐
- Use external service:** ☐
- Server name:** (Empty text field)

An 'OK' button is located at the bottom of the window.

For more information please refer to the **Scheduler** chapter in **Kentico CMS Developer's guide**.

# Part

---



VII

**Site separation**

---



## 7 Site separation

### 7.1 Overview

Kentico CMS allows you to use in the E-commerce module both site-specific objects available for the respective sites only and global objects shared across all your sites. This approach not only gives you more flexibility but is also highly effective in multi-site environment, e.g. when hosting several e-commerce sites for different customers.

To learn how to allow global data on the given site, please refer to the [Enabling the use of global objects](#) topic.

Security matters related to your E-commerce package site separation are described in the [Security](#) topic.

In the E-commerce module, there are three groups of objects available for a particular site from the point of view of site separation:

- [Objects with both site and global option](#) include objects which can be used either as site-specific or site-specific together with global.
- [Objects with only site or global option](#) include objects which can be used only separately, either as site-specific or global.
- [Objects with special cases](#) include objects which cannot be put into either of the above groups.

Before you start configuring and using global data in the E-commerce module, you should learn what basic E-commerce module configuration [scenarios](#) are available with regard to site separation.

### 7.2 Basic configuration scenarios

This topic describes basic E-commerce module configuration scenarios with regard to site separation. There are three such scenarios:

#### Single site (on-line store)

Typically, if you have only one on-line store, you should always use site data only. It helps you build other on-line stores in the future without any problem.

#### Multiple sites (on-line stores)

This scenario concerns multiple sites (on-line stores) which are owned and managed by a single organization. Typically, there is a need for global data which should be shared across all sites. The set of settings for [objects with both site and global option](#) and [objects with only site or global option](#) covers all typical scenarios for usage of global data.


#### Multiple sites (on-line stores)

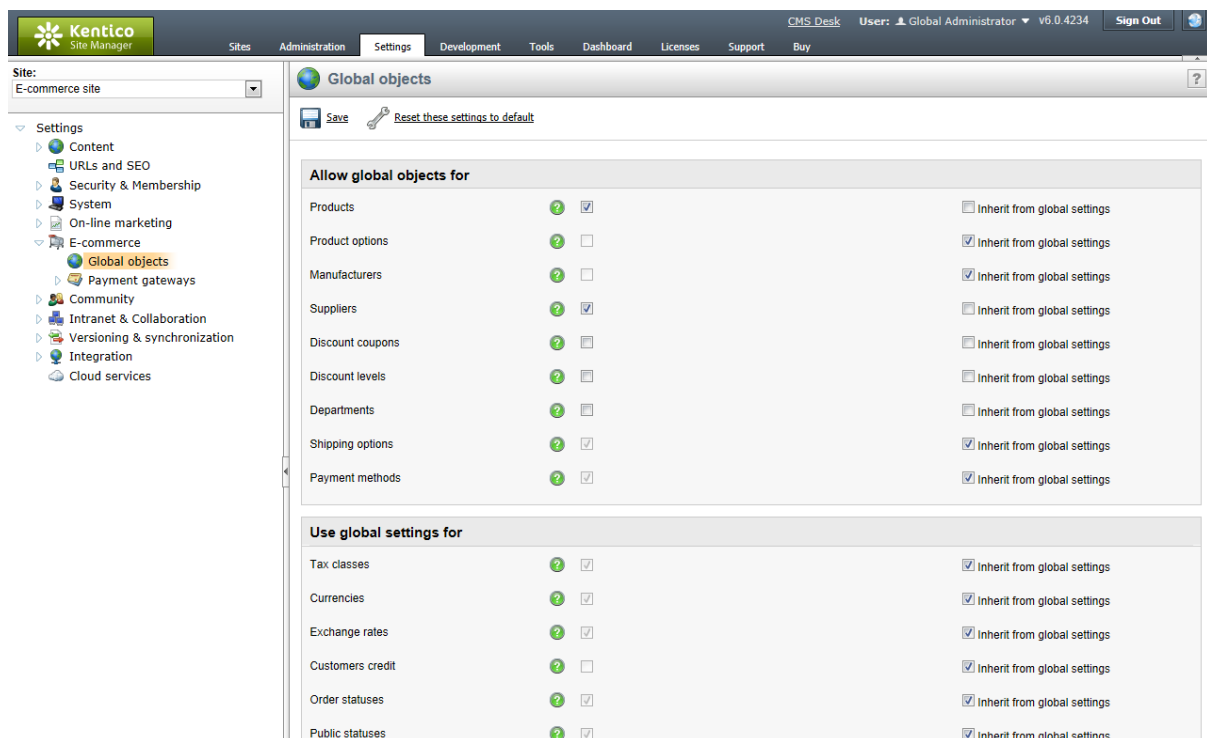
This scenario concerns multiple sites (on-line stores) which are owned and managed by different organizations. Typically, global data cannot be used because of the different owners of the on-line stores. Therefore, each site should be configured to use site data only.

## 7.3 Enabling the use of global objects

Here you will learn how certain [authorized](#) users can enable the use of global objects for the E-commerce module. There are three options to perform this task:

### Site Manager -> Settings

You can navigate to **Site Manager -> Settings**. Using the **Site** drop-down list select (*global*) to perform the configuration globally or choose one of the available sites to perform configuration for this particular site. From the Settings tree choose **E-commerce -> Global objects** and mark the check boxes next to objects to be used as global objects. Click  **Save**. For general information on how to configure your settings, please refer to [Developer's Guide -> Website settings](#).




The screenshot shows the Kentico Site Manager interface. The top navigation bar includes links for Sites, Administration, Settings, Development, Tools, Dashboard, Licenses, Support, and Buy. The user is logged in as 'Global Administrator' with version 'v6.0.4234'. The left sidebar shows the 'Settings' tree with 'Global objects' selected under 'E-commerce'. The main content area is titled 'Global objects' and contains two sections: 'Allow global objects for' and 'Use global settings for'.

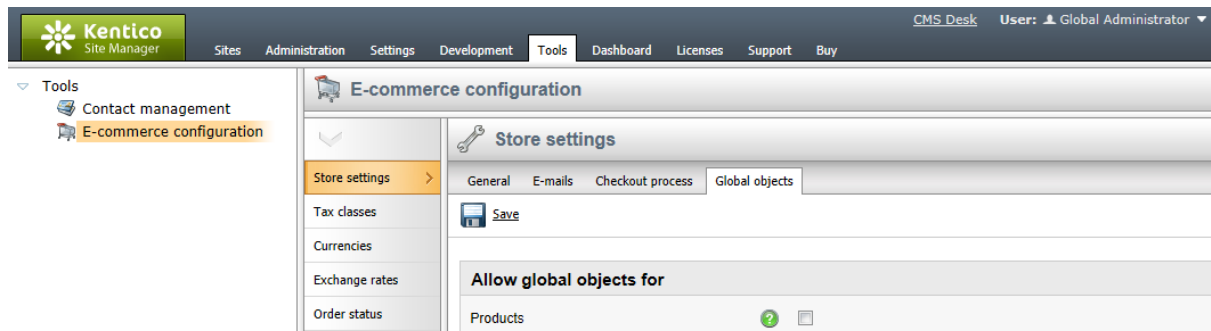
Allow global objects for		
Products	<input checked="" type="checkbox"/>	<input type="checkbox"/> Inherit from global settings
Product options	<input type="checkbox"/>	<input checked="" type="checkbox"/> Inherit from global settings
Manufacturers	<input type="checkbox"/>	<input checked="" type="checkbox"/> Inherit from global settings
Suppliers	<input checked="" type="checkbox"/>	<input type="checkbox"/> Inherit from global settings
Discount coupons	<input type="checkbox"/>	<input type="checkbox"/> Inherit from global settings
Discount levels	<input type="checkbox"/>	<input type="checkbox"/> Inherit from global settings
Departments	<input type="checkbox"/>	<input type="checkbox"/> Inherit from global settings
Shipping options	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> Inherit from global settings
Payment methods	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> Inherit from global settings

Use global settings for		
Tax classes	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> Inherit from global settings
Currencies	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> Inherit from global settings
Exchange rates	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> Inherit from global settings
Customers credit	<input type="checkbox"/>	<input checked="" type="checkbox"/> Inherit from global settings
Order statuses	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> Inherit from global settings
Public statuses	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> Inherit from global settings

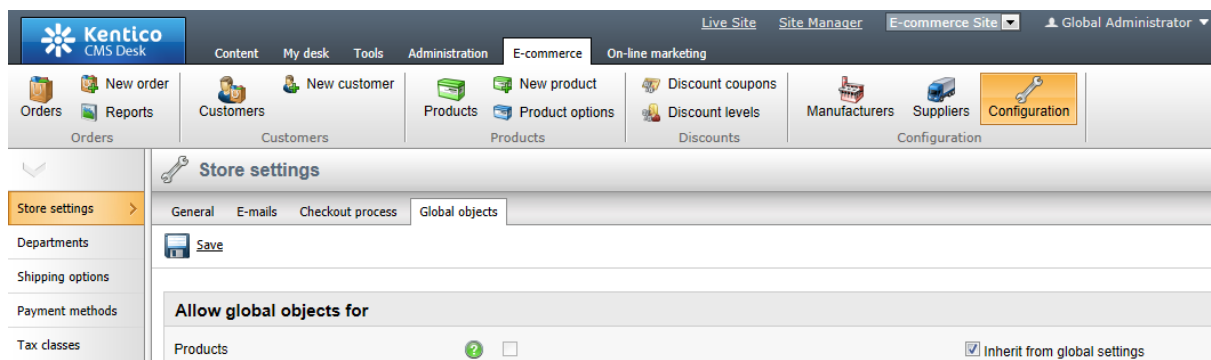
### Site Manager -> Tools

You can perform this configuration also in **Site Manager -> Tools -> E-commerce configuration -> Store settings** on the **Global objects** tab, where you need to mark the check boxes next to objects that should be used as global objects. Click  **Save** to save the changes. However, only global configuration can be performed from here; i.e. you cannot select a site. The system thus behaves as if you selected (*global*) from the **Site** drop-down list in **Site Manager -> Settings**, as described in the [Site Manager -> Settings](#) section.



## CMS Desk -> E-commerce

The third option is to go to **CMS Desk -> E-commerce -> Configuration -> Store settings**. Switch to the **Global objects** tab and mark the check boxes next to objects to be used as global objects. As you are now performing configuration for a particular site, you can optionally allow inheriting settings for the given objects from your global settings. This can be done using the **Inherit from global settings** check boxes; more details can be found in the [Website settings](#) chapter of the Developer's Guide. To save your configuration, click **Save**.



Marking check boxes in the **Allow global objects for** category allows global objects which can be used together with site-specific objects. Marking check boxes in the **Use global settings** category allows global objects which need to be used separately from their site-specific counterparts.



### Please note

In the settings of their properties, global objects can only use data from other global objects. When editing a global object, only global data are thus available in the global object's editing interface, regardless of the global objects settings for the given site. A global product can have only a global department, global manufacturer and global supplier etc. assigned. Also the price of a global product is set in the global currency.

## 7.4 Objects with both site and global option

This group includes objects which can be used on a particular site either as site-specific objects or as a combination of site-specific and global objects. The following objects belong to this group:

- Products
- Product options
- Manufacturers
- Suppliers
- Discount coupons
- Discount levels
- Departments
- Shipping options
- Payment methods

1. If global objects belonging to this group are [allowed](#) for the current site, the **Site** drop-down list with the following options is available on the object page:

- **(global and this site)** - both global and current site data is listed; global objects are marked Yes in the **Is global** column. The option does not allow you to create a new object and the link to create one is disabled.
- **(global)** - only global data is listed. The option allows [authorized](#) users to create a new global object.
- **current site** - only current site data is listed. The option allows users to create a new site-specific object; the default option.

Site: (global and this site)

New product Please choose global or particular site to create new item.

Product name:

Product type: (all)

Department: (all my departments)

Show

Display advanced filter

Actions	Product name	Product number	Product price	Available items	Public status	Internal status	Enabled	Is global
	Acer Aspire 3105WLMi		€ 4949.39	0			Yes	Yes
	Acer configurator		€ 9080.91				Yes	No
	Acer Extensa 7620G		€ 1000.00	0			Yes	No

2. If global objects belonging to this group are not allowed for the current site, the **Site** drop-down list is hidden and only current site-specific objects are listed.

Product name:

Product type: (all)

Department: (all my departments)

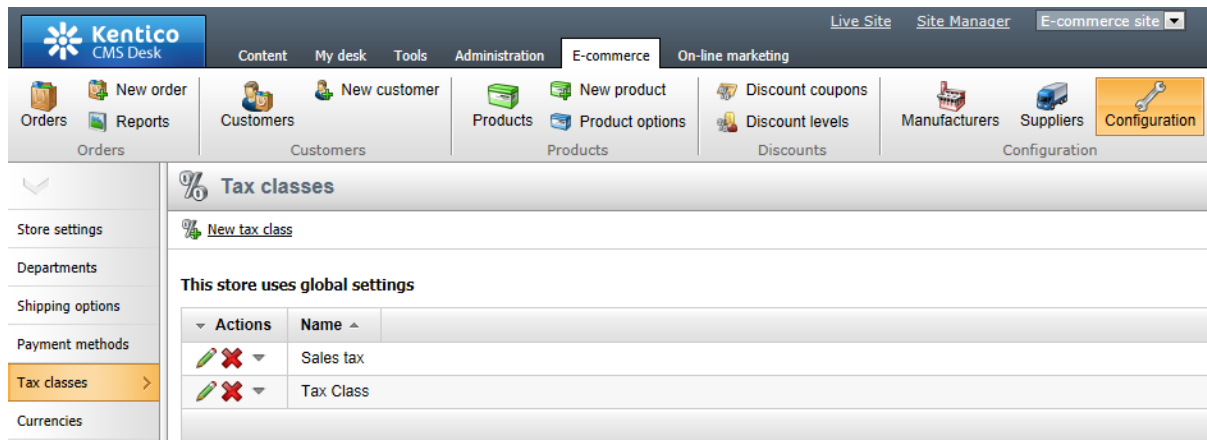
Show

## 7.5 Objects with only site or global option

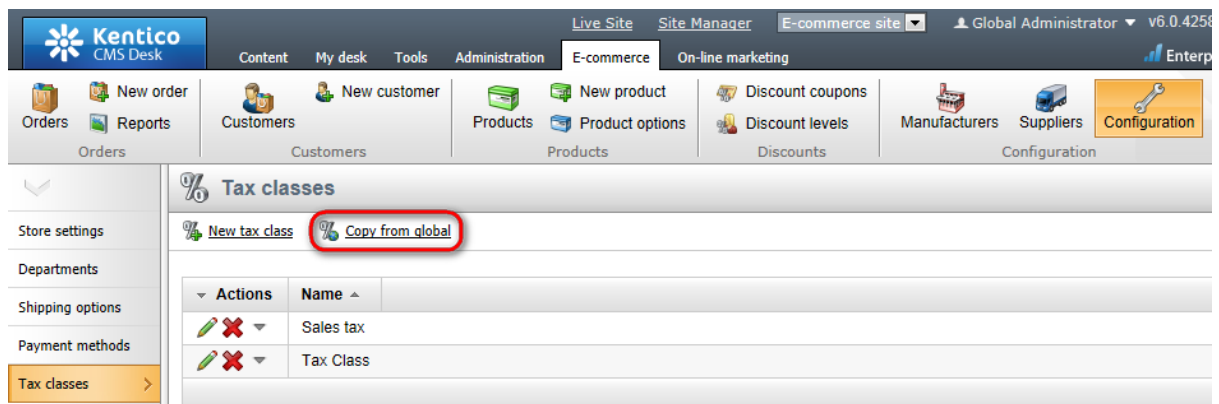
This group includes objects which can be used on a particular site only separately, either as site-specific objects or as global objects. The following objects belong to this group:

- Tax classes
- Currencies
- Exchange rates
- Customers credit
- Order statuses
- Public statuses
- Internal statuses
- Invoice

1. If global objects belonging to this group are [allowed](#) for the current site, only global objects are listed on the object page.



2. If global objects belonging to this group are not allowed for the current site, only site-specific objects are listed on the object page. Besides, the **Copy from global** link is visible, enabling you to copy global settings for the given object.



If you are copying from global settings, providing a site-specific object with the same code name like a corresponding global object is found, the site-specific object is updated with global data. On condition

that an existing site-specific object has no counterpart among global objects, it is lost. However, only if there is no dependency on other objects in the system. Otherwise, the object is disabled and remains in the system.

## 7.6 Objects with special cases

This group includes special objects from the point of view of site separation, which cannot be regarded as [objects with both site and global option](#) or [objects with only site or global option](#). The following objects belong to this group:

- [Orders](#) - are bound to a particular site.
- [Reports](#) - in most cases are bound to a particular site (Number of orders, Sales, Top 100 customers by volume of sale, Inventory). However, Kentico CMS offers also global reports (Inventory).
- [Customers](#) - an anonymous customer is bound to a particular site, same as a registered one who is bound through their user's account. This dependency enables registered customers to have preferred currency, payment option and shipping option on different sites. As for customers' credit, users can use both their global credit and credit bound to a given site. However, these cannot be mixed together, i.e. the customer can pay on the given site either with global credit or site-specific credit.

## 7.7 Security

### Enabling the use of global objects

Only global administrators can enable the use of [global objects](#).

### Managing global objects

Only global administrators and users with the **Modify global data** and **Modify global configuration** permissions (see [Developer's Guide -> Development -> Membership, permissions and security -> Permissions](#)) can create, modify and delete global objects.

For more detailed information about these permissions, same as for a comprehensive description of security in the E-commerce module, please refer to the [Security](#) chapter.

# Part

---



VIII

**Product types**

---

## 8 Product types

### 8.1 Overview

You can choose from five types of product when defining a product in **CMS Desk -> E-commerce**. Setting special properties for the given product type allows you to better control your product behavior:

- [Product](#)
- [Membership](#)
- [E-product](#)
- [Donation](#)
- [Bundle](#)

Besides, a [product with the text product option](#) is described in this chapter.

### 8.2 Product

This is the basic type of product from which the other four types, i.e. [membership](#), [e-product](#), [donation](#) and [bundle](#), are derived. All these special types share properties of this basic type, as described in [Configuration settings -> Products](#).

### 8.3 Membership

Membership as a type of product allows a user to buy membership. Detailed information about memberships in Kentico CMS can be found in [Developer's Guide -> Development -> Membership, permissions and security -> Memberships](#).

Membership as a type of [product](#) has the following special properties:

The screenshot shows the Kentico CMS Desk interface. The top navigation bar includes 'Live Site', 'Site Manager', 'E-commerce Site', and 'Global Administrator'. The main menu has tabs for 'Content', 'My desk', 'Tools', 'Administration', 'E-commerce', and 'On-line marketing'. The 'E-commerce' tab is active, showing sub-tabs for 'Orders', 'Customers', 'Products', 'Discounts', and 'Configuration'. The 'New product' button is highlighted. Below the navigation bar, the 'Product type' dropdown is set to 'Membership'. The 'Membership' section contains the following fields:



- Membership group:** A dropdown menu with '(none)' selected.
- Membership validity:** Radio buttons for 'Days', 'Weeks', 'Months', and 'Years'. The 'Until' radio button is selected.
- Until:** A text input field with a 'Now' button and a 'N/A' button.

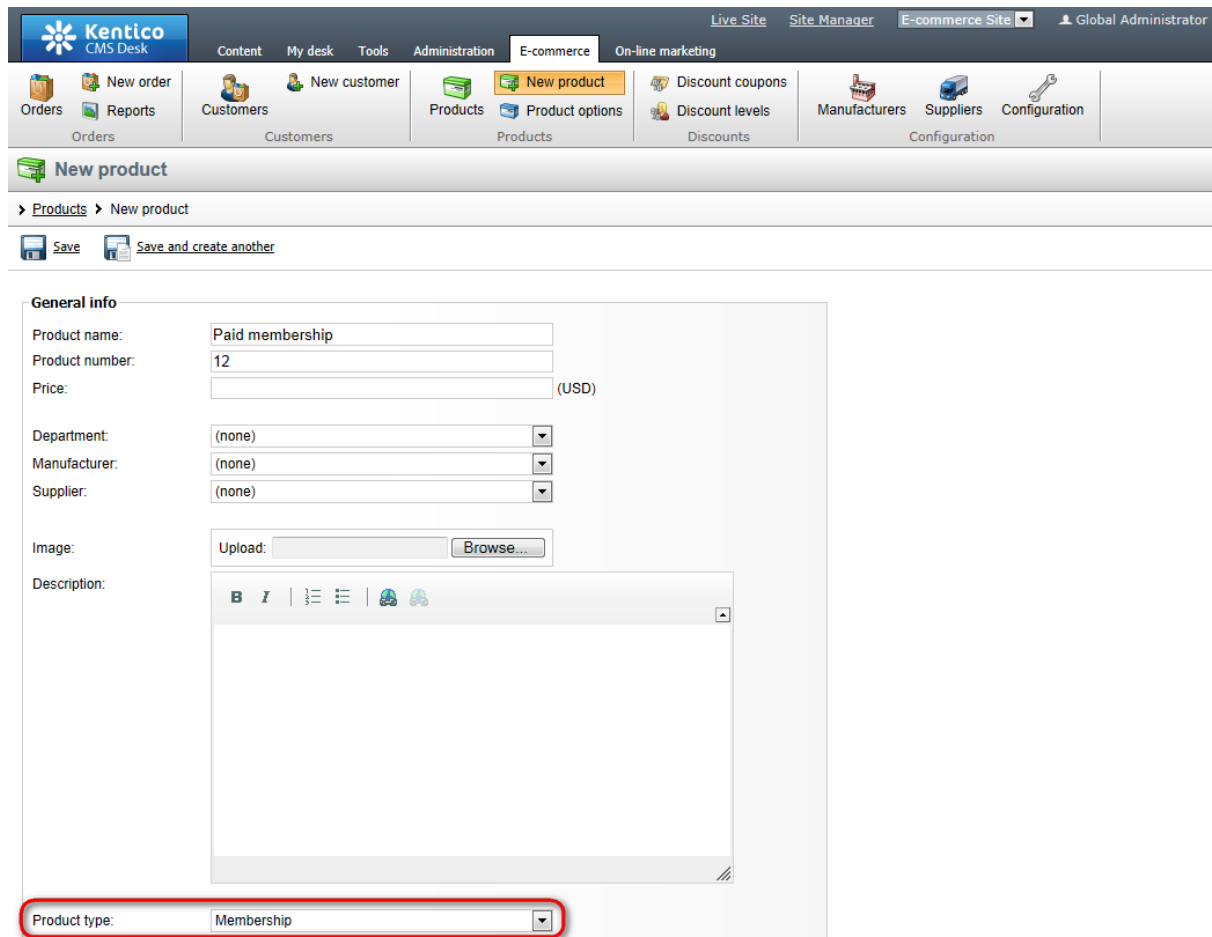
- **Membership group** - indicates to which membership group this paid membership belongs.
- **Membership validity** - can be set either for a limited period of time (days, weeks, months, years, until a given date) or for unlimited period of time. To set your membership validity for unlimited period of time, choose **Until** and leave the text box below empty.

Further in this topic you will learn how to [create](#), [buy](#) and [renew](#) your paid membership. Besides, [notifying](#) about membership expiration will be mentioned briefly:



## Creating a paid membership product

To create a new product of this type, go to **CMS Desk -> E-commerce** and on the ribbon click  **New product**. You will be redirected to the **New product** page. From the **Product type** drop-down list choose *Membership* and by filling in the form fields set your product properties as required. Click  **Save**.




The screenshot shows the Kentico CMS Desk interface for creating a new product. The top navigation bar includes 'Live Site', 'Site Manager', 'E-commerce Site', and 'Global Administrator'. The main ribbon has tabs for 'Content', 'My desk', 'Tools', 'Administration', 'E-commerce', and 'On-line marketing'. Under the 'E-commerce' tab, there are icons for 'New order', 'New customer', 'New product', 'Discount coupons', 'Discount levels', 'Manufacturers', 'Suppliers', and 'Configuration'. The 'New product' icon is highlighted. Below the ribbon, the breadcrumb path is 'Products > New product'. There are 'Save' and 'Save and create another' buttons. The 'General info' section contains the following fields:

- Product name: Paid membership
- Product number: 12
- Price: (empty) (USD)
- Department: (none)
- Manufacturer: (none)
- Supplier: (none)
- Image: Upload: (empty) Browse...
- Description: (empty text area with rich text editor toolbar)
- Product type: Membership (highlighted with a red box)

## Buying membership

From the point of view of the customer, there is no difference in the way memberships and any other types of product are bought; i.e. memberships can also be bought by going through the [standard checkout process](#).

However, the customer gains membership only when the order is marked as paid. This can be done either automatically by the system on condition that the order changes its [status](#) to one with the **Mark order as paid** property enabled (available in **CMS Desk -> E-commerce -> Configuration -> Order status -> Edit (  ) order status**)

The screenshot shows the 'Order status properties' configuration page in the Kentico CMS Desk. The 'Order status' is set to 'Completed'. The 'Mark order as paid' checkbox is checked and highlighted with a red circle. Other visible fields include 'Order status display name' (Completed), 'Order status code name' (Completed), and 'Order status color' (#DDEBFA). The 'Send notification' and 'Enabled' checkboxes are also checked. An 'OK' button is at the bottom right.

or manually by a store administrator by enabling the **Order is paid** property (available in **CMS Desk -> E-commerce -> Orders -> Edit (✎) order** on the **Billing** tab).

The screenshot shows the 'Order properties' - Billing tab in the Kentico CMS Desk. The 'Order is paid' checkbox is checked and highlighted with a red circle. Other visible fields include 'Payment method' (Cash on delivery), 'Currency' (U.S. Dollar), and 'Billing address' (Global Administrator, 45 Holcombe Crescent, Denv). There are 'Edit' and 'New' buttons next to the address field. An 'OK' button is at the bottom.

Marking a memberships order as paid, both automatically by the system and manually by an administrator, results either in renewing the existing memberships or assigning new ones.



#### Please note

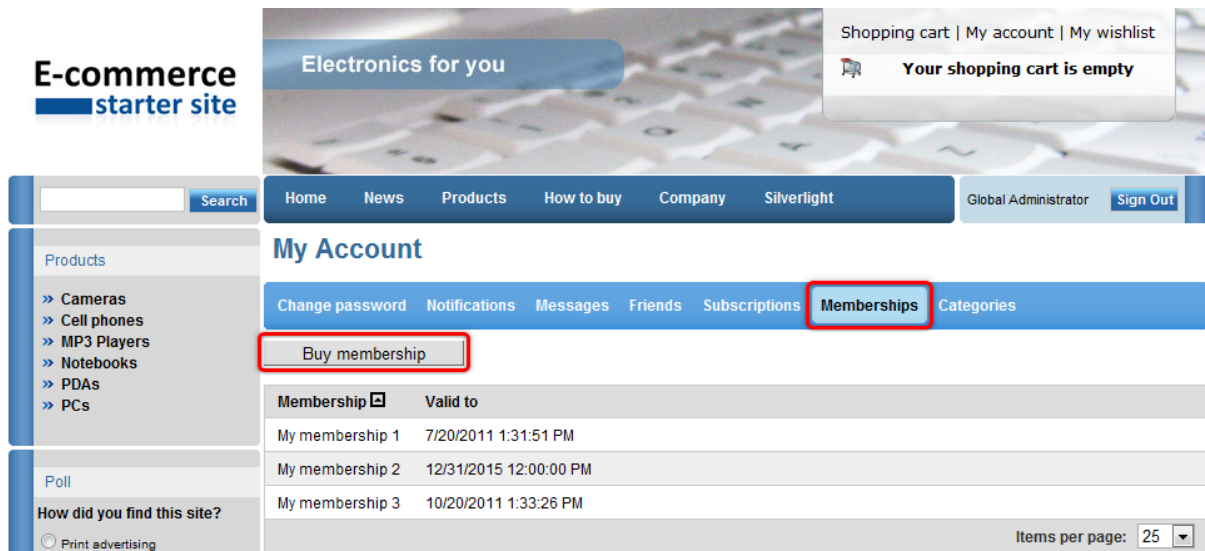
If an order for memberships with validity in time units already marked as paid is unmarked, the validity is reduced by these unpaid units.  
If an order for memberships with unlimited validity or validity set until a particular date already marked as paid is unmarked, the memberships are removed completely.

## Notifying about membership expiration

If membership is close to its expiration, the system can be configured to send the customer an automatic notification reminder. For more details, please refer to the [Memberships](#) chapter of the Developer's Guide.

## Renewing membership

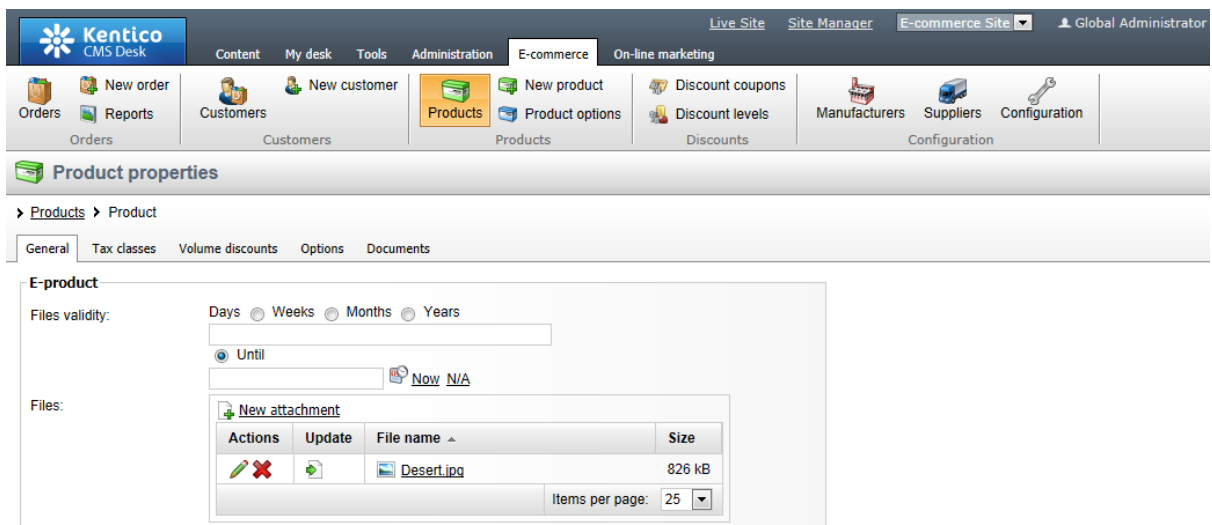
To have an expired membership renewed, the user can either ask the administrator to perform the renewal or they can buy a new membership on the My Account page on the live site. By clicking the [Memberships tab](#), the customer can view a list of their memberships and gets access to the [Buy membership button](#). By clicking this button, they will be redirected to a page allowing them to buy new memberships or renew the existing ones.



## 8.4 E-product

E-product as a type of product allows a user to buy anything which has a form of electronic file and can be downloaded. Audio files, videos and e-books are thus typically offered as e-products.



E-product as a type of [product](#) has the following special properties:

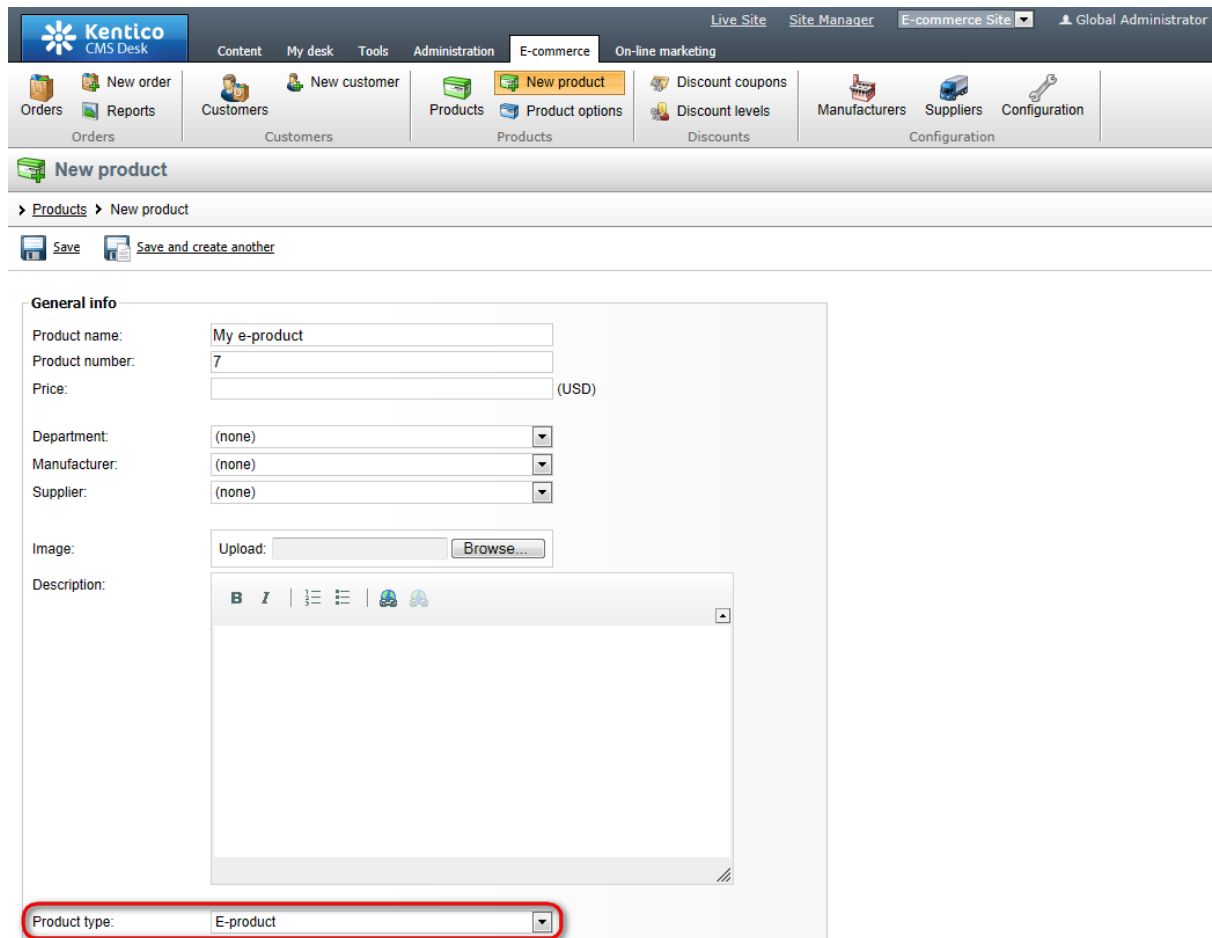


- **Files validity** - sets the period of time during which the files can be downloaded. Can be set either for a limited period of time (days, weeks, months, years, until a given date) or for unlimited period of time. To set your files validity for unlimited period of time, choose **Until** and leave the text box below empty.
- **Files** - allows to upload files to be shipped within the e-product. Only files with allowed extensions as defined in **Site Manager -> Settings -> System -> Files** can be uploaded.

Further in this topic you will learn how to [create](#) and [buy](#) your e-product and how to [download](#) files within the e-product. Besides, [notifying](#) about e-product expiration will be mentioned:

## Creating an e-product

To create a new product of this type, go to **CMS Desk -> E-commerce** and on the ribbon click  **New product**. You will be redirected to the **New product** page. From the **Product type** drop-down list choose *E-product*, set your product properties as required by filling in the form fields and upload files to be shipped within the e-product. Click  **Save**.





**Kentico CMS Desk** | Live Site | Site Manager | E-commerce Site | Global Administrator

Content | My desk | Tools | Administration | **E-commerce** | On-line marketing

Orders | New order | Reports | Customers | New customer | Products | **New product** | Discount coupons | Discount levels | Manufacturers | Suppliers | Configuration

**New product**

> Products > New product

 Save  Save and create another

**General info**

Product name:


Product number:

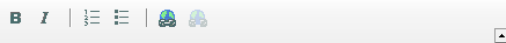
Price:  (USD)

Department:

Manufacturer:

Supplier:

Image: Upload:  

Description: 

Product type:

## Buying an e-product

From the point of view of the customer, there is no difference in the way e-products and any other types of product are bought; i.e. e-products can also be bought by going through the [standard checkout](#)

[process](#).

## Enabling e-product download links

However, the customer is able to [download the files](#) only when the order is marked as paid. This can be done either automatically by the system on condition that the order changes its [status](#) to one with the **Mark order as paid** property enabled (available in **CMS Desk -> E-commerce -> Configuration -> Order status -> Edit (✎) order status**) or manually by a store administrator by enabling the **Order is paid** property (available in **CMS Desk -> E-commerce -> Orders -> Edit (✎) order** on the **Billing** tab).

Marking an e-product order as paid, both automatically by the system and manually by an administrator, results in creating new download links for the respective e-product files with validity set as of the time of marking.



### Please note

If an order for e-products already marked as paid is unmarked, the download links related to these e-products are deleted so the customer is not able to download the files.

## Downloading e-product files

The user can download the files on the My Account page on the live site. By clicking the [Orders tab](#) they can view a list of all their orders, where each order containing at least one e-product has a **View** link in the **Downloads** column. By clicking the link, the user can view a list of e-products bought in the given order, together with download links details.

**E-commerce starter site**

Electronics for you

Shopping cart | My account | My wishlist  
Your shopping cart is empty

Home News Products How to Company Silverlight Andrew Jones Sign Out

**My Account**

Details Addresses **Orders** Credit Change password Notifications Messages Subscriptions Memberships

Order ID	Order date	Total price	Status	Tracking number	Invoice	Downloads
4	10/3/2011 3:10:02 PM	\$300.00	Payment received		<a href="#">View</a>	<a href="#">View</a>
3	10/3/2011 3:03:22 PM	\$100.00	Payment received		<a href="#">View</a>	<a href="#">View</a>

Items per page: 25

## Notifying about e-product expiration

If an e-product is close to its expiration, the system can be configured to send the customer an automatic notification reminder.

A site-specific scheduled task sends a notification e-mail about all expiring e-products bought on the current site. A global scheduled task named **E-product reminder** (in Site Manager -> Administration -> Scheduled tasks) sends a notification e-mail about all expiring e-products bought on all available sites.

The content of the reminder e-mails is taken from the **Ecommerce.EproductExpirationNotification** e-mail template, which can be edited via the **Site Manager -> Administration -> E-mail templates** interface. However, if there is a site-specific template with the same code name and a site-specific scheduled task had been configured, an e-mail based on this site-specific template is sent.

Detailed information on scheduled tasks in Kentico CMS can be found in [Developer's Guide -> Development -> Scheduler](#).

## 8.5 Donation

Donation as a type of product allows a user to donate a sum of money. It behaves as a standard product and can be offer stand-alone or with various product options. However, you cannot put a donation into a [bundle](#) or select it as a [product option](#).

Donation as a type of [product](#) has the following special properties:

The screenshot shows the Kentico CMS Desk interface with the 'E-commerce' tab selected. The 'New product' button is highlighted. Below the ribbon, the 'Donation' form is displayed with the following fields:

- Minimum donation:** 10 (USD)
- Maximum donation:** 10000 (USD)
- Allow private donation:** ☒

- **Minimum donation** - the minimum sum of money that the user can donate.
- **Maximum donation** - the maximum sum of money that the user can donate.
- **Allow private donation** - indicates whether the user is allowed to donate privately. If true, the option is displayed during the purchase process when adding the product into the shopping cart (the user will thus not be present in public donation lists, reports etc).

Further in this topic you will learn how to [create](#) and [buy](#) a donation and how to view donations [reports](#):

### Creating a donation

To create a new product of this type, go to **CMS Desk -> E-commerce** and on the ribbon click **New product**. You will be redirected to the **New product** page. From the **Product type** drop-down list choose *Donation*, set your product properties as required by filling in the form fields and click **Save**.

Kentico CMS Desk

Content My desk Tools Administration E-commerce On-line marketing

Orders Reports Customers New customer Products Product options Discount coupons Discount levels Manufacturers Suppliers Configuration

New product

Products > New product

Save Save and create another

**General info**

Product name: My donation

Product number: 288

Price: 10000 (USD)

Department: (none)

Manufacturer: (none)

Supplier: (none)

Image: Upload: C:\Users\Public\Pictures\ Browse...

Description: This is my donation.

Product type: Donation

## Buying a donation

From the point of view of the customer, there is no difference in the way donations and any other types of product are bought; i.e. donations can also be bought by going through the [standard checkout process](#).

## Reporting donations

In Kentico CMS, you can view also donation reports. If you decide to do so, navigate to **CMS Desk -> E-commerce -> Reports** and from the **Reports** tree choose **Donations**. You can select to view all donations at once or you can select to view public and [private](#) donations separately. However, individual donors are displayed in the list only when the order containing the donation is paid.

**Donations**

Save Print

Donation name:

Donation amount from:

Donation amount to:

Donations: (all)

Donor:

Donor company:

Order date from:  Now

Order date to:  Now

Update

Donation name	Donation amount	Donation is private	Donor	Donor company	Order date
Donation 3	200000	<input type="checkbox"/>	John Burrow		10/06/2011
Donation 3	50000	<input type="checkbox"/>	John Burrow		10/06/2011
Donation 2	3000	<input checked="" type="checkbox"/>	John Burrow		10/06/2011
Donation 3	2500	<input type="checkbox"/>	Laura Summers		10/06/2011
Donation 1	575	<input checked="" type="checkbox"/>	John Burrow		10/06/2011
Donation 1	50	<input checked="" type="checkbox"/>	Brad Stanton		10/06/2011

## 8.6 Bundle

Bundle as a type of product allows a user to buy several different products as one combined product. This marketing strategy can be highly effective if selling related products or products which wouldn't sell well enough separately. However, you can sell any individual products within separate product bundles.

Bundle as a type of [product](#) has the following special properties:

**Bundle**

Remove from inventory: ☒ Remove bundle only  
☐ Remove each product separately  
☐ Remove both bundle and products

Products:

Product name
<input type="checkbox"/> Microsoft Office Professional 2010
<input type="checkbox"/> Microsoft Windows 7 Ultimate



Remove selected Add products

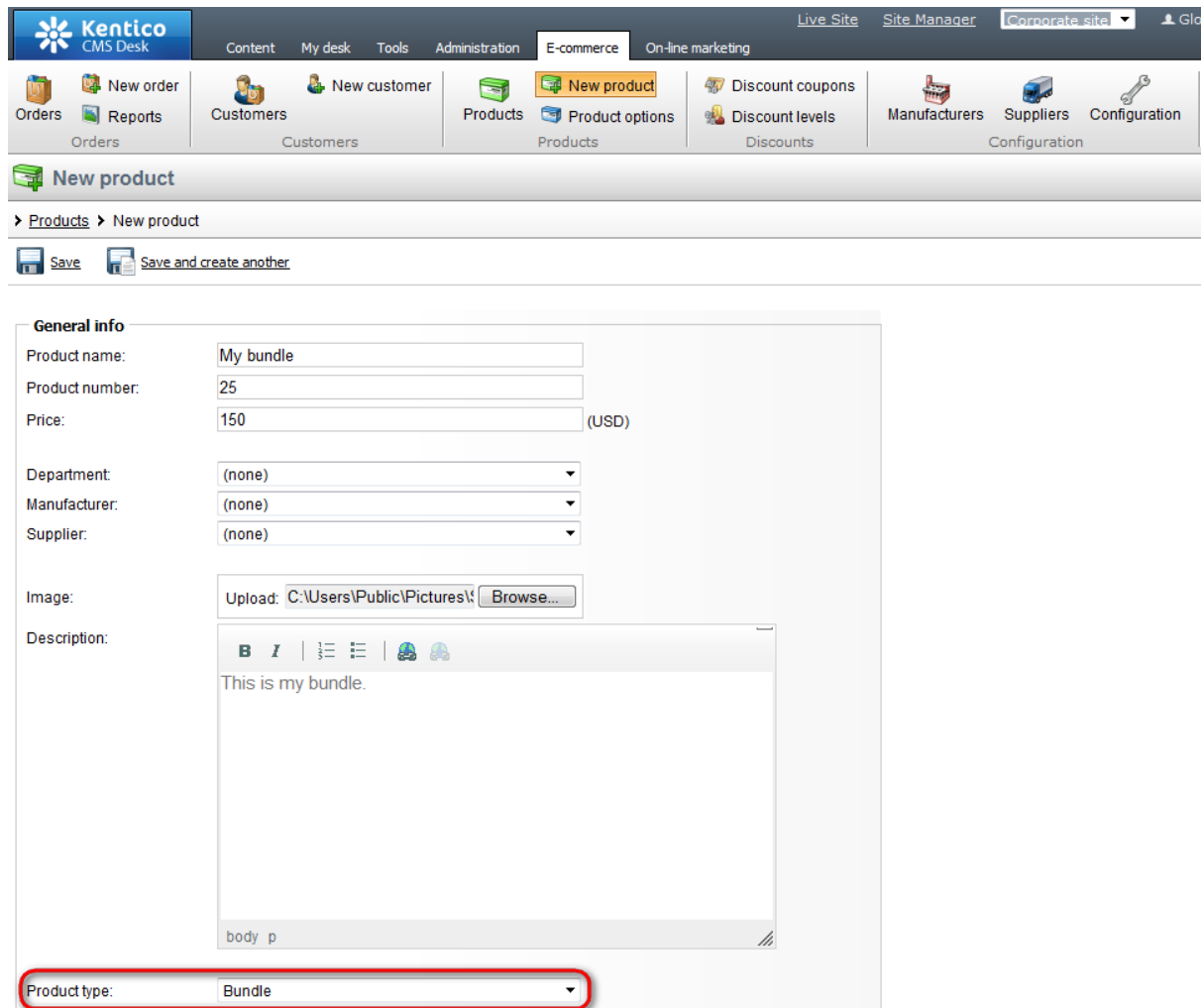
- **Remove from inventory** - defines the behavior of the inventory.
  - **Remove bundle only** - the bundle is removed from the inventory as a whole.
  - **Remove each product separately** - individual items of the bundle are removed from the respective inventories.
  - **Remove both bundle and products** - both the bundle and individual items of the bundle are removed from the respective inventories.
- **Products** - allows to add products to be shipped within the bundle. The following types of product cannot be added: bundle, donation.

Further in this topic you will learn how to [create](#) and [buy](#) your bundle:



## Creating a bundle

To create a new product of this type, go to **CMS Desk -> E-commerce** and on the ribbon click  **New product**. You will be redirected to the **New product** page. From the **Product type** drop-down list choose *Bundle*, set your product properties as required by filling in the form fields and add products to be shipped within the bundle. Click  **Save**.





**Kentico CMS Desk** | Content | My desk | Tools | Administration | **E-commerce** | On-line marketing | Live Site | Site Manager | *Current site* | User

Orders | New order | Reports | Customers | New customer | Products | **New product** | Discount coupons | Discount levels | Manufacturers | Suppliers | Configuration

**New product**

> Products > New product

 Save  Save and create another

**General info**

Product name:


Product number:


Price:  (USD)

Department:

Manufacturer:

Supplier:

Image: Upload:  

Description:   

This is my bundle.

body p

Product type:

## Buying a bundle

From the point of view of the customer, there is no difference in the way bundles and any other types of product are bought; i.e. bundles can also be bought by going through the [standard checkout process](#).

### Please note

During its price calculation, the bundle behaves as a separate product. Neither the taxes nor discounts related to individual bundle items are thus reflected in this



calculation.

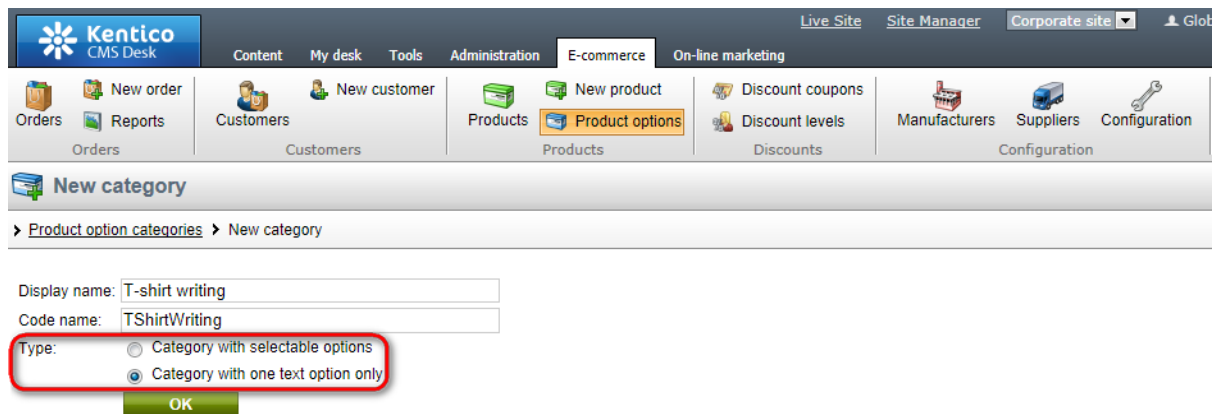
## 8.7 Product with the text product option

In Kentico CMS any type of product can be assigned with the text [product option](#). This is a special product option enabling you e.g. to put writing on a T-shirt etc. From the point of view of the user, the option is defined for and assigned to a selected product as any other product option.

Further in this topic you will learn how to [create](#) a new text product option, how to [assign](#) the category containing this option to a product, how to [buy](#) this product and how to [edit](#) the product option text in the product order.

### Creating the text product option

1. To create a new text product option, you first need to define a text option category. Go to **CMS Desk - > E-commerce**, click  **Product options** on the ribbon and on the **Product option categories** page click the  **New category** link.
2. Enter the category display name and code name and as the category type choose **Category with one text option only**. Click **OK**.



The screenshot shows the Kentico CMS Desk interface. The top navigation bar includes 'Live Site', 'Site Manager', and 'Corporate site'. The main ribbon is 'E-commerce', with 'Product options' selected. Below the ribbon, the 'New category' form is displayed. The form has the following fields:

- Display name: T-shirt writing
- Code name: TShirtWriting
- Type: ☒ Category with one text option only (selected), ☐ Category with selectable options

The 'OK' button is highlighted with a red box.

3. Set the [category properties](#) on the **General** tab as required by filling in the form fields and click **OK** to save the changes. Please note that the **Text type** property gives you the option to choose from either a text box or text area.

**Category properties**

Product option categories > T-shirt writing

General Options

Display name: T-shirt writing

Code name: TShirtWriting

Text type: Text box

Display price: ☒

Default text: DEFAULT (+ \$0.00)

Description: T-shirt writing

Enabled: ☒


OK

4. If you now switch to the **Options** tab, you can see that the new text product option was created automatically, with the same name like the name of the given category. **Edit** (✎) the option as required by filling in the form fields on its **General** tab and add or remove tax classes on its **Tax classes** tab.

**Category properties**

Product option categories > T-shirt writing

General Options

Actions	Product name	Product number	Product price	Available items	Enabled
	T-shirt writing		\$0.00		Yes

## Assigning the category containing the text product option to a product

To assign the category containing the text product option to a product, please follow the standard product options assignment procedure, as described in detail in the [Adding product options](#) chapter.

**Kentico CMS Desk** Live Site Site Manager Corporate site GLOB

Content My desk Tools Administration E-commerce On-line marketing

Orders Reports Orders Customers New customer Customers Products New product Product options Products Discounts Discount coupons Discount levels Discounts Manufacturers Suppliers Configuration

**Product properties**

Products > T-shirt

General Tax classes Volume discounts Options Documents

The changes were saved.

The following categories with product options are assigned to the product:

<input type="checkbox"/>	Category name
<input type="checkbox"/>	T-shirt writing

Remove selected Add categories

## Buying the product

From the point of view of the customer, there is no difference in the way products with text product options and any other types of product are bought; i.e. products with text product options can also be bought by going through the [standard checkout process](#).

The screenshot displays the Kentico CMS E-commerce interface. The main navigation bar includes 'Content', 'My desk', 'Tools', 'Administration', 'E-commerce', and 'On-line marketing'. The 'E-commerce' tab is active. The left sidebar shows 'New order' and 'New customer' buttons. The main content area shows 'Step 2 of 6 - Add some products to the shopping cart'. A 'Shopping cart' section is visible with an 'Add item' button. A red arrow points from the 'Add item' button to the 'Add order items' dialog box. The dialog box is titled 'Add order items' and shows a 'T-shirt' product with a unit price of \$40.00. It prompts the user to specify product options before adding to the shopping cart. The 'Color' option is set to 'Blue (+ \$0.00)'. The 'T-shirt writing' option is set to 'DEFAULT (+ \$20.00)'. The total price (without tax) is \$60.00. A red box highlights the 'T-shirt writing' option. The dialog box also includes a quantity input field set to '1' and an 'Add to cart' button.

## Editing the product option text in the product order

If the **order is not marked as paid**, the product option text together with other order item properties can be edited in the given product order. However, editing is enabled only if the *web.config* file is properly configured as described in detail in the [Web.config settings](#) topic in the Configuration settings section of this guide.

The screenshot displays the Kentico CMS Desk interface. A modal dialog titled 'Order item properties' is open, showing fields for 'Product name' (T-shirt writing), 'Unit price' (20.00 USD), 'Units' (1), and 'Text' (DEFAULT). Below the dialog, the 'Shopping cart' section is visible, showing a table with columns: Remove, Product name, Units, Unit price, Tax, and Subtotal. The cart contains three items: 'T-shirt' (1 unit, 40.00), '- Green' (1 unit, 0.00), and '- T-shirt writing "DEFAULT"' (1 unit, 20.00). A red circle highlights the third item, and a red arrow points from it to the 'Order item properties' dialog. The bottom right of the interface shows the total shipping (\$8.00) and total price (\$68.00), with an 'Update' button.

Remove	Product name	Units	Unit price	Tax	Subtotal
<input type="checkbox"/>	T-shirt	1	40.00	0.00	40.00
<input type="checkbox"/>	- Green	1	0.00	0.00	0.00
<input type="checkbox"/>	- T-shirt writing "DEFAULT"	1	20.00	0.00	20.00

Total shipping: \$8.00  
Total price: \$68.00  
[Update](#)

Please note that an order is marked as paid either automatically by the system if it changes its [status](#) to one with the **Mark order as paid** property enabled (available in **CMS Desk -> E-commerce -> Configuration -> Order status -> Edit (✎) order status**) or manually by a store administrator by enabling the **Order is paid** property (available in **CMS Desk -> E-commerce -> Orders -> Edit (✎) order** on the **Billing** tab).

# Part

---



IX

**Purchase process and payment gateways**

## 9 Purchase process and payment gateways

### 9.1 Purchase process overview

The standard purchase (checkout) process looks like this:

1. The user browses the website and adds the products to the **shopping cart**. They can modify the number of items in the cart. They can also insert a discount coupon.
2. The user clicks the **Checkout** button. They can choose from three options:
  - **Sign in using an existing account** - if they already have a user account
  - **Create a new account**
  - **Continue as anonymous customer** - this option doesn't require the user to create a user account, but they need to enter their details with every purchase. You can enable/disable this option at **CMS Desk -> Tools -> E-commerce -> Configuration -> Store Settings -> Allow anonymous customers**.
3. The user enters/updates customer data, including first name, last name, e-mail and company name.
4. The customer enters billing/shipping address or chooses from the list of previously used addresses. The user can also choose to enter an additional company headquarters address that can be enabled at **CMS Desk -> Tools -> E-commerce -> Configuration -> Store Settings -> Use an extra company address**.
5. The user chooses the payment and shipping method.
6. The user previews the order.
7. The user completes the order or continues with on-line payment. The order status is set to **New**. The order notification e-mail is sent to store owner and to the customer if it's configured in **CMS Desk -> Tools -> E-commerce -> Configuration -> Store Settings** dialog.
8. The payment is processed by the payment gateway. If the payment succeeds/fails, the order status is set to values specified at **CMS Desk -> Tools -> E-commerce -> Configuration -> Payment methods -> <select some payment method> -> Order status when payment succeeds/fails**. A payment notification e-mail is sent to store owner and to the customer if it's configured in **CMS Desk -> Tools -> E-commerce -> Configuration -> Store Settings** dialog and if the payment is successfully completed.
9. The user is redirected to the page configured in the properties of the **Shopping cart** web part, in the **Default URL after purchase** value.

#### Shopping cart content

The shopping cart content is stored in the database and it's bound to the current user name. If the user is anonymous, the cart ID is stored in a browser cookie. It ensures that the cart content is preserved even if the session is lost during application restart. You can find more details on how the shopping cart is retrieved in [Figure D - Shopping cart retrieval](#).

When the checkout process is completed (regardless if the payment succeeds or fails), a new order is created and the shopping cart content is moved to the order. The cart is empty then.

The cart content is accessible in your code using the **ShoppingCartInfoObj** class in the checkout process steps.

## 9.2 Customizing the purchase process

The purchase process can be enhanced with your custom steps or you can remove some steps. This can be done at **CMS Desk -> Tools -> E-commerce -> Configuration -> Store settings -> Checkout process**.

For each step, you can specify the following values:

<b>Caption</b>	The step caption in the checkout process.
<b>Code name</b>	The name of the site used in the code.
<b>Image file name</b>	Name of the file used in the checkout process header. The image must be stored in folder ~\App_Themes\<stylesheet name>\Images\ShoppingCart.
<b>Control path</b>	Virtual path to the ASCX control representing the dialog in the given step. Example:  ~/CMSEcommerce/ShoppingCart/ShoppingCartOrderAddresses.ascx
<b>Show on the live site</b>	Indicates if this step should be used in the checkout process on the live site. (see below)
<b>Show in Customer section</b>	Indicates if this step should be used in the checkout process when creating a new order for the given customer. (see below)
<b>Show in Order section</b>	Indicates if this step should be used in the checkout process for a new order created from the administration interface. (see below)
<b>Show in Order items section</b>	Indicates if this step should be used in the checkout process when editing items of an existing order. (see below)

See chapter [Developing custom dialogs for the checkout process](#) for more details on development of custom steps.

### Types of the checkout process

There are four different types of checkout process:

- Checkout process 1 - New order on the live site
- Checkout process 2 - New order from section **CMSDesk -> Tools -> E-commerce -> Customers/ (some customer) -> Orders**.
- Checkout process 3 - New order from section **CMSDesk -> Tools -> E-commerce -> Orders**.
- Checkout process 4 – Editing existing order from section **CMSDesk -> Tools -> E-commerce -> Orders -> (some order) -> Items**.

Below, you can find the screenshots of these types:

#### Checkout process 1 – New order on the live site





### Checkout process 3 - New order from section CMSDesk -> Orders

The screenshot shows the Kentico CMS Desk interface. The top navigation bar includes 'Content', 'My desk', 'Tools', 'Administration', 'E-commerce', and 'On-line marketing'. The 'E-commerce' section is active, showing sub-sections like 'Orders', 'Customers', 'Products', 'Discounts', 'Manufacturers', 'Suppliers', and 'Configuration'. The 'New order' button is highlighted. Below the navigation bar, the 'New order' page is displayed, showing 'Step 1 of 6 - Select customer'. A 'Customer' dropdown menu is present, with a 'Select' button next to it. A 'Next' button is located at the bottom right of the page.

### Checkout process 4 - Editing existing order from section CMSDesk -> Tools -> E-commerce -> Orders -> (some order) -> Items.

The screenshot shows the Kentico CMS Desk interface. The top navigation bar is the same as in the previous screenshot. The 'Order properties' page is displayed, showing 'Step 1 of 2 - Add some products to the shopping cart'. The 'Shopping cart' section is active, showing a table with columns: 'Remove', 'Product name', 'Units', 'Unit price', 'Tax', and 'Subtotal'. The table contains one item: 'Apple MacBook Air' with a unit price of 2094.00 and a subtotal of 2094.00. Below the table, there is a 'Currency' dropdown set to 'U.S. Dollar', a coupon code input field, and a 'Total price' of \$2103.60. An 'Update' button is located at the bottom right of the page.

## 9.3 Developing custom dialogs for the checkout process

### Creating a custom checkout process step

1. Create new web user control (\*.ascx) and place your own form controls into it.
2. Go to its code behind and set control class to inherit from **CMS.EcommerceProvider**.

#### ShoppingCartStep.

3. There are several methods you can override to reach the required functionality:
  - **IsValid()** – Validates current step custom data and returns validation result. True – all step data is correct and can be processed, False – some step data is not correct or missing and cannot be processed. In this case appropriate error message should be displayed. By default, it returns True.
  - **ProcessStep()** – Processes current step information (updates shopping cart data) and returns result of this action. True – shopping cart data was updated successfully and customer can be moved to the next checkout process step, False – shopping cart update failed and customer cannot be moved to the next step. In this case appropriate error message should be displayed.

By default, it returns True.

- **ButtonBackClickAction()** – Defines action which is run after the Back button is clicked. By default, the parent shopping cart control method ButtonBackClickAction() is called which moves customer one step backward in the checkout process.

- **ButtonNextClickAction()** – Defines action which is run after the “Next button” is clicked. By default, the parent shopping cart control method ButtonNextClickAction() is called which moves customer one step forward in the checkout process when the current step data is valid and processed successfully.

4. There are several properties you should use to get or set required information:

- **ShoppingCartControl** – parent shopping cart control the step belongs to

- **ShoppingCartInfoObj** – shopping cart object which stores all data during the checkout process

- **CheckoutProcessStep** – checkout process step information

5. The step control is created and can be registered as your custom checkout process step.

If the control represents checkout process steps in different [checkout process types](#) and these steps differ from each other only a little, you can create one control and specify a different behavior depending on the checkout process type as follows:

**[C#]**

```
using System;
using CMS.EcommerceProvider;
using CMS.GlobalHelper;

switch (this.ShoppingCartControl.CheckoutProcessType)
{
    case CheckoutProcessEnum.LiveSite:
        // Here comes the code which will be run only
        // when it is a checkout process on the live site
        break;

    case CheckoutProcessEnum.CMSDeskOrder:
        // Here comes the code which will be run only
        // when it is a checkout process in the section CMSDesk/Tools/E-commerce/
        Orders
        break;

    default:
        // Here comes the code which will be run in all other cases
        break;
}
```

## Example 1 - My step

The following example shows a simple checkout process step definition. It displays the total price and an editable field to insert a customer comment. After the **Next** button is clicked, the editable field is checked for emptiness. If it is not empty the customer comment is saved and shopping cart data is updated. Otherwise, an appropriate error message is displayed. If the customer is a member of role “VipCustomers”, an extra step with an additional form for VIP customers is loaded (MyVipStep.ascx). The **Back** button action is not overridden.

**[MyStep.ascx]**

Please note: If you installed the Kentico CMS project as a web application, you need to rename the *CodeFile* property on the first line to *Codebehind* for the code example to be functional.

```
<%@ Control Language="C#" AutoEventWireup="true" CodeFile="MyStep.ascx.cs"
Inherits="MyStep" %>
<asp:Label ID="lblError" runat="server" EnableViewState="false" Visible="false"></
asp:Label>
<table>
  <tr>
    <td><asp:Label ID="lblTotalPrice" runat="server" /></td>
    <td><asp:Label ID="lblTotalPriceValue" runat="server" /></td>
  </tr>
  <tr>
    <td><asp:Label ID="lblComment" runat="server" /></td>
    <td><asp:TextBox ID="txtComment" runat="server" TextMode="MultiLine" Rows
="3" /></td>
  </tr>
</table>
```

[MyStep.ascx.cs]

[C#]

```
using System;
using System.Data;
using System.Configuration;
using System.Collections;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Web.UI.HtmlControls;

using CMS.GlobalHelper;
using CMS.Ecommerce;
using CMS.EcommerceProvider;

public partial class MyStep : ShoppingCartStep
{
    protected void Page_Load(object sender, EventArgs e)
    {
        // Initialize labels
        lblTotalPrice.Text = ResHelper.GetString("MyStep.TotalPrice");
        lblComment.Text = ResHelper.GetString("MyStep.Comment");

        // Display rounded and formatted total price
        lblTotalPriceValue.Text =
CurrencyInfoProvider.GetFormattedPrice(ShoppingCartInfoObj.RoundedTotalPrice,
ShoppingCartInfoObj.CurrencyInfoObj);

        if (!this.ShoppingCartControl.IsCurrentStepPostBack)
        {
            // Load customer comment

```

```
        txtComment.Text = this.ShoppingCartInfoObj.ShoppingCartCustomData[
"CustomerComment"];
    }

}

/// <summary>
/// Validates current step data.
/// </summary>
/// <returns></returns>
public override bool IsValid()
{
    // Check customer comment for emptiness
    if (txtComment.Text.Trim() == "")
    {
        // Display error message
        lblError.Text = ResHelper.GetString("MyStep.Error.CommentMissing");
        lblError.Visible = true;

        // Data are not correct - customer comment missing
        return false;
    }
    else
    {
        // Data are correct
        return true;
    }
}

/// <summary>
/// Process current step data
/// </summary>
/// <returns></returns>
public override bool ProcessStep()
{
    // Update shopping cart with customer comment
    this.ShoppingCartInfoObj.ShoppingCartCustomData["CustomerComment"] =
txtComment.Text.Trim();

    try
    {
        // Update shopping cart in database
        ShoppingCartInfoProvider.SetShoppingCartInfo(this.
ShoppingCartInfoObj);

        // Current step data were processed and saved succesfully
        return true;
    }
    catch
    {
        // Display error message
        lblError.Text = ResHelper.GetString("MyStep.Error.ShoppingCartUpdate"
);
        lblError.Visible = true;
    }
}
```

```

        // Current step data update failed
        return false;
    }
}

/// <summary>
/// Action after the "Next button" is clicked
/// </summary>
public override void ButtonNextClickAction()
{
    // If customer is registered and is a member of role "VipCustomers"
    if ((this.ShoppingCartInfoObj.UserInfoObj != null)
        && (this.ShoppingCartInfoObj.UserInfoObj.IsInRole("VipCustomers",
this.ShoppingCartInfoObj.SiteName)))
    {
        if (IsValid() && ProcessStep())
        {
            try
            {
                // Load extra step for VIP customers which is not included
                // in standard checkout process definition
                ShoppingCartStep ctrl =
(ShoppingCartStep)this.Page.LoadControl("~/CMSEcommerce/ShoppingCart/MyVipStep.
ascx");

                this.ShoppingCartControl.LoadStep(ctrl);

                // Note: Current step index is not increased
            }
            catch
            {
                // Error while loading extra step control -> Do standard
action
                base.ButtonNextClickAction();
            }
        }
    }
    else
    {
        // Do standard action (validate step data, process step data, load
next step)
        base.ButtonNextClickAction();
    }
}
}

```

### My VIP step

This is an external checkout process step which is not included in standard checkout process definition. It is loaded only when the current customer is a member of role "VipCustomers", however, you will need to use your own condition for loading your external steps. There is no data validation (Vip customer comment can be empty). Neither "Back button action" nor "Next button action" is overridden so it means the standard methods are called after the **Back** button and **Next** button are clicked.

**[MyVipStep.ascx]**

Please note: If you installed the Kentico CMS project as a web application, you need to rename the *CodeFile* property on the first line to *Codebehind* for the code example to be functional.

```
<%@ Control Language="C#" AutoEventWireup="true" CodeFile="MyVipStep.ascx.cs"
Inherits="MyVipStep" %>
<asp:Label ID="lblError" runat="server" EnableViewState="false" Visible="false"></
asp:Label>
<asp:Label ID="lblComment" runat="server" />
<asp:TextBox ID="txtComment" runat="server" TextMode="MultiLine" Rows="3" />
```

**[MyVipStep.ascx.cs]****[C#]**

```
using System;
using System.Data;
using System.Configuration;
using System.Collections;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Web.UI.HtmlControls;

using CMS.GlobalHelper;
using CMS.Ecommerce;
using CMS.EcommerceProvider;

public partial class CMSEcommerce_TestVipCustomers : ShoppingCartStep
{
    protected void Page_Load(object sender, EventArgs e)
    {
        // Initialize label
        lblComment.Text = ResHelper.GetString("MyVipStep.Comment");

        if (!this.ShoppingCartControl.IsCurrentStepPostBack)
        {
            // Load VIP customer extra comment
            txtComment.Text =
Convert.ToString(this.ShoppingCartInfoObj.ShoppingCartCustomData["VipComment"]);
        }
    }

    /// <summary>
    /// Process current step data
    /// </summary>
    /// <returns></returns>
    public override bool ProcessStep()
```

```

{
    // Update shopping cart with VIP customer extra comment
    this.ShoppingCartInfoObj.ShoppingCartCustomData[ "VipComment" ] =
txtComment.Text.Trim();

    try
    {
        // Update shopping cart in database
        ShoppingCartInfoProvider.SetShoppingCartInfo(this.
ShoppingCartInfoObj);

        // Current step data were processed and saved succesfully
        return true;
    }
    catch
    {
        // Display error message
        lblError.Text = ResHelper.GetString( "MyStep.Error.ShoppingCartUpdate"
);

        lblError.Visible = true;

        // Current step data update failed
        return false;
    }
}
}

```



#### Important notes on step order

Notice that data of all steps (including Kentico CMS standard shopping cart steps) are always processed by themselves, not by the parent shopping cart control. It means if you reorder standard checkout process steps you can experience a strange behavior because of missing information or omitting some important action.

For example if you move step “Order preview” in the standard checkout process definition for the live site before steps “Select billing and shipping address” and “Select payment and shipping methods” some order preview data will be missing (billing and shipping addresses, payment method and shipping option) because they were not entered yet. But even worse is that entered data of the following two steps (“Select billing and shipping address”, “Select payment and shipping methods”) will not be saved.

This happens because the order is saved after the “Next button” of the step “Order preview” is clicked and these two steps include completely different actions after their “Next buttons” are clicked.

## 9.4 Payment gateways

Once the customer enters appropriate information, you can redirect them to an on-line payment gateway. You can choose from the following options:



- **No on-line payment** - the customer is only displayed with some "Thank you" page.
- **Customer credit** - the customer can pay using their credit. The credit must be entered in the customer details by the store owner. This option is useful for customer loyalty programs. For further information please refer to the [Customer credit](#) chapter.
- **Authorize.NET** - the customer pays using their card. See [Authorize.NET configuration](#) for more details.
- **PayPal** - the customer pays using their card or PayPal account. See [PayPal configuration](#) for more details.
- **Custom payment gateway** - you can integrate some other payment gateway using your code. See [Developing custom payment gateways](#) for more details.

## 9.5 Authorize.NET configuration

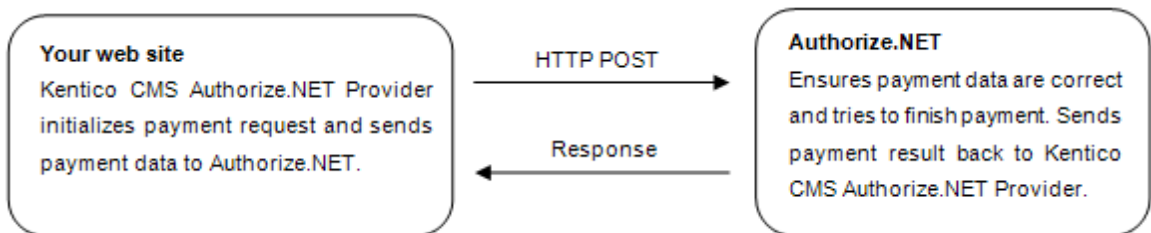
Authorize.NET is one of the most popular gateway providers. It uses plain HTTPS POST operations against its gateway server.

### What do I need?

1. **Kentico CMS 3.0 or higher** with built in Authorize.NET support.
2. **Internet Merchant Account** - A type of bank account that allows a business to accept internet credit card payments (the card is not physically presented to the merchant).
3. **Payment Gateway Account** - A secure internet bridge between your website and the credit card processing networks.

See **Authorize.NET Getting Started Guide** at <http://www.authorize.net/files/ecommerceguide.pdf> for more details.

### How does it work?



### Authorize.NET settings in Kentico CMS 3.1 or higher

Before you can offer customers to use Authorize.NET payment gateway you will need to do some necessary settings:

1. Go to the section *CMSSiteManager -> Settings -> Payment Gateways – Authorize.NET*
2. Enter **API Login** (API Login ID for the payment gateway account) and **Transaction key** (Transaction key obtained from the Merchant Interface) to identify your payments.
3. Turn off **Test Mode** of the Authorize.NET payment gateway. Gateway behavior depends on both test mode settings: in CMSSiteManager and in Authorize.NET Merchant Interface. See table below for more details about test mode settings:

Kentico CMS	Merchant Interface	Transaction processed as
ON	ON	test transaction
OFF	ON	test transaction
ON	OFF	test transaction
OFF	OFF	live transaction

4. Ensure that Authorize.NET payment method is registered and enabled.

5. Check Authorize.NET payment method payment gateway settings:

- Payment gateway url: *https://secure.authorize.net/gateway/transact.dll*
- Payment gateway assembly name: *CMS.EcommerceProvider*
- Payment gateway class name: *CMS.EcommerceProvider.CMSAuthorizeNetProvider*

**Please note:** If you want your transaction to be processed as a test transaction turn on the Test Mode and use *https://test.authorize.net/gateway/transact.dll* as your payment gateway url.

While using the Authorize.NET payment method, a customer is required to fill their credit card information in the last step of the checkout process to finish payment. After the payment is finished the order payment result is updated.

Sign in to [CMS Desk](#). Sign in to [CMS Site Manager](#). The default account is administrator with blank password.

**E-commerce starter site**

Electronics for you

Shopping cart | My account | My wishlist

**Total price: \$537.30**

Home News Products How to buy Company Silvertight Global Administrator Sign out

Step 6 of 6 - Payment

Your order was saved.

**Payment summary**

Order ID: 4  
Payment method: Credit card  
Total price: \$537.30

**Your credit card details**

Credit card number: 1222365948790015  
Credit card CCV: 528  
Credit card expiration: 01/2019

Skip payment Finish payment

Powered by Kentico

## 9.6 PayPal configuration

### What do I need?

1. **Kentico CMS 3.0 or higher** with built in PayPal support.
2. **PayPal account**, see [https://www.paypal.com/us/cgi-bin/webscr?cmd=\\_registration-run](https://www.paypal.com/us/cgi-bin/webscr?cmd=_registration-run) for more details.

### PayPal settings in Kentico CMS 3.1 or higher

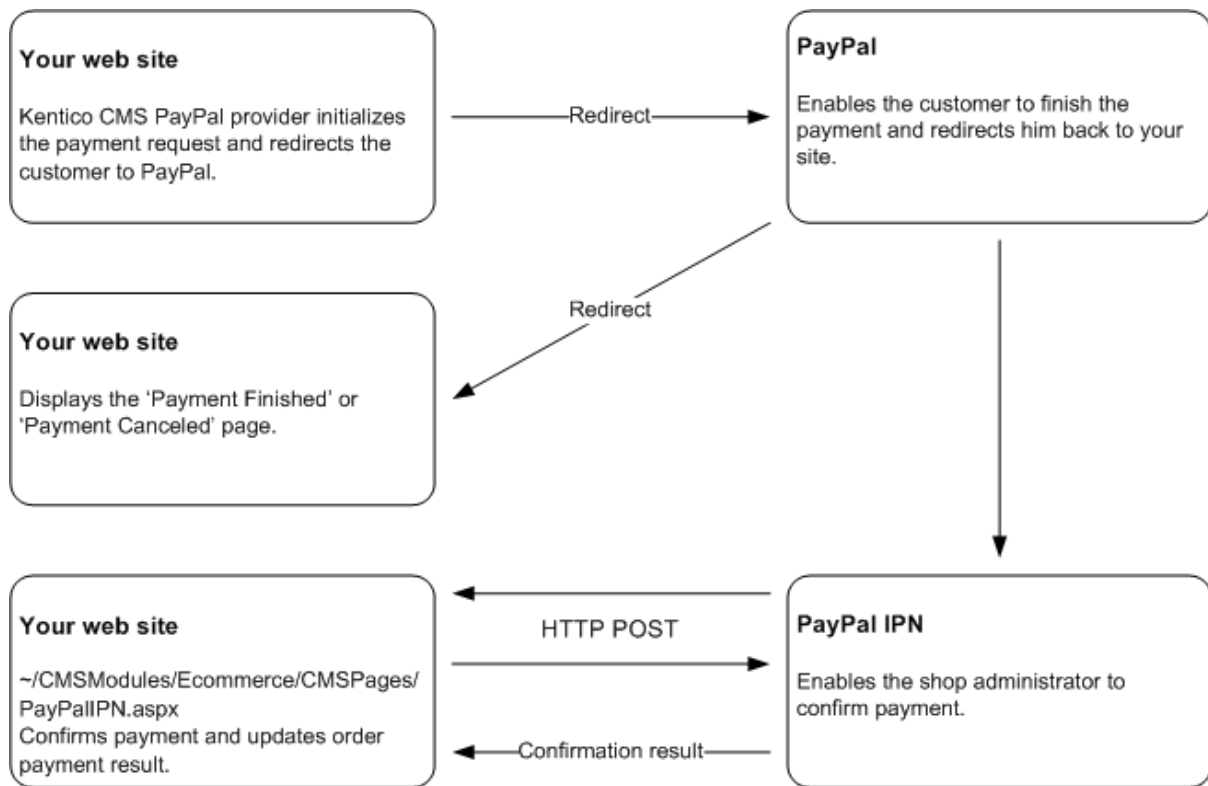
Before you can offer customers to use PayPal payment gateway you will need to do some necessary settings:

1. Go to the section *CMSSiteManager -> Settings -> Payment Gateways – PayPal*
2. Enter **Business** (E-mail address for merchant's PayPal account) to identify your payments. Other settings are optional:
  - **Notify Url**: The URL to which PayPal posts information about the transaction, see *PayPal IPN* for more details. If set overrides settings in PayPal merchant's interface.
  - **Cancel Return Url**: The URL to which the buyer's browser is redirected if payment is cancelled; for example, a URL on your website that displays a "Payment Canceled" page. Browser is redirected to a PayPal web page by default.
  - **Return Url**: The URL to which the buyer's browser is redirected after completing the payment; for example, a URL on your site that displays a "Thank you for your payment" page. Buyer is redirected to a PayPal web page by default.
3. Ensure that PayPal payment method is registered and enabled.
4. Check PayPal payment method payment gateway settings:
  - Payment gateway url: *https://www.paypal.com/cgi-bin/webscr*
  - Payment gateway assembly name: *CMS.EcommerceProvider*
  - Payment gateway class name: *CMS.EcommerceProvider.CMSPayPalProvider*

#### Please note:

- If you want your transaction to be processed as test transaction use *https://www.sandbox.paypal.com/cgi-bin/webscr* as your payment gateway url and sign up for PayPal SandBox testing environment, see [https://www.paypal.com/IntegrationCenter/ic\\_sandbox.html](https://www.paypal.com/IntegrationCenter/ic_sandbox.html) for more details.
- The price is always rounded to two decimal places, otherwise the PayPal payment gateway wouldn't allow the payment.

### How does it work?

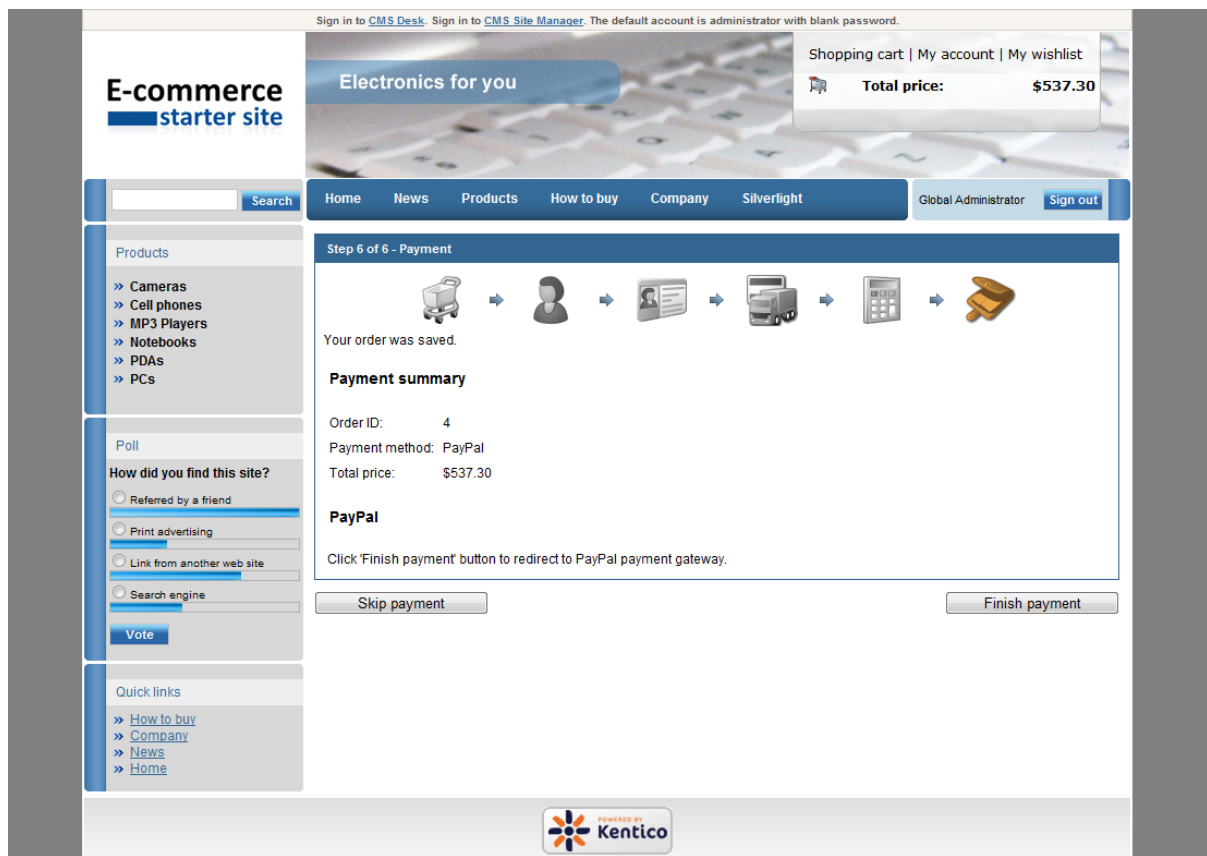


## IPN - Instant Payment Notification

It's easy to spoof the return URL you sent to PayPal since it's visible on the query string. Therefore a user could simply type the Confirmation Url in directly and you should not confirm the order at this point. You can manually check for orders on the PayPal site or wait for PayPal's confirmation e-mails etc. all of which let you know for sure that the order was processed in a 'manual' way.

To automate this process, PayPal can optionally ping you back at another URL with order completion information. It uses a mechanism called Instant Payment Notification (IPN) which is essentially a web based callback mechanism that calls a pre-configured url on your site. IPN must be enabled on the PayPal site and when enabled IPN sends a confirmation to you at this url after the order was processed. PayPal then expects a return from you within a certain timeframe (a few minutes) and return a response to you to confirm you that the customer has paid. To do this you have to POST the data back to PayPal by echoing back all the form data that PayPal sends to you. IPN is optional, but it's a requirement if you need to confirm your orders immediately with your customers.

While using the PayPal payment method, a customer is not required to fill any additional information in the last step of the checkout process. After the "Finish payment" button is clicked the user is redirected to the PayPal payment gateway to finish payment:



## 9.7 Developing custom payment gateways

### Creating a custom payment gateway

Here's a general overview of the process of integrating a custom payment gateway:

1. Create a payment gateway form with your custom controls to enable customers to enter their payment data such as credit card number, see section [Creating custom payment gateway form](#).
2. Create your custom payment gateway class and override required methods for payment processing, see section [Creating custom payment gateway class](#).
3. Go to the **CMSDesk -> Tools -> E-commerce -> Configuration -> Payment methods**.
4. Create a new payment method and register your custom payment gateway as described in the [Payment methods](#) chapter.

### Creating a custom payment gateway form

1. Create a new web user control (\*.ascx) and place it into your site folder which is located in the root of your web project.

Since the control is located in the site folder it is included in export package of your site.

2. Set control class to inherit from abstract class *CMS.EcommerceProvider.CMSPaymentGatewayForm*.
3. There are several methods you need to override to reach your required functionality:
  - **LoadData()** – Initializes form controls with customer payment data.
  - **ValidateData()** – Validates customer payment data.
  - **ProcessData()** – Process customer payment data and saves them to the *ShoppingCartInfo* object.

4. There are several properties to get or set required information:
  - **ShoppingCartControl** – Parent shopping cart control the current shopping cart step belongs to.
  - **ShoppingCartInfoObj** – Shopping cart object which stores all data during the checkout process.

**Please note:** Payment data, such as credit card numbers, credit card codes and others are not saved into the database because of security reasons.

## Creating custom payment gateway class

1. Create a new library (assembly) as a part of your solution and a new class inside this library.
2. Add a reference to *System.Web* assembly into the project with the payment gateway (Right-click the References folder, choose Add reference, select .NET -> System.Web).
3. Set your class to inherit from abstract class *CMS.EcommerceProvider.CMSPaymentGatewayProvider*.
4. There are several methods you can override to reach your required functionality:
  - **AddCustomData()** – Adds payment gateway custom controls to the current shopping cart step. By default *CMSPaymentGatewayForm* control is added to the payment data container and its data are loaded.
  - **RemoveCustomData()** – Removes payment gateway custom controls from the current shopping cart step. By default all controls from payment data container are removed.
  - **ValidateCustomData()** – Validates payment gateway custom data of the current shopping cart step. By default *CMSPaymentGatewayForm* control data are validated.
  - **ProcessCustomData()** – Process payment gateway custom data of the current shopping cart step. By default *CMSPaymentGatewayForm* data are processed.
  - **ProcessPayment()** – Process payment - you need to override this method to process payment by your payment processor.
  - **GetPaymentDataForm()** – Loads payment gateway form with custom controls – you need to override this method to get your own payment gateway form.
5. There are several properties to get or set required information:
  - **ShoppingCartControl** – Parent shopping cart control the current shopping cart step belongs to.
  - **ShoppingCartInfoObj** – Shopping cart object which stores all data during the checkout process .If *OrderId* is set it is created from existing order, otherwise it is returned from current shopping cart control.
  - **OrderId** – Order ID. Set this value when you want to process payment for the existing order outside the checkout process.
  - **PaymentDataContainer** – Payment gateway custom controls container of the current shopping cart step control.
  - **PaymentResult** – Payment result (see caption *Payment Result*)
  - **IsPaymentCompleted** – Indicates whether order payment was already completed. It is determined by order payment result.
  - **InfoMessage** - Payment result message displayed to user when payment succeeds.
  - **ErrorMessage** - Payment result message displayed to user when payment fails.
6. Compile the library.
7. Ensure the library file (\*.dll) is included in <your web project folder>/Bin directory.

## Example

The following example shows a custom payment processor implementation. It allows customers to pay for their orders using some external payment gateway similar to PayPal, let's call it Custom Gateway. Customer is asked for their credit card number in the last step of checkout process. Credit card number is validated for emptiness and processed after the "Finish payment" button is clicked. If it succeeds payment process is performed – required payment data are attached to the payment url and customer is redirected to Custom Gateway. If payment process fails (payment gateway url is not defined) order

payment result is updated and appropriate error message is displayed. Notice that order is saved before the customer is asked to pay for it, it happens immediately after the “Order now” button is clicked.

**Please note:**

- It is not secure to send credit card information as a part of payment gateway url. Customers usually asked for their credit card number after they are redirected to the payment gateway itself; otherwise another way of sending credit card information should be used instead.
- For more details about how payment gateway could inform merchant about the result of the payment which was finished outside their website, see caption PayPal (IPN – Instant Payment Notification).

Example of custom payment gateway definition.

## Custom payment gateway form

It is a simple form with one input field to enter customer credit card number, see image bellow.

## CustomGatewayForm.ascx

Please note: If you installed the Kentico CMS project as a web application, you need to rename the *CodeFile* property on the first line to *Codebehind* for the code example to be functional.

```
<%@ Control Language="C#" AutoEventWireup="true" CodeFile="CustomGatewayForm.ascx.
cs" Inherits="CMSEcommerce_Example_CustomGatewayForm" %>
<asp:Label ID="lblTitle" runat="server" EnableViewState="false" CssClass
="BlockTitle" />
<asp:Label ID="lblError" runat="server" EnableViewState="false" CssClass
="ErrorLabel" Visible="false" />
<asp:Label ID="lblCardNumber" EnableViewState="false" runat="server" />
<asp:TextBox ID="txtCardNumber" runat="server" />
```

## CustomGatewayForm.ascx.cs

[C#]

```
using System;
using System.Data;
using System.Configuration;
using System.Collections;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Web.UI.HtmlControls;

using CMS.EcommerceProvider;
using CMS.GlobalHelper;

public partial class CMSEcommerce_Example_CustomGatewayForm :
CMSPaymentGatewayForm
{

    protected void Page_Load(object sender, EventArgs e)
    {
        // Initialize label
        lblTitle.Text = "Your credit card details";
        lblCardNumber.Text = "Credit card number:";
    }

    /// <summary>
    /// Initializes form controls with customer payment data.
    /// </summary>
    public override void LoadData()
    {
        // Display customer credit card number
        txtCardNumber.Text = ValidationHelper.GetString(this.ShoppingCartInfoObj.
PaymentGatewayCustomData["CustomGatewayCardNumber"], "");
    }

    /// <summary>
    /// Validates customer payment data.
    /// </summary>
    /// <returns></returns>
    public override string ValidateData()
```



```

    {
        if (txtCardNumber.Text.Trim() == "")
        {
            lblError.Visible = true;
            lblError.Text = "Please enter your credit card number";
            return lblError.Text;
        }
        return "";
    }

    /// <summary>
    /// Process customer payment data.
    /// </summary>
    /// <returns></returns>
    public override string ProcessData()
    {
        // Save credit card number
        this.ShoppingCartInfoObj.PaymentGatewayCustomData[
            "CustomGatewayCardNumber"] = txtCardNumber.Text.Trim();
        return "";
    }
}

```

## Custom payment gateway class

The following example uses assembly name *CMS.CustomProvider* and class *CMS.CustomProvider.CustomGateway*, however, you will need to use your own names.

### CustomGateway.cs

[C#]

```

using System;
using System.Collections.Generic;
using System.Text;
using System.Web;
using System.Collections;

using CMS.EcommerceProvider;
using CMS.GlobalHelper;
using CMS.UIControls;
using CMS.ExtendedControls;

namespace CMS.CustomProvider
{
    /// <summary>
    /// Class representing Custom Gateway processor.
    /// </summary>
    public class CustomGateway : CMSPaymentGatewayProvider
    {
        /// <summary>
        /// Returns payment gateway form with custom controls.
        /// </summary>
        /// <returns></returns>
    }
}

```

```

public override CMSPaymentGatewayForm GetPaymentDataForm()
{
    try
    {
        return (CMSPaymentGatewayForm)this.ShoppingCartControl.LoadControl
        ("~/CMSEcommerce/Example/CustomGatewayForm.ascx");
    }
    catch
    {
        return null;
    }
}

/// <summary>
/// Process payment.
/// </summary>
public override void ProcessPayment()
{
    // Get payment gateway url
    string url = this.GetPaymentGatewayUrl();

    if (url != "")
    {
        // Initialize payment parameters
        Hashtable parameters = InitializePaymentParameters();

        // Add required payment data to the url
        url = GetFullPaymentGatewayUrl(url, parameters);

        // Redirect to payment gateway to finish payment
        this.ShoppingCartControl.Page.Response.Redirect(url);
    }
    else
    {
        // Show error message - payment gateway url not found
        this.ErrorMessage = "Unable to finish payment: Payment gateway url
not found.";

        // Update payment result
        this.PaymentResult.PaymentDescription = this.ErrorMessage;
        this.PaymentResult.PaymentIsCompleted = false;

        // Update order payment result in database
        this.UpdateOrderPaymentResult();
    }
}

/// <summary>
/// Returns table with initialized payment parameters.
/// </summary>
/// <returns></returns>
private Hashtable InitializePaymentParameters()
{
    Hashtable parameters = new Hashtable();

```

```

        parameters["orderid"] = this.ShoppingCartInfoObj.OrderId;
        parameters["price"] = this.ShoppingCartInfoObj.TotalPrice;
        parameters["currency"] = this.ShoppingCartInfoObj.CurrencyInfoObj.
CurrencyCode;
        parameters["cardnumber"] = Convert.ToString(this.ShoppingCartInfoObj.
PaymentGatewayCustomData["CustomGatewayCardNumber"]);

        return parameters;
    }

    /// <summary>
    /// Returns payment gateway url with payment data in query string.
    /// </summary>
    /// <param name="url">Payment gateway url.</param>
    /// <param name="parameters">Initialized payment paremeters.</param>
    /// <returns></returns>
    private string GetFullPaymentGatewayUrl(string url, Hashtable parameters)
    {
        foreach (DictionaryEntry parameter in parameters)
        {
            // Add payment data to the url
            url = URLHelper.AddParameterToUrl(url, Convert.ToString(parameter.
Key), HttpUtility.UrlEncode(Convert.ToString(parameter.Value)));
        }
        return url;
    }
}

```

## 9.8 Payment results

### Payment Results

Payment result is stored in xml format which is represented by the object *CMS.Ecommerce.PaymentResultInfo*. Each payment result xml node is equal to the single payment result item which is represented by the object *CMS.Ecommerce.PaymentResultItemInfo*.

Base payment result items are:

- **Payment date** – Date and time when the payment result was last updated.
- **Payment method** – Payment method which was used for payment.
- **Payment is completed** – Yes/No. Indicates whether payment was already completed.
- **Payment status** – Completed/Failed/(your custom status). Status of the payment.
- **Payment transaction ID** – Completed payment unique identifier generated by the payment gateway.
- **Payment description** – Describes payment result in more details.

Payment result item properties are:

- **Name**: Unique identifier of the item.
- **Header**: Friendly name of the item visible to user (simple text or localizable string).
- **Text**: Outer representation of the item value visible to user (simple text or localizable string).
- **Value**: Inner representation of the item value used by developers.

Following example shows xml definition of the order payment result extended by the item “

*authorizationcode*” used by the Authorize.NET:

```
<result>
  <item name="date" header="{ $PaymentGateway.Result.Date$}" value="1/27/2008
5:01:41 PM" />
  <item name="method" header="{ $PaymentGateway.Result.PaymentMethod$}" text="
Credit card" value="230" />
  <item name="completed" header="{ $PaymentGateway.Result.IsCompleted$}" value
="1" text="{ $PaymentGateway.Result.PaymentCompleted$}" />
  <item name="status" header="{ $PaymentGateway.Result.Status$}" text="
{ $PaymentGateway.Result.Status.Completed$}" value="completed" />
  <item name="transactionid" header="{ $PaymentGateway.Result.TransactionID$}"
value="0" />
  <item name="description" header="{ $PaymentGateway.Result.Description$}" />
  <item name="authorizationcode" header="{ $AuthorizeNet.AuthorizationCode$}"
value="000000" />
</result>
```

The following example shows order payment result which is visible to the user in CMSDesk:

*Date: 1/27/2008 5:01:41 PM*  
*Method: Credit card*  
*Is completed: YES*  
*Status: Completed*  
*Transaction ID: 0*  
*Authorization code: 000000*

**Please note:**

- The order payment result remains empty until it is updated by the payment gateway processor.
- You don't need to specify both item value and item text if they are identical because payment result rendering method can manage this and renders payment result as follows: try to render item text, if not found, try to render item value.

## How to customize payment result

You can use *PaymentResultInfo* properties to get or set specified item text or value:

- *PaymentDate*
- *PaymentMethodID*
- *PaymentMethodName*
- *PaymentIsCompleted*
- *PaymentStatusName*
- *PaymentStatusCode*
- *PaymentTransactionID*

You will need to use public methods ***GetPaymentResultItemInfo(string itemName)*** and ***SetPaymentResultItemInfo(PaymentResultItemInfo itemObj)*** to get and set your custom payment result items.

The following example shows how to get and set custom payment result item while payment processing by your custom payment gateway provider:

**[C#]**

```
using CMS.Ecommerce;

// Set authorization code
PaymentResultItemInfo item = new PaymentResultItemInfo();
item.Header = "{$AuthorizeNet.AuthorizationCode$}";
item.Name = "authorizationcode";
item.Value = "00000";
this.PaymentResult.SetPaymentResultItemInfo(item);
```

[C#]

```
using CMS.Ecommerce;

// Get authorization code
PaymentResultItemInfo item = this.PaymentResult.GetPaymentResultItemInfo(
    "authorizationcode");
```

## 9.9 Customer credit

Customers may receive a credit on their account. They can then purchase products using this credit. This feature is useful for customer loyalty competitions where customers receive bonus points/credit for their previous purchases and can order some products once they achieve an appropriate amount of credit.

Please note: the purchase can be made only using one payment method, so it's not possible to combine credit payment with another form of payment and the whole order must be paid using the credit.

The customer receives credit when the store owner adds some credit event to the customer history. This can be done in **Customer properties** dialog, on the **Credit** tab.

The credit is added to/deducted from customers' site-specific or global credit balance. This depends on the site separation settings of a given site. Specifically, it depends on whether the customers use site-specific or global credit on the site. More details can be found in the [Enabling the use of global objects](#) topic in the Site separation section of this guide.

Before you can offer registered customers to use the **Customer credit** you will need to do some necessary settings:

1. Ensure that the **Customer credit** payment method is registered and enabled.
2. Check the **Customer credit** payment method settings:
  - Payment gateway url: leave blank
  - Payment gateway assembly name: *CMS.EcommerceProvider*
  - Payment gateway class name: *CMS.EcommerceProvider.CMSCreditPaymentProvider*

While using the customer credit, a customer is not required to fill any additional information in the last step of the checkout process. After the "Finish payment" button is clicked, their credit is decreased by specified amount which is equal to the order total price in default currency and the order payment result is updated.

**Step 6 of 6 - Payment**

 →  →  →  →  → 

Your order was saved.

**Payment summary**

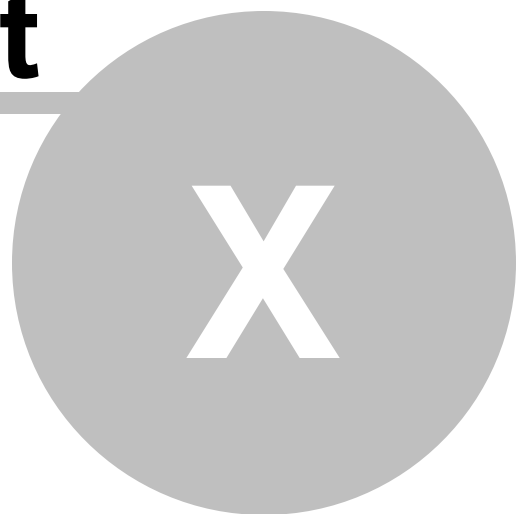
Order ID: 6  
Payment method: Customer credit  
Total price: \$155.25

**Use credit for payment**

Available credit: \$1000.00

# Part

---



**Security**

---

## 10 Security

### 10.1 Overview

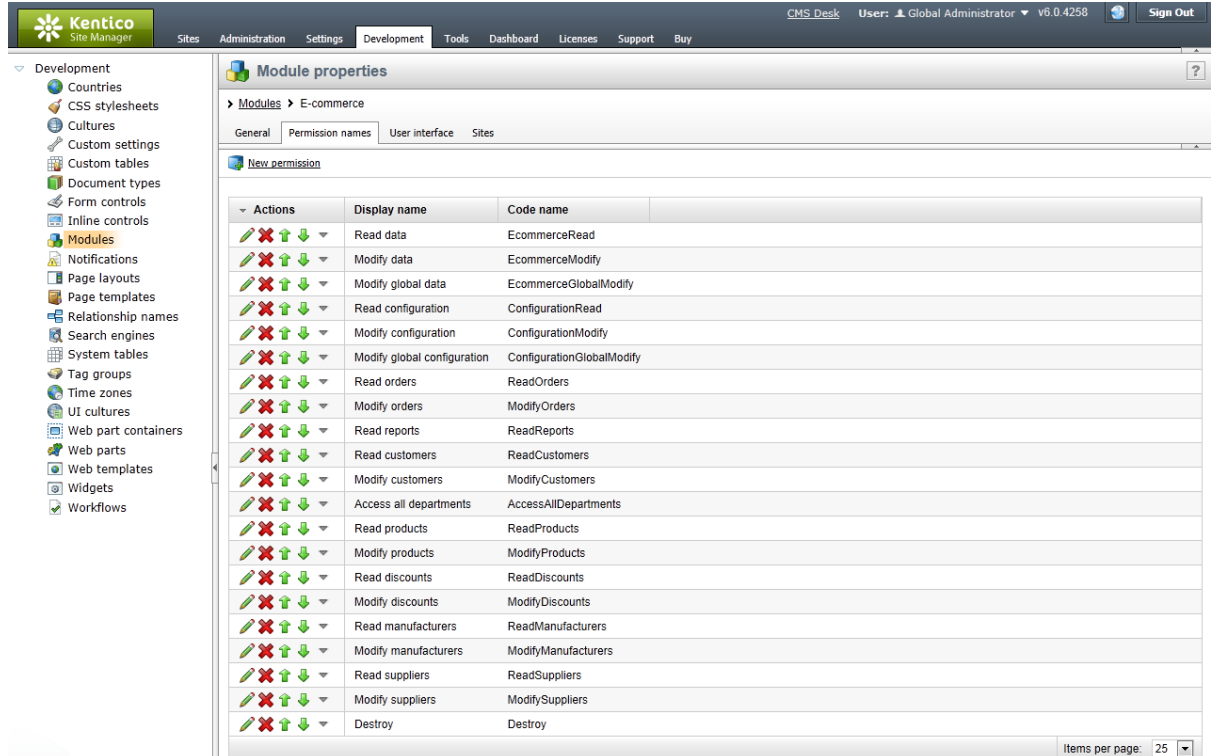
This chapter describes security of the E-commerce module:

- To learn what security settings are available in the E-commerce module to define access and configuration rights for users of your website, please refer to the [Permissions](#) topic.
- To learn what needs to be set for a user to be able to manage products from a specific department, please refer to the [Department administrators](#) topic.
- To learn how to personalize your user interface of the E-commerce module, please refer to the [UI personalization](#) topic.
- To see UI personalization and permissions sample settings for this module, please refer to the [E-commerce sample roles](#) topic.

### 10.2 Permissions

To prevent users from accessing and modifying certain on-line store data and from configuring the on-line store, you will need to assign E-commerce module permissions. This procedure is described in more detail in the [Membership, permissions and security -> Permissions](#) chapter in the Development section of the Developer's Guide.

The E-commerce module has the following permissions:



The screenshot displays the 'Module properties' window for the 'E-commerce' module, specifically the 'Permission names' tab. A 'New permission' button is visible. Below it is a table listing various permissions with their actions, display names, and code names.

Actions	Display name	Code name
	Read data	EcommerceRead
	Modify data	EcommerceModify
	Modify global data	EcommerceGlobalModify
	Read configuration	ConfigurationRead
	Modify configuration	ConfigurationModify
	Modify global configuration	ConfigurationGlobalModify
	Read orders	ReadOrders
	Modify orders	ModifyOrders
	Read reports	ReadReports
	Read customers	ReadCustomers
	Modify customers	ModifyCustomers
	Access all departments	AccessAllDepartments
	Read products	ReadProducts
	Modify products	ModifyProducts
	Read discounts	ReadDiscounts
	Modify discounts	ModifyDiscounts
	Read manufacturers	ReadManufacturers
	Modify manufacturers	ModifyManufacturers
	Read suppliers	ReadSuppliers
	Modify suppliers	ModifySuppliers
	Destroy	Destroy

Items per page: 25

- **Read data** - allows to access Orders, Reports, Customers, Products, Product options, Discount coupons, Discount levels, Manufacturers and Suppliers.



- **Modify data** - allows to create, modify and delete data; see Read data.
- **Modify global data** - allows to create, modify and delete global Customers, global Products, global Product options, global Discount coupons, global Discount levels, global Manufacturers and global Suppliers.
- **Read configuration** - allows to access E-commerce module configuration (Store settings, Departments, Shipping options, Payment methods, Tax classes, Currencies, Exchange rates, Order status, Public status, Internal status, Invoice).
- **Modify configuration** - allows to modify E-commerce module configuration; see Read configuration.
- **Modify global configuration** - allows to modify E-commerce module global configuration.
- **Read orders** - allows to access Orders.
- **Modify orders** - allows to create, modify and delete Orders.
- **Read reports** - allows to access Reports.
- **Read customers** - allows to access Customers.
- **Modify customers** - allows to create, modify and delete Customers.
- **Access all departments** - allows to access Products from all departments.
- **Read products** - allows to access Products and Product options.
- **Modify products** - allows to create, modify and delete Products and Product options.
- **Read discounts** - allows to access Discount coupons.
- **Modify discounts** - allows to create, modify and delete Discount coupons.
- **Read manufacturers** - allows to access Manufacturers.
- **Modify manufacturers** - allows to create, modify and delete Manufacturers.
- **Read suppliers** - allows to access Suppliers.
- **Modify suppliers** - allows to create, modify and delete Suppliers.
- **Destroy** - allows to destroy E-commerce module object version history.

## Example

To allow members of a particular role to edit e.g. a site-bound Manufacturer, you will need to assign this role permissions in one of the following combinations:

- **Read data + Modify data**
- **Read data + Modify manufacturers**
- **Read manufacturers + Modify data**
- **Read manufacturers + Modify manufacturers**

Similarly, to edit e.g. a global Customer, you will need to assign permissions in one of the following combinations:

- **Read data + Modify global data**
- **Read manufacturers + Modify global data**

This can be configured in the E-commerce module permissions matrix in **Site Manager -> Administration -> Permissions**, as described in detail in the [Permissions](#) chapter in the Development -> Membership, permissions and security section of the Developer's Guide.

## Other permissions

You may also need to restrict access to modifications of product documents in the content tree. In such case, you can use document-level permissions as described in detail in the [Permissions -> Document permissions](#) chapter in the same section of the Developer's Guide.

## 10.3 Department administrators

Here you will learn how to assign certain users, i. e. your department administrators, to departments containing products for which they are responsible. This will prevent them from accidentally modifying other products.

To allow a user to manage products in given departments, you need to:

1. Assign the role whose member the user is appropriate E-commerce module permissions. Specifically, you need to assign the Read and Modify permissions for products or data. The user will thus be able to access and manage products. If you haven't assigned these permissions yet, please refer to the [Permissions](#) topic for more details.
2. Assign the user to [departments](#) whose products they should manage OR assign the role whose member the user is the **Access all departments** permission; see note below.



### Please note

If you need to allow certain users to access Products from all departments regardless of the departments to which they are assigned, please assign the **Access all departments** permission to roles whose members they are.

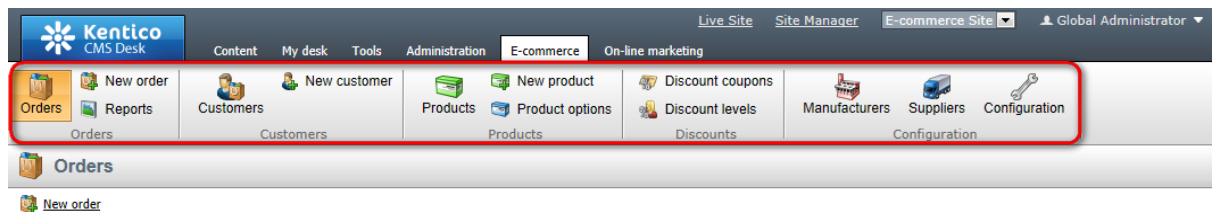
## 10.4 UI personalization

UI personalization enables you to provide certain users with **customized user interface**. You can hide any E-commerce module UI item represented by the corresponding UI element and you can also create your custom UI elements. To learn how to do this, please refer to the [Membership, permissions and security -> UI personalization](#) chapter in the Development section of the Developer's Guide.

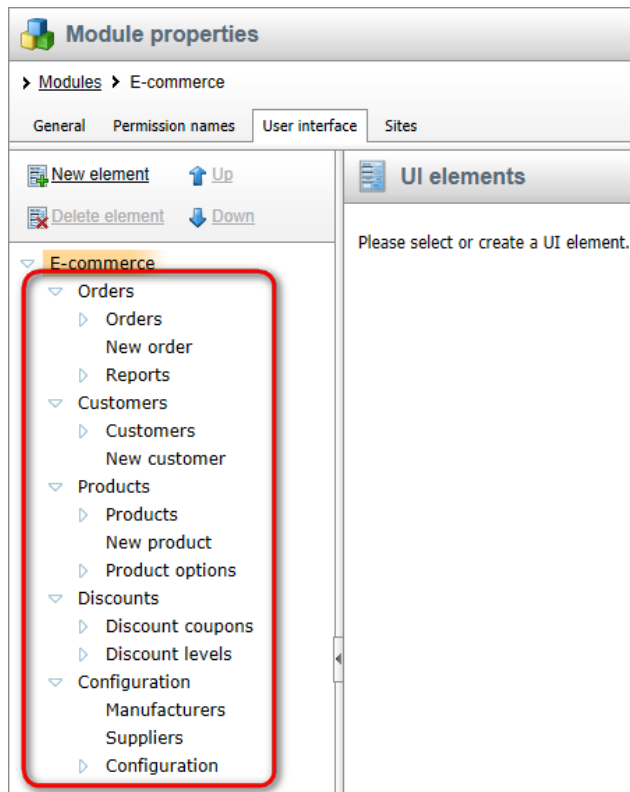
Here follows a survey of E-commerce module UI elements and the real UI which these elements represent.

### E-commerce toolbar

The **E-commerce** module provides ample possibilities of UI personalization. As you can see in the screenshot below, it is divided in five categories: [Orders](#), [Customers](#), [Products](#), [Discounts](#) and [Configuration](#).



These categories together with items they contain are represented by the first- and second-level E-commerce module UI elements as highlighted below.



#### Please note

It is not recommended to move any UI elements that come with the installation. The reason is to ensure the possibility of a correct security check in the real UI.



#### Please note

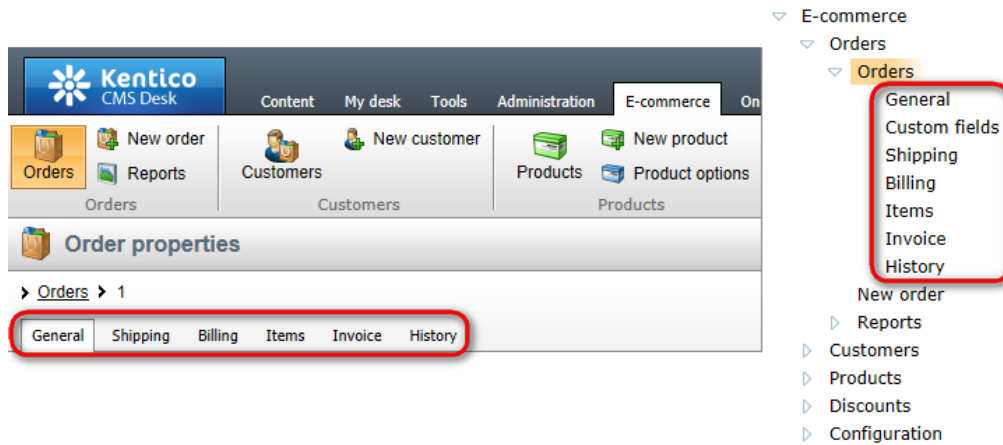
You can adjust the size of the icon in the ribbon. This can be done in **Site Manager -> Development -> Modules -> Edit (🔧) E-commerce** on the **User interface -> General** tab of the corresponding UI element. More details can be found in the [UI personalization -> UI elements management](#) chapter in the same section of the Developer's Guide.

## Orders

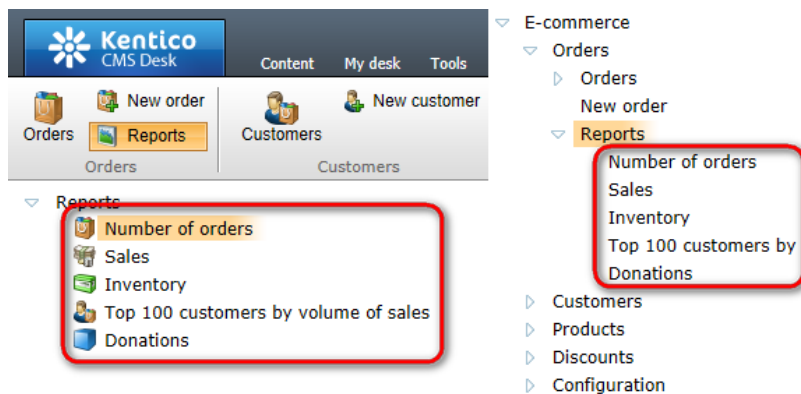
The Orders category provides information about orders.

These tabs contain information about your order. Please note that the **Custom fields** tab is visible only if

you added your custom fields via **Site Manager -> Development -> System tables**.



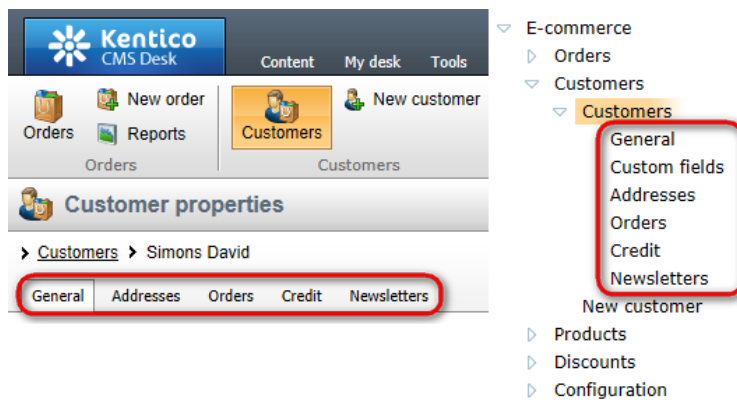
The Reports tree items provide reporting related to your on-line store.



## Customers

The Customers category provides information about customers.

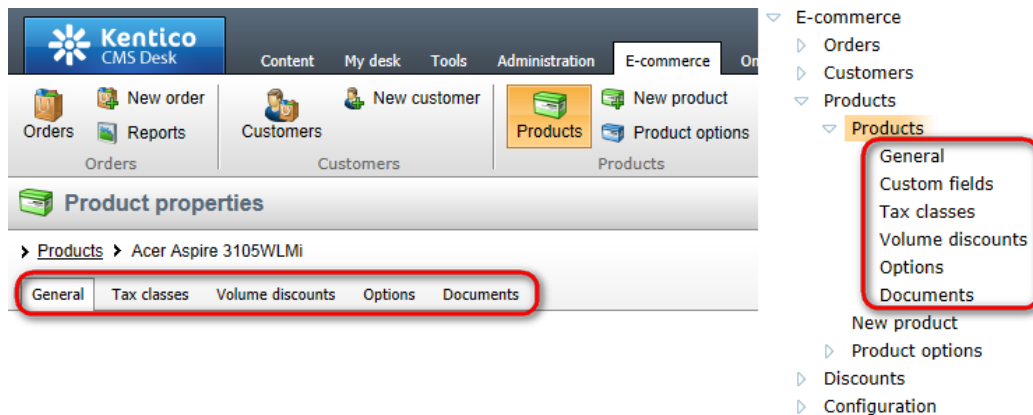
These tabs contain information about your customer. Please note that the **Custom fields** tab is visible only if you added your custom fields via **Site Manager -> Development -> System tables**.



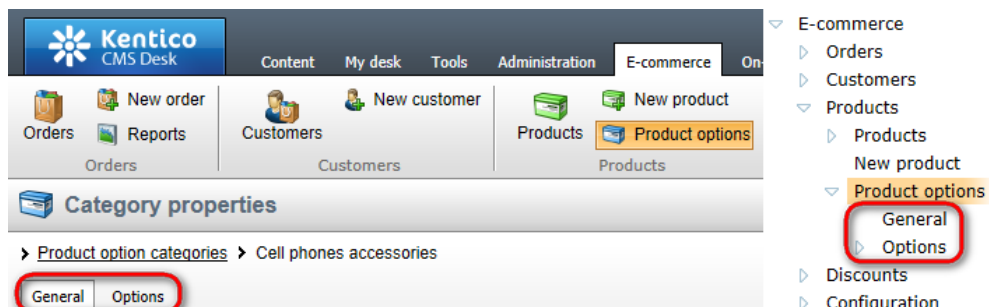
## Products

The Products category provides information about products.

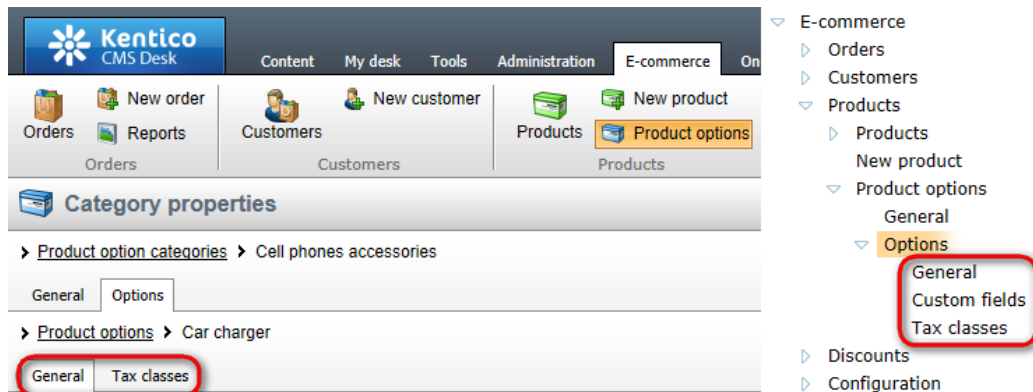
These tabs contain information about your product. Please note that the **Custom fields** tab is visible only if you added your custom fields via **Site Manager -> Development -> System tables**.



These tabs contain information about your product option.



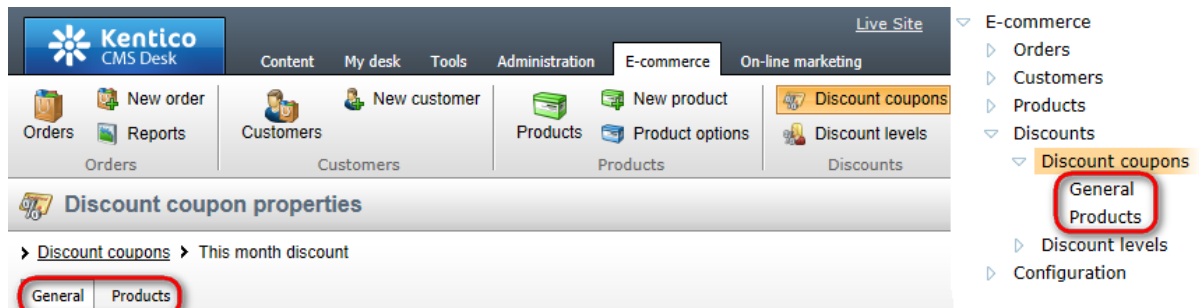
These tabs contain your product option details. Please note that the **Custom fields** tab is visible only if you added your custom fields via **Site Manager -> Development -> System tables**.



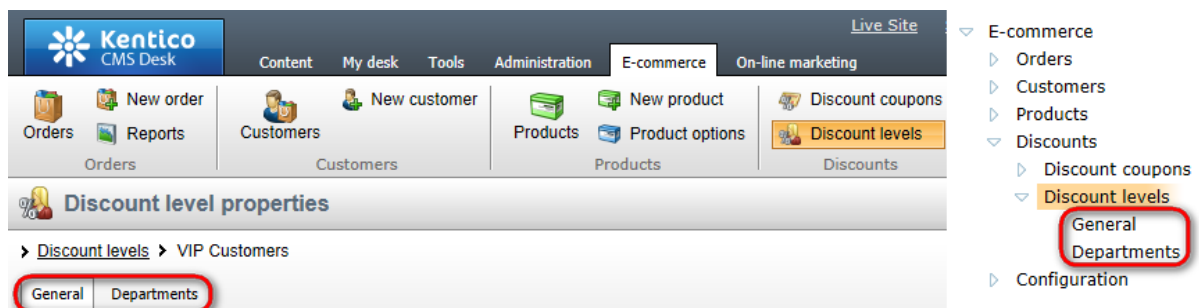
## Discounts

The Discounts category provides information about discounts.

These tabs contain information about your discount coupon.

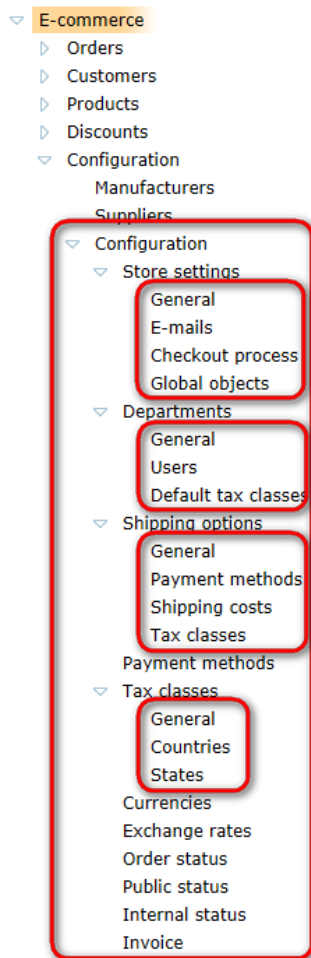


These tabs contain information about your discount level.

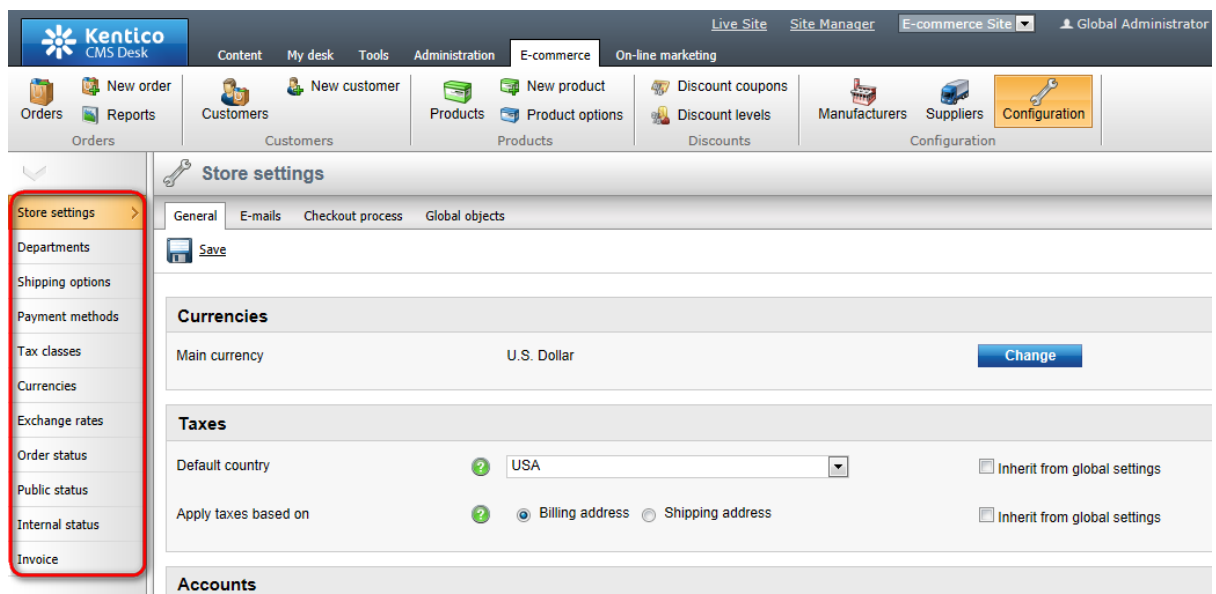


## Configuration

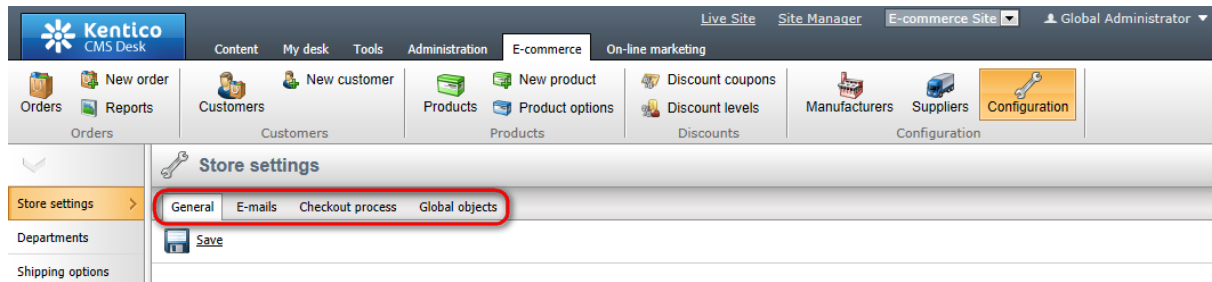
The Configuration category provides access to configuration-related features of the E-commerce module.



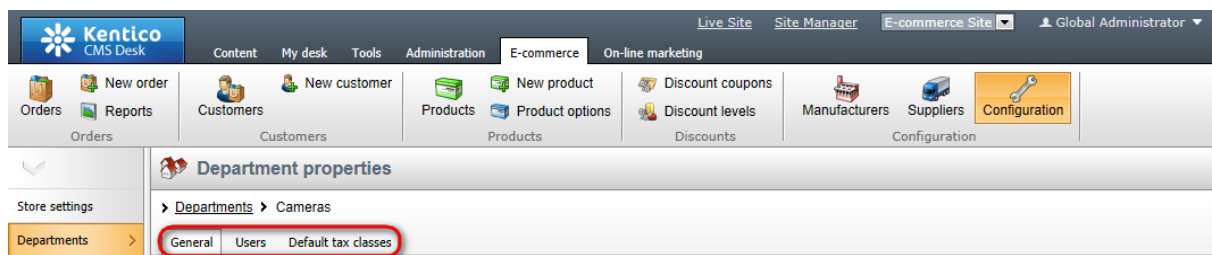
The Configuration menu items enable you to configure your E-commerce module features.



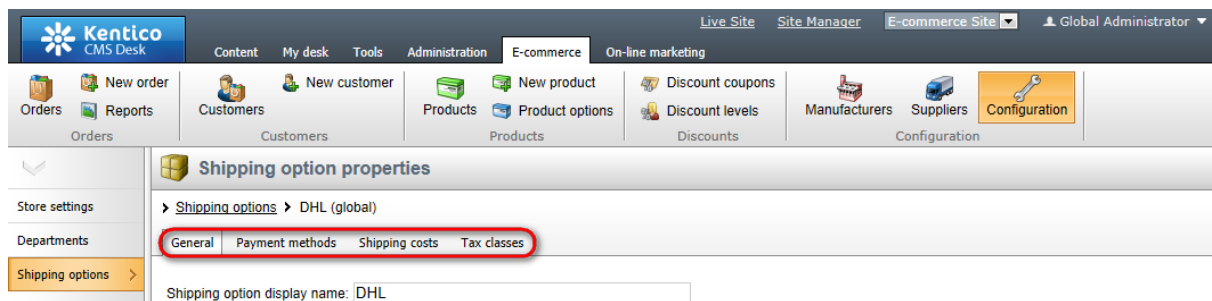
These tabs contain information about your on-line store settings.



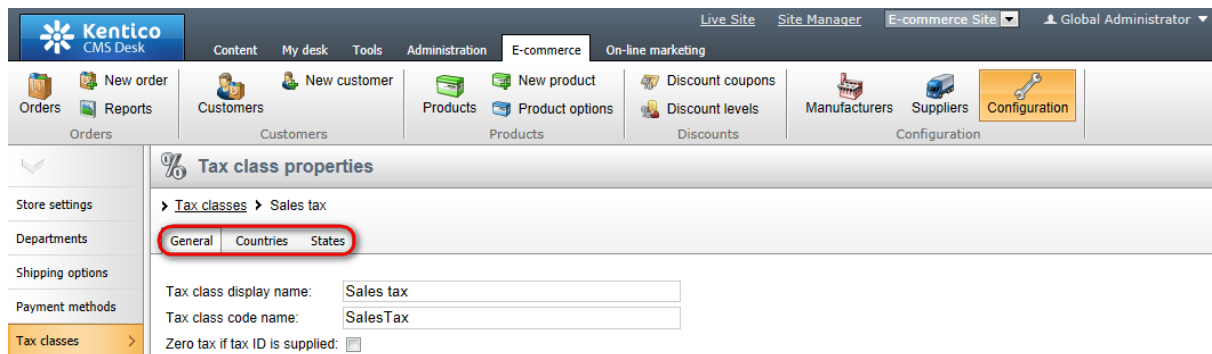
These tabs contain information about your department.



These tabs contain information about your shipping option.



These tabs contain information about your tax class.





## 10.5 E-commerce sample roles

After installing your E-commerce sample site, you can see [UI personalization](#) and [Permissions](#) sample settings for the E-commerce module. For this purpose, the following sample roles are pre-configured on this sample site: [CMS E-commerce Account Managers](#), [CMS E-commerce Administrators](#) and [CMS E-commerce Editors](#). However, these roles are not available on other Kentico CMS sample sites.

These roles can be edited in **Site Manager -> Administration -> Roles**. More details can be found in the [Membership, permissions and security -> Role management](#) topic in the Development section of the Developer's Guide.

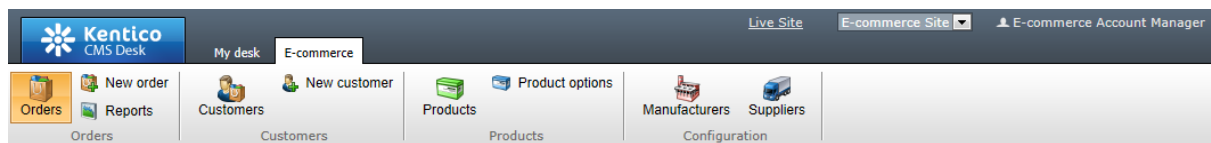


### Please note


These roles serve only as inspiration for security settings of your on-line store administration. You can create and configure your own roles for your on-line store.

### CMS E-commerce Account Managers

Members of this role can access the following features, based on the role UI profile settings as described in detail in the [UI personalization](#) topic:



Based on the role [permissions](#) settings, members of this role have full access to Orders, Customers, Manufacturers and Suppliers and they can also view Reports. However, Products and Product options they can only read:

 **Role properties**

» [Roles](#) » CMS E-commerce Account Managers

General Users Memberships **Permissions** UI personalization

Permissions for: Module  E-commerce

Permission	Allow
Read data	<input type="checkbox"/>
Modify data	<input type="checkbox"/>
Modify global data	<input type="checkbox"/>
Read configuration	<input type="checkbox"/>
Modify configuration	<input type="checkbox"/>
Modify global configuration	<input type="checkbox"/>
Read orders	<input checked="" type="checkbox"/>
Modify orders	<input checked="" type="checkbox"/>
Read reports	<input checked="" type="checkbox"/>
Read customers	<input checked="" type="checkbox"/>
Modify customers	<input checked="" type="checkbox"/>
Access all departments	<input type="checkbox"/>
Read products	<input checked="" type="checkbox"/>
Modify products	<input type="checkbox"/>
Read discounts	<input type="checkbox"/>
Modify discounts	<input type="checkbox"/>
Read manufacturers	<input checked="" type="checkbox"/>
Modify manufacturers	<input checked="" type="checkbox"/>
Read suppliers	<input checked="" type="checkbox"/>
Modify suppliers	<input checked="" type="checkbox"/>

« < 1 2 > »

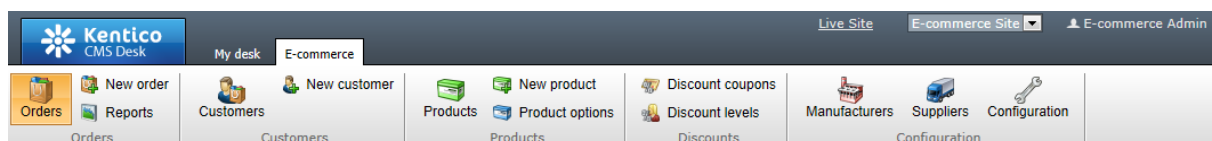


### Please note


To try this role out, please log in as the role default user, i.e. *E-commerce Account Manager*, with *ECAccountManager* user name and blank password.

## CMS E-commerce Administrators

Members of this role can access the following features, based on the role UI profile settings as described in detail in the [UI personalization](#) topic:



Based on the role [permissions](#) settings, members of this role have full access within the E-commerce module. However, they cannot modify global data or global configuration:

 **Role properties**

> [Roles](#) > CMS E-commerce Administrators

General Users Memberships **Permissions** UI personalization

Permissions for: Module  E-commerce

Permission	Allow
Read data	<input checked="" type="checkbox"/>
Modify data	<input checked="" type="checkbox"/>
Modify global data	<input type="checkbox"/>
Read configuration	<input checked="" type="checkbox"/>
Modify configuration	<input checked="" type="checkbox"/>
Modify global configuration	<input type="checkbox"/>
Read orders	<input checked="" type="checkbox"/>
Modify orders	<input checked="" type="checkbox"/>
Read reports	<input checked="" type="checkbox"/>
Read customers	<input checked="" type="checkbox"/>
Modify customers	<input checked="" type="checkbox"/>
Access all departments	<input checked="" type="checkbox"/>
Read products	<input checked="" type="checkbox"/>
Modify products	<input checked="" type="checkbox"/>
Read discounts	<input checked="" type="checkbox"/>
Modify discounts	<input checked="" type="checkbox"/>
Read manufacturers	<input checked="" type="checkbox"/>
Modify manufacturers	<input checked="" type="checkbox"/>
Read suppliers	<input checked="" type="checkbox"/>
Modify suppliers	<input checked="" type="checkbox"/>

« < 1 2 > »



#### Please note

To try this role out, please log in as the role default user, i.e. *E-commerce Admin*, with *ECAdmin* user name and blank password.

## CMS E-commerce Editors

Members of this role can access the following features in **CMS Desk -> E-commerce**, based on the role UI profile settings as described in detail in the [UI personalization](#) topic:



Based on the role [permissions](#) settings, members of the *CMS E-commerce Editors* role have full access to Products, Product options, Discount coupons and Discount levels. However, Manufacturers and Suppliers they can only read:

**Role properties**

► Roles ► CMS E-commerce Editors

General Users Memberships **Permissions** UI personalization

Permissions for: Module ▼ E-commerce ▼

Permission	Allow
Read data	<input type="checkbox"/>
Modify data	<input type="checkbox"/>
Modify global data	<input type="checkbox"/>
Read configuration	<input type="checkbox"/>
Modify configuration	<input type="checkbox"/>
Modify global configuration	<input type="checkbox"/>
Read orders	<input type="checkbox"/>
Modify orders	<input type="checkbox"/>
Read reports	<input type="checkbox"/>
Read customers	<input type="checkbox"/>
Modify customers	<input type="checkbox"/>
Access all departments	<input type="checkbox"/>
Read products	<input checked="" type="checkbox"/>
Modify products	<input checked="" type="checkbox"/>
Read discounts	<input checked="" type="checkbox"/>
Modify discounts	<input checked="" type="checkbox"/>
Read manufacturers	<input checked="" type="checkbox"/>
Modify manufacturers	<input type="checkbox"/>
Read suppliers	<input checked="" type="checkbox"/>
Modify suppliers	<input type="checkbox"/>

« < 1 2 > »

### Please note

To try this role out, please log in as the role default user, i.e. *E-commerce Editor*, with *ECEditor* user name and blank password.

# Part

---



XI

**Developing custom providers**

---

## 11 Developing custom providers

### 11.1 Overview

Customization of the behavior of the CMS application or its specific module according to your exact requirements is described in detail in the [Custom providers](#) chapter in the API programming and Kentico CMS internals section of the Developer's Guide.

As this type of customization is available also for the E-commerce module, you can override the default behavior and calculations by using custom providers that ensure various operations.

E-commerce module customization samples can be found in **<web project folder>\App\_Code\CMSModules\Ecommerce\Samples** (or in **<web project folder>\Old\_App\_Code\CMSModules\Ecommerce\Samples** if you installed Kentico CMS as a web application).

# Part

---



XII

**E-commerce internals and API**

---

## 12 E-commerce internals and API

### 12.1 Overview

In this chapter, you will learn which [database tables](#) and [API classes](#) are used in the E-commerce module. You will also see the most common [API examples](#).

Advanced API examples can be found in the [API programming and Kentico CMS internals](#) section of the Developer's Guide. For detailed API documentation, such as a list of all methods from the module classes, please refer to [Kentico CMS API Reference](#).

### 12.2 Database tables

The following database tables are used in the E-commerce module. As this is a very complex module, the topic is divided into the following content-based sections:

- [Customers](#)
- [Products](#)
- [Orders](#)
- [Shopping cart](#)
- [Currencies](#)
- [Shipping and payment](#)
- [Discounts](#)
- [Taxes](#)

#### Customers

- **COM\_Customer** - contains records representing customers.
- **CMS\_User** - contains records representing registered users.
- **CMS\_UserSite** - contains records representing relationship between users and sites; a given user is assigned to a given site.
- **COM\_Address** - contains records representing customer addresses.
- **COM\_CustomerCreditHistory** - contains records representing customer credit history.

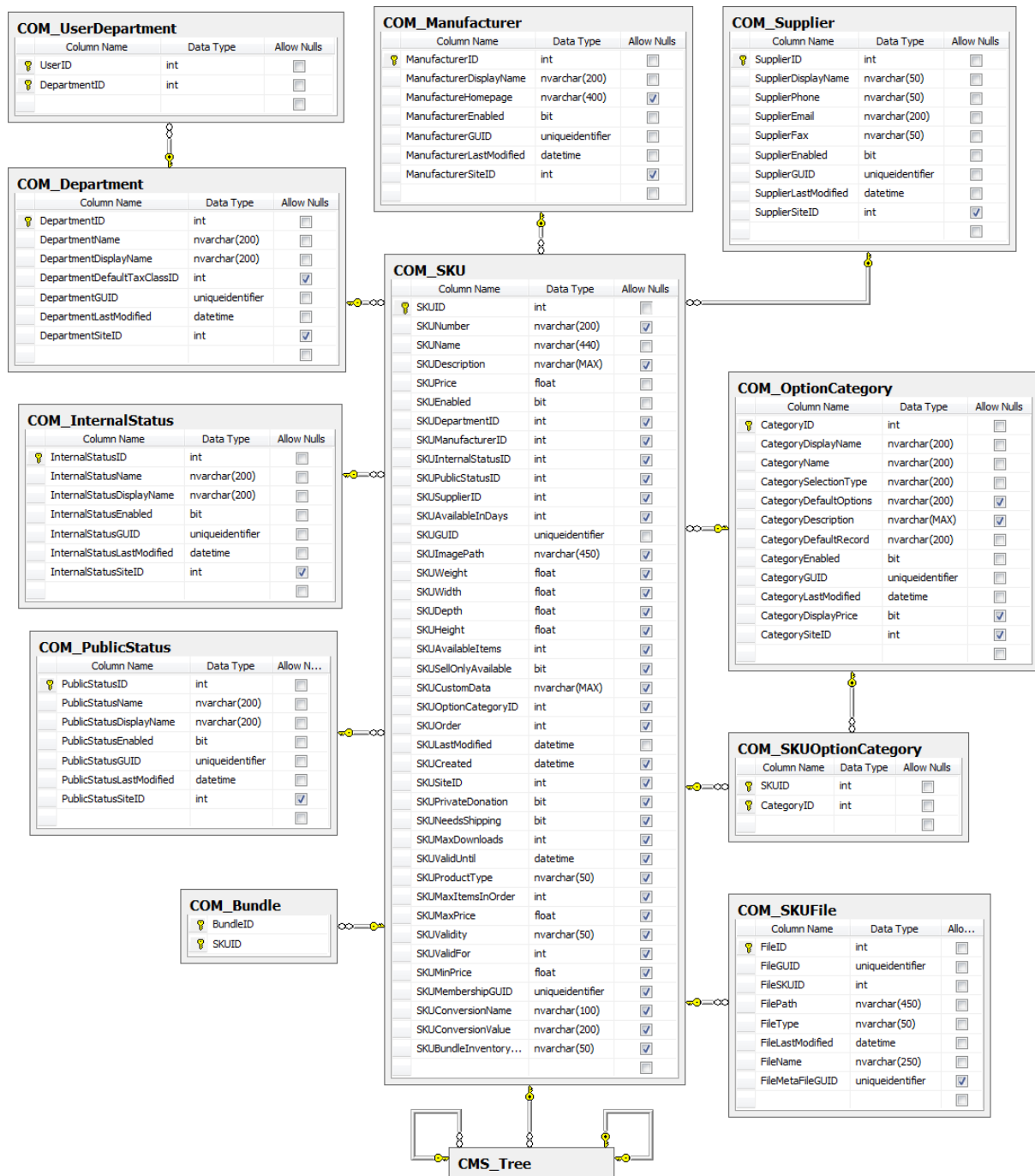




- © 2011 Kentico Software

type and other products, indicating that these products are within these bundles.

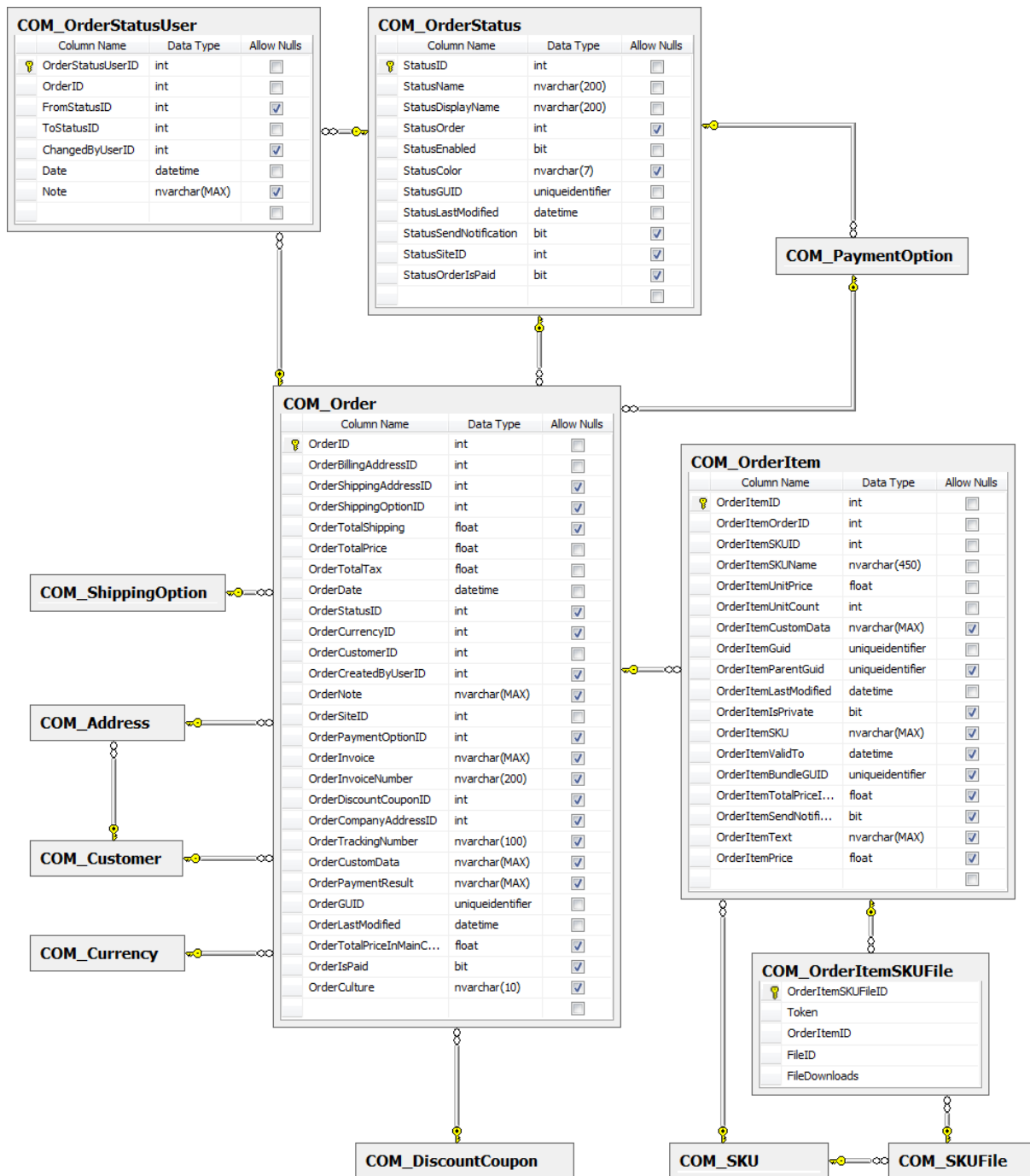
- **COM\_SKUFile** - contains records representing e-product files.
- **COM\_Manufacturer** - contains records representing manufacturers.
- **COM\_Supplier** - contains records representing suppliers.
- **COM\_Department** - contains records representing departments.
- **COM\_UserDepartment** - contains records representing relationships between users and departments, indicating that a particular user can manage products of a particular department.
- **COM\_InternalStatus** - contains records representing product internal statuses.
- **COM\_PublicStatus** - contains records representing product public statuses.



## Orders

- **COM\_Order** - contains records representing orders.
- **COM\_OrderItem** - contains records representing order items.
- **COM\_OrderItemSKUFile** - contains records representing relationships between order items and e-product files, indicating that a particular order item comes with a particular e-product file. This relationship is created when an order containing an e-product is paid.
- **COM\_OrderStatus** - contains records representing order statuses.
- **COM\_OrderStatusUser** - contains records representing relationships between order statuses and

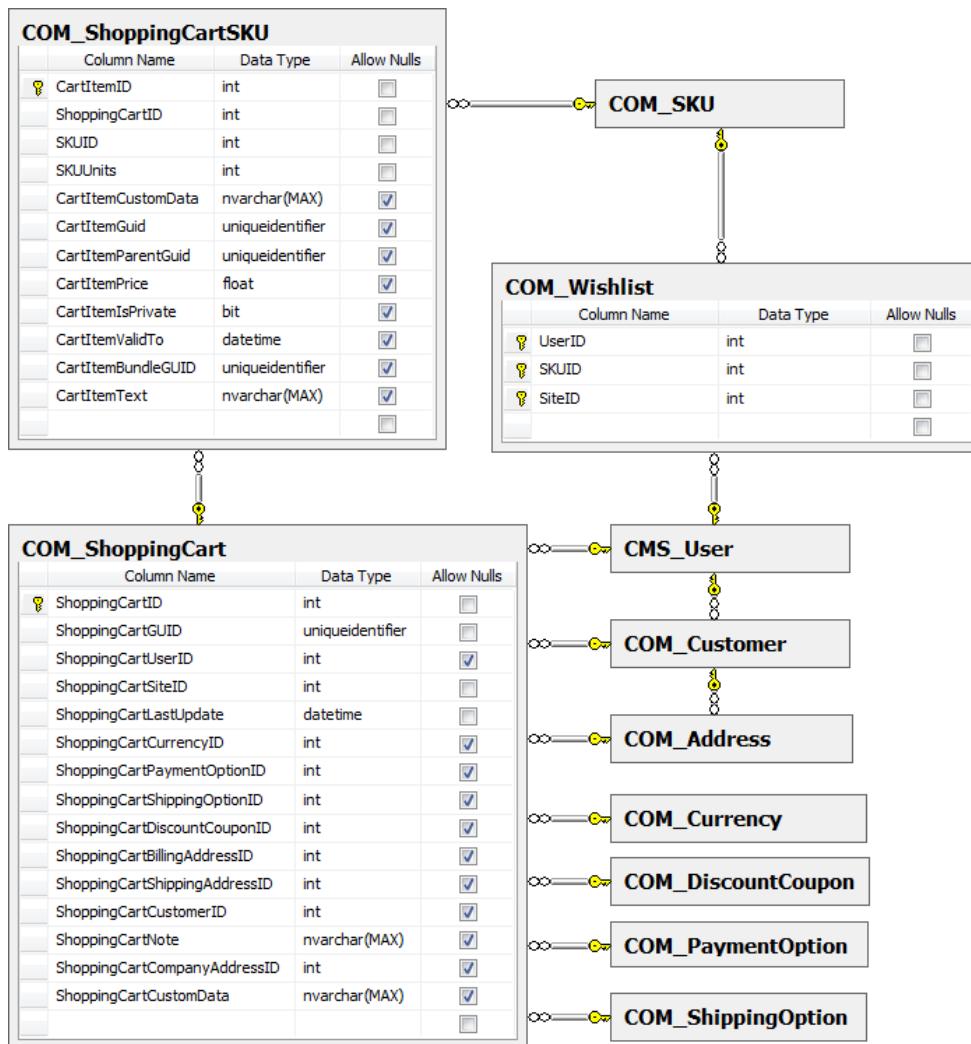
users, indicating that a particular user changed the status of a particular order (i.e. order status history).



## Shopping cart

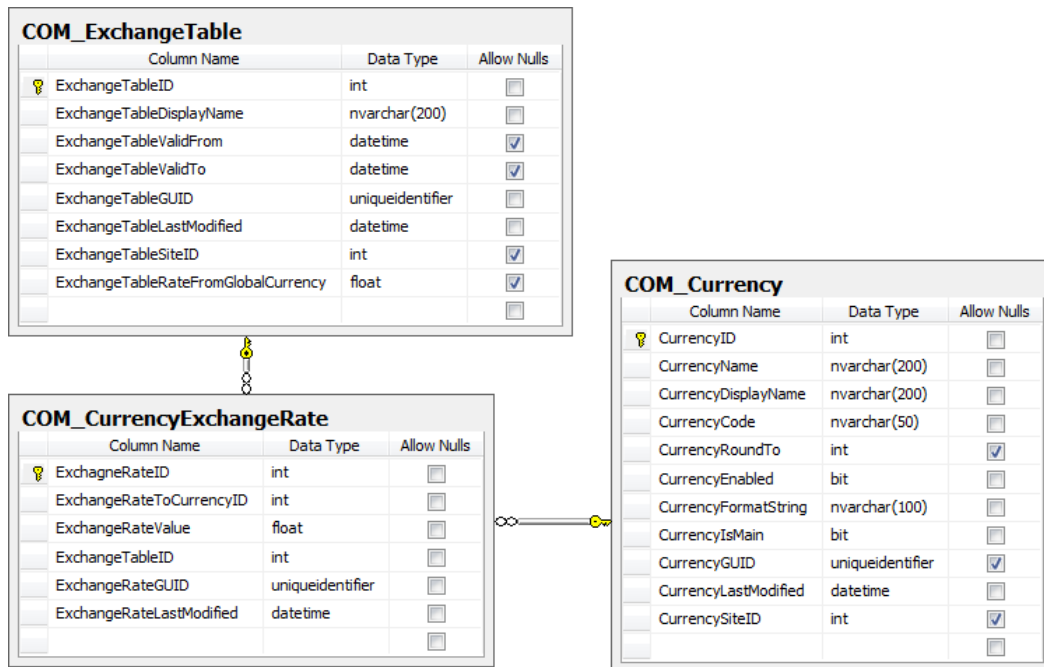
- **COM\_ShoppingCart** - contains records representing shopping carts.
- **COM\_ShoppingCartSKU** - contains records representing relationships between SKUs and shopping carts, indicating that particular SKUs are placed in a particular shopping cart.

- **COM\_Wishlist** - contains records representing wishlists.



## Currencies

- **COM\_Currency** - contains records representing currencies.
- **COM\_ExchangeTable** - contains records representing exchange tables.
- **COM\_CurrencyExchangeRate** - contains records representing exchange rates.



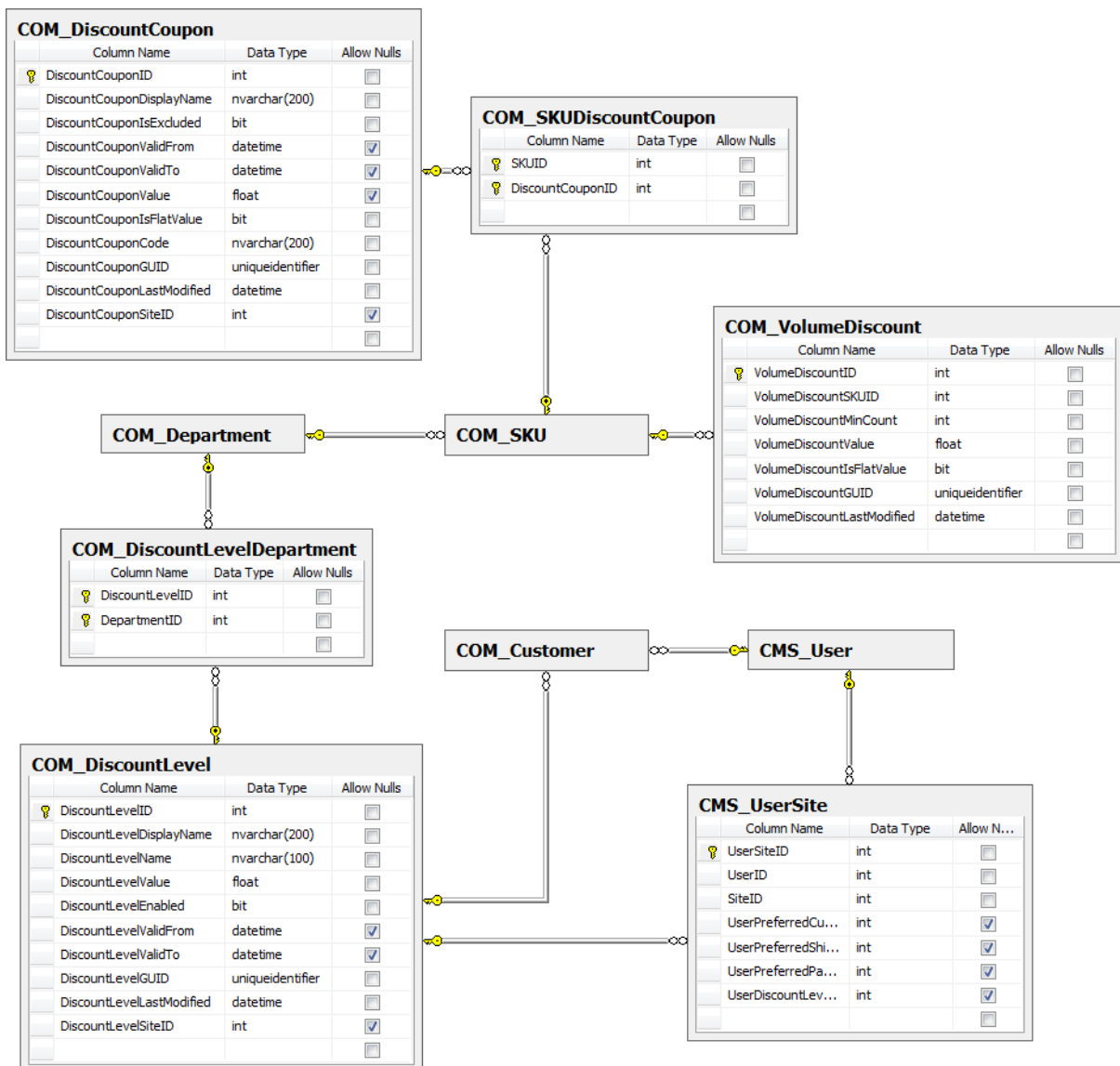
## Shipping and payment

- **COM\_PaymentOption** - contains records representing payment options.
- **COM\_PaymentShipping** - contains records representing relationships between payment and shipping, indicating that a given payment method is allowed for a given shipping.
- **COM\_ShippingOption** - contains records representing shipping options.
- **COM\_ShippingCost** - contains records representing shipping costs.



## Discounts

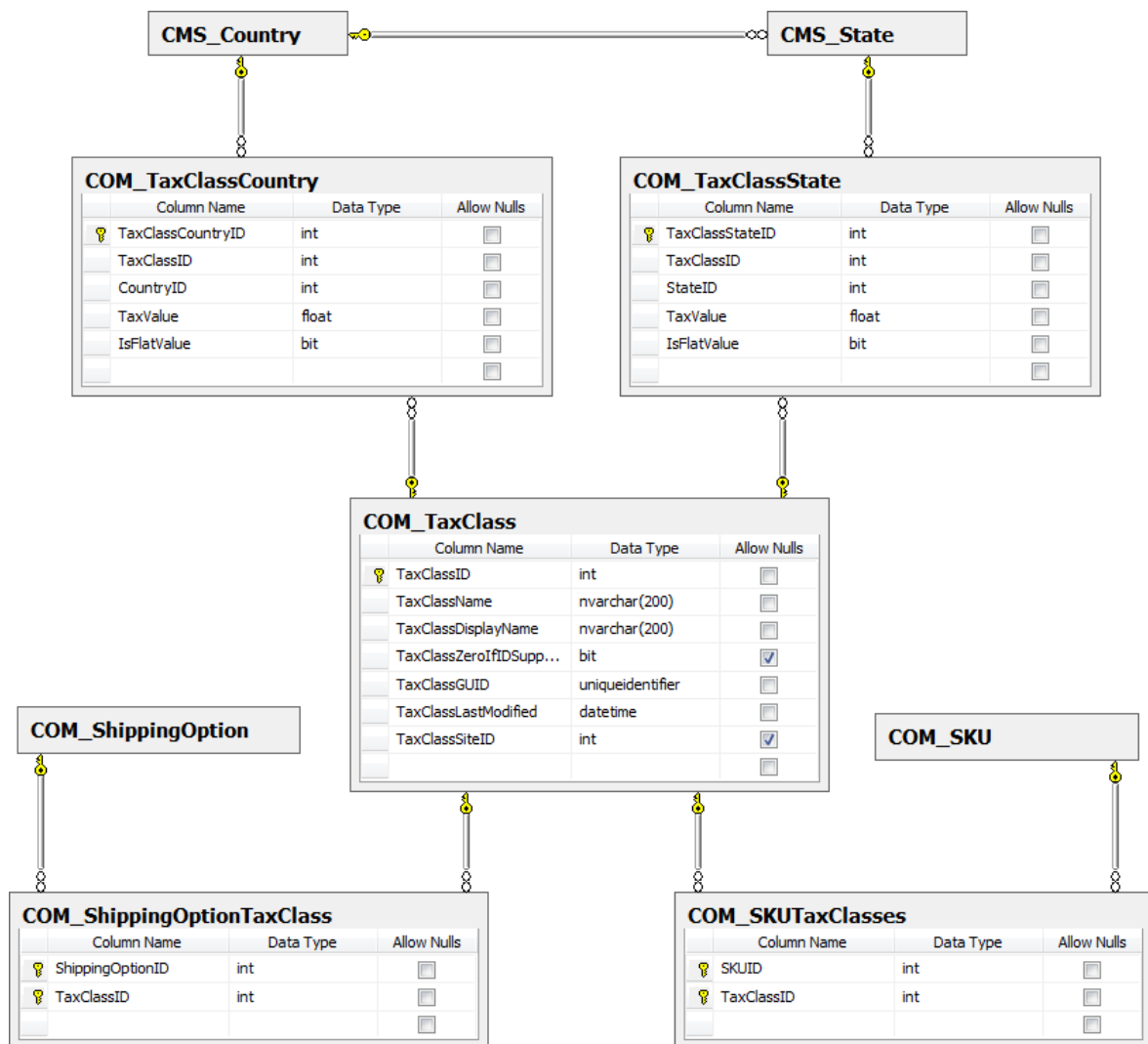
- **COM\_VolumeDiscount** - contains records representing volume discounts.
- **COM\_DiscountCoupon** - contains records representing discount coupons.
- **COM\_SKUDiscountCoupon** - contains records representing relationships between discount coupons and SKUs, indicating that particular discount coupons can be applied to a particular SKU.
- **COM\_DiscountLevel** - contains records representing discount levels.
- **COM\_DiscountLevelDepartment** - contains records representing relationships between discount levels and departments, indicating that a particular discount level can be applied to all products from a particular department.



## Taxes

- **COM\_TaxClass** - contains records representing tax classes.
- **COM\_TaxClassCountry** - contains records representing relationships between tax classes and countries, indicating the value of a particular tax in a particular country.
- **COM\_TaxClassState** - contains records representing relationships between tax classes and states, indicating the value of a particular tax in a particular state.
- **COM\_SKUTaxClasses** - contains records representing relationships between tax classes and SKUs, indicating that particular tax classes are assigned to a particular SKU.
- **COM\_DepartmentTaxClass** - contains records representing relationships between departments and tax classes, indicating that all new products in a particular department will have automatically assigned a particular tax.
- **COM\_ShippingOptionTaxClass** - contains records representing relationships between shipping options and tax classes, indicating that particular taxes are assigned to a particular shipping.





## 12.3 API classes

If you are not familiar with the database table data management by Info and Provider classes, we strongly recommend that you first refer to the [Database table API](#) topic in the API programming and Kentico CMS internals section of the Developer's Guide.

### Please note

The E-commerce module classes use the **CMS.Ecommerce** namespace.

- [Customers](#)
- [Products](#)
- [Orders](#)
- [Shopping cart](#)

- [Currencies](#)
- [Shipping and payment](#)
- [Discounts](#)
- [Taxes](#)

## Customers

### COM\_Customer table API:

- **CustomerInfo** - represents one customer.
- **CustomerInfoProvider** - provides management of customers.

### CMS\_User table API:

- **UserInfo** - represents one registered user.
- **UserInfoProvider** - provides management of registered users.

### CMS\_UserSite table API:

- **UserSiteInfo** - represents a relationship between one user and one site; a given user is assigned to a given site.
- **UserSiteInfoProvider** - provides management of relationships between users and sites.

### COM\_Address table API:

- **AddressInfo** - represents one customer address.
- **AddressInfoProvider** - provides management of customer addresses.

### CMS\_CustomerCreditHistory table API:

- **CreditEventInfo** - represents one customer credit history.
- **CreditEventInfoProvider** - provides management of customer credit histories.

## Products

### COM\_OptionCategory table API:

- **OptionCategoryInfo** - represents one product option category.
- **OptionCategoryInfoProvider** - provides management of product option categories.

### COM\_SKU table API:

- **SKUInfo** - represents one product (SKU).
- **SKUInfoProvider** - provides management of products (SKUs).

### COM\_SKUOptionCategory table API:

- **SKUOptionCategoryInfo** - represents a relationship between one product option category and one SKU, expressing that a particular product option category is available for a particular SKU.

- **SKUOptionCategoryInfoProvider** - provides management of relationships between product option categories and SKUs.

#### COM\_Bundle table API:

- **BundleInfo** - represents a relationship between one product of the bundle product type and one other product, expressing that this product is within this bundle.
- **BundleInfoProvider** - represents management of relationships between products of the bundle product type and other products.

#### COM\_SKUFile table API:

- **SKUFileInfo** - represents one e-product file.
- **SKUFileInfoProvider** - provides management of e-product files.

#### COM\_Manufacturer table API:

- **ManufacturerInfo** - represents one manufacturer.
- **ManufacturerInfoProvider** - provides management of manufacturers.

#### COM\_Supplier table API:

- **SupplierInfo** - represents one supplier.
- **SupplierInfoProvider** - provides management of suppliers.

#### COM\_Department table API:

- **DepartmentInfo** - represents one department.
- **DepartmentInfoProvider** - provides management of departments.

#### COM\_UserDepartment table API:

- **UserDepartmentInfo** - represents a relationship between one user and one department, expressing that a particular user can manage products of a particular department..
- **UserDepartmentInfoProvider** - provides management of relationships between users and departments.

#### COM\_InternalStatus table API

- **InternalStatusInfo** - represents one product internal status.
- **InternalStatusInfoProvider** - provides management of product internal statuses.

#### COM\_PublicStatus table API

- **PublicStatusInfo** - represents one product public status.
- **PublicStatusInfoProvider** - provides management of product public statuses.

### Orders

**COM\_Order table API:**

- **OrderInfo** - represents one order.
- **OrderInfoProvider** - provides management of orders.

**COM\_OrderItem table API:**

- **OrderItemInfo** - represents one order item.
- **OrderItemInfoProvider** - provides management of order items.

**COM\_OrderItemSKUFile table API:**

- **OrderItemSKUFileInfo** - represents a relationships between one order item and one e-product file, expressing that a particular order item comes with a particular e-product file. This relationship is created when an order containing an e-product is paid.
- **OrderItemSKUFileInfoProvider** - provides management of order items and e-product files.

**COM\_OrderStatus table API:**

- **OrderStatusInfo** - represents one order status.
- **OrderStatusInfoProvider** - provides management of order statuses.

**COM\_OrderStatusUser table API:**

- **OrderStatusUserInfo** - represents a relationship between one order status and one user, expressing order status changes, i.e. order status history.
- **OrderStatusUserInfoProvider** - provides management of relationships between order statuses and users.

**Shopping cart****COM\_ShoppingCart table API:**

- **ShoppingCartInfo** - represents one shopping cart.
- **ShoppingCartInfoProvider** - provides management of shopping carts.

**COM\_ShoppingCartSKU table API:**

- **ShoppingCartItemInfo** - represents a relationship between one SKU and one shopping cart expressing that a particular SKU is placed in a particular shopping cart.
- **ShoppingCartItemInfoProvider** - provides management of relationships between SKUs and shopping carts.

**COM\_Wishlist table API:**

- **WishlistItemInfo** - represents one item (product) in the wishlist.
- **WishlistItemInfoProvider** - represents management of items in the wishlists.

**Currencies**

**COM\_Currency table API:**

- **CurrencyInfo** - represents one currency.
- **CurrencyInfoProvider** - provides management of currencies.

**COM\_ExchangeTable table API:**

- **ExchangeTableInfo** - represents one exchange table.
- **ExchangeTableInfoProvider** - provides management of exchange tables.

**COM\_CurrencyExchangeRate table API:**

- **ExchangeRateInfo** - represents one exchange rate.
- **ExchangeRateInfoProvider** - provides management of exchange rates.

**Shipping and payment****COM\_PaymentOption table API:**

- **PaymentOptionInfo** - represents one payment option.
- **PaymentOptionInfoProvider** - provides management of payment options.

**COM\_PaymentShipping table API:**

- **PaymentShippingInfo** - represents a relationship between one payment method and one shipping, expressing that a given payment method is allowed for a given shipping.
- **PaymentShippingInfoProvider** - provides management of relationships payment methods and shipping.

**COM\_ShippingOption table API:**

- **ShippingOptionInfo** - represents one shipping option.
- **ShippingOptionInfoProvider** - provides management of shipping options.

**COM\_ShippingCost table API:**

- **ShippingCostInfo** - represents one shipping cost.
- **ShippingCostInfoProvider** - provides management of shipping costs.

**Discounts****COM\_VolumeDiscount table API:**

- **VolumeDiscountInfo** - represents one volume discount.
- **VolumeDiscountInfoProvider** - provides management of volume discounts.

**COM\_DiscountCoupon table API:**

- **DiscountCouponInfo** - represents one discount coupon.
- **DiscountCouponInfoProvider** - provides management of discount coupons.

#### **COM\_SKUDiscountCoupon table API:**

- **SKUDiscountCouponInfo** - represents a relationship between one discount coupon and one SKU expressing that a particular discount coupon can be applied to a particular SKU.
- **SKUDiscountCouponInfoProvider** - provides management of relationships between discount coupons and SKUs.

#### **COM\_DiscountLevel table API:**

- **DiscountLevelInfo** - represents one discount level.
- **DiscountLevelInfoProvider** - provides management of discount levels.

#### **COM\_DiscountLevelDepartment table API:**

- **DiscountLevelDepartmentInfo** - represents a relationship between one discount level and one department expressing that a particular discount level is available for a particular department.
- **DiscountLevelDepartmentInfoProvider** - provides management of relationships between discount levels and departments.

### **Taxes**

#### **COM\_TaxClass table API:**

- **TaxClassInfo** - represents one tax class.
- **TaxClassInfoProvider** - provides management of tax classes.

#### **COM\_TaxClassCountry table API:**

- **TaxClassCountryInfo** - represents a relationship between one tax class and one country, expressing the value of a particular tax in a particular country.
- **TaxClassCountryInfoProvider** - provides management of relationships between tax classes and countries.

#### **COM\_TaxClassState table API:**

- **TaxClassStateInfo** - represents a relationship between one tax class and one state, expressing the value of a particular tax in a particular state.
- **TaxClassStateInfoProvider** - provides management of relationships between tax classes and states.

#### **COM\_SKUTaxClasses table API:**

- **SKUTaxClassInfo** - represents a relationship between one tax class and one SKU, expressing that a particular tax class is assigned to a particular SKU.
- **SKUTaxClassInfoProvider** - provides management of relationships between tax classes and SKUs.

### COM\_DepartmentTaxClass table API:

- **DepartmentTaxClassInfo** - represents a relationship between one department and one tax class, expressing that all new products in a particular department will have automatically assigned a particular tax.
- **DepartmentTaxClassInfoProvider** - provides management of relationships between departments and tax classes.

### COM\_ShippingOptionTaxClass table API:

- **ShippingOptionTaxClassInfo** - represents a relationship between one shipping option and one tax class, expressing that a particular tax is assigned to a particular shipping.
- **ShippingOptionTaxClassInfoProvider** - provides management of relationships between shipping options and tax classes.

## 12.4 API examples

### 12.4.1 Overview



#### Please note

All API examples can be simulated in the API examples user interface. For more details, please refer to the [API examples](#) chapter in the API programming and Kentico CMS internals section of the Developer's Guide.

If you would like to check the code of the examples, please refer to **<web project>\CMSAPIExamples\Code\Modules\Ecommerce\Default.aspx.cs**.

All API examples are site-specific. To prevent malfunction, it is thus necessary to configure the E-commerce module to use [site data](#) only (in **Site Manager - > Settings -> E-commerce -> Global objects**).

**Part**

---

**XIII**

**Figure A: Tax calculation process**

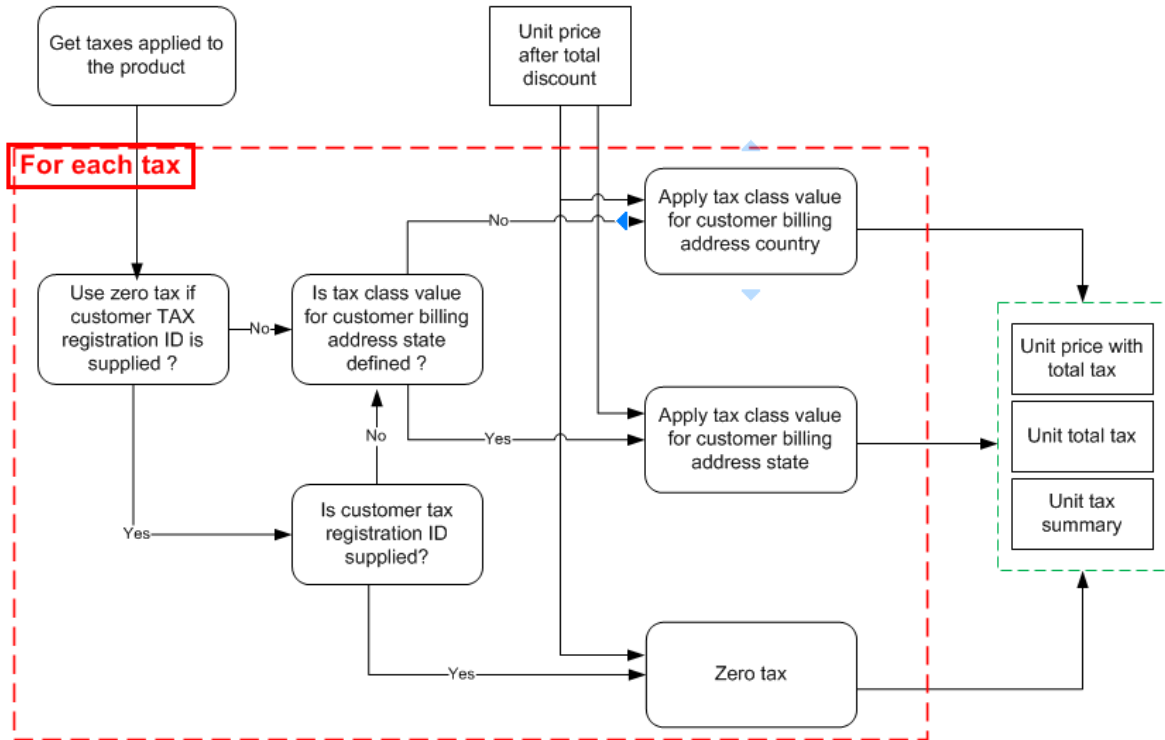
---



## 13 Figure A: Tax calculation process

The following figure shows how the tax is calculated:

### Taxes



**Part**

---



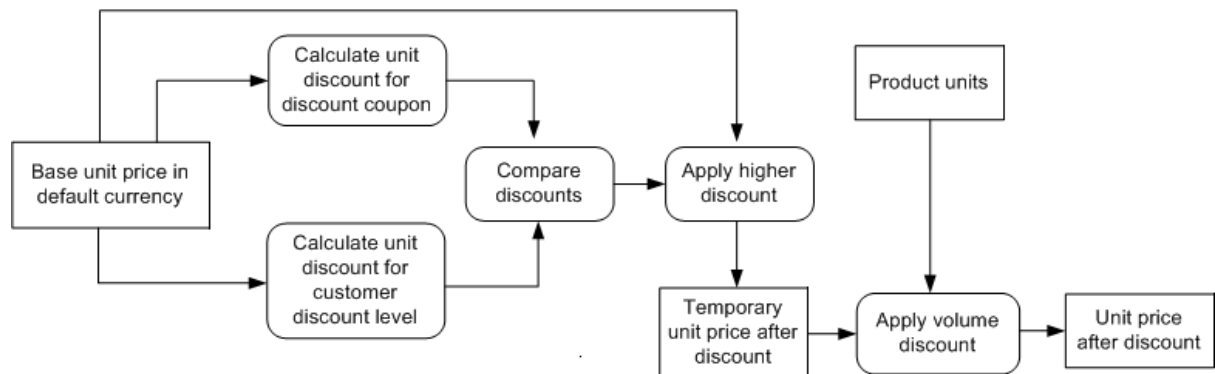
**Figure B: Discount calculation**

---

## 14 Figure B: Discount calculation

The following figure shows how discounts are calculated:

### Discounts



# Part

---



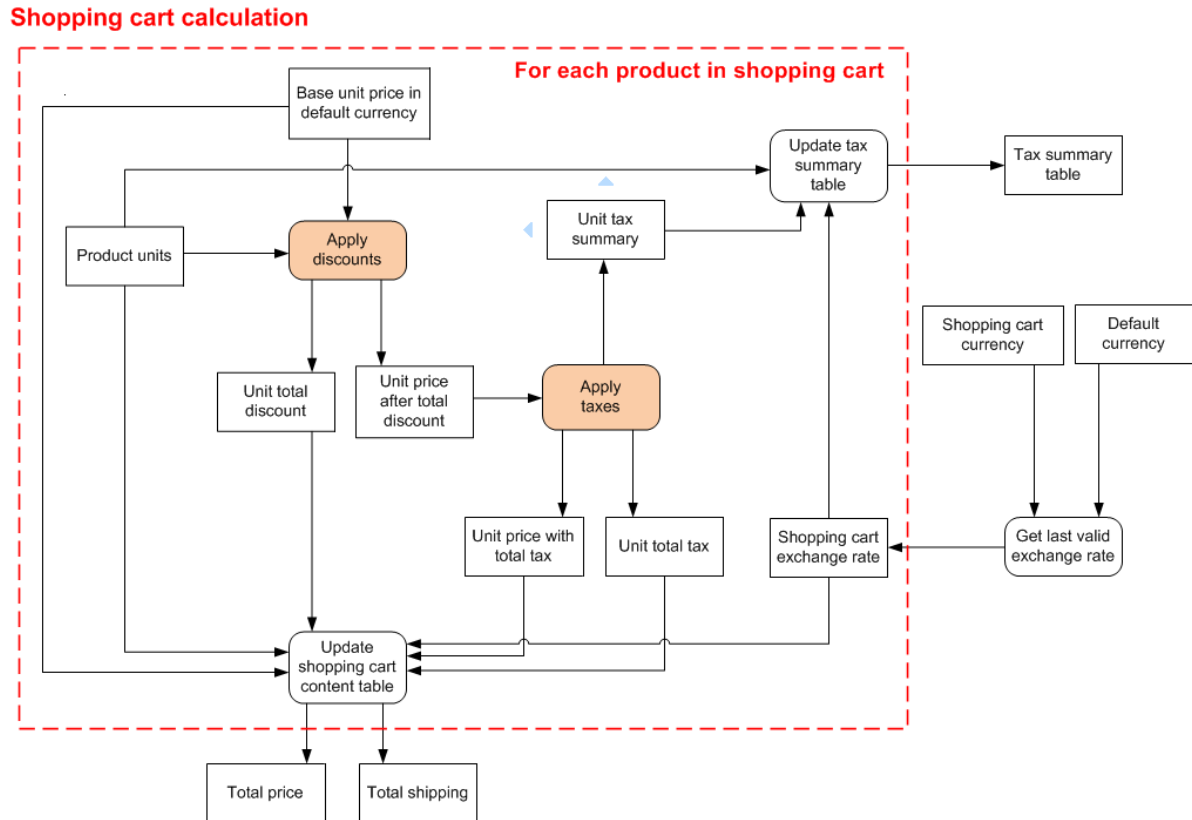
XV

**Figure C: Shopping cart price calculation**

---

## 15 Figure C: Shopping cart price calculation

The following figure shows how the total price is calculated:



**Part**

---

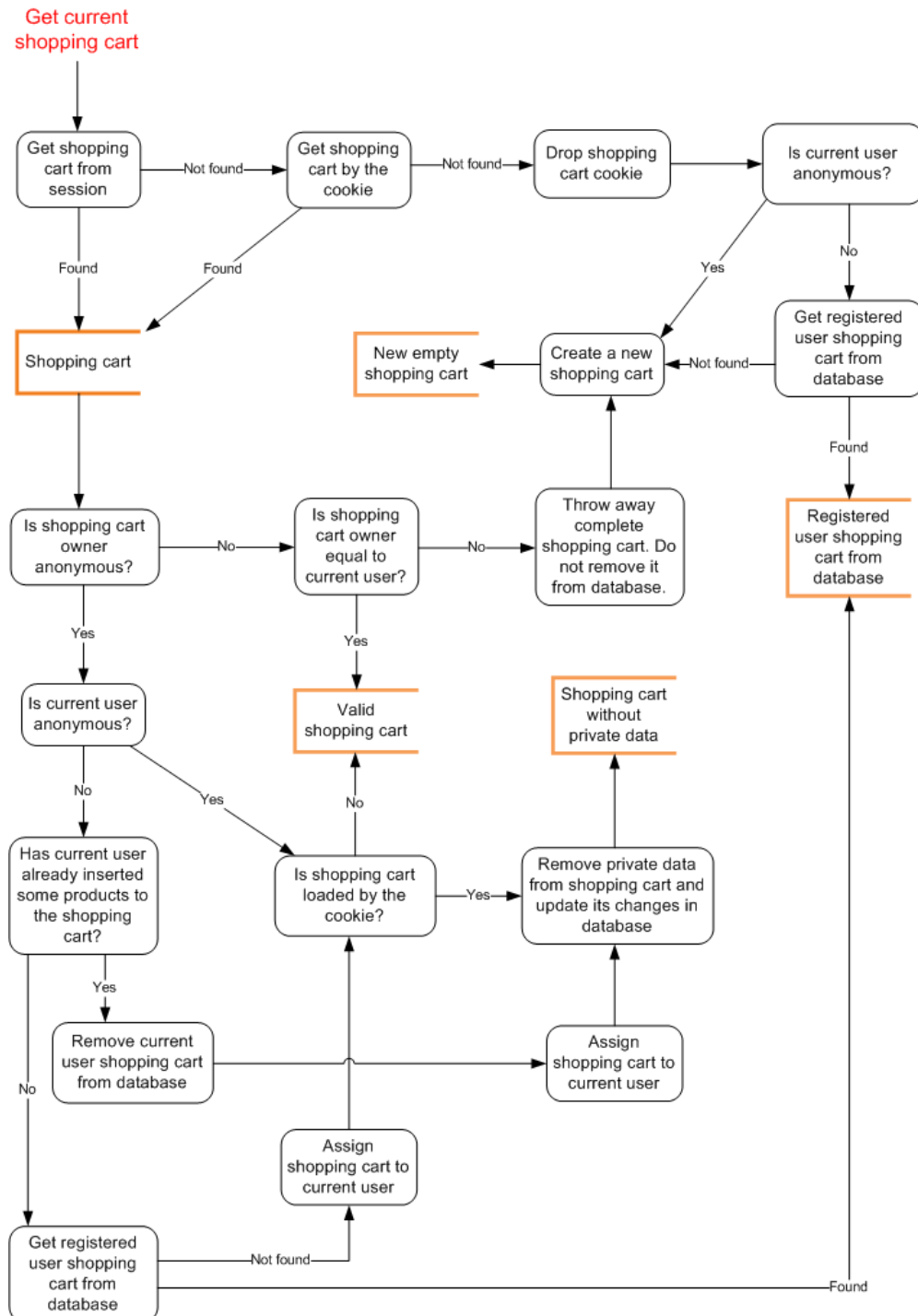
**XVI**

**Figure D: Shopping cart retrieval**

---

## 16 Figure D: Shopping cart retrieval

The following figure shows how the shopping cart is retrieved for the current visitor:



# Index

## - A -

Adding items to the shopping cart 111  
 Adding items to the wish list 114

## - C -

Company details  
   changing 36  
 Configuration settings  
   countries and states 139  
   currencies 134  
   customers 121  
   customizing an e-mail template 147  
   customizing invoice template 147  
   deleting old shopping carts 162  
   departments 129  
   discount coupons 127  
   discount levels 127  
   discounts overview 157  
   enabling the e-commerce module 137  
   exchange rates 135  
   internal status 136  
   invoice 136  
   localization 145  
   manufacturers 127  
   mapping document fields 141  
   multi-site configuration 140  
   order status 135  
   orders 120  
   overview 119  
   payment methods 132  
   product options 126  
   products 123  
   public status 135  
   reports 128  
   shipping options 130  
   sitemanager settings 141  
   store 128  
   suppliers 127  
   tax classes 134  
   using product inventory 152  
   web.config settings 144

Custom providers  
   overview 226

## - D -

Discount calculation 247

## - F -

First purchase 9

## - I -

Images  
   displaying 93  
   overview 93  
   resizing 96  
   storing 96  
 Integration with existing site  
   overview 73

## - O -

Overview 7

## - P -

Product  
   adding new 14  
   adding product options 18  
   adding to multiple categories 27  
   customizing product categories 23  
   displaying featured products 30  
   sorting and paging 32  
 Product fields  
   adding custom fields 56  
   adding images 53  
   customizing product design 51  
   defining new product type 43  
 Product price  
   displaying 115  
 Product URL  
   getting 116  
 Purchasing and payment  
   Authorize.NET configuration 197  
   custom checkout dialogs 190



Purchasing and payment  
    custom payment gateways 201  
    customer credit 209  
    customizing 188  
    overview 187  
    payment gateways 196  
    payment results 207  
    PayPal configuration 199  
    similar products by sale 86  
    top N products by sale 87  
    wishlist 91

## - S -

Searching 116  
Shopping cart price calculation 249  
Shopping cart retrieval 251

## - T -

Tax calculation process 245  
Tools  
    BizForms 100  
    Blogs 101  
    Booking system 101  
    Content staging 108  
    File import 102  
    Forums 102  
    GeoMapping 103  
    Image gallery 104  
    Messaging 105  
    Newsletters 106  
    overview 99  
    Polls 107  
    Reporting 107  
    RSS feeds 108  
    User contributions 109  
    Web analytics 109  
    Web farm synchronization 110

## - W -

Web design  
    colors using CSS styles 69  
    customizing master page 60  
Web parts  
    my account 74  
    random products 77  
    shopping cart 81  
    shopping cart preview 84