

When designing the pages of your store's [checkout process](#), you may want to display various types of information related to the current shopping cart, such as tax summaries or discount details.


## Tax summaries

A tax summary in Kentico is a list of all [tax classes](#) that apply to a shopping cart or order (including contained [products](#) and the selected [shipping option](#)), along with the sums of the tax values for each tax class.

By default, the system displays tax summaries only in [invoices](#). Checkout pages on the live site often display only the tax total.

However, if you wish to display detailed tax information on your checkout pages (for example in the order review step of the checkout process), you can add the tax summary using a combination of the **Static text** web part, [macros](#) and [transformations](#).

1. Open the **Pages** application.
2. Select the shopping cart page where you wish to display the tax summary.
3. Switch to the **Design** tab.
4. [Place](#) the **Static text** web part onto the page.
5. In the web part's configuration, enter an appropriate macro expression into the web part's **Text** property.

 The tax summary data is available in macros within the **TaxSummary** property of the current shopping cart object.

### Example

```
{% EcommerceContext.CurrentShoppingCart.TaxSummary.Any() ?  
    EcommerceContext.CurrentShoppingCart.TaxSummary.ApplyTransformation(  
        "Ecommerce.Transformations.Live_TaxTable",  
        "Ecommerce.Transformations.Order_TaxesTableHeader",  
        "Ecommerce.Transformations.Order_TaxesTableFooter")  
: "No taxes applied." %}
```

6. Click **Save & Close**.

Define the appearance of the tax summary through the [transformations](#) assigned into the parameters of the *ApplyTransformation* method in the macro expression. The sample macro above uses the default tax summary table transformations, with the main item transformation modified for use on live site pages.

1. Open the **Page types** application.
2. Edit the page type where you wish to store the tax summary transformations. The example uses the default **E-commerce - Transformations** page type.
3. Select the **Transformations** tab and click **New transformation**.
4. Set the transformation properties and code:

- **Transformation name:** Live\_TaxTable (*example*)
- **Transformation type:** Text / XML
- **Code:**

```
<tr>  
  <td>{% HTMLEncode(Localize(Name)) %}</td>  
  <td>{% FormatPrice(Value) %}</td>  
</tr>
```

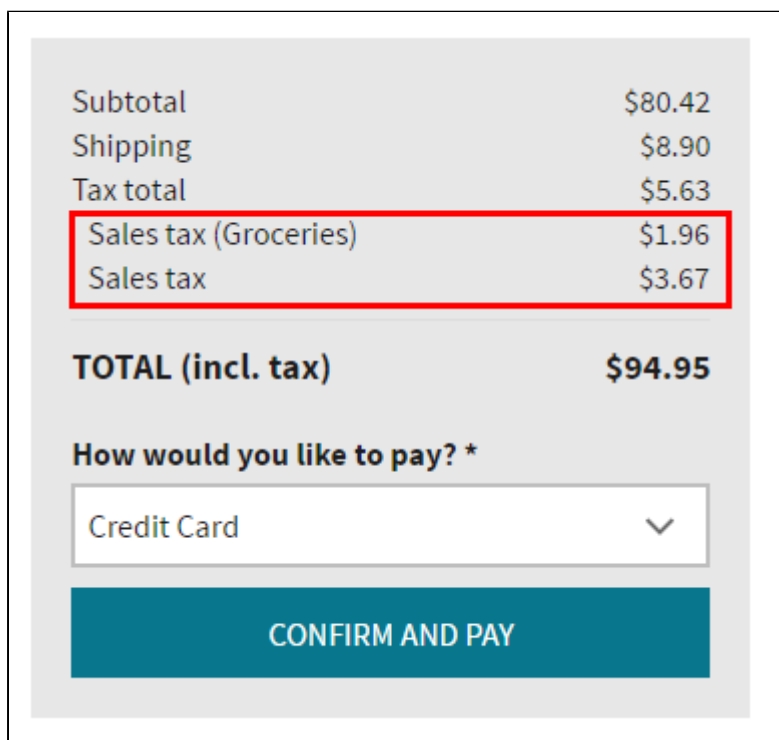
**i** You can add the transformations under any page type and use any transformation name. You only need to adjust the parameters of the **ApplyTransformation** method in the tax summary macro correspondingly.

When writing transformations for tax summary data items, the tax class data is available in the **Name** and **Value** fields.

5. Click **Save**.

6. (Optional) Create custom header and footer transformations if you use them in the *ApplyTransformation* macro method. The example uses the default tax summary table transformations.

The page where you added the *Static text* web part now displays a tax summary for the current shopping cart. The appearance of the tax summary is determined by the applied transformations.



The screenshot shows a checkout page with a tax summary table and a payment section. The tax summary table has the following items:

Subtotal	\$80.42
Shipping	\$8.90
Tax total	\$5.63
Sales tax (Groceries)	\$1.96
Sales tax	\$3.67

The 'Sales tax (Groceries)' and 'Sales tax' rows are highlighted with a red border. Below the table, the **TOTAL (incl. tax)** is \$94.95. The payment section is titled 'How would you like to pay? \*' and features a dropdown menu with 'Credit Card' selected. A large teal button labeled 'CONFIRM AND PAY' is at the bottom.

## Discounts applied to shopping cart items

**i** **Note:** The information in this section does not apply to [catalog](#) and [volume discounts](#). These discounts are included directly in the basic unit price of each product.

When a customer's shopping cart fulfills the conditions of a [Buy X Get Y discount](#) or the customer enters a [product coupon](#), the system applies discounts to individual items in the shopping cart. Each shopping cart item represents one or more units of a specific [product](#).

To display information about the applied discounts on your store's checkout pages, include the appropriate code in the [transformation](#) that you use to format the content of the shopping cart. A typical way of displaying the shopping cart items is using the **S hopping cart content** web part together with **ASCX** transformations.

When editing your shopping cart item ASCX transformations, call the **GetMultiBuyDiscountNames** method to generate the names and values of Buy X Get Y discounts and product coupons applied to the given item, enclosed within **<li>** HTML tags.

```
<%# IfEmpty(GetMultiBuyDiscountNames(), "", "<span>Special discounts:</span><ul>" +  
GetMultiBuyDiscountNames() + "</ul>") %>
```

If you want full control over the HTML output of the discount content, you can instead add inline code and process the **DiscountSummary** property of the transformed shopping cart item.

#### Example

```
<% var summary = ((ShoppingCartItemInfo)DataItem).DiscountSummary;  
if ((summary != null) && summary.Count > 0) { %>  
    <span>Special discounts:</span>  
    <ul>  
        <% foreach(var discountItem in summary) { %>  
            <li><%= HTMLEncode(Localize(discountItem.Name)) %> <strong>(<%= FormatPrice  
(discountItem.Value) %>)</strong></li>  
        <% } %>  
    </ul>  
<% } %>
```



The summary items stored in **DiscountSummary** provide the following properties:

- **Name** – the display name of the discount as specified in the **Buy X Get Y discounts** or **Product coupons** application.
- **Value** – the total amount saved after applying the discount to the given shopping cart item (for all units). To format the value into the [format string of the currently selected currency](#), use the *FormatPrice* method.



## Your shopping cart



**Bolivia Finca Illimani (4 oz) (In stock)**

**\$12.60**

Qty

6

[UPDATE](#)

[REMOVE](#)

Special discounts:

Coffee discount coupon **(\$2.10)**

Coffee on 3 for 2 discount **(\$6.30)**



**Clever Dripper (In stock)**

**\$21.90**

Qty

1

[UPDATE](#)

[REMOVE](#)



**Bleached Paper Filters V60-01 (In stock)**

**\$0.00**

Qty

1

[UPDATE](#)

[REMOVE](#)

Special discounts:

Free filters w/ Clever Dripper **(\$3.90)**

### Displaying discount summaries separately from the main cart content

Alternatively, you can display a summary of Buy X Get Y discounts and product coupons separately from the main shopping cart content. For example, the following steps show how to use the **Static text** web part and [macros](#) to generate a basic list of all applied Buy X Get Y discounts and product coupons.

1. Open the **Pages** application.
2. Select the shopping cart page where you wish to display the discount summary.
3. Switch to the **Design** tab.
4. [Place](#) the **Static text** web part onto the page.
5. In the web part's configuration, enter an appropriate macro expression into the web part's **Text** property.



### Example

```
{%
  result = "";
  cartItems = ECommerceContext.CurrentShoppingCart.ShoppingCartItems;
  if (cartItems.Any()) {
    result += "<ul>";
    foreach (item in cartItems) {
      result += "<li>" + HTMLEncode(Localize(item.SKU.SKUName));
      itemDiscountSummary = item.DiscountSummary;
      if (itemDiscountSummary.Any()) {
        result += "<ul>";
        foreach (summaryItem in itemDiscountSummary) {
          result += "<li>" + HTMLEncode(Localize(summaryItem.Name)) + " (" +
FormatPrice(summaryItem.Value) + ")</li>";
        };
        result += "</ul>";
      }
      result += "</li>";
    };
    result += "</ul>";
  }
  return result;
}%}
```

#### 6. Click **Save & Close**.

The first level of the list displays the names of all items in the current shopping cart. Items that have their price reduced by Buy X Get Y discounts or product coupons have a second list level with the names and values of the given discounts.

## Order discounts

Any [order discounts](#) that apply to the displayed shopping cart reduce its price total.

To display the total value of applied order discounts, add the [Shopping cart totals web part](#) onto the page, with the **Total to display** property set to *Order discounts total*.



Alternatively, you can display the order discount total value by using the following [macro expression](#):

```
{% FormatPrice(ECommerceContext.CurrentShoppingCart.OrderDiscount) %}
```

If you wish to display a detailed order discount summary on your checkout pages, you can use a combination of the **Static text** web part and macros:

1. Open the **Pages** application.
2. Select the shopping cart page where you wish to display the order discount summary.
3. Switch to the **Design** tab.
4. [Place](#) the **Static text** web part onto the page.
5. In the web part's configuration, enter an appropriate macro expression into the web part's **Text** property.



#### Example

```
{%
result = "";
orderDiscountSummary = ECommerceContext.CurrentShoppingCart.OrderDiscountSummary;
if (orderDiscountSummary.Any()) {
    result += "<ul>";
    foreach (item in orderDiscountSummary) {
        result += "<li>" + HTMLEncode(Localize(item.Name)) + " (- " + FormatPrice
(item.Value) + ")</li>"
    };
    result += "</ul>";
}
return result;
%}
```

#### 6. Click **Save & Close**.

The `{% ECommerceContext.CurrentShoppingCart.OrderDiscountSummary %}` macro provides a collection of summary items representing order discounts applied to the current shopping cart. You can access the display name and value of each order discount in the **Name** and **Value** fields of the summary items, and format the data according to your site's requirements.

## Gift cards

When a customer redeems a [gift card](#) during checkout, the system subtracts the gift card's value from the final price of the shopping cart.

To display the value of applied gift cards, add the [Shopping cart totals web part](#) onto the page, with the **Total to display** property set to *Other payments total*.



Alternatively, you can display the gift card total value by using the following [macro expression](#):

```
{% FormatPrice(ECommerceContext.CurrentShoppingCart.OtherPayments) %}
```

If you wish to display a detailed gift card summary on your checkout pages, you can use a combination of the **Static text** web part and macros:

1. Open the **Pages** application.
2. Select the shopping cart page where you wish to display the gift card summary.
3. Switch to the **Design** tab.
4. [Place](#) the **Static text** web part onto the page.
5. In the web part's configuration, enter an appropriate macro expression into the web part's **Text** property.



#### Example

```
{%
result = "";
giftCardSummary = ECommerceContext.CurrentShoppingCart.OtherPaymentsSummary;
if (giftCardSummary.Any()) {
    result += "<ul>";
    foreach (item in giftCardSummary) {
        result += "<li>" + HTMLEncode(Localize(item.Name)) + " (- " + FormatPrice
(item.Value) + ")</li>"
    };
    result += "</ul>";
}
return result;
%}
```

#### 6. Click **Save & Close**.

The `{% ECommerceContext.CurrentShoppingCart.OtherPaymentsSummary %}` macro provides a collection of summary items representing gift cards applied to the current shopping cart. You can access the display name and value of each gift card in the **Name** and **Value** fields of the summary items, and format the data according to your site's requirements.