Bulk Quantity Promotion Plugin

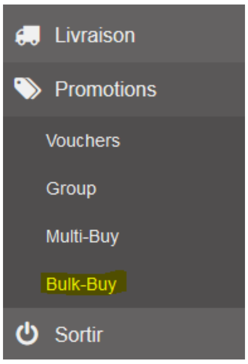
Introduction

This document explains how to install and integrate the Bulk Quantity promotion plugin for NBSv3.

Installation

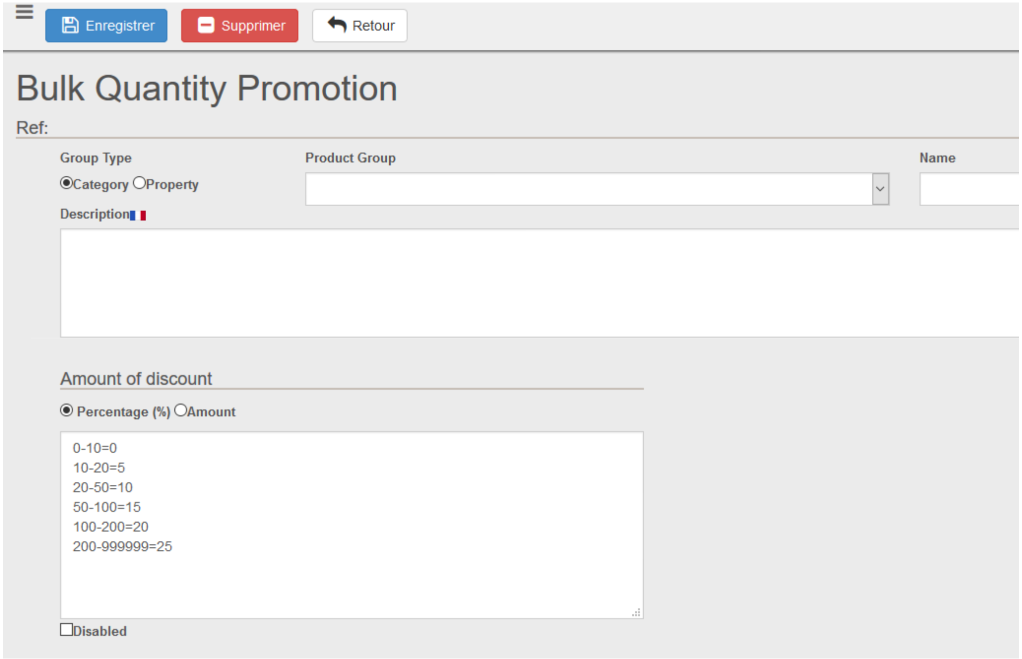
Install the plugin Install zip like you would any normal DNN module. Then go into the NBSv3 Back Office and goto the Admin>Plugin menu. This should install the required plugin data into the Back Office menu. You should then find a BO>Promotions>Bulk-Buy option on the menu.

NOTE: If you using a NBS version prior to v3.5.2.2 then the plugin it injected at the top of the list, this pushes the dashboard option down and therefore you need to move the dashboard plugin back to the top of the list.



Adding a Promotion

Go into the Bulk-Buy option and you can create a new promotion by click the “Add” button above the list. This will open a new promotion detail page for you.



Bulk Quantity Promotion

Ref:

Group Type

● Category ○ Property

Product Group

Name

Description

Amount of discount

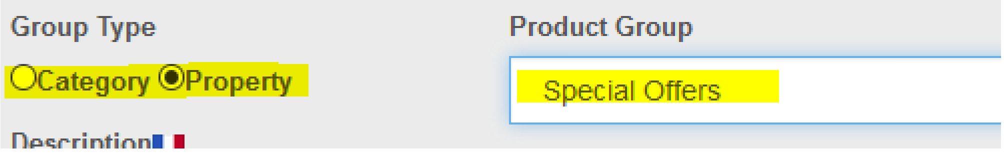
● Percentage (%) ○ Amount

```
0-10=0
10-20=5
20-50=10
50-100=15
100-200=20
200-999999=25
```

□ Disabled

Assign to Property or Category

Bulk Promotions work by assigning the promotion to a category or property. All products that are in the category or have the property will have the promotion activate.



Group Type

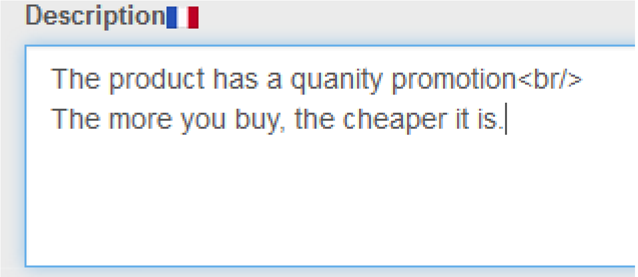
○Category ●Property

Product Group

Special Offers

Description Information

The description field allows you to enter a description which can be displayed on the product when the promotion is activate. This is localized to the language being displayed. HTML code can be entered, to make the display appear as formatted on the product detail.

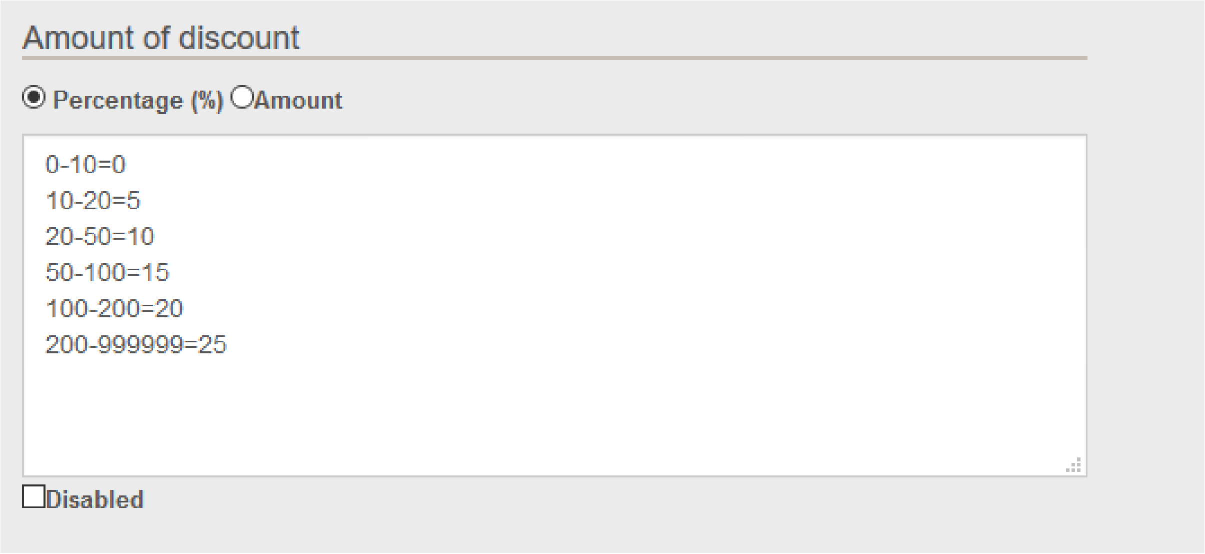


Description

The product has a quanity promotion
The more you buy, the cheaper it is.

Promotion Qty Range

The range of quantity promotion is entered into the "Amount of discount" textbox and can be a percentage to an amount. Each row represents a new quantity range.



Amount of discount

● Percentage (%) ○Amount

0-10=0
10-20=5
20-50=10
50-100=15
100-200=20
200-999999=25

□Disabled

Displaying the Promotion detail on the product display

The description of the promotion or any other data you wish can be displayed on the product display by altering the display template being used. (By default “NBS_ProductDisplayDetail.cshtml” or NBS_ProductDisplayList.cshtml)

There are 2 methods to displaying data

1. Inherit QtyPromo Razor Tokens (Recommended)

This method involves altering the display templates to inherit the “QtyPromoRazorTokens” class.

In the display template the 1st line of the template should define the “NBrightBuyRazorTokens”

@inherits NBrightBuy.render.NBrightBuyRazorTokens<NBrightRazor>

This line needs to be removed and replaced with the “QtyPromoRazorTokens” class.

@inherits NBrightBuy.QtyPromo.render.QtyPromoRazorTokens<NBrightRazor>

You can now use the Qty Promo tokens which have been defined in the Qty promo Plugin.

Example:

```
@{
   var hasQtyPromo = HasQtyPromo(info.ItemID);
   if (hasQtyPromo.ToString() == "true")
   {
           <div>
                      FROM: @QtyPromoFromPrice(info.ItemID)
           </div>
           <div>
                      @QtyPromoDescription(info.ItemID)
           </div>
   }
   else
   {
      <div>NO PROMOTION</div>
   }
}
```

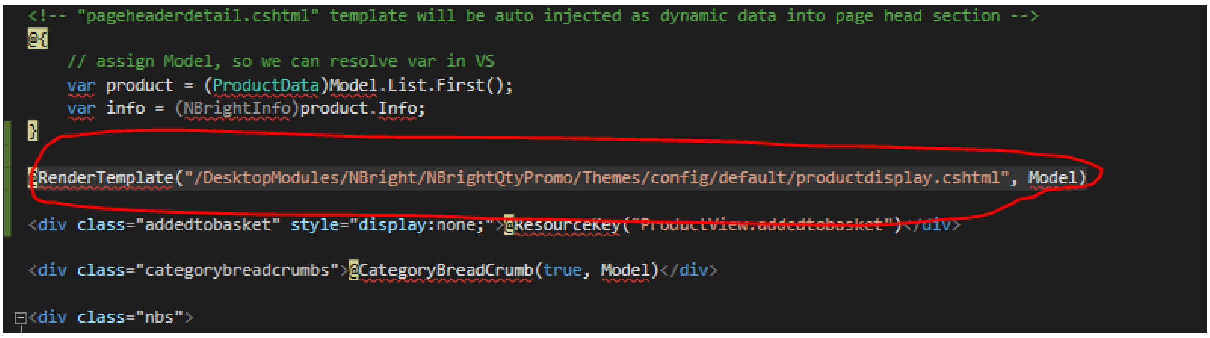
2. Use @RenderTemplate token

Note: Depending on which version of NBS you use, this method may be affected by cache and cause duplicate display across products.

This works by editing the template and adding a token to display a template. The token to be entered into the display template is the “@RenderTemplate”

In the example below we've linked the token to an example template which is delivered with QtyPromo install.

@RenderTemplate("/DesktopModules/NBright/NBrightQtyPromo/Themes/config/default/productdisplay.cshtml", Model)



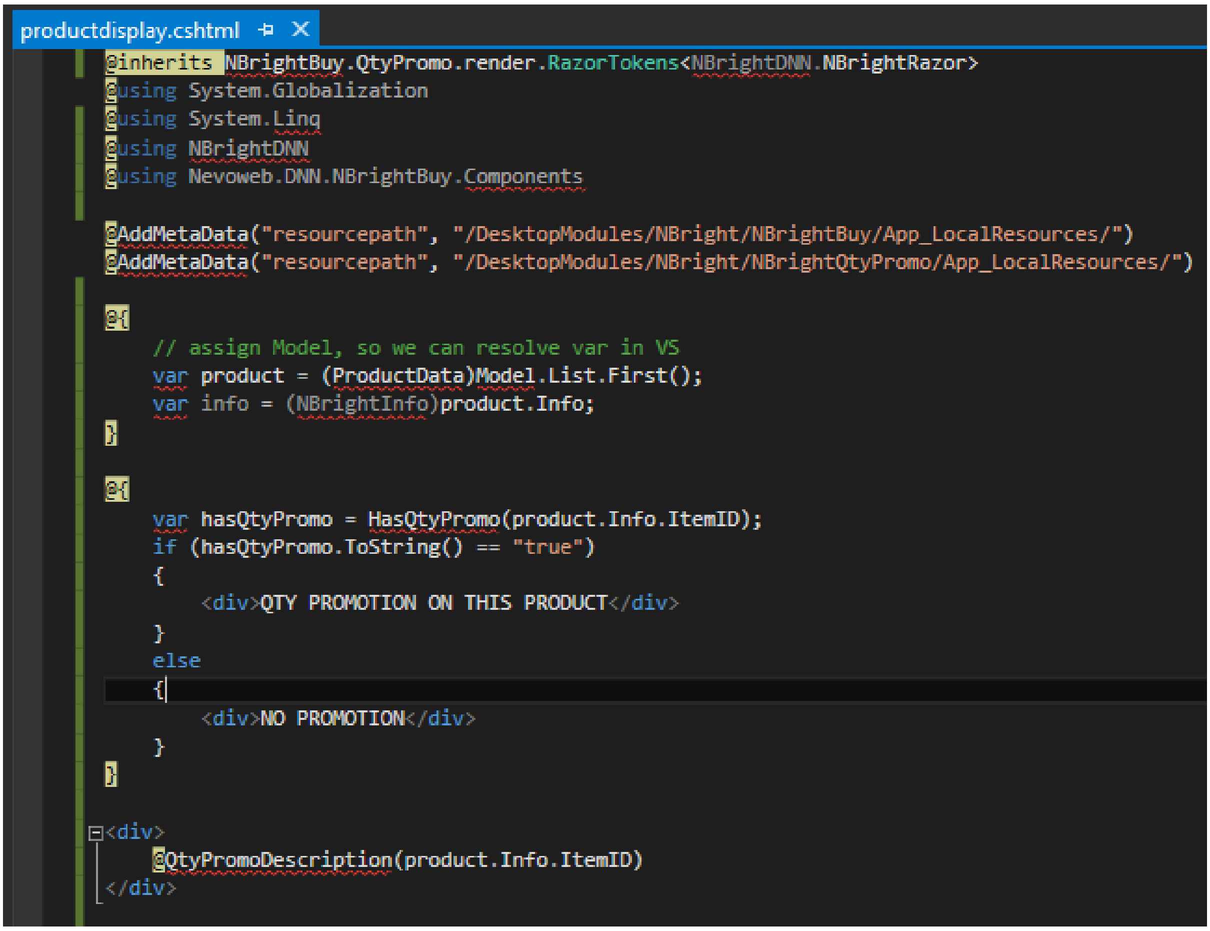
```
<!-- "pageheaderdetail.cshtml" template will be auto injected as dynamic data into page head section -->
@{
    // assign Model, so we can resolve var in VS
    var product = (ProductData)Model.List.First();
    var info = (NBrightInfo)product.Info;
}

@RenderTemplate("/DesktopModules/NBright/NBrightQtyPromo/Themes/config/default/productdisplay.cshtml", Model)

<div class="addedtobasket" style="display:none;">@ResourceKey("ProductView.addedtobasket")</div>

<div class="categorybreadcrumbs">@CategoryBreadCrumb(true, Model)</div>

<div class="nbs">
```

This token is rendering the example template which renders the data using special 2 tokens.

@HasQtyPromo

@QtyPromoDescription.

Look at the example template in the plugin as an example how it can be edited to match your needs.



productdisplay.cshtml

```
@inherits NBrightBuy.QtyPromo.render.RazorTokens<NBrightDNN.NBrightRazor>
@using System.Globalization
@using System.Linq
@using NBrightDNN
@using Nevoweb.DNN.NBrightBuy.Components

@AddMetaData("resourcepath", "/DesktopModules/NBright/NBrightBuy/App_LocalResources/")
@AddMetaData("resourcepath", "/DesktopModules/NBright/NBrightQtyPromo/App_LocalResources/")

@{
    // assign Model, so we can resolve var in VS
    var product = (ProductData)Model.List.First();
    var info = (NBrightInfo)product.Info;
}

@{
    var hasQtyPromo = HasQtyPromo(product.Info.ItemID);
    if (hasQtyPromo.ToString() == "true")
    {
        <div>QTY PROMOTION ON THIS PRODUCT</div>
    }
    else
    {
        <div>NO PROMOTION</div>
    }
}

<div>
    @QtyPromoDescription(product.Info.ItemID)
</div>
```