

Promotions

- Overview
 - Promotion types
 - Best deal strategy and combined discounts
 - Promotion calculation process
 - Eligibility conditions
 - Available variables for eligibility condition
 - Eligibility conditions examples
 - Fixed amount off
 - Percent Off
 - Percent Off (non sale)
 - Gift
 - Customer segmentation
- Promotion management
 - Basic information
 - Promotion condition
 - Coupon codes management
 - Multi use coupons
 - Single use coupons
 - Mixed use coupons
- Examples
 - Fully Annotated Example of Product Promotion
 - Fully Annotated Example of Product Coupon Promotion
- Validating promotions configuration

Overview

Promotions allow to apply discounts and gifts dynamically depending on contents of shopping cart, specific customer, triggered by coupon code (also known as promo code or voucher) or a combination of the three. In contrast to [price lists](#) which provide fixed predefined discounts promotions are **determined and calculated during runtime** to give the final price. More specifically **every time shopping cart is modified** in any way (e.g. currency change, items added or removed, billing or shipping address selected, shipping or payment method selected) it is recalculated taking into account all active promotions.

Promotions allow business user to **define various rules which allow customer to receive special discounts and gifts**. The rules consist of **promotion type**, **eligibility condition**, **action required** if condition is met and **action context** (a parameter passed to action).

Promotion types

The following types of promotions are supported:

Promotion Type	Description	Run target
Item level	Order line discounts or gifts triggered by properties of SKU	Storefront, during cart calculation
Order level	Order subtotal discounts or gifts triggered by overall order (usually total reaching a certain limit)	Storefront, during cart calculation
Shipping level	Shipping costs discount	Storefront, during cart calculation
Customer	Automatic customer profile tagging to put customers in different segments. Each tag can be used in promotion conditions at item, order and shipping level.	Admin, periodic bulk evaluation using customerTagJob

Results of the item and order promotions being applied can be observed on mini cart and shopping cart page. Figure 1 shows an excessively complicated but illustrative demonstration of different promotions contributing to order at different levels, including coupon triggered discounts, gifts and compound discounts which are given on top of the multi buy sale price.

Figure 1: Default theme shopping cart page with promotions

[Sale](#)
[Special Offer](#)
[New Arrivals](#)
[Demo](#)
[Contacts](#)

[Login / Register](#)
[Language Eng](#)
[Currency €](#)

[Web site View \(shopping cart\)](#)

[Contact form](#)

(9) **€4717⁵⁸**

Product	Price	Quantity	Amount
DS215+ SKU code DS215---2X-ST2000VX000 Add to wishlist Save for later	657¹⁷ Discount 7% SHOP10EURC0001:50EUROFF	- 2 + ↺ ↻	1314³⁴ ×
CB5-571-C4Y3 SKU code NX-MUNET-002 Add to wishlist Save for later	575⁸⁴ Discount 28% SHOP10EURITEMQTY10P	- 6 + ↺ ↻	3453²⁴ ×
Trigger-Z SKU code SGK-6010-GKCM1-DE Add to wishlist Save for later	0⁰⁰ Discount 100% SHOP10EURGIFT	1 ↺ ↻	

Shopping cart
€4717⁵⁸
 Discount 45%
 SHOP10EURGIFT,SHOP10EURORD50F
[Checkout](#)

Add Coupons
 50EUROFF [applied] ×

[Add coupon](#)

Order Message
 Add order message

[Edit](#)

Compound discount:
20% off price list sale, leaving sale price at 80% and 10% off via promotion leaving 72% of the price i.e. 28% discount

SKU triggered gift

Coupon triggered item discount

Order level promotions applied

Mini cart displays discounted sub total

Coupon codes applied

[Profile](#)
[View shopping cart](#)
[Order guide](#)
[Order history](#)

[FAQ](#)
[Return policy](#)
[Site map](#)
[Call center](#)

[Google+ 0](#)
[Twitter 0](#)
[Facebook Like 10](#)

[Contacts](#)
[Licences](#)
[Site map](#)

YesCart.org 2015

Shipping promotions can be seen on the shipping method selection page. Figure 2 demonstrated a free shipping promotion given when order exceeds a certain value (as stated in this particular shipping promotion condition)

Figure 2: Default theme checkout shipping method selection page with promotions

John Smith


Language

Eng

Currency

€

Web site View (checkout)



(9)

€4717⁵⁸

1 Log In

2 Your Address

3 Shipping Method

4 Payment

Select carrier and shipping method

Carrier

Courier

Shipping method

1-3 day EUR

Shipping cost

€0⁰⁰

Discount 100%
SHOP10EURSHIPFREE

Shipping level promotion

< Your Address

Payment >

Profile

View shopping cart

Order guide

Order history

FAQ

Return policy

Site map

Call center

Item, order and shipping promotion discounts applied to an order can be viewed on the order review page just before payment. The review page shows less detail nonetheless states all promotions applied and the overall discount for the order.

Figure 3: Default theme checkout order review page with promotions

Web site
View
(checkout)

John Smith  Language Eng Currency €



YES CART
DEMO STORE

 (9) €4717⁵⁸

1 Log In

2 Your Address

3 Shipping Method

4 Payment

Verify order and choose payment method

Delivery number 150821122308-13-0

Product	Final (discounted) prices	Price	Quantity	Amount
 DS215+ DS215----2X-ST2000VX000		657.17	2.00	1314.34
 CB5-571-C4Y3 NX-MUNET-002		575.54	6.00	3453.24
 Trigger-Z SGK-6010-GKCM1-DE		0.00	1.00	0.00

Order total
Tax
Order total (inc. tax)

Order final total
with order
discounts with
grand total
discount given

4717.58
786.31
€4717⁵⁸

Address SHOP10EURSHIPFREE, SHOP10EURITEMQTY10P, SHOP10EURGIFT, SHOP10EURORD50F,
SHOP10EURC0001:50EUROFF
Buckingham Palace SW1A 0AA Londonh GB United Kingdom, John Smith,
123412341324

Items sub
total with
discounts

Delivery sub total 4767.58
Tax 794.64
Sub total (inc. tax) 4767.58

Method 1-3 day EUR
Shipping cost 0.00
Tax 0.00
Shipping cost (inc. tax) 0.00
Buckingham Palace SW1A 0AA
Londonh GB United Kingdom,
John Smith, 123412341324

Shipping cost
total with
discounts

All promotions
applied to this
order

Payment

Payment method Payment via courier

Place order

← Shipment method

[Profile](#) | [View shopping cart](#) | [Order guide](#) | [Order history](#) | [FAQ](#) | [Return policy](#) | [Site map](#) | [Call center](#)

Note that customer promotion is specific to customer profile tagging operation. These promotions are designed for very complex customer data specific conditions that create a business decision on the customer segment - a **tag**, which can represent purchasing habit (e.g. number of orders), geo location (e.g. current billing address), profile properties (e.g. what type of account they have? have they signed up for newsletter?). The customer promotion evaluations occurs periodically in Admin (as a cron job) and if condition is satisfied a tag is added to customer profile. This allows to do all the data heavy lifting away from storefront servers. Then item, order and shipping promotions can use customer profile tag to perform lightweight conditions such as "if customer has tag X then they are eligible for promotion Y".

Best deal strategy and combined discounts

The promotion calculation strategy attempts to provide **best deal strategy** (in the same way as price resolution) by determining which of the currently active promotions will yield greatest overall discount.

By default each promotion is treated in isolation as the most typical case is to give a single promotion for a given condition. For example if there are two promotions A and B that give 10% and 5% discount respectively, A will be applied during cart calculation process since 10% is greater than 5%.

In some cases business rules require multiple promotions to provide combined effect. In this case promotions that can be combined should be flagged using **Can be combined** property. Combinable promotions will be assessed to test the overall discount they give when enforcing **best deal strategy**. For example if there are three promotions A, B and C that give discounts 10%, 5% and 7% respectively and A and B are marked as combinable then AB will give cumulative discount of 14.5% ($10\% + 90\% \times 5\%$) which is greater than 7% so A and B promotions are applied. If however A, B and C that give discount 10%, 5% and 15% then C will be applied since cumulative discount of AB is only 14.5%.

Promotion calculation process

Every time shopping can is recalculated the following processes are performed:

Phase	Process
1	All gifts and promotions are removed from cart
2	For each item in cart currently active item level promotions are evaluated and the best value promotion(s) applied against sale price
3	Sub total for cart items is calculated
4	Order level promotions are evaluated against sub total object from phase 3
5	Order total is calculated
6	Shipping level promotions are evaluated against order total from phase 5
7	Final order totals are evaluated

Note that during phases 2, 4 and 6 the promotions are applied sequentially. This means that discounts are not simply calculated in isolation but rather each promotion applies virtual discount and gradually decreases the price when there is a combination of several applicable promotions. This concept is best explained by few examples.

Example 1: non combinable

Rank	Promotion	Can be combined	Action
1	A		3%
2	B		5 off
3	C		5%

If an item in cart costs 100 then the following calculations will apply

```
Discount value for A = 3%
Discount value for B = 5/100 = 5%
Discount value for C = 5%

B and C are highest at 5%

B is chosen as its rank is higher than C

Item price = 100.00 - 5% = 95.00
```

Example 2: non combinable

Rank	Promotion	Can be combined	Action
1	A		3%
2	B		5 off
3	C		5%



If an item in cart costs 150 then the following calculations will apply

```
Discount value for A = 3%
Discount value for B = 5/150 = 3.33%
Discount value for C = 5%

C is highest at 5% and is chosen

Item price = 100 - 5% = 142.50
```

Example 3: combinable

Rank	Promotion	Can be combined	Action
1	A		3%
2	B		5 off
3	C		5%

If an item in cart costs 150 then the following calculations will apply

```
Discount value for AB (A part) = 3%
Because we have combination of discounts we subtract 3% before next
assessment, thus price = 150 - 3% = 145.50
Discount value for AB (B part) = 5/145.50 = 3.44%
Discount value for AB = 3% + 3.44% = 6.44%
Discount value for C = 5%

AB has highest value at 6.44% and is chosen

Item price = 100 - 5% - 5 = 140.50 (overall discount is 6.33%)
```

Hopefully these example have provided some understanding at inner workings of the cumulative discounts and made the reader aware of the peculiarities of the process.

Eligibility conditions

Eligibility condition is a boolean expression (i.e. expression that evaluates either to true or false) which represents the qualifying criteria for promotion. Default promotion expression engine is Groovy (a java library).

Promotion engine is just another service in the platform with default implementation written to support Groovy as expression language. However this implementation can be replaced to use any expression language of your choice. If you need advice on custom promotion engine implementations please contact us using [contact form](#)

Since the expression written in Groovy here are some not so obvious things:

1	Equals operator is "==" (double equals) or .equals()	<pre>// A equals B A == B A.equals(B)</pre>
2	The condition is a script and can have multiline Groovy code as long as the last line evaluates to true or false	<pre>// Check item SKU is one in the list of promo SKU codes def list = ['PROMOSKU001', 'PROMOSKU002', 'PROMOSKU003']; list.contains(shoppingCartItem.productSkuCode)</pre>
3	There are some pred efined variables that can be used in expression	<pre>// customerTags is variable containing list of tags from Customer profile customerTags.contains('bigspender')</pre>

Using variables and Groovy syntax any condition can be written to represent promotion rule with conditions resembling a natural English language sentences.

Available variables for eligibility condition

Variable	Type	Description	Item Promo	Order Promo	Shipping Promo
actionContext	String	Promotion context as specified by the user	✓	✓	✓
promotion	Promotion ¹	Promotion domain object	✓	✓	✓
registered	Boolean	Flag to check if current user is logged in. true means registered logged in customer, false means anonymous customer	✓	✓	✓
customer	Customer ¹	Customer domain object	✓	✓	✓
customerTags	List of String	Customer profile tags. Used for customer segmentation	✓	✓	✓
shoppingCart	ShoppingCart ¹	ShoppingCart object. ⚠ Total of the cart must not be used as it does not contain correct information	✓	✓	✓
shoppingCartItem	CartItem ¹	Current cart item being assessed for promotion eligibility	✓	✗	✗
shoppingCartItemTotal	Total ¹	Cart items total, which contains correct cart items total amount including item level promotions	✗	✓	✗
shoppingCartOrderTotal	Total ¹	Order sub total, which contains correct cart items total amount including item level promotions and order level promotions	✗	✗	✓

¹ Links for specified variable types lead to their java definitions, which means is that you can reference any object's properties in the expression. For example to reference SKU code in a particular shopping cart the following syntax can be used **shoppingCartItem.productSkuCode** which corresponds to `CartItem.getProductSkuCode()` property or to reference first name of the customer use **customer.firstname** etc. To simplify: use variable name then full stop to indicate property then property name, which is method name (see Method Summary in javadocs) without "get" prefix and without brackets.

Although this sounds complicated at first, once you get the hang of the syntax it becomes very easy to create very complex expressions that would be quite difficult to write using GUI editor.

If you are struggling with Groovy syntax our professional team can create GUI interfaces for promotion conditions that will suit your business. Use our [contact form](#) to find out more.

Eligibility conditions examples

Here are some typical conditions examples that should improve understanding of the syntax:

At item level a promotion might target specific SKU to give extra discounts. Targeted SKU can be written as list denoted by square brackets with comma separated elements inside. Each code must be enclosed in single quotes to denote that these are actual SKU codes (and not variables or keywords). List has **.contains()** function to check if element exists. The parameter to this function will be currently evaluated cart item SKU code.

Check item SKU is either 'CC_TEST4', 'CC_TEST5'

```
[ 'CC_TEST4', 'CC_TEST5' ].contains(shoppingCartItem.productSkuCode)
```

In some cases condition is irrelevant. Using **true** keyword makes condition to always evaluate to true. This is useful for general coupon promotions such as "10% off on any purchase" where only coupon code validation is required.

Always true

```
true
```

Customer segmentation can be achieved through customer profile tagging. Profile tags are provided by **customerTags** variable, which is a list that can be assessed for presence of a tag, say 'frequentbuyer', in the same way as SKU codes were checked in the example above. Note that frequentbuyer is enclosed in single quotes as this is the actual value and not a variable.

Customer has tag 'frequentbuyer'

```
customerTags.contains('frequentbuyer')
```

Order level promotions may use total variables, which are calculated after cart items promotions have been evaluated. Mathematical signs 'greater', 'less' or 'equals' can be used to create various conditions. In example below order items sub total with item level discounts should be greater than 200. Note that combination of signs can be used as well, such as 'greater or equal to' **>=** or 'less than or equal to' **<=**.

Cart items subtotal including item level promotions is more than 200

```
shoppingCartItemTotal.priceSubTotal > 200.00
```

At shipping level final order total, which also includes the order level promotions, can be assessed much in the same way as cart items sub total.

Order subtotal including item level and order level promotions is more than 200

```
shoppingCartOrderTotal.subTotal > 200.00
```

In some cases it is required to formulate customer specific condition, which can be accomplished via **customer** variable.

Discount for Bob's

```
customer.firstname == 'Bob'
```

Since customer variable references all customer data it is also possible to go deeper and access orders information. Note that these are **expensive calculations as they pull all customer data from the database**. In order to make the conditions lighter complex conditions that are customer specific must be of **tagging promotions** type, as these conditions are evaluated in Admin as background task and do not interfere with storefront. For example condition below checks that the customer has at least 2 orders so the tagging action could be to give 'frequentbuyer' tag. Then item, order and shipping level promotions can assess if 'frequentbuyer' tag is present as demonstrated in an example above.

Discount for customers who have 2 or more orders

```
customer.orders.size() >= 2
```

Another more complicated customer specific example is checking current default shipping address of the customer. This could also be used for tagging promotion, say 'ukcustomer'. Then this tag can be referenced in item, order and shipping promotions.











Discount for customers living in UK

```
def address = customer.getDefaultAddress('S');  
address != null && address.countryCode == 'GB'
```

Promotion Actions and Action Context

Each promotion has an action, which defines the operation to be performed that range from applying a discount to creating a gift line in cart to amending customer profile for customer segmentation purposes.

The following table shows actions supported by promotion types stated above:

Action	Context	Description	Item	Order	Shipping	Customer
Fixed Amount Off	Number representing amount. Currency is deduced from promotion currency.	Fixed amount off	Applied to item price. E.g. if sale price is 50 and amount off is 10 then final price would be 40 per unit	Applied to order subtotal. E.g. if subtotal (i.e. sum of all amounts for each cart item) is 1000 and amount off is 50 then final subtotal amount is 950	Applied to shipping cost. E.g. if shipping is 10 and amount off is 5 then shipping cost due is 5	
Percent off	Number representing discount percentage (0-100)	Percent off the price, total or cost in proportion to sale price	Applied to sale price of the item. E.g. if sale price is 50 and percent off is 10 then final price would be 45 per unit	Applied to order subtotal. E.g. if subtotal (i.e. sum of all amounts for each cart item) is 1000 and percent off is 10 then final subtotal amount is 900	Applied to shipping cost. E.g. if shipping is 10 and percent off is 5 then shipping cost due is 9.5	
Percent discount (non sale) ¹	Number representing discount percentage (0-100)	Percent off the price, total or cost in proportion to list price	Applied to list price of the item. E.g. if list price is 50 and percent off is 10 then final price would be 45 per unit	Applied to order subtotal (sum of all amounts for each cart items that do not have sale price).		
Gift	SKU code representing gift with optional multiplier parameter	Gift product added free to cart	Gift(s) are added to cart	Gift(s) are added to cart		
Tagging	Tag to be added to customer profile					Used by cron job to apply tags to customer profiles, which later on can be used in promotion eligibility conditions

¹Non sale percent discount action is used to prevent double discounts on items that are on sale in price list.

Fixed amount off

Fixed amount off is an amount in currency defined by the promotion configuration that represents total discount to be applied. Note that if value off is greater than actual purchase amount then the amount off becomes purchase amount thus there will never be a negative value.

At item level the amount is applied to price. At order level the amount is applied to cart items total.

Example 1: item promotion "50 EUR off ASUS ME181C-A1, MeMO Pad".

Promotion configuration would have the following parameters:

Parameter	Value	Memo
Promotion type	Item	
Currency	EUR	
Condition	'ME181C-A1-BK'.equals(shoppingCartItem.productSkuCode)	
Action	Fixed Amount off	
Action Context	50	EUR is inferred from currency set on promotion, thus this is 50EUR

Discounts calculation:

Sale Price	Quantity	Discount	Final price	Total	Memo
45EUR	1	45EUR	0EUR	0EUR	Discount cannot be greater than price
150EUR	1	50EUR	100EUR	100EUR	
150EUR	2	50EUR	100EUR	200EUR	

Example 2: order promotion "10 EUR off orders placed in August 2016".

Promotion configuration would have the following parameters:

Parameter	Value	Memo
Promotion type	Order	
Start date	01-Aug-2016	
End date	31-Aug-2016	
Currency	EUR	
Condition	true	
Action	Fixed Amount off	
Action Context	10	EUR is inferred from currency set on promotion, thus this is 10EUR

Discounts calculation:

Cart items total	Discount	Total	Memo
5EUR	5EUR	0EUR	Discount cannot be greater than price
100EUR	10EUR	90EUR	

Example 3: shipping promotion "5 EUR off shipping on orders above 100EUR".

Promotion configuration would have the following parameters:

Parameter	Value	Memo
Promotion type	Shipping	
Currency	EUR	
Condition	shoppingCartOrderTotal.subTotal >= 100.00	
Action	Fixed Amount off	
Action Context	5	EUR is inferred from currency set on promotion, thus this is 5EUR

Discounts calculation:

Order total	Number of deliveries	Delivery charge	Discount	Total	Memo
50EUR	1	10EUR	0EUR	60EUR	
150EUR	1	10EUR	5EUR	155EUR	
150EUR	2	10EUR	5EUR	160EUR	

Percent Off

Percent off is calculated from current price of the item to produce the final price to be paid.

Example 1: item promotion "10% off items over 100EUR".

Promotion configuration would have the following parameters:

Parameter	Value	Memo
Promotion type	Item	
Currency	EUR	
Condition	shoppingCartItem.price > 100.00	
Action	Percent Off	
Action Context	10	% is inferred from the action type, thus this is 10%

Discounts calculation:

Sale Price	Quantity	Discount	Final price	Total	Memo
45EUR	1	45EUR	4.50EUR	40.50EUR	
45EUR	2	45EUR	4.50EUR	81.00EUR	

Example 2: order promotion "10% off orders placed by frequent buyers".

Promotion configuration would have the following parameters:

Parameter	Value	Memo
Promotion type	Order	
Currency	EUR	
Condition	customerTags.contains('frequentbuyer')	
Action	Percent Off	
Action Context	10	% is inferred from the action type, thus this is 10%

Discounts calculation:

Cart items total	Discount	Total	Memo
------------------	----------	-------	------

5EUR	0.50EUR	4.50EUR	
100EUR	10.00EUR	90.00EUR	

Example 3: shipping promotion "Free shipping on orders above 100EUR".

Promotion configuration would have the following parameters:

Parameter	Value	Memo
Promotion type	Shipping	
Currency	EUR	
Condition	shoppingCartOrderTotal.subTotal >= 100.00	
Action	Percent Off	
Action Context	100	% is inferred from the action type, thus this is 100% i.e. free

Discounts calculation:

Order total	Number of deliveries	Delivery charge	Discount	Total	Memo
50EUR	1	10EUR	10EUR	50EUR	
150EUR	1	10EUR	10EUR	150EUR	
150EUR	2	10EUR	10EUR	150EUR	

Percent Off (non sale)

Percent off is calculated from list price of the item to produce the final price to be paid and is applied only if new discounted value is less than sale price. This means that if item had sale price which is better than discount given by this promotion then sale price from price list is used (customer best deal strategy).

Example 1: item promotion "10% off items over 100EUR".

Promotion configuration would have the following parameters:

Parameter	Value	Memo
Promotion type	Item	
Currency	EUR	
Condition	shoppingCartItem.price > 100.00	
Action	Percent Off (non sale)	
Action Context	10	% is inferred from the action type, thus this is 10%

Discounts calculation:

List Price	Sale Price	Quantity	Discount	Final price	Total	Memo
45EUR	40EUR	1	45EUR	4.50EUR	40.00EUR	Sale price is used because 4.50 produced by promotion is less than sale discount 5.00EUR
45EUR	42EUR	1	45EUR	4.50EUR	40.50EUR	Promotion price is used because 4.50 produced by promotion is less than sale discount 3.00EUR
45EUR	42EUR	2	45EUR	4.50EUR	81.00EUR	

Example 2: order promotion "10% off orders placed by frequent buyers".

Promotion configuration would have the following parameters:

Parameter	Value	Memo
Promotion type	Order	

Currency	EUR	
Condition	customerTags.contains('frequentbuyer')	
Action	Percent Off (non sale)	
Action Context	10	% is inferred from the action type, thus this is 10%

Discounts calculation:

List price cart items total	Cart items total	Discount	Total	Memo
6EUR	5EUR	0.60EUR	4.40EUR	The discount is calculated using list price total
100EUR	100EUR	5.00EUR	95.00EUR	The discount is calculated using list price total

Gift

Gifts are items automatically added to cart with zero final price. Since every type of promotion is subject to best deal strategy, gifts promotional value is calculated from gift product price. The quantity of gifts to be added to the cart depends on the **multiplier** and differs slightly depending on the promotion level (item or order).

The gift action context has the following syntax:

```
[SKU CODE] (=|~) [MULTIPLIER]
```

The multiplier part is optional and it allows to control how many gifts are added. At **item level multiplier** refers to **quantity** of the currently evaluated cart item. At **order level multiplier** refers to **cart total** amount.

Example 1: Item level, no multiplier

At item level no multiplier denotes that for every whole SKU quantity ordered a gift will be added (**Buy 1 get 1 gift**)

Promotion configuration would have the following parameters:

Parameter	Value	Memo
Promotion type	Item	
Action	Gift	
Action Context	ABC001	Simple SKU code, no multiplier

Gifts calculation:

Ordered SKU Quantity	Gifts Added	Memo
1	1	
2	2	
N	N	

Example 2: Item level, exact multiplier

At item level exact multiplier denotes that for every whole multiplier's worth SKU quantity ordered a gift will be added (**Buy X get 1 gift**).

Promotion configuration would have the following parameters:

Parameter	Value	Memo
Promotion type	Item	
Action	Gift	
Action Context	ABC001=2	SKU code with exact multiplier of 2, i.e. buy 2 get 1 gift

Gifts calculation:

Ordered SKU Quantity	Gifts Added	Memo
1	0	
2	1	
3	1	
4	2	
N	ROUND DOWN (N / multiplier)	

Example 3: Item level, approximate multiplier

At item level exact multiplier denotes that for every whole multiplier's worth SKU quantity ordered a gift will be added (**Get 1 gift if you buy at least X**).

Promotion configuration would have the following parameters:

Parameter	Value	Memo
Promotion type	Item	
Action	Gift	
Action Context	ABC001~2	SKU code with exact multiplier of 2, i.e. buy 2 get 1 gift

Gifts calculation:

Ordered SKU Quantity	Gifts Added	Memo
1	0	
2	1	
3	2	
4	2	
5	3	
N	ROUND UP (N / multiplier)	

Example 4: Order level, no multiplier

At order level this configuration means single gift no matter what the sub total is.

Promotion configuration would have the following parameters:

Parameter	Value	Memo
Promotion type	Order	
Currency	EUR	
Action	Gift	
Action Context	ABC001	One gift for order

Gifts calculation:

Order Sub Total	Gifts Added	Memo
25 EUR	1	
50 EUR	1	
75 EUR	1	
100 EUR	1	

X	1	
---	---	--

Example 5: Order level, exact multiplier

At order level this configuration means one gift per multiplier amount.

Promotion configuration would have the following parameters:

Parameter	Value	Memo
Promotion type	Order	
Currency	EUR	
Action	Gift	
Action Context	ABC001=50	One gift for order each 50 EUR

Gifts calculation:

Order Sub Total	Gifts Added	Memo
25 EUR	0	
50 EUR	1	
75 EUR	1	
100 EUR	2	
X	ROUND DOWN (X / multiplier)	

Example 6: Order level, approximate multiplier

At order level this configuration means one gift per multiplier amount.

Promotion configuration would have the following parameters:

Parameter	Value	Memo
Promotion type	Order	
Currency	EUR	
Action	Gift	
Action Context	ABC001~50	One gift for order each 50 EUR

Gifts calculation:

Order Sub Total	Gifts Added	Memo
25 EUR	1	With approximation multiplier event 0.01 will trigger gift, therefore condition has to carefully formulated (e.g. order total greater than X)
50 EUR	1	
75 EUR	2	
100 EUR	2	
X	ROUND UP (X / multiplier)	

Customer segmentation

In some cases promotion must apply to a specific **segment of people**, such as 'frequent buyers', 'UK buyers', 'Customers registered in August' or 'Newsletter subscribers'. It is easy to see that each **segment can define very different groups of people** with very diverse qualifying criteria, where **each person may belong to none, few or all segments**. In order to accomplish this it is recommended to use customer tags, where each **tag represents a segment** that is easily understood by business user. Once customer profile tagging has been performed item, order and shipping level promotion can have simple conditions that check if customer belongs to a certain segment:

Example A

```
customerTags.contains('frequentbuyer')
```

Each **customer profile can contain zero or more tags** associated with them (space separated). The **tags can be manually managed** via customer [profile management](#) but it may be troublesome if there are many accounts. To solve this problem the platform defines **customer level promotions to tag profiles automatically**. Customer level promotion allows to select customers that qualify to be put into a segment. Then customer profile can be automatically tagged using tag specified in promotion action context. The **customer promotions are evaluated in YUM by a recurring job** that is run in the background. Therefore all that is necessary is to **define customer level promotions and enable them** and the platform will automatically start tagging customer profiles.

Defining customer promotion condition is very similar to any other promotion. For example if criteria for frequent buyer is to have more than 10 orders the following condition can be formulated:

Example B

```
customer.orders.size() >= 10
```

And action context would have *frequentbuyer* tag as a single word.

After having setup promotions in example A and B the following steps will happen:

1. Customer tagging job will run periodically on YUM node examining if there are customer accounts that meet the promotion criteria in example B. Those that meet the criteria will be tagged with 'frequentbuyer' tag.
2. When customer browsing storefront performs an action that alters the shopping cart re-calculation process will be triggered including promotion evaluation. Active promotion with condition in example A will be evaluated and if current customer profile was tagged the promotion will be applied.

Because all promotion conditions are in fact Groovy scripts very complex rules can be crafted in customer promotion condition with ease. This is where the promotion engine truly shines. Another benefit of this separation is that all complexity related to customer segmentation can be isolated into customer level promotions, which could be managed by skilled personnel with ease, leaving the day to day promotions based on tag checking very simple to craft by regular business users. This approach is straightforward, clean and simple, and flexible to **avoid code changes and limitation** to complex business marketing strategies.

We recommend above mechanism for all customer specific promotions.

Promotion management

Promotions are completely autonomous objects. They can be [imported](#) or entered through YUM as long as there is a reference to shop code and currency. Each promotion is bound to this pair to avoid confusion and simplify the process of discount calculation for shopping cart. This also gives flexibility to have separate promotion strategies defined for different currencies.

Basic information

The most important properties on the promotion summary tab are:

- **Promotion code** (visible to customer in [default theme](#)) which cannot be changed after the promotion is saved
- **Shop code** for the [shop](#) where this promotion will be used which cannot be changed after the promotion is saved
- **Currency** for discount calculations, which cannot be changed after the promotion is saved
- **Valid from/to** time period for which promotion is active for **if it is enabled**

Tags are used purely to allow categorisation of promotions and make searching easier. These can be anything you want (space separated list of

words). Some ideas would be names for your promotion campaign (e.g. denimhalfprice2015) or seasonal sale (e.g. xmas2016).

Promotions have to be explicitly enabled to be active. Active promotions cannot be changed. Therefore if modifications are required promotion has to be disabled first.

Current Activity

! This promotion is enabled, disable it to make changes

Code

SHOP10EURITEMQTY10P

Shop code

SHOP10

Currency

EUR

Rank

0

Valid from

GMT yyyy-MM-dd HH:mm:ss

Valid to

GMT yyyy-MM-dd HH:mm:ss

Tags

Tags

Promotion on/off switch and active period

Promotion on/off switch
and active period

Name	default value
	10% off when buying 5 or more items

Information about the promotion	
en	10% off when buying 5 or more items
ru	Скидка 10% при покупке свыше 5 единиц
uk	Знижка 10% при покупці більше 5 одиниць
Language	Value
Description	
default value	
Language	Value

Other properties refer to localised name and description, which can be used to communicate the information regarding this promotion to the customers. Alternatively business user can list this information on a promotion [micro site](#) where currently active promotions, their conditions and benefits can be clearly articulated to the customers.

Promotion condition

Condition tab displays all information regarding the promotion criteria and configuration for promotion action.

The promotion calculation is quite complex process in its own right and becomes exponentially difficult when trying to assess its effects during auditing of orders. To simplify this task updating existing promotion parameters is limited to disabled promotions only (i.e. use enable/disable button).

If promotion needs to be changed we strongly recommend to use "copy" facility to create a different promotion and disable old one. This will help immensely when auditing the discounts already applied to orders, especially in the long run.

Type refers to one of the supported types and **Action** defined instructions if conditions are satisfied. (Note that action selection is influenced by type)

Text area at the top contains the **condition**, which has been discussed in the overview section. Right below the condition **action context parameter** is located, which is labeled after the action type defined on the "summary" tab.

In order to enable coupons support **triggered by coupon** flag must be checked, which will enable controls on the "Coupons" tab.

Can be combined flag determines if this promotion can be part of cumulative discount, which has been discussed in the overview section above.

Main
Localization
Condition

Active Tab

☒ Can be combined
☐ Coupon triggered

Type

Product

Action

Percentage off

Percentage off

10

Action context parameter

Rule

shoppingCartItem.qty > 5

Condition expression

Coupon codes management

Coupon management only applies to promotions that are **triggered by coupon** (see "Condition" tab). Any promotion can become coupon triggered at any time.

Coupon codes allow business user to specify promotions triggered by user entering a special code to their shopping cart. This gives a whole new dimension to promotions.

Simple use case is to create a promotion (say 5% discount) that marketer would like to issue to client on per order basis (say customer is unhappy, or asks for additional discount if they have large order history). Therefore we can create a 5% order discount promotion, which is triggered by coupon code. Then we can add a single use coupon code and tell it to the customer. When they enter the code they will get a one off 5% discount.

Another use case would be distribution of promotion flyers or coupons in newspapers, whereby a special single use or multi use code is specified that can trigger discounts.

The platform supports **single use** and **multi use coupon codes**, as well as **coupon codes with limited use** (e.g. first 1000 customers). The validity of the coupon is determined by the promotion (see enabled flag and active to/from configuration on the "summary" tab).

Promotions / YesCart shop (EUR) / 10% off with coupon

?

T

\$

▶

⏻

+

📄

📝

✖

↺

↻

📁

Code	Name	Configuration	Tags	Valid from	Valid to	Enabled
SHOP10EURITEMQTY10P	10% off when buying 5 or more items	🔗 ⚙️ 📄				🟢
SHOP10EURCOUPON10	10% off with coupon	🔗 ⚙️ 📄 📝				🟢
SHOP10EURITEMQTY15P	15% off when buying 10 or more items	🔗 ⚙️ 📄				🟢

The screenshot shows the 'Coupons' configuration page in Magento 2. The 'Condition' tab is selected, and a red dashed box highlights the 'Condition' tab and the 'Coupon triggered' dropdown. The 'Coupon triggered' dropdown is set to 'On checkout'. The 'Can be combined' checkbox is unchecked. The 'Type' dropdown is set to 'Order', and the 'Action' dropdown is set to 'Percentage off'. The 'Percentage off' value is set to 10.

The screenshot shows the 'Coupons' tab selected in the 'Condition' section. A red dashed box highlights the 'Coupons' tab, and a red dashed arrow points from it to a red dashed box around the '+' icon in the top right corner of the table. The table has the following data:

Code	Usage limit	Usage limit (per customer)	Usage count
SHOP10EURCOUPON10	100		1

Coupons

To generate random code single use coupons do not enter code and enter the number of coupons to generate in usage limit, to generate multiple use coupon enter code and usage limit to specify how many times this coupon can be used

Code

Usage limit

Usage limit (per customer)

10

1

Cancel

Generate

Generated coupon codes can be exported and saved as CSV

```

promotioncouponid,promotionid,code,usageLimit,usageLimitPerCustomer,usageCount,
8,9,SHOP103LDAH5WGH,1,0,0,
9,9,SHOP104V0E1CQR5Z,1,0,0,
6,9,SHOP106O0GSE6SIR,1,0,0,
13,9,SHOP10H1DVKXBKHL,1,0,0,
12,9,SHOP10JYSIUX5X1,1,0,0,
7,9,SHOP10Q9X1JXH0TB,1,0,0,
4,9,SHOP10SGRI5WOMHB,1,0,0,
5,9,SHOP10UWXMZGP6M,1,0,0,
11,9,SHOP10VX6TKGEZTO,1,0,0,
10,9,SHOP10ZKWQARHRGM,1,0,0,

```

Because list of coupon codes can be very large (e.g. thousands of auto generated coupon codes) it is not loaded by default. "show coupons" button must be clicked to view them. However viewing this list when there are multiple single use coupons may be impractical. In this case we recommend to download coupon codes for given promotion as a CSV file using "download coupon list" button.

Coupon code list displays the coupon codes and their configuration (usage limit and usage limit per customer) as well as used count, which is updated whenever order is placed with coupon being used.

Coupon codes with any configuration can be added at any point in time so business user can setup promotion with specific rules and then add coupon codes as necessary on day to day basis.

Consider the following example: newspaper advertisement with coupon code giving 5% discount during summer limited to 1000 customers. Marketing department could issue a multi use coupon with usage limit of 1000. Say it is used up very quickly and other customers (beyond 1000) start complaining. Marketing department could decide re-issuing additional single use personal codes to specific customers. So the promotion is still the same but business users can control additional coupons inflow to keep customers happy.

Another example whereby: 1000 flyers with single use coupon codes giving 5% discount are given out. Say that this limit is used up and business users would like to reissue another 1000 flyers. All that is necessary is to generate additional 1000 coupons and new flyers can be printed.

Final example is: gesture of good will, giving 5EUR discount to customers dissatisfied with their order. This can be a permanent promotion set for 5 EUR fixed amount off. Then if a customer contacts call centre with a complaint (say shipping took too much time) a single use coupon can be

generated to give them 5EUR in their next order.




Above examples are just few use cases, which demonstrate flexibility of the promotion engine. How promotions and coupons are used will depend on individual business marketing strategy.

Lets see how above is achieved through "Coupons" tab.

Multi use coupons

Multi use coupons usually refer to codes which are publicly available (e.g. newspapers or TV advertisement) and represent easy to remember single coupon.

In order to create a multi use coupon values in the coupon code generation form have to be filled in as follows:




Input	Required	Example value	Memo
Code		2015AUG10OF	Simple easy to remember code, but it has to be unique
Usage limit		100	100 means that this code can be used 100 times
Usage limit per customer		1	1 means that each customer will only be able to use this code once

Usage limits must be based on the marketing strategy and can contain any value that business user sees fit for the purpose - the platform just makes sure that those limits are not violated.

Single use coupons

Single use coupons usually refer to discounts given to specific users. They are generated in bulk sometimes in thousands, which is impractical when done manually. The platform allows to auto generate codes. All that is necessary is to specify usage limits and **omit code**. The platform will interpret this request as auto-generation of single use codes in quantity set by usage limit.

In order to create X single use coupons values in the coupon code generation form have to be filled in as follows:

Input	Required	Example value	Memo
Code			Omitting code means that it needs to be auto generated
Usage limit		10	10 means that 10 coupon codes are required
Usage limit per customer		1	1 means that each customer will only be able to use this code once

Mixed use coupons

As already mentioned it is possible to **add any type of coupons to any coupon triggered promotion at any time**.

We recommend the following usage:

1. for one off discount - generate single coupon code with usage limit one. E.g. for user Bob the code may be 'BOB0001'.
2. for large campaigns - auto generate multiple coupon codes by entering number of coupons necessary into usage limit, which will generate that many single use coupons. Then download this list and maintain via [import](#) functionality from then on.

Examples

Fully Annotated Example of Product Promotion

Promotion: 10% discount for registered users buying more than 5 items

Promotions / YesCart shop (EUR) / 10% off when buying 5 or more items

This promotion is enabled, disable it to make changes

Main Localization Condition Coupons

Code

SHOP10EURITEMQTY10P

☐ Can be combined

☒ Coupon triggered

Type

Product

Action

Percentage off

Percentage off

10

Rule

(registered) && (shoppingCartItem.qty >= 5)

Sale Special Offer New Arrivals Demo Contacts Bob Doe Language Eng Currency €

Web site View


YES CART DEMO STORE

Contact form

Search...

Customer must be logged in

Quantity has to be above 5 items

Product	Price	Quantity	Amount
 ME301T-1A045A SKU code ME301T-1A045A Add to wishlist Save for later	2071⁴² Discount 10% SHOP10EURITEMQTY10P	- 6 + Refresh	12428⁵² x

Shopping cart

€ 12328⁵²

Discount 11%
SHOP10EURORD100F

Checkout

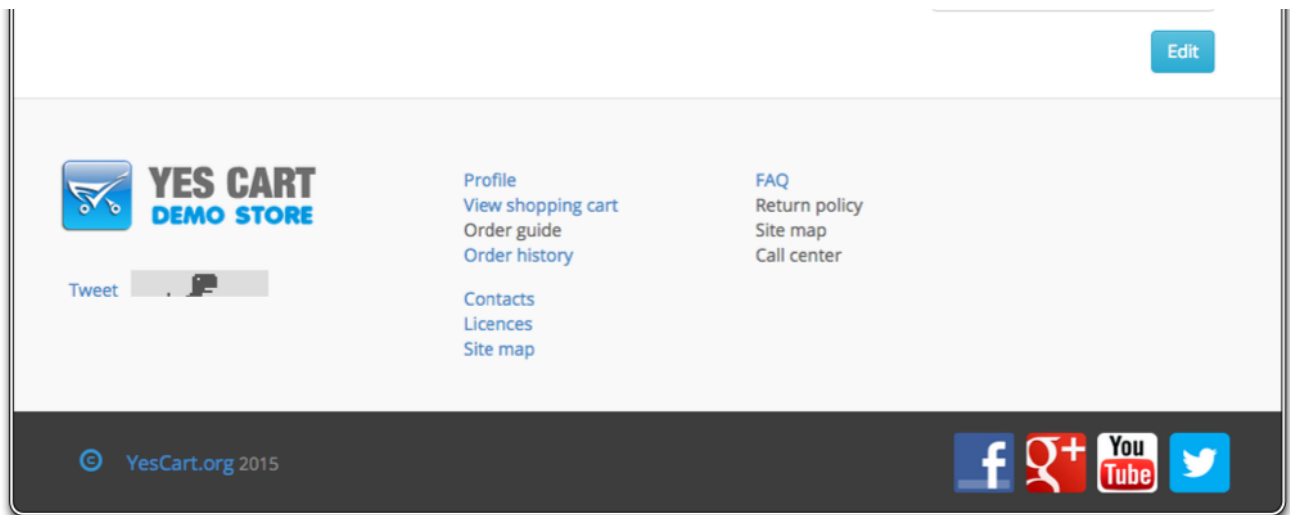
Add Coupons

Enter coupon

Add coupon

Order Message

Add order message



Fully Annotated Example of Product Coupon Promotion

Promotion: 10% discount with coupon code

The condition in this example is simply **true** to make it always available. However **any condition** can be used with coupon promotions. Say if coupon only works if there is a certain item in cart, order order total is checked. Below is just the simplest example.

Promotions / YesCart shop (EUR) / 10% off when buying 5 or more items

This promotion is enabled, disable it to make changes

Main Localization Condition Coupons

Code

SHOP10EURITEMQTY10P

☐ Can be combined

☒ Coupon triggered

Type

Order

Action

Percentage off

Percentage off

10

Rule

Rule

Empty rule is always applicable

Main Localization Condition Coupons

Code

Usage limit

Usage limit (per customer)

Usage count

2015AUG10OFF

100

0 0

Sale Special Offer New Arrivals Demo Contacts

Bob Doe Language Eng Currency €

Web site View

YES CART DEMO STORE

Contact form

Search...

Shopping cart

Product Price Quantity Amount

ME301T-1A045A 2071⁴² - 3 + 6214²⁶

SKU code ME301T-1A045A

Discount 10%

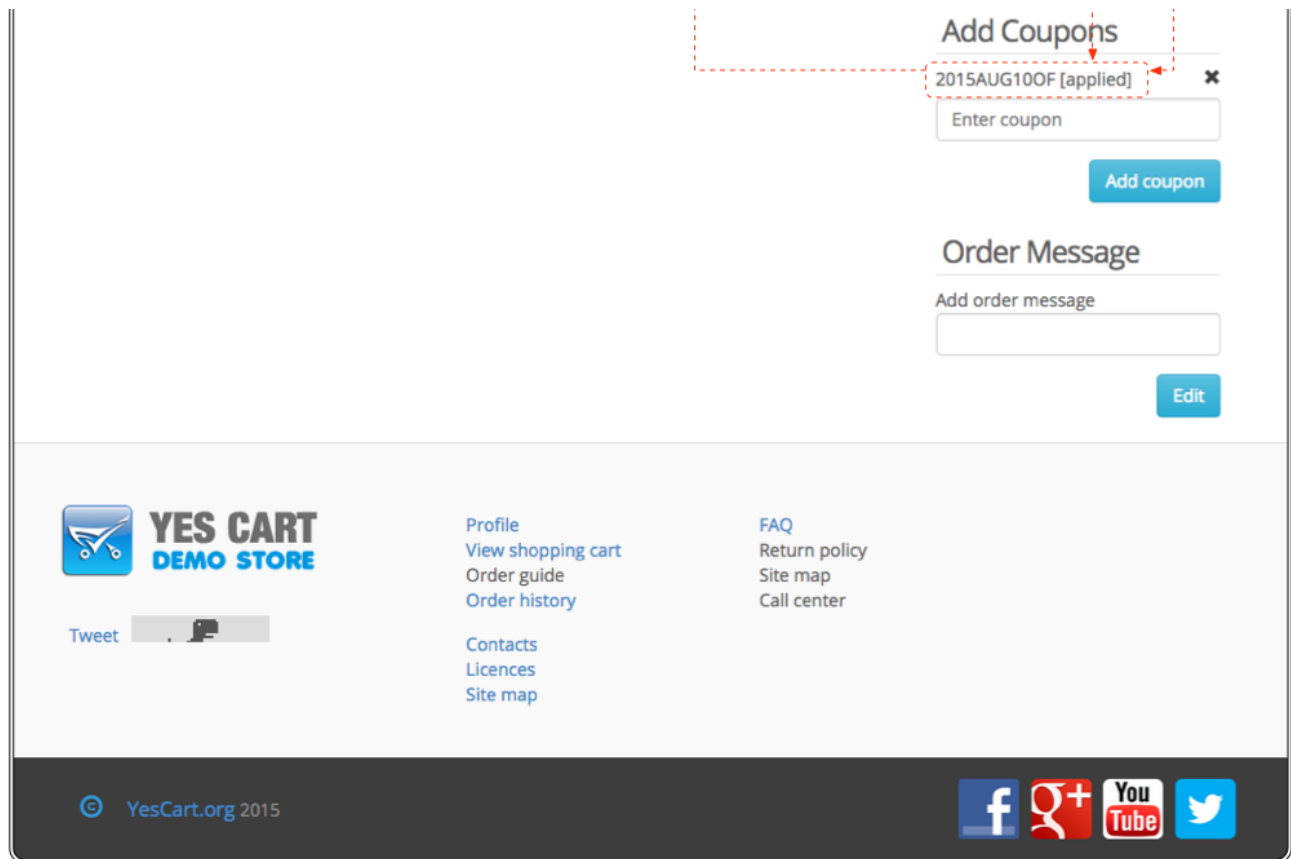
SHOP10EURITEMCPNB10P:2015AUG10OF

€6114²⁶

Discount 11%

SHOP10EURORD100F

Checkout



Validating promotions configuration

[Price tester](#) is a great tool that is available on promotion section of Admin allows validating various scenarios and see how promotions calculations are affected.