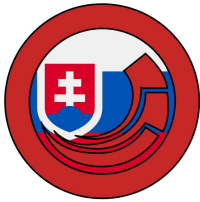# Setting up Commerce Connect

Compiled by Peter Prochazka

Setting up Commerce Connect

Compiled from [official Sitecore Commerce documentation](#).

<u>Compiled by:</u>

Peter Prochazka ([@chorpo](#) / [tothecore.sk](#))

4th July 2018

More Sitecore guidelines and Sitecore / Sitecore Commerce related topics can be found on my blog [tothecore.sk](#).

You can find them also directly [in my github repositories](#).

Table of Contents

Setting up Commerce Connect

The items installed with Commerce Connect

When you install the Commerce Connect package in Sitecore, a number of items are installed.

This topic describes the installed items in the following sections:

- Finding Commerce Connect items in the Content Editor
- Page events, goals, and outcomes
- Personalization rules
- Marketing automation campaigns

Finding Commerce Connect items in the Content Editor

Commerce Connect items are installed in the following folders in the Content Editor:

Item type	Item path
Page events	/sitecore/System/Settings/Analytics/Page Events/Commerce
Goals	/sitecore/System/Marketing Control Panel/Goals/CommerceC
Outcomes	/sitecore/System/Marketing Control Panel/Outcomes
Conditional renderings	/sitecore/System/Settings/Rules/Definitions/Elements/Con Conditional Renderings
Marketing automation plan folder	/sitecore/system/Marketing Control Panel/Automation Plan
Experience Analytics Dimensions	/sitecore/system/Marketing Control Panel/Experience Analytics/Dimensions/Connect
Marketing	/sitecore/system/Settings/Rules/Definitions/Elements/Con

automation conditions	`/sitecore/system/Settings/Rules/Definitions/Elements/Com` `Email`
Templates	`/sitecore/Templates/CommerceConnect`

In addition, a number of Experience Analytics [reports](#) that specifically handle data from Sitecore Commerce Connect are installed with Commerce Connect.

Page events, goals, and outcomes

To access page events, goals, and outcomes included with Commerce Connect, you need to use the API provided with Commerce Connect. For more information, please see section 1.2 of the [Commerce Connect Developer's Guide]().

The table below shows details of the Commerce Connect page events, goals and outcomes.

Note
From Sitecore XP 9.0, when Sitecore saves events, goals and outcomes, Sitecore stores the related personal data in contact facets and assigns the facets a PIISensitive attribute (Personally Identifiable Information) so that the facets can be deleted if the customer exercises their right to be forgotten.

Service layer the item belongs to	Item Type	Item Name	Data Stored with the Item
Cart	Page event	Cart Created	`ExternalId, UserId, CartName, CartStatus`
Cart	Page event	Cart Deleted	`ExternalId (Cart ID), UserId, CartName, CartStatus`
Cart	Page event	Cart Locked	`ExternalId, UserId, CartName, CartStatus`
Cart	Page event	Cart Resumed	`ExternalId, UserId, CartName, CartStatus, PreviousState`
Cart	Page event	Cart Unlocked	`ExternalId, UserId, CartName, CartStatus`
Cart	Page event	Cart Updated	`CustomerId, CartName, ShopName`
Cart	Page event	Lines Added To Cart	`ListOfCartLines (ProductId Quantity, Price)`
Cart	Page	Lines	`ListOfCartLines (ProductId Quantity, Price)`

	event	Removed From Cart	
Cart	Page event	Lines Updated On Cart	`ListOfCartLines (ProductIc` `Quantity, Price)`
Cart	Outcome	Abandoned Cart	`Shopname, Cart ID, Cart ar` `Cart Total`
Catalog	Page event	Visited Category Page	`Catalog Name, Category Nam`
Catalog	Page event	Facet Applied	`Facet name, Applied` `(boolean)`
Catalog	Page event	Product Sorting	`Sort key, Sort direction` `(ASC/DSC)`
Catalog	Page event	Visited Product Details Page	`Product ID, ProductName,` `ParentCategoryName,` `ShopName, ParentCategoryIL`
Customers and Users	Page event	Customer Account Deleted	`CustomerName, ShopName`
Customers and Users	Page event	Customer Account Disabled	`CustomerName, ShopName`
Customers and Users	Page event	Customer Account Enabled	`CustomerName, ShopName`
Customers and Users	Page event	Customer Account Updated	`CustomerName, ShopName`
Customers and Users	Page event	User Account Deleted	`UserName, ShopName`
Customers and Users	Page event	User Account Disabled	`UserName, ShopName`
Customers and	Page	User Account	`UserName, ShopName`

Users	event	Enabled	
Customers and Users	Page event	User Account Updated	`UserName, ShopName`
Customers and Users	Goal	Customer Account Created	`CustomerName, ShopName`
Customers and Users	Goal	User Account Created	`UserName, ShopName`
Globalization	Page event	Culture chosen	`ShopName, Culture ID`
Inventory	Page event	Add To Cart Stock Status	`CartId, UserId, ShopName, ListOfCartLines(ProductId, StockStatus, InStockDate, ShippingDate)`
Inventory	Page event	Back In Stock Subscription	`Product, Email, Location, InterestDate, ShopName`
Inventory	Page event	Back In Stock Unsubscription	`Product, Location, ShopName, VisitorId`
Inventory	Page event	Products Are Back In Stock	`ShopName, ListOfProducts`
Inventory	Page event	Visited Product Stock Status	`ShopName, Product, Location, StockStatus, AvailabilityDate, StockCount`
LoyaltyProgram	Goal	Loyalty Program Joined	`ExternalId, UserId, ShopName, CardNumber`
Orders	Page event	Gift Card Puchase	`Order ID, Gift CardI ID, the order total, the amour used on the gift card and the remaining amount`
Orders	Page event	Loyalty cart Puchase	`Order ID, Order Total, Loyalty CardID, Loyalty program ID (if available), Loyalty points used and th total amount that the`

			loyalty points equals
Orders	Page event	Order Status Changed	`Order ID, Status (string)`
Orders	Page event	Ordered Product Stock Status	`ProductID, ProductName, StockStatus, InStockDate, ShippingDate, PreOrderable (true/false)`
Orders	Page event	Visitor Cancel Order	`CustomerId, ShopName, OrderId, Total`
Orders	Page event	Visitor Viewed Order Details	`CustomerId, ShopName, OrderId, Total amount`
Orders	Page event	Visitor Viewed Order History	`CustomerId, ShopName`
Orders	Page event	Offline Orders Synchronized	`UserId, ShopName, LastOrderSynchronizedDate, ExternalSystem, NumberOfOrdersSynchronized, SynchronizedOrderIDsList`
Orders	Goal	Visitor Order Created	`ExternalId (Order ID), ShopName, Totals, Order`
Orders	Outcome	Product Purchase (Visitor Order Created)	`Shopname, Order ID (External ID), a cut down order (order header + order lines with product ids and names, line total, subtotal and total)`
Pricing	Page event	Currency chosen	`ShopName, Currency ID`
WishLists	Page event	Lines Added To WishList	`ExternalId, UserId, WishListName, ShopName, ListOfCartLines(ProductId, Quantity, Price)`
WishLists	Page event	Lines Removed From WishList	`ExternalId, UserId, WishListName, ShopName, ListOfCartLines(ProductId, Quantity, Price)`
WishLists	Page event	Lines Updated On WishList	`ExternalId, UserId, WishListName, ShopName,`

			ListOfCartLines(ProductId, Quantity, Price)
WishLists	Page event	WishList Created	ExternalId, UserId, WishListName, ShopName
WishLists	Page event	WishList Deleted	ExternalId, UserId, WishListName, ShopName
WishLists	Page event	WishList Emailed	ListOfWishLists(ExternalId, UserId, WishListName, ShopName)
WishLists	Page event	WishList Printed	ExternalId, UserId, WishListName, ShopName
[Part of Platform]	Page event	Search	ShopName, SearchTerm, NumberOfHits

Personalization rules

The table below shows details of the personalization rule conditions that come with Commerce Connect.

Service layer	Item name	Description	Rule parameters
Cart	Total Product Quantity Condition	Tests the current quantity of items in the cart against a specified value.	TotalProductQuantity, compares-to operator
Cart	Cart Total Condition	Tests the current cart total against a specified amount.	CartTotal, compares-to operator
Cart	Specific Product Quantity Condition	Tests the current cart content for the quantity of a specified product against a specified value.	ProductId, ProductQuantity, compares-to operator
Inventory	Current Product Stock Count	Tests the stock count of a product given by the Commerce Context implementation against a specified value. Requires Commerce Context. The default implementation alone throws *Not Implemented* exceptions.	StockCount, compares-to operator
Inventory	Current Product	Tests the stock count of a product	StockCount, Location, compares-to operator

|	Stock Count By Location	provided by Commerce Context against a specified value. Also tests for the product being in a specified location. Requires Commerce Context. The default implementation alone throws *Not Implemented* exceptions.	
Inventory	Current Product Stock Status	Tests the stock status of a product provided by Commerce Context against a specified status. Requires Commerce Context. The default implementation alone throws *Not Implemented* exceptions.	StockStatus, compares-to operator
Inventory	Current Product Stock Status By Location	Tests the stock status of a product provided by Commerce Context against a specified status. Also tests for the product being in a specified location. Requires Commerce Context. The default implementation alone throws *Not*	StockStatus, Location, compares-to operator

		Implemented exceptions.	
Inventory	Specific Product Stock Count	Tests the stock count of a specified product against a specified value.	ProductId, StockCount, compares-to operator
Inventory	Specific Product Stock Count By Location	Tests the stock count of a specified product against a specified value. The condition also tests for the product being in a specified location.	ProductId, StockCount, Location, compares-to operator
Rules	Has Triggered Page Events	Tests the number of occurrences of a specified page event within a specified time range against a specified value. Not recommended for use in production systems; not optimized as it does not use KeyBehaviorCache, which was introduced with 8.1.	PageEventName, MinimalNumberOfTimes, TimeRange

Marketing automation campaigns

Service layer item belongs to	Item type	Item name	Description
Cart	Marketing Automation campaign condition	Abandoned Cart	Follows up on contacts who visit the storefront and leave a shopping cart with products in it. If a contact has a known email address, Sitecore sends a notification email to encourage the contact to finalize the purchase. The marketing automation campaign stores the cart ID in custom values.
Cart	Marketing Automation campaign condition	Where Cart Has Been Manipulated	Checks if any actions have been performed on the cart since the contact was enrolled into the campaign or since last checked. Used in Abandoned Cart campaign.
Cart	Marketing Automation campaign condition	Where Contact Has Email	Checks if an email exists for the contact. Used in Abandoned Cart campaign.
Cart	Marketing Automation campaign condition	Where Contact Cart Is Empty	Checks if the cart with the specified ID is empty or not. Used in Abandoned Cart campaign.

Set up the Storefront Abandoned Cart marketing automation campaign

From version 9.0.2, Commerce Connect contains a default *Storefront Abandoned Cart* [marketing automation campaign](#), which helps you track shopping carts to determine if they are abandoned, and then follow-up on them.
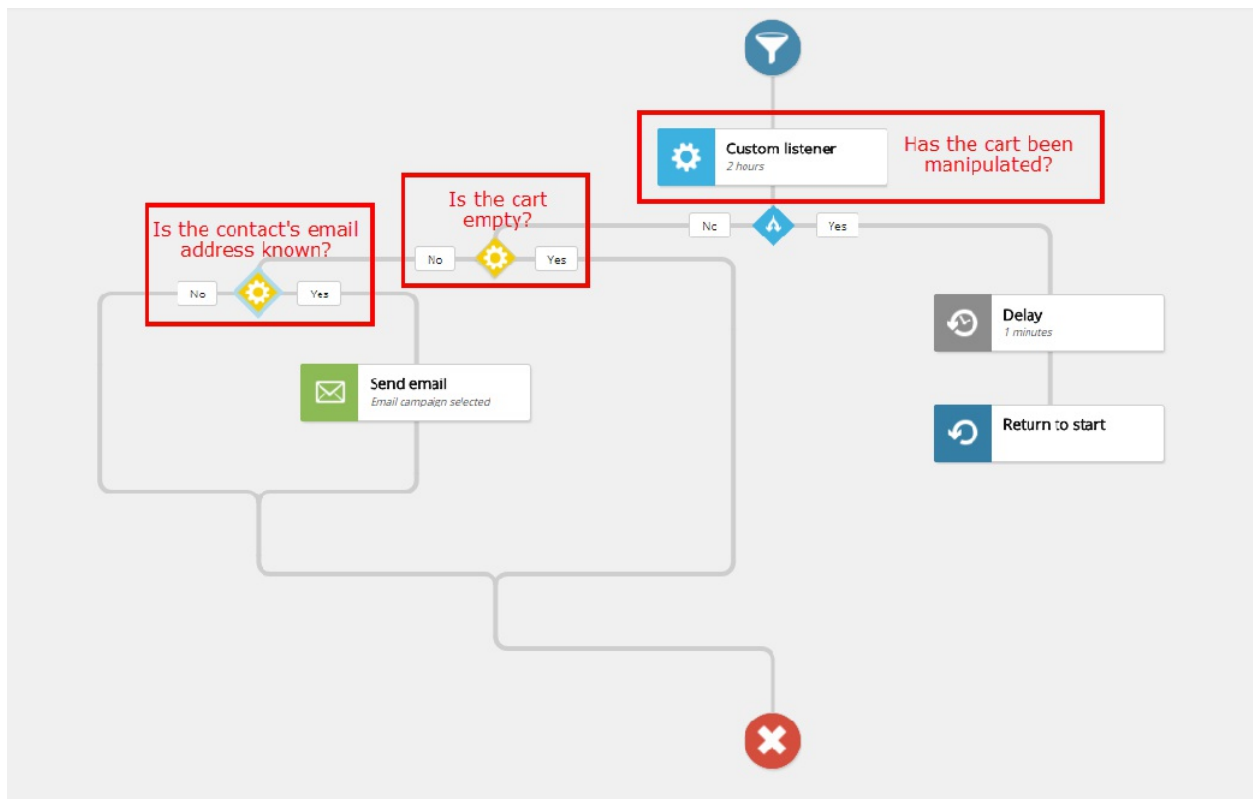
When a new contact arrives to the storefront and a new cart is created, the contact is enrolled into the *Storefront Abandoned Cart* marketing automation campaign.

By default, the campaign works as follows:

- If the cart is not manipulated during a default 2-hour period, the system then checks whether the cart is empty, and whether an email address is known for the contact.

- If the cart is empty or an email address is not known, the contact is removed from the campaign.

- If the cart is not empty and the email address is known, an email message is sent to the contact reminding them that they have abandoned their cart. The intention is to get the contact to return and place an order. After the email message is sent, the contact is removed from the campaign.

When you deploy the SXA Storefront solution, the *Storefront Abandoned Cart* automation campaign is disabled by default.

To set up the *Storefront Abandoned Cart* automation campaign:

1. [Create an email campaign](#) in the Email Experience Manager. You can personalize the email message by inserting tokens for the contact's name and similar information.

2. Assign the email campaign to the Send email action in the marketing automation campaign.

3. To [activate your campaign](#), click Actions ▾, and in the drop-down menu click Activate.

Note

In Commerce Connect 9.0.2, you cannot include information about the content of the cart in the email message due to limitations in the integration of Email Experience Manager with Marketing Automation.