

Klarna Payments

Version 19.1.4



Table of Contents

1.	Summary	1-3
2.	Component Overview.....	2-4
2.1	Functional Overview.....	2-4
2.2	Use Cases.....	2-5
2.3	Limitations, Constraints.....	2-22
2.4	Compatibility	2-23
2.5	Privacy, Payment Data.....	2-23
3.	Implementation Guide	3-28
3.1	Setup	3-28
3.1.1	int_klarna_payments.....	3-28
3.1.2	int_klarna_payments_pipelines	3-29
3.1.3	int_klarna_payments_controllers	3-29
3.2	Configuration.....	3-30
3.2.1	Metadata Import	3-30
3.2.2	Cartridge Path.....	3-31
3.2.3	Klarna Payments Configurations	3-32
3.2.4	Klarna Payments Logo and Payment Option Name.....	3-34
3.2.5	Services Configuration.....	3-35
3.2.6	Custom attributes.....	3-36
3.3	Custom Code	3-37
3.3.1	Integration efforts	3-37
3.3.2	Templates modifications	3-38
3.3.3	Pipeline modifications	3-44
3.3.3.1	COBilling pipeline	3-44
3.3.3.2	COSummary pipeline.....	3-45
3.3.3.3	COPlaceOrder pipeline.....	3-46
3.3.4	Controller modifications.....	3-47
3.3.4.1	COBilling controller.....	3-47
3.3.4.2	COSummary controller.....	3-49
3.3.4.3	OrderModel.js	3-50
4.	Release History.....	4-52

1. Summary

This cartridge enables a Commerce Cloud store to use the Klarna Payments service.

The developer has to install the cartridge and integrate it into the online store following the instructions from this document. Merchant teams are required also to configure the cartridge with the valid merchant credentials and site configurations in Commerce Cloud Business Manager to enable Klarna payments methods in the checkout. It is a requirement that Merchant sign a contract for integration support and production go-live with Klarna.

Klarna offers a test environment, so the integration can be tested before switching to the Klarna production environment. Based on the contract, Klarna shall provide assistance with integration and testing prior to sign-off for go-live.

The integration consists of an archive, which contains the following contents:

- Cartridge called 'int_klarna_payments', and optionally either "int_klarna_payments_pipelines" or "int_klarna_payments_controllers" depending on which architecture is to be employed.
- A site-template archive containing new attributes and settings
- This document
- The integration is based on the Site Genesis demo store provided by Commerce Cloud.

2. Component Overview

2.1 Functional Overview

Klarna payment can be integrated via the Klarna Payment REST API and the JavaScript SDK on the storefront. Klarna Payment enables consumers to choose from the different payment method products offered by Klarna. Multiple Klarna products are available within the categories Pay Now, Pay Later and Pay Over Time. The cartridge integration displays payment options via a widget (iframe) added inline on the billing page, referred to as Klarna widget or just “the widget”. The widget with information about the payment method is displayed to the customer when the individual clicks on the Klarna payment method.

Customers can authorize the payment after reviewing the payment method terms and clicking Place Order button. The Klarna order is confirmed once order is placed and customer re-directed to the confirmation page.

Klarna Payment integration sent customer browser to the location returned in redirect_url as Klarna needs to interact with customer's browser as first party before redirecting to the confirmation page on the store. This ensures a faster checkout experience for returning customers, as the customers device is recognized.

Orders successfully placed with Klarna return a Fraud Status: APPROVED and displayed in Business manager (BM). Klarna orders with Fraud Status PENDING, updates are sent to the notification_url on the merchant Commerce Cloud site within 4-24 hours. The updated Fraud status depends on the fraud screening, (e.g. FRAUD_RISK_ACCEPTED, FRAUD_RISK_REJECTED, FRAUD_RISK_STOPPED) returns a Fraud Status and displayed in BM. The push notification is repeatedly sent until (up-to 24 hours, every 10 mins) the POST request is acknowledged with a 200 response.

In case order is ACCEPTED, order creation in SCC proceeds as usual, in case of REJECTED status, order is failed and in case of PENDING status the 'two phase ordering' approach is used, where the order is first created but only 'placed' after notification for ACCEPTED status is asynchronously received. If neither ACCEPTED, nor REJECTED status is received the order will stay in status 'created' until an action is taken by the merchant. Klarna payment status is saved in a custom attribute with id kpFraudStatus in the PaymentTransaction system object, and can be seen in BM on the order details Payment tab as below:

Sandbox - klarna01
SiteGenesis

Merchant Tools Administration Storefront

Merchant Tools > Ordering > Orders > Order: 00005402(SiteGenesis)

General Attributes Payment Notes History

Payment Information for Order '00005402'

Order Total:	\$265.63
<u>Amount Paid:</u>	\$0.00
Balance Due:	\$265.63
Invoice Number:	00023003
<u>Payment Status:</u>	Not Paid
Payment Method:	Klarna Processor: KLARNA_PAYMENTS Transaction: f909b950-faa5-5b8e-b8f7-b6857aba0a59 Amount: \$265.63
Fraud Status: ACCEPTED	

2.2 Use Cases

1. Create an order in SCC using Klarna Payments as a payment method though Klarna Payments widget on the billing page. All the SCC OOTB checkout functionality remains in place, such as but not limited to: cart updates during checkout, checkout with applied coupon(s) code(s), checkout with applied product level promotion, checkout with applied order level promotion, checkout with applied shipping level promotion, checkout with applied order level promotion with bonus product.

Select 'Klarna' as the payment method on Billing page of checkout process and click the 'CONTINUE TO PLACE ORDER' button:

SELECT PAYMENT METHOD • REQUIRED

Credit Card Pay Pal Bill Me Later

Buy now, pay later
Klarna

Pay later in 4 interest-free installments.

\$14.17	\$14.17	\$14.17	\$14.17
Today	In 2 weeks	In 4 weeks	In 6 weeks

• Buy now. Pay in four \$14.17 installments.
• No interest.

How it works

By continuing, I accept [Klarna Services terms](#) [Privacy Policy](#).
[Pay Later in 4 terms](#) and request electronic communication.

TESTDRIVE

CONTINUE TO PLACE ORDER >

SELECT PAYMENT METHOD • REQUIRED

Credit Card Pay Pal Bill Me Later

Buy now, pay later [Klarna](#)

Slice It [TESTDRIVE](#) [Klarna](#)

Pay a little every month

12 months	from \$17.50/mo.
18 months	from \$11.67/mo.

- Minimum total: \$209.95
- APRs from 0%-19.99%*
- No impact on credit score to check eligibility

* Promotional offer. Subject to a potential \$2 monthly minimum interest charge and [credit terms](#). Accounts are issued by WebBank.

You will see the Order Confirmation page with payment method Klarna and can press the 'PLACE ORDER' button:

commerce cloud

Enter Keyword or item No.

NEW ARRIVALS WOMENS MENS ELECTRONICS TOP SELLERS

STEP 1: Shipping > STEP 2: Billing > STEP 3: Place Order

Help? 800-555-0199

PRODUCT	QTY	TOTAL
Quilted Jacket Item No.: 701642853695 Color royal Size L	1 In Stock	\$110.99
Pull On Pant Item No.: 701642867098 Color Grey Heather Size L	1 In Stock	\$69.00
Zerrick Item No.: 740357357531 Color Black Size 6.5 Width M	1 In Stock	\$99.00

ORDER SUMMARY [Edit](#)

Subtotal	\$278.99
Shipping Ground	\$9.99
Sales Tax	\$14.45
Order Total:	\$303.43

SHIPPING ADDRESS [Edit](#)

Alexander Gayardzhiev Sofia Washington, DC 20230 United States Method: Ground

BILLING ADDRESS [Edit](#)

Alexander Gayardzhiev Sofia Washington, DC 20230 United States

PAYMENT METHOD [Edit](#)

Klarna Amount: \$303.43

« Edit Cart **PLACE ORDER**

The customer's browser is sent to the redirect_url and immediately thereafter shown the Commerce cloud Order Confirmation page

The screenshot shows the Order Number page for order 00000907. At the top, there's a navigation bar with links for NEW ARRIVALS, WOMENS, MENS, ELECTRONICS, and TOP SELLERS. A search bar at the top right contains the placeholder "Enter Keyword or Item No." Below the navigation, a message says "Thank you for your order." followed by "If you have questions about your order, we're happy to take your call (800-555-0199) Monday - Friday, 8AM - 8PM".

Order Number: 00000907

Order Placed: Apr 5, 2017

PAYMENT METHOD:
Klarna
Amount: \$303.43

BILLING ADDRESS:
Alexander Gaydardzhiev
Sofia
Washington, DC 20230
United States
Phone: 3333333333

SHIPMENT NO. 1

SHIPPING STATUS:		SHIPPING TO:	
I Not Shipped		Alexander Gaydardzhiev	Sofia
METHOD:	Ground	Washington, DC 20230	United States
		3333333333	
ITEM	QTY	PRICE	
Quilted Jacket	1	\$110.77	
ITEM NO.: 701642853495			
COLOR: royal			
SIZE: L			
Pull On Pant	1	\$69.00	
ITEM NO.: 701642867098			
COLOR: Grey Heather			
SIZE: L			
Zerrick	1	\$99.00	
ITEM NO.: 740357357531			
COLOR: Black			
SIZE: 6.5			
WIDTH: M			

The newly created order can be inspected in BM

The screenshot shows the Orders page in Business Manager. The top navigation bar includes links for Merchant Tools, Administration, and Storefront. The main content area displays a table of orders. The table has columns for Number, Order Date, Site, Created By, Registration Status, Customer, Email, Total, and Status.

Number	Order Date	Site	Created By	Registration Status	Customer	Email	Total	Status
00000907	4/5/17 12:59:31 pm Etc/UTC	SiteGenesis	Customer	Unregistered	Alexander Gaydardzhiev	alexander.gaydardzhiev@zautre.com	\$303.43	New
00000906	4/5/17 7:39:01 am Etc/UTC	SiteGenesis	Customer	Registered	Yalomova	dentisa.yalomova@zautre.com	\$1,749.26	Open
00000905	4/5/17 7:20:49 am Etc/UTC	SiteGenesis	Customer	Registered	Yalomova	dentisa.yalomova@zautre.com	\$56.69	New

Klarna Payments order id can be inspected in the Attributes tab of the order

The screenshot shows the Attributes tab for order 00000907. The top navigation bar includes links for Merchant Tools, Administration, and Storefront. The main content area shows a single attribute entry: "Klarna Payments Order ID: 9547289d-e2dd-593a-89e5-04439bb54806".

Payment method details can be inspected on the Payment tab of the order, and it should be Klarna

Payment Method:
Name: Processor: KLARNA PAYMENTS
Transaction: 95472954-265593a-9fe4-04d1bb6c4806
Amount: \$303.43

Billing Address:
Alexander Gavrilov
Sofia
Washington DC 20230
us

Order can be further inspected in Klarna Merchant Portal:

EU: eu.portal.klarna.com, US: us.portal.klarna.com, OC: us.portal.klarna.com

#FWZ0CSLP Captured £19.19

Merchant reference 1 00067625	Created 22 Jun 2020, 11:27	Expires 20 Jul 2020, 02:00	Merchant ID K500670
---	--------------------------------------	--------------------------------------	-------------------------------

Customer

Shipping address
UK approved
34 Cavendish Square, Marylebone
33 Cavendish Square
London
W1G 0PW
GB
Tel
+447911123456

Email
tester+uk-2@klarna.com

[Edit shipping address](#)

Billing address

Additional Info

Order lines (2)

Refund	Qty	Item	Reference	Unit price	Discount	Tax	Amount
<input type="checkbox"/>	1	Копринена Tie	79377537003 3	19.19	0.00	20% 3.20	19.19
<input type="checkbox"/>	1	Free Shipping	GBPFree	0.00	0.00	0% 0.00	0.00

PAYMENT DETAILS

Initial Payment Method
Statement
Resend statement

ORDER TOTAL £19.19

Captured	£19.19
Refunded	£0.00
Not Captured	£0.00

CUSTOMER BILLED £19.19

Activity Log

- 26 Jun 2020 a few seconds ago 13:58 Order customer details updated By vivek.poulose via Merchant Portal
- 22 Jun 2020 4 days ago 11:27 Captured: £19.19 Via API
- 11:27 Order placed: £19.19

2. Refuse Klarna Payments as a payment method

Upon selecting 'Klarna' as the payment method on Billing page of checkout process and based on the information provided Klarna Payments can be refused as a payment method.

SELECT PAYMENT METHOD • REQUIRED

- Credit Card Pay Pal Bill Me Later
- Buy now, pay later Buy now, pay later
- Klarna.

Option not available

TESTDRIVE

Unfortunately this option is not available. Please choose a different payment method.

[CONTINUE TO PLACE ORDER >](#)

3. Klarna Payments not available – if Klarna API is not available, Klarna is not presented as a payment option

4. Order accepted after review - Instead of immediately accepting the order, it may happen that Klarna flags this transaction for additional review and accept it later on. The order is then created in the SCC system:

Number	Order Date	Site	Created By	Registration Status	Customer	Email	Total	Status
000000909	4/5/17 1:41:08 pm Etc/UTC	SiteGenesis	Customer	Registered	Gayardzhev	ted.smith+pend-accept-02@example.com	\$102.67	Created
000000908	4/5/17 1:31:25 pm Etc/UTC	SiteGenesis	Customer	Registered	Gayardzhev	ted.smith+pend-accept-02@example.com	\$110.24	Open
000000907	4/5/17 12:59:31 pm Etc/UTC	SiteGenesis	Customer	Unregistered	Alexander Gayardzhev	alexander.gayardzhev@castrc.com	\$103.43	Open

This order is marked with EXPORT_STATUS_NOTEEXPORTED, confirmation status NOTCONFIRMED and NOTPAID.

Shipping Status	Not Shipped	Confirmation Status	Not Confirmed	Export Status	Not Exported
-----------------	-------------	---------------------	---------------	---------------	--------------

Shipping	Line Item	Qty	Product ID	Name	Manufacturer	Tax Rate	Unit Sales Price	Tax Basis	Item Total
		2	701642953695	Cuffed Jacket		5.00 %	\$10.99	\$221.98	\$221.98
		2	701642957698	Pull On Pant		5.00 %	\$69.00	\$138.00	\$138.00
		2	748195736751	Zerlick		5.00 %	\$99.00	\$198.00	\$198.00

Upon receiving approval from Klarna the order is placed:

Merchant Tools - Sandbox - Klarna! Decisions

Merchant Tools Administration Storefront

Details for Order #000000909

Information: Contains 3 line items to 1 shipping location. The total price is \$602.67.

Date Received: 4/17/14 1:08 pm Etc/UTC

Site: SiteGenesis

Created By: Guest

Customer: GarySmithKiev

Customer No.: 00000001

IP Address: 31.13.219.164

Email: test.smith+pend-accept-02@example.com

Phone: 3333333333

Order Status: Open Confirmation Status: Confirmed

Shipping Status: Not Shipped Export Status: Ready for Export

Shipment #000000909	Qty	Product ID	Name	Manufacturer	Tax Rate	Unit Sales Price	Tax Basis	Item Total
2	701642953695	Quilted Jacket			5.00 %	\$110.99	\$221.98	\$221.98
2	701642967098	Pull On Pant			5.00 %	\$60.00	\$118.00	\$118.00
2	740357357531	Zenick			5.00 %	\$99.00	\$198.00	\$198.00

Shipment Details:

- Total: \$115.99
- Total Shipping Cost (001): \$15.99
- Shipping Total: \$15.99
- Tax Total: \$20.70
- Total: \$602.67

[Send Email](#) [Print Order](#)

In Klarna's Merchant Portal (Orders) the order is marked as accepted after review:

Klarna. Merchant Portal

Overview > Order #DR7XBFL5

#DR7XBFL5 Uncaptured **US\$37.09**

Merchant reference 1 00045902 Edit	Merchant reference 2 a2c79d78ca1b673d64f4163557 Edit	Created 26 Jun 2020, 14:18	Expires 24 Jul 2020, 02:00	Merchant ID N100141			
Customer	Order lines (4)						
Shipping address	<input type="button" value="Capture"/> <input type="button" value="Edit order lines"/> <input type="button" value="Cancel order"/>						
US-norm Person 629 N High St Columbus 43215 OH US	<input type="checkbox"/> Qty	Item	Reference	Unit price	Discount	Tax	Amount
	<input type="checkbox"/> 1	\$50 Fixed Products Amount Above 100	\$50FixedProductsAmountAbove100	-475.00	0.00	0% 0.00	-475.00
Tel +491607891234	<input type="checkbox"/> 1	Woven Trimmed Cardigan.	701644391737M	500.00	0.00	0% 0.00	500.00
Email tester+pend-accept-05@klarna.com	<input type="checkbox"/> 1	2-Day Express	002	9.99	0.00	0% 0.00	9.99
Edit shipping address	<input type="checkbox"/> 1	Sales Tax	Sales Tax	2.10	0.00	0% 0.00	2.10
Billing address	<input type="checkbox"/> 1						
Additional Info	PAYMENT DETAILS Initial Payment Method Pay later in parts VISA ****1111				ORDER TOTAL US\$37.09		
					Captured	US\$0.00	
					Refunded	US\$0.00	
					Not Captured	US\$37.09	
					CUSTOMER BILLED	US\$0.00	

Activity Log

26 Jun 2020 6 minutes ago

14:21 Order accepted after review By Klarna

14:18 Order placed: US\$37.09 By Klarna

5. Order rejected after review - Instead of immediately accepting the order, it may happen that Klarna flags this transaction for additional review and reject it later on. The order is then created in the SCC system:

Orders								
<p>You are using our new Search service. This page allows you to search for orders by order number. Select Advanced to use more search options. Select By Number to search by providing a list of order numbers. Order numbers can be separated by either ';' or '-' or space or newline. Entered text is treated as case-sensitive; substring matching is not supported.</p>								
<p>Order Search</p>								
Number	Order Date	Site	Created By	Registration Status	Customer	Email	Total	Status
00001001	4/5/17 10:40:50 am Etc/UTC	SiteGenesis	Customer	Unregistered	Alexander Gaydardzhiev	ted.smith+pend-reject-02@example.com	\$303.43	Created
00000999	4/5/17 1:41:08 pm Etc/UTC	SiteGenesis	Customer	Registered	Gaydardzhiev	ted.smith+pend-accept-02@example.com	\$602.67	Open
00000998	4/5/17 1:31:25 pm Etc/UTC	SiteGenesis	Customer	Registered	Gaydardzhiev	ted.smith+pend-accept-02@example.com	\$110.24	Open

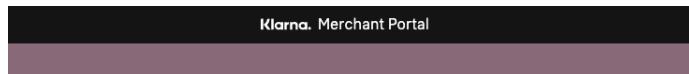
This order is marked with EXPORT_STATUS_NOTEXPORTED, confirmation status NOTCONFIRMED and NOTPAID.

Details for Order '00001001'								
<p>Information: Contains 3 line items to 1 shipping location The total price is \$303.43</p>								
Date Received:	4/5/17 10:40:50 am Etc/UTC	Site:	SiteGenesis	Created By:	Customer	Customer:	Alexander Gaydardzhiev (unregistered customer)	
Customer No.:	n/a	IP Address:	31.13.219.164	Email:	ted.smith+pend-reject-02@example.com	Phone:	3333333333	
Order Status:	Failed	Confirmation Status:	Not Confirmed	Shipment Status:	Not Shipped	Export Status:	Not Exported	
Shipment								
Qty	Product ID	Name	Status	Manufacturer	Tax Rate	Unit Sales Price	Tax Basis	Item Total
1	701642883695	Oufted Jacket	CANCELLED		5.00 %	\$110.99	\$110.99	\$110.99
1	701642867098	Pull On Pant	CANCELLED		5.00 %	\$69.00	\$69.00	\$69.00
1	740357357531	Zenck	CANCELLED		5.00 %	\$99.00	\$99.00	\$99.00
Shipment Shipping Cost: CANCELLED Total Shipping Cost (0.00): \$9.99 Shipping Total: \$9.99 Tax Total: \$14.45 Total: \$303.43								
Send Email Print Order								
<< Back to List								

Upon receiving rejection from Klarna the order is failed:

Orders								
<p>You are using our new Search service. This page allows you to search for orders by order number. Select Advanced to use more search options. Select By Number to search by providing a list of order numbers. Order numbers can be separated by either ';' or '-' or space or newline. Entered text is treated as case-sensitive; substring matching is not supported.</p>								
<p>Order Search</p>								
Number	Order Date	Site	Created By	Registration Status	Customer	Email	Total	Status
00001001	4/5/17 10:40:50 am Etc/UTC	SiteGenesis	Customer	Unregistered	Alexander Gaydardzhiev	ted.smith+pend-reject-02@example.com	\$303.43	Failed
00000999	4/5/17 1:41:08 pm Etc/UTC	SiteGenesis	Customer	Registered	Gaydardzhiev	ted.smith+pend-accept-02@example.com	\$602.67	Open
00000998	4/5/17 1:31:25 pm Etc/UTC	SiteGenesis	Customer	Registered	Gaydardzhiev	ted.smith+pend-accept-02@example.com	\$110.24	Open

In Klarna's Orders system the order is marked as rejected after review:



[Overview](#) > Order #19FTMKQV

#19FTMKQV [Uncaptured](#)

US\$329.65

Merchant reference 1: 00045903 Merchant reference 2: ff5a31ea398fddcf857ef0ba Created: 26 Jun 2020, 14:27 Expires: 24 Jul 2020, 02:00 Merchant ID: N100141

[Edit](#)

Rejected after review

Customer

Shipping address

US-norm Approved
620 N High St
Columbus
43215 OH
US

Tel: +491607898065

Email: tester+pend-reject-05@klarna.com

[Edit shipping address](#)

Billing address

Additional Info

Order lines (3)

[Capture](#) [Edit order lines](#)

Qty	Item	Reference	Unit price	Discount	Tax	Amount
<input type="checkbox"/>	Casual Spring Easy Jacket	8833605242 52M	295.00	0.00	0% 0.00	295.00

<input type="checkbox"/>	1	2-Day Express	002	15.99	0.00	0% 0.00	15.99
--------------------------	---	---------------	-----	-------	------	---------	-------

<input type="checkbox"/>	1	Sales Tax	Sales Tax	18.66	0.00	0% 0.00	18.66
--------------------------	---	-----------	-----------	-------	------	---------	-------

PAYMENT DETAILS

Initial Payment Method
Pay later in parts

VISA **** *1111

Currency: USD

[Cancel order](#)

ORDER TOTAL

Captured: US\$0.00

Refunded: US\$0.00

Not Captured: US\$329.65

CUSTOMER BILLED: US\$0.00

Activity Log

26 Jun 2020

14 minutes ago

14:30
Order rejected after review
By Klarna

14:27
Order placed: US\$329.65
By Klarna

6. Klarna Payment option not available for the current purchase – the customer is presented with an appropriate message when attempt to choose Klarna is made.

SELECT PAYMENT METHOD • REQUIRED

Credit Card Pay Pal Bill Me Later

Buy now, pay later

Klarna

Buy now, pay later

Klarna

Option not available

[TESTDRIVE](#)

Unfortunately this option is not available. Please choose a different payment method.

[CONTINUE TO PLACE ORDER >](#)

7. Gift Certificate / Klarna Payments split payment

Use Klarna Payments in combination with OOTB Site Genesis Gift Certificate functionality. When a gift certificate(s) is(are) redeemed in the checkout flow and remaining order amount is paid with Klarna Payments an order line item(s) 'Gift Certificate' is added to the Klarna Order reducing the amount owned with the gift certificate Payment Instrument(s).

8. Virtual Card settlements as an alternative to standard order management

The virtual card solution can be enabled through the site preference as shown below:

The screenshot shows the Site Preferences page in the SFCC interface. The top navigation bar includes 'Merchant Tools', 'Administration', and 'Storefront'. The left sidebar has sections for 'Salesforce', 'Sandbox - klarna01 SiteGenesisGlobal', and 'Site Preferences'. The main content area displays several configuration options under 'Klarna Payments'.

Name	Value	Default Value	Actions
SecondaryTextColorPreference	CSS hex color to be used in Klarna Payments iFrame	#9C9CAU	Edit Across Sites
BorderRadiusPreference	30	5px	Edit Across Sites
Attachments	Yes	No	Edit Across Sites
Not available message on billing page	{ "GB": "Klarna Payment not available", "FR": "String In French", "AT": "Klarna Rechnung nicht verfügbar", "default": "Klarna Payment not available" }		Edit Across Sites
Virtual Card Network Enabled	Yes	No	Edit Across Sites

A red box highlights the 'Virtual Card Network Enabled' section, which contains the note: 'If set to true SFCC will create Virtual Card Network settlement from every Klarna or...'. The 'Edit Across Sites' button is also visible in this row.

This option is disabled by default. However, if standard order management is not a reasonable option for a Klarna integration, instead Klarna's virtual card solution (MCS), can be enabled. When a customer places an order, the order is first booked in SFCC. If the order has a `fraud_status` of `PENDING`, action is not taken on the order until receiving Klarna's push notification that the `fraud_status` has changed to `FRAUD_RISK_ACCEPTED`. Once an order has been accepted by Klarna, the merchant platform creates a virtual card settlement, per the merchant card services (MCS) API.

Once a settlement has been created, the merchant platform can authorize the virtual card until the Klarna order is valid. Then, once the order has been fulfilled, the card funds should be captured. (For delays in capture, or other special use cases, please speak with the Klarna key account manager in advance) While Klarna is the original payment method of the order, the order will be settled with a credit card instead of direct bank account transfer. You can find more information [here](#) around other use cases. To utilize virtual card integration option the merchant should:

- Enable VCN option in Site Preferences as shown above
- Enter the VCN Public Key ID. Unique UUIDv4 value, which is different for playground testing and Production (live site)

Name	Value	Default Value	
The Klarna Payments not available message on billing page: JSON string holdin...			
Virtual Card Network Enabled	Yes	No	Edit Across Sites
VCN Public Key ID	6c5b99c5-b0de-4689-b569-1ae12ec898eb		Edit Across Sites
Unique identifier for the public key used for encryption of the card data			
VCN Private Key	MIIJKQIAKACgEAtirQQ1705IOj5GLAGyngcKEK2V3eX2Mxjh+q05X5AG KS1a Y6mpACVNsLjYrHk42QV1C5n8vn1Dz56bhUKNKFyUwSgEusn1ewYkPZ0 audwj yJYWbwKqC50ahHTYAlieKGSwAh5lx5c+o25Xr6wx8M3Rv57mQta4sWM GeU1z9Z bpBuJTR738js3/6cxCO9usU229Fc4r2RifGD8kyIV7TvnrbDlC8DelpBkZl 3w OB11FDwOBuNIE+jTy0UkPcmw8yScR+Xe5PgLBj6M7RQtOkLjlLIS18LKR IKCK LMk3rbaNTPn+1N5LqxRdjrnh2HnaDFkKbbInHq2n93jYS72fZzaHSF02O6 Your 4096 bit RSA Private Key		Edit Across Sites
	MIICljANBgkqhkiG9w0BAQEFAOCAg8AMIICCgKCgEAtirQQ1705IOj5GLA C		

- Generate a 4096-bit RSA key pair. Set the custom preference 'vcnPublicKey' with the value of the public key without the header and footer lines (begin and end public key) and the custom preference 'vcnPrivateKey' with the value of the private key without the header and footer lines (begin and end private key). As shown below:

How to generate a 4096-bit RSA key pair:

In order to generate an RSA keypair with a 4096-bit private key you can use the following openssl command:

```
openssl genpkey -algorithm RSA -out private_key.pem -pkeyopt rsa_keygen_bits:4096
```

In order to extract the public key from an RSA keypair, you can use the following openssl command:

```
openssl rsa -pubout -in private_key.pem -out public_key.pem
```

In the folder where you have executed the above commands two new files will be created - public key.pem and private key.pem.

The contents of the files should look something like:

public_key.pem

-----BEGIN PUBLIC KEY-----

```
MIICljANBqkqhkiG9w0BAQEFAOCg8AMIICCgKCAgEAoNYG7I2G8nZa+22oBYZktV228lw3UE9WO4oxfnJtKEdHn84x55ULT8KQTh9NVtdeKC8nTfTgyvMt/GNCa18xuZv/IGYDftKt85hbV5EjOum+StAlufEXvIBX7nMOMc1KyWm9kp2kbqd88mFIX63KV94OoNExcNatRDFYR+qz53+ifadDQtQ1sIVNStdroCZDJ1+LxtBy9V+BdmsBK1ERLsKh/JLxyWE24FJKV+z00s7TQkdWW/5ET12OGQYZsWo1yqgi9HplNvrisve8vWPxaL4m8iZ3I/9yYdg7yANQbTxSjccbRCgaaagPo30CNxeqU6qafY5g8vY3E52CoXHDdO4UsIX1qcuYIDhqaDzey6W+b8m755xLi+rqQyM4PBWL0J0dM3FVid8+4YKILex3AKBFciqRCMHSOGaEeyrXKTjlAsghr9RS8PifvQRrl440cHzqw2vX0DvpjSWcmUJtW4wUq5RNSsobrxnVmoV6fj1z67Q/1P+l5le+oowdahR5ztVqjIO+2PNoX415VDs/Pkz3f8wWVc3Mp2oNT244o+/NIiyRfPFajJx7JAgrcvZt2nFAmY4QApXLFCpgEMwYucE4AH4gJKsh3KZbxRERrrO72bL2rxvWqBp/0h7DcMsV9sQs4BvxxII6CF506FThzmclaKLBAyd5LALiXiPfkCAwEAAQ==
```

-----END PUBLIC KEY-----

private_key.pem

-----BEGIN PRIVATE KEY-----

```
MIIJQQIBADANBgkqhkiG9w0BAQEFAASCCSswggknAgEAAoICAQCg1gbuXYbydlr7bagFhmS1XbbyXDdQT1Y7ijF+Scm0oR0efzjHnIQu3wpBOH01W114oLydN9ODK8y38Y0JrXzG5IX+UZgN+0q3zmFtXkSM66b5K0Ai58Re+UFucw4xzUrJab2SnaRup3zyUhfrcpX3g6g0Rdw1q1EMVhH6rPnf6J9p0NC1DWyVU1K12ugJkMnX4vG0HL1X4F2awErUREuwqH8ktfJYTbgUkpX7PTSztNCR1Zb/kRPXY4ZBhmxajXKqCL0emU2+uKy97y9Y/FovibyJncj/3Jh2DvIA1BtPFIlxttEKBppqA+jfQI3F6pTqpp9jmDy9jcTnYKhccN07hSyVfWpy5ggOGpoPN7Lpb5vybvnnEuL6upDlzf8FYvQnR0zcVWJ3z7hgogt7hCaOEVYKpElwdl4ZoR7KtcpOOUCyCGv1FLw+j+9BGsvjjRwfOrDa9fQO+mNJzyZQm1bjBSrlE1KyhuvGdWahXp+PXPrdT/U/6Xkh76ijB1qFHnO1WomU77Y82hfgjIUOz8+TPd/zBZVzcynag1Pbjij780iLJF88VoknHskCCty9m3acUCZjhAClcsUkkMmAQzBi5wTgAfikqyHcplvFERGus7vZsvavG9aoGn/SHsNwyxX2xCzgG/HEiXoiXnToVOHOZyVooEDJ3ksAuJeI9+QIDAQABAoICACRkaUsUNI22RB3yEPu3DiCPpO6v+QAeA4gTW+GUdqR9dCZLaSCZ7bhxVVUuoX4qPzslO6hjUmOyzG6upFgVPk+pHNQfyEUzoC148Eib9OziAXUN2URMpV1KbwVm+BO814X8zguai7uru0PHTG1oy6774Ct1OknxAxxHQDlxatT6XJFo5SA4EinUfnz2Bo3/xry/QjxW/mCK0GwDd4PNp9TGMFPTv2SgdSDOWzGQIOH5N3owuzMpI8NV6z74wv+i5PtV41Dzu8WhyXpiYSsk00SRKHPC68j2bAzTPghp5aSz9976SGm2SPonJXyboXdiHbl/osdyqDxeIT3iB9GmrHX/iKHPGJCh7fRZvqj39Hc+IxYjabwW3rDeDIPB7ab9z1KLF4z1D6AZOKCPyTaDRdQ1QeDi7LwDmk7NHEPrmF/nlcguQdqblbmFO2zEsOT0e6y4uBMndRsbQprTNSMUdBkrAINaYVSTQ1Z0Y/8DZDpGcyS1OnJv74F15uDjKN6/ov991mZ1JrZ+v2sdS3EDUlmvP6thQKwl7Ln6h+ApHtWUG1NmVQe5gJE0qAeJ9b45clUzIRUwhVmEp8Noljh0kAjaNd4lk7xy9ZRDUY5yekPeYrJPShjsHAyEoktJljRufI2UUq3uxNjjlCoQcOVGfNDISYTTPwpu1pmC0C+rh2fgBAoIBAQDRultRArvtc2JKhVOUyZk88zd9kvrl6fNiyKmiHgiWf7qkTPD9xhOQWDw3iwRFQAD+YkgV5MCBO8wp8oO8GEsOCI+XZWExOcPT0VfjPZHiQrTFnlfG/+fAO14xLf3j3ED4YQXdHOKI3xoLknQx/EydLoctxgkkpgWLrsA7DwdSAg1/0sBvaHY27ogAfdimHdaKZ5OAe4a9k1qP3xVZBuOe8Sd65unBavUJLDuvikeNmkSVgW1sm55/729Jlr63USHF76lt+vE1cdZ+vKg5vYotsQgPzvNBmUO/E8GjzMXQRfqfvEDINXEX0rCupTkw1G6AGTwQc/NPzyr/LTpLe6UBAoIBAQDEUjTiG11Vhf7WjdG3gctRlr+mYapQHgXdVlx2QSaqUYid+0QXK11YfJlsRB6nwa+OED83RfP0
```

```
lIFqxpzudSLPmoDulBT7DI5c/aleyKs/siUusP8QVDXk6OAR84XSytC35sIRV7pE  
VMuBL91jfkQ0Lf/PresIK/kI6Yvwwwp4qrHK6/f9TgciHclYtf+/oti4ky6GJgfmP  
fmuCqjxmUKbXXFPd5RbL2THGOowilb8zDLjf3RlbjlQFqogAk6H9hp2V0VZLiJHp  
UWM3z3zxDWedaqJ08sHuk/rA9QpsVTu8IGTQsxdj8JwluN1Q+YZiOuPiSENBqPzT  
V3exezo3sD5AoIBAGU3qEyPojz1+9D1Sal8LW2CABzlq4z9g84ABAZOslxX5q7W  
x1PinZyDSQRXg1B13jt29ZdIR79ygnQlg1YOBjvtgVQHPuafk3RIBQbbCh+val  
9dn/tUxMGqhnunKaby1rovJHfdqnPpKwzNAjYUqaGkJ822xhmmke/fEyAanlPa4  
stDRvIPEWPTLx5xcOCdx13khPKsnkgRvaLEfpwkVX7Vr7hK/2OSFaYTNmrxXYBQ7  
c6D/9d3Oo4nLb/mu+Tq67S19t53Qg/GEgTfkpuRoVPi0KyhUnKKCGWIBMZLTwyIG  
S9eTFDKoJ0cSTGipjW7bPua93wZ8eEbRABpf4QECggEANNhQBeEJ0aCdBVHtdrEI  
crDaa8X0W1aJi5dol4hYCRajaKsfHAF/QfdgMQVxHwUC5YG4En/Q+DAVWhGWYpXD  
RhC3zeFy5FVsyzk0sx/fAOIKGvRn5BRW4YRR9GMRzbjsT+RcruBnckdE9ERXGpX9  
c/JB3rxZBlt+oliFM8yfWKtMwsrmNKtFuDftvJeok4KejycFF4eWDqsf828xjPT+  
xA/FP4CQD1UqkcpmuFSlwAwXo6LXVY7NTS0nKMiUnTLkLTIHtLnO9+9jmNapWRP  
Tc+hZUuHKlpI8DHFmX2j87LgkFD05eD5lynY4RgZtU1W1C1RdVYwoA72WB7knEaB  
uQKCAQAH9s67P/7fFX9dfEans3PHU4nGjD8dJ8eoNQ6DhBMydZpGWI5ZUeEBZDRk  
0cBOeRs5BOcS43Em9kETpzawyCwxmnwzl+CzoPzMQcTw9tXomF9HG6RJ9XBdjfGA  
ALAwCd4bASxmFM6guSP5GKnZ9aY3tR3tWWDFr7f9z8wOewzzpPclwRh009fPe4TC  
NXoEm1MELJVeUieDSLKZgjgCw8WHGqltONpA0/fwSM2glcxETVV7qx3aPuJzCVh  
LQZoBLQk3UMKsWDdpzeBdiERe66NAgVk92Xe7SY9EY2vymaq761i1x1vlprT27qp  
240LDJawqM0IraKmdCvWjofWSaOU  
-----END PRIVATE KEY-----
```

- Finally, you need to send the generated key_id/public key in JWK format to Klarna prior to testing/go-live. It will be used to encrypt the virtual card pan and csc on Klarna side when settlement request is made. After confirmation from Klarna that the key has been successfully added to your merchant profile you would be able to use virtual card-based settlement option for Klarna payment methods.

Note:

The recommend RSA keypair size of 4096 bits. This key pair must be associated with a unique key_id (UUIDv4). The public key must be shared in JWK format with Klarna contact. Note that for production and playground, the Key id and corresponding keypair combination shared are different and must be configured prior to testing/go-live with the agreed virtual card product

- Authorize the virtual credit card

The virtual credit card details are stored as order level custom attributes, and can be accessed through the Order object as follows:

- Order.custom.kpVCNBrand – virtual credit card brand;
- Order.custom.kpVCNCSC – virtual credit card csc;
- Order.custom.kpVCNExpirationMonth – virtual credit card expiration month;

- Order.custom.kpVCNExpirationYear - virtual credit card expiration month ;
- Order.custom.kpVCNHolder - virtual credit card holder;
- Order.custom.kpVCNPAN - virtual credit card number;
- Order.custom.kplIsVCN – Boolean property indicating whether VCN is used.

Note: Please review the details in the order export XML prior to go-live. This is to ensure that card details are fetched from the correct custom attribute fields in your Orders XML file for post purchase order processing.

These attributes can be inspected in the BM order details in the attributes tab as shown below:

[Merchant Tools](#) > [Ordering](#) > [Orders](#) > Order: 00045802(RefArch)

General **Attributes** Payment Notes History

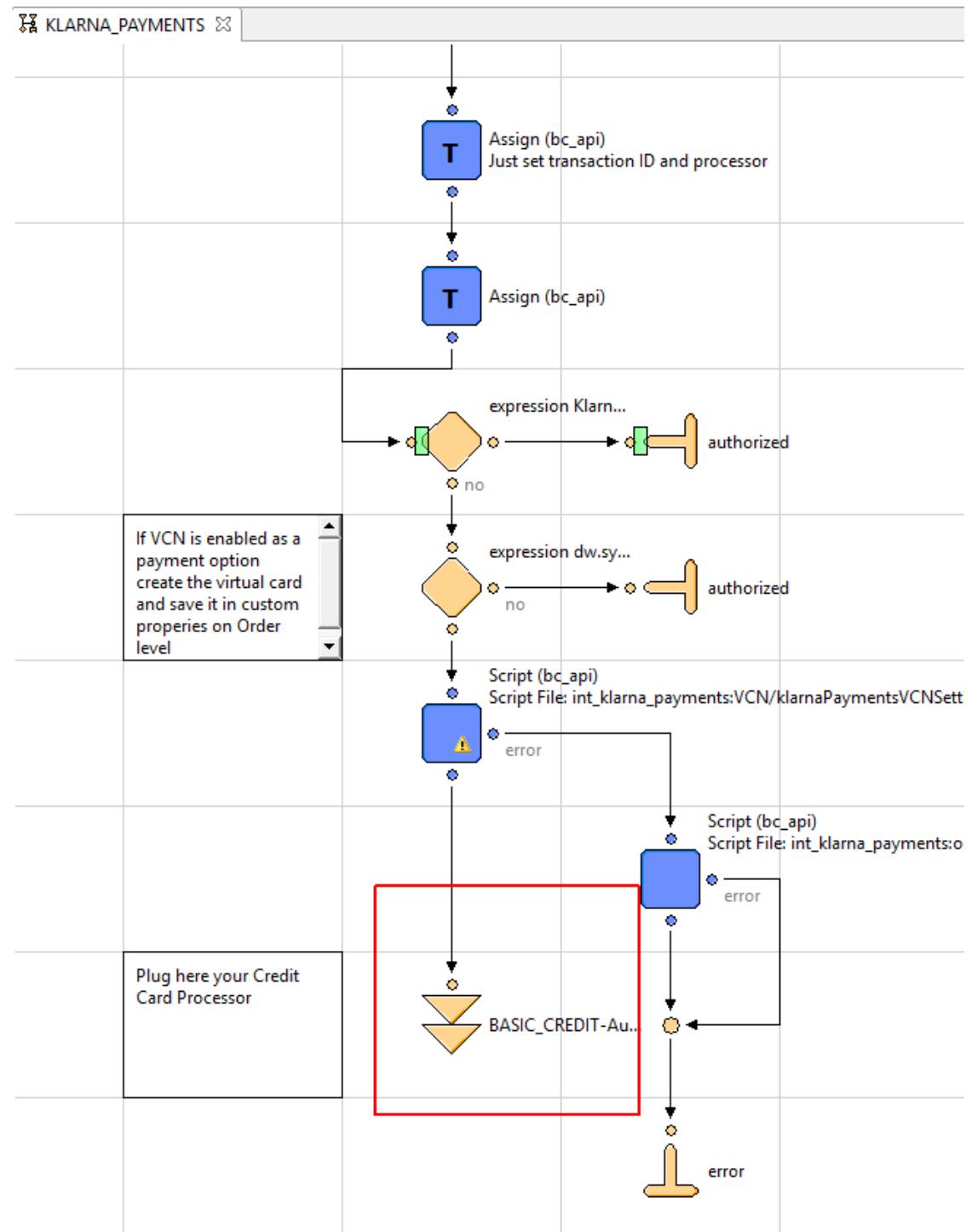
Attributes for Order '00045802'

On this page you can edit the attributes of the order. Fields with a red asterisk (*) are mandatory. Click **Apply** to save changes. Click **Reset** to revert your changes.

Klarna Payments Order ID:	7cb6837b-3978-22e0-846c-4bbbde8de39f
Is VCN Used:	<input checked="" type="checkbox"/>
VCN Holder:	Klarna Bank AB
VCN PAN:	1111118556588816
VCN Expiration Year:	20
VCN Expiration Month:	02
VCN CSC:	794
VCN Brand:	VISA

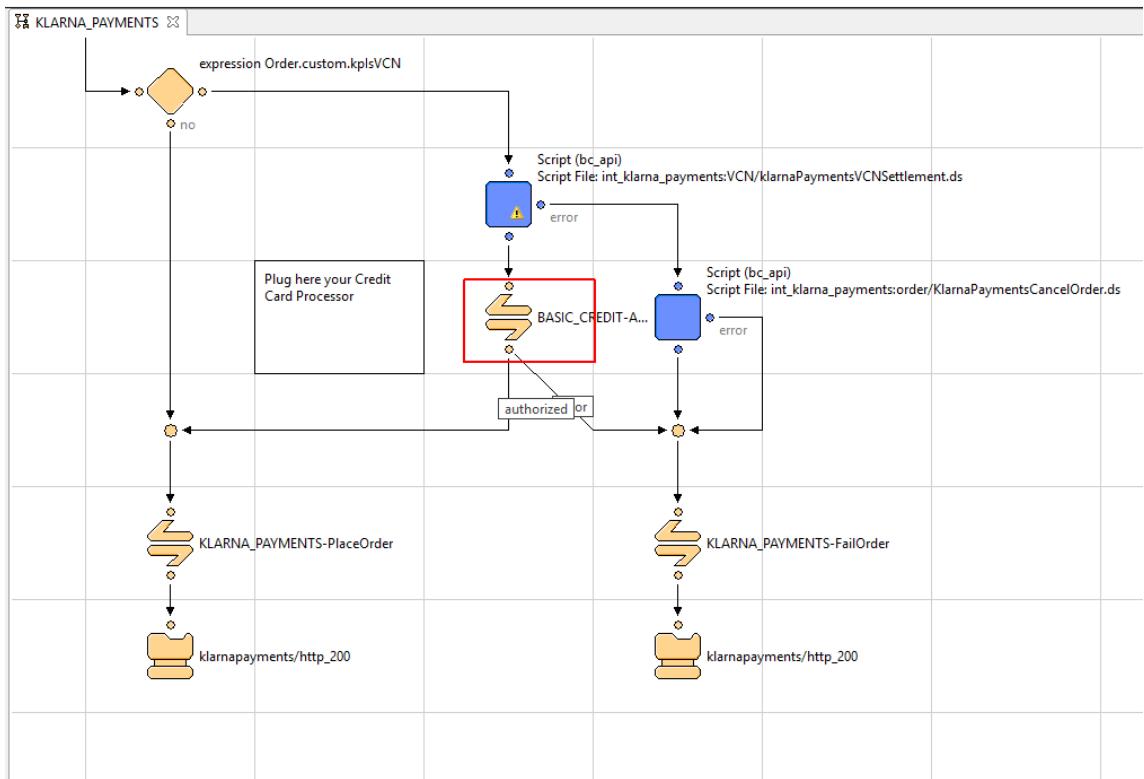
<< Back to List

It is up to the merchant to use the credit card details as described above to authorize the virtual credit card. As an example, authorization is done with calling BASIC_CREDIT-Authorize as below (pipeline version):



You need to add your own logic in KLARNA_PAYMENTS-Authorize in the place of BASIC_CREDIT-Authorize call node.

You would also need to add your own card authorization login in the place of BASIC_CREDIT-Authorize call node in the KLARNA_PAYMENTS-Notification pipeline as shown below:



9. Configure On-Site Messaging

Klarna On-Site Messaging (OSM) is configured per-site per-locale via the **KlarnaCountries** custom object. To configure the OSM settings for a locale, you must visit “*Merchant Tools – Custom Object Editor*” and search for **KlarnaCountries** custom object. Provide a valid locale for the OSM tag based on the country being configured. The OSM Data Client ID and Data keys required are available in Klarna Merchant Portal – On-site Messaging App

Merchant Tools > Custom Objects > Custom Objects > US - General

General																			
Manage 'US' (KlarnaCountries)																			
Fields with a red asterisk (*) are mandatory. You can view and edit the name and description in other languages, if required. Click Apply to save the details.																			
custom <table border="1"> <tr> <td>Country Code:</td> <td>US</td> </tr> <tr> <td>On-site Messaging Data Locale:</td> <td>en-US</td> </tr> <tr> <td>Service Credential ID:</td> <td>klarna.http.uscredentials</td> </tr> <tr> <td>On-site messaging Data Client ID:</td> <td>60a22a39-c2fd-5d09-bfe1-771459318a4d</td> </tr> <tr> <td>Cart Placement Tag Enabled:</td> <td><input checked="" type="checkbox"/></td> </tr> <tr> <td>Cart Placement Data Key:</td> <td>credit-promotion-standard</td> </tr> <tr> <td>PDP Placement Tag Enabled:</td> <td><input checked="" type="checkbox"/></td> </tr> <tr> <td>PDP Placement Data Key:</td> <td>credit-promotion-standard</td> </tr> <tr> <td>Library URL:</td> <td>https://na-library.playground.klarnaservices.com/lib.js</td> </tr> </table>		Country Code:	US	On-site Messaging Data Locale:	en-US	Service Credential ID:	klarna.http.uscredentials	On-site messaging Data Client ID:	60a22a39-c2fd-5d09-bfe1-771459318a4d	Cart Placement Tag Enabled:	<input checked="" type="checkbox"/>	Cart Placement Data Key:	credit-promotion-standard	PDP Placement Tag Enabled:	<input checked="" type="checkbox"/>	PDP Placement Data Key:	credit-promotion-standard	Library URL:	https://na-library.playground.klarnaservices.com/lib.js
Country Code:	US																		
On-site Messaging Data Locale:	en-US																		
Service Credential ID:	klarna.http.uscredentials																		
On-site messaging Data Client ID:	60a22a39-c2fd-5d09-bfe1-771459318a4d																		
Cart Placement Tag Enabled:	<input checked="" type="checkbox"/>																		
Cart Placement Data Key:	credit-promotion-standard																		
PDP Placement Tag Enabled:	<input checked="" type="checkbox"/>																		
PDP Placement Data Key:	credit-promotion-standard																		
Library URL:	https://na-library.playground.klarnaservices.com/lib.js																		
<< Back to List																			

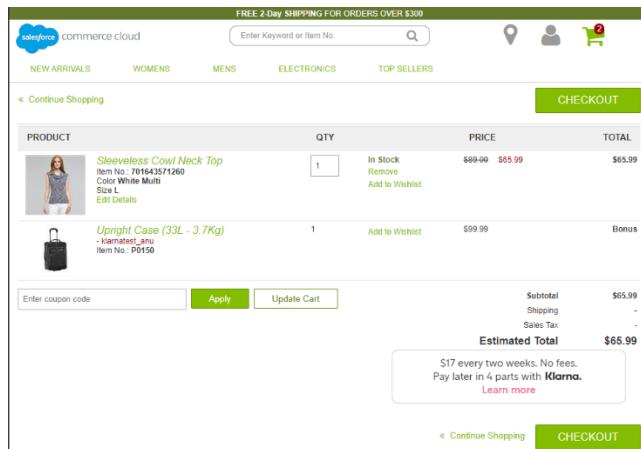
To enable Placement tag for the Cart Page, the “*Cart Placement Data Key*” must be filled with Data Key value and the “*Cart Placement Tag Enabled*” must be checked

To enable Placement tag for the PDP Page, the “*PDP Placement Data Key*” must be filled with Data Key value and the “*PDP Placement Tag Enabled*” must be checked

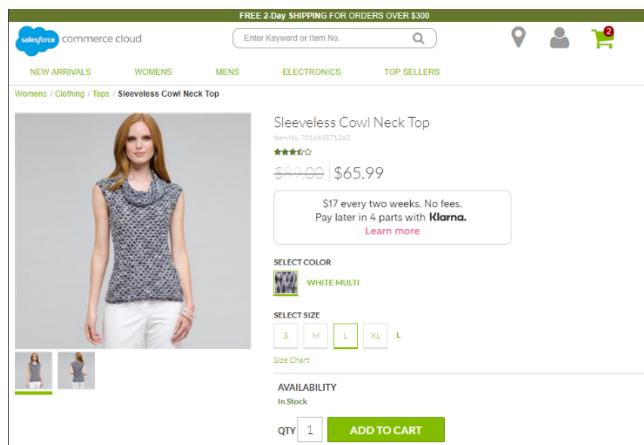
In Library URL, please input the full URL to the On-Site Messaging JavaScript Library.

In On-site messaging Data Client ID, please input the On-Site Messaging Data client ID value

On-site messaging Data locale, please input the valid On-Site Messaging Data locale



Klarna On-Site Messaging Enabled on Cart Page



Klarna On-Site Messaging Enabled on PDP Page

For more information regarding customizations and best practices, please refer to the Klarna Developer: <https://developers.klarna.com/documentation/on-site-messaging/customization/>

10. Configure Auto-capture

When the preference is enabled (disabled by default), a full amount capture is attempted prior to acknowledging the order with Klarna. If the capture is successful, the SFCC order's payment transaction is marked as Paid, and viewable in the Business Manager.

[Merchant Tools](#) > [Ordering](#) > [Orders](#) > Order: 00029204(RefArchGlobal)

General Attributes **Payment** Notes History

Payment Information for Order '00029204'

Order Total:	£199.98
Amount Paid:	£0.00
Balance Due:	£199.98
Invoice Number:	00119002
Payment Status:	Paid
Payment Method:	KLARNA_PAYMENTS Processor: KLARNA_PAYMENTS Transaction: 91ab359d-ada0-71e0-834e-6eb59b83bf9f Amount: £199.98 Klarna Payment Category ID: pay_later Klarna Payment Category Name: Pay later. Fraud Status: ACCEPTED

The order will be marked as “*Captured*” in the Klarna’s Merchant Dashboard.

Merchant reference 1 00029205	Klarna reference BJJGH9P1	Created Jun 25, 2019, 5:00 PM	Expires Jul 23, 2019, 3:00 AM	Merchant ID K500726
Edit				
Customer Shipping address Angela Gill 4939 Wyatt Street West Palm Beach SW42 4RG Florida GB Tel 01222 555 555 Email ivan.zanev@tryzens.com Edit shipping address	Order lines (2) Refund Qty Item Reference Unit price Discount Tax Amount 1 Black Flat Front Wool Suit 7505187030... 191.99 0.00 5% 9.14 191.99 1 Наземен транспорт GBP001 7.99 0.00 5% 0.38 7.99	Currency: GBP		
Billing address	PAYOUT DETAILS Initial Payment Method Statement Resend statement			
Additional Info	ORDER TOTAL Captured £199.98 Refunded £0.00 Not Captured £0.00			
	CUSTOMER BILLED £199.98			
Activity Log Jun 25, 2019 9 minutes ago	5:00 PM Order acknowledged Via API			
	5:00 PM Captured: £199.98 Via API			
	5:00 PM Order placed: £199.98			

If the capture is unsuccessful, an error will be logged in the custom error log.

2.3 Limitations, Constraints

The merchant will need a configured Klarna Payments account/MID for playground and production. Note: The later requires a contract with Klarna

The merchant can style the Klarna Payments widget (skin), to make it matching the look and feel of their store. The list with the graphic elements that can be customized, through site preferences is as follows:

- "color_details" (site preference kpColorDetails): "#COFFEE"
- "color_button" (site preference kpColorButton): "#COFFEE"
- "color_button_text" (site preference kpColorButtonText): "#COFFEE"
- "color_checkbox" (site preference kpColorCheckbox): "#COFFEE"
- "color_checkbox_checkmark" (site preference kpCheckboxCheckmark): "#COFFEE"
- "color_header"(site preference kpColorHeader): "#COFFEE"
- "color_link"(site preference kpColorLink): "#COFFEE"
- "color_border"(site preference kpColorBorder): "#COFFEE"
- "color_border_selected"(site preference kpBorderSelected): "#COFFEE"
- "color_text"(site preference kpColorText): "#COFFEE"
- "color_text_secondary"(site preference kpColorTextSecondary): "#COFFEE"
- "radius_border"(site preference kpRadiusBorder): "0px"

Successful payment authorizations (Klarna Payments status APPROVED) lead to a 'Paid' order payment status and 'Ready for Export' export status

Refused and pending payment authorizations (Klarna Payments statuses REJECTED and PENDING) lead to a 'Not Paid' order payment status, and 'Not Exported' export status

Cancels and refunds lead to a 'Not Paid' order payment status (even if the status was 'Paid' before), and 'Not Exported' export status

The cartridge makes use of the Klarna Payments JSON REST API and a JavaScript SDK.

2.4 Compatibility

Compatible with Commerce Cloud 18.1.0 and SiteGenesis 105.0.0

2.5 Privacy, Payment Data

1. The cartridge is compliant with GDPR recommendation and follows the best practice mentioned [here](#) and implementation follows practice mentioned [here](#)
2. EMD (Extra Merchant Data): The cartridge supports sending additional information on the customer's past purchase history when turned on in custom preferences: Attachements(kpAttachments). The type of data that can be send as an attachment is mentioned here. EMD is required for certain types of merchant order and inclusion of EMD is (e.g.customer_account_info: past purchase with merchant store) generally beneficial to improve acceptance rates. EMD is included as part of Authorization step in Commerce Cloud checkout. The data send to Klarna is customizable through a hook. An example hook contents are provided in int_klarna_payments/scripts/EMD/KlarnaPaymentsBuildEMD_hooks.js. The merchant can customize the contents of the BuildEMD function. Current Basket or Order object can be accessed thorough args. LineItemCtnr. This script should return a JSON string to be used as a value for the body sub-field of the attachment field as [described here](#). If the example KlarnaPaymentsBuildEMD_hooks.js file is used unchanged the data send to Klarna is by the following schema:

{

```
"$schema": "http://json-schema.org/draft-03/schema#",
"id": "http://klarna.com/v2/emd#",
"description": "Extended Merchant Data Payload Schema",
"type": "object",
"properties": {
    "customer_account_info": {
        "type": "array",
        "items": {
            "type": "string"
        }
    }
}
```

```

    "type": "object",
    "properties": {
        "unique_account_identifier": {
            "type": "string",
            "maxLength": 24
        },
        "account_registration_date": {
            "description": "ISO 8601 e.g. 2012-11-24T15:00",
            "type": "string",
            "format": "date-time",
            "pattern": "^[0-9][0-9][0-9][0-9]-[0-1][0-9]-[0-3][0-9]T[0-2][0-9]:[0-5][0-9](:[0-5][0-9])\\{0,1}Z\\{0,1}\\$"
        },
        "account_last_modified": {
            "description": "ISO 8601 e.g. 2012-11-24T15:00",
            "type": "string",
            "format": "date-time",
            "pattern": "^[0-9][0-9][0-9][0-9]-[0-1][0-9]-[0-3][0-9]T[0-2][0-9]:[0-5][0-9](:[0-5][0-9])\\{0,1}Z\\{0,1}\\$"
        }
    }
},
"payment_history_full": {
    "type": "array",
    "items": {
        "type": "object",

```

```

    "additionalProperties": false,
    "properties": {
        "unique_account_identifier": {
            "type": "string"
        },
        "payment_option": {
            "type": "string",
            "enum": ["card", "direct banking", "non klarna credit",
                     "sms", "other"]
        },
        "number_paid_purchases": {
            "type": "integer"
        },
        "total_amount_paid_purchases": {
            "type": "number"
        },
        "date_of_last_paid_purchase": {
            "description": "ISO 8601 e.g. 2012-11-24T15:00",
            "type": "string",
            "format": "date-time",
            "pattern": "^[0-9][0-9][0-9][0-9]-[0-1][0-9]-[0-3][0-9]T[0-2][0-9]:[0-5][0-9](:[0-5][0-9])\\{0,1\\}Z\\{0,1\\}\\$"
        },
        "date_of_first_paid_purchase": {
            "description": "ISO 8601 e.g. 2012-11-24T15:00",
            "type": "string",
            "format": "date-time",
        }
    }
}

```

```

        "pattern": "[0-9][0-9][0-9][0-9]-[0-1][0-9]-[0-3][0-9]T[0-2][0-9]:[0-5][0-9](:[0-5][0-9]){{0,1}}Z{{0,1}}$"
    }
}
}
}
}

```

Example data:

```
{
  "extra_merchant_data": {
    "content_type": "application/vnd.klarna.internal.emd-v2+json",
    "body": {
      "customer_account_info": [
        {
          "unique_account_identifier": "5509d9f7c8720c0e4575154b",
          "account_registration_date": "2015-03-18T20:03:03Z",
          "account_last_modified": "2015-03-18T20:03:03Z"
        },
        {
          "purchase_history_full": [
            {
              "unique_account_identifier": "5509d9f7c8720c0e4575154b",
              "payment_option": "card",
              "number_paid_purchases": "23",
              "total_amount_paid_purchases": "140023",
              "date_of_last_paid_purchase": "2015-03-18T20:03:03Z",
              "date_of_first_paid_purchase": "2015-03-18T20:03:03Z"
            }
          ]
        }
      }
    }
}
```

3. PCI-DSS compliance

The virtual card (MCS) solution enables settlements using individual virtual card issued against a Klarna order. To be compliant with PCI-DSS requirements, merchant must ensure the data is securely maintained and transmitted as part of their operation in their live shop environment. This maybe be done in consultation with your payment service provider/acquirer and completed prior to go-live.

3. Implementation Guide

To setup and implement Klarna Payments Salesforce Commerce Cloud Cartridge, the following steps must be followed:

3.1 Setup

Klarna Payments integration has three cartridges:

- `int_klarna_payments` which implements the core storefront functionality;
- `int_klarna_payments_pipelines` which implements the storefront functionality with pipeline based approach;
- `int_klarna_payments_controllers` which implements the storefront functionality with controllers based approach;

3.1.1 *int_klarna_payments*

The `int_klarna_payments` cartridge has the following components:

Cartridge name	<ul style="list-style-type: none">• <code>int_klarna_payments</code>
Scripts	<ul style="list-style-type: none">• <code>/checkout</code><ul style="list-style-type: none">▪ <code>SetPendingOrderStatus.ds</code>• <code>/common</code><ul style="list-style-type: none">▪ <code>KlarnaPaymentsApiClientContext.js</code>▪ <code>KlarnaPaymentsHttpService.ds</code>• <code>/EMD</code><ul style="list-style-type: none">▪ <code>KlarnaPaymentsBuildEMD_hooks.js</code>• <code>/locale</code><ul style="list-style-type: none">▪ <code>KlarnaPaymentsGetCountryFromLocale.ds</code>▪ <code>KlarnaPaymentsGetLocale.ds</code>• <code>/order</code><ul style="list-style-type: none">▪ <code>KlarnaPaymentsAcknowledgeOrder.ds</code>▪ <code>KlarnaPaymentsCancelOrder.ds</code>▪ <code>KlarnaPaymentsCreateOrder.ds</code>▪ <code>KlarnaPaymentsOrderModel.js</code>▪ <code>KlarnaPaymentsOrderRequestBuilder.js</code>• <code>/session</code><ul style="list-style-type: none">▪ <code>KlarnaPaymentsCreateSession.ds</code>▪ <code>KlarnaPaymentsSessionModel.js</code>▪ <code>KlarnaPaymentsSessionRequestBuilder.js</code>▪ <code>klarnaPaymentsUpdateSession.ds</code>• <code>/util</code><ul style="list-style-type: none">▪ <code>Builder.js</code>▪ <code>KlarnaPaymentsConstants.js</code>• <code>/VCN</code><ul style="list-style-type: none">▪ <code>klarnaPaymentsVCNSettlement.ds</code>• <code>createKlarnaPaymentInstrument.ds</code>• <code>hooks.json</code>• <code>klarnaRemovePreviousPI.ds</code>

- Static*
 - /default
 - Js
 - *klarna-payments.js*
- Templates*
 - /default
 - *klarnapayments*
 - *http_200.isml*
 - *modules.isml*
 - *paymentmethodhelper.isml*
 - *shippingaddresshelper.isml*
 - *klarnapaymentblock.isml*
 - *klarnapaymentscategories.isml*
 - /resources
 - *klarnapaymentsresources.isml*

3.1.2 int_klarna_payments_pipelines

The int_klarna_payments_pipelines cartridge has the following components:

- Cartridge name*
 - int_klarna_payments_pipelines
- Pipelines*
 - *KLARNA_PAYMENTS.xml*

3.1.3 int_klarna_payments_controllers

The int_klarna_payments_controllers cartridge has the following components:

- Cartridge name*
 - int_klarna_payments_controllers
- Controllers*
 - *KLARNA_PAYMENTS.js*
- Scripts*
 - *hooks.json*
- Other*
 - *package.json*

If the base cartridges are named differently the variables shown above need to be renamed to the correct paths based on the individual integration.

3.2 Configuration

3.2.1 Metadata Import

Go to ‘metadata’ folder, review and edit, if needed, the site-template contents. (Site template is prepared to setup SiteGenesis and RefArch sites - you may want to change that to your actual sites and delete the ones that are not needed). Zip the directory and you’ll have ‘site-template.zip’ installation package. Import it through BM Administration > Site Development > Site Import & Export section.

The following configuration items will be added after you import ‘site-template.zip’ archive:

This screenshot shows the configuration interface for Klarna Payments within Site Development. It includes sections for 'Send product_url and image_url', 'merchant_reference2 mapping', and 'Pre-Assessment is on for:'.

Name	Value	Default Value
Send product_url and image_url	None	If set to true product_url and image_url fields in Klarna Payments create session call will be included in API call, otherwise product_url and image_url fields will not be populated
merchant_reference2 mapping		The field from SCC order (basket) object that is mapped to merchant_reference2 field from klarna API request
Pre-Assessment is on for:		Pre-assessment flag. Comma separated string values with country codes, which should have the pre-assessment flag ON. For example the following flag "US, CN"

See full list below.

Besides configuration parameters, the following order attributes will be added:

This screenshot shows the 'Object Type: Order' attribute definitions. It lists a single attribute named 'Klarna Payments Order ID'.

ID or Name	Type	Attribute Settings	Values
Klarna Payments Order ID	String		New Delete

Custom Object Definition:

After importing metadata there will be a site specific custom object to store Klarna countries:

Object Type 'KlarnaCountries'

This page lists the attribute definitions of your object type. Use the search to find attribute definitions by ID and name.

Click New to create new attribute definitions. Click Delete to delete existing attribute definitions.

Search Attribute Definitions						
ID or Name:	Find		Type	Attribute Settings	Values	
Select All	ID	Name				
<input type="checkbox"/>	UUID	UUID	String	*	0	Edit
<input type="checkbox"/>	country	Country Code	String	*	0	Edit
<input type="checkbox"/>	creationDate	Creation Date	Date+Time	*	0	Edit
<input type="checkbox"/>	credentialID	Service Credential ID	String		0	Edit
<input type="checkbox"/>	klarna_locale	Klarna Locale	String		0	Edit
<input type="checkbox"/>	lastModified	Last Modified	Date+Time	*	0	Edit
<input type="checkbox"/>	osmCartEnabled	Cart Placement Tag Enabled	Boolean		0	Edit
<input type="checkbox"/>	osmCartTagId	Cart Placement Tag ID	String		0	Edit
<input type="checkbox"/>	osmLibraryUrl	Library URL	String		0	Edit
<input type="checkbox"/>	osmPDPEnabled	PDP Placement Tag Enabled	Boolean		0	Edit
<input type="checkbox"/>	osmPDPTagId	PDP Placement Tag ID	String		0	Edit
<input type="checkbox"/>	osmUCI	On-site messaging UCI	String		0	Edit

Note: The same custom object is used by Klarna Checkout

KlarnaCountries custom object holds the corresponding locale for each Klarna Payments endpoint. Even if you have locales that are not supported by Klarna Payments we recommend to make a corresponding entry in the custom object for that locale (you can use one of the other endpoints). Thus, on the billing page of the unsupported locale you will have the Klarna Payments widget showing an appropriate message like below:

If Klarna Payments is not supported as a payment method and you have not updated the locale entry in the KlarnaCountries custom object as stated above, the customer will be presented with the following more general message on billing page:

3.2.2 Cartridge Path

If using the controller based integration, cartridge path should be setup as such under Sites / Manage Sites /

app_storefront_controllers:app_storefront_core:int_klarna_payments_controllers:int_klarna_payments

Instance Type: Sandbox/Development ▾

Deprecated. The preferred way of configuring HTTP and HTTPS hostnames is by using new features of the site aliases configuration ("Site URLs/Aliases Configuration"). The HTTP/HTTPS hostnames values set in this section will be used if no hostnames are defined by the aliases.

HTTP Hostname:	<input type="text"/>
HTTPS Hostname:	<input type="text"/>
Instance Type:	All
Cartridges:	app_storefront_controllers:app_storefront_core:int_klarna_payments_controllers
Effective Cartridge Path:	app_storefront_controllers:app_storefront_core:int_klarna_payments_controllers:int_klarna_payments_plugin_apple_pay_plugin_facebook_plugin_pinterest_commerce_bc_api_core

If using pipelines, the cartridge path should be similar to:

app_storefront_pipelines:app_storefront_core:int_klarna_payments_pipelines:int_klarna_payments

Instance Type: Sandbox/Development ▾

Deprecated. The preferred way of configuring HTTP and HTTPS hostnames is by using new features of the site aliases configuration ("Site URLs/Aliases Configuration"). The HTTP/HTTPS hostnames values set in this section will be used if no hostnames are defined

HTTP Hostname:

HTTPS Hostname:

Instance Type: All

Cartridges:

Effective Cartridge Path:

3.2.3 Klarna Payments Configurations

Also, there is a page in Business Manager where you have to configure Klarna Payments cartridge, you can find it in Site Preferences > Custom Preferences > Klarna_Payments:

The screenshot shows the Klarna Payments configuration page within the Site Preferences section of the Business Manager. The top navigation bar includes 'Merchant Tools' (selected), 'Administration', and 'Storefront'. The left sidebar shows 'Merchant Tools / Site Preferences / Custom Site Preference Groups / Klarna Payments'. The main content area has a header 'Klarna Payments' with tabs for 'Edit Across Sites' and 'Save'. It displays three configuration items:

- Send product_url and image_url**: A dropdown menu set to 'None'. A note below says: "If set to true product_url and image_url fields in Klarna Payments create session call will be included in API call, otherwise product_url and image_url fields will not be populated".
- merchant_reference2 mapping**: A dropdown menu showing a single option. A note below says: "The field from SCC order (basket) object that is mapped to merchant_reference2 field from klarna API request".
- Pre-Assessment is on for:** A dropdown menu set to 'None'. A note below says: "Pre-assessment flag. Comma separated string values with country codes, which should have the pre-assessment flag 'On'. For example the following flag 'US, CN'".

Parameter Name	Attribute ID	Description
Auto-capture	kpAutoCapture	If enabled, a full capture will be attempted automatically prior to order acknowledge. If disabled, the merchant needs to manually capture the order amount from merchant OMS or via Klarna Merchant Portal. Auto-capture is possible for orders when virtual card solution is not enabled.
Send product_url and image_url	sendProductAndImageURLs	If set to true product_url and image_url fields in Klarna Payments create session call will be included in API call, otherwise product_url and image_url fields will not be populated.
Merchant Reference 2 Mapping	merchant_reference2_mapping	The field from SCC order (basket) object that is mapped to merchant_reference2 field from klarna API request.

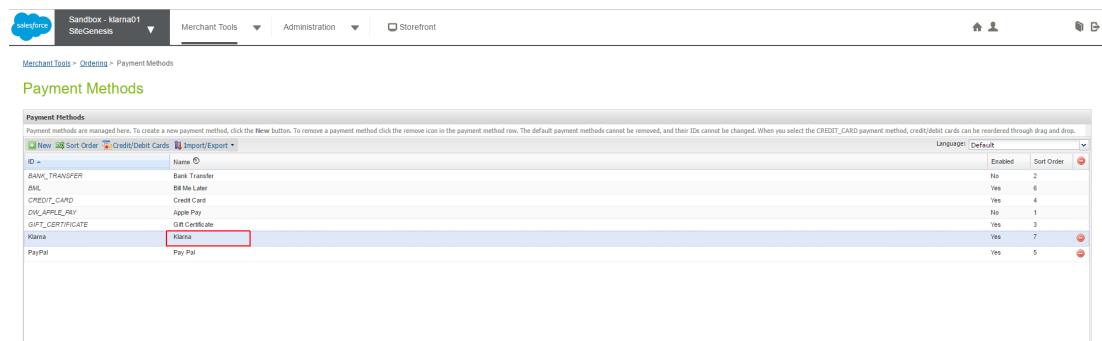
		Has to be one of the class attributes of SCC LineItemCtnr. Note that for complex data structures result may not always be as expected.
Border Color Preference	kpColorBorder	CSS hex color to be used in Klarna Payments iFrame
Border Selected Color Preference	kpColorBorderSelected	CSS hex color to be used in Klarna Payments iFrame
Button Color Preference	kpColorButton	CSS hex color to be used in Klarna Payments iFrame
Button Text Color Preference	kpColorButtonText	CSS hex color to be used in Klarna Payments iFrame
Checkbox Color Preference	kpColorCheckbox	CSS hex color to be used in Klarna Payments iFrame
Checkbox Checkmark Color Preference	kpColorCheckboxCheckmark	CSS hex color to be used in Klarna Payments iFrame
Details Color Preference	kpColorDetails	CSS hex color to be used in Klarna Payments iFrame
Header Color Preference	kpColorHeader	CSS hex color to be used in Klarna Payments iFrame
Link Color Preference	kpColorLink	CSS hex color to be used in Klarna Payments iFrame
Text Color Preference	kpColorText	CSS hex color to be used in Klarna Payments iFrame
Secondary Text Color Preference	kpColorTextSecondary	CSS hex color to be used in Klarna Payments iFrame
Border Radius Preference	kpRadiusBorder	Size (in pixels) of the border radius to be used in Klarna Payments iFrame
Attachments	kpAttachments	Flag to switch on/off the using of attachments when creating a session. Default is OFF.
Not available message on billing page	kpNotAvailableMessage	The Klarna Payment not available message on billing page. JSON string holding country code and corresponding message string. For example: { " GB ":"Klarna Payment not available", " default ":"Klarna Payment not available" }
Virtual Card Network Enabled	kpVCNEnabled	If set to true SFCC will create Virtual Card Network settlement from every Klarna order. Default is false

VCN Public Key ID	kpVCNkeyId	Unique identifier for the public key used for encryption of the card data
VCN Private Key	vcnPrivateKey	Your VCN 4096 bit RSA Private Key
VCN Public Key	vcnPublicKey	Your VCN 4096 bit RSA Public Key

Here is the description of the Site Preferences fields which have to be configured:

3.2.4 Klarna Payments Logo and Payment Option Name

Payment option name can be changed in BM Merchant Tools > Ordering > Payment Methods as shown below.



Payment option name can be localized for the specific country and language through the language dropdown menu on the right, as shown below:



Logo can be changed from BM Merchant Tools > Ordering > Payment Methods. Enter the file name with the logo in the image field on payment method details view (see screenshot below)

3.2.5 Services Configuration

After the metadata import you should have Klarna service ready in Administration > Operations > Services like so:

You need to configure Klarna Payments services credentials. Go to Credentials tab as shown below:

You need to edit the existing credentials or create new ones using the credentials provided to you by Klarna and the API URL for the specific location, see for example below:

The Klarna Payments API is accessible through a few different URLs. There are different URLs for testing and for making live purchases as well as different URLs for depending on if you are based in Europe or in the U.S.

Live environment

The API for the European production environment can be found at

- <https://api.klarna.com/>

The API for the U.S. production environment can be found at

- <https://api-na.klarna.com/>

The API for the Oceania production environment can be found at

- <https://api-oc.klarna.com/>

Testing environment

The API for the European Playground/testing environment can be found at

- <https://api.playground.klarna.com/>

The API for the U.S. Playground/testing environment can be found at

- <https://api-na.playground.klarna.com/>

The API for the Oceania Playground/testing environment can be found at

- <https://api-oc.playground.klarna.com/>

3.2.6 *Custom attributes*

ORDER system object has been extended with the following custom attributes:

Parameter Name	Attribute ID	Description
Klarna Payments Order ID	kpOrderID	Holding the Klarna Payments Order ID, if Klarna Payments is used for the specific order
VCN Brand	kpVCNBrand	Holding the Klarna Payments Virtual Card Network card Brand

VCN CSC	kpVCNCSC	Holding the Klarna Payments Virtual Card Network card csc (encrypted)
VCN Expiration Month	kpVCNEExpirationMonth	Holding the Klarna Payments Virtual Card Network card expiration month
VCN Expiration Year	kpVCNEExpirationYear	Holding the Klarna Payments Virtual Card Network card expiration year
VCN PAN	kpVCNPAN	Holding the Klarna Payments Virtual Card Network card pan (encrypted)
VCN Holder	kpVCNHolder	Holding the Klarna Payments Virtual Card Network card holder name
Is VCN Used	kplIsVCN	True if VCN is used for payment of the order, otherwise false

PAYMENT TRANSACTION system object has been extended with the following custom attributes:

Parameter Name	Attribute ID	Description
Fraud Status	kpFraudStatus	Klarna Payments order fraud status

3.3 Custom Code

Integration may vary based on the customers' storefront. Site Genesis version 2 is used to demonstrate Klarna Payments integration.

3.3.1 *Integration efforts*

The following steps are needed to complete the integration:

- Install the cartridges
- Change and import 'site-template.zip' (system objects and custom objects)
- Do the required code changes
- Configure your merchant's Klarna Payments Account in the Klarna back office
- Configure Klarna Payments parameters in Commerce Cloud

- Test
- Create styling for Klarna Widget (optional)

These steps should not require more than a few hours.

Note that the name of your storefront site ID should match the name in the ‘site-template.zip’, otherwise you will get import errors.

Also, note that the payment methods ‘Klarna’ and the payment processors ‘KLARNA_PAYMENTS’ will be created or updated from the ‘site-template.zip’.

3.3.2 *Templates modifications*

The following changes should be made regardless of whether a controller or a pipeline integration approach are being used.

To integrate Klarna Payments the following storefront cartridge templates should be updated:

- default/checkout/billing/billing.isml
- default/checkout/summary/summary.isml
- default/checkout/billing/paymentmethods.isml
- default/checkout/shipping/minishipments.isml
- default/components/footer/footer_UI.isml
- default/product/producttopcontentPS.isml
- default/product/productcontent.isml
- default/checkout/cart/cart.isml

default/checkout/summary/summary.isml

Add Code:

```
<isif condition="${session.privacy.KlarnaPaymentsFinalizeRequired}">
    <script><isinclude template="/resources/klarnapaymentsresources.isml"/></script>
        <script type="text/javascript" src="${URLUtils.staticURL('/js/klarna-
payments-finalize.js')}"></script>
```

```

<script src=" https://x.klarnacdn.net/kp/lib/v1/api.js"
async></script>
</isif>

```

Before </isdecorate> closing tag as shown below:

```

214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239

    <form action="${URLUtils'https('COSummary-Submit')}" method="post" class="submit-order">
        <fieldset>
            <div class="form-row">
                <a class="back-to-cart" href="${URLUtils.url('Cart-Show')}">
                    <input value="${Resource.msg('summary.editcart','checkout',null)}" encoding="off" />
                </a>
                <button class="button-fancy-large" type="submit" name="submit" value="${Resource.msg('global.submitorder','locale',null)}" ${Resource.msg('global.submitorder','locale',null)}>
                </button>
            </div>
            <input type="hidden" name="${dw.web.CSRFProtection.getTokenName()}" value="${dw.web.CSRFProtection.generateToken()}"/>
        </fieldset>
    </form>
</div>

<isif condition="${session.privacy.KlarnaPaymentsFinalizeRequired}">
    <script><isinclude template="/resources/klarnapaymentsresources.isml"/></script>
    <script type="text/javascript" src="${URLUtils.staticURL('/js/klarna-payments-finalize.js')}"></script>
    <script src="https://x.klarnacdn.net/kp/lib/v1/api.js" async></script>
</isif>
</isdecorate>

```

default/checkout/billing/billing.isml

Add Code:

```

<script><isinclude template="/resources/klarnapaymentsresources.isml"/></script>
<script type="text/javascript" src="${URLUtils.staticURL('/js/klarna-payments.js')}"></script>
<script src=" https://x.klarnacdn.net/kp/lib/v1/api.js" async></script>

```

Before </isdecorate> closing tag as shown below:

```

@ billing.isml 32
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199<iscomment> ++++++
200 payment methods
201 ++++++
202
203
204 <iscomment>payment method area</iscomment>
205 <isinclude template="checkout/billing/paymentmethods"/>
206 <isbonusdiscountlineitem p_alert_text="${Resource.msg('billing.bonusproductalert','checkout',null)}" p_discount_line_item="${pdict.BonusDiscountLineItem}"/>
207
208
209<div class="form-row form-row-button">
210     <button class="button-fancy-large" type="submit" name="${pdict.CurrentForms.billing.save.htmlName}" value="${Resource.msg('global.continueplaceorder',)}>
211     </button>
212     <input type="hidden" name="${dw.web.CSRFProtection.getTokenName()}" value="${dw.web.CSRFProtection.generateToken()}">
213
214 </form>
215<iscript>
216 importScript("util/ViewHelpers.ds");
217 var addressForm = pdict.CurrentForms.billing.billingAddress.addressFields;
218 var countries = ViewHelpers.getCountryNamesAndRegions(addressForm);
219 var json = JSON.stringify(countries);
220 </iscript>
221 window.Countries = <isprint value="${json}" encoding="off"/></script>
222 <script><isinclude template="/resources/klarnapaymentsresources.isml"/></script>
223 <script type="text/javascript" src="${URLUtils.staticURL('/js/klarna-payments.js')}"></script>
224 <script src="https://x.klarnacdn.net/kp/lib/v1/api.js" async></script>
225 </isdecorate>

```

default/checkout/billing/paymentmethods.isml

Add code:

```
<isif condition="${paymentMethodType.value === 'Klarna'}">hide</isif>
```

After <div class="form-row label-inline" close to line 18 as shown below:

```
paymentmethodisml :: 
1 <iscomment type="text/html" charset="UTF-8" compact="true"/>
2 <iscomment templateName="paymentmethods.isml" />
3 <isinclude template="util/modules">
4<isif condition="${pdict.orderTotal > 0}">
5   <fieldsets>
6     ...
7     <legend>
8       ${Resource.msg('billing.paymentheader','checkout',null)}
9       <div class="dialog-required"> <span class="required-indicator">&#8226; <em>${Resource.msg('global.requiredfield','locale',null)}</em></span></div>
10      </legend>
11    <div class="payment-method-options form-indent">
12      <isloop items="${pdict.CurrentForms.billing.paymentMethods.selectedPaymentMethodID.options}" var="paymentMethodType">
13        ...
14        <iscomment>Ignore GIFT_CERTIFICATE method, GCs are handled separately before other payment methods.</iscomment>
15        <isif condition="${paymentMethodType.value.equals(dw.order.PaymentInstrument.METHOD_GIFT_CERTIFICATE)}"><iscontinue/></isif>
16        ...
17        <div class="form-row label-inline" <isif condition="${paymentMethodType.value === 'Klarna'}">hide</isif>
18          <isset name="radioID" value="${paymentMethodType.value}" scope="page"/>
19          <div class="field-wrapper">
20            <input id="is-$radioID" type="radio" class="input-radio" name="${pdict.CurrentForms.billing.paymentMethods.selectedPaymentMethodID.htmlName}" v
21            ...
22            <label for="is-$radioID"><isprint value="${Resource.msg(paymentMethodType.label,'forms',null)}"/><isif condition="${!empty(dw.order.PaymentMgr.getF
23            ...
24            </div>
25          </isloop>
26        </div>
27      </iscomment>
28    </isinclude template="klarnapayments/klarnapaymentscategories.isml"/>
29  </div>
30</div>
31</div>
32</div>
```

Add Code:

```
<isinclude template="klarnapayments/klarnapaymentscategories.isml"/>
```

After </isloop> Close to line 28 as shown below:

```
paymentmethodisml :: 
1 <iscomment type="text/html" charset="UTF-8" compact="true"/>
2 <iscomment templateName="paymentmethods.isml" />
3 <isinclude template="util/modules">
4<isif condition="${pdict.orderTotal > 0}">
5   <fieldsets>
6     ...
7     <legend>
8       ${Resource.msg('billing.paymentheader','checkout',null)}|
9       <div class="dialog-required"> <span class="required-indicator">&#8226; <em>${Resource.msg('global.requiredfield','locale',null)}</em></span></div>
10      </legend>
11    <div class="payment-method-options form-indent">
12      <isloop items="${pdict.CurrentForms.billing.paymentMethods.selectedPaymentMethodID.options}" var="paymentMethodType">
13        ...
14        <iscomment>Ignore GIFT_CERTIFICATE method, GCs are handled separately before other payment methods.</iscomment>
15        <isif condition="${paymentMethodType.value.equals(dw.order.PaymentInstrument.METHOD_GIFT_CERTIFICATE)}"><iscontinue/></isif>
16        ...
17        <div class="form-row label-inline" <isif condition="${paymentMethodType.value === 'Klarna'}">hide</isif>
18          <isset name="radioID" value="${paymentMethodType.value}" scope="page"/>
19          <div class="field-wrapper">
20            <input id="is-$radioID" type="radio" class="input-radio" name="${pdict.CurrentForms.billing.paymentMethods.selectedPaymentMethodID.htmlName}" v
21            ...
22            <label for="is-$radioID"><isprint value="${Resource.msg(paymentMethodType.label,'forms',null)}"/><isif condition="${!empty(dw.order.PaymentMgr.getF
23            ...
24            </div>
25          </isloop>
26        </div>
27      </iscomment>
28      <isinclude template="klarnapayments/klarnapaymentscategories.isml"/>
29    </div>
30  </div>
31</div>
32</div>
```

Add Code:

```
<iscomment>Klarna Payments
```

```
</iscomment>
```

```
<isinclude template="klarnapayments/klarnapaymentblock.isml"/>
```

Before </fieldset> closing tag, close to line 150 as shown below:

```
paymentmethods.isml
144      <isinputfield formfield="$pdict.CurrentForms.billing.paymentMethods.bml.year" type="select" rowclass="year"/>
125      <isinputfield formfield="$pdict.CurrentForms.billing.paymentMethods.bml.month" type="select" rowclass="month"/>
126      <isinputfield formfield="$pdict.CurrentForms.billing.paymentMethods.bml.day" type="select" rowclass="day"/>
127      <isinputfield formfield="$pdict.CurrentForms.billing.paymentMethods.bml.ssn" type="input"/>
128
129      <div class="bml-terms-and-conditions form-caption">
130          <isinputfield formfield="$pdict.CurrentForms.billing.paymentMethods.bml.termsandconditions" type="checkbox"/>
131          <div>
132              <div class="form-row form-caption">
133                  <isinputfield formfield="$pdict.CurrentForms.billing.paymentMethods.bml.toc" type="checkbox"/>
134              </div>
135          </div>
136      </div>
137
138      <div>
139
140          <iscomment>
141              Custom processor
142          -----
143
144      </iscomment>
145
146      <div class="payment-method <isif condition="${!empty(pdct.selectedPaymentID) && pdct.selectedPaymentID=='PayPal'}>paym
147          <!-- Your custom payment method implementation goes here. -->
148          ${Resource.msg('billing.custompaymentmethod','checkout',null)}
149      </div>
150
151      <div>
152          <iscomment>
153              Klarna Payments
154          -----
155      </iscomment>
156
157      <isinclude template="klarnapayments/klarnapaymentblock.isml"/>
158
159  </fieldset>
160  <div class="gift-cert-used form-indent">
161      <isif condition="$pdct.gcPITotal>0">${Resource.msg('billing.giftcertnomethod','checkout',null)}<iselse>${Resource.msg
162      <isnput type="hidden" name="$pdict.CurrentForms.billing.paymentMethods.selectedPaymentMethodID" value="${pdct.curri
```

default/checkout/shipping/minishipments.isml

Add the following code:

```
<isinclude template="klarnapayments/modules.isml"/>
```

In the end begining of the file as shown below:

```
minishipments.isml
1 <iscontent type="text/html" charset="UTF-8" compact="true"/>
2 <isinclude template="util/modules.isml"/>
3 <isinclude template="klarnapayments/modules.isml"/>
4
5@<iscomment>
6     This template renders a summary of all shipments of the basket which is
7     used below the order summary at the right hand side in the checkout
8     process.
9 </iscomment>
10 <isset name="Shipments" value="$pdict.Basket.shipments" scope="page"/>
11
12 <iscomment>the url to edit shipping addresses depends on the checkout scenario</iscomment>
13 <isset name="editUrl" value="${URLUtils_https('COShipping-Start')}" scope="page"/>
14@<isif condition="$pdict.CurrentForms.multishipping.entered.value">
15     <isset name="editUrl" value="${URLUtils_https('COShippingMultiple-Start')}" scope="page"/>
16 </isif>
17
```

Add the following code:

```
<iskpaddresshelper p_address="${shipment.shippingAddress}" />
```

After <isminicheckout_address p_address="\${shipment.shippingAddress}" /> as as shown below:

```

29         <isif condition="${Shipments.size() > 1 && pdict.Basket.productLineItems.size() > 0}"><div class="name">${Resource.msg('minishipments.name')}</div>
30             <h3 class="section-header">
31                 <isif condition="${shipment.giftCertificateLineItems.size() > 0}">
32                     ${Resource.msg('minishipments.shipping','checkout',null)} <span>${Resource.msg('minishipments.giftcertdeliver')}</span>
33                 <iselseif condition="${shipment.custom.shipmentType == 'instore'}">
34                     <isset name="editUrl" value="${URLUtils_https('Cart-Shov')}" scope="page"/>
35                     <a href="#" class="section-header-note">${Resource.msg('global.edit','locale',null)}</a>
36                     ${Resource.msg('cart.store.instorepickup','checkout',null)}
37                 <iselseif condition="${shipment.shippingAddress != null && pdict.Basket.productLineItems.size() > 0}">
38                     <a href="#" class="section-header-note">${Resource.msg('global.edit','locale',null)}</a>
39                     ${Resource.msg('minishipments.shippingaddress','checkout',null)}
40             </isif>
41         </h3>
42     </div>
43     <div class="details">
44         <iscomment>
45             render the detail section depending on whether this is a physical shipment with products
46             (shipped to an address) or if this is a gift certificate (send via email)
47         </iscomment>
48         <isif condition="${shipment.giftCertificateLineItems.size() > 0}">
49             <isloop items="${shipment.giftCertificateLineItems}" var="giftCertLI">
50                 <div><isprint value="${giftCertLI.recipientName}" /></div>
51                 <div><isprint value="${giftCertLI.recipientEmail}" /></div>
52             </isloop>
53         <iselseif condition="${shipment.shippingAddress != null && pdict.Basket.productLineItems.size() > 0}">
54             <isminishophelper p_address="${shipment.shippingAddress}" />
55             <isif condition="!empty(shipment.shippingMethod)">
56                 <div class="minishipments-method">
57                     <span>${Resource.msg('order.orderdetails.shippingmethod','order',null)}</span>
58                     <span><isprint value="${shipment.shippingMethod.displayName}" /></span>
59                 </div>
60             </isif>
61         </iselseif>
62     </div>
63     </isif>
64 </div>
65
66     </div>
67 </isif>
68 </isloop>
69 </isif>

```

default/components/footer/footer_UI.isml

Add Code:

```
<isinclude template="klarnapayments/scripts.isml"/>
```

Before `<script src="${URLUtils.staticURL('/js/app.js')}"></script>` script tag as shown below:

```

27
28     <isinclude template="klarnapayments/scripts.isml"/>
29
30 <script src="${URLUtils.staticURL('/js/app.js')}"></script>

```

default/product/producttopcontentPS.isml

Add the following code:

```
<isinclude template="klarnapayments/modules.isml"/>
```

```
<iskosmpdp p_product="${psProduct}" />
```

, right under the `pricing` template include as shown below:

```

165         <label>${Resource.msg('product.setpricelabel','product',null)}</label>
166         <isinclude template="product/components/pricing"/>
167
168         <isinclude template="klarnapayments/modules.isml"/>
169         <iskosmpdp p_product="${psProduct}" />
170

```

default/product/productcontent.isml

Add the following code:

```
<isif condition="${!isQuickView}">  
    <isinclude template="klarnapayments/modules.isml"/>  
    <iskosmpdp p_product="${pdict.Product}" />  
</isif>
```

, right under the `pricing` template include as shown below:

```
77    <isinclude template="product/components/pricing"/>  
78  
79@    <isif condition="${!isQuickView}">  
80        <isinclude template="klarnapayments/modules.isml"/>  
81        <iskosmpdp p_product="${pdict.Product}" />  
82    </isif>  
83  
84    <isset name="pam" value="${pdict.Product.getAttributeModel()}" scope="page"/>
```

js/pages/product/variant.js

Add the following code:

```
window.KlarnaOnsiteService = window.KlarnaOnsiteService || [];  
window.KlarnaOnsiteService.push({eventName: 'refresh-placements'});
```

in the variation update callback, on line 35

```
29    |     callback: function () {  
30    |         if (SitePreferences.STORE_PICKUP) {  
31    |             productStoreInventory.init();  
32    |         }  
33    |         image.replaceImages();  
34    |         tooltip.init();  
35    |         window.KlarnaOnsiteService = window.KlarnaOnsiteService || [];  
36    |         window.KlarnaOnsiteService.push({eventName: 'refresh-placements'});  
37    |     }
```

default/checkout/cart/cart.isml

Add the following code:

```
<isinclude template="klarnapayments/modules.isml"/>  
  
<iskosmcart p_lineitemctnr="${pdict.Basket}" />
```

, right after the `<isordertotals>`:

```
820@    <div class="cart-order-totals">  
821        <isordertotals p_lineitemctnr="${pdict.Basket}" p_totallabel="${Resource.msg('global.estimatedtotal','Locale',null)}"  
822            <isinclude template="klarnapayments/modules.isml"/>  
823            <iskosmcart p_lineitemctnr="${pdict.Basket}" />  
824        </div>  
825    ...
```

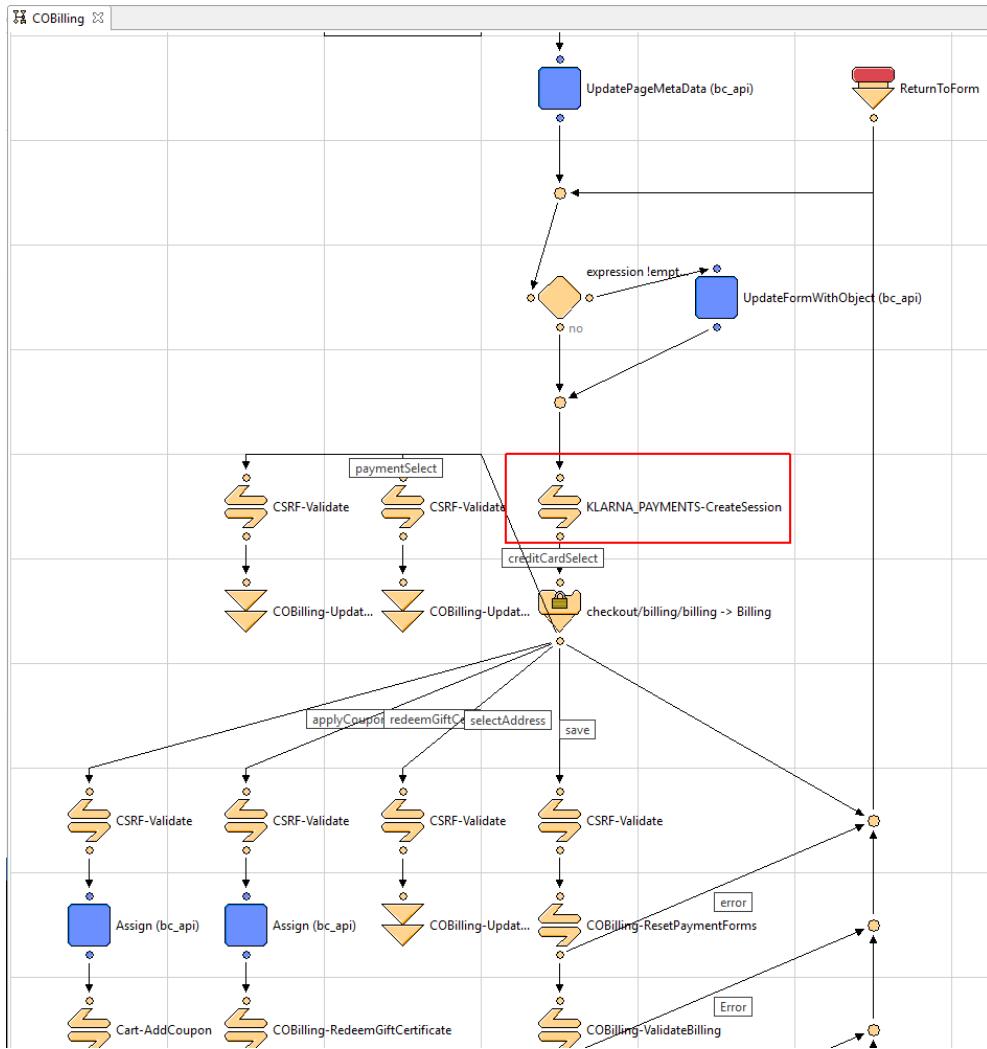
3.3.3 Pipeline modifications

If using a pipeline based SiteGenesis integration, additionally follow the instructions in this chapter. If integrating via the controller based model see next chapter.

To use Klarna Payments cartridge you will need to make the following changes in pipelines.

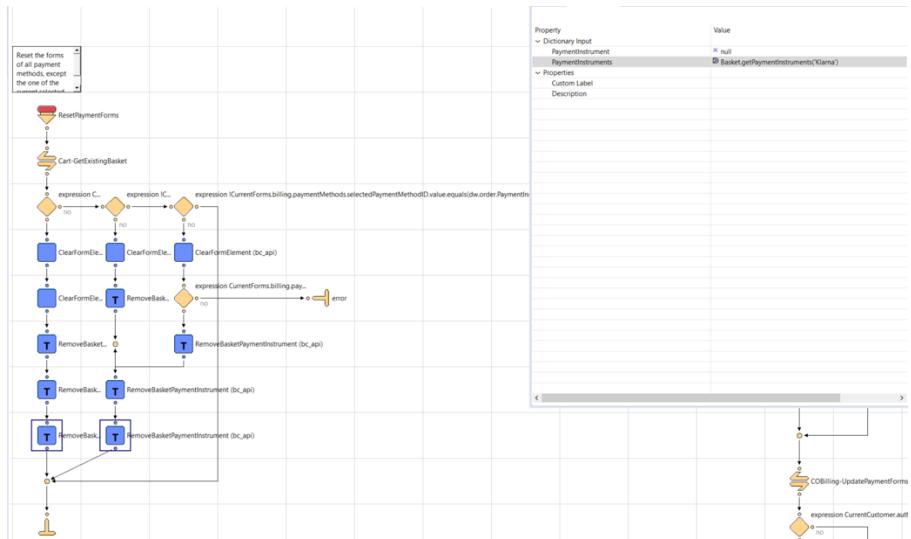
3.3.3.1 COBilling pipeline

Go to COBilling pipeline, Start node and add KLARNA_PAYMENTS-CreateSession call node entry point before checkout/billing/billing -> Billing interaction continue node (see screen shot below)



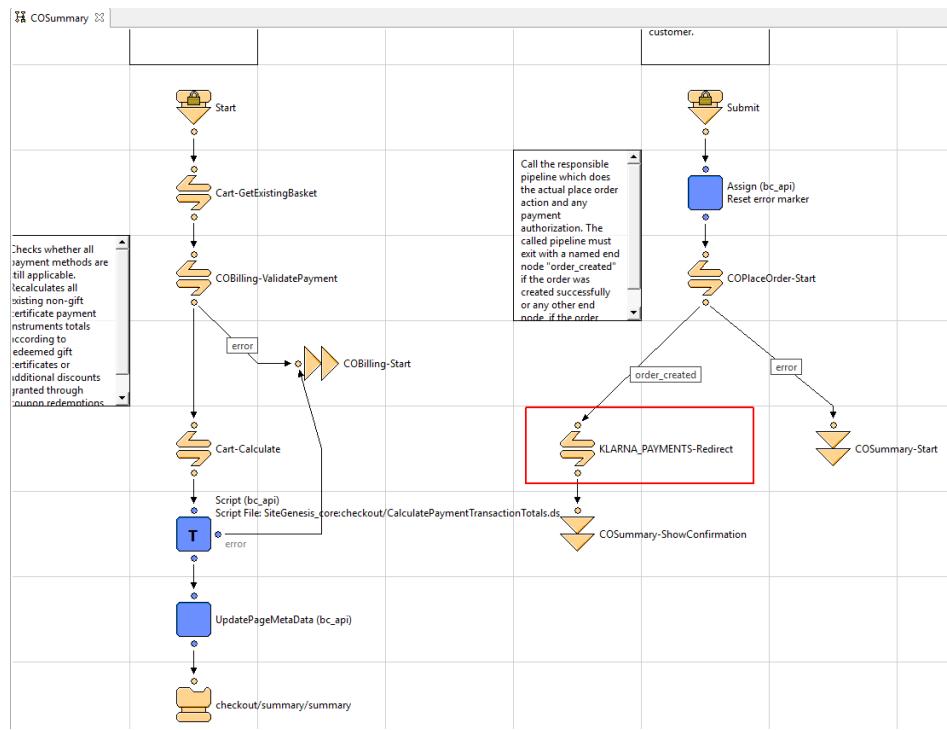
In pipeline COBilling-ResetPaymentForms, add two pipelets
 'com.demandware.pipelet.basket.RemoveBasketPaymentInstrument' at the end of the pipeline.

In Dictionary Input 'PaymentInstruments', set it with value:
 Basket.getPaymentInstruments('Klarna').



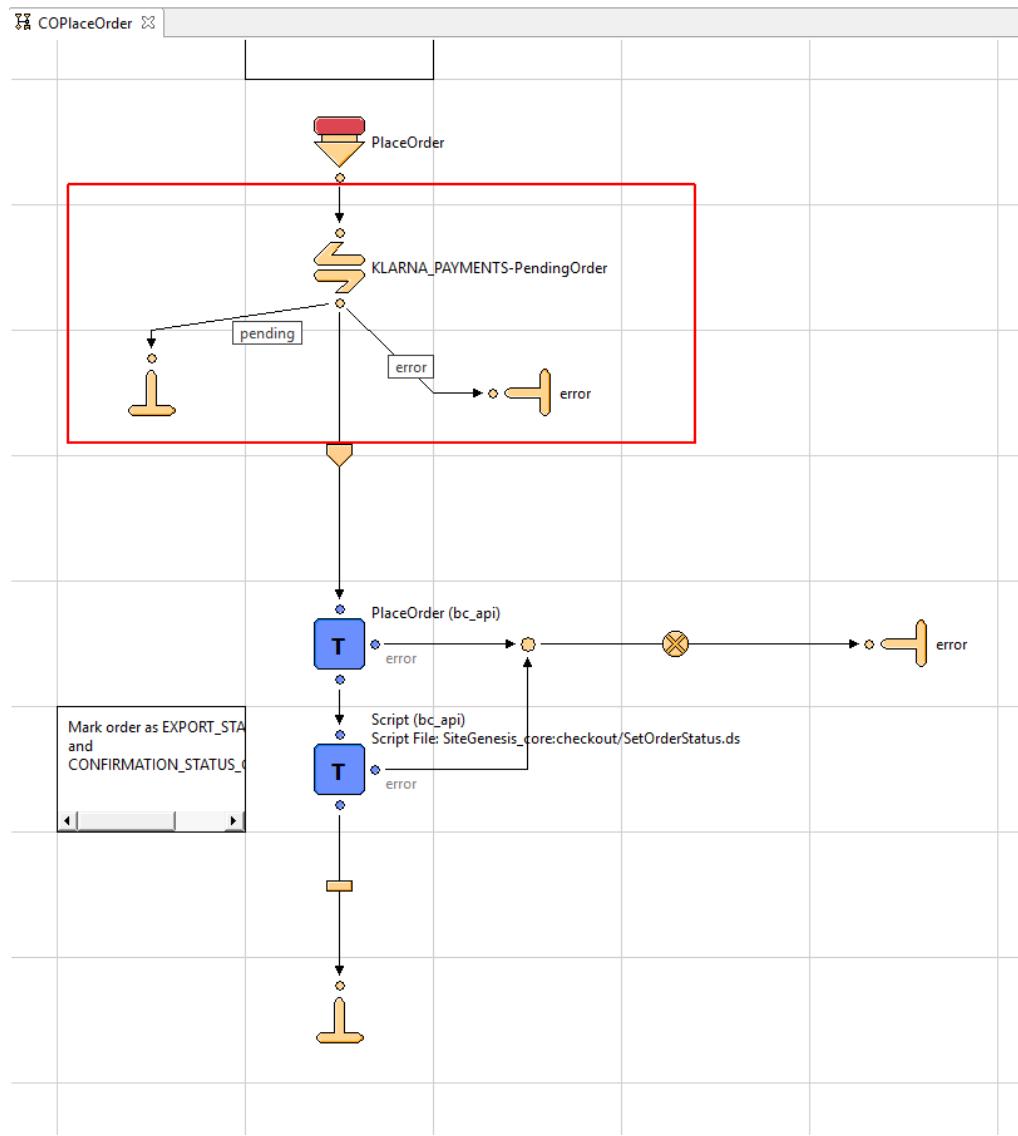
3.3.3.2 *COSummary pipeline.*

Go to COSummary pipeline, Submit node and add KLARNA_PAYMENTS-Redirect call node entry point before COSummary-ShowConfirmation jump node (see screen shot below)



3.3.3.3 COPlaceOrder pipeline.

Go to COPlaceOrder pipeline, PlaceOrder node and add KLARNA_PAYMENTS-PendingOrder call node entry point before PlaceOrder script node node (see screen shot below). Create 'pending' and 'error' transitions and end nodes as shown on the screenshot



3.3.4 Controller modifications

If using a controller based SiteGenesis integration, additionally follow the instructions in this chapter. If integrating via the pipeline method please see Pipeline modifications chapter above.

3.3.4.1 COBilling controller.

Go to COBilling controller, returnToForm method and add the following code block:

```
try
{
```

```

    require('int_klarna_payments_controllers/cartridge/controllers/KLARNA_PAYMENTS.js')
    .CreateSession();
  } catch( e )
  {
    require( 'dw/system/Logger' ).getLogger( 'COBilling.js' ).error( 'Klarna Create Session
Error: {0}', e );
  }

```

after pageMeta.update (see screen shot below)

```

79  function returnToForm(cart, params) {
80    var pageMeta = require('~cartridge/scripts/meta');
81
82    // if the payment method is set to gift certificate get the gift certificate code from the form
83    if (!empty(cart.getPaymentInstrument()) && cart.getPaymentInstrument().getPaymentMethod() === PaymentInstrument.METHOD_GIFT_CERTIFICATE) {
84      app.getForm('billing').copyFrom({
85        giftCertCode: cart.getPaymentInstrument().getGiftCertificateCode()
86      });
87    }
88
89    pageMeta.update({
90      pageTitle: Resource.msg('billing.meta.pagetitle', 'checkout', 'SiteGenesis Checkout')
91    });
92
93    try {
94      require('int_klarna_payments_controllers/cartridge/controllers/KLARNA_PAYMENTS.js').CreateSession();
95    } catch( e ) {
96      require( 'dw/system/Logger' ).getLogger( 'COBilling.js' ).error( 'Klarna Create Session Error: {0}', e );
97    }
98
99    if (params) {
100      app.getView(require(~cartridge/scripts/object').extend(params, {
101        Basket: cart.object,
102        ContinueURL: URLUtils_https('COBilling-Billing')
103      })).render('checkout/billing/billing');
104    } else {
105      app.getView({
106        Basket: cart.object,
107        ContinueURL: URLUtils_https('COBilling-Billing')
108      }).render('checkout/billing/billing');
109    }
110  }
111}

```

In the method resetPaymentForms(), add the command for the three conditions:

```
cart.removePaymentInstruments(cart.getPaymentInstruments('Klarna'));
```

```
function resetPaymentForms() {
```

```
  var cart = app.getModel('Cart').get();
```

```
  var status = Transaction.wrap(function () {
```

```
    if
      (app.getForm('billing').object.paymentMethods.selectedPaymentMethodID.value.equals('Pa
yPal')) {
        app.getForm('billing').object.paymentMethods.creditCard.clearFormElement();
        app.getForm('billing').object.paymentMethods.bml.clearFormElement();

    cart.removePaymentInstruments(cart.getPaymentInstruments(PaymentInstrument.METHO
D_CREDIT_CARD));
```

```
    cart.removePaymentInstruments(cart.getPaymentInstruments(PaymentInstrument.METHO
D_BML));
  
```

```
  // remove Klarna PaymentInstrument
```

```

        cart.removePaymentInstruments(cart.getPaymentInstruments('Klarna'));
    } else if
(app.getForm('billing').object.paymentMethods.selectedPaymentMethodID.value.equals(PaymentInstrument.METHOD_CREDIT_CARD)) {
    app.getForm('billing').object.paymentMethods.bml.clearFormElement();

cart.removePaymentInstruments(cart.getPaymentInstruments(PaymentInstrument.METHOD_BML));
    cart.removePaymentInstruments(cart.getPaymentInstruments('PayPal'));
    // remove Klarna PaymentInstrument
    cart.removePaymentInstruments(cart.getPaymentInstruments('Klarna'));
} else if
(app.getForm('billing').object.paymentMethods.selectedPaymentMethodID.value.equals(PaymentInstrument.METHOD_BML)) {
    app.getForm('billing').object.paymentMethods.creditCard.clearFormElement();

if (!app.getForm('billing').object.paymentMethods.bml.ssn.valid) {
    return false;
}

cart.removePaymentInstruments(cart.getPaymentInstruments(PaymentInstrument.METHOD_CREDIT_CARD));
    cart.removePaymentInstruments(cart.getPaymentInstruments('PayPal'));
    // remove Klarna PaymentInstrument
    cart.removePaymentInstruments(cart.getPaymentInstruments('Klarna'));
}
return true;
});

return status;
}

```

3.3.4.2 COSummary controller.

Go to COSummary controller, submit() method and add the following code block:

```

try
{
require('int_klarna_payments_controllers/cartridge/controllers/KLARNA_PAYMENTS.js').Redirect();
} catch( e )
{
require( 'dw/system/Logger' ).getLogger( 'COsummary.js' ).error( 'Klarna Redirect Error: {0}', e );
}
before showConfirmation(placeOrderResult.Order) (see screen shot below)

```

```

57 /**
58 * This function is called when the "Place Order" action is triggered by the
59 * customer.
60 */
61 function submit() {
62     // Calls the COPlaceOrder controller that does the place order action and any payment authorization.
63     // COPlaceOrder returns a JSON object with an order_created key and a boolean value if the order was created successfully.
64     // If the order creation failed, it returns a JSON object with an error key and a boolean value.
65     var placeOrderResult = app.getController('COPlaceOrder').start();
66     if (placeOrderResult.error) {
67         start({
68             PlaceOrderError: placeOrderResult.PlaceOrderError
69         });
70     } else if (placeOrderResult.order_created) {
71         try {
72             require('int_klarna_payments_controllers/cartridge/controllers/KLARNA_PAYMENTS.js').Redirect();
73         } catch( e ) {
74             require('dw/system/Logger').getLogger( 'COSummary.js' ).error( 'Klarna Redirect Error: {0}', e );
75         }
76     }
77     showConfirmation(placeOrderResult.Order);
78 }
79
80 }
81
82 }
83

```

3.3.4.3 OrderModel.js

Go to OrderModel.js, placeOrder method and add the following code block:

```

if( session.privacy.KlarnaPaymentsFraudStatus === 'PENDING' )
{
    try
    {
        require('int_klarna_payments_controllers/cartridge/controllers/KLARNA_PAYMENTS
.js').PendingOrder();
    } catch( e )
    {
        require('dw/system/Logger').getLogger( 'OrderModel.js' ).error( 'Klarna Payments
Pending Order Error: {0}', e );
    }
    return;
}

```

before var placeOrderStatus = OrderMgr.placeOrder(order); (see screen shot below)

```

1 'use strict';
2
3 /**
4 * Module for ordering functionality.
5 * @module models/OrderModel
6 */
7
8 /* API Includes */
9 var AbstractModel = require('./AbstractModel');
10 var Order = require('dw/order/Order');
11 var OrderMgr = require('dw/order/OrderMgr');
12 var Resource = require('dw/web/Resource');
13 var Status = require('dw/system>Status');
14 var Transaction = require('dw/system/Transaction');
15
16 /**
17 * Place an order using OrderMgr. If order is placed successfully,
18 * its status will be set as confirmed, and export status set to ready.
19 * @param {dw.order.Order} order
20 */
21 function placeOrder(order) {
22
23     if( session.privacy.KlarnaPaymentsFraudStatus === 'PENDING' )
24     {
25         try
26         {
27             require('int_klarna_payments_controllers/cartridge/controllers/KLARNA_PAYMENTS.js').PendingOrder( order );
28         } catch( e )
29         {
30             require('dw/system/Logger').getLogger( 'OrderModel.js' ).error( 'Klarna Payments Pending Order Error: {0}', e );
31         }
32         return;
33     }
34
35     var placeOrderStatus = OrderMgr.placeOrder(order);
36     if (placeOrderStatus === Status.ERROR) {
37         OrderMgr.failOrder(order);
38         throw new Error('Failed to place order.');
39     }
40     order.setConfirmationStatus(Order.CONFIRMATION_STATUS_CONFIRMED);
41     order.setExportStatus(Order.EXPORT_STATUS_READY);
42 }

```


4. Release History

Version	Date	Changes
17.1.0		Initial release
18.1.0		Implement payments endpoint
19.1.0		Added SFRA version.
19.1.1		Updated VCN to use the newest API version
19.1.2		Fix auto capture for the pipeline's cartridge
19.1.3		On-site Messaging updates OC endpoint added
19.1.4		New country locales added. Minor bug fixes. Cartridge templates and forms updated for latest SFRA.