

# Klarna for Salesforce Commerce Cloud

Version 24.4.0 SFRA

Klarna

# Contents

<b>Contents</b>	<b>2</b>
<b>1. Summary</b>	<b>5</b>
<b>2. Klarna Payments</b>	<b>6</b>
2.1. Key features	6
2.2. Klarna Payments cartridge integration	7
2.3. Locales	8
2.4. Use Cases	9
2.4.1. Multiple payment options	9
2.4.2. Klarna Display Conditions and Authorization Handling	10
2.4.3. Authorizing and Placing Klarna Orders in Checkout	11
2.4.4. Refusal of Klarna Payments on Authorization	13
2.4.5. Klarna Payment Option Not Available for Current Purchase - Billing Page	14
2.4.6. Handling Unavailability of Klarna API or Inapplicable Storefronts	15
2.4.7. Notification Handling and Fraud Status Updates	15
2.4.8. Virtual Card Settlements	17
2.4.9. Enabling and Managing Auto-Capture for Payments	23
2.4.10. Widget Customizations	24
2.4.11. Customizing Payment Method Name	25
2.5. Advanced features	27
2.5.1. Klarna Payment Method Based Promotions	27
2.5.2. Price Adjustment Taxation Handling	27
2.5.3. Buy Online, Pickup in Store (BOPIS)	29
2.5.4. Service Rate Limits	30
2.5.5. Klarna Subscriptions	30
2.5.5.1. Configuration	31
2.5.5.2. Cart Page	31
2.5.5.3. Checkout	32
2.5.6. Account Subscription Dashboard	33
2.5.7. Recurring Subscription Order Creation	34
2.6. Klarna Express Button (KEB)	34
2.7. Compatibility	35
2.8. Privacy and Payment	35
2.8.1. GDPR Compliance	35
2.8.2. EMD (Extra Merchant Data)	35

2.8.3. PCI-DSS Compliance	39
<b>3. Conversion Boosters</b>	<b>41</b>
3.1. Klarna On-site Messaging	41
3.1.1. Configuration	41
3.1.1.1. Configuring OSM via KlarnaCountries custom object	41
3.1.1.2. Configuring OSM via site preferences	44
3.2. Klarna Express Checkout	47
3.2.1. Configuration	47
3.2.2. Placements	48
<b>4. Implementation Guide</b>	<b>50</b>
4.1. Setup of Business Manager	50
4.1.1. Cartridge Upload & Assignment	50
4.1.2. Metadata Import	51
4.1.3. Build Klarna JS & CSS	52
4.2. Configuration	52
4.2.1. Configure Klarna Activation Site Preferences for single Klarna API credentials per site	52
4.2.2. Configure Klarna Activation Custom Object for multiple Klarna API credentials on one site	53
4.2.3. Add Account Settings to KlarnaCountries Custom Objects	54
4.2.4. Configure Klarna Payment Custom Preferences	55
4.2.5. Configure Klarna Payment Service	56
4.2.6. Configure Klarna Rate Limited Service Profile	57
4.2.7. Configure Custom Rate Limits	57
4.3. Extended Controllers	58
4.4. Template Updates	58
4.5. Jobs	59
4.5.1. Job "OrderCleanUp" (Optional)	59
4.5.2. Job "RecurringOrders"	63
4.6. Custom Code	64
4.6.1. app_storefront_base\cartridge\client\default\js\checkout\checkout.js [optional]	64
4.6.2. app_storefront_base\cartridge\client\default\js\cart\cart.js [required]	64
4.7. Integration with Other Payment Cartridges	65
4.8. External Interfaces	67
<b>5. Testing</b>	<b>68</b>
<b>6. Operations and Maintenance</b>	<b>69</b>
6.1. Data Storage	69
6.1.1. System Object Extensions	69

6.1.1.1. Basket	69
6.1.1.2. Order	70
6.1.1.3. Order Payment Instrument	71
6.1.1.4. Payment Transaction	72
6.1.1.5. Site Preferences	72
6.1.1.6. Product	80
6.1.1.7. ProductLineItem	81
6.1.1.8. Profile	81
6.1.2. Custom Objects	82
6.1.2.1. Klarna Express Button	82
6.1.2.2. KlarnaCountries	83
6.1.2.3. Klarna Activation	89
6.1.3. Session Attributes & Cookies	90
6.1.4. Library	92
6.1.5. Services	92
6.2. Logs	93
6.3. Availability	93
6.4. Failover/Recovery Process	93
6.5. Support	93
6.5.1. Merchant Support	94
This information helps speed up the investigation and resolution process.	94
<b>7. User Guide</b>	<b>95</b>
7.1. Cartridge Upgrade	95
7.1.1. Upgrade Process	95
7.2. Roles and Responsibilities	96
7.3. Storefront Functionality	96
<b>8. Release History</b>	<b>101</b>
8.1. Known issues	107
<b>9. Additional Information</b>	<b>108</b>
9.1. Klarna API Information	108
9.1.1. Live Environment	108
9.1.2. Testing Environment	108
9.2. Generate Key Pair and Key ID for Virtual Card Settlements (VCN)	108
9.3. Decrypt VCN Card Details	111
9.4. Update KlarnaCountries Definition	114

# 1. Summary

The Klarna cartridge seamlessly integrates with Salesforce Commerce Cloud Storefront, allowing merchants to offer a variety of Klarna products. This guide is tailored for developers, providing detailed instructions on installing and integrating the cartridge into a Salesforce Commerce Cloud site, ensuring full compatibility with the Storefront Reference Architecture (SFRA).

Key features:

- **Klarna Payments:** Offers a variety of payment options at checkout, including Pay Now, Pay Later, and Financing.
- **Express checkout:** Enhances conversion rates by simplifying and speeding up the checkout process.
- **On-site Messaging:** Promotes the availability of Klarna's flexible payment methods throughout the shopping experience.

Integration components:

- **Cartridges:** The integration involves two cartridges - `int_klarna_payments` and `int_klarna_payments_sfra`.
- **Site-template Archive:** Includes new attributes and settings necessary for the integration.
- **Documentation:** This SiteGenesis integration guide provides comprehensive instructions.

To enable Klarna as a payment method, merchants need to configure the cartridge using valid credentials and site configurations within the Commerce Cloud Business Manager. The integration is designed based on the SFRA demo store provided by Commerce Cloud. Merchants are required to sign a contract with Klarna to receive integration support and go-live in production. Klarna provides a playground environment for testing the integration before switching to the production environment. Additionally, Klarna offers assistance with integration and testing prior to the go-live sign-off.

# 2. Klarna Payments

Klarna Payments enables merchants to integrate flexible payment options like Pay Now, Pay Later, and Financing. It supports easy configuration through the Business Manager, enhancing the checkout experience and boosting conversion rates. Advanced features include virtual card settlements and streamlined fraud management for secure and efficient transactions.

## 2.1. Key features

### Integrate Klarna Payments using Best Practices

- Available for international markets including North America, Europe, and Oceania.
- Klarna Payments on SFRA can be configured independently on each site by region.
- Supports multiple Klarna payment methods: Pay Now, Pay Later, and Pay Over Time.
- Quick integration with a virtual card-based approach for settlement.
- Compliant with GDPR (EU) standards for checkout flow.
- Supports multi-shipping addresses.

### Payment and Notification Handling

- Handles notifications for pending status updates (**reject/accept**) for suspected orders after review.
- Supports Klarna authorization with finalization for bank transfer methods (Pay Now).
- Supports Auto-Capture of payments.

### Advanced Features

- BOPIS (Buy Online, Pickup in Store) support, including additional merchant data.
- Supports Klarna payment method-based promotions.
- Supports adjusted price promotions under the Gross Tax Policy.

## 2.2. Klarna Payments cartridge integration

The Klarna cartridge utilizes the Klarna Payments JSON REST API and JavaScript SDK for storefront integration. This integration displays various Klarna payment options via a widget (iframe) embedded on the billing page, known as the Klarna widget.

- **Customer Interaction:** Customers select Klarna as their payment method, review the payment terms, and authorize payment by clicking the "Place Order" button.
- **Order Creation:** Upon authorization, a Klarna order is created, and the customer is redirected to the confirmation page.
- **Fraud Status Management:**
  - Orders with a fraud status of **ACCEPTED** proceed with standard order creation in the SCC.

The screenshot shows the 'Payment' tab of the Order details page in the Billing Module (BM). The URL in the address bar is [Merchant Tools > Ordering > Orders > Order: 00005307\(RefArchGlobal\)](#). The 'Payment' tab is selected, showing the following payment information:

<b>Order Total:</b>	£25.19
<b>Amount Paid:</b>	£0.00
<b>Balance Due:</b>	£25.19
<b>Invoice Number:</b>	00024502
<b>Payment Status:</b>	Paid
<b>Payment Method:</b>	KLARNA_PAYMENTS Processor: KLARNA_PAYMENTS Transaction: 40ab1ccf-1a51-266b-98a2-0d00fb59261d Amount: £25.19
Klarna Payment Category ID: pay_over_time Klarna Payment Category Name: Buy now, pay later <b>Fraud Status: ACCEPTED</b>	

At the bottom left is a link [<< Back to List](#).

Figure 1. Klarna Payment Details in BM

- Orders with a **PENDING** status undergo further review. If later accepted as **FRAUD\_RISK\_ACCEPTED**, the SCC order status updates accordingly. [Figure 2]

Merchant Tools > Ordering > Orders > Order: 00005338(RefArchGlobal)

General Attributes **Payment** Notes History

**Payment Information for Order '00005338'**

Order Total:	£191.99
<u>Amount Paid:</u>	£0.00
Balance Due:	£191.99
Invoice Number:	00024524
<u>Payment Status:</u>	Paid
Payment Method:	KLARNA_PAYMENTS Processor: KLARNA_PAYMENTS Transaction: 08ae0f0-22f0-2138-9334-978b637d74dc Amount: £191.99
Klarna Payment Category ID: pay_over_time Klarna Payment Category Name: Buy now, pay later <b>Fraud Status: FRAUD_RISK_ACCEPTED</b>	

<< Back to List

Figure 2. Klarna Payment Details in BM

## Key Attributes and Business Manager Details

- Custom Attribute for Payment Status:** The fraud status of Klarna Payments is stored in a custom attribute `kpFraudStatus` within the `PaymentTransaction` system object.
- Business Manager View:** This status is visible in the Business Manager on the order details Payment tab.

### 2.3. Locales

The Klarna Payments cartridge supports multiple locales, ensuring a seamless experience for a global customer base. Supported locales include:

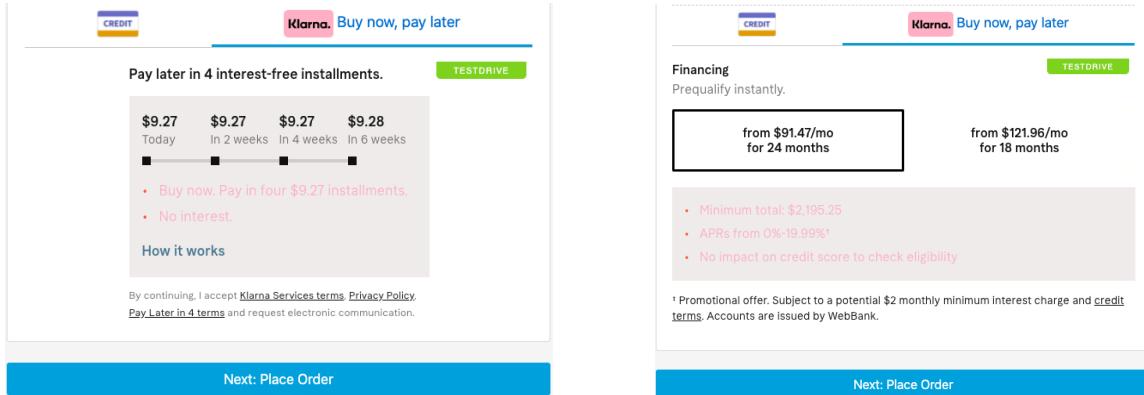
- |           |           |           |
|-----------|-----------|-----------|
| • English | • French  | • Spanish |
| • Danish  | • German  | • Swedish |
| • Dutch   | • Italian |           |
| • Finnish | • Polish  |           |

For a comprehensive list of the 25+ supported locales, please check [Klarna's technical docs site](#).

## 2.4. Use Cases

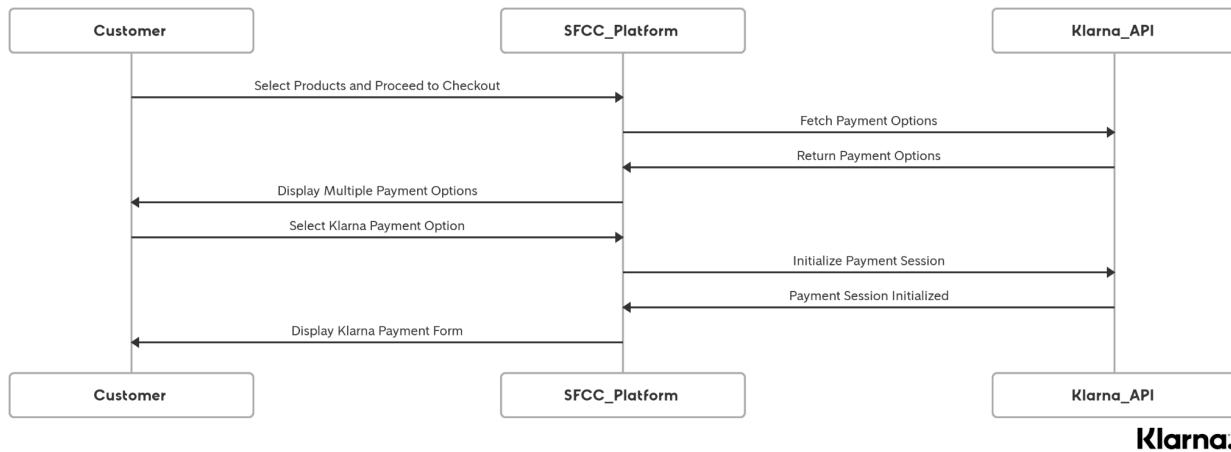
### 2.4.1. Multiple payment options

During the checkout billing step, Klarna payment options are dynamically displayed based on the customer's cart details and the payment method categories available for the current Klarna session.



**Figure 3. Example of payment options displayed at checkout**

When the customer selects a payment method, a widget displaying additional information about the Klarna product will appear.



**Figure 4. Klarna payment options retrieval**



**Note:** The payment methods shown depend on market availability and the contractual agreement with Klarna.

## 2.4.2. Klarna Display Conditions and Authorization Handling

### Display Conditions

Klarna payment methods may not be displayed under the following conditions:

1. **Payment Categories Not Returned:** Klarna is not displayed if payment categories are not returned in the session response.
2. **Session Linking Issue:** Payment method content is not displayed if we have payment methods in the session, but no session is linked to the basket.
3. **Authorization Denial:** Payment method category is hidden or grayed out if `show_form` is returned as false in the Klarna response on an authorization request and the payment method is denied.
4. **Session Creation Failure:** Overall, if Klarna is unavailable and a session is not created, payment categories will not be displayed.

### Authorization Handling Preference



#### This functionality is deprecated as of release 24.4.0

By default no action will be applied upon authorization rejections. No configuration changes required.

We've implemented a site preference, `kpRejectedMethodDisplay`, allowing merchants to choose how to handle authorization rejections (refer to section 2.3.5 Refusal of Klarna Payments on Payment Method – Authorization). The available options are:

- **No action:** No specific handling for authorization rejections.
- **Hide:** Hide the payment method upon authorization rejection.
- **Gray out:** Gray out the payment method upon authorization rejection.

### 2.4.3. Authorizing and Placing Klarna Orders in Checkout

The cartridge implementation follows best practices for integrating Klarna payment methods, offering the inclusion of Extra Merchant Data (EMD) to optimize acceptance rates. EMD can include customer information and Buy Online, Pickup in Store (BOPIS) details when the custom site preference `attachments` is enabled. Merchants should review and validate EMD data based on their data privacy requirements prior to going live.

#### Authorization Process

When a customer selects a Klarna payment method and clicks "Continue to Place Order," their personally identifiable information (PII) is sent. This information is essential for assessing and verifying the customer's data to display available payment method options.

Steps:

1. **Customer Action:** The customer selects a Klarna payment method and clicks "Continue to Place Order."
2. **Authorization Initiation:** Authorization is initiated, sending PII to Klarna for verification.
3. **Authorization Outcome:**
  - a. **Success:** The customer is taken to the summary page.
  - b. **Additional Information:** Depending on the payment method and market, additional information may be requested before completing authorization.
  - c. **Failure:** If authorization is not successful (`approved = false`), the customer remains on the billing page, no changes to payment options display.

#### Special Case: Pay Now

For the Pay Now payment category, an additional "finalize" call is required when the customer clicks the "Place Order" button on the review page. This ensures that funds are transferred only when the customer decides to place the order. A cookie called `selectedKlarnaPaymentCategory` is set in the customer's browser to initialize the "finalize" call to Klarna.

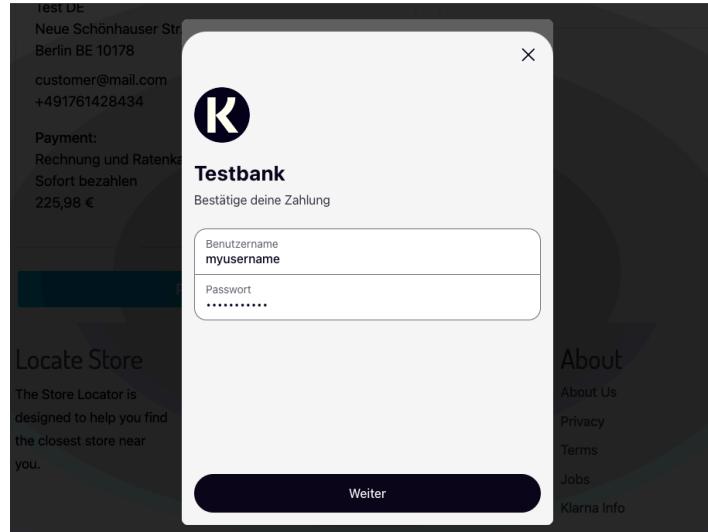


Figure 5. Finalize Call Screen

## Klarna Order Placement

The Klarna order is placed just before the SFCC order is created:

- **Approved:** Successful payment authorizations (Klarna Payments authorization status: APPROVED) result in the creation of both Klarna and SFCC orders with a "Paid" order payment status and "Ready for Export" export status.
- **Rejected or Pending:** Refused and pending Klarna Payment orders (statuses REJECTED and PENDING) result in a "Not Paid" order payment status and "Not Exported" export status.
- **VCN Settlement Errors:** Klarna orders are canceled, and SFCC orders are set to a "Not Exported" export status.

## Changing Payment Methods

If a customer authorizes a Klarna payment and then returns to the billing page to select a different non-Klarna payment method, an automatic cancelAuthorization call is triggered. This releases the authorized funds and frees up the available purchase amount. Merchants can utilize this function for specific use cases.

```
server.get('SaveAuth', function (req, res) {
    var KlarnaSessionManager =
        require('*-/cartridge/scripts/common/klarnaSessionManager');
```

```

// var processor = require('*-/cartridge/scripts/payments/processor');

var token = req.httpHeaders['x-auth'];
var finalizeRequired = req.httpHeaders['finalize-required'];

//Cancel any previous authorizations
//processor.cancelAuthorization();

var klarnaSessionManager = new KlarnaSessionManager();
klarnaSessionManager.saveAuthorizationToken(token, finalizeRequired);

res.setStatusCode (200);
});

```

## Testing

MERCHANTS should thoroughly test the checkout flow, including scenarios such as:

- Klarna session flow
- Payment method switching
- Order amount updates
- Checkout with external payment methods

### 2.4.4. Refusal of Klarna Payments on Authorization



#### This functionality is deprecated as of release 24.4.0

By default no action will be applied upon authorization rejections. No configuration changes required.

Upon selecting Klarna as the payment method on the billing step of checkout and based on the customer information provided, Klarna Payments can be refused as a payment method. If the payment method is rejected with `show_form=false` and `approved=false` (i.e., hard reject), the merchant can choose how the payment option is displayed on the Billing page using the Business Manager (BM) preference **Hide Payment Methods on Deny** (`kpRejectedMethodDisplay`):

- **No:** Leave the payment visible to the customers.

- **Hide:** The payment option will be hidden from the customer.
- **Gray Out:** The payment option will be grayed out and not clickable.

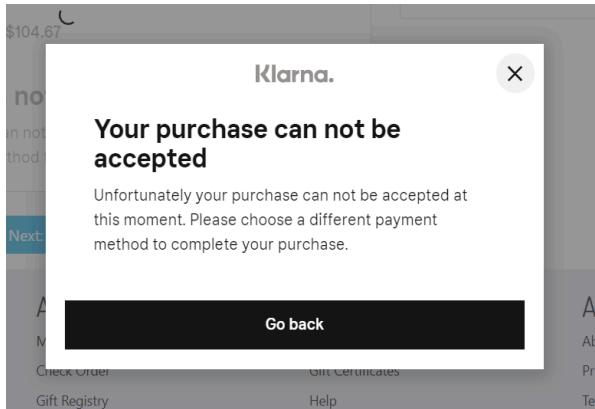


Figure 6. Denied order popup

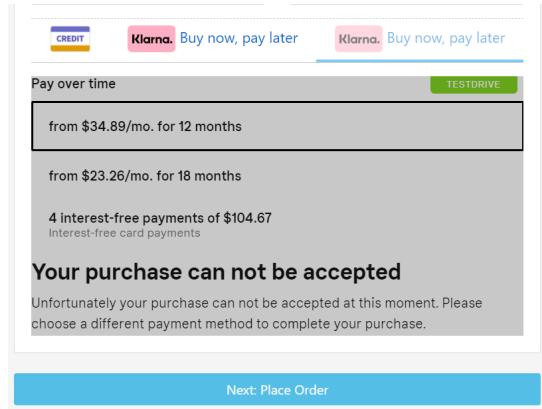


Figure 7. Greyed Out Payment Option



**Note:** Reloading the page will show the denied Klarna payment method again.

#### 2.4.5. Klarna Payment Option Not Available for Current Purchase - Billing Page

The customer is presented with an appropriate message in the Klarna widget when the customer attempts to choose Klarna in the billing step.

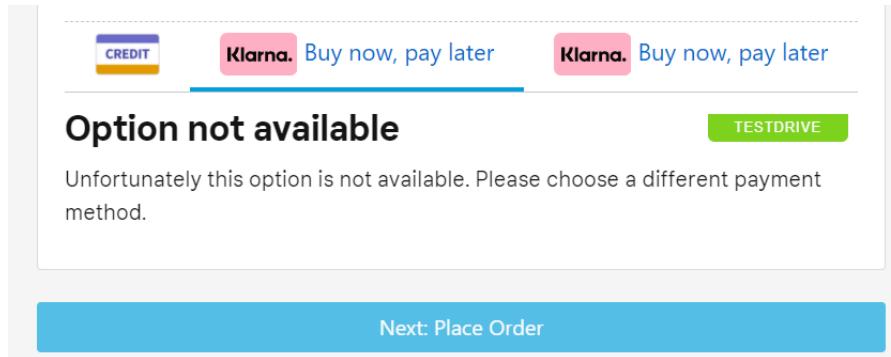


Figure 8. Option Not Available

## 2.4.6. Handling Unavailability of Klarna API or Inapplicable Storefronts

If the Klarna API is unavailable or the site/storefront is inapplicable, Klarna will not be presented as a payment option on the billing page. It is recommended that a Klarna session not be created when a customer selects a non-Klarna market or merchant store. This also applies to cases where merchants have multi-currency storefronts with a basket currency not supported by Klarna.

## 2.4.7. Notification Handling and Fraud Status Updates

When a Klarna order is created but flagged for additional review, the Commerce Cloud order remains in the `Created` status with a Fraud Status of `PENDING`. This order is marked with `EXPORT_STATUS_NOTEEXPORTED`, confirmation status `NOTCONFIRMED`, and `NOTPAID`.

### Notification Process

For orders with a Fraud Status of `PENDING`, updates are sent to the pre-configured `notification_url` on the merchant's Commerce Cloud site once the review is complete. The updated Fraud Status, such as `FRAUD_RISK_ACCEPTED`, is displayed in Business Manager (BM). Push notifications are sent repeatedly (up to 24 hours, every 10 minutes) until the POST request is acknowledged with a `200` response.

### Event Types

Klarna sends one of the following event types in the notification to SFCC to update the risk status:

- `FRAUD_RISK_ACCEPTED`
- `FRAUD_RISK_REJECTED`
- `FRAUD_RISK_STOPPED`

Notifications are generally received within 4-24 hours. The order's payment transaction is updated accordingly (see `kpFraudStatus`). This status can be viewed in BM on the order details Payment tab.

Merchant Tools > Ordering > Orders > Order: 00005338(RefArchGlobal)

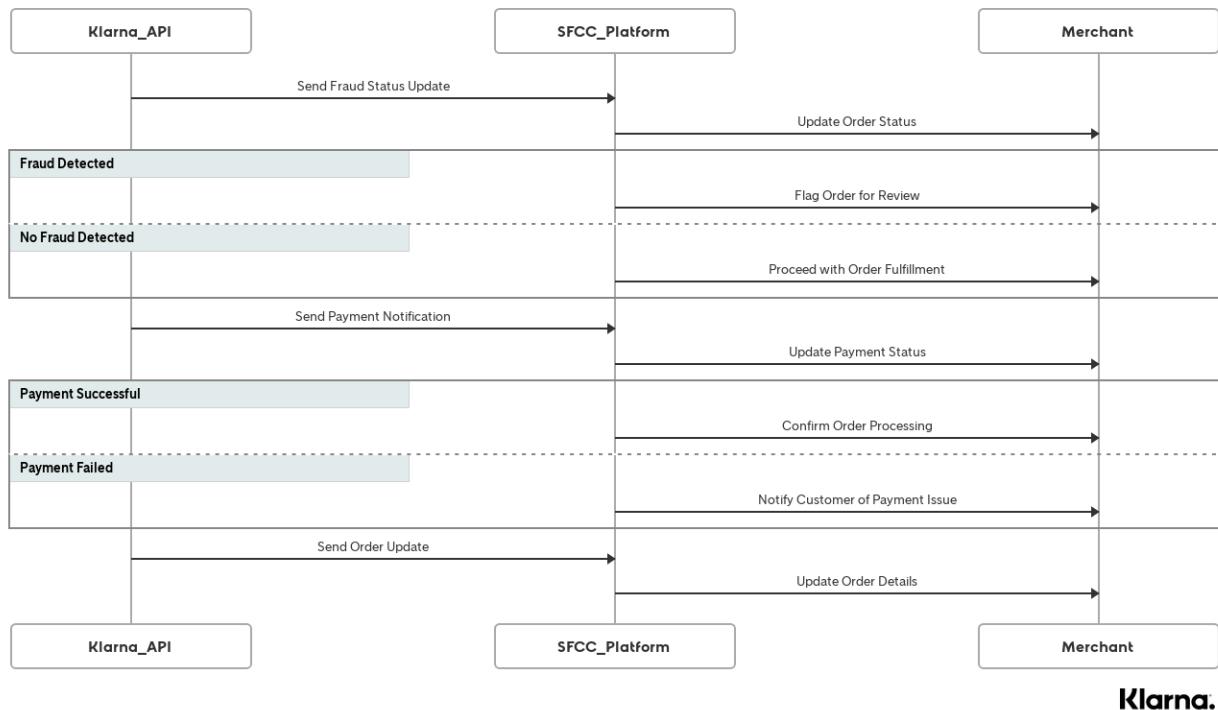
General Attributes **Payment** Notes History

**Payment Information for Order '00005338'**

Order Total:	£191.99
Amount Paid:	£0.00
Balance Due:	£191.99
Invoice Number:	00024524
Payment Status:	Paid
Payment Method:	KLARNA_PAYMENTS Processor: KLARNA_PAYMENTS Transaction: 08ae0f0-22f0-2138-9334-978b637d74dc Amount: £191.99
Klarna Payment Category ID: pay_over_time Klarna Payment Category Name: Buy now, pay later <b>Fraud Status: FRAUD_RISK_ACCEPTED</b>	

<< Back to List

**Figure 9. Fraud Status in Payment Details**



**Figure 10. Notification handling and fraud status updates**

## Handling Fraud Risk Updates

- **FRAUD\_RISK\_ACCEPTED**
  - **Order Status:** Changes to **OPEN**.
  - **Confirmation Status:** Changes to **CONFIRMATION\_STATUS\_CONFIRMED**.

- **Export Status:** Changes to EXPORT\_STATUS\_READY.
- **Auto-Capture Orders:** The payment status will be set to PAYMENT\_STATUS\_PAID, and the full order amount will be captured.
- FRAUD\_RISK\_REJECTED or FRAUD\_RISK\_STOPPED
  - **Order Status:** Marked as failed (FAIL) in SFCC.



**Note:** The Klarna pending functionality availability is dependent on markets and enabled based on the contractual agreement with Klarna.

#### 2.4.8. Virtual Card Settlements

This feature is disabled by default. However, if standard order management is not suitable for your Klarna integration, you can utilize Klarna's Merchant Card Service-based virtual card solution. The virtual card is issued against a Klarna order to capture the authorized order amount using standard card rails.

When a customer places an order, it is first booked in SFCC. Once Klarna accepts the order, the Klarna cartridge integration creates a virtual card-based settlement using the Merchant Card Services (MCSv3) API.

After the settlement is created (virtual card returned), the encrypted card details are saved in SFCC. These details can later be used by the merchant's OMS platform or custom PSP integration to authorize the virtual card until the Klarna order is valid. On successful order fulfillment, the funds on the virtual card may be captured. (For delays in capture or special use cases, please speak with the Klarna Key Account Manager in advance). While Klarna is the original payment method, the order amount will be settled with the merchant using the issued virtual credit card instead of a direct bank account transfer.

Please review the procedures with the Klarna team for details of the settlement process using virtual cards. Refer to the code in scripts/payments/processor.js and update accordingly to the integrated cards processor.

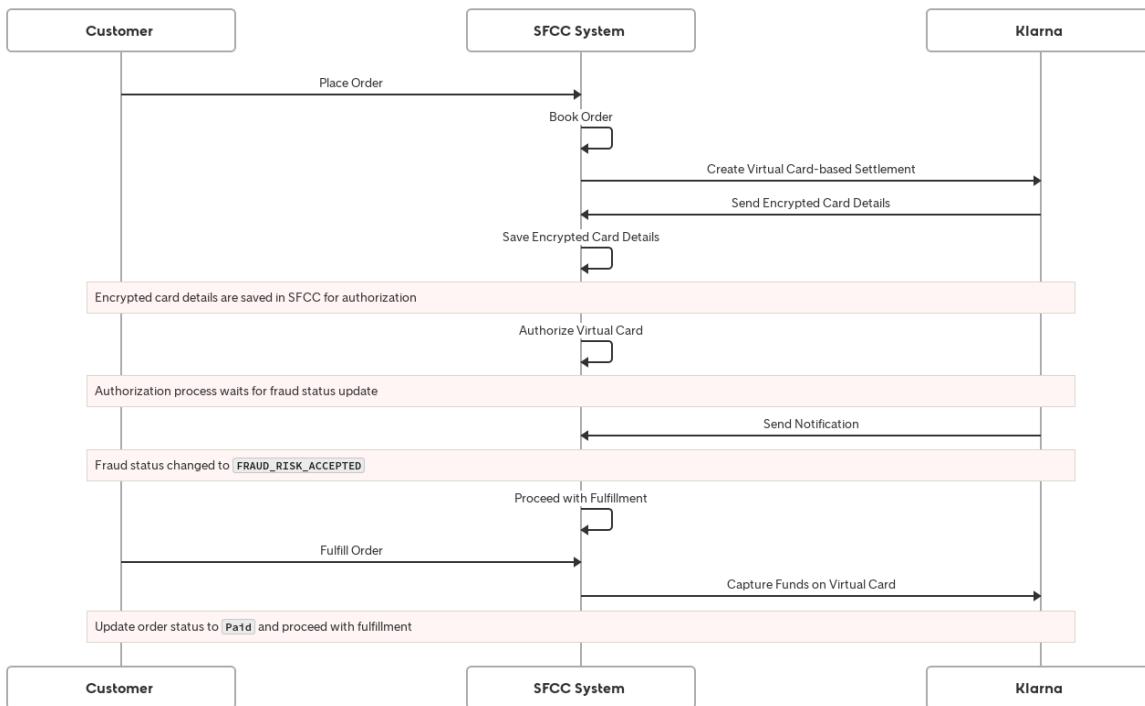


Figure 11. VCN Process in Klarna Payments Cartridge



**Note:** If the Klarna 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. The virtual card issued is limited to 1 single successful authorization per order for a given MID. For decrypting the credit card details refer to section [Decrypt VCN Card Details](#).

Credit card authorization call:

```

/**
 * Call Credit Card Authorization Hook (for VCN settlement)
 * @param {dw.order.order}.order DW Order
 * @returns {processorResult} authorization result
 */

function callCreditCardAuthorizationHook(order) {
    var processorResult = null;
    var paymentInstrument = order getPaymentInstruments(PAYMENT_METHOD)[0];
    var paymentProcessor = PaymentMgr
  
```

```

        .getPaymentMethod(paymentInstrument.paymentMethod)
        .paymentProcessor;
    var transactionID =
        paymentInstrument.getPaymentTransaction().getTransactionID();

    var hook = 'app.payment.processor.' + CREDIT_CARD_PROCESSOR_ID;
    if (!HookMgr.hasHook(hook)) {
        throw new Error('File of app.payment.processor.' +
CREDIT_CARD_PROCESSOR_ID + 'hook is missing or the hook is not configured'
    }

    processorResult = HookMgr.callHook('app.payment.processor.' +
CREDIT_CARD_PROCESSOR_ID, 'Authorize', transactionID, paymentInstrument,
paymentProcessor);
    return processorResult;
}

```

If enabled and fully configured, a virtual card settlement request is made successfully. For orders placed with the VCN settlement option, the related custom attributes are shown below:

[Merchant Tools](#) > [Ordering](#) > [Orders](#) > Order: 00049309(RefArch)

General    **Attributes**    Payment    Notes    History

## Attributes for Order '00049309'

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

Klarna Payments Order ID: 72bf2c96-6523-2add-8c50-f2af87712019

Is VCN Used:

VCN Card ID: befbb0e9-5e98-4e39-9c00-75aba0c3372b

[<< Back to List](#)

**Figure 12. VCN Details in Order**

If required, the additional virtual card details can be assigned to this group in **Administration > Site Development > System Object Types > select "Order"**. In the Attribute Grouping tab, select **Klarna\_Payments** and click on "edit". Assign the new attributes and save the data.

[Administration](#) > [Site Development](#) > [System Object Types](#) > [Order - Attribute Groups](#) > Klarna Payments

## Object Type 'Order' - Attribute Definition Assignments

On this page you can assign existing attribute definitions to your attribute group.

Assign Attribute Definition

ID:		Add			
Select All	ID	Name	Type	Attribute Settings	Sorting
<input type="checkbox"/>	<b>kpOrderID</b>	Klarna Payments Order ID	String		
<input type="checkbox"/>	<b>kplVCN</b>	Is VCN Used	Boolean		
<input type="checkbox"/>	<b>kpVCNCardID</b>	VCN Card ID	String		
<input type="checkbox"/>	<b>kpVCNHolder</b>	VCN Holder	String		
<input type="checkbox"/>	<b>kpVCNBrand</b>	VCN Brand	String		
<input type="checkbox"/>	<b>kpVCNPcidata</b>	VCN PCI Data	String		
<input type="checkbox"/>	<b>kpVCNIV</b>	VCN Initialization Vector	String		
<input type="checkbox"/>	<b>kpVCNAESKey</b>	VCN AES Key	Text		

[Unassign](#)

[<< Back](#)

**Figure 13. Full List of VCN Attributes**

Work with your Klarna Account Manager and Delivery contact in advance to select the appropriate virtual card product based on your business requirements and use cases. You can find information [here](#) about other supported use cases.



### Important!

#### DO NOT SAVE DECRYPTED PCI DATA ON THE SERVER

It is the responsibility of the merchant to ensure PCI-DSS compliance and to ensure the card data is handled securely in coordination with required partners/Payment Service Provider/Acquirer. Please review in advance the order export details required for virtual card-based Klarna orders. Any historical decrypted PCI data should also be expunged, regardless of the validity date (see the ["OrderCleanUp"](#) job section).

Depending on their credentials configuration, merchants should follow these steps to utilize the virtual card integration option:

**For single credentials per site configuration**

1. Enable VCN option in Site Preferences as shown below.

The screenshot shows the Site Preferences page in the Klarna interface. The top navigation bar includes 'RefArchGlobal', 'Merchant Tools', 'Administration', 'Storefront', 'Toolkit', and a user icon. The main content area displays a table with three columns: 'Name', 'Value', and 'Default Value'. A specific row for 'Enable virtual card number (VCN)' is highlighted with a purple border. The 'Value' field contains 'Yes', and the 'Default Value' field contains 'No'. A tooltip for the 'Value' field states: 'When set to "Yes," each Klarna order will automatically generate a Virtual Card Number upon settlement request.' An 'Edit Across Sites' link is also present. Other settings listed include 'VCN - Public Key ID' and 'VCN - Enable settlement retry'.

Name	Value	Default Value
Enable virtual card number (VCN) (kpVCNEnabled)	Yes	No
VCN - Public Key ID (kpVCNkeyId)	6c5b99c5-b0de-4689-b569-1ae12ec898e	UUIDv4 value corresponding to the key pair. Shared with Klarna respectively for Production & Playground (test) environment.
VCN - Enable settlement retry (kpVCNRetry)	None	No

**Figure 14. VCN enablement setting**

2. Enter the VCN Public Key ID. Unique UUIDv4 value, which should be different for playground testing and Production (live site).

This screenshot shows the same Site Preferences page as Figure 14, but the 'Enable virtual card number (VCN)' setting has been changed to 'No'. The 'VCN - Public Key ID' setting is highlighted with a purple border. The 'Value' field contains '6c5b99c5-b0de-4689-b569-1ae12ec898e', and the 'Default Value' field contains 'No'. A tooltip for the 'Value' field states: 'UUIDv4 value corresponding to the key pair. Shared with Klarna respectively for Production & Playground (test) environment.' An 'Edit Across Sites' link is also present. The other settings remain the same as in Figure 14.

Name	Value	Default Value
Enable virtual card number (VCN) (kpVCNEnabled)	No	No
VCN - Public Key ID (kpVCNkeyId)	6c5b99c5-b0de-4689-b569-1ae12ec898e	UUIDv4 value corresponding to the key pair. ...
VCN - Enable settlement retry (kpVCNRetry)	None	No

**Figure 15. VCN Public Key ID**

3. Generate a 4096-bit RSA key pair (Refer to section [Generate Key Pair and Key Id for Virtual Card Settlements](#)).

4. Update the VCN settlement retry setting. By default, this is disabled. However, if enabled and in cases when a VCN creation error is returned, the application will retry the settlement request once again with `order_id` as the idempotency key.
5. Send the generated unique `key_id` + `public key` combination in JWK format to Klarna before testing and going live. It will be used to encrypt the AES key, which encrypts the PCI data on Klarna's side when the settlement request is made. After confirmation from Klarna that the key has been successfully added to your merchant profile, you will be able to use the virtual card-based settlement option for Klarna payment methods.

### For different credentials per site - (Using `KlarnaActivation` custom object)

1. Go to **Merchant Tools > Custom Objects > Manage Custom Objects** and select `KlarnaActivation`.
2. Search and locate the custom object entry.
3. Enable the VCN option.
4. Enter the VCN Public Key ID, a unique UUIDv4 value, which should be different for Playground (testing) and Production (live site) environments.
5. Update the VCN settlement retry setting. By default, this is disabled. If enabled, in cases when a VCN creation error is returned, the application will retry the settlement request once again with `order_id` as the idempotency key.

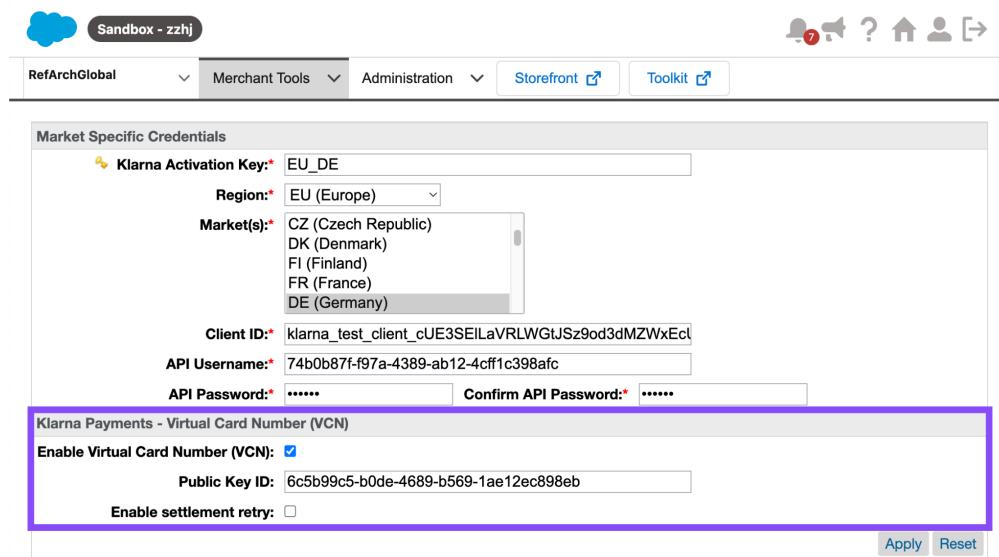


Figure 16. VCN settings in Klarna activation custom object

6. Generate a 4096-bit RSA key pair (Refer to the section "Generate Key Pair and Key Id for Virtual Card Settlements").

- Send the generated unique `key_id + public key` combination in JWK format to Klarna before testing and going live. It will be used to encrypt the AES key, which encrypts the PCI data on Klarna's side when the settlement request is made. After confirmation from Klarna that the key has been successfully added to your merchant profile, you will be able to use the virtual card-based settlement option for Klarna payment methods.

#### 2.4.9. Enabling and Managing Auto-Capture for Payments

Auto-capture is enabled via a site preference `kpAutoCapture` located in the `Klarna_Payments` preference group. By default, auto-capture is disabled. When enabled, the system attempts to capture the full payment amount automatically.

The screenshot shows the Klarna Merchant Portal interface. At the top, there are navigation tabs for 'Payments' and 'Conversion boosters'. Below the tabs, the order details are displayed for Ref #1: 00001502. The order contains two items: a 'Sleeveless Cowl Neck Top' and 'Ground'. On the right side, there is a 'Customer' section with shipping and billing address details. At the bottom left, a purple box highlights the 'Activity log (2)' section, which lists two events: 'Merchant captured 2 items for: €53.50' (21 Jun 2024, 11:01, via API) and 'Order created by Klarna: €53.50' (21 Jun 2024, 11:01). The URL <http://demo.klarnatest.com/K500726> K500726 is visible at the bottom of the page.

Figure 17. Order Details in Klarna Portal

#### Successful Capture

- The SFCC order's payment transaction is marked as `Paid`.
- The transaction status is viewable in the Business Manager.
- The order status in Klarna's Merchant Portal is updated to `Captured`.

## Unsuccessful Capture

- An error is logged in the custom error log.
- The issue must be reviewed with the Klarna delivery team before testing and going live.



### Note:

- Auto-capture is possible only for orders where Virtual Card Number (VCN) is not enabled.
- Ensure the setting is reviewed with the Klarna delivery team before testing and going live.

### 2.4.10. Widget Customizations

The Klarna Payments widget can be styled to align with the marketing and branding needs of the merchant's store. Various graphic elements of the widget can be customized through site preferences.

#### Customizable Graphic Elements



### This functionality is partly deprecated as of release

#### 24.4.0

Widget customization preferences must be entered using the new JSON object `kpColorCustomisation` instead of separate attributes.

For more details, refer to the [Klarna Payments API documentation](#).

Below is a table of the graphic elements that can be customized along with their corresponding site preferences:

Element	Site Preference deprecated with release	New site preference since 24.4.0 release	Example Value
---------	--	---	------------------

<b>24.4.0</b>			
Details Color	<code>kpColorDetails</code>	<code>kpColorCustomization</code>	#COFFEE
Button Color	<code>kpColorButton</code>	Enter the needed customizations in the JSON object, make sure to follow the <a href="#">options object</a> specified in the Klarna documentation	#COFFEE
Button Text Color	<code>kpColorButtonText</code>		#COFFEE
Checkbox Color	<code>kpColorCheckbox</code>		#COFFEE
Checkbox Checkmark Color	<code>kpCheckboxCheckmark</code>		#COFFEE
Header Color	<code>kpColorHeader</code>		#COFFEE
Link Color	<code>kpColorLink</code>		#COFFEE
Border Color	<code>kpColorBorder</code>		#COFFEE
Selected Border Color	<code>kpBorderSelected</code>		#COFFEE
Text Color	<code>kpColorText</code>		#COFFEE
Secondary Text Color	<code>kpColorTextSecondary</code>		#COFFEE
Border Radius	<code>kpRadiusBorder</code>		0px

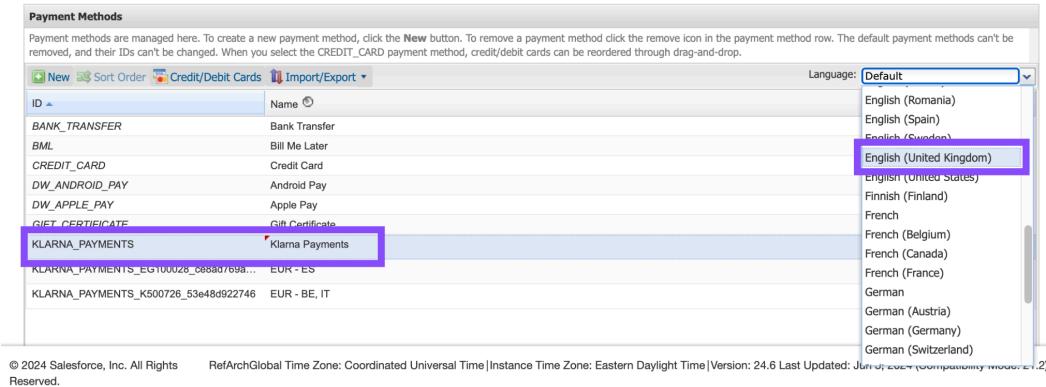
**Table 1. Klarna Payments widget customization elements**

## 2.4.11. Customizing Payment Method Name

The payment method name “Klarna Payments” can be customized in the Business Manager.

### Steps to Customize

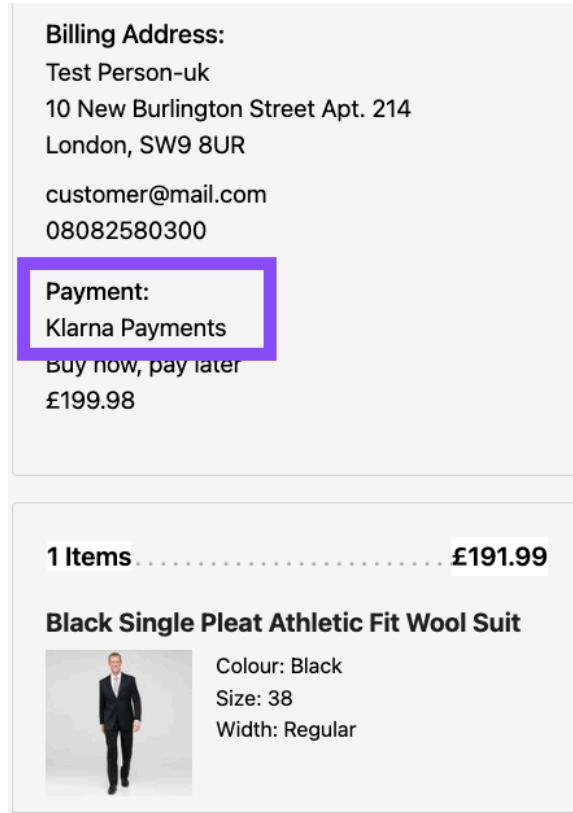
1. Navigate to Merchant Tools > Ordering > Payment Methods in Business Manager.
2. Select the Klarna payment method.
3. Choose the desired language from the drop-down menu.



**Figure 18. Customize Payment Name**

The customized payment method name will be displayed in:

- The mini summary and confirmation screens
- Confirmation emails
- The My Account Order Details section



**Figure 19. Payment Method Name in confirmation screen**

## 2.5. Advanced features

### 2.5.1. Klarna Payment Method Based Promotions

Starting with the B2C 20.7 release, merchants can use payment methods as qualifiers for product, order, and shipping promotions.

By default, when a promotion is configured to use a payment method as a qualifier, the total order amount is shown to the customer upon reaching the review page. This can cause the Klarna authorization call to be made for a higher amount than the final total.

To resolve this, when a customer selects a payment option in the billing section, a backend call is triggered. This call recalculates the basket totals, including any applicable promotions, and updates the Klarna session details. Consequently, the Klarna iframe widgets and the mini summary section on the storefront are refreshed to display the final order details.

For payment methods other than Klarna, this logic should be customized by the merchant to handle any third-party payment integrations.

### 2.5.2. Price Adjustment Taxation Handling

Out of the box, the Klarna API sends product and shipping method details along with relevant discounts as separate line items, as shown below:

```
"order_lines": [
    {
        "type": "discount",
        "name": "5 Off Ties Promotion",
        "reference": "682875540326M_$5_off_ties_promotion",
        "quantity": 1,
        "merchant_data": "5ties",
        "unit_price": -500,
        "tax_rate": 500,
        "total_amount": -500,
        "total_tax_amount": 0,
        "total_discount_amount": 0,
        "product_url": null,
```

```

        "image_url": null
    },
    {
        "type": "physical",
        "name": "Checked Silk Tie",
        "reference": "682875540326M",
        "quantity": 1,
        "unit_price": 1919,
        "tax_rate": 500,
        "total_amount": 1919,
        "total_tax_amount": 68,
        "total_discount_amount": 0
    }
}

```

## Handling Gross Taxation with Adjusted Prices

MERCHANTS using gross taxation might opt to enable the “Tax Products and Shipping Only Based on Adjusted Price” preference. This preference is located under Merchant Tools > Site Preferences > Pricing and Promotion, and ensures that price adjustments are not taxed.

The setting `kpPromoTaxation` has been introduced for this purpose. MERCHANTS should update this setting to match their promotion configuration as follows:

- **Price (Based on Price):** The product, shipping, and their discounts will be sent as separate line items. This is the default setting.
- **Adjustment (Based on Adjusted Price):** When selected, the product or shipping method line item will be sent with the attribute `total_amount` matching the prorated price and the attribute `total_discount_amount` matching the total sum of all discounts for this item.

```

"order_lines": [
    {
        "type": "physical",
        "name": "Checked Silk Tie",
        "reference": "682875540326M",
        "quantity": 1,
        "unit_price": 1919,
        "tax_rate": 2200,
        "total_amount": 1919,
        "total_tax_amount": 68,
        "total_discount_amount": 0
    }
]

```

```

        "total_amount": 1419,
        "total_tax_amount": 256,
        "total_discount_amount": 500,
        "product_url": null,
        "image_url": null
    }
}

```



**Note:** Enabling this setting is not required for storefronts using net taxation, as the tax is not included in the product's base price. In such cases, the total order sales tax is sent to Klarna as a separate line item, not at the product or shipping line-item level.

### 2.5.3. Buy Online, Pickup in Store (BOPIS)

When store pickup is enabled on the storefront, the integration sends store details to Klarna during the authorization request and when placing the Klarna order. Store information is only sent after the customer interacts with the Klarna payment method widgets.

#### Address Handling

The store address(es) are included in the Enhanced Merchant Data (EMD) attachment under the `other_delivery_address` attribute when applicable. The handling of addresses in Klarna orders with store pickup is as follows:

- **Orders with Only Store Pickup Shipments:** If there is no home delivery address, the shipping address in the Klarna order will be set to the first store shipment's details.
- **Orders with Both Store Pickup and Home Delivery Shipments:** The home delivery address will be used as the shipping address in the Klarna calls.
- **Orders with No Store Pickups:** No information is sent in the `other_delivery_address` attribute.

```
{
    "attachment": {
        "content_type": "application/vnd.klarna.internal.emd-v2+json",
        "body": {
            "other_delivery_address": [
{

```

```

        "shipping_method": "store pick-up",
        "shipping_type": "normal",
        "first_name": "Test",
        "last_name": "Customer",
        "street_address": "1487 Bay St",
        "street_number": "",
        "postal_code": "01109",
        "city": "Springfield",
        "country": "US"
    ]
}
}
}

```

For more information on the options, refer to the Klarna documentation [here](#).

#### 2.5.4. Service Rate Limits



#### This functionality is deprecated as of release 24.4.0

- **Rate limits are managed by Klarna.** The plugin handles the errors in case these are exceeded. Default service `klarna.http.defaultendpoint` will be used for all API payments calls.
- For more information refer to the Klarna technical documentation [here](#).

Klarna Payment API sets rate limits by operation (session creation, order creation, etc) to maintain a high quality of service for all its customers. A merchant has the flexibility to request higher rate limits for a specific duration. The duration of such events may last the period of high traffic events (e.g.: Flash sales or Holiday shopping). To enable these agreed rate limits, contact your Klarna account manager.

#### 2.5.5. Klarna Subscriptions

The Klarna cartridge supports subscription handling, allowing merchants to offer products as subscriptions, manage subscription details, and handle recurring orders seamlessly.

### 2.5.5.1. Configuration

Subscription details are configured at the product level. Products can be set as subscription-only, standard, or both. The trial period must be an integer value.

Klarna Subscription

Is Klarna Subscription Product: Yes

Klarna Trial Days Usage: 15 (Integer)

Is Klarna Standard Product: No

Figure 20. Product subscription configuration

### 2.5.5.2. Cart Page

There are two types of subscription products:

- Subscription-only Products:** These are automatically added to the shopping cart as subscription line items.
- Dual-purpose Products:** For products that can be either subscription or standard, users can select their preference on the cart page.

<b>3/4 Sleeve V-Neck Top</b>	Color: Butter Size: L In Stock <a href="#">Edit</a>	Each <del>\$24.00</del> <b>\$21.60</b> 10% off Summer Flats	Quantity 1	Total <del>\$24.00</del> <b>\$21.60</b>	X
<input type="checkbox"/> Subscribe This product can be purchased as a Subscription with Klarna payments. Subscription Trial period: 12 days					
<b>3/4 Sleeve Solid Drape Neck Sweater</b>	Color: Carnation Size: M In Stock <a href="#">Edit</a>	Each <del>\$46.99</del> <b>\$42.29</b> 10% off Summer Flats	Quantity 1	Total <del>\$46.99</del> <b>\$42.29</b>	X
<input checked="" type="checkbox"/> Subscribe This product can be purchased as a Subscription with Klarna payments. Subscription Trial period: 12 days					

Figure 21. Subscription products on cart page



## Important!

The checkout process will not proceed if the cart contains a mix of standard and subscription products with different trial periods, or if some products have a trial period while others do not.

### Subscription Details

Dropdown menus with predefined values appear on the cart page when there is at least one subscription product in the cart. These values can be configured in the Administration panel under **Site Development > System Object Types > Basket - Attribute Definitions**. The attributes available for configuration are:

- `kpSubscriptionFrequency`
- `kpSubscriptionPeriod`

The screenshot shows a shopping cart interface with a single item: "Cluster Drop Earring" priced at \$26.00. A checkbox labeled "Subscribe This product can be purchased as a Subscription with Klarna payments." is checked. Below it, a note says "Subscription Trial period: 7 days". A promotional message offers "Purchase \$74.00 or more and receive Free Shipping with USPS (7-10 Business Days)". On the right, there's a "Subscription Info" section with dropdowns for "Subscription Period" (set to "Week") and "Subscription Frequency" (set to "1"). A callout highlights the "Subscription Frequency" dropdown, which includes options like Day, Week, Month, Year, and a range from 1 to 15. Other sections include "Enter Promo Code", "Shipping" (set to "2-Day Express (2 Business Days)"), and a summary of "Estimated Total" (\$37.79) with financing options ("4 interest-free payments of \$15.74 with Klarna TESTDRIVE").

**Figure 22. Subscriptions details in cart page**

#### 2.5.5.3. Checkout

Only logged-in users can complete a subscription checkout. Session intent is defined based on the basket content:

- **tokenize**: Basket contains products with a trial period; user is not charged on order creation.
- **buy\_and\_tokenize**: Basket contains subscription products without a trial period.

- **buy:** Basket contains only standard products, no subscription products.

For intents `tokenize` and `buy_and_tokenize`, a Klarna customer token for recurring payments is created and stored in the customer profile for future use.

## User Profile Updates

Upon order creation, the user profile is updated with subscription data:

1. **token:** Subscription token.
2. **enabled:** Status of the subscription.
3. **nextChargeDate:** Calculated date for the next subscription charge.
4. **subscriptionPeriod:** Enumerated value representing the subscription period ('week', 'month', or 'year').
5. **subscriptionFrequency:** Numeric value representing the frequency of the subscription (1, 2, 3, 4, 5, 6, 15).
6. **subscriptionProductID:** ID of the corresponding subscription product.
7. **lastOrderID:** ID of the last order for the subscription.

### 2.5.6. Account Subscription Dashboard

Users can view a full list of their subscriptions in the My Account section. They can cancel subscriptions, which will be displayed with an Inactive status. The Cancel Subscription button deactivates the customer token, preventing further charges.

The screenshot shows a web interface for managing subscriptions. At the top, there's a breadcrumb navigation: "Home / My Account". Below it, a header says "Subscriptions". There are two main card-like components, each representing a subscription.

**Subscription 1 (Top):**  
Subscription ID: 69c0aeb2-e34e-4d11-847a-6a217dbce435  
Cancel  
Last Order ID: 00050402  
Next Charge Date: 10/5/23  
Status: Inactive  
Total subscription amount: \$38.15

**Subscription 2 (Bottom):**  
Subscription ID: 9404e832-2c91-4e8b-8535-6e26c881d9bf  
Cancel  
Last Order ID: 00050403  
Next Charge Date: 10/5/23  
Status: Active  
Total subscription amount: \$38.15

Figure 23. My Account Subscription dashboard

## 2.5.7. Recurring Subscription Order Creation

A back-end job processes recurring subscriptions for each user in SFCC if subscriptions exist. The job iterates through customers and checks for subscriptions due for payment on the same day.

### Order Creation Process

1. **Create new order:** Using data from `lastOrderID`, a new order is created, and a charge call is made to Klarna using the token and price from `lastOrderID`.
2. **Update details on success:** On success, `nextChargeDate` is updated based on `subscriptionPeriod`, and `lastOrderID` is set to the ID of the newly placed order. The Subscription Dashboard is updated accordingly.

### Handling Failures

- **Retry mechanism:** Merchants can configure a retry mechanism with specific retry intervals. If retries fail, the subscription can be deactivated.
  - **Retry:** Boolean field (Yes/No).
  - **Number of retries:** Number of retries (1, 2, etc.).
  - **Retry frequency:** Interval in days (1, 2, etc.).
- **Cancel subscription:** If retry is disabled, the subscription is canceled.

Orders with trial period subscriptions are paid after the trial period ends. On the next charge date, a new order is created with the channel type set to `SUBSCRIPTIONS`.

## 2.6. Klarna Express Button (KEB)



### This functionality is deprecated as of release 24.4.0

Klarna Express Button has been replaced with [Klarna Express Checkout](#).

The cartridge supports the Klarna Express button (KEB) on the standard Cart page and mini-Cart.

By enabling Klarna's Express button on the cart page of your website, shoppers can choose to log into their (or sign up for a) Klarna account and have their personal details pre-filled in the checkout. Klarna's payment method will be pre-selected for the shopper. In supported

markets, Klarna's network of shoppers benefit from a pre-filled checkout experience, which also includes first time shoppers on a merchant storefront.

Once the credentials have been provided and the shopper has successfully authenticated with their Klarna account, they are redirected to the checkout page with relevant details pre-populated and Klarna Payment (e.g.: "Pay in 4") method pre-selected on the billing page.

For storefronts with custom checkout design or address field validations, please ensure that the authenticated users address data is populated without modification to ensure the Klarna Payment method authorization is completed successfully.



**Note:** On redirect, in the standard checkout, any existing email ID and phone number are updated with the latest provided by the shopper.

## 2.7. Compatibility

This cartridge has been tested against API Version 22.6 (Compatibility Mode: 22.7) and SFRA version 6.0.0.

## 2.8. Privacy and Payment

### 2.8.1. GDPR Compliance

The cartridge is compliant with GDPR recommendations and follows best practices to ensure only necessary Personally Identifiable Information (PII) is transmitted to authorize the payment method. For detailed guidelines, refer to the implementation best practices [here](#).

### 2.8.2. EMD (Extra Merchant Data)

The cartridge supports sending additional information on the customer's past purchase history and "Buy Online, Pickup in Store" (BOPIS) store addresses when enabled in custom preferences under "Attachments" (`kpAttachments`). The types of data that can be sent as an attachment are detailed [here](#).

EMD is required for certain types of merchant orders and generally improves acceptance rates (e.g., `customer_account_info`: past interactions with the merchant store). EMD is included as part of the authorization step in the Commerce Cloud checkout. The data sent to Klarna is customizable and can be viewed in

`int_klarna_payments/scripts/payments/additionalCustomerInfo.js`. This script should return a JSON string to be used as the value for the body sub-field of the attachment field as described [here](#).

## Example Schema for Additional Customer Information

If the example `additionalCustomerInfo.js` file is used unchanged, the data sent to Klarna follows this 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": "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]{4}-[0-1][0-9]-[0-3][0-9]T[0-2][0-9]:[0-5][0-9](:[0-5][0-9])?Z?"
                    },
                    "account_last_modified": {
                        "description": "ISO 8601 e.g. 2012-11-24T15:00",
                        "type": "string",
                        "format": "date-time",
                        "pattern":
                            "[0-9]{4}-[0-1][0-9]-[0-3][0-9]T[0-2][0-9]:[0-5][0-9](:[0-5][0-9])?Z?"
                    }
                }
            }
        }
    }
}
```

```

        }
    },
    "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]{4}-[0-1][0-9]-[0-3][0-9]T[0-2][0-9]:[0-5][0-9](:[0-5][0-9])?Z?$$"
                },
                "date_of_first_paid_purchase": {
                    "description": "ISO 8601 e.g. 2012-11-24T15:00",
                    "type": "string",
                    "format": "date-time",
                    "pattern":
                    "^[0-9]{4}-[0-1][0-9]-[0-3][0-9]T[0-2][0-9]:[0-5][0-9](:[0-5][0-9])?Z?$$"
                }
            }
        },
        "other_delivery_address": {
            "type": "array",
            "items": {
                "type": "object",
                "additionalProperties": false,

```

```
"properties": {
    "shipping_method": {
        "type": "string",
        "enum": ["store pick-up", "pick-up point", "registered box",
"unregistered box"]
    },
    "shipping_type": {
        "type": "string",
        "enum": ["normal", "express"]
    },
    "first_name": {
        "type": "string"
    },
    "last_name": {
        "type": "string"
    },
    "street_address": {
        "type": "string"
    },
    "street_number": {
        "type": "string"
    },
    "postal_code": {
        "type": "string"
    },
    "city": {
        "type": "string"
    },
    "country": {
        "type": "string"
    }
}
}
}
}
```

## Example data

```
{
    "attachment": {
        "content_type": "application/vnd.klarna.internal.emd-v2+json",
        "content": "base64 encoded file content"
    }
}
```

```

"body": {
    "customer_account_info": [
        {
            "unique_account_identifier": "5509d9f7c8720c0e4575154b",
            "account_registration_date": "2015-03-18T20:03:03Z",
            "account_last_modified": "2015-03-18T20:03:03Z"
        }
    ],
    "payment_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"
        }
    ],
    "other_delivery_address": [
        {
            "shipping_method": "store pick-up",
            "shipping_type": "normal",
            "first_name": "Test",
            "last_name": "Customer",
            "street_address": "1487 Bay St",
            "street_number": "",
            "postal_code": "01109",
            "city": "Springfield",
            "country": "US"
        }
    ]
}
}
}
}

```



**Note:** When the customer uses Guest Checkout, the EMD sent includes

`payment_history_full[0].unique_account_identifier` (cqcid value set by SFCC), and all other fields are empty.

### 2.8.3. PCI-DSS Compliance

The virtual card (MCSv3) solution enables settlements using individual virtual cards issued against a Klarna order. To comply with PCI-DSS requirements, merchants must ensure that data is securely maintained and transmitted as part of their operation in their live store

environment. The required steps must be completed in consultation with your payment service provider/acquirer before going live. Review the order export details required for virtual card-based Klarna orders in advance. Any historical decrypted PCI data should be expunged, regardless of the VCN validity date.



**Important!**

**DO NOT SAVE UNENCRYPTED PCI DATA ON THE SERVER.**

# 3. Conversion Boosters

## 3.1. Klarna On-site Messaging

On-site messaging is a platform that enables you to add tailored messaging to your website. With on-site messaging, you can inform shoppers about the different payment options available as they browse your site. By using Klarna, customers have access to flexible payment options in the checkout; on-site messaging is a great way to let them know even before they decide to buy.

The Klarna Payment cartridge provides multiple options in the standard implementation based on the reference architecture:

- Product Page and Cart-based promotions
- Sitewide
  - Top Banner strip
  - Footer logo
- Custom Info-page

### 3.1.1. Configuration

#### 3.1.1.1. Configuring OSM via KlarnaCountries custom object

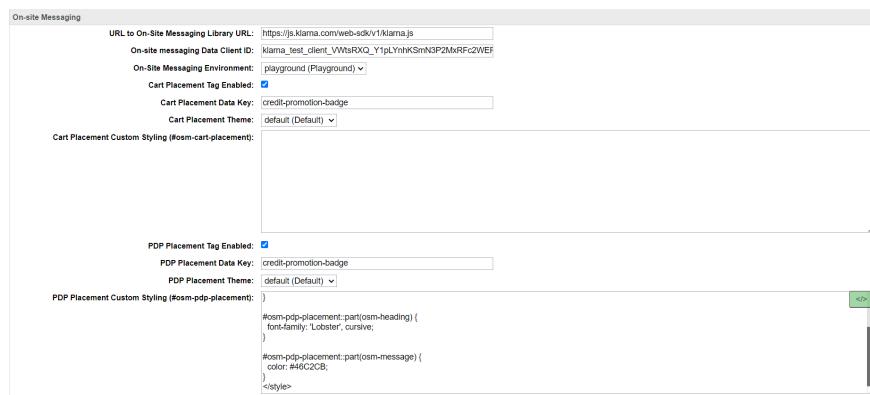


#### This functionality is deprecated as of release 24.4.0

Current configuration using attributes from Custom Object is deprecated with version 24.4.0. In the new implementation, settings are retrieved from Site Preferences as explained in [section 3.1.1.2](#).

Klarna On-site Messaging (OSM) is configured by site and by locale via the **KlarnaCountries** custom object. To configure the OSM settings for a locale, visit "Merchant Tools – Custom Object Editor" and search for **KlarnaCountries** custom object. Select the country key, e.g., "US".

In the custom country-specific configuration, 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 the Klarna Merchant Portal within the On-site Messaging App.



**Figure 24. OSM Settings in BM**

To enable the Placement tag for the Cart Page, the "Cart Placement Data Key" must be filled with the Data Key value and the "Cart Placement Tag Enabled" must be checked. To customize the Cart Placement, select a theme and/or enter custom CSS in the "Cart Placement Custom Styling" attribute using the placement id **#osm-cart-placement**.



**Note:** Cart placements amount must be updated, and the latest Klarna credit offering placement (where required) displayed to the customer when the order line quantity is updated on the cart page. Refer to section [app\\_storefront\\_base\cartridge\client\default\js\cart\cart.js \(required\)](#).

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

customize the PDP Placement, select a theme and/or enter custom CSS in the "PDP Placement Custom Styling" attribute using the placement id `#osm-pdp-placement`.

To enable the Placement tag for the header, the "Header Placement Data Key" must be filled with the Data Key value and the "Header Placement Tag Enabled" must be checked. To customize the Header Placement, select a theme and/or enter custom CSS in the "Header Placement Custom Styling" attribute using the placement id `#osm-header-placement`.

To enable the Placement tag for the footer, the "Footer Placement Data Key" must be filled with the Data Key value and the "Footer Placement Tag Enabled" must be checked. To customize the Footer Placement Tag, select a theme and/or enter custom CSS in the "Footer Placement Custom Styling" attribute using the placement id `#osm-footer-placement`.

To enable the Placement tag for the Info Page, the "Info Page Placement Data Key" must be filled with the Data Key value and the "Info Page Placement Tag Enabled" must be checked. To customize the Info Page Placement Tag, select a theme and/or enter custom CSS in the "Info Page Placement Custom Styling" attribute using the placement id `#osm-info-page-placement`.

In the Library URL, input the full URL to the On-Site Messaging JavaScript Library. In On-Site Messaging Environment, select the corresponding environment (playground for test or production). In On-site Messaging Data Client ID, input the On-Site Messaging Data client ID value. For On-site Messaging Data locale, input the valid On-Site Messaging Data locale.

All placements can be customized with custom CSS using the placement component id in the brackets of the attribute name. The customization value should be wrapped in a `<style>` element.

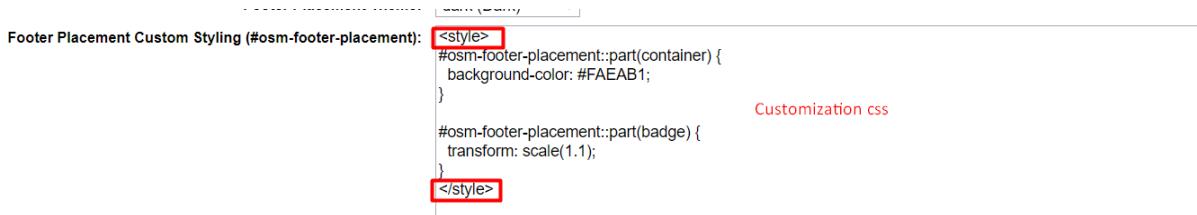


Figure 25. OSM Placements CSS customization

### 3.1.1.2. Configuring OSM via site preferences

Current configuration under BM is placed in Merchant Tools – Site Preferences-Custom Preference under **Klarna\_OSM** group:

- To enable Klarna OSM implementation in storefront, **osm\_enable** must be set to **true**.
- **osm\_theme** dropdown is used to apply a theme to OSM. Options include **default**, **dark**, and **custom**.
- To enable placement in PDP, cart, header, footer, and info page, **osm\_placement** is enabled in the multi-select dropdown. Placement tag ID is hardcoded from code.

Placement	TagID
Product Page	<b>credit-promotion-auto-size</b>
Cart Page	<b>credit-promotion-badge</b>
Site Wide Banner	<b>top-strip-promotion-badge</b>
FAQ	<b>info-page</b>
Footer	<b>footer-promotion-auto-size</b>

Table 2. Klarna On-site messaging placement tagID for different pages

Placement styling is taken from **osm\_placement\_styling**, which is a JSON. **osmLibraryURL** is hardcoded and retrieved from constants. **osmEnvironment** is used from **KlarnaPayments**.

CSS customizations are available only for the new OSM library version. Please follow Klarna guidance for placement styling - [Styling On-Site Messaging with CSS](#).

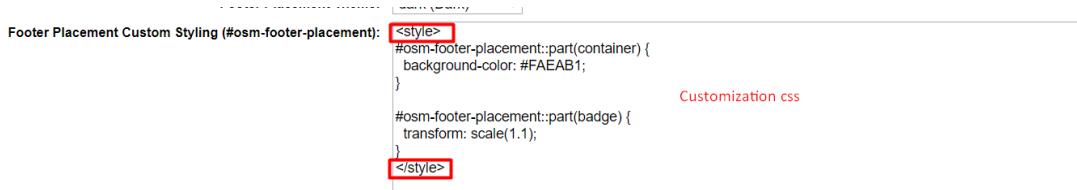


Figure 26. OSM Configuration in BM

For Canada only, update `osmDataInlineEnabled` Enabled on PDP/Cart placement (Canada only) value as follows:

- For PayBright enabled payment methods – set to `true`.
- For Klarna enabled payment methods – set to `false`.

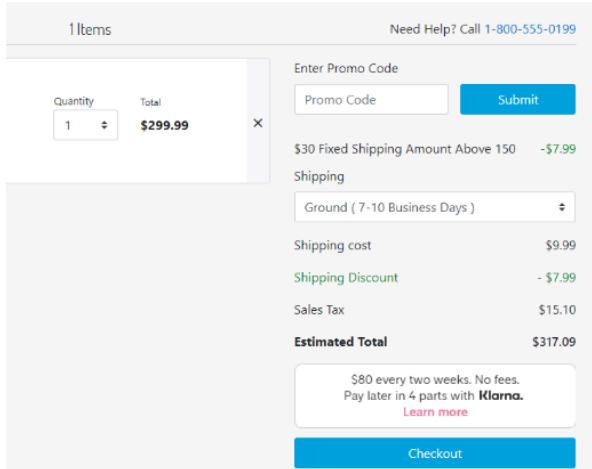


Figure 27. On-Site Messaging on Cart Page

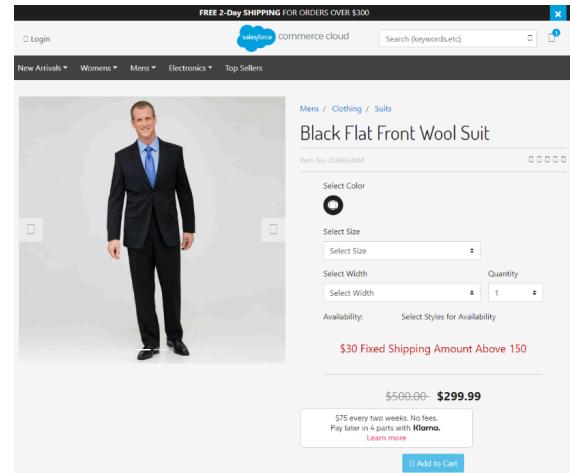


Figure 28. On-Site Messaging on PDP Page

In addition to the above, if you wish to display the dedicated (custom) Klarna info OSM page you can use the following controller endpoint `KlarnaPayments-InfoPage`. For example, update the `footer-about` content asset to include this line of code as shown below:

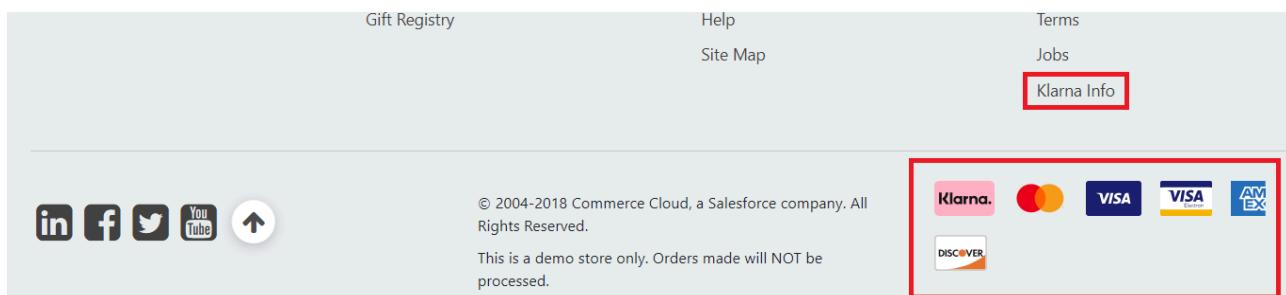
```
<li><a href="$url('KlarnaPayments-InfoPage')$" title="Go to Klarna Info">Klarna Info</a></li>
```

The screenshot shows a code editor interface with a red box highlighting the line of code: <li><a href="\$url('KlarnaPayments-InfoPage')\$" title="Go to Klarna Info">Klarna Info</a></li>

```

⑤ Body: <h3><a class="title" href="#">About</a></h3>
      <ul class="menu-footer content">
        <li><a href="$url('Page-Show', 'cid', 'about-us')$" title="Go to About Us">About Us</a></li>
        <li><a href="$url('Page-Show', 'cid', 'privacy-policy')$" title="Go to Privacy">Privacy</a></li>
        <li><a href="$url('Page-Show', 'cid', 'terms')$" title="Go to Terms">Terms</a></li>
        <li><a href="$url('Page-Show', 'cid', 'jobs_landing')$" title="Go to Jobs">Jobs</a></li>
        <li><a href="$url('KlarnaPayments-InfoPage')$" title="Go to Klarna Info">Klarna Info</a></li>
      </ul> <!-- END .footer__about -->
    
```

**Figure 29. Footer asset update**



**Figure 30. On-Site Messaging on footer and linked to Klarna Info Page**

The header features a "FREE 2-DAY SHIPPING FOR ORDERS OVER \$300" banner, a "salesforce commerce cloud" logo, a search bar, and a login link. Below the header, a navigation bar offers "New Arrivals", "Womens", "Mens", "Electronics", and "Top Sellers". A promotional message "Buy now, pay later with Klarna. [Learn more](#)" is displayed. The main content area features a large headline "Shop now, pay later with Klarna." and a subtext "We've partnered with Klarna to give you a better shopping experience." To the right, there is a section titled "It's smooth and simple" with descriptive text and a small image of a person exercising on a purple mat.

**Figure 31. On-Site Messaging on header & dedicated Klarna info page**

For more information regarding OSM customizations and best practices, please refer to the Klarna Developer Docs. Integration Best Practices and information about Klarna Branding and Co-marketing options can be found [here](#).



**Note:** It is the merchant's responsibility to ensure that user consent is collected when required for OSM placements for your local market to abide by legal requirements, e.g., EU cookie-guide in Klarna Merchant Portal.

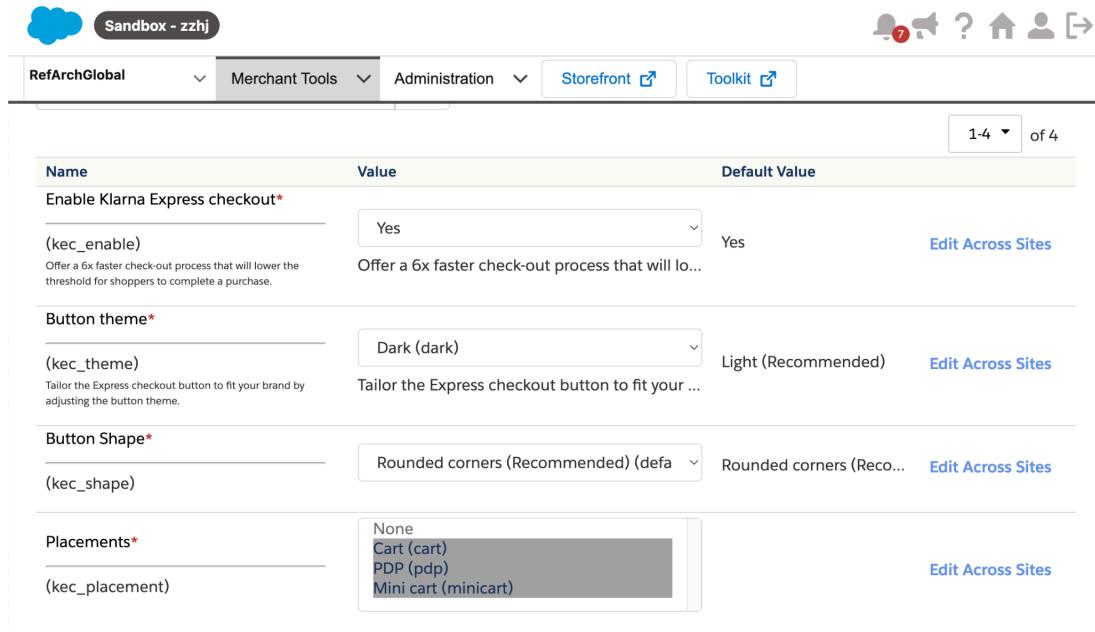
## 3.2. Klarna Express Checkout

Klarna Express checkout (KEC) is a new feature introduced in Storefront that displays an Express Checkout Button on the Product Detail Page (PDP), Cart, and Mini Cart. Users are redirected to Klarna upon clicking the Express Checkout button. This feature enables a quick and easy checkout process where the Shipping Address, Billing Address, and Payment details are preselected, allowing the checkout to be completed in fewer clicks. For more information, refer to the [Klarna Documentation](#). The multi-step checkout process includes a finalize call at the place order stage.

### 3.2.1. Configuration

The basket includes a custom attribute, `kpIsExpressCheckout`, which is set to `true` to enable the Express Checkout process.

To configure Express Checkout, navigate to **Merchant Tools > Site Preferences > Custom Site Preference Groups > Klarna Express checkout**. Here, you can select the locations within your store where you want the button to appear and customize its appearance by choosing the theme and shape.



**Figure 32. Klarna Express checkout configuration**

### 3.2.2. Placements

#### Product Detail Page (PDP)

The Express checkout button is placed near the “Add to cart” button to offer a seamless checkout alternative. The button is not visible if the product is subscription-only.

#### Minicart and cart

The product is added to the cart, and the checkout starts with a new basket created exclusively for this product. Once the checkout is completed, the old basket is restored based on a session attribute.

The Express Checkout button is not displayed in the minicart if there are only subscription-only products in the cart.

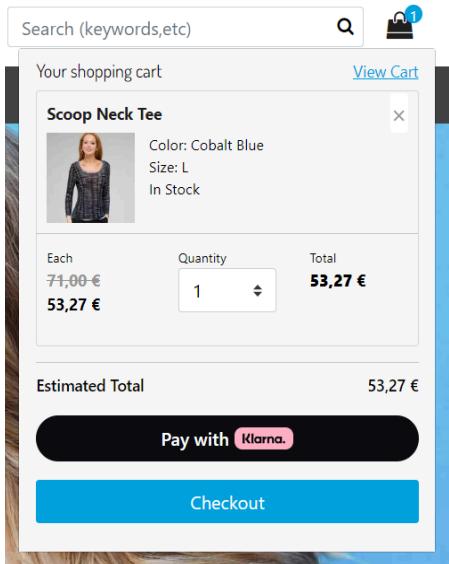


Figure 33. Klarna Express checkout on minicart

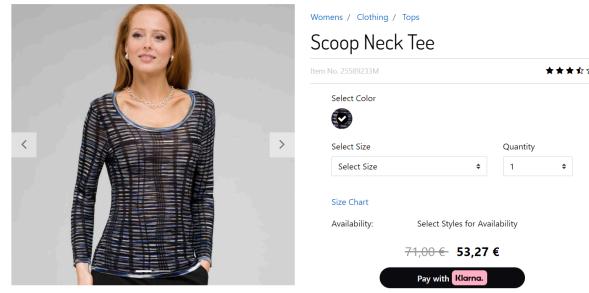


Figure 34. Klarna Express checkout on PDP

# 4. Implementation Guide

## 4.1. Setup of Business Manager

The Klarna Payments LINK Cartridge contains two cartridges required for full functionality. Controller and SFRA support are separated into two distinct cartridges, facilitating the installation and use of either model:

- **int\_klarna\_payments**: Implements the core storefront functionality.
- **int\_klarna\_payments\_sfra**: Implements the storefront functionality with SFRA code.

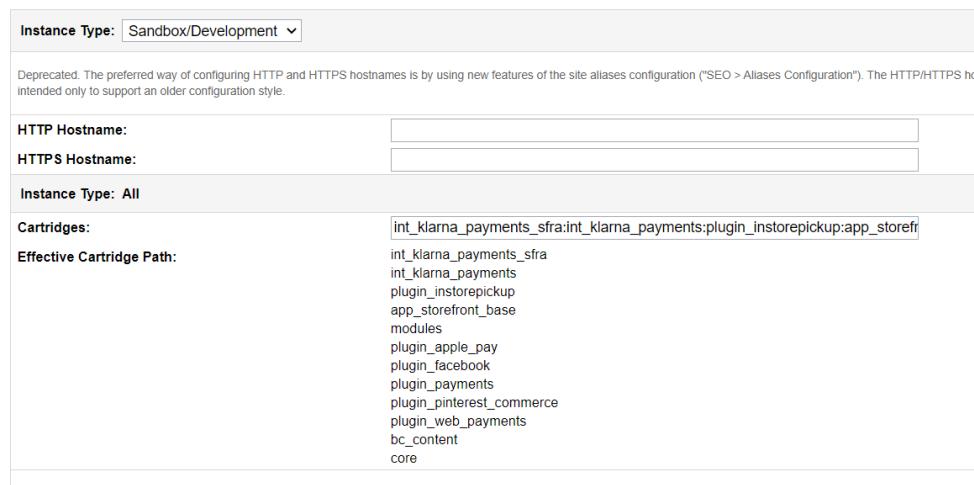
### 4.1.1. Cartridge Upload & Assignment

#### Step-by-Step Instructions

- 1) Import the **int\_klarna\_payments** cartridge into the SCC Studio Workspace:
  - a) Open SCC Studio.
  - b) Click **File -> Import -> General -> Existing Projects into Workspace**.
  - c) Browse to the directory where you saved the **int\_klarna\_payments** cartridge.
  - d) Click **Finish**.
  - e) Click **OK** when prompted to link the cartridge to the sandbox.
- 2) Import the **int\_klarna\_payments\_sfra** cartridge into the SCC Studio Workspace:
  - a) Open SCC Studio.
  - b) Click **File -> Import -> General -> Existing Projects into Workspace**.
  - c) Browse to the directory where you saved the **int\_klarna\_payments\_sfra** cartridge.
  - d) Click **Finish**.
  - e) Click **OK** when prompted to link the cartridge to the sandbox.

**3) Prepend the Klarna cartridges to the effective site cartridge path:**

- a) Log into the SCC Business Manager.
- b) Click **Administration -> Sites -> Manage Sites**.
- c) Select the desired site.
- d) Click on the **Settings** tab.
- e) Prepend `int_klarna_payments_sfra:int_klarna_payments` to the **Cartridges** field.
- f) Click **Apply**.



**Figure 37. Effective Cartridge Path**

#### **4.1.2. Metadata Import**

- 1) Go to the main directory “metadata” folder, review the site-template content, and edit if needed. (The site template is prepared to set up “SiteGenesis” and “RefArch” sites. You may want to change that to your actual sites and delete the ones that are not needed.)
- 2) Zip the directory to create the “site-template.zip” installation package.
- 3) Log into the SCC Business Manager.
- 4) Click **Administration -> Sites Development -> Site Import & Export**.
- 5) Browse to the directory where you saved the “site-template.zip”.
- 6) Click **Upload**.
- 7) Select the uploaded site zip and click **Import**.



**Note:** Review the default `service.xml` file in the `site-template.zip` and update the configuration for Playground and Production accordingly before importing.

#### 4.1.3. Build Klarna JS & CSS

- 1) Update the path to your base SFRA installation in the `package.json` file from the same root folder if necessary.

##### Example Path Configuration:

```
/**  
 * Call Credit Card Authorization Hook (for VCN settlement)  
 * @param {dw.order.order}.ord
```

- 2) Ensure the correct path to the SFRA cartridge is configured.
- 3) Run the following commands from the root folder of the Klarna repository:
  - a) `npm run compile:sscss` to compile the SCSS files.
  - b) `npm run compile:js` to build the Klarna-specific JavaScript files.

## 4.2. Configuration

### 4.2.1. Configure Klarna Activation Site Preferences for single Klarna API credentials per site

#### Steps:

- 1) **Log into the SCC Business Manager.**
- 2) **Select the Desired Site:** Choose the site from the tabs across the top of the page.
- 3) **Access Custom Preferences:** Click **Site Preferences** -> **Custom Preferences** -> **Klarna Activation**
- 4) **Update all the fields:**
  - a) Enter `client_id` to activate OSM and KEC.
  - b) API Username and password.

- c) Select region.
- d) Select markets on which Klarn should be available.
- e) Select environment - yes for playground, no for production.

The screenshot shows the Klarna activation settings page within the Salesforce Commerce Cloud Business Manager. The top navigation bar includes tabs for RefArchGlobal, Merchant Tools, Administration, Storefront (selected), and Toolkit. A sidebar on the left lists categories like Client ID, API Username, API Password, Region, Market, and Run plugin in TEST mode. The main content area contains fields for Client ID (value: klarna\_test\_client\_cUE3SEILaVRLWGtJSz9od3dN), API Username (value: 74b0b87f-f97a-4389-ab12-4cff1c398afc), API Password (value: masked), Region (selected: Europe (EU)), Market (dropdown menu showing options like Australia (AU), Austria (AT), Belgium (BE), Canada (CA), Czech Republic (CZ), Denmark (DK), Finland (FI), France (FR), Germany (DE), and None, with Czech Republic selected), and Run plugin in TEST mode (selected: Yes). Each field has an "Edit Across Sites" link to its right.

Name	Value	Default Value
Client ID (KP_client_id) (String)	klarna_test_client_cUE3SEILaVRLWGtJSz9od3dN	To unlock the plugin's features, enter your credentials. Get the client identifier and API credentials from the Klarna Merchant Portal, under Settings.
API Username (KP_API_Username) (String)	74b0b87f-f97a-4389-ab12-4cff1c398afc	<a href="#">Edit Across Sites</a>
API Password (KP_API_Password)	*****	<a href="#">Edit Across Sites</a>
Region (KP_Region)	Europe (EU)	<a href="#">Edit Across Sites</a>
Market (KP_Market)	None Australia (AU) Austria (AT) Belgium (BE) Canada (CA) <b>Czech Republic (CZ)</b> Denmark (DK) Finland (FI) France (FR) Germany (DE)	Select the region where your store operates. <a href="#">Edit Across Sites</a>
Run plugin in TEST mode*(KP_environment)	Yes	No <a href="#">Edit Across Sites</a>

Figure 38. Klarna activation settings

#### 4.2.2. Configure Klarna Activation Custom Object for multiple Klarna API credentials on one site

##### Steps:

- 1) **Log into the SCC Business Manager.**
- 2) **Select the Desired Site:** Choose the site from the tabs across the top of the page.
- 3) **Access Custom Objects:** Click **Custom Objects** -> **Custom Object Editor**.

- 4) **Change Object Type:** Set the Object Type dropdown to "KlarnaActivation".
- 5) **Create Object:** enter Klarna activation key (something meaningful to you).
- 6) **Enter Fields:** Enter the required fields as mentioned in the "KlarnaActivation" section.
- 7) **Repeat:** Repeat for the other regions or markets as necessary.

New Custom Object (KlarnaActivation)

Fields with a red asterisk (\*) are mandatory. Click Apply to save the details.

<b>Market Specific Credentials</b>	
 <b>Klarna Activation Key:</b> *	<input type="text"/>
<b>Region:</b> *	- None -
<b>Market(s):</b>	<input checked="" type="checkbox"/> AU (Australia) <input type="checkbox"/> AT (Austria) <input type="checkbox"/> BE (Belgium) <input type="checkbox"/> CA (Canada) <input type="checkbox"/> CZ (Czech Republic)
<b>Client ID:</b> *	<input type="text"/>
<b>API Username:</b> *	<input type="text"/>
<b>API Password:</b> *	<input type="password"/> Confirm API Password:*
<b>Klarna Payments - Virtual Card Number (VCN)</b>	
<b>Enable Virtual Card Number (VCN):</b>	<input type="checkbox"/>
<b>Public Key ID:</b>	<input type="text"/>
<b>Enable settlement retry:</b>	<input type="checkbox"/>
<b>Apply</b> <b>Cancel</b>	

Figure 39. Klarna activation custom object settings

#### 4.2.3. Add Account Settings to KlarnaCountries Custom Objects



#### This functionality is deprecated as of release 24.4.0

This configuration has been deprecated with release 24.4.0 and will be removed in the future. Use sections [4.2.1](#) and [4.2.2](#) for the updated configuration.

#### Steps:

- 1) **Log into the SCC Business Manager.**
- 2) **Select the Desired Site:** Choose the site from the tabs across the top of the page.
- 3) **Access Custom Objects:** Click **Custom Objects** -> **Custom Object Editor**.
- 4) **Change Object Type:** Set the Object Type dropdown to "KlarnaCountries".
- 5) **Find and Edit:** Click the **Find** button. Select the desired country to edit (see screenshot below).
- 6) **Update Fields:** Update the required fields as mentioned in the "KlarnaCountries" section.

- 7) **Repeat:** Repeat for the other countries as necessary.

The screenshot shows the 'Manage 'US' (KlarnaCountries)' page under 'Custom Objects > Custom Objects > US - General'. The 'General' tab is selected. The page displays various configuration settings for the 'US' country, including:

- Country Code: **US**
- On-site Messaging Data Default Locale: **en-US**
- Service Credential ID: **klarna.http.uscredentials**
- On-site messaging Data Client ID: **60a22a39-c2fd-5d09-bfe1-771459318a4d**
- Cart Placement Tag Enabled: **checked**
- Cart Placement Data Key: **info-page-standard**
- PDP Placement Tag Enabled: **checked**
- PDP Placement Data Key: **credit-promotion-small**
- Header Placement Tag Enabled: **checked**
- Header Placement Data Key: **top-strip-promotion-standard**
- Footer Placement Tag Enabled: **checked**
- Footer Placement Data Key: **footer-promotion-auto-size**
- Info Page Placement Tag Enabled: **checked**
- Info Page Placement Data Key: **info-page**
- Library URL: **https://na-library.playground.klarnaservices.com/lib.js**

Figure 40. KlarnaCountries Settings

#### 4.2.4. Configure Klarna Payment Custom Preferences



#### This functionality is deprecated as of release 24.4.0

This configuration has been deprecated with release 24.4.0 and will be removed in the future. Use sections [4.2.1](#) and [4.2.2](#) for the updated configuration.

#### Steps:

- 1) **Log into the SCC Business Manager.**
- 2) **Select the Desired Site:** Choose the site from the tabs across the top of the page.
- 3) **Access Custom Preferences:** Click **Site Preferences** -> **Custom Preferences** -> **KlarnaPayment**.
- 4) **Fill Out Settings:** Complete the settings as desired. Descriptions of the site preferences are in the Site Preferences section.

#### 4.2.5. Configure Klarna Payment Service



#### This functionality is deprecated as of release 24.4.0

This configuration has been deprecated with release 24.4.0 and will be removed in the future. Use sections [4.2.1](#) and [4.2.2](#) for the updated configuration.

##### Steps:

- 1) **Log into the SCC Business Manager.**
- 2) **Navigate to Services:** Click Administration -> Operations -> Services.
- 3) **Access Credentials:** Click the Credentials tab. Each Klarna credential corresponds to one of the KlarnaCountries custom objects. Click on the one you want to edit.
- 4) **Enter Credentials:** Enter the MID API username and API password received from Klarna.
- 5) **Edit URL Field:** Update the URL field if using the Production environment. For Klarna API URLs, refer to [Klarna API URLs](#).

Administration > Operations > Services > Service Credentials > klarna.http.gbcredentials - Details

#### klarna.http.gbcredentials

Fields with a red asterisk (\*) are mandatory. Click **Apply** to save the details. Click **Reset** to revert to the last saved state.  
These credentials are used by 0 services.

Name:*	klarna.http.gbcredentials
URL:	https://api.playground.klarna.com/
User:	your Merchant ID
Password:	*****

**Figure 41. Service Settings**

#### 4.2.6. Configure Klarna Rate Limited Service Profile



#### This functionality is deprecated as of release 24.4.0

- **Rate limits are managed by Klarna.** The plugin handles the errors in case these are exceeded.
- Default service `klarna.http.defaultendpoint` will be used for all API payments calls.
- For more information refer to the Klarna technical documentation [here](#).

##### Steps:

- 1) **Log into the SCC Business Manager.**
- 2) **Navigate to Klarna Payments Preferences:** Go to *Merchant Tools -> Site Preferences -> Custom Site Preferences Group -> Klarna Payments*.
- 3) **Enable Service Limit Configuration:** Set **Rate Limit By Operation** to **Yes**. If set to **No**, the default service will control the rate limit.

#### 4.2.7. Configure Custom Rate Limits



#### This functionality is deprecated as of release 24.4.0

**Rate limits are managed by Klarna.** The plugin handles the errors in case these are exceeded.

##### Steps:

- 1) **Log into the SCC Business Manager.**
- 2) **Navigate to Services:** Go to **Administration -> Operations -> Services**.
- 3) **Select Required Profile:** Choose a required profile (e.g., `Klarna.http.createSession`).
- 4) **Enable Rate Limit:** Check the **Enable Rate Limit** box.
- 5) **Set Rate Limit:**
  - a) **Max Rate Limit Calls:** Set to 50 (example: Higher rate limit of 50 requests/sec agreed with Klarna).

- b) **Rate Limit Interval (milliseconds):** Set to 1000.
- 6) **Repeat:** Repeat for all service operations with respective agreed rate limits.

The screenshot shows the 'Administration > Operations > Services > Service Profiles' section. A specific profile, 'klarna.http.createSession', is selected. The configuration details are as follows:

- Name:** klarna.http.createSession
- Connection Timeout (ms):** 30,000
- Enable Circuit Breaker:**
- Max Circuit Breaker Calls:** 0
- Circuit Breaker Interval (ms):** 0
- Enable Rate Limit:**
- Max Rate Limit Calls:** 50
- Rate Limit Interval (ms):** 1000

At the bottom left, there is a link to '<< Back to List'.

Figure 42. Example of create session rate limit profile

### 4.3. Extended Controllers

Controller	Start Node	Remarks
Checkout.js	Begin	Extended to call Klarna session manager
CheckoutServices.js	Get, SubmitPayment, PlaceOrder	Klarna payment method/category and totals are being stored
CheckoutShippingServices.js	SubmitShipping, ToggleMultiShipping	Calling the Klarna session manager
Order.js	Confirm	Extending Klarna order data to view data

Table 3. Extended Controllers List

### 4.4. Template Updates

Templates have been updated to support On-site messaging and Addresses forms for Klarna. To be used as reference but feel free to customize the templates to match your specific needs. Final review and sign-off as per project requirements and contract agreements.

## 4.5. Jobs

### 4.5.1. Job “OrderCleanUp” (Optional)



**Note:** This one-time clean-up job is applicable only to merchants integrated with Klarna Payments cartridge version earlier than 19.1.6, utilizing (or previously used) virtual card-based settlement (VCN) and stored decrypted card details within Business Manager.

The job iterates over orders with status “Exported” and the attribute `custom.kpIsVCN=true` to remove sensitive details saved in fields `kpVCNPAN`, `kpVCNCSC`, `kpVCNExpirationMonth`, and `kpVCNExpirationYear` from previous releases. No parameters are passed to the script.

Upon successful run, the job logs the result of processed orders in the custom debug log located in `webdav/Sites/Logs`. You will receive a message indicating the processed orders count for each storefront or a message indicating that there are no orders needing update.

```
Wed, 09 Sep 2020 09:45:38 GMT DEBUG CustomJobThread[1748004063]OrderCleanUp|executeRefArch custom [] Job [OrderCleanUp] - [RefArch] No orders require processing
Wed, 09 Sep 2020 09:45:38 GMT DEBUG CustomJobThread[1748004063]OrderCleanUp|executeRefArchGlobal custom [] Job [OrderCleanUp] - [RefArchGlobal] No orders require processing
Wed, 09 Sep 2020 09:45:38 GMT DEBUG CustomJobThread[1748004063]OrderCleanUp|executeSiteGen custom [] Job [OrderCleanUp] - [SiteGenesis] Orders processed: 2
Wed, 09 Sep 2020 09:45:38 GMT DEBUG CustomJobThread[1748004063]OrderCleanUp|executeSiteGenGlobal custom [] Job [OrderCleanUp] - [SiteGenesisGlobal] No orders require processing
```

Figure 43. Job Logs

In case of an error, the cause of the failure (message and stack trace) will be logged in the standard error log.

```
Wed, 09 Sep 2020 09:05:22 GMT ERROR CustomJobThread[618064@18]OrderCleanUp|execute com.demandware.beehive.core.internal.domain.SystemObjectQueryMgrImpl Sites-RefArch-Site 308 41e0d91e96 e67f0cb0ade56d99ae0c4bc12b 1648624161284573184 - Exception while parsing system object query.
System
RequestID: e67f0cb0ade56d99ae0c4bc12b
SessionType: J00
Truncated SessionID: 41e0d91e96
User Profile UUID: 01e37ff77529fda91ae75408ab
```

Figure 44. Job Logs

### Steps to Setup the Job

1. **Import the Job Configuration:** Navigate to **Administration > Operations > Import & Export.** Import the file **jobs.xml**.

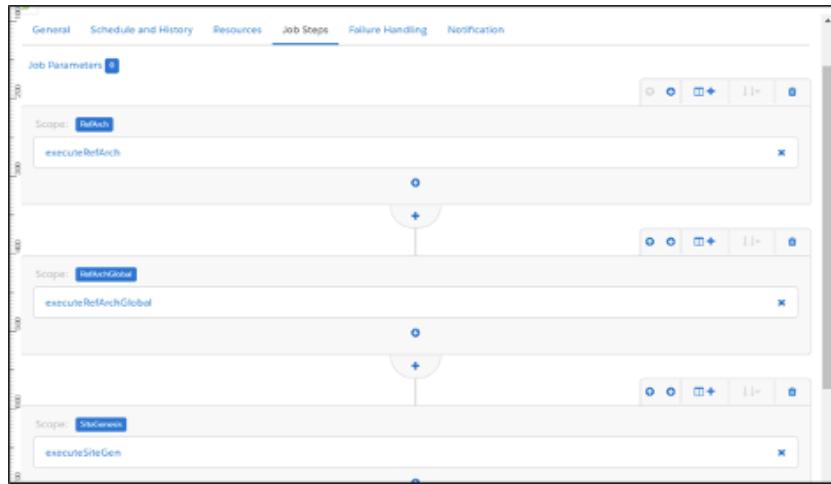


Figure 45. Job Steps



**Note:** The XML file includes only the RefArch scope by default, but it can be configured with multiple flows if you have more than one site using this functionality. Each site should be added as a separate flow.

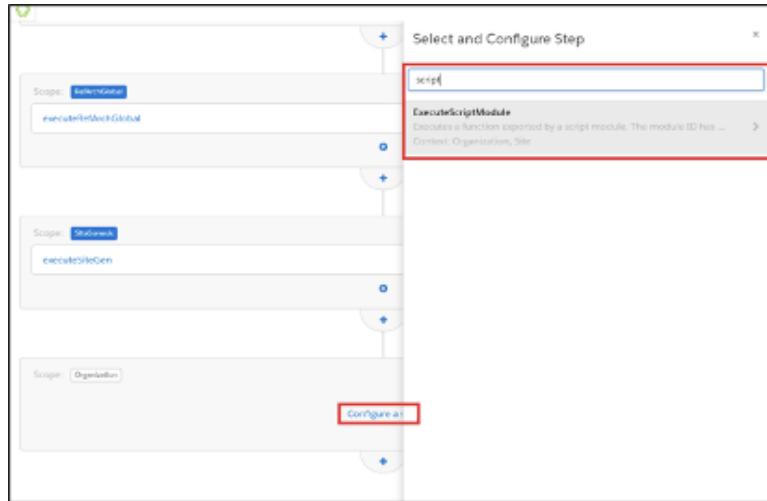
2. **Add a Sequential Flow:** Click on the **Add a sequential flow** button at the bottom of the current flow.



Figure 46. Add New Job Step

### 3. Configure the Step:

- Click on the **Configure a step** button within the flow.
- In the flyout, search for “script” and select **ExecuteScriptModule**.

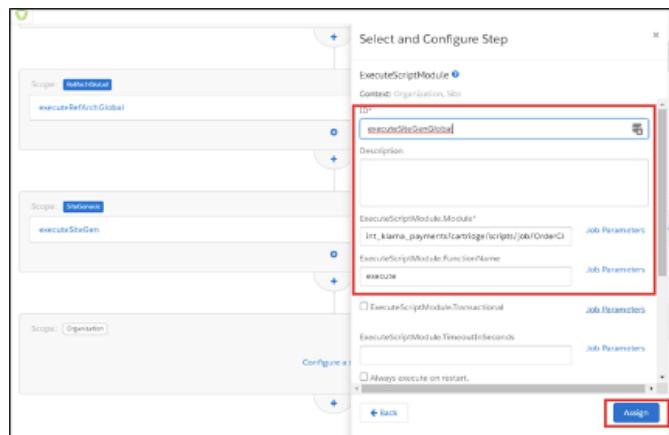


**Figure 47. Configure Step**

### 4. Populate Step Fields:

In the flyout, populate these fields and click the **Assign** button.

- **ID:** Enter any meaningful name. If you have multiple flows, ensure this name is unique. Duplicate names will not save, and an error message will appear.
- **ExecuteScriptModule.Module:** Enter the location of the `OrderCleanUpJob.js` file. By default, it should be `int_klarna_payments/cartridge/scripts/job/OrderCleanUpJob.js` or the location where you have placed the script file.
- **ExecuteScriptModule.FunctionName:** Leave this field value as `execute`.



**Figure 48. Configure Step (cont.)**

##### 5. Assign to Correct Site Scope:

- Click on **Organization** and select **Specific Sites** from the drop-down.
- From the list of sites, select the respective site ID (e.g., **SiteGenesisGlobal**) and click on **Assign**.

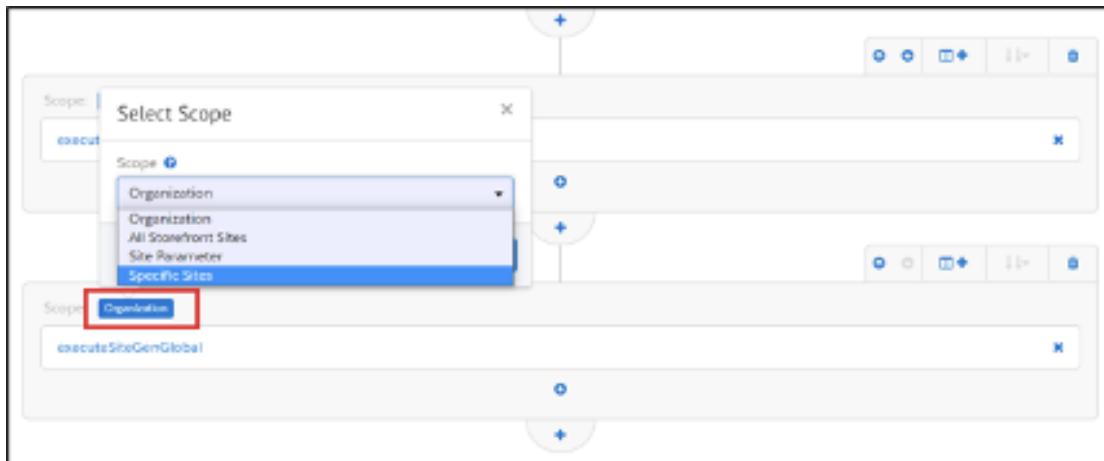


Figure 49. Job Scope

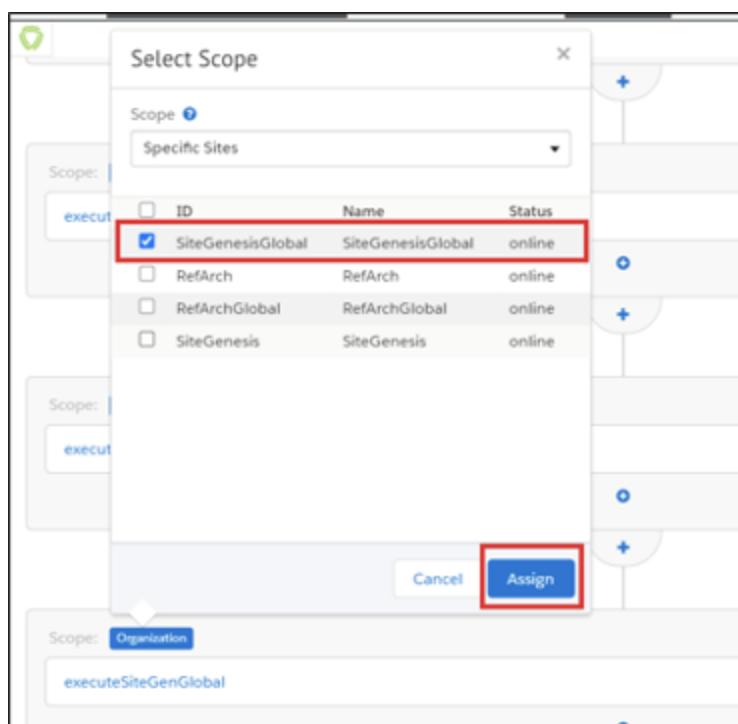


Figure 50. Job Scope (cont.)

Repeat steps 1-5 for each site/storefront that you have using Klarna VCN and need additional configuration. This ensures that the clean-up job runs for all relevant sites and removes sensitive data as required.

### 4.5.2. Job “RecurringOrders”

The **RecurringOrders** job is designed to process subscription entries for all customers. The job performs the following functions:

- **Eligibility Check:** It verifies that the subscription is enabled and that either the `nextChargeDate` or `nextRetryDate` matches the current date.
- **Order Creation:** It creates new Salesforce Commerce Cloud (SFCC) orders for eligible subscriptions, replacing the old ones.
- **Trial Period Handling:** It processes orders with expiring trial periods for charges.

### Configuration

By default, the job is set to run on the **RefArch** site, as specified in the `jobs.xml` file. This setting can be modified either in the `jobs.xml` file or through the storefront configuration.

The job consists of a single step, `createOrder`, with the following configuration:

- **ExecuteScriptModule.Module:**  
`int_klarna_payments/cartridge/scripts/job/RecurringOrdersJob.js`
- **ExecuteScriptModule.FunctionName:** `execute`

The job operates at site level.

Ensure your configuration matches these details to maintain the proper functionality of the RecurringOrders job.

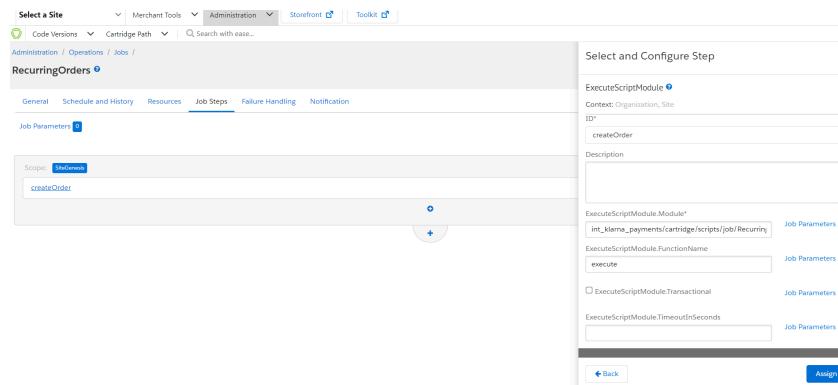


Figure 51. RecurringOrdersJobs

## 4.6. Custom Code

The Storefront Reference Architecture (SFRA) does not require modifications to the core cartridge to enable any of the LINK integration cartridges. The `int_klarna_payments_sfra` cartridge follows Salesforce's best practices. Below are two cases where changes may be applicable:

#### 4.6.1. `app_storefront_base\cartridge\client\default\js\checkout\checkout.js` [optional]

After placing an order, every customer is redirected to Klarna and then sent back to the site with the order confirmation page. To prevent sending any additional URL parameters to Klarna, follow these steps:

```
310 else if (stage === 'placeOrder') {
311   // disable the placeOrder button here
312   $('#body').trigger('checkout:disableButton', '.next-step-button button');
313   $.ajax({
314     url: $('.place-order').data('action'),
315     method: 'POST',
316     success: function (data) {
317       // enable the placeOrder button here
318       $('#body').trigger('checkout:enableButton', '.next-step-button button');
319       if (data.error) {
320         if (data.cartError) {
321           window.location.href = data.redirectUrl;
322           defer.reject();
323         } else {
324           // go to appropriate stage and display error message
325           defer.reject(data);
326         }
327       } else {
328         var continueUrl = data.continueUrl;
329         var urlParams = {
330           ID: data.orderID,
331           token: data.orderToken
332         };
333         continueUrl += (continueUrl.indexOf('?') !== -1 ? '&' : '?') +
334           Object.keys(urlParams).map(function (key) {
335             return key + '=' + encodeURIComponent(urlParams[key]);
336           }).join('&');
337       }
338     }
339   });
340   // enable the placeOrder button here
341   $('#body').trigger('checkout:enableButton', '.next-step-button button');
342   $.ajax({
343     url: $('.place-order').data('action'),
344     method: 'POST',
345     success: function (data) {
346       // enable the placeOrder button here
347       $('#body').trigger('checkout:enableButton', '.next-step-button button');
348       if (data.error) {
349         if (data.cartError) {
350           window.location.href = data.redirectUrl;
351           defer.reject();
352         } else {
353           // go to appropriate stage and display error message
354           defer.reject(data);
355         }
356       } else {
357         var continueUrl = data.continueUrl;
358         var urlParams = {};
359         if (data.orderID && data.orderToken) {
360           urlParams.ID = data.orderID;
361           urlParams.token = data.orderToken;
362         }
363         continueUrl += (continueUrl.indexOf('?') !== -1 ? '&' : '?') +
364           Object.keys(urlParams).map(function (key) {
365             return key + '=' + encodeURIComponent(urlParams[key]);
366           }).join('&');
367       }
368     }
369   });
370 }
```

**Figure 52. Changes in checkout.js**

#### 4.6.2. app\_storefront\_base\cartridge\client\default\js\cart\cart.js [required]

To get updates for the cart On-Site Messaging (OSM) widget, make the following changes in `cart.js` or in your app cartridge.

In the `updateCartTotals(data)` function, add the following code at the bottom:

```

if (data.totals.klarnaTotal) {
    $('klarna-placement').attr('data-purchase-amount', data.totals.klarnaTotal);
    if (window.Klarna && window.Klarna.OnsiteMessaging) {
        window.Klarna.OnsiteMessaging.refresh();
    }
}

```

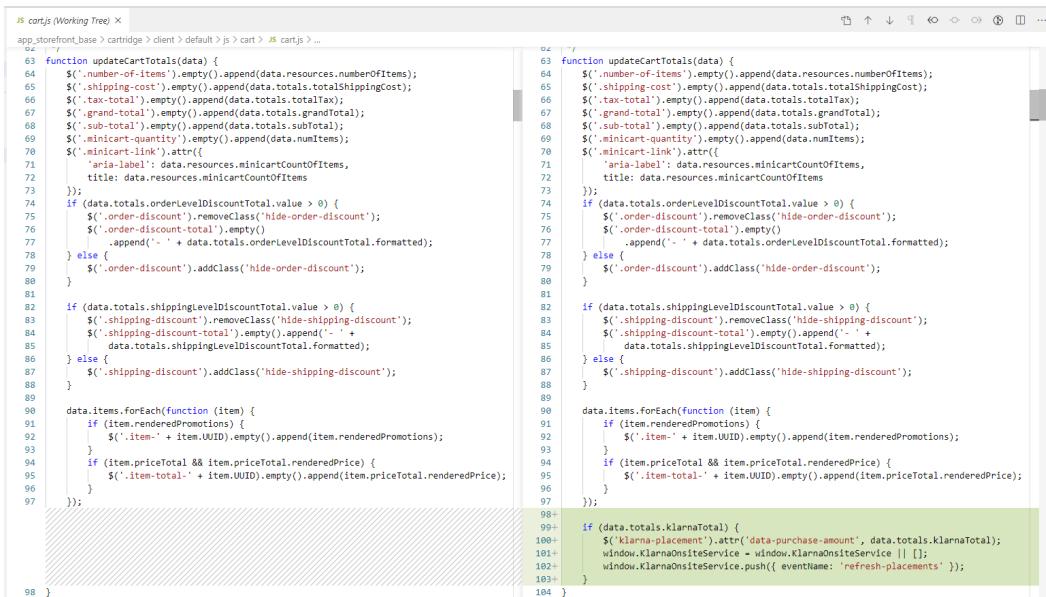


Figure 53. Changes in cart.js

Ensure these modifications are implemented to maintain proper integration with Klarna's services.

## 4.7. Integration with Other Payment Cartridges

This section is relevant if there are other payment cartridge integrations within the cartridge path besides Klarna Payments SFRA. Regardless of the order of those cartridges in the cartridge path, certain templates need to be overwritten by adding a new `if`

condition and including the correct sub-template. The templates that require modifications are:

- \templates\default\checkout\billing\paymentOptions\paymentOptionsContent.isml
- \templates\default\checkout\billing\paymentOptions\paymentOptionsSummary.isml
- \templates\default\checkout\billing\paymentOptions\paymentOptionsTabs.isml

If the website owner requires PayPal as well as Klarna, each of these templates must be copied to a new custom cartridge. The example below shows the new code for the paymentOptionsContent.isml template:

```
<isloop items="${pdict.order.billing.payment.applicablePaymentMethods}"
var="paymentOption" status="loopSate">
    <isif condition="${paymentOption.ID === 'CREDIT_CARD'}">
        <isinclud template="checkout/billing/paymentOptions/creditCardContent" />
    </isif>
    <isif condition="${paymentOption.ID === 'KLARNA_PAYMENTS'}">
        <isinclud
template="checkout/billing/paymentOptions/klarnaPaymentsContent" />
    </isif>
    ...
    <isif condition="${paymentOption.ID === 'PayPal'}">
        <isinclud template="paypal/checkout/paypalContent" />
    </isif>
</isloop>
```

The same applies to the other two templates (paymentOptionsSummary.isml and paymentOptionsTabs.isml).

## 4.8. External Interfaces

All requests are made through Klarna's REST API and are encrypted using SHA-256 with the shared secret provided by Klarna. Only HTTPS is allowed, and JSON is used for all communications.

For a full reference guide, along with the resource structure for requests and responses, refer to the developer portal: [Klarna Payments API](#).

# 5. Testing

Klarna provides a set of testing credentials and triggers to facilitate the testing process. These resources ensure that integrations with Klarna Payments can be thoroughly tested before going live.

For detailed information and to obtain the necessary testing credentials, please refer to the following URL: [Klarna Testing Environ.](#)

# 6. Operations and Maintenance

## 6.1. Data Storage

### 6.1.1. System Object Extensions

#### 6.1.1.1. Basket

Parameter Name	Attribute ID	Description
Klarna Session ID	kpSessionId	The Klarna session ID returned after “Create Session” API endpoint is called (Applicable since version 21.2.0)
Klarna Client Token	kpClientToken	Client token returned by “Create Session” API endpoint and used to initialize the JS SDK (Applicable since version 21.2.0)
Klarna Client Token	kpClientToken	Client token returned by “Create Session” API endpoint and used to initialize the JS SDK.
Klarna Subscription Frequency	kpSubscriptionFrequency	Subscription frequency values (day, month, etc.)

Klarna Subscription Period	<code>kpSubscriptionPeriod</code>	Predefined subscription period in numbers
Klarna Is Express Checkout	<code>kpIsExpressCheckout</code>	Determines if the basket is an express checkout
Klarna Session ID	<code>kpSessionId</code>	The Klarna session ID returned after "Create Session" API endpoint is called (Applicable since version 21.2.0)

**Table 4. Basket Attributes**

#### 6.1.1.2. Order

Parameter Name	Attribute ID	Description
Klarna Payments Order ID	<code>kpOrderID</code>	The Klarna Payments Order ID for Klarna payment method selected by customer
VCN Brand	<code>kpVCNBrand</code>	Klarna Payments virtual card scheme name
VCN Holder	<code>kpVCNHolder</code>	Klarna Payments virtual card holder name
VCN Card ID	<code>kpVCNCardID</code>	Klarna Payments Virtual Card - Card ID
VCN PCI Data	<code>kpVCNPCTData</code>	Klarna Payments Virtual Card PCI Data in encrypted format
VCN Initialization Vector	<code>kpVCNIV</code>	Klarna Payments Virtual Card Initialization Vector

VCN AES Key	<code>kpVCNAESKey</code>	Klarna Payments Virtual Card AES Key
Is VCN Used	<code>kpIsVCN</code>	True if virtual card is enabled & used for payment of the order, otherwise false
Klarna Session ID	<code>kpSessionId</code>	The Klarna session ID returned after "Create Session" API endpoint is called (Applicable since version 21.2.0)
Klarna Client Token	<code>kpClientToken</code>	Client token returned by "Create Session" API endpoint and used to initialize the JS SDK (Applicable since version 21.2.0)
Klarna Subscription Frequency	<code>kpSubscriptionFrequency</code>	Subscription frequency values (day, month, etc.)
Klarna Subscription Period	<code>kpSubscriptionPeriod</code>	Predefined subscription period in numbers

**Table 5. Order Attributes**

#### 6.1.1.3. Order Payment Instrument

Parameter Name	Attribute ID	Description
Klarna Payment Category ID	<code>klarnaPaymentCategoryID</code>	ID of Klarna payment category
Klarna Payment Category Name	<code>klarnaPaymentCategoryName</code>	Name of Klarna payment category

**Table 6. Order Payment Instrument Attributes****6.1.1.4. Payment Transaction**

<b>Parameter Name</b>	<b>Attribute ID</b>	<b>Description</b>
Fraud Status	<code>kpFraudStatus</code>	Klarna Payments order fraud status
Klarna Authorization Token	<code>kpAuthorizationToken</code>	<p>This attribute stores the Klarna authorization token, which is a string value used to authenticate and finalize the order creation process with Klarna's payment system.</p> <p><b>(Applicable since version 23.2.0)</b></p>
Klarna Redirect URL	<code>kpRedirectURL</code>	<p>This attribute holds the URL to which the customer is redirected after the payment authorization is successfully completed by Klarna's system.</p> <p><b>(Applicable since version 23.2.0)</b></p>

**Table 7. Payment Transaction Attributes****6.1.1.5. Site Preferences****Klarna Payments****This functionality is deprecated as of release 24.4.0**

As of version 24.4.0, the **Klarna\_Payments** preferences group is deprecated. The attributes have been reorganized as follows:

- All Klarna Payments related attributes have been moved to the new **Klarna\_KP** preferences group.

- Attributes related to Klarna On-site Messaging have been moved to the **Klarna\_OSM** custom preferences group.
- Klarna Express Checkout attributes have been moved to the **Klarna\_KEC** preferences group.

Parameter Name	Attribute ID	Description
Auto-capture	kpAutoCapture	When enabled "Yes", a full order capture will be attempted automatically. The standalone order management API capture request will include total order amount value for "captured_amount". Default value is "No"
Klarna Payments Service Name	kpServiceName	The service name used for the current site
Send product_url and image_url	sendProductAndImageURLs	If set to true, product_url and image_url fields will be included in the Klarna session and order API calls. This enhances shopper experience post purchase. Default value is "Yes"
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 results may vary. Note: Merchant Reference 1 value is always set to the SCC order ID
Border Color Preference	kpColorBorder	CSS (hex value) color set for Border in Klarna Payments iFrame

Border Selected Color Preference	kpColorBorderSelected	CSS (hex value) color set for selected element Border in Klarna Payments iFrame
Button Color Preference	kpColorButton	CSS (hex value) color set for Button in Klarna Payments iFrame
Button Text Color Preference	kpColorButtonText	CSS (hex value) color set for Button text in Klarna Payments iFrame
Checkbox Color Preference	kpColorCheckbox	CSS (hex value) color set for Checkbox in Klarna Payments iFrame
Checkbox Checkmark Color Preference	kpColorCheckboxCheckmark	CSS (hex value) color set for checkbox checked(selected) in Klarna Payments iFrame
Details Color Preference	kpColorDetails	CSS (hex value) color set for details in Klarna Payments iFrame
Header Color Preference	kpColorHeader	CSS (hex value) color set for Header in Klarna Payments iFrame
Rate limit By Operation	kpRateLimitByOperation	Select “Rate Limit By Operation” to Yes. If it is selected to NO, the default service profile will be utilized. The standard API rate limit for the Klarna service is as mentioned listed on <a href="#">docs.klarna.com</a> . The default service id is <a href="#">klarna.http.defaultendpoint</a> .
Klarna Payment Create New Session When Expires	kpCreateNewSessionWhenExpires	If set to Yes, then a new Klarna session will be created if Klarna session expires before SFCC basket expires

Link Color Preference	kpColorLink	CSS (hex value) color set for link in Klarna Payments iFrame
Text Color Preference	kpColorText	CSS (hex value) color set for text in Klarna Payments iFrame
Secondary Text Color Preference	kpColorTextSecondary	CSS (hex value) color set for secondary text in Klarna Payments iFrame
Border Radius Preference	kpRadiusBorder	Value (in pixels) of the border radius to be used in Klarna Payments iFrame
Attachments	kpAttachments	Toggle (Yes/No) for the inclusion of attachments when creating an order. Specific to inclusion of EMD (customer_account_info, other_delivery_address) when applicable. Default is "No".
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.
Virtual Card Number Enabled	kpVCNEnabled	If this option is set to "Yes", Klarna settlement request will generate a Virtual Card Number for every Klarna order. Note: the option will only work if VCN private/public keys are configured properly as mentioned below and public key shared in advance with Klarna
VCN Public Key ID	kpVCNkeyId	UUIDv4 value corresponding to the key pair. Shared with Klarna representative for Production & Playground (test) env configuration

VCN Private Key	vcnPrivateKey	SSL private key used only to decode Virtual Card information (used with kpVCNEncoded). Refer to section 9.3 Decrypt VCN Card Details
VCN Public Key	vcnPublicKey	SSL public key used with Virtual Card integration (used with kpVCNEncoded). Shared with Klarna and stored here for reference.
VCN Settlement Retry Enabled	kpVCNRetryEnabled	If set to "Yes", SFCC will retry the VCN settlement once again in case of service error. Default is "No"
Promotion Price Taxation	kpPromoTaxation	Only use "Based on Adjusted Price" value if you have enabled the corresponding value in "Merchant Tools > Site Preferences > Promotions > Discount Taxation" and use gross taxation. Default: Based on Price
Hide Payment Methods on Deny	kpRejectedMethodDisplay	If set to value other than "No", the Klarna payment method options on the checkout will be grayed out or not displayed to customer in the current view when Klarna authorization request is rejected in the response (.i.e hard reject - "show_form" and "approved" values are both "false")
Alternative Klarna Payment Flow	kpUseAlternativePaymentFlow	If set to "Yes", Klarna Authorization and Order creation steps will be triggered on the Checkout Review page when a customer clicks CTA/Place Order Button. Default: No
Enable OMS	kpOMSEnabled	If set to "Yes", it updates the order information ingested in SFOMS (SalesForce Order Management System) in the

		format that Klarna expects. It is required when there is an integration between SFCC and SFOMS. Default: No
Klarna Payments Additional Logging	kpAdditionalLogging	If set to "Yes", it writes additional logging info to get more order details when an issue occurs. Default: No
Agent User Name	kpAgentUserName	User Agent Name for orders on behalf
Agent User Password	kpAgentUserPassword	User Agent Password for orders on behalf
Enable Retry for Recurring Orders	kpEnableRecurringOrderRetry	Enable retry for failed orders
Number Of Retries	kpRecurringNumberOfRetry	Recurring number of retries
Recurring Retry Frequency	kpRecurringRetryFrequency	Retry frequency for recurring orders
Klarna Create Order Token	kpCreateOrderToken	Token used to verify the caller for recurring orders
Use Bank Transfer callback	kpBankTransferCallback	If set to "Yes", SFCC will wait for Klarna callback to place an order. Default is "No"

#### **Klarna Express checkout (Klarna\_ExpressCheckout)**

Klarna Express checkout Enabled	kpECEnabled	Boolean flag to enable/disable Klarna Express checkout
Klarna Express checkout Button Theme	kpECButtonTheme	The theme of the button. Options include default, light & dark

Klarna Express checkout Button Shape	kpECButtonShape	The shape of the button. Options include default, rect & pill
Placement	kec_placement	Multiselect attribute to choose where to display express checkout buttons. Values: cart, pdp, minicart

**Table 8. Klarna Payments Site Preferences****Klarna Payments (Klarna\_KP)**

Parameter Name	Attribute ID	Description
Enable Klarna Payments	kp_enable	Enable/Disable Klarna Payments.Boolean attribute with Default value as true
Color Customization	kpColorCustomization	JSON attribute to customize color
Enable Extra Merchant Data	kpEMD	Enable this option to include customer_account_info and other_delivery_address as attachments when creating an order.Boolean with default value false
VCN - Enable settlement retry	kpVCNRetry	When set to "Yes", SFCC will automatically retry the VCN settlement in the event of a service error. Boolean with default value false
Log Extra debug data	kpLogExtraData	Attribute to log debug data.Boolean with default value true
Merchant Reference 2	merchant_reference2	Enter the attribute from the SCC order (basket) object that you would like to forward as an additional merchant reference to Klarna. This attribute must be a class attribute of the SCC LineItemCtnr. String attribute

Agent username for subscriptions	<code>kpSubsUsername</code>	Agent username used to login on behalf of customers to create subscription orders.String attribute
Agent password for subscriptions	<code>kpSubsPassword</code>	Agent password to login on behalf of customers to create subscription orders.Password Attribute
Enable retry for subscriptions	<code>kpSubsRetryEnable</code>	Boolean with default value false
Number of Retries	<code>kpSubsRetryNumber</code>	Number attribute
Recurring retry frequency	<code>kpSubsRetryFrequency</code>	Number attribute

### Klarna Express checkout (Klarna\_KEC)

Parameter Name	Attribute ID	Description
Enable Express Checkout	<code>kec_enable</code>	Boolean flag to enable/disable Klarna Express checkout
Theme	<code>kec_theme</code>	The theme of the button. Options include default, light & dark
Button Shape	<code>kec_shape</code>	The shape of the button. Options include default, rect & pill
Placement	<code>kec_placement</code>	Multiselect attribute to choose where to display express checkout buttons. Values: cart, PDP, minicart

Table 9. Klarna Express checkout Site Preferences

## Klarna On-site messaging

Parameter Name	Attribute	Description
Enable On-Site Messaging	osm_enable	Enable/Disable OSM functionality
Theme	osm_theme	Theme for on site messaging .Options include default,dark and custom
Placement	osm_placement	Multiselect dropdown which includes cart,pdp,header,footer and info. Placement Tag IDs are hardcoded in code.
Custom styling	osm_custom_styling	JSON to store custom styling

Table 11. Klarna On-site messaging Site Preferences

### 6.1.1.6. Product

Parameter Name	Attribute	Description
Is Klarna Standard Product	kpIsStandardProduct	Boolean attribute to define if the product is standard.
s Klarna Subscription Product	kpIsSubscriptionProduct	Boolean attribute to define if the product is eligible for subscription.
Klarna Trial Days Usage	kpTrialDaysUsage	Numeric value used for free trial definition.

Table 12. Product parameters for Klarna subscriptions

### 6.1.1.7. ProductLineItem

Parameter Name	Attribute ID	Description
Is Selected for Subscription product	kpSubscription	Boolean attribute to define if the product is selected for subscription in the basket.
Enriched data for product line item.	klarna_oms_lineItemJSON	String data for product line item.

Table 13. ProductLineItem parameters

### 6.1.1.8. Profile

Parameter Name	Attribute ID	Description
Klarna Subscriptions	kpSubscriptions	Text attribute to store customer subscriptions to Klarna

Table 14. Klarna Subscriptions profile

## 6.1.2. Custom Objects

### 6.1.2.1. Klarna Express Button



### This functionality is deprecated as of release 24.4.0

Klarna Express Button has been deprecated and replaced with [Klarna Express Checkout](#).

The Klarna Express Button (KEB) can be configured and adapted to fit your storefront's needs. Follow these steps to enable the KEB button:

#### 1) White-listing MID and Domains:

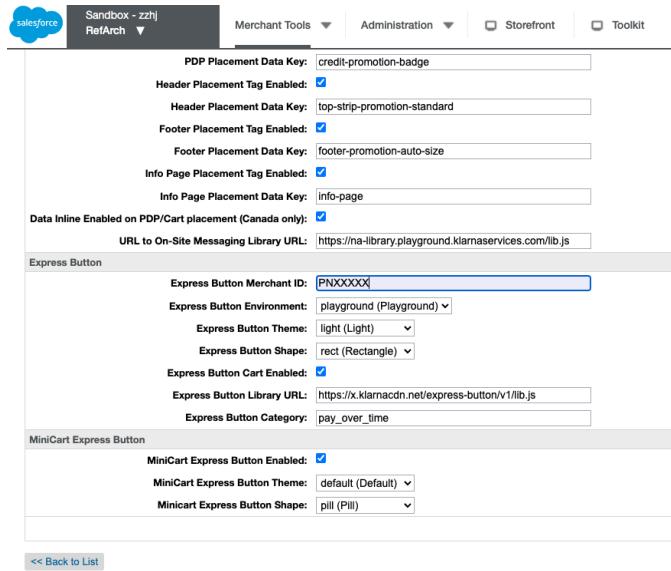
- a) Your Klarna Merchant ID (MID) and domains must be white-listed. Contact your Klarna delivery manager for support with this process before testing and going live, as the white-listed domains may vary.
- b) Ensure that the MID matches the one configured in the Business Manager (BM) service credentials for your storefront.

#### 2) Configuring KEB Settings:

- a) Navigate to "Merchant Tools – Custom Object Editor" and search for the **KlarnaCountries** custom object.
- b) Select the respective country key (e.g., "US") and provide the following details:
  - i) **Express Button Merchant ID:** The allow-listed Merchant ID.
  - ii) **Express Button Environment:** Choose the Klarna environment (Production/Playground).
  - iii) **Express Button Cart Enabled:** Check this box to display the button on the cart page.
  - iv) **MiniCart Express Button Enabled:** Check this box to display the button on the minicart page.
  - v) **MiniCart Express Button Theme/Shape:** Set the theme/shape for the button on the minicart page.
  - vi) **Express Button Library URL:** The URL for the button library.
  - vii) **Express Button Category:** The Klarna category to be pre-selected Refer to [Figure 54](#) for a visual guide on completing the configuration.at checkout (e.g., "pay\_over\_time").

#### 3) Shopper Experience:

- a) The shopper lands on the cart page where the KEB is displayed.
- b) The shopper clicks on the KEB button.
- c) The shopper is prompted to share their Klarna registered credentials to complete authentication.
- d) The shopper is redirected to checkout with the Klarna Payment Method pre-selected.



**Figure 54. Klarna Express Button completed configuration**

More customization options for the button can be found [here](#).



**Note:** The Klarna Express Button is currently available in specific markets, with more to follow.

#### 6.1.2.2. KlarnaCountries



#### This functionality is deprecated as of release 24.4.0

This object is deprecated from version 24.4.0, and replaced with [Klarna Activation](#) object.

The respective object is dynamically selected based on the requested locale country, e.g., SFCC site with locale `de_DE` or `en_DE` will use the `DE` custom object. In cases when the requested locale country can't be dynamically resolved (i.e. with "default" SFCC locale) – attribute `klarnaLocale` can be utilized to pass the proper locale to Klarna. For all other cases, this field can be left blank and will not be taken into consideration.

Even if you have locales that are not supported by Klarna Payments, we recommend creating a corresponding entry in the custom object for that locale. Thus, on the billing page of the unsupported locale you will have the Klarna Payments widget showing an appropriate message.

The custom objects store data such as Klarna default locale, service credential IDs and Klarna Payments placement data keys to ensure that Klarna Payments integration is correctly configured.

Administration > Site Development > Custom Object Types > KlarnaCountries - Attribute Definitions																																																																																																	
		General		Attribute Definitions																																																																																													
				Attribute Grouping																																																																																													
<b>Object Type 'KlarnaCountries'</b>																																																																																																	
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.																																																																																																	
<div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 5px;"> <b>Search Attribute Definitions</b>          ID or Name: <input type="text"/> Find       </div> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>Select All</th> <th>ID</th> <th>Name</th> <th>Type</th> <th>Attribute Settings</th> <th>Values</th> <th></th> </tr> </thead> <tbody> <tr> <td><input type="checkbox"/></td> <td> <code>UUID</code></td> <td>UUID</td> <td>String</td> <td>*</td> <td>0</td> <td><a href="#">Edit</a></td> </tr> <tr> <td><input type="checkbox"/></td> <td> <code>country</code></td> <td>Country Code</td> <td>String</td> <td>*</td> <td>0</td> <td><a href="#">Edit</a></td> </tr> <tr> <td><input type="checkbox"/></td> <td> <code>creationDate</code></td> <td>Creation Date</td> <td>Date+Time</td> <td>*</td> <td>0</td> <td><a href="#">Edit</a></td> </tr> <tr> <td><input type="checkbox"/></td> <td><code>credentialID</code></td> <td>Service Credential ID</td> <td>String</td> <td></td> <td>0</td> <td><a href="#">Edit</a></td> </tr> <tr> <td><input type="checkbox"/></td> <td><code>klarna_locale</code></td> <td>Klarna Locale</td> <td>String</td> <td></td> <td>0</td> <td><a href="#">Edit</a></td> </tr> <tr> <td><input type="checkbox"/></td> <td> <code>lastModified</code></td> <td>Last Modified</td> <td>Date+Time</td> <td>*</td> <td>0</td> <td><a href="#">Edit</a></td> </tr> <tr> <td><input type="checkbox"/></td> <td><code>csmCartEnabled</code></td> <td>Cart Placement Tag Enabled</td> <td>Boolean</td> <td></td> <td>0</td> <td><a href="#">Edit</a></td> </tr> <tr> <td><input type="checkbox"/></td> <td><code>csmCartTagId</code></td> <td>Cart Placement Tag ID</td> <td>String</td> <td></td> <td>0</td> <td><a href="#">Edit</a></td> </tr> <tr> <td><input type="checkbox"/></td> <td><code>csmLibraryUrl</code></td> <td>Library URL</td> <td>String</td> <td></td> <td>0</td> <td><a href="#">Edit</a></td> </tr> <tr> <td><input type="checkbox"/></td> <td><code>csmPDFEnabled</code></td> <td>PDF Placement Tag Enabled</td> <td>Boolean</td> <td></td> <td>0</td> <td><a href="#">Edit</a></td> </tr> <tr> <td><input type="checkbox"/></td> <td><code>csmPDFTagId</code></td> <td>PDF Placement Tag ID</td> <td>String</td> <td></td> <td>0</td> <td><a href="#">Edit</a></td> </tr> <tr> <td><input type="checkbox"/></td> <td><code>csmUCI</code></td> <td>On-site messaging UCI</td> <td>String</td> <td></td> <td>0</td> <td><a href="#">Edit</a></td> </tr> </tbody> </table>							Select All	ID	Name	Type	Attribute Settings	Values		<input type="checkbox"/>	<code>UUID</code>	UUID	String	*	0	<a href="#">Edit</a>	<input type="checkbox"/>	<code>country</code>	Country Code	String	*	0	<a href="#">Edit</a>	<input type="checkbox"/>	<code>creationDate</code>	Creation Date	Date+Time	*	0	<a href="#">Edit</a>	<input type="checkbox"/>	<code>credentialID</code>	Service Credential ID	String		0	<a href="#">Edit</a>	<input type="checkbox"/>	<code>klarna_locale</code>	Klarna Locale	String		0	<a href="#">Edit</a>	<input type="checkbox"/>	<code>lastModified</code>	Last Modified	Date+Time	*	0	<a href="#">Edit</a>	<input type="checkbox"/>	<code>csmCartEnabled</code>	Cart Placement Tag Enabled	Boolean		0	<a href="#">Edit</a>	<input type="checkbox"/>	<code>csmCartTagId</code>	Cart Placement Tag ID	String		0	<a href="#">Edit</a>	<input type="checkbox"/>	<code>csmLibraryUrl</code>	Library URL	String		0	<a href="#">Edit</a>	<input type="checkbox"/>	<code>csmPDFEnabled</code>	PDF Placement Tag Enabled	Boolean		0	<a href="#">Edit</a>	<input type="checkbox"/>	<code>csmPDFTagId</code>	PDF Placement Tag ID	String		0	<a href="#">Edit</a>	<input type="checkbox"/>	<code>csmUCI</code>	On-site messaging UCI	String		0	<a href="#">Edit</a>
Select All	ID	Name	Type	Attribute Settings	Values																																																																																												
<input type="checkbox"/>	<code>UUID</code>	UUID	String	*	0	<a href="#">Edit</a>																																																																																											
<input type="checkbox"/>	<code>country</code>	Country Code	String	*	0	<a href="#">Edit</a>																																																																																											
<input type="checkbox"/>	<code>creationDate</code>	Creation Date	Date+Time	*	0	<a href="#">Edit</a>																																																																																											
<input type="checkbox"/>	<code>credentialID</code>	Service Credential ID	String		0	<a href="#">Edit</a>																																																																																											
<input type="checkbox"/>	<code>klarna_locale</code>	Klarna Locale	String		0	<a href="#">Edit</a>																																																																																											
<input type="checkbox"/>	<code>lastModified</code>	Last Modified	Date+Time	*	0	<a href="#">Edit</a>																																																																																											
<input type="checkbox"/>	<code>csmCartEnabled</code>	Cart Placement Tag Enabled	Boolean		0	<a href="#">Edit</a>																																																																																											
<input type="checkbox"/>	<code>csmCartTagId</code>	Cart Placement Tag ID	String		0	<a href="#">Edit</a>																																																																																											
<input type="checkbox"/>	<code>csmLibraryUrl</code>	Library URL	String		0	<a href="#">Edit</a>																																																																																											
<input type="checkbox"/>	<code>csmPDFEnabled</code>	PDF Placement Tag Enabled	Boolean		0	<a href="#">Edit</a>																																																																																											
<input type="checkbox"/>	<code>csmPDFTagId</code>	PDF Placement Tag ID	String		0	<a href="#">Edit</a>																																																																																											
<input type="checkbox"/>	<code>csmUCI</code>	On-site messaging UCI	String		0	<a href="#">Edit</a>																																																																																											

**Figure 55. KlarnaCountries Attributes**



**Note:** The same custom object is used by Klarna Checkout cartridge integration.

The table below describes attributes of the KlarnaCountries custom object:

Attribute Name	Attribute ID	Description
Country Code	<code>country</code>	Two-letter country code

On-site Messaging Data Default Locale	klarnaLocale	Fallback, if the request locale can't be dynamically resolved, i.e., when using "default" SFCC locale
Service Credential ID	credentialID	The ID of service credentials for this locale.
On-site messaging Data Client ID	osmUCI	The Klarna On-site messaging "data-client-id" applicable for a given country
Cart Placement Tag Enabled	osmCartEnabled	To enable Cart Placement for a given locale.
Cart Placement Tag ID	osmCartTagId	The Klarna On-site messaging "data-key" of placement applicable for Cart Page for a given locale.
Cart Placement Theme	osmCartTheme	The theme for cart placement tag. It could be default or dark.
Cart Placement Custom Styling (#osm-cart-placement)	osmCartCustomStyling	Custom styling for cart placement. The css should be wrapped in <style>...customCss...</style> element.
PDP Placement Tag Enabled	osmPDPEnabled	To enable PDP Placement for a given locale.
PDP Placement Tag ID	osmPDP TagId	The Klarna On-site messaging "data-key" of placement applicable for the Product Display page for a given locale.
PDP Placement Theme	osmPDPTheme	The theme for PDP placement tag. It could be default or dark.

PDP Placement Custom Styling (#osm-pdp-placement)	osmPDPCustomStyling	Custom styling for pdp placement. The css should be wrapped in <style>...customCss...</style> element.
Header Placement Tag Enabled	osmHeaderEnabled	To enable Klarna Header Placement in a given storefront
Header Placement Data Key	osmHeaderTagId	The Klarna On-site messaging "data-key" of placement applicable for the Header
Header Placement Theme	osmHeaderTheme	The theme for header placement tag. It could be default or dark.
Header Placement Custom Styling (#osm-header-placement)	osmHeaderCustomStyling	Custom styling for header placement. The css should be wrapped in <style>...customCss...</style> element.
Footer Placement Tag Enabled	osmFooterEnabled	To enable Klarna footer Placement in a given storefront
Footer Placement Data Key	osmFooterTagId	The Klarna On-site messaging "data-key" of placement applicable for the footer
Footer Placement Theme	osmFooterTheme	The theme for footer placement tag. It could be default or dark.
Footer Placement Custom Styling (#osm-footer-placement)	osmFooterCustomStyling	Custom styling for footer placement. The css should be wrapped in <style>...customCss...</style> element.

Info Page Placement Tag Enabled	osmInfoPageEnabled	To enable Klarna Info Page Placement in a given storefront
Info Page Placement Data Key	osmInfoPageTagId	The Klarna On-site messaging "data-key" of placement applicable for the Info Page
Info Page Placement Theme	osmInfoPageTheme	The theme for info page placement tag. It could be default or dark.
Info Page Placement Custom Styling (#osm-info-page-placement)	osmInfoPageCustomStyling	Custom styling for info page placement. The css should be wrapped in <style>...customCss...</style> element.
Data Inline Enabled on PDP/Cart placement (Canada only)	osmDataInlineEnabled	Enable this when using PayBright payment method in Canada
URL to On-Site Messaging Library URL	osmLibraryUrl	<p>URL for On-Site Messaging library, applicable for testing or production must be saved.</p> <p>Please use only Klarna production URL in live storefront. Verify test environment URL which includes "playground" in URL For production or live environment, ensure URL includes "production". E.g: Old library Test URL&gt;</p> <p><a href="https://na-library.playground.klarnaservices.com/lib.js">https://na-library.playground.klarnaservices.com/lib.js</a></p> <p>, e.g: Live URL&gt;<a href="https://na-library.production.klarnaservices.com/lib.js">https://na-library.production.klarnaservices.com/lib.js</a></p>

		New library - <a href="https://js.klarna.com/web-sdk/v1/klarna.js">https://js.klarna.com/web-sdk/v1/klarna.js</a>
On-Site Messaging Environment	osmEnvironment	Environment on which the OSM is used. It could be playground or production.
Express Button Environment	kebEnvironment	The express button environment. Default is playground
Express Button Merchant ID	kebMerchantID	The merchant ID used to display the button
Express Button Library URL	kebLibraryUrl	URL of the express button library
Express Button Cart Enabled	kebCartEnabled	To enable the Klarna Express Button on cart page
Express Button Theme	kebTheme	The theme of the button. Options include default, light & dark
MiniCart Express Button Enabled	kebMCEnabled	To enable the Klarna Express Button on minicart
MiniCart Express Button Theme	kebMCTheme	The theme of the button. Options include default, light & dark
Express Button Category	kebCategory	The Klarna category to be pre-selected on Checkout Page, e.g., "pay_over_time"
Klarna Express checkout Client Key	expressCheckoutClientKey	The Klarna Express client key used for express checkout

**Table15. KlarnaCountries Attributes**

The `data-client-id` and `data-key` values used in the OSM placements are available in the Klarna Merchant Portal (Europe/US (CA included)/Oceania) within the On-site Messaging App. When selecting the `data-key` values, ensure that the filter is set to the right country and language.

### 6.1.2.3. Klarna Activation

KlarnaActivation custom object gives an opportunity to configure multiple MIDs per site. Merchants could have different credentials, clientIDs and VCN keys per country or group of countries. It has a similar structure to the Klarna Activation Site preferences group. Klarna Activation Key is the unique key for each Klarna Activation CO entry. It should be defined by the merchant and could have any unique string value.

This screenshot shows the 'Attribute Definitions' tab for the 'KlarnaActivation' object type. It lists various attributes with their IDs, names, types, and settings. The attributes include:

ID	Name	Type	Attribute Settings	Values	Action
<code>KP_API_Password_countries</code>	API Password	Password	*	0	Edit
<code>KP_API_Username_countries</code>	API Username	String	*	0	Edit
<code>UUID</code>	UUID	String	*	0	Edit
<code>creationDate</code>	Creation Date	Date+Time	*	0	Edit
<code>kpVCNEnabled_countries</code>	Enable Virtual Card Number (VCN)	Boolean		0	Edit
<code>kpVCNRetry_countries</code>	Enable settlement retry	Boolean		0	Edit
<code>kpVCNkeyId_countries</code>	Public Key ID	String		0	Edit
<code>kp_activation_key</code>	Klarna Activation Key	String	*	0	Edit
<code>kp_client_id_countries</code>	Client ID	String	*	0	Edit
<code>kp_market_countries</code>	Market(s)	Enum of Strings	*	26	Edit
<code>kp_region_countries</code>	Region	Enum of Strings	*	3	Edit
<code>lastModified</code>	Last Modified	Date+Time	*	0	Edit

Figure 56. Klarna Activation attributes

Attribute Name	Attribute ID	Description
Klarna Activation Key	<code>kp_activation_key</code>	Unique identifier for a single KlarnaActivation entry. It is a free text entered by the merchant.
Region	<code>kp_region_countries</code>	Klarna regions (Europe, North America, Oceania)
Market(s)	<code>kp_market_countries</code>	Klarna available countries - <a href="#">List of Klarna Countries</a>

Attribute Name	Attribute ID	Description
Client ID	kp_client_id_countries	Client ID generated by Klarna
API Username	KP_API_Username_countries	Username for Klarna services
API Password	KP_API_Password_countries	Password for Klarna services
Enable Virtual Card Number (VCN)	kpVCNEnabled_countries	Flag to enable virtual card number for selected countries. This will override the globally selected in Klarna Payments.
Public Key ID	kpVCNkeyId_countries	Countries specific public key for VCN
Enable settlement retry	kpVCNRetry_countries	Flag to enable settlement retry for selected countries. This will override the globally selected in Klarna Payments.

Table 16. Klarna Activation site preference

### 6.1.3. Session Attributes & Cookies

The following session custom attributes are saved in `session.privacy` storage and accessible in checkout. The attributes are retained for the session lifetime & cleared when the customer logs out of their profile.

Attribute	Description
<code>KlarnaLocale</code>	The Klarna locale in use
<code>KlarnaPaymentsSessionID</code> <b>(Not applicable since version 21.2.0)</b>	The Klarna session ID returned after "Create Session" API endpoint is called

<code>KlarnaPaymentsClientToken</code> <b>(Not included as session attributes since version 21.2.0)</b>	Client token returned by "Create Session" API endpoint and used to initialize the JS SDK
<code>KlarnaPaymentMethods</code>	The available payment method categories for the respective Klarna session; Saved in JSON format
<code>KlarnaPaymentsAuthorizationToken</code>	The authorization token returned by JS SDK "Authorize" call
<code>KPAuthInfo</code>	Whether finalization is required for the payment method; Returned by JS SDK "Authorize" call; Saved in JSON format
<code>KlarnaExpressCategory</code>	The KEB payment category; Currently applicable for US and defaults to "pay_over_time"
<code>KlarnaPaymentsRedirectURL</code>	The URL to redirect the customer to after placing the order; Returned by the "Create Order" API call
<code>'kpActive_ ' + countryCode</code>	Flag to indicate if Klarna is enabled for current site country
<code>'kpActivationSource_ ' + countryCode</code>	Klarna Activation source per country - Custom Object or Site Preferences. It is empty for old Klarna Countries config
<code>'kpActivationKey_ ' + countryCode</code>	Klarna Activation key in case of Custom object usage.

**Table 17 Klarna Session Attributes**

The following cookies are being set by Klarna integration:

Cookie Name	Description
<code>selectedKlarnaPaymentCategory</code>	The selected payment method on checkout (e.g. "pay_later")

Table 18. Klarna Cookies

### 6.1.4. Library

In addition to the configurations, the following two library assets will be added:

- **footer-about**: An updated Out-of-the-Box (OOTB) asset that includes a link to the Klarna On-site messaging (OSM) dedicated page in the footer.
- **klarna-email-info**: An asset containing links to review Klarna Payment information, which is used in the confirmation email sent to customers.

### 6.1.5. Services

An HTTP service, `klarna.http.defaultendpoint`, has been added with the `klarna.http.service` profile.



### Deprecation notice

- **Deprecation Notice for Version 24.4.0:** Service credentials and the `KlarnaCountries` custom object have been deprecated as of version 24.4.0. Please use the Klarna Activation Site Preferences or the Klarna Activation custom object to enter API credentials.
- **Replication Guidance for Version 21.2.0:** Prior to version 21.2.0, the `KlarnaCountries` custom object was replicable. To avoid issues with service credentials during replication, merchants should use the same service credential name across staging, development, and production environments.

For more details on updating the `KlarnaCountries` definition in your instances, please review Section [Update KlarnaCountries Definition](#).

## 6.2. Logs

The integration includes the following types of logs:

- **Service Communication Logs:** These logs start with `service-klarna-***` and contain every request and response to the Klarna endpoints. Personal information, such as emails and names required for the Klarna API calls, is masked in these logs.
- **Custom Errors and Debug Info:** Depending on the case, custom errors and debug information are logged under `customerror-***`, `custodebug-***`, and `custominfo-***` files.

## 6.3. Availability

Cartridge functionality is dependent on the availability of the Klarna API service. The current operational status of Klarna can be viewed at [Klarna Status](#).

## 6.4. Failover/Recovery Process

If the Klarna API is not available, Klarna will not be presented as a payment option. In the event of any failure within the Klarna API, contact Klarna support for assistance.

## 6.5. Support

A customer service workshop can be conducted during the implementation process before going live to align operational processes and ensure customer satisfaction. Klarna provides all customers with access to the Klarna App via the website [Klarna App Login](#) or by downloading the Klarna App (available for free on Android/iOS). Through the app, customers can contact support, view their statements, pay for purchases, track delivery updates, and extend due dates if they choose to pay after delivery.

### 6.5.1. Merchant Support

For reporting core SFCC functionality issues in the Klarna cartridge technical integration, please contact: [commercecloud@klarna.com](mailto:commercecloud@klarna.com).

For production issues related to Klarna API availability, merchant representatives should reach out to their Klarna Account Manager after reviewing the current operational status at [Klarna Status](#). If there are suspicions about degraded performance or issues with Klarna's service, report the problem in production (Post Go-live). The Klarna contact will then report this internally to the incident management team, which has established routines to handle and resolve reported incidents. The Klarna contact may request additional information from the individual reporting the problem to help the internal team ascertain and identify the issue. The KAM may also advise the merchant to follow updates on the status page if it is a known incident with ongoing updates.

#### **Pre-requisite Information to Provide When Reporting an Incident:**

- Merchant's affected MID or market
- Impact and examples of customer orders (order\_id or Klarna session\_id if available)
- Screenshots, timeframe, and any additional information as required

This information helps speed up the investigation and resolution process.

# 7. User Guide

## 7.1. Cartridge Upgrade

Regular updates to our cartridge code include bug fixes, performance enhancements, security patches, and new features. Staying up to date with these changes is essential for maintaining compatibility, security, and optimal functionality of your integration.

### 7.1.1. Upgrade Process

Follow these steps to upgrade your custom code with the latest version of our cartridge:

#### 1) Review Release Notes

Start by reviewing the release notes for the latest version of the cartridge. The release notes outline changes, improvements, and any potential compatibility considerations associated with the upgrade.

#### 2) Assess Custom Code Changes

Identify any customizations or modifications you have made to the SFCC cartridge code in your custom files. These may include storefront customizations, controller adjustments, or custom business logic built on top of our cartridge.

#### 3) Backup Custom Files

Before proceeding with the upgrade, ensure that you have a backup of your custom files, including any modifications made to the SFCC cartridge code. This backup will serve as a safety net in case any issues arise during the upgrade process.

#### 4) Compare Code Differences

Use a version control system or a file comparison tool to compare the differences between your custom code and the latest version of the SFCC cartridge code. Pay close attention to areas where changes have been made to ensure compatibility and maintain functionality.

## **5) Update Integration Code**

Integrate the latest version of the SFCC cartridge code into your custom files, replacing any outdated or deprecated code with the new implementations. Follow the migration guides and best practices provided to ensure a smooth transition.

## **6) Test and Validate**

After updating your custom code, thoroughly test the integration to ensure that all functionality works as expected. Test various scenarios, including user interactions, data processing, and third-party integrations, to identify any potential issues or regressions.

## **7) Address Compatibility Issues**

If you encounter any compatibility issues or conflicts with your existing custom code, troubleshoot and resolve them accordingly. Consult our support resources or reach out to our team for assistance in addressing compatibility concerns.

## **8) Deploy Changes**

Once you are satisfied with the upgrade and have validated its functionality, deploy the changes to your production environment. Monitor the integration closely following deployment to ensure ongoing stability and performance.

## **7.2. Roles and Responsibilities**

There are no recurring tasks required by the merchant. Once configurations are set up, the functionality runs on demand.

## **7.3. Storefront Functionality**

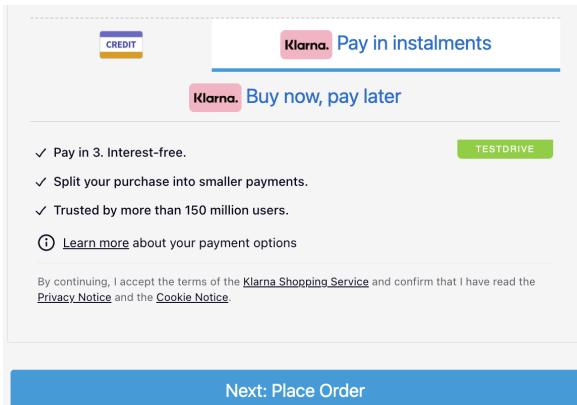
When Klarna is set up, Klarna Payment options and iframe widgets will be shown on the billing step. All SFCC out-of-the-box (OOTB) checkout functionality remains in place, such as:

- 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 a bonus product

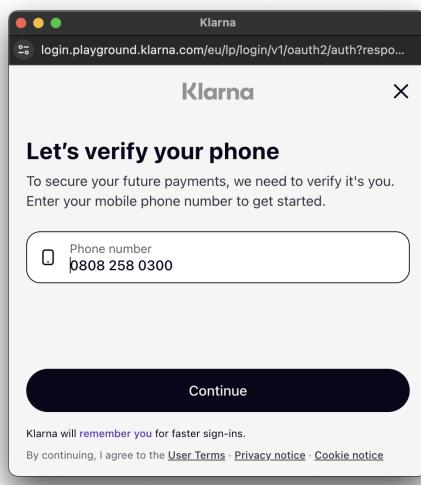
To use Klarna's payment options on the billing step of the checkout process:

- 1) Select one of Klarna's payment options as the payment method.
- 2) Click the "Next: Place Order" button:



**Figure 57. Payment Options on Checkout**

- 3) Depending on the payment method selected and the region, a Klarna popup window will appear. Follow the steps on the screen:



**Figure 58. Klarna Pop Up Screen**

- 4) On the Review step click on “Place Order” button:

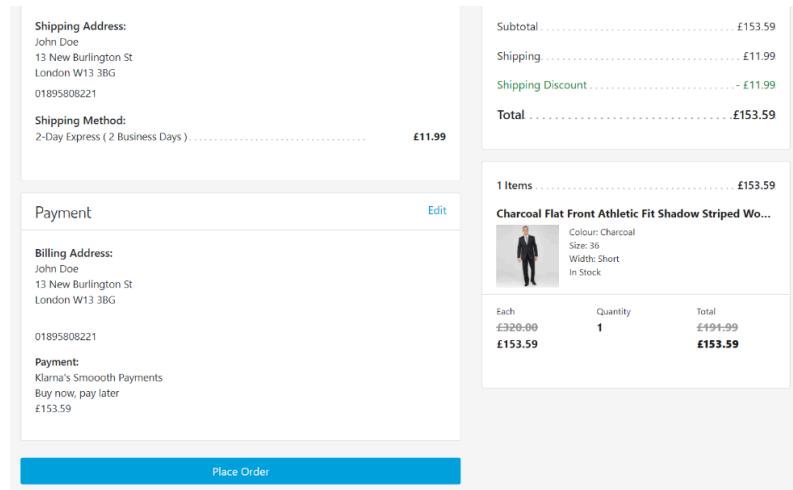


Figure 59. Payment Review Screen

- 5) The customer’s browser is sent to the `redirect_url` and immediately thereafter shown the Commerce Cloud Order Confirmation page:

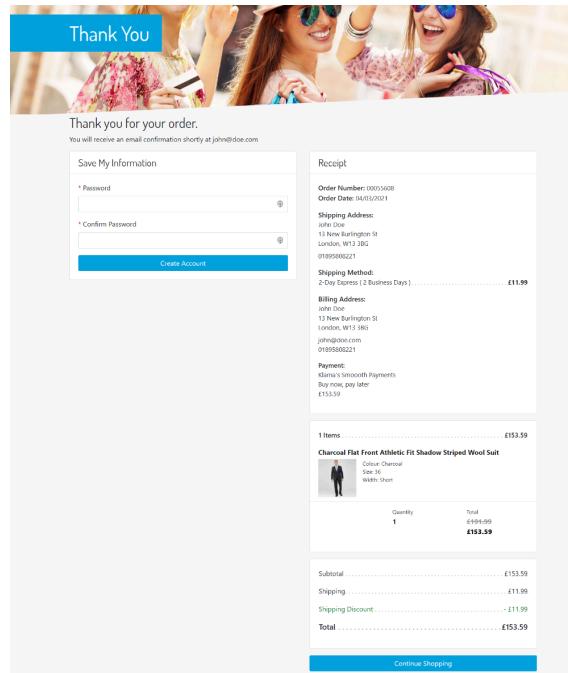


Figure 60. Order Confirmation Page

6) The newly created order can be inspected in Business Manager:

Orders

You're using the 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 or newline. Entered text is treated as case-sensitive; substring matching isn't supported.

Order Search					
Number	Order Date	Site	Created By	Registration Status	
00055608	3/4/21 4:38:34 pm Etc/UTC	RefArchGlobal	Customer	Unregistered	
00055506	3/3/21 6:13:04 pm Etc/UTC	RefArchGlobal	Customer	Registered	
00055409	3/2/21 10:07:53 am Etc/UTC	RefArchGlobal	Customer	Unregistered	
00055311	3/1/21 10:07:01 pm Etc/UTC	RefArchGlobal	Customer	Registered	
00055309	3/1/21 9:55:21 pm Etc/UTC	RefArchGlobal	Customer	Registered	
00055307	3/1/21 9:47:04 pm Etc/UTC	RefArchGlobal	Customer	Registered	
00055306	3/1/21 6:57:04 pm Etc/UTC	RefArchGlobal	Customer	Registered	
00054903	2/19/21 10:10:07 am Etc/UTC	RefArchGlobal	Customer	Registered	
00054802	2/18/21 9:15:16 am Etc/UTC	RefArchGlobal	Customer	Registered	
00054505	2/10/21 12:33:17 pm Etc/UTC	RefArchGlobal	Customer	Registered	

Showing 1 - 10 of 393 items

Show [50](#) [100](#) [All](#) items

Figure 61. Orders List in BM

a) Klarna Payments order ID can be inspected in the Attributes tab of the order:

Merchant Tools > Ordering > Orders > Order: 00055608(RefArchGlobal)

General    **Attributes**    Payment    Notes    History

Attributes for Order '00055608'

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	
Klarna Payments Order ID:	6fa5c3ad-411d-2629-b543-4f5ea672ea9e
Is VCN Used:	<input type="checkbox"/>
VCN Card ID:	<input type="text"/>

[<< Back to List](#)

Figure 62. Order Attributes

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

Merchant Tools > Ordering > Orders > Order: 00055608(RefArchGlobal)

General Attributes Payment Notes History

**Payment Information for Order '00055608'**

Order Total:	£153.59
Amount Paid:	£0.00
Balance Due:	£153.59
Invoice Number:	00242008
Payment Status:	Paid
Payment Method:	KLARNA_PAYMENTS Processor: KLARNA_PAYMENTS Transaction: 6fa5c3ad-411d-2629-b543-4f5ea672ea9e Amount: £153.59
Klarna Payment Category ID: pay_over_time Klarna Payment Category Name: Buy now, pay later Fraud Status: ACCEPTED	

<< Back to List

**Figure 63. Order Payment Detail**

- c) The order can be further inspected in the Klarna Merchant Portal:
  - i) EU: [eu.portal.klarna.com](https://eu.portal.klarna.com)
  - ii) US: [us.portal.klarna.com](https://us.portal.klarna.com)
  - iii) OC: [us.portal.klarna.com](https://us.portal.klarna.com)

#7BRVXW4H Captured £153.59

Merchant reference 1 <b>00055608</b> Edit	Merchant reference 2 <b>cf6bf55f8f98bc392030178105</b> Edit	Created <b>Mar 4, 2021, 6:38 PM</b>	Expires <b>Apr 1, 2021, 3:00 AM</b>	Merchant ID <b>K500726</b>
---	---	--	--	-------------------------------

**Customer**

**Shipping address**

**John Doe**  
13 New Burlington St  
London  
W13 3BG  
GB  
Tel  
**01895808221**  
Email  
**john@doe.com**  
Edit shipping address

**Billing address**

**Additional Info**

**Order lines (2)**

Item / Reference	Qty	Unit price	Discount	Tax	Amount
Charcoal Flat Front Athletic Fit Shadow Striped Wool Suit 640188017003M	1	191.99	38.40	5% 7.31	£153.59 Captured
2-Day Express GBP002	1	11.99	11.99	5% 0.00	£0.00 Captured

**Currency: GBP**

**PAYMENT DETAILS**

Initial Payment Method  
Pay later in parts  
VISA 41111\*\*\*\*\*1111  
Resend statement

**ORDER TOTAL** £153.59

Captured £153.59  
Refunded £0.00  
Not Captured £0.00

**CUSTOMER BILLED** £153.59

**Activity Log**

Mar 4, 2021 6:38 PM  
8 minutes ago  
Captured: £153.59  
Via API

6:38 PM  
Order placed: £153.59  
By Klarna

**Figure 64. Klarna Portal Order View**

# 8. Release History

Version	Date	Changes
18.1.0		Initial release of Klarna Payments SFRA.
19.1.0	June 2019	Added SFRA version
19.1.1		Updated VCN to use the newest API version
19.1.2		Fix auto capture for the pipelines cartridge
19.1.4		New country locales added. Minor bug fixes. Cartridge templates and forms updated for latest SFRA.
19.1.5		Added additional verification for all notifications. Minor fixes around the configuration objects. Added Canadian support. Documentation updates.

<b>Version</b>	<b>Date</b>	<b>Changes</b>
19.1.6		<p>New country locales added.</p> <p>Updated VCN to store encrypted card details</p>
21.1.0	March 2021	<p>Fixes around discounts taxation &amp; VCN error handling.</p> <p>Added VCN improvements, additional OSM placements, BOPIS support. New IT, CA, FR &amp; NZ country locales.</p> <p>Removed acknowledge call.</p> <p>Documentation updates.</p>
21.1.1	March 2021	<p>New On-Site Messaging attribute for Canada.</p> <p>Remove not required locale templates for SG Spain &amp; Belgium.</p> <p>Documentation updates.</p>
21.1.2	April 2021	<p>Fixed core file naming convention issues in 21.1.0 and 21.1.1. Please upgrade to the latest version if you are currently using 21.1.0 or 21.1.1.</p> <p>Removed deprecated "scripts/util/Builder.js" file.</p>
21.2.0	June 2021	<p>Added Klarna Express Button.</p> <p>Moved Klarna session ID &amp; client token from SFCC session privacy to Basket attributes.</p> <p>Changed KlarnaCountries definition to not replicable.</p>

<b>Version</b>	<b>Date</b>	<b>Changes</b>
		Code cleanup. Documentation updates.
21.3.0	November 2021	Improvements for create_session errors Expired user session issues related to empty shipment.shippingMethod Additional locale (PL) included in config files
21.3.1	November 2021	Documentation updates.
22.1.0	February 2022	Improvements for create and update session errors Added Klarna Express Button in minicart. Support for long running basket Rate-limits by operations
22.2.0	March 2022	One Klarna Optimisation Mexico locale support
22.2.1	April 2022	Rollback of One Klarna Optimisation

<b>Version</b>	<b>Date</b>	<b>Changes</b>
22.3.0	May 2022	SFRA ver. 6.0.0 support Rollback hide VAT from Checkout functionality
22.3.1	May 2022	Fix User-agent version sent to Klarna services
22.4.0	July 2022	Intent field addition in Klarna Payment session creation Combine Klarna Authorization and Create Order in Checkout Review Step
22.5.0	January 2023	OMS support
23.1.0	July 2023	Fix issue with incorrect values for EMD Improvement Klarna Auto Capture and error handling Logging information for troubleshooting bugs Add Auto_finalise=True to the review checkout flow
23.1.1	September 2023	Fixed an issue where sessions with negative order_tax_amount occurred due to SFCC session expiration Compatibility mode 21.2 support Replace deprecated window.KlarnaOnsiteService.push with window.Klarna.OnsiteMessaging.refresh.

<b>Version</b>	<b>Date</b>	<b>Changes</b>
23.2.0	December 2023	<p>Subscription Payments support: recurring payments and subscription handling directly within the SFCC environment. This update includes configuration options, subscription management in the cart and checkout pages, and a customer dashboard for subscription oversight.</p> <p>Klarna Bank Transfer payments: added a new server-side authorization callback feature for Klarna Bank Transfer payments, enhancing reliability across EU markets and supporting all existing KP cartridge functionalities.</p>
24.1.0	January 2024	<p>Klarna Express checkout: a new feature introduced in Storefront where Express Checkout Button will be displayed in PDP, Cart and Mini Cart and the user will be redirected to Klarna after clicking the Express Checkout button. Klarna Express checkout enabled quick and easy checkout where Shipping Address, Billing Address and Payment details will be preselected so that Checkout can be completed in fewer clicks.</p> <p>Fix for creating order service calls with 500 status responses: now orders are not created and error is thrown for these cases.</p>
24.2.0	March 2024	<p>Update of OSM functionality that is more aligned with current web standards, provides a consistent identifier system as used in KEC, and offers enhanced customization options to our merchants. This will not only improve the user experience but also reinforce Klarna's commitment to providing versatile and state-of-the-art e-commerce solutions. CSS customizations are available only in the new library version.</p> <p>Documentation update - new section added for cartridge upgrade process.</p>
24.3.0	May 2024	Update of Klarna Express checkout configuration. The display of Klarna Express checkout buttons is now customizable by selecting preferred placements. By default, none is selected.

<b>Version</b>	<b>Date</b>	<b>Changes</b>
24.4.0	June 2024	<p>Documentation update – new section added “Klarna Display Conditions and Authorization Handling”.</p> <p><b>New features</b></p> <ul style="list-style-type: none"> <li>• RO and CZ countries support</li> </ul> <p><b>Configuration Structure Updates</b></p> <ul style="list-style-type: none"> <li>• New Settings Points <ul style="list-style-type: none"> <li>◦ Klarna Activation Custom Object</li> <li>◦ Site Preferences: <ul style="list-style-type: none"> <li>■ Klarna Activation (Klarna_Activation)</li> <li>■ Klarna Payments (Klarna_KP)</li> <li>■ Klarna Express Checkout (Klarna_KEC)</li> <li>■ Klarna On-site Messaging (Klarna_OSM)</li> </ul> </li> </ul> </li> </ul> <p><b>Deprecations</b></p> <ul style="list-style-type: none"> <li>• Klarna Countries Custom Object</li> <li>• Site Preferences: <ul style="list-style-type: none"> <li>◦ Klarna Payments (Klarna_Payments)</li> <li>◦ Klarna Recurring Payments (Klarna_RecurringPayments)</li> <li>◦ Klarna Express Checkout (Klarna_ExpressCheckout)</li> </ul> </li> <li>• Removed Site Preferences Attributes: <ul style="list-style-type: none"> <li>◦ kpServiceName</li> <li>◦ kpBankTransferCallback</li> <li>◦ kpRejectedMethodDisplay</li> <li>◦ kpNotAvailableMessage</li> </ul> </li> </ul>

Version	Date	Changes
		<ul style="list-style-type: none"><li>○ vcnPvateKey</li><li>○ vcnPublicKey</li><li>○ KpRateLimitByOperation</li><li>○ kpCreateNewSessionWhenExpires</li><li>○ sendProductAndImageURLs</li></ul> <p><b>Note:</b> Site preferences and service credentials for the deprecated items are now obsolete and must be migrated to the new site preferences or activation custom object.</p>

## 8.1. Known issues

There are no known issues in version 24.4.0.

# 9. Additional Information

## 9.1. Klarna API Information

The Klarna Payments API is accessible through different endpoints based on the context of the webstore. There are separate endpoints for testing and live and the Klarna merchant identifier (MID) is configured for respective markets in regions (EU, NA, OC) by endpoint.

### 9.1.1. Live Environment

To access the Klarna API in the production environment, use the following URLs based on your region:

- **Europe:** <https://api.klarna.com/>
- **North America:** <https://api-na.klarna.com/>
- **Oceania:** <https://api-oc.klarna.com/>

### 9.1.2. Testing Environment

To access the Klarna API in the testing environment, use the following URLs based on your region:

- **Europe:** <https://api.playground.klarna.com/>
- **North America:** <https://api-na.playground.klarna.com/>
- **Oceania:** <https://api-oc.playground.klarna.com/>

## 9.2. Generate Key Pair and Key ID for Virtual Card Settlements (VCN)

To generate an RSA key pair with a 4096-bit private key, follow these steps:

**Generate RSA Key Pair:** Use the following OpenSSL command to generate a 4096-bit private key:

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

**Extract Public Key:** Use the following OpenSSL command to extract the public key from the RSA key pair:

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

**File Output:** These commands will create two files in the folder where you executed the commands:

- `public_key.pem`
- `private_key.pem`

**File Contents:** The contents of the files should look something like this:

#### **Public\_key.pem:**

```
-----BEGIN PUBLIC KEY-----  
MIICIJANBhkjhkiG9wOBAQEFAOCAg8AMIIICCgKCAGEAoNYG712G8nZa+22oBYZk  
tV2281w3UE9W04oxfkNjtKEdHn84x55ULt8KQTh9NVtdeKC8nTfTgyvMt/GNCa18  
xuZV/lGYDftKt85hbV5Ej0um+StAIufEXv1BX7nMOMc1KyWm9kp2kbqd88mFIX63  
KV940oNExcNatRDFYR+qz53+ifadDQtQ1s1VNstdroCZDJ1+LxtBy9V+BdmsBK1E  
RLsKh/JLxyWE24FJKV+z0Os7TQkdWW/5ET120GQYZsWo1yqgi9HplNvrise8vWP  
xaL4m8iz3I/9yYdg7yANQbTxSJccbRCgaaagPo30CNxeqU6qafY5g8vY3E52CoXH  
Dd04Us1X1qcuYIDhqaDzey6W+b8m755xLi+rqQyM4PBWL0J0dM3FVid8+4YKILex  
3AKBFciqRCMHSOGaEeyrXKTj1Asghr9RS8PifvQRrL440chzqw2vX0DvpjSWcmUJ  
tW4wUq5RNSSobrxnVmoV6fj1z67Q/1P+15Ie+oowdahR5ztVqJ10+2PNoX4I5VDs  
/Pkz3f8wWVc3Mp2oNT244o+/NIiyRfPFaJJx7JAgrcvZt2nFAmY4QApXLFJCpgEM  
wYucE4AH4gJKsh3KZbxRERrr072bL2rxvWqBp/0h7DcMsV9sQs4BvxxI16CF506F  
ThzmclaKLBAyd5LALiXiPfkCAwEAAQ==  
-----END PUBLIC KEY-----
```

#### **Private\_key.pem:**

```
-----BEGIN PRIVATE KEY-----  
MIIJQQIBADANBhkjhkiG9wOBAQEFAASCCSswggknAgEAAoICAQCg1gbuXYbydlr7
```

bagFhmS1XbbYXDdQT1Y7ijF+ScmOoR0efzjHn1Qu3wpBOHO1W114oLydN90DK8y3  
8YOJrXzG51X+UZgN+0q3zmFtXkSM66b5KOAi58Re+UFfucw4xzUrJab2SnaRup3z  
yUhfrpX3g6g0Rdw1q1EMVhH6rPnf6J9pONC1DWyVU1K12ugJkMnX4vGOHL1X4F  
2awErUREuwqH8ktfJYTbgUkpX7PTSztNCR1Zb/kRPXY4ZBhmxajXKqCL0emU2+uK  
y97y9Y/FovibyJncj/3Jh2DvIA1BtPFI1xttEKBppqA+jfQI3F6pTqpp9jmDy9jc  
TnYKhccN07hSyVfWpy5gg0GpoPN7Lpb5vybvnnEuL6upDIzg8FYvQnR0zcVWJ3z7  
hgogt7HcAoEVyKpEIwdI4ZoR7Ktcp00UCyCGv1FLw+J+9BGsvjjRwfOrDa9fQ0+m  
NJZyZQm1bjBSrlE1KyhuvGdWahXp+PXPrtd/U/6Xkh76ijB1qFHn01WomU77Y82h  
fgj1U0z8+TPd/zBZVzcynag1Pbjij780i1JF88VoknHskCCty9m3acUCZjhAC1cs  
UkKmAQzBi5wTgAfikqyHcp1vFERGus7vZsvavG9aoGn/SHsNwyxX2xCzgG/HEiX  
oIXnToVOHOZyVooEDJ3ksAuJeI9+QIDAQABAoICACRkaUsUNI22RB3yEPu3DiCP  
p06v+QAEA4gTW+GUdqR9dCZLaSCZ7bhxVV0uoX4qPzs106hjUmOyzG6upFgVPk+P  
HNQfyEUZoC148Eib90ziAXUN2URMp1KbwVm+B0814X8zguai7uru0PHTG1oy677  
4Ct10knxAxxHQDIaxT6XJFo5SA4EinUfnz2Bo3/xry/QjxW/mCKOGwDd4PNp9TGM  
FPtv2SgdSDOWzGQ10H5N3owuzMpI8NV6z74wv+i5Ptv41Dzu8WhyXpiYSsk00SRK  
HPC68j2bAzTPghp5aSZ9976SGm2SPonJXyboXdiHbI/osdyqDxeIT3iB9GmrHX/i  
kHPGJCh7fRZvqj39Hc+IxYjabwW3rDeDIPB7ab9z1KLF4z1D6AZOKCPyTaDRdQ1Q  
eDi7LwDmk7NHEPrmF/nIcgudqbIbmF02zEsOT0e6y4uBMndRsbQprTNSMuDbkrA  
1NaYVSTQ1ZOY/8DZDpGcyS10nJv74F15uDjKN6/ov991mZ1JrZ+V2sdS3EDU1mvP  
6thQKwI7Ln6h+ApHtWUG1NmvmQe5gJE0qAej9b45c1UzIRUwhVmEp8NoIJh0kAjaN  
d41k7xy9ZRDUY5yekPeYrJPShjsHAyEoktJIjRufI2UUq3uxNjjICoQcOVGfNDIS  
YTPpu1pmCOC+rh2fgBAoIBAQDRultRArvtc2JKhVOUyZk88zd9kvrI6fNiyKmi  
HgiWf7qkTPD9xh0QWDw3iwRFQAD+YkgV5MCBO8wp8o08GEsOCI+XZWEEx0cPT0Vfj  
PZHiQrTFn1fG/+fAO14xLf3j3ED4YQXdH0KI3xoLknQx/EydLoctxgkkpgWLrsA7  
DwdSAg1/OsBvaHY27ogAfdimHdaKZ50Ae4a9k1qP3xVZBu0e8Sd65unBavUJLDuv  
ikeNmksVgW1sm55/729JIr63USHF76It+vE1cdZ+vKg5vYotsQgPzvNBmU0/E8Gj  
zMXQRfqfvED1NXEXOrCupTkw1G6AGTwQc/NPzyr/LTpLe6UBAoIBAQDEUjtG11V  
hf7WjdG3gctR1r+mYapQHgXdVLx2QSaqUYid+OQXK11YfJlsRB6nwa+OED83RfPO  
1IFqxpzudSLPmoDuIBT7D15c/aleyKs/siuUsP8QVDXk60AR84XSytC35sIRV7pE  
VMuBL91jfkQOLf/Pres1K/kI6Yvwwp4qrHK6/f9TgciHclYtf+/oti4ky6GJgfmp  
fmuCqjxmUKbXXFPd5RbL2THG0owlb8zDLjf3R1bj1QFqogAk6H9hp2VOVZLiJHp  
UWM3z3xDWeDaqJ08sHuk/rA9QpsVTu8IGTQsxdj8JwluN1Q+YZiOuPiSENbqPzT  
V3exexzo3sD5AoIBAGU3qEyPojz1+9D1SaI8LW2CABzlq4z9g84ABAZOs1xX5q7W  
x1PinZyDSQRXg1B13jt29ZdIR79ygnQ1g1YOBjcvtgVQHPuafk3R1BQbbCh+vaI  
9dn/tUxMGqhnunKaby1rovJHfdqnPpkwzNAjYUqaGkJ822xhmmke/fEyAanIPa4  
stDRvIPEWPTLx5xc0Cdx13khkPSnkgRvaLEfpwkVX7Vr7hK/20SFaYTNmrzXYBQ7  
c6D/9d30o4nLb/mu+Tq67S19t53Qg/GEgTfkpuRoVPI0KyhUnKKCGW1BMZLTwyIG  
s9eTFDKoJ0cSTGipjW7bPua93wZ8eEbRABpf4QECggEANNhQBeEJ0aCdBVHtdrEI  
crDaa8X0W1aJi5dol4hYCRajaKsfHAF/QfdgMQVxHwUC5YG4En/Q+DAVWhGWYpXD  
RhC3zeFy5FVszyk0sx/fA01KGvRn5BRW4YRR9GMRzbjsT+RcrubnckdE9ERXGpX9  
c/JB3rxZB1t+oIiFM8yfWKtMwsrmNKtFuDftvJeok4KejycFF4eWDqs f828xjPT+  
xA/FP4CQD1UqkcpmuFSIwAwXo6LXVY7NTSOmKMiUnTLkL1TIHtLn09+9jmNapWRP  
Tc+hZUuHK1pI8DHFmX2j87LgkFD05eD51ynY4RgZtU1W1C1RdVYwoA72WB7knEaB  
uQKCAQAH9s67P/7fFX9dfEans3PHU4nGjD8dJ8eoNQ6DhBMydZpGWI5ZUeEBZDRk

```
0cB0eRs5B0cS43Em9kETpzawyCwxmnwz1+CzoPzMqcTw9tXomF9HG6RJ9XBdJfGA
ALAwCd4bASxmFM6guSP5GKnZ9aY3tR3tWWdfr7f9z8w0ewzzpPclwRh009fPe4TC
NXoEm1MELJVieUieDSLKZgjgCw8WHGqLItONpAO/fwSM2gIcxETVv7qx3aPuJzCVh
LQZoBLQk3UMKsWDdpzeBdiERe66NAgVk92Xe7SY9EY2vymaq761i1x1vlprT27qp
240LDJawqMOIrakmdCvWjofWSaOU
-----END PRIVATE KEY-----
```

**Key Pair Association:** This key pair must be associated with a `key_id` (UUIDv4). The public key must be shared in JWK format with your Klarna contact. Note that for both production and playground environments, different `key_id` and key pair combinations must be configured before testing or going live with the virtual card product.

### 9.3. Decrypt VCN Card Details

To decrypt the virtual card details stored at the order level and authorize the credit card processor, you can use the following code snippet. This script utilizes the Salesforce Commerce Cloud (SFCC) APIs and classes for cryptographic operations.

```
var OrderMgr = require( 'dw/order/OrderMgr' );
var Cipher = require( 'dw/crypto/Cipher' );
var Encoding = require( 'dw/crypto/Encoding' );
var Site = require( 'dw/system/Site' );

var Order = OrderMgr.getOrder( "order_id" );
var VCNPrivateKey = Site.getCurrent().getCustomPreferenceValue( 'vcnPrivateKey' );
var cipher = new Cipher();

var keyEncryptedBase64 = Order.custom.kpVCNAESKey;
var keyEncryptedBytes = Encoding.fromBase64( keyEncryptedBase64 );
var keyDecrypted = cipher.decryptBytes( keyEncryptedBytes, VCNPrivateKey,
"RSA/ECB/PKCS1PADDING", null, 0 );
var keyDecryptedBase64 = Encoding.toBase64( keyDecrypted );
var cardDataEncryptedBase64 = Order.custom.kpVCNPCIData;
var cardDataEncryptedBytes = Encoding.fromBase64( cardDataEncryptedBase64 );
var cardDecrypted = cipher.decryptBytes( cardDataEncryptedBytes,
keyDecryptedBase64, "AES/CTR/NoPadding", Order.custom.kpVCNIV, 0 ); 102 Klarna
Payments for SFRA v24.2.0
var cardDecryptedUtf8 = decodeURIComponent( cardDecrypted );
```

```

var cardObj = JSON.parse( cardDecryptedUtf8 );
var expiryDateArr = cardObj.expiry_date.split( "/" );

// Retrieve encrypted card details
var cardPAN = cardObj.pan, cardCVV = cardObj.cvv,
cardExpiryMonth = expiryDateArr[0], cardExpiryYear = expiryDateArr[1];

```

**Steps:**

- 1) **Retrieve the Order:** Use `OrderMgr.getOrder("order_id")` to fetch the order by its ID.
- 2) **Store the VCN private key in SFCC:** Create custom attribute to store the private key:
  - a) **Site Preference for single site credentials**
    - i) Go to **Administration > System Object Types**.
    - ii) Select **Site Preferences** from the list of system object types.
    - iii) Click on **Attribute Definitions**.
    - iv) Click on **New** to create a new custom attribute.
    - v) Define the Attribute:
      - (1) **ID:** Enter '**vcnPrivateKey**' (this is the unique identifier for the attribute).
      - (2) **Display Name:** Enter '**VCN Private Key**'.
      - (3) **Description:** Enter '**SSL private key used only to decode Virtual Card information (used with kpVCNEncoded)**'.
      - (4) **Type:** Select '**Text**' from the dropdown.
      - (5) **Mandatory:** Ensure the mandatory flag is set to '**false**'.
    - vi) Save the attribute by click on **Apply**
    - vii) Click on **Attribute Grouping**
    - viii) Select **Klarna\_KP** group
    - ix) Add the new attribute '**vcnPrivateKey**' and save
    - x) Open **Merchant Tools > Site Preferences > Custom Site Preference Groups > Klarna Payments** and update the **vcnPrivateKey** value
  - b) **Custom Object attribute for multiple Klarna credentials per site**
    - i) in Business Manager, go to **Administration > Site Development > Custom Object Types**.
    - ii) Locate the custom object type named **KlarnaActivation** and click to edit

- iii) Navigate to the **Attributes** tab.
  - iv) Click on **New** to create a new attribute.
  - v) Fill in the following details:
    - (1) **Attribute ID:** `vcnPrivateKey`
    - (2) **Display Name:** VCN Private Key
    - (3) **Description:** SSL private key used only to decode Virtual Card information (used with `kpVCNEnabled`).
    - (4) **Value Type:** Select **Text**
    - (5) **Mandatory:** Ensure this is **unchecked**, as *mandatory-flag* is *false*.
  - vi) Still within the Custom Object **KlarnaActivation**, navigate to **Attribute Groups** tab.
  - vii) Locate and click on Edit of '**vcn**' attributes group
  - viii) Find the attribute **vcnPrivateKey** and add it to the group
  - ix) Go to **Merchant Tools > Custom Objects > Manage Custom Objects**
  - x) Select **KlarnaActivation** ObjectType and search
  - xi) Enter the **vcnPrivateKey** for each entry
- 3) **Get the VCN Private Key:** Retrieve the VCN private key from the custom site preferences using:
- a) From Site Preferences (single credentials)  
`Site.getCurrent().getCustomPreferenceValue('vcnPrivateKey')`
  - b) From Custom Object - please take a look at `KlarnaHelper.getVCNKeyId()` and create similar function to get the private key using its id **vcnPrivateKey**.
- 4) **Decrypt the AES Key:**
- a) Convert the encrypted AES key from Base64 to bytes using  
`Encoding.fromBase64`.
  - b) Decrypt the AES key using `cipher.decryptBytes` with the RSA algorithm.
- 5) **Decrypt the Card Data:**
- a) Convert the encrypted card data from Base64 to bytes using  
`Encoding.fromBase64`.
  - b) Decrypt the card data using `cipher.decryptBytes` with the AES algorithm.
- 6) **Parse and Extract Card Details:**
- a) Decode the decrypted card data from URL encoding to UTF-8 format.
  - b) Parse the JSON string to get the card details object.
  - c) Extract the card PAN, CVV, and expiry date.
- 7) **Authorize the Credit Card Processor:**

- a) Use the decrypted card details to proceed with the credit card authorization process.



## Important!

- Ensure that sensitive data such as card details are handled securely and comply with PCI DSS standards.
- Remove or secure any logging of sensitive information in a production environment.
- For more information about the decryption process, refer to the [Klarna documentation](#).

## 9.4. Update KlarnaCountries Definition



### This functionality is deprecated as of release 24.4.0

This object is deprecated from version 24.4.0, and replaced with the [Klarna Activation](#) object.

With version 21.2.0 of the cartridge, the `KlarnaCountries` custom object definition changed to non-replicable. If you are using an earlier version, to mitigate any issues that may be present in your environments, please follow these steps to update the definitions:

- 1) Back-up `KlarnaCountries` configurations per country for each site as changing the definition will remove all configurations on your environment.
  - a) **For merchants with one site** - Go to "**Merchant Tools > Custom Objects > Import & Export**", select `KlarnaCountries` & export the data.
  - b) **For merchants with multiple sites using Klarna** – You can export the configurations via "**Admin > Site Development > Site Import & Export**" and select each site that uses Klarna to be included in the export zip file. This

section requires Account Manager access for users and will export all custom objects, not just the Klarna ones.

- 2) Export custom object definitions from "**Admin > Site Development > Import & Export**". This action will export all custom object definitions that you have.
- 3) Update `KlarnaCountries.xml` definitions in file exported in step 2 and set "`<staging-mode>source-to-target</staging-mode>`" to "`<staging-mode>no-staging</staging-mode>`"
- 4) Import the updated `KlarnaCountries.xml` definition in "**Admin > Site Development > Import & Export**".
- 5) Import the `KlarnaCountries` configs exported in step 1 (manually or via bulk import)
  - a) If you've followed step **1.b** with multiple sites, you may want to edit the zip file and remove everything else apart from `KlarnaCountries.xml`