****

*SFRA LINK Cartridge Integration guide*

*Version 23.1.0 – SFRA*



Table of Contents

[**1. Summary**](#_gjdgxs) **3**

[**2. Component Overview**](#_30j0zll) **4**

[*Functional Overview*](#_1fob9te) *4*

[*Limitations, Constraints*](#_3znysh7) *4*

[Contract](#_yguvnvqwbhf5) 4

[*Compatibility*](#_2et92p0) *4*

[*Privacy, Payment*](#_tyjcwt) *4*

[**3. Implementation Guide**](#_3dy6vkm) **5**

[*Setup of Business Manager*](#_1t3h5sf) *5*

[Importing cartridges](#_4d34og8) 5

[Assigning cartridges to the site](#_2s8eyo1) 7

[Import of metadata](#_17dp8vu) 8

[Service](#_35nkun2) 8

[*Custom Code*](#_1ksv4uv) *8*

[Perform changes in scripts](#_gj02rffv5pcc) 8

[Perform changes in templates](#_ps7qc6jo8oei) 9

[Perform changes in client script](#_hjccpohuzf96) 22

[Configuration](#_qm9di0ith0b8) 31

[Customer Groups](#_2y8scq8zd9wc) 31

[Custom Site Preferences](#_26in1rg) 31

[*Firewall Requirements*](#_z337ya) *33*

[**4. Testing**](#_3j2qqm3) **33**

[**5. Operations, Maintenance**](#_1y810tw) **33**

[*Data Storage*](#_4i7ojhp) *33*

[*Availability*](#_2xcytpi) *33*

[*Failover/Recovery Process*](#_1ci93xb) *33*

[*Support*](#_3whwml4) *33*

[**6. Known Issues**](#_2p2csry) **33**

[**7. Release History**](#_147n2zr) **33**

# Summary

The Catch LINK cartridge allows you to quickly integrate a Catch instance into a Salesforce Commerce Cloud website. The Catch service allows the customer to pay for purchases in the online store and saves money on purchases.

This LINK cartridge contains instructions that allow you to integrate with a reference Catch application.

**Glossary**

| Term | Description |
| --- | --- |
| SFCC | Salesforce Commerce Cloud |
| Business Manager (BM) | The primary tool used to configure the SFCC platform and customer storefront  SFRA |
| SFRA | StoreFront Reference Architecture |

# Component Overview

## Functional Overview

The Catch cartridge enables customers to select the Catch payment method on checkout and pay with the Catch platform. Service supports the following payment methods:

* Catch payment processing

## Limitations, Constraints

Use of the Catch cartridge requires credentials and API keys from Catch. Current implementation of the Catch payment instrument supports only US and USD in terms of countries/currencies.

### Contract

To work with the cartridge, you need to contact the Catch support service to get your personal credentials to gain access to the Catch services.

## Compatibility

The cartridge is designed for Salesforce Commerce Cloud API version 23.9 (Compatibility Mode: 22.7) with

SFRA version 6.3.0. The cartridge is designed with US locale but multi-locale and US currency support.

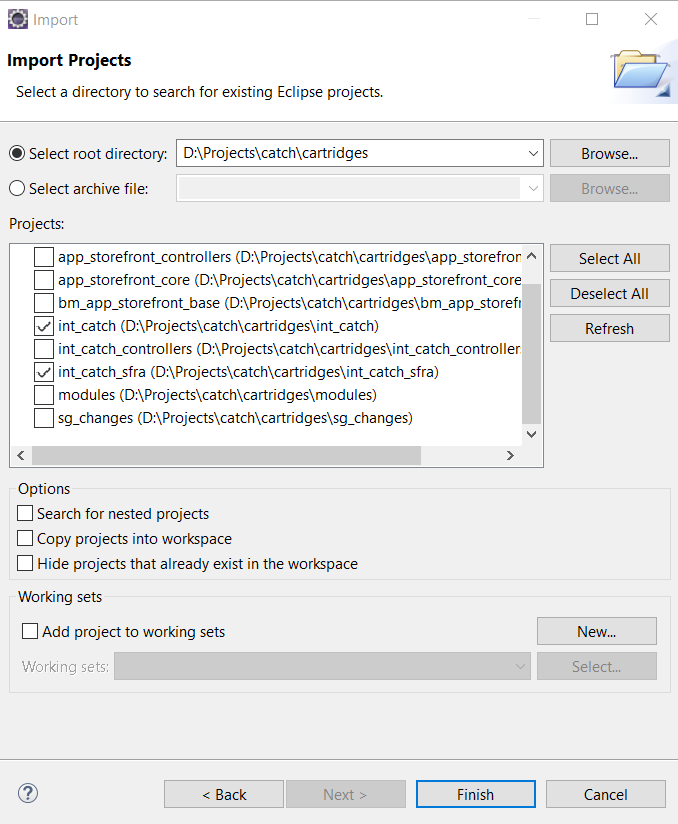
## Privacy, Payment

The Catch cartridge collects and processes user profile information or billing information. For additional privacy information, please contact your Catch Account Manager.

# Implementation Guide

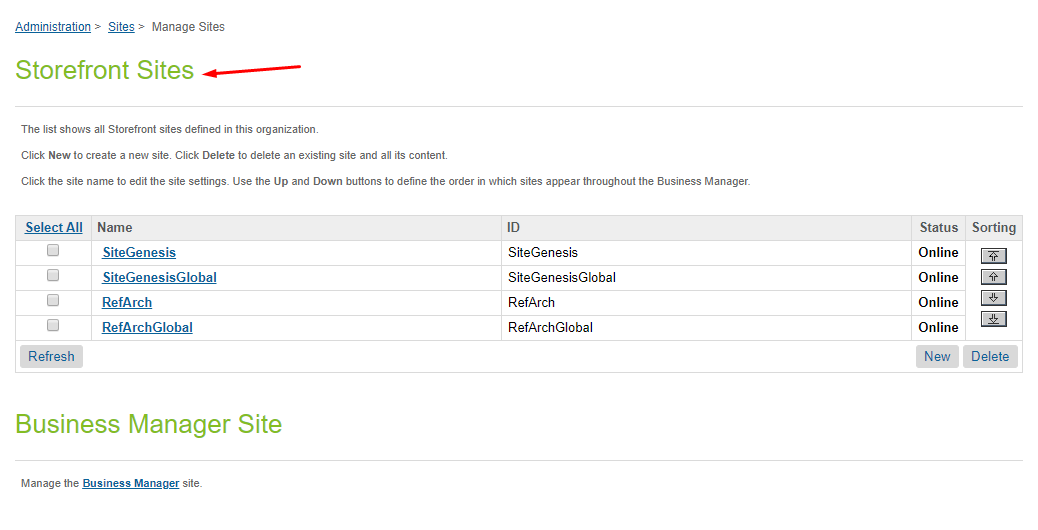
## Setup of Business Manager

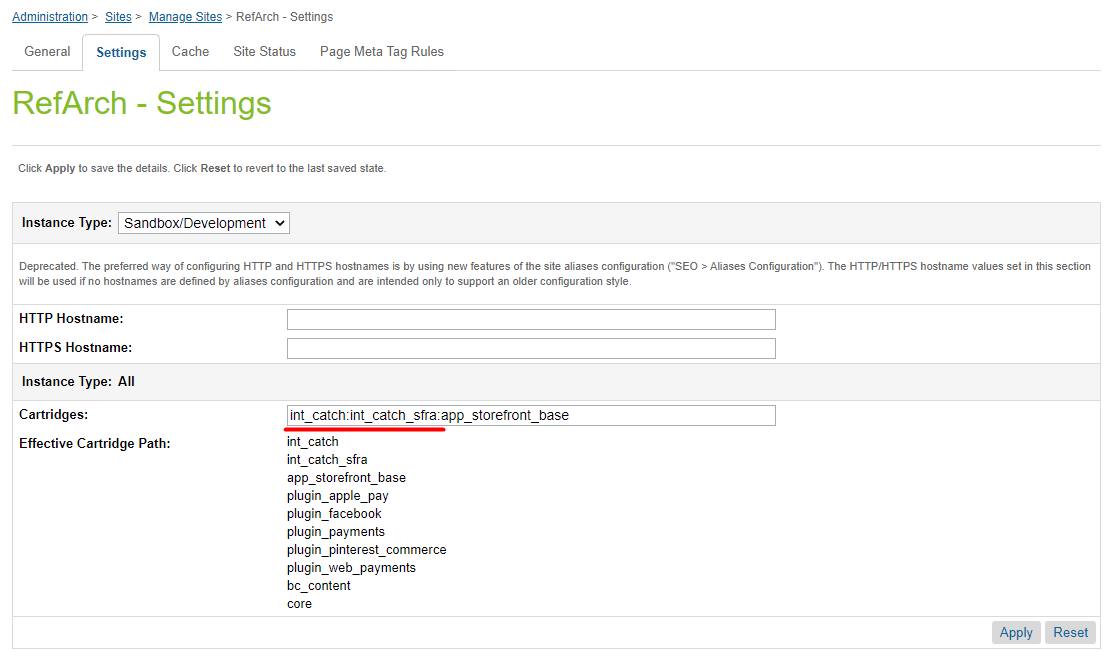
### Importing cartridges

1. Download the cartridge source code.
2. Establish a new digital server connection with your SFCC Instance
3. Import cartridges to a workspace in Salesforce UX Studio.  
   
4. Add cartridges to Project Reference of Server Connection.  
   
5. Wait until Studio completes workspace build and uploads source codes to the sandbox.

### Assigning cartridges to the site

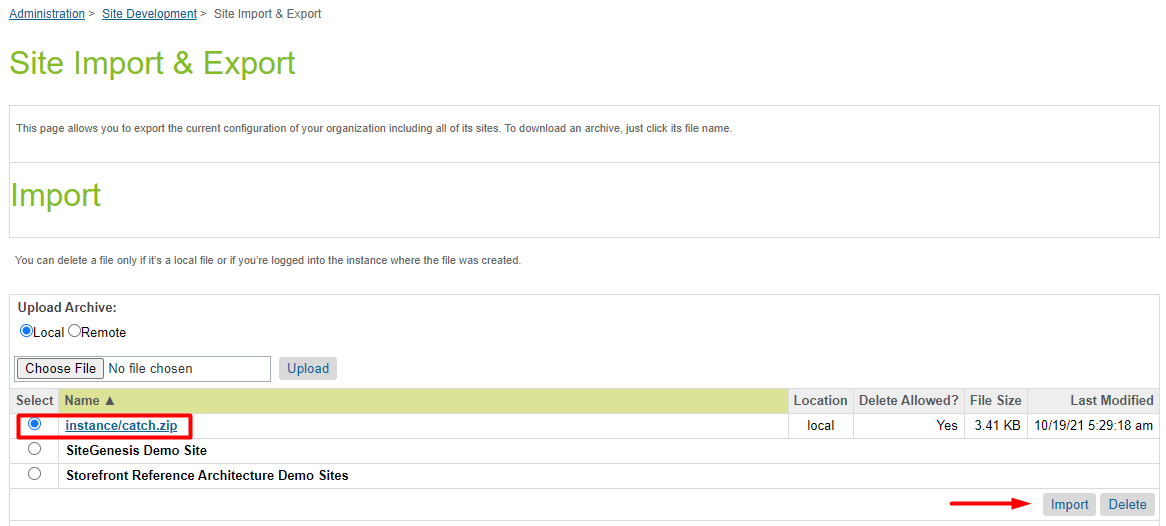
Go to Administration > Sites > Manage Sites > Select your site from the list “Storefront Sites”.

  
  
Select Settings tab. In the beginning of the cartridge path add the following:  
int\_catch:int\_catch\_sfra:.



### Import of metadata

Go to Administration > Site Development > Site Import & Export. Upload *catch.zip* from *metadata* folder.



Select catch.zip and finish the import process.

### Service

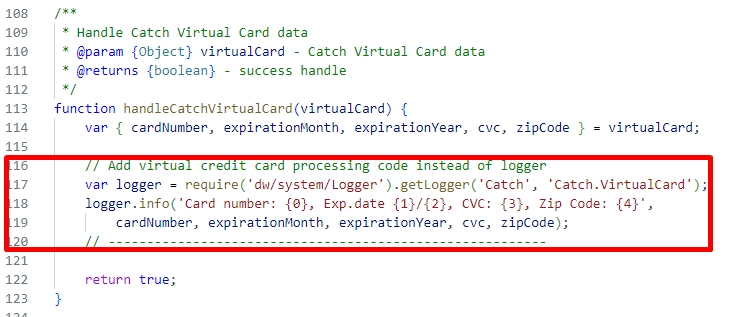
The service does not require additional settings.

## Custom Code

### Perform changes to processing virtual card data

**Note: the cartridge does not include virtual card payment processing.**

When using a Virtual Card integration type, after the completion of the Catch checkout, the cartridge gets the data of the virtual card from Catch for processing. The cartridge does not include virtual card payment processing because different payment services may be used for each merchant. When integrating the cartridge, developers have to add the virtual card processing code to the function **handleCatchVirtualCard** in the int\_catch/cartridge/scripts/catch/helper/catchCheckoutHelper.js module.



### Perform changes in templates

**Note: all changes are already made in the int\_catch\_sfra cartridge.**

1. app\_storefront\_base/cartridge/templates/default/cart/cart.isml

Add Catch styles and script on the page in lines 4 and 6 :

assets.addJs('/js/catch.js');

assets.addCss('/css/catch.css');



Put the following rows in line 81:

<iscomment> Catch code </iscomment>

<div class="row">

<div class="col d-flex justify-content-center callout-mini-cart-wrapper">

<isinclude url="${URLUtils.url('Catch-IncludeWidget', 'widgetName', 'callout', 'page', 'cart')}" />

</div>

</div>

<iscomment> Catch code </iscomment>



1. app\_storefront\_base/cartridge/templates/default/checkout/checkout.isml

Add Catch styles on the page in line 10:

assets.addCss('/css/catch.css');

Put the following rows in line 13:

<isset name="catchHelper" value="${require('\*/cartridge/scripts/catch/helper/catchHelper')}" scope="page" />

<isset name="catchPublicKey" value="${catchHelper.getPreference('publicKey')}" scope="page" />

Extend <div> tag with id="checkout-main" in line 23:

<div

id="checkout-main"

class="catch-data container data-checkout-stage <isif condition="${pdict.order.usingMultiShipping && pdict.order.shipping.length > 1}">multi-ship</isif>"

data-customer-type="${pdict.customer.registeredUser ? 'registered' : 'guest'}"

data-checkout-stage="${pdict.currentStage}"

data-checkout-get-url=${URLUtils.https('CheckoutServices-Get')}

data-key="${catchPublicKey}"/>

>



Extend <button> in line 117:

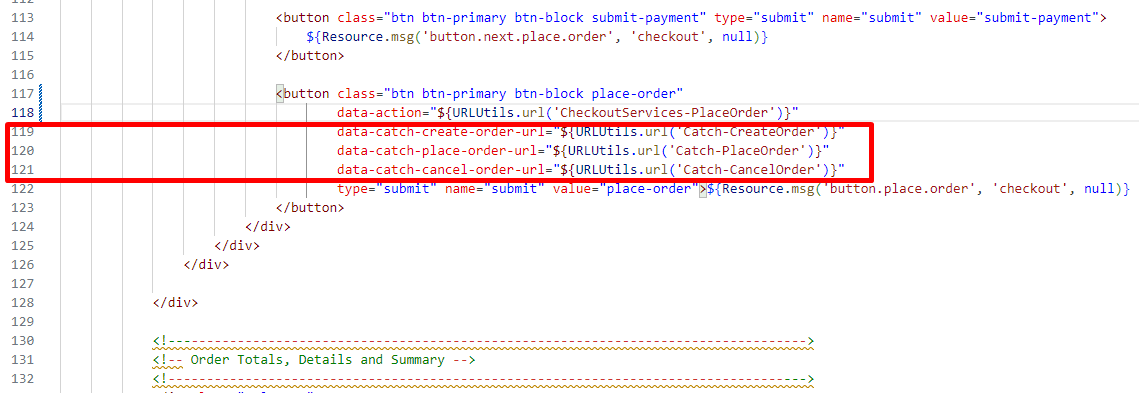
<button class="btn btn-primary btn-block place-order"

data-action="${URLUtils.url('CheckoutServices-PlaceOrder')}"

data-catch-create-order-url="${URLUtils.url('Catch-CreateOrder')}"

data-catch-place-order-url="${URLUtils.url('Catch-PlaceOrder')}"

data-catch-cancel-order-url="${URLUtils.url('Catch-CancelOrder')}"



1. app\_storefront\_base/cartridge/templates/default/checkout/orderTotalSummary.isml

Add the following line to the end of the file:

<isif condition="${pdict.order.totals.appliedRewards}">

<div class="col-6 start-lines">

<p class="order-receipt-label"><span>${Resource.msg('label.order.rewards','catch', null)}</span></p>

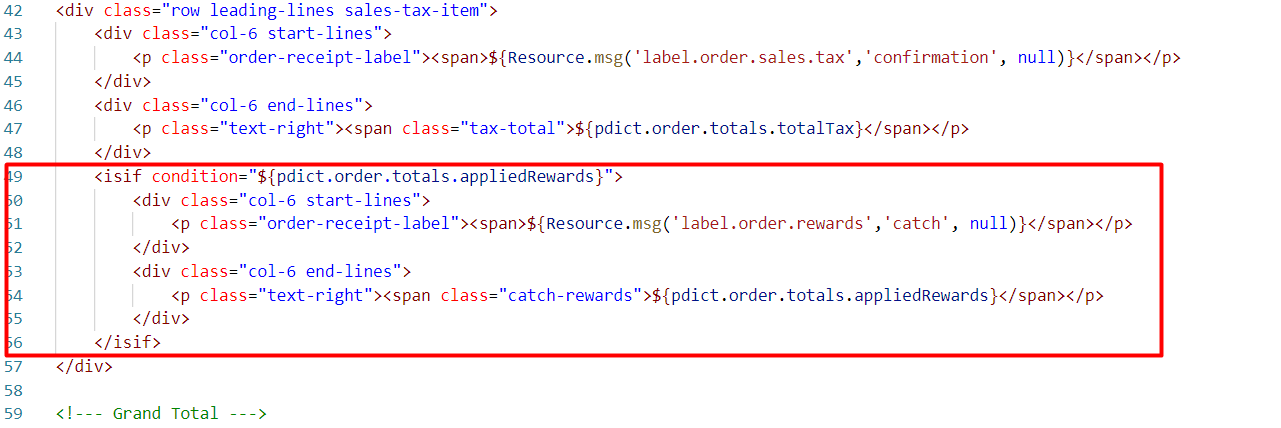
</div>

<div class="col-6 end-lines">

<p class="text-right"><span class="catch-rewards">${pdict.order.totals.appliedRewards}</span></p>

</div>

</isif>



Add the following line to the end of the file:

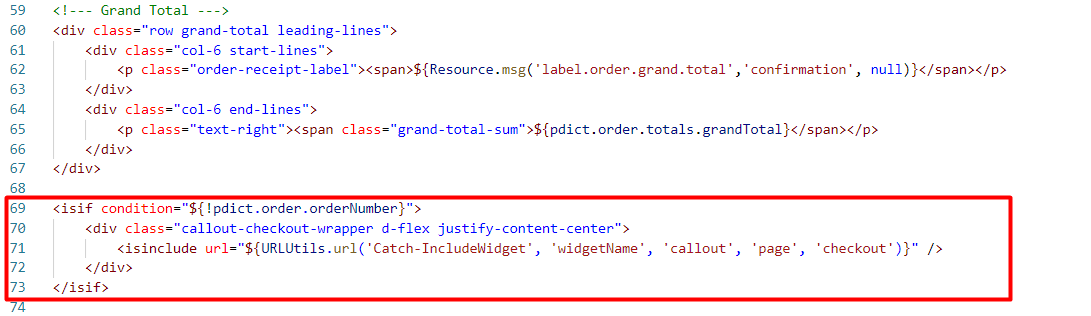
<isif condition="${!pdict.order.orderNumber}">

<div class="callout-checkout-wrapper d-flex justify-content-center">

<isinclude url="${URLUtils.url('Catch-IncludeWidget', 'widgetName', 'callout', 'page', 'checkout')}" />

</div>

</isif>



1. app\_storefront\_base/cartridge/templates/default/checkout/billing/paymentOptions/paymentOptionsContent.isml

Add the line to the beginning of the file

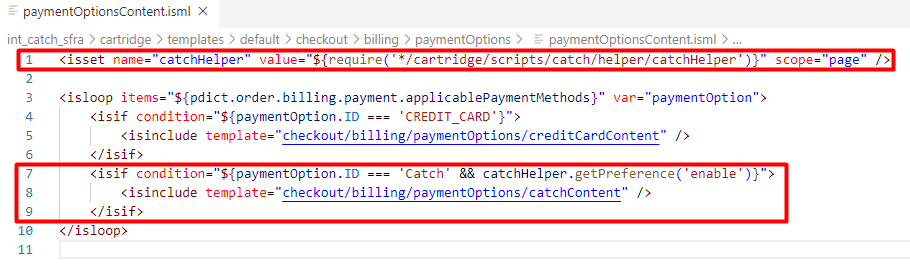
<isset name="catchHelper" value="${require('\*/cartridge/scripts/catch/helper/catchHelper')}" scope="page" />

Put the following rows in line 7:

<isif condition="${paymentOption.ID === 'Catch' && catchHelper.getPreference('enable')}">

<isinclude template="checkout/billing/paymentOptions/catchContent" />

</isif>



1. app\_storefront\_base/cartridge/templates/default/checkout/billing/paymentOptions/paymentOptionsSummary.isml

Put the following rows in line 6:

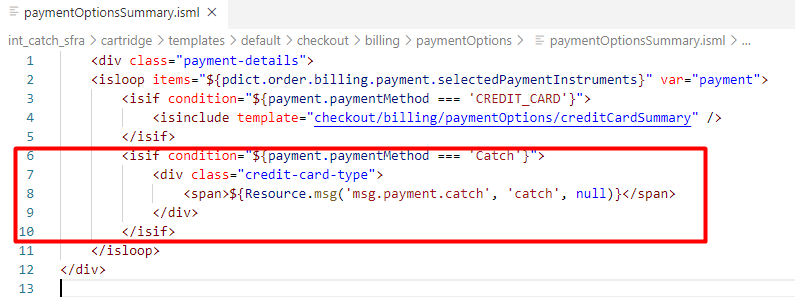
<isif condition="${payment.paymentMethod === 'Catch'}">

<div class="credit-card-type">

<span>${Resource.msg('msg.payment.catch', 'catch', null)}</span>

</div>

</isif>



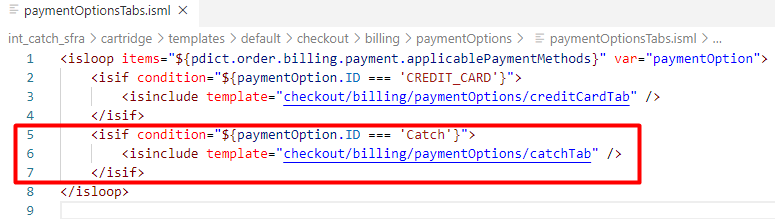
1. app\_storefront\_base/cartridge/templates/default/checkout/billing/paymentOptions/paymentOptionsTabs.isml

Put the following rows in line 5:

<isif condition="${paymentOption.ID === 'Catch'}">

<isinclude template="checkout/billing/paymentOptions/catchTab" />

</isif>



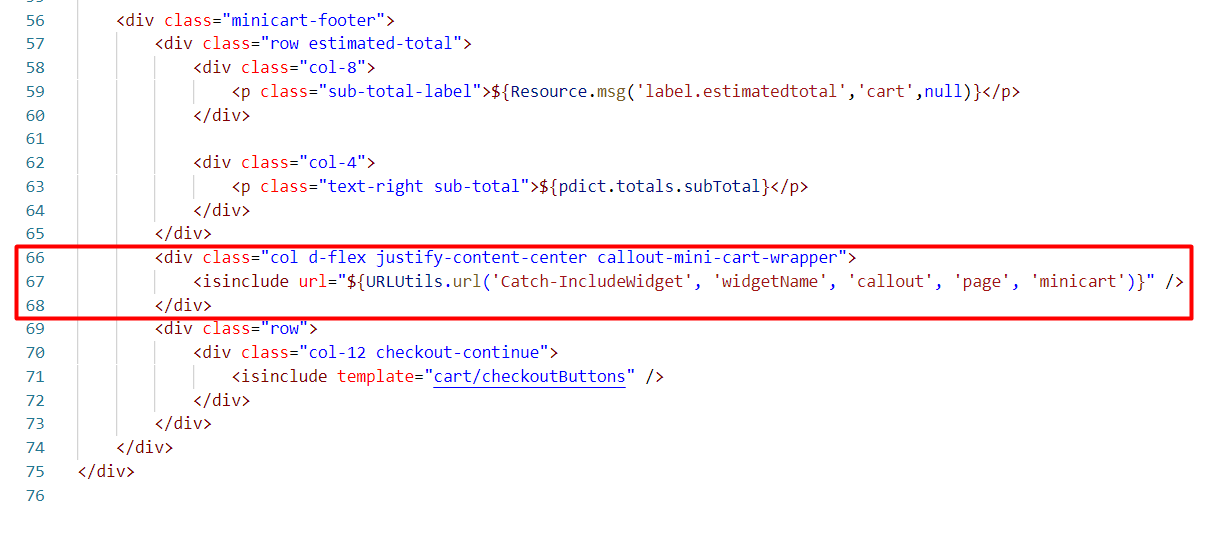
1. app\_storefront\_base/cartridge/templates/default/checkout/cart/miniCart.isml

Put the following rows in line 66:

<div class="col d-flex justify-content-center callout-mini-cart-wrapper">

<isinclude url="${URLUtils.url('Catch-IncludeWidget', 'widgetName', 'callout', 'page', 'minicart')}" />

</div>



1. app\_storefront\_base/cartridge/templates/default/checkout/confirmation/confirmation.isml

Add Catch scripts and styles on the page in the beginning of the file:

assets.addCss('/css/catch.css');

assets.addJs('/js/catchInitOnConfirmation.js');

Put the following rows in line 34:

<isloop items="${pdict.order.billing.payment.selectedPaymentInstruments}" var="payment">

<isif condition="${payment.paymentMethod === 'Catch' && pdict.order.catch.catch\_earned !== 0}">

<isset name="paidWithCatch" value="${true}" scope="page" />

<div class="row">

<isinclude url="${URLUtils.url('Catch-IncludeWidget', 'widgetName', 'purchase', 'pdictOrderNo', pdict.order.orderNumber, 'pdictOrderToken', pdict.order.order\_token)}" />

</div>

</isif>

</isloop>

<isset name="catchHelper" value="${require('\*/cartridge/scripts/catch/helper/catchHelper')}" scope="page" />

<isif condition="${catchHelper.getPreference('enable')

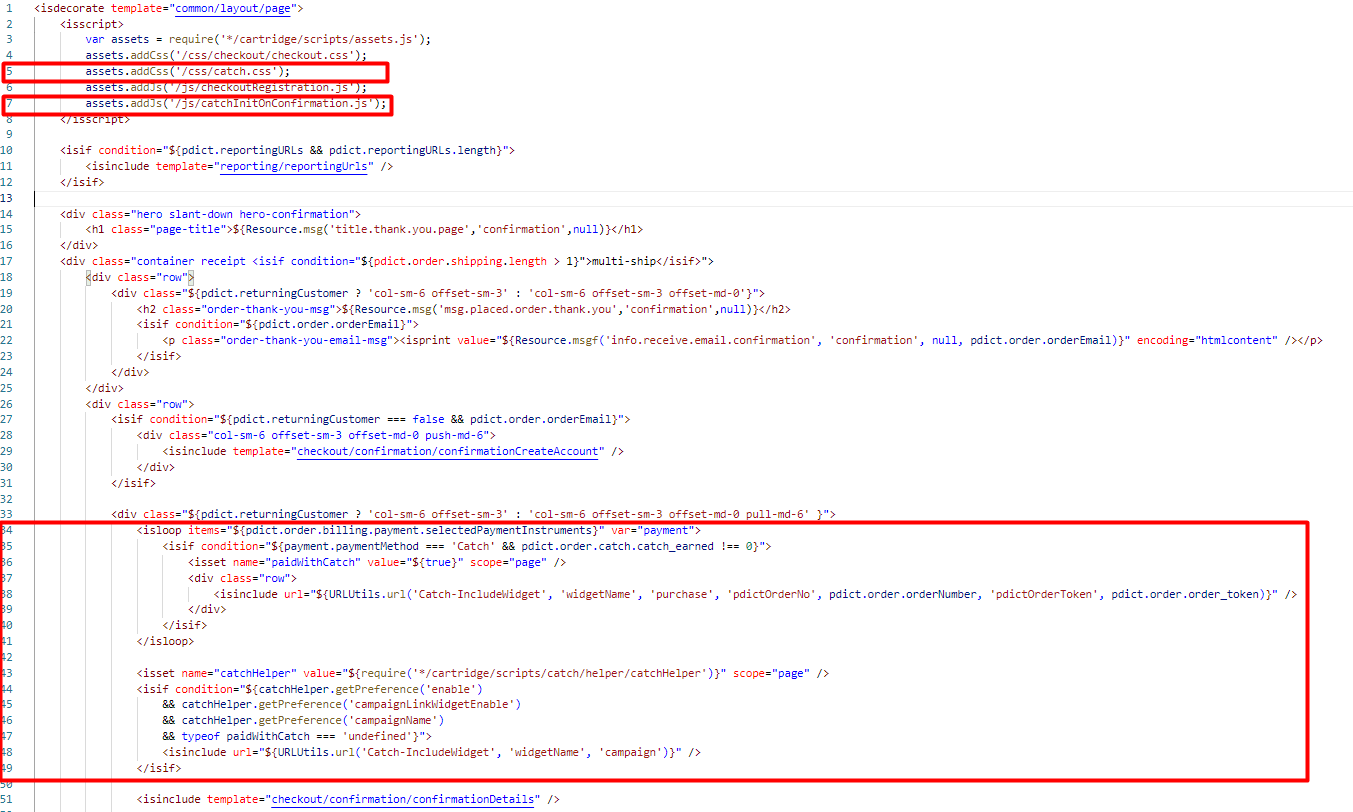
&& catchHelper.getPreference('campaignLinkWidgetEnable')

&& catchHelper.getPreference('campaignName')

&& typeof paidWithCatch === 'undefined'}">

<isinclude url="${URLUtils.url('Catch-IncludeWidget', 'widgetName', 'campaign')}" />

</isif>



1. app\_storefront\_base/cartridge/templates/default/checkout/confirmation/confirmationEmail.isml

Put the following rows in line 5:

<iscomment>Catch Code Start</iscomment>

<isloop items="${pdict.order.billing.payment.selectedPaymentInstruments}" var="payment">

<isif condition="${payment.paymentMethod === 'Catch'}">

<div>

<p>${Resource.msg('msg.confirmation.catch.mail', 'catch', null)}</p>

</div>

</isif>

</isloop>

<iscomment>Catch Code End</iscomment>



1. app\_storefront\_base/cartridge/templates/default/checkout/confirmation/confirmationOrderTotal.isml

Put the following rows in line 29:

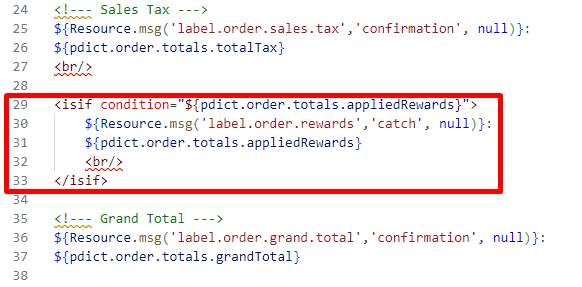
<isif condition="${pdict.order.totals.appliedRewards}">

${Resource.msg('label.order.rewards','catch', null)}:

${pdict.order.totals.appliedRewards}

<br/>

</isif>



1. app\_storefront\_base/cartridge/templates/default/checkout/confirmation/confirmationPaymentInfo.isml

Put the following rows in line 29:

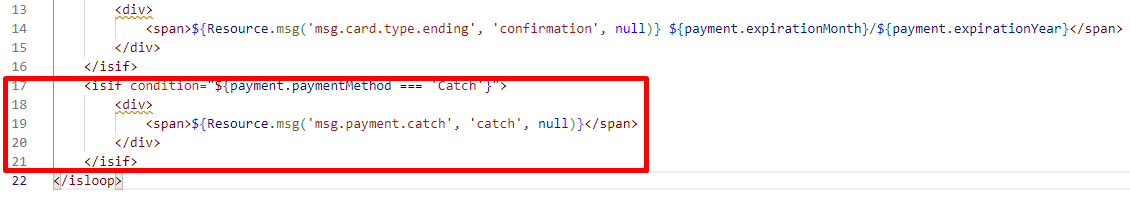
<isif condition="${payment.paymentMethod === 'Catch'}">

<div>

<span>${Resource.msg('msg.payment.catch', 'catch', null)}</span>

</div>

</isif>



1. app\_storefront\_base/cartridge/templates/default/common/htmlHead.isml

Put the following rows in line 32:

<link rel="stylesheet" href="${URLUtils.staticURL('/css/catch.css')}" />

<isinclude template="catch/catchSDK" />



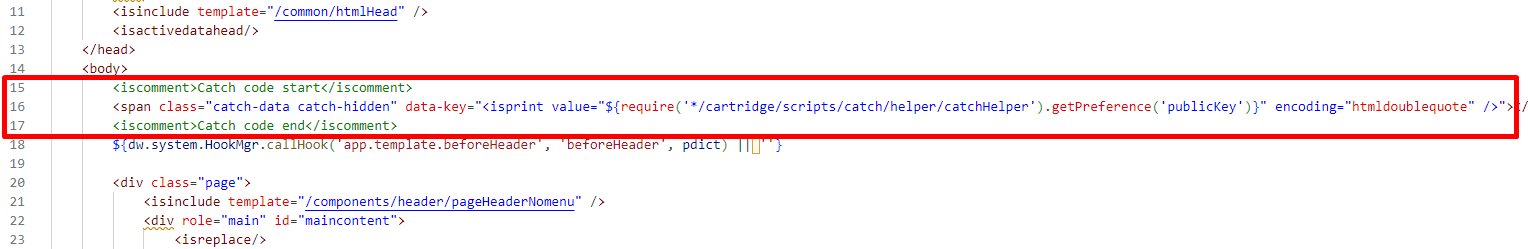
1. app\_storefront\_base/cartridge/templates/default/common/layout/checkout.isml

Put the following rows in line 15:

<iscomment>Catch code start</iscomment>

<span class="catch-data catch-hidden" data-key="<isprint value="${require('\*/cartridge/scripts/catch/helper/catchHelper').getPreference('publicKey')}" encoding="htmldoublequote" />"></span>

<iscomment>Catch code end</iscomment>



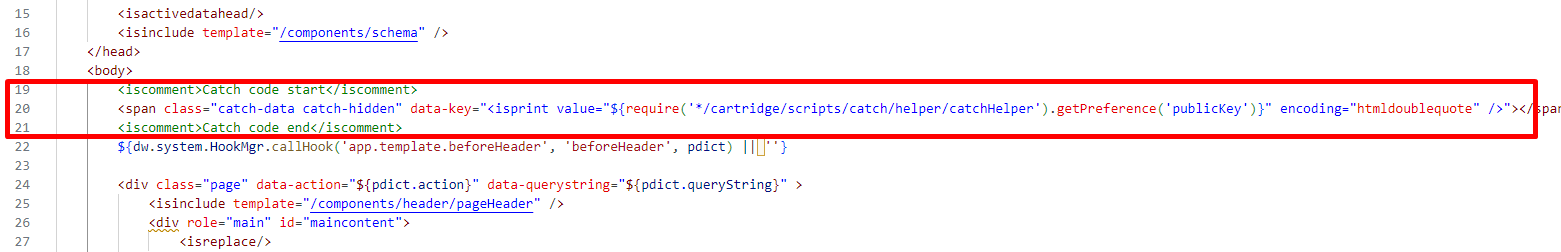
1. app\_storefront\_base/cartridge/templates/default/common/layout/page.isml

Put the following rows in line 19:

<iscomment>Catch code start</iscomment>

<span class="catch-data catch-hidden" data-key="<isprint value="${require('\*/cartridge/scripts/catch/helper/catchHelper').getPreference('publicKey')}" encoding="htmldoublequote" />"></span>

<iscomment>Catch code end</iscomment>



1. app\_storefront\_base/cartridge/templates/default/common/layout/pdStorePage.isml

Put the following rows in line 32:

<iscomment>Catch code start</iscomment>

<span class="catch-data catch-hidden" data-key="<isprint value="${require('\*/cartridge/scripts/catch/helper/catchHelper').getPreference('publicKey')}" encoding="htmldoublequote" />"></span>

<iscomment>Catch code end</iscomment>

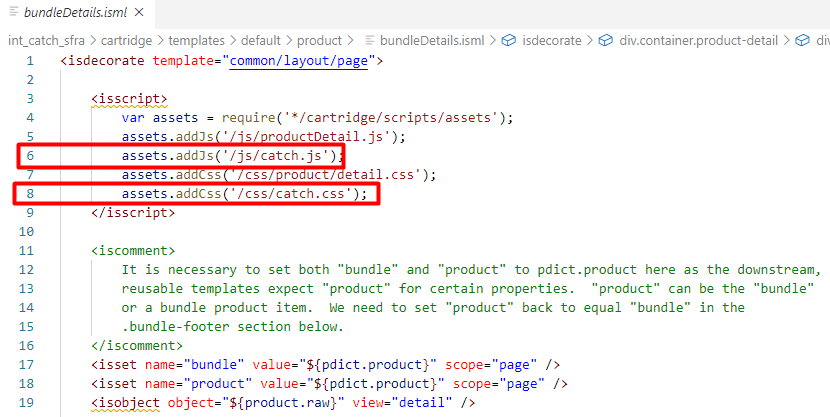


1. app\_storefront\_base/cartridge/templates/default/product/bundleDetails.isml

Add Catch scripts and styles on the page in the beginning of the file:

assets.addJs('/js/catch.js');

assets.addCss('/css/catch.css');



Add the Catch widget to the PDP bundle page. Put the following rows in line 49:

<iscomment> Catch Code </iscomment>

<div class="row">

<div class="col">

<isinclude url="${URLUtils.url('Catch-IncludeWidget', 'widgetName', 'callout', 'page', 'product', 'productId', product.id, 'productType', product.productType)}" />

</div>

</div>

<iscomment> Catch Code </iscomment>



1. app\_storefront\_base/cartridge/templates/default/product/productDetails.isml

Add Catch scripts and styles on the page in the beginning of the file:

assets.addJs('/js/catch.js');

assets.addCss('/css/catch.css');



Add the Catch widget to the PDP page. Put the following rows in line 58:

<iscomment> Catch Code </iscomment>

<div class="row">

<div class="col">

<isinclude url="${URLUtils.url('Catch-IncludeWidget', 'widgetName', 'callout', 'page', 'product', 'productId', product.id, 'productType', product.productType)}" />

</div>

</div>

<iscomment> Catch Code </iscomment>



1. app\_storefront\_base/cartridge/templates/default/product/setDetails.isml

Add Catch scripts and styles on the page in the beginning of the file:

assets.addJs('/js/catch.js');

assets.addCss('/css/catch.css');



1. app\_storefront\_base/cartridge/templates/default/product/components/setItems.isml

Add the Catch widget to the PDP page. Put the following rows in line 42:

<iscomment> Catch Code </iscomment>

<div class="row">

<div class="col">

<isinclude url="${URLUtils.url('Catch-IncludeWidget', 'widgetName', 'callout', 'page', 'product', 'productId', product.id, 'productType', product.productType)}" />

</div>

</div>

<iscomment> Catch Code </iscomment>

### 

### Perform changes in client script

**Note:** *Don’t forget to compile client side scripts after implementing changes in source code.*

1. app\_storefront\_base/cartridge/client/default/js/cart/cart.js

Put the following row in line 5:

var catchHelper = require('../catch/helper/catchHelper');

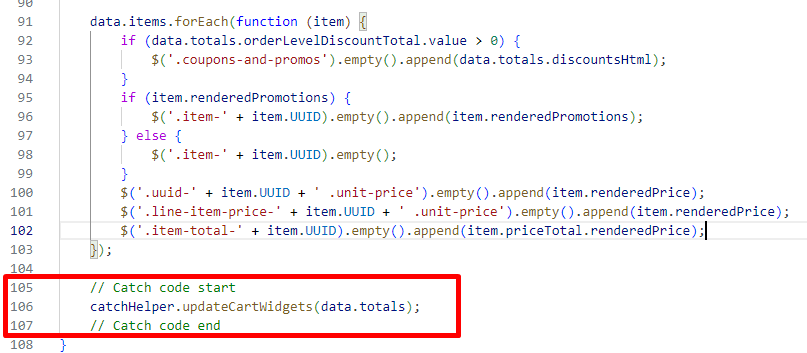


Put the following rows in end of updateCartTotals: function():

// Catch code start

catchHelper.updateCartWidgets(data.totals);

// Catch code end



**Note: Also make sure to call *catchHelper.updateCartWidgets();* on your custom Storefront every time totals are changed on the Cart page. This way Catch widgets will be updated with an actual total price.**

Put the following rows to the line 373:

// Catch code start

var miniCartWidget = $('.minicart .catch-pdp-widget');

var cartCalloutWidget = $('.cart-page .catch-pdp-widget');

if (data.isIneligibleProductInBasket) {

catchHelper.hideWidget(miniCartWidget);

catchHelper.hideWidget(cartCalloutWidget);

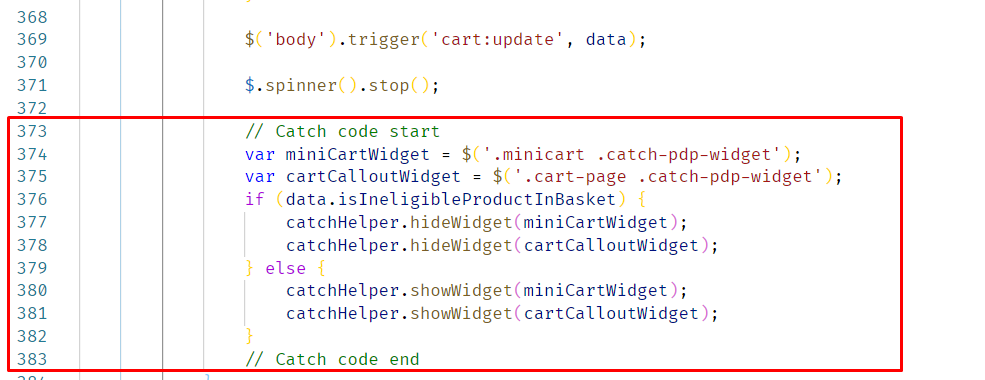
} else {

catchHelper.showWidget(miniCartWidget);

catchHelper.showWidget(cartCalloutWidget);

}

// Catch code end



1. app\_storefront\_base/cartridge/client/default/js/checkout/billing.js

Modified function updatePaymentInformation(). Add the switch statement in line 189:

switch (order.billing.payment.selectedPaymentInstruments[0].paymentMethod) {

case 'Catch':

htmlToAppend += '<div data-payment="catch"><span>'

+ order.resources.catch.paymentMethod

+ '</span></div>';

break;

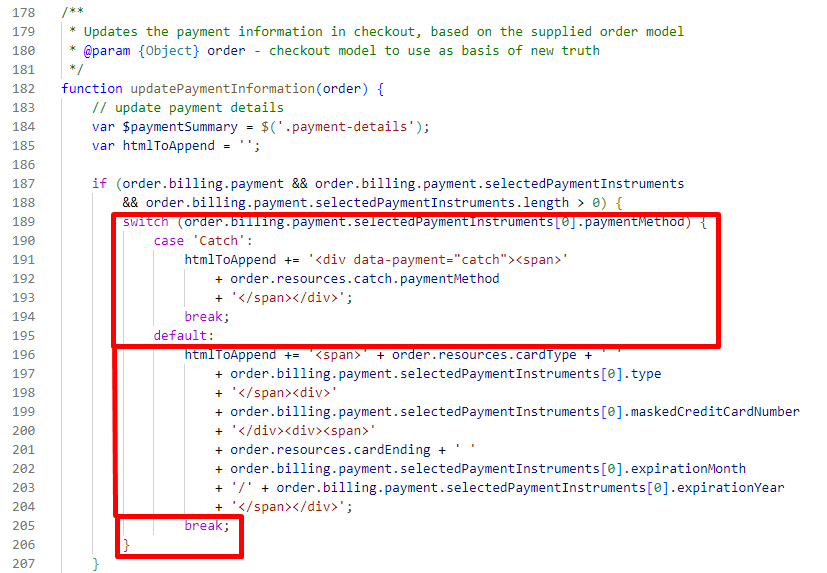
default:

**… 'htmlToAppend' content goes here**

break;

}

}

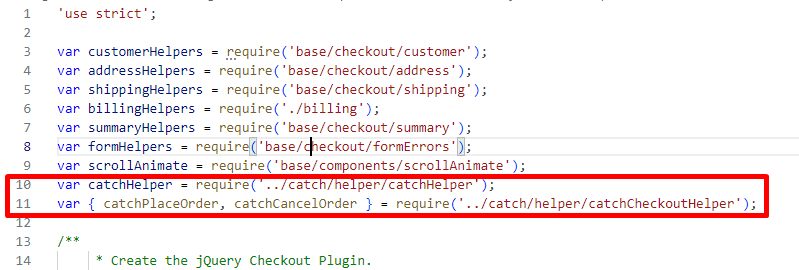


1. app\_storefront\_base/cartridge/client/default/js/checkout/checkout.js

Put the following row in line 10:

var catchHelper = require('../catch/helper/catchHelper');

var { catchPlaceOrder, catchCancelOrder } = require('../catch/helper/catchCheckoutHelper');

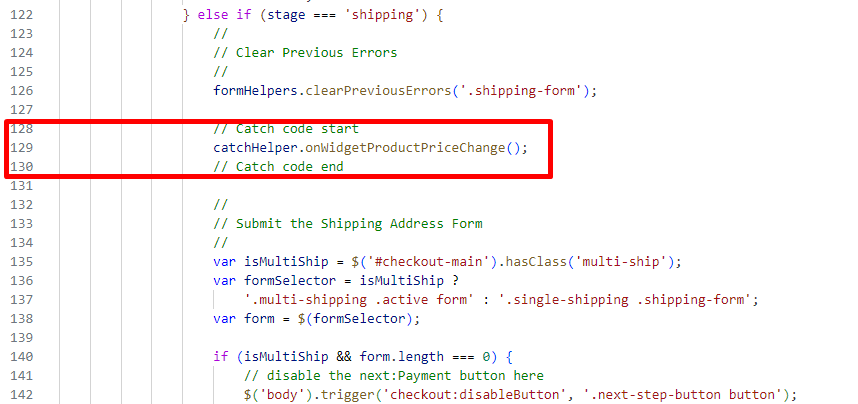


Put the following rows in line 128:

// Catch code start

catchHelper.onWidgetProductPriceChange();

// Catch code end



Modify the 'placeOrder' stage code:

var isCatchPaymentMethod = $('[data-payment="catch"]').length;

if (isCatchPaymentMethod) {

$.ajax({

url: $('.place-order').data('catch-create-order-url'),

method: 'GET',

success: function (data) {

// enable the placeOrder button here

$('body').trigger('checkout:enableButton', '.next-step-button button');

if (data.error) {

if (data.cartError) {

window.location.href = data.redirectUrl;

defer.reject();

} else {

// go to appropriate stage and display error message

defer.reject(data);

}

} else {

$('.error-message').hide();

var catchCheckoutID = data.order.catch.catch\_checkoutId;

var isCatchVirtualCard = data.catch.isCatchVirtualCard;

var option = {

inline: true,

autoCloseOnConfirm: data.catch.autoCloseOnConfirm,

hideHeader: data.catch.hideHeader,

oprefill: catchHelper.getCatchPrefilledData(data)

};

if (catchCheckoutID) {

option.onCancel = function () {

catchCancelOrder(data.order.orderNumber, data.order.order\_token)

.then(defer.reject);

};

if (isCatchVirtualCard) {

option.onConfirm = function (cardDetails) { // eslint-disable-line no-param-reassign

$('body').trigger('checkout:disableButton', '.next-step-button button');

catchPlaceOrder(data.order.orderNumber, data.order.order\_token, cardDetails)

.then(defer.resolve, defer.reject);

};

} else {

option.onConfirm = function () { // eslint-disable-line no-param-reassign

$('body').trigger('checkout:disableButton', '.next-step-button button');

catchPlaceOrder(data.order.orderNumber, data.order.order\_token)

.then(defer.resolve, defer.reject);

};

}

catchHelper.initializeCatchSDK('checkout', catchCheckoutID, option, isCatchVirtualCard);

}

}

},

error: function () {

// enable the placeOrder button here

$('body').trigger('checkout:enableButton', $('.next-step-button button'));

}

});

} else {

**… 'placeOrder' stage content goes here**

}



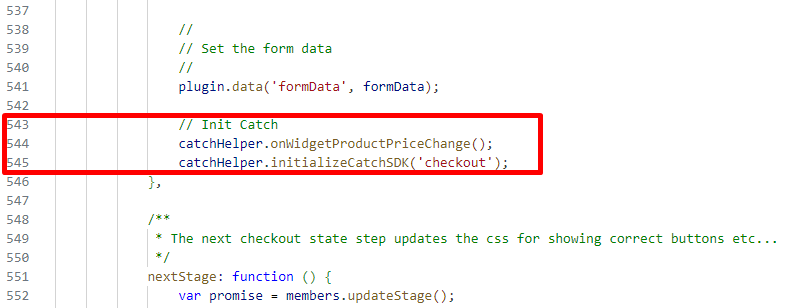


Add to the end of initialize: function() initialization Catch widget. Put the following rows in line 543:

// Init Catch

catchHelper.onWidgetProductPriceChange();

catchHelper.inititializeCatchSDK('checkout');



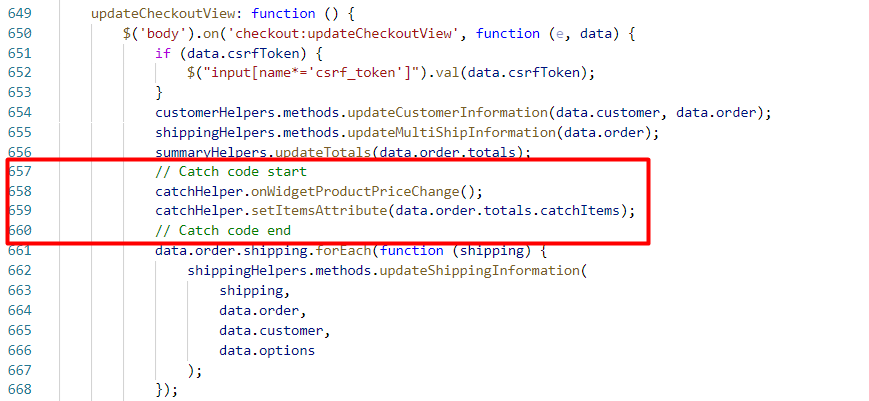
Put the following rows in line 657:

// Catch code start

catchHelper.onWidgetProductPriceChange();

catchHelper.setItemsAttribute(data.order.totals.catchItems);

// Catch code end

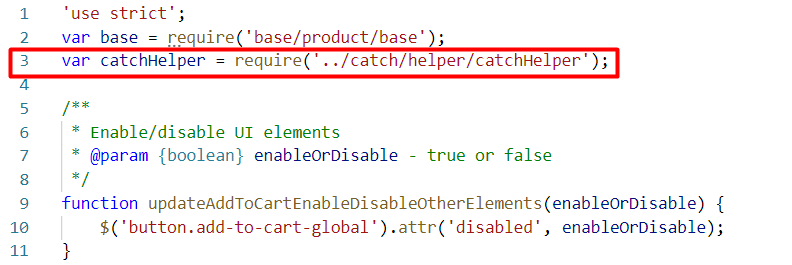


**Note: Also make sure to call *catchHelper.onWidgetProductPriceChange();* on your custom Storefront every time totals are changed during checkout. This way Catch widgets will be updated with an actual total price.**

1. app\_storefront\_base/cartridge/client/default/js/product/detail.js

Put the following row in line 3:

var catchHelper = require('../catch/helper/catchHelper');



Put the following rows to updateAttribute: function() in line 61:

// Catch code start

var calloutWidget = response.container.find('.catch-pdp-widget');

catchHelper.onWidgetProductPriceChange();

catchHelper.updateItemsAttributePDP(response);

if (response.data.isVisibleCallout) {

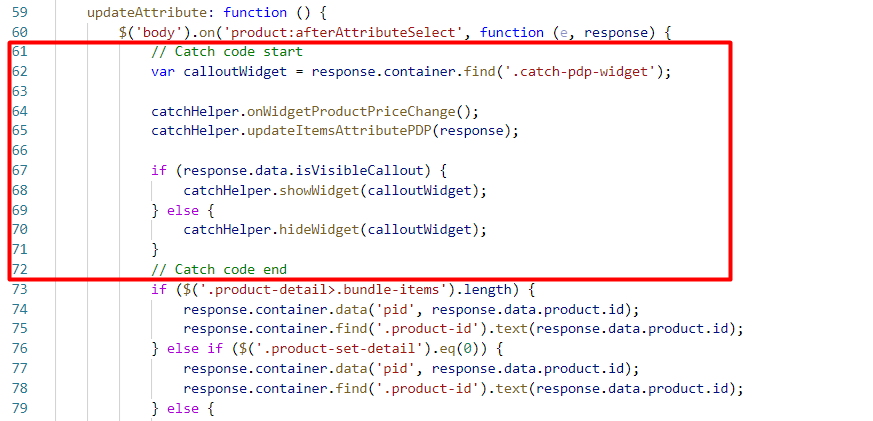
catchHelper.showWidget(calloutWidget);

} else {

catchHelper.hideWidget(calloutWidget);

}

// Catch code end



1. app\_storefront\_base/cartridge/client/default/js/main.js

Put the following row in line 15:

processInclude(require('./catchInit'));



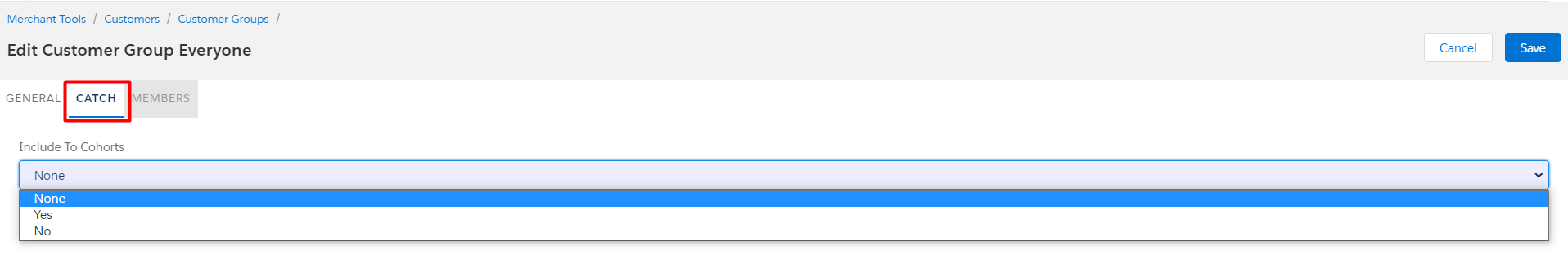
## Configuration

### Customer Groups

After import is successfully done, you should see Catch settings tab for Customer Groups in BM > Merchant Tools > Customers > Customer Groups

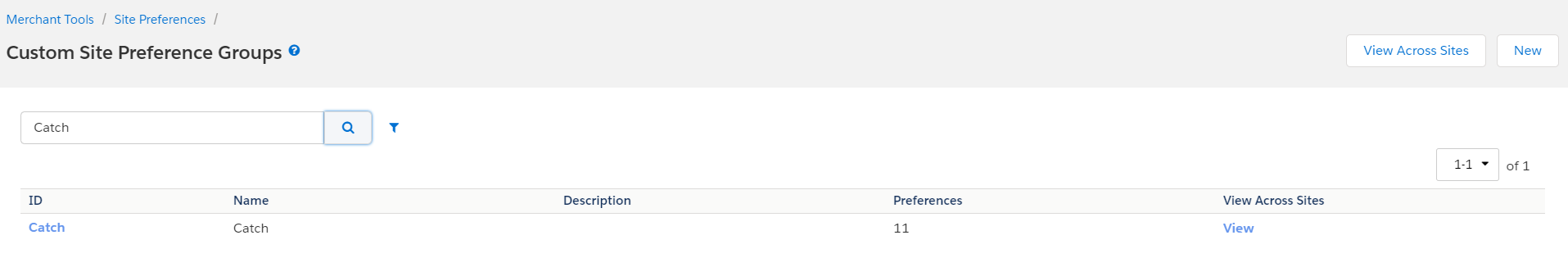
If the customer belongs to a particular Customer Group and you want to send this data to Catch in order to tailor Catch to each consumer’s context set “Include To Cohorts” attribute value to “Yes”.

Learn more about User Cohorts at <https://catch.readme.io/reference/user-cohorts>

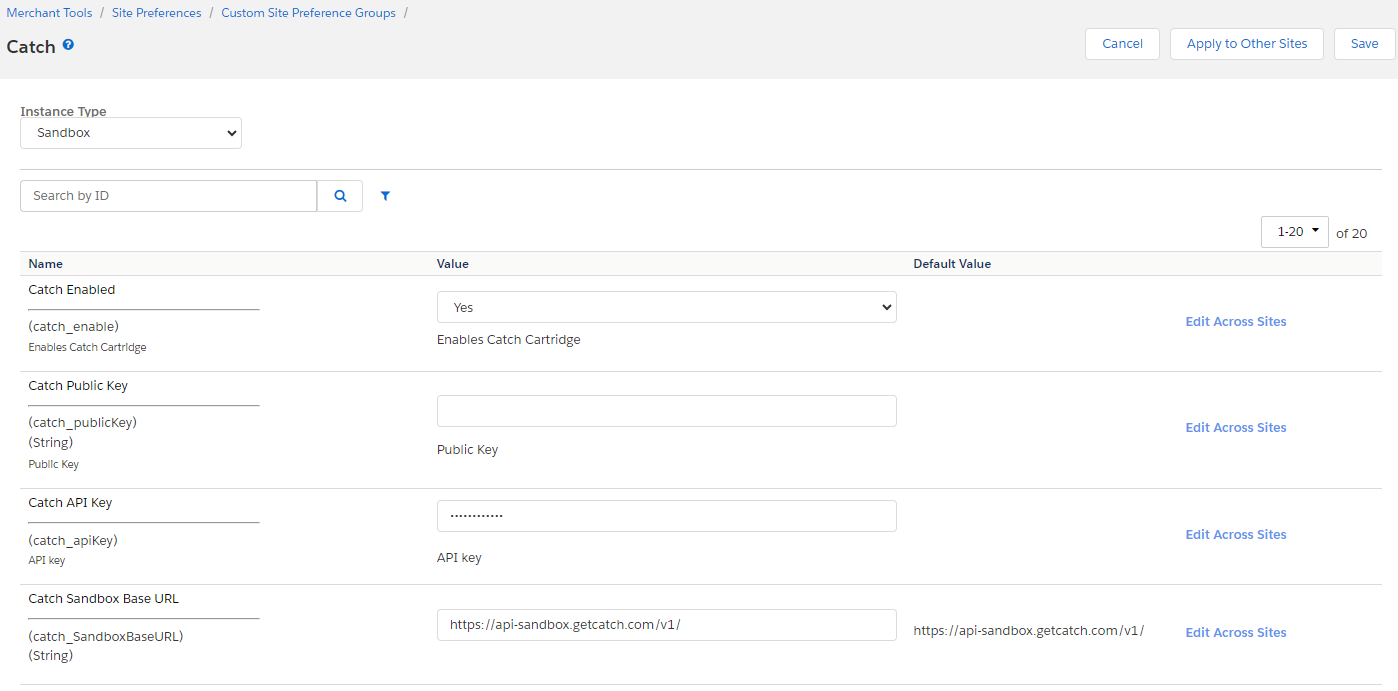


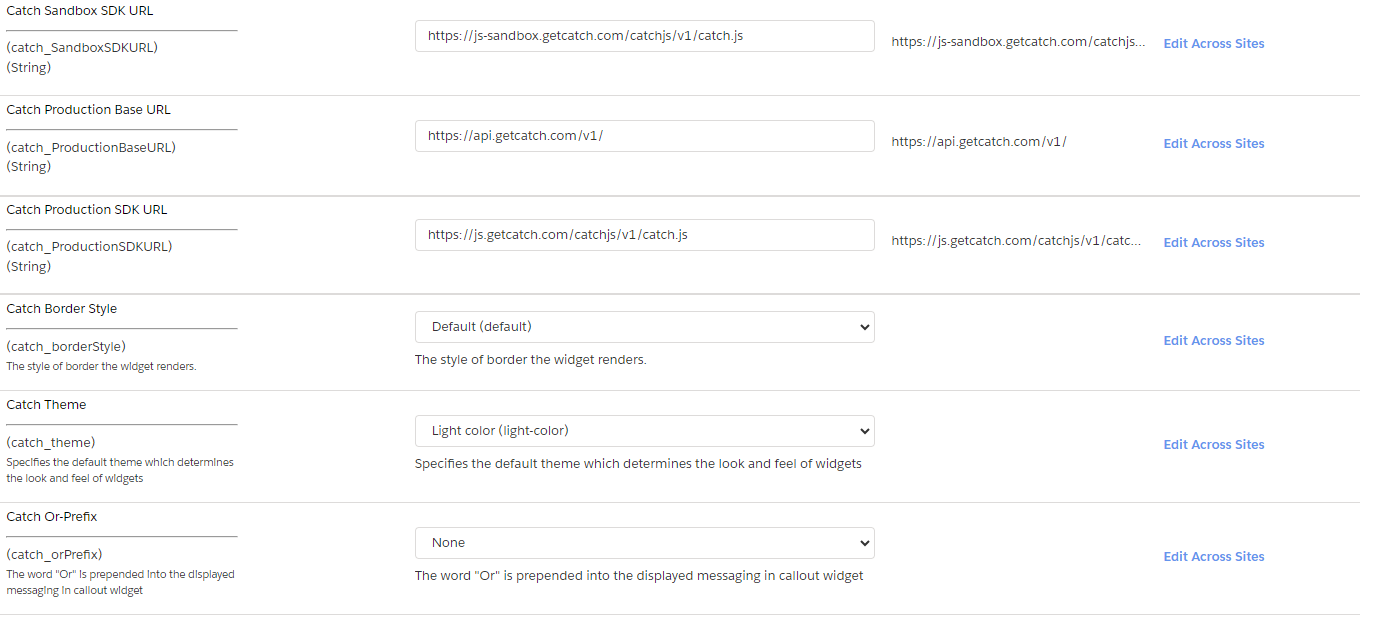
### Custom Site Preferences

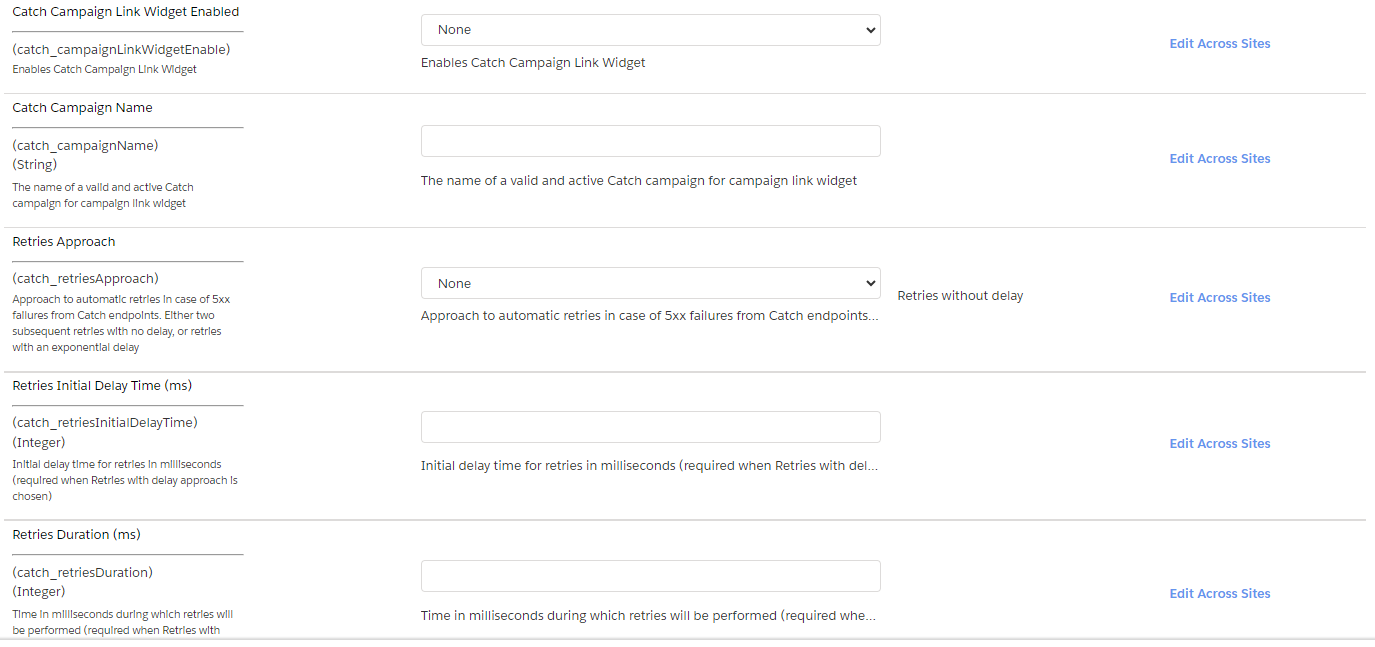
After import is successfully done, you should see Site Preferences in BM > Merchant Tools > Site Preferences > Custom Preferences.

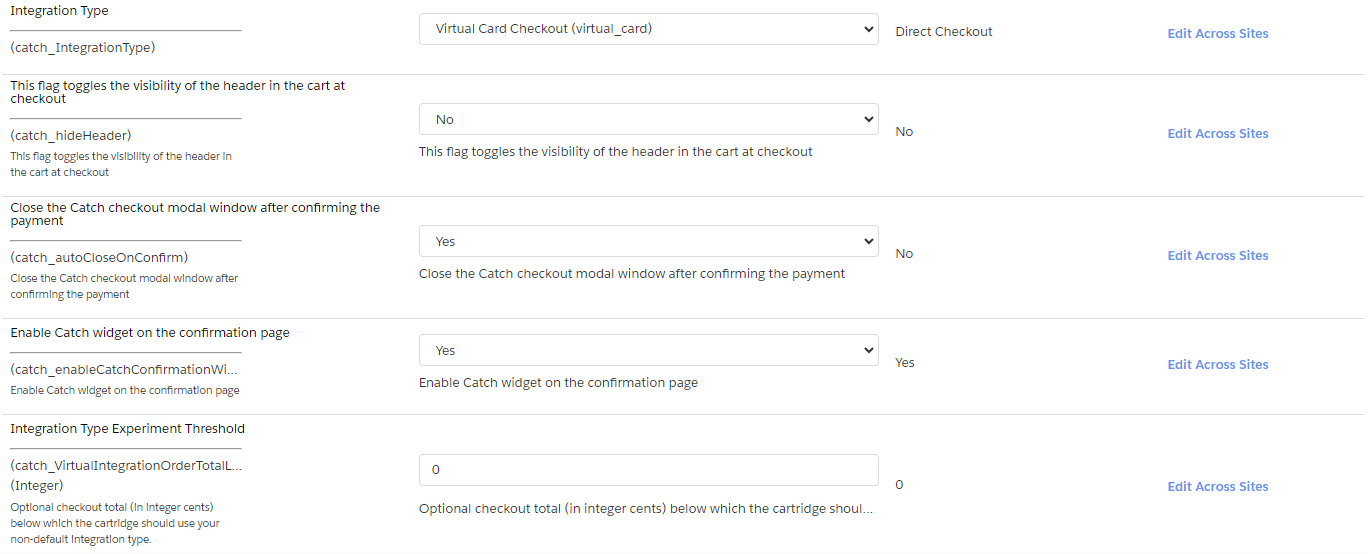


Select Catch and you should see fields for cartridge setup.









| # | Field Title | Description |
| --- | --- | --- |
| 1 | Catch Enable | Enable/Disable all Catch functionality. |
| 2 | Catch Public Key | Catch Public Key assigned by Catch Team. |
| 3 | Catch API Key | Catch API Key assigned by Catch Team. |
| 5 | Catch Sandbox Base URL | Sandbox URL to access the Catch REST API. |
| 6 | Catch Sandbox SDK URL | Sandbox URL to access to Catch frontend JavaScript SDK. |
| 7 | Catch Production Base URL | Production URL to access the Catch REST API. |
| 8 | Catch Production SDK URL | Production URL to access to Catch frontend JavaScript SDK. |
| 9 | Catch Border Style | Catch widget attribute border style. |
| 10 | Catch Theme | Catch widget theme attribute. |
| 11 | Catch Or-Prefix | Catch widget or-prefix attribute. |
| 12 | Catch Campaign Link Widget Enabled | Enables Catch Campaign Link Widget. |
| 13 | Catch Campaign Name | The name of a valid and active Catch campaign for campaign link widget. |
| 14 | Retries Approach | Approach to automatic retries in case of 5xx failures from Catch endpoints. Either two subsequent retries with no delay, or retries with an exponential delay. |
| 15 | Retries Initial Delay Time (ms) | Initial delay time for retries in milliseconds (required when Retries with delay approach is chosen). |
| 16 | Retries Duration (ms) | Time in milliseconds during which retries will be performed (required when Retries with delay approach is chosen). |
| 17 | Integration Type | Select Direct checkout or Virtual Card checkout integration type. |
| 18 | This flag toggles the visibility of the header in the cart at checkout | This flag toggles the visibility of the header in the cart at checkout. The header contains a logo image and a back button. When the header is gone, the consumer can still use the browser back button to navigate away from checkout. This flag is intended to be used when your site is embedded in a web view in a mobile application that already has a static header with your logo. |
| 19 | Close the Catch checkout modal window after confirming the payment | Automatically close the Catch checkout modal window after confirming the payment. |
| 20 | Enable Catch widget on the confirmation page | Show Catch widget on the confirmation page. |
| 21 | Integration Type Experiment Threshold | Order amount at which the choice of integration type changes.  If the order amount is higher than the specified value, use the integration type specified in the “Integration Type”. Otherwise another is Integration Type used. |

## Firewall Requirements

No special firewall requirements

# Testing

Test cases for checking cartridge performance are described in the “Catch test cases” document.

# Operations, Maintenance

## Data Storage

Intentionally left blank

## Availability

If the Catch service becomes unavailable, the storefront continues to work without the service. All errors work of the service are saved in the log file.

## Failover/Recovery Process

Intentionally left blank

## Support

For questions please contact your Catch Success Manager.

# Known Issues

# Release History

| **Version** | **Date** | **Changes** |
| --- | --- | --- |
| 22.1.0 | 2021-01-01 | Initial release |
| 22.2.0 | 2022-05-31 | - Fixed SFRA checkout bug  - Hide Catch Callout on >$5000 items and carts  - Catch Redeemed Credits as a line item on confirmation page  - Updated Catch logo on payment method selection step  - Updated transactional emails |
| 22.3.0 | 2022-11-23 | - Fixed shipping address “name” field bug  - Added user cohort data  - Added item data  - Added campaign link widget  - Added Retry logic to calls to POST /checkout, POST /purchase, and POST /refund  - Upgraded POST /checkout API request |
| 23.1.0 | 2023-08-20 | - Added Virtual Cart Checkout functionality  - Improve error handling |
|  |  |  |