****

*SG LINK Cartridge Integration guide*

*Version 23.1.0 – SG*



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) 7

[Service](#_35nkun2) 8

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

[Perform changes in controllers](#_478b18pjwbtj) 8

[Perform changes in templates](#_zd5dztkdrlg4) 12

[Perform changes in resources](#_yohzpc8qdyxz) 22

[*Configuration*](#_3rdcrjn) *22*

[Customer Groups](#_2y8scq8zd9wc) 22

[Custom Site Preferences](#_26in1rg) 22

[*Firewall Requirements*](#_z337ya) *24*

[**4. Testing**](#_3j2qqm3) **24**

[**5. Operations, Maintenance**](#_1y810tw) **24**

[*Data Storage*](#_4i7ojhp) *24*

[*Availability*](#_2xcytpi) *24*

[*Failover/Recovery Process*](#_1ci93xb) *24*

[*Support*](#_3whwml4) *24*

[**6. Known Issues**](#_2p2csry) **25**

[**7. Release History**](#_147n2zr) **25**

# 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 |
| SG | Site Genesis |

# 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

SG version 105.2.0. The cartridge is designed with US locale but multi-locale and US currency support.

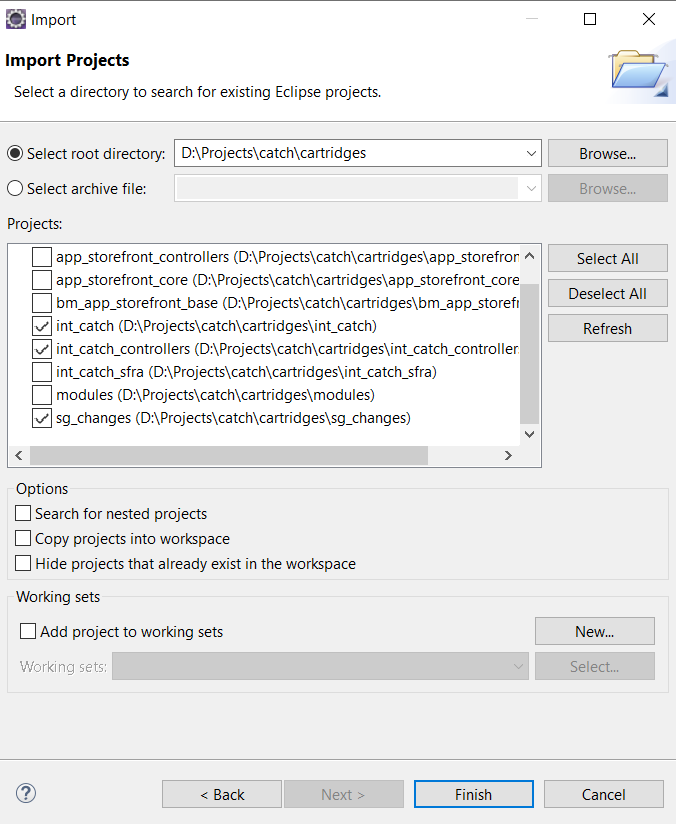
## Privacy, Payment

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

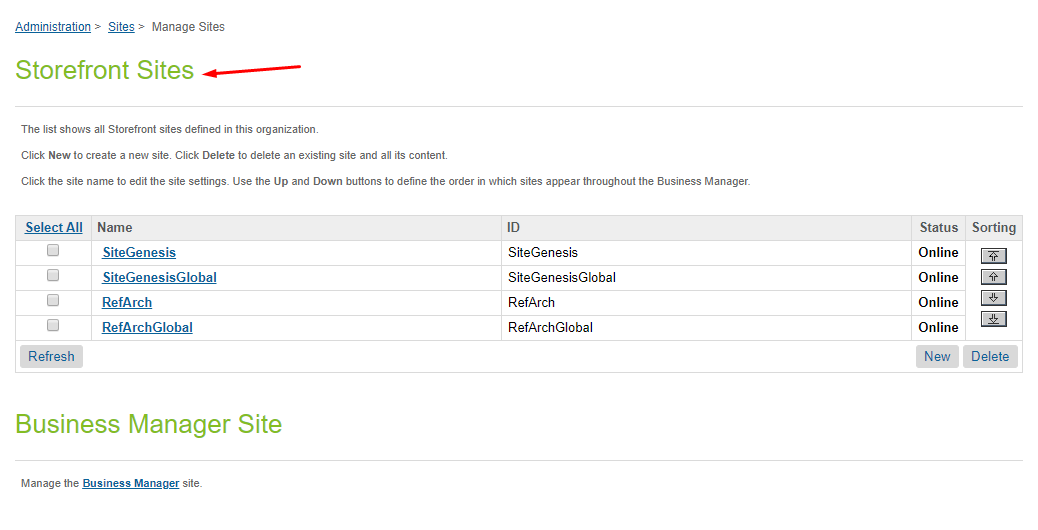
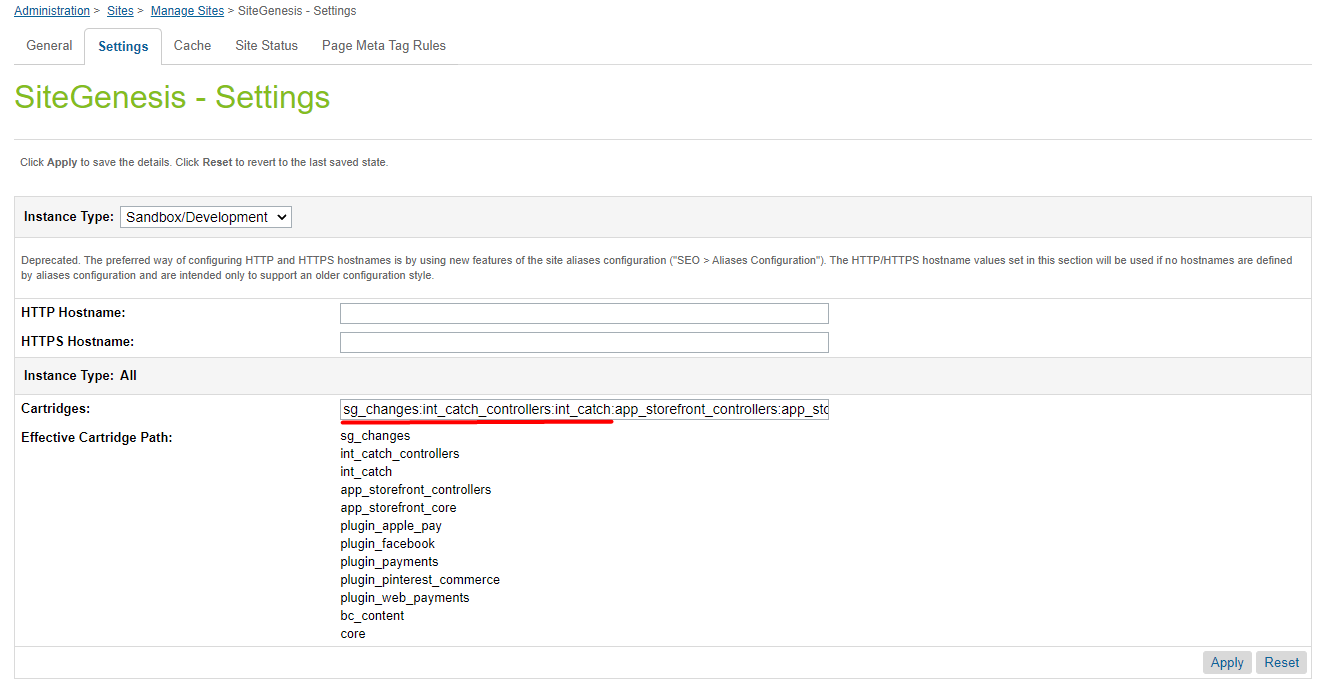
# Implementation Guide

## Setup of Business Manager

### Importing cartridges

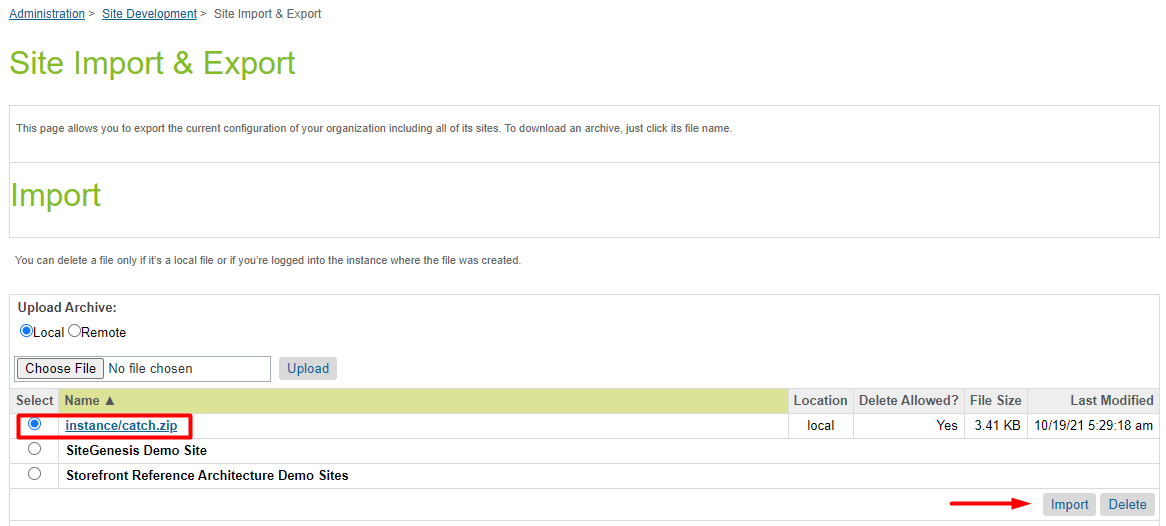
1. Download the cartridge source code.
2. Establish a new digital server connection with your SFCC Instance
3. Import cartridges to a workspace in Salesforce UX Studio.  
   
4. Add cartridges to Project Reference of Server Connection.  
   
5. Wait until Studio completes workspace build and uploads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


### Import of metadata

Go to Administration > Site Development > Site Import & Export. Upload *catch.zip* from the metadatafolder.



Select catch.zip and finish the import process.

### Service

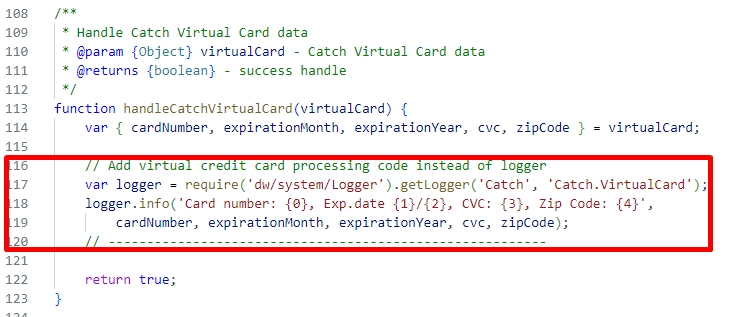
The service does not require additional settings.

## 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.



**Note: all changes are already made in the sg\_changes cartridge.**

### Perform changes in controllers

1. YOUR\_CARTRIDGE\_controllers/cartridge/controllers/COBilling.js

Put the following rows to the function resetPaymentForms() in line 371:

} else if (app.getForm('billing').object.paymentMethods.selectedPaymentMethodID.value.equals('Catch')) {

var paymentInstruments = cart.object.paymentInstruments.toArray();

paymentInstruments.forEach(function (paymentInstrument) {

var paymentMethod = paymentInstrument.paymentMethod;

if (paymentMethod !== 'Catch') {

cart.removePaymentInstruments(cart.getPaymentInstruments(paymentMethod));

}

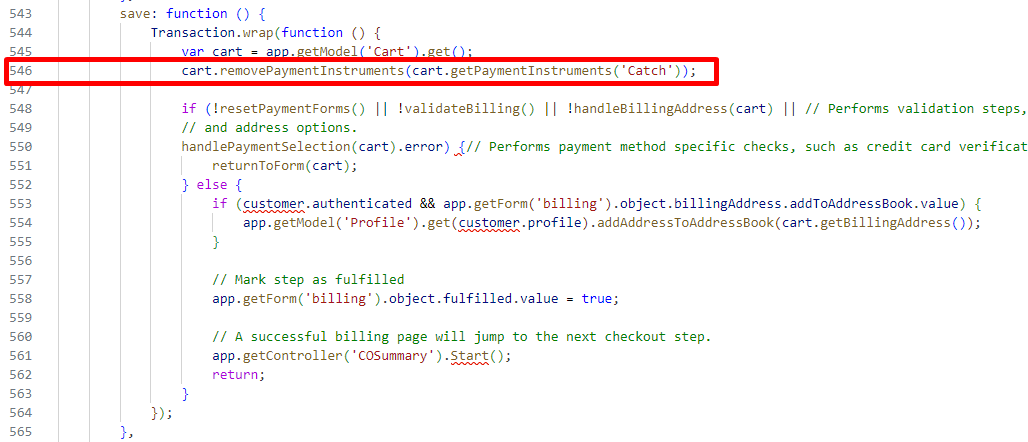
});

}



Put the following rows to the function save() in line 546:

cart.removePaymentInstruments(cart.getPaymentInstruments('Catch'));



1. YOUR\_CARTRIDGE\_controllers/cartridge/controllers/COPlaceOrder.js

Add parameter ‘stage’ to the function start() in line 87:

function start(stage) {

Put the following rows to the begin of the function start() in line 88:

var serviceHelper = require('\*/cartridge/scripts/catch/services/serviceHelper');

var catchHelper = require('\*/cartridge/scripts/catch/helper/catchHelper');

var catchCheckoutHelper = require('\*/cartridge/scripts/catch/helper/catchCheckoutHelper');

Wrap handling shipping and billing form to the if statement in the function start() in line 91:

if (!stage || stage.stage === 'create\_order') {

var cart = Cart.get();

**… handling shipping and billing form code goes here**

// --- Catch code start ---

var catchIntegrationType = (stage && stage.stage === 'create\_order')

? catchHelper.getCheckoutIntegrationType(cart) : null;

// --- Catch code end ---

**… create order code goes here**

// --- Create Catch checkout ---

if (stage && stage.stage === 'create\_order') {

var catchCheckoutResult = serviceHelper.createCheckout(order, catchIntegrationType);

if (catchCheckoutResult.error) {

Transaction.wrap(function () { OrderMgr.failOrder(order); });

return catchCheckoutResult;

}

try {

Transaction.wrap(function () {

order.custom.catch\_checkoutId = catchCheckoutResult.data.id;

order.custom.catch\_purchaseId = order.UUID;

order.custom.catch\_integrationType = catchIntegrationType;

});

} catch (e) {

return {

error: true,

PlaceOrderError: new Status(Status.ERROR, 'confirm.error.technical')

};

}

return {

orderNo: order.orderNo,

orderToken: order.orderToken,

catchCheckoutId: catchCheckoutResult.data.id,

isCatchVirtualCard: catchIntegrationType === catchHelper.VIRTUAL\_CARD\_CHECKOUT

};

}

// ---- Catch code end ----

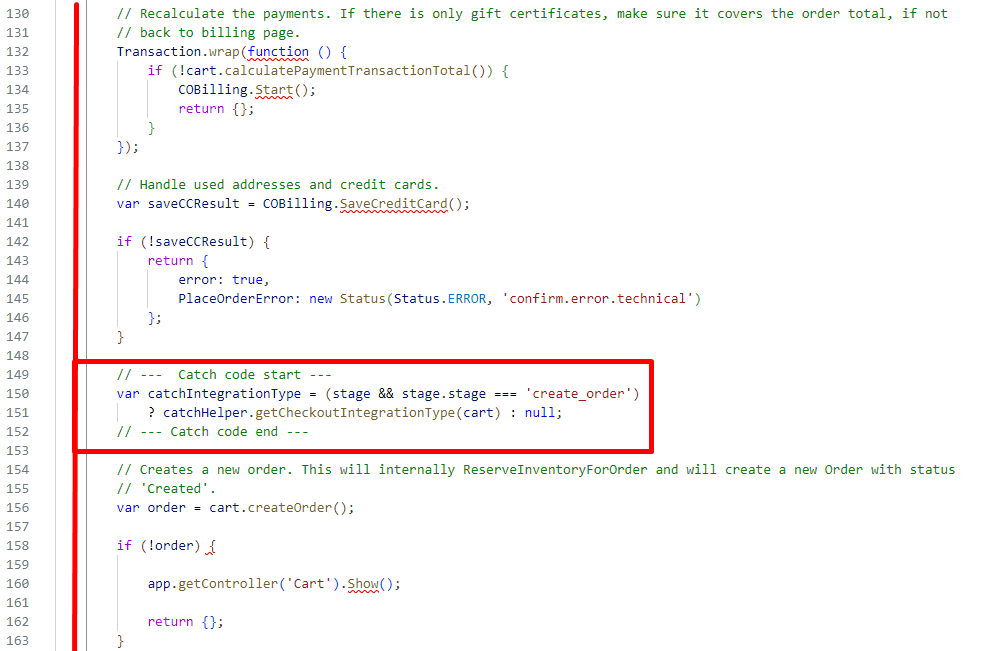
}

if (stage && stage.stage === 'place\_order') {

order = stage.order;

}







Put the following rows to the function save() in line 220:

// --- Catch code start ----

if (stage && stage.stage === 'place\_order') {

if (order.getPaymentInstruments('Catch').length === 0) {

return {

error: true,

PlaceOrderError: new Status(Status.ERROR, 'confirm.error.technical')

};

}

if (order.custom.catch\_integrationType === catchHelper.VIRTUAL\_CARD\_CHECKOUT) {

// --- Handle Catch Virtual Card Checkout ---

if (!catchCheckoutHelper.handleCatchVirtualCard(stage.virtualCard, order)) {

return {

error: true,

PlaceOrderError: new Status(Status.ERROR, 'confirm.error.technical')

};

}

} else {

// --- Handle Catch purchase for Direct Checkout ---

var catchPurchaseResult = serviceHelper.createPurchase(order);

if (!catchPurchaseResult) {

return {

error: true,

PlaceOrderError: new Status(Status.ERROR, 'error.purchase.id')

};

}

if (!catchCheckoutHelper.handleCatchPurchaseResponce(order, catchPurchaseResult)) {

return {

error: true,

PlaceOrderError: new Status(Status.ERROR, 'confirm.error.technical')

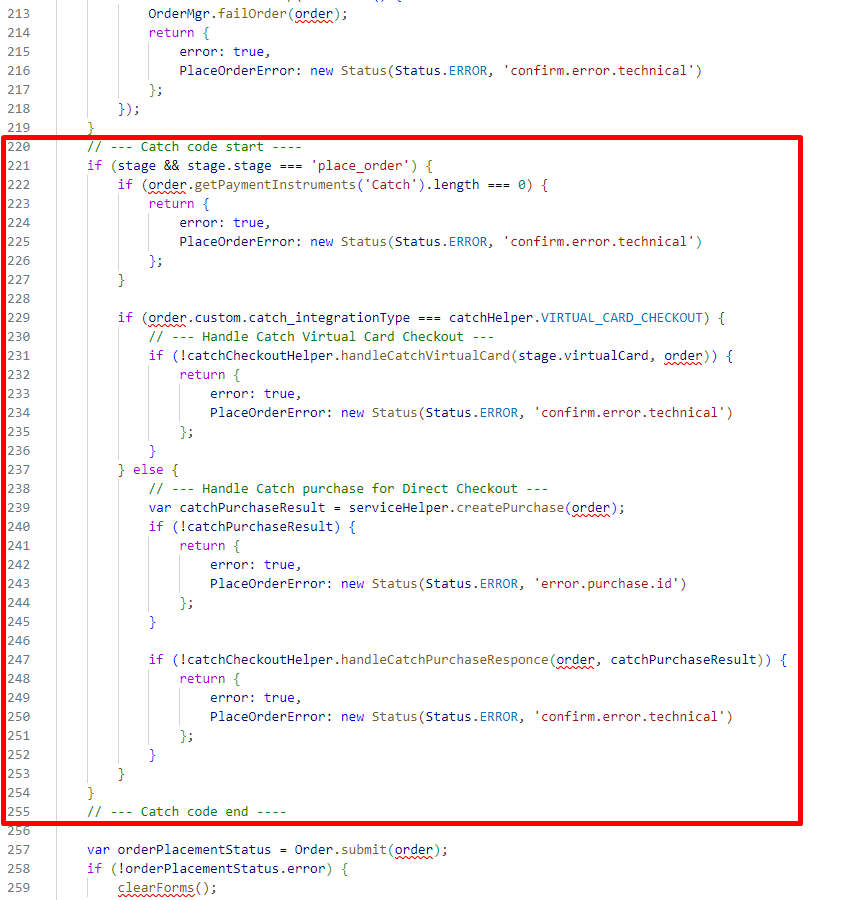
};

}

}

}

// --- Catch code end ----



### Perform changes in templates

1. YOUR\_CARTRIDGE\_core/cartridge/templates/default/checkout/minisummary.isml

Put the following rows in line 44:

<iscomment>Catch Code</iscomment>

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

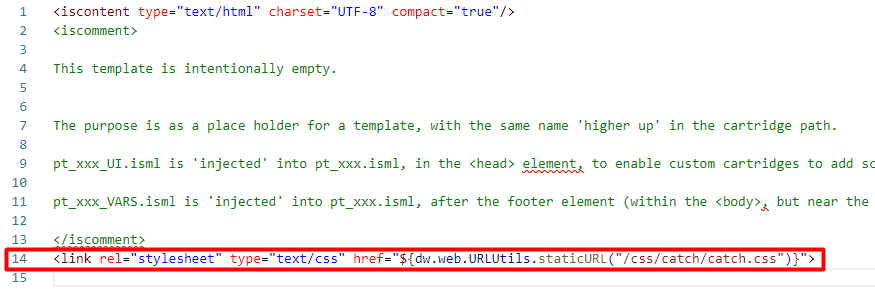
<iscomment>Catch Code</iscomment>



1. YOUR\_CARTRIDGE\_core/cartridge/templates/default/checkout/pt\_checkout\_UI.isml

Add Catch styles to the end of the file:

<link rel="stylesheet" type="text/css" href="${dw.web.URLUtils.staticURL("/css/catch/catch.css")}">

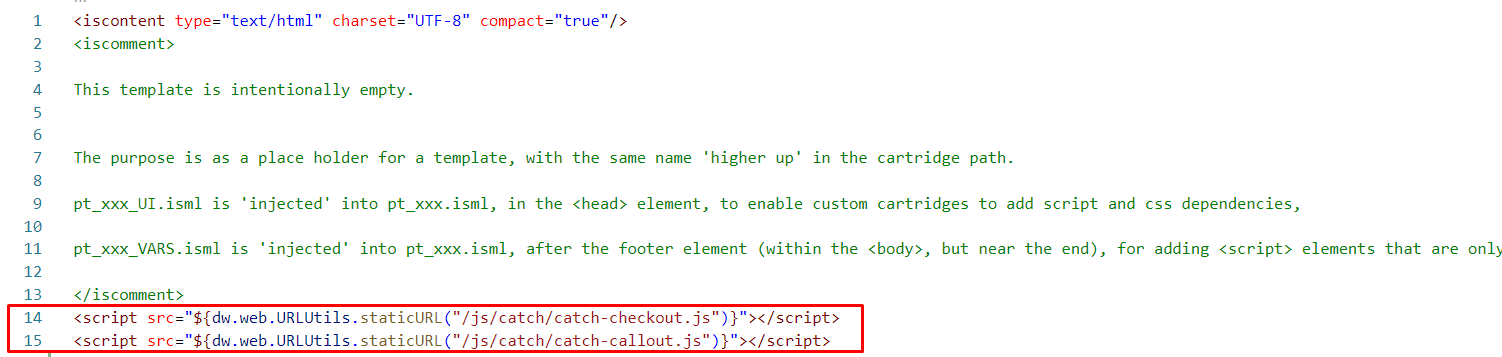


1. YOUR\_CARTRIDGE\_core/cartridge/templates/default/checkout/pt\_checkout\_VARS.isml

Add Catch scripts to the end of the file:

<script src="${dw.web.URLUtils.staticURL("/js/catch/catch-checkout.js")}"></script>

<script src="${dw.web.URLUtils.staticURL("/js/catch/catch-callout.js")}"></script>



1. YOUR\_CARTRIDGE\_core/cartridge/templates/default/checkout/billing/billing.isml

Put the following rows in line 31:

<isif condition="${pdict.error === true}">

<div class="error-message catch-checkout-error">

${pdict.errorMessage}

</div>

</isif>



1. YOUR\_CARTRIDGE\_core/cartridge/templates/default/checkout/billing/paymentmethods.isml

Extend <div> tag with class="payment-method-options" in line 12:

<div class="payment-method-options form-indent catch-flex">

Put the following row in line 13:

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

Put the following row in line 20:

<isif condition="${paymentMethodType.value.equals('Catch') && !catchHelper.getPreference('enable')}"><iscontinue/></isif>

Extend <div> tag with id="checkout-main" in line 22:

<div class="form-row label-inline <isif condition="${paymentMethodType.value.equals('Catch')}">label-hidden</isif>">

Extend <div> tag with class="field-wrapper" in line 24:

<div class="field-wrapper catch-flex">

Put the following row in line 26:

<isif condition="${radioID === 'Catch'}">

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

</isif>

Extend <label for="is-${radioID}"> tag in line 30:

<label for="is-${radioID}">

<isif condition="${radioID !== 'Catch'}">

<isprint value="${Resource.msg(paymentMethodType.label,'forms',null)}"/>

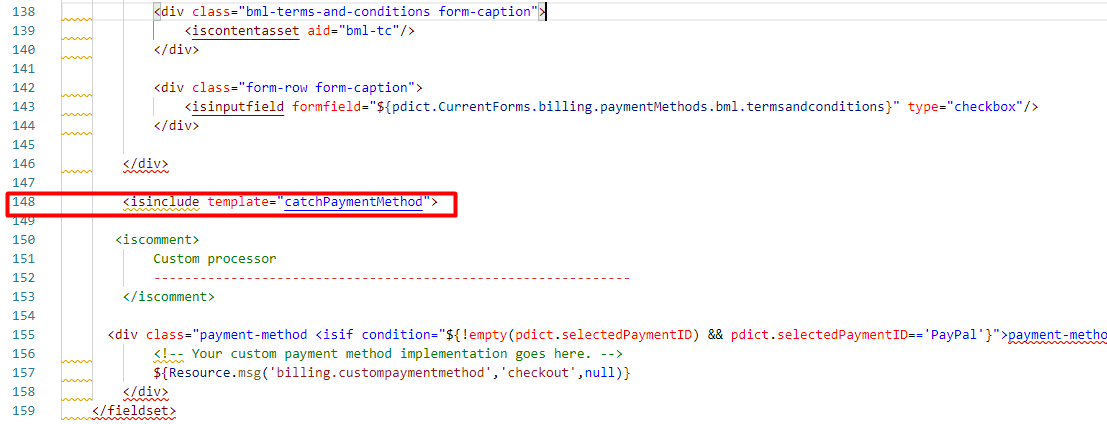
</isif>

</label>



Put the following row in line 148:

<isinclude template="catchPaymentMethod">



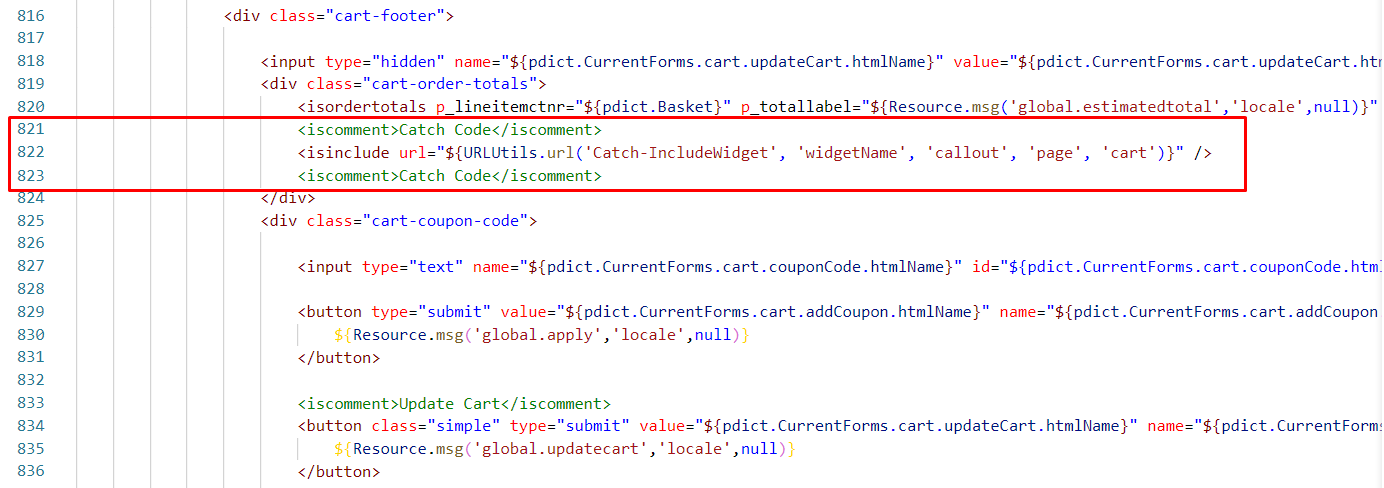
1. YOUR\_CARTRIDGE\_core/cartridge/templates/default/checkout/cart/cart.isml

Put the following rows in line 821:

<iscomment>Catch Code</iscomment>

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

<iscomment>Catch Code</iscomment>



1. YOUR\_CARTRIDGE\_core/cartridge/templates/default/checkout/cart/minicart.isml

Put the following rows in line 61:

<div class="callout-mini-cart-wrapper">

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

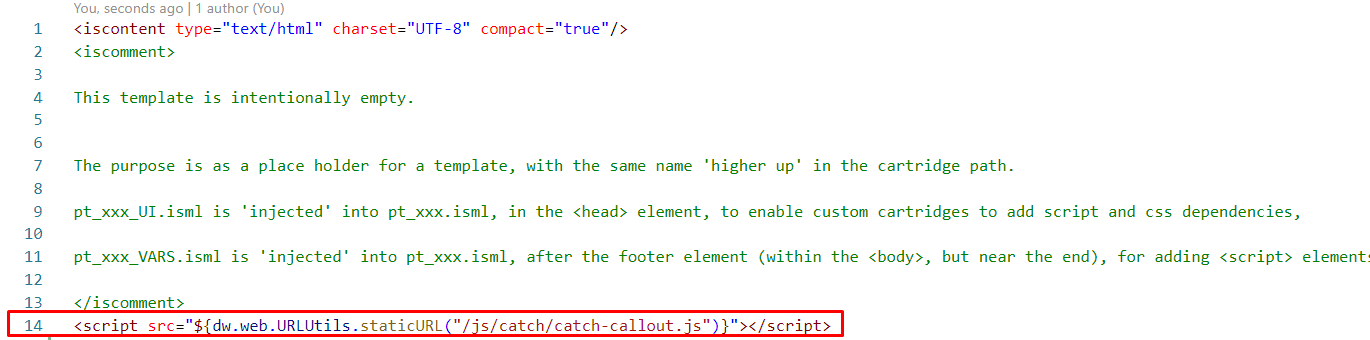
</div>



1. YOUR\_CARTRIDGE\_core/cartridge/templates/default/checkout/summary/pt\_cart\_VARS.isml

Put the following row to the end if the file:

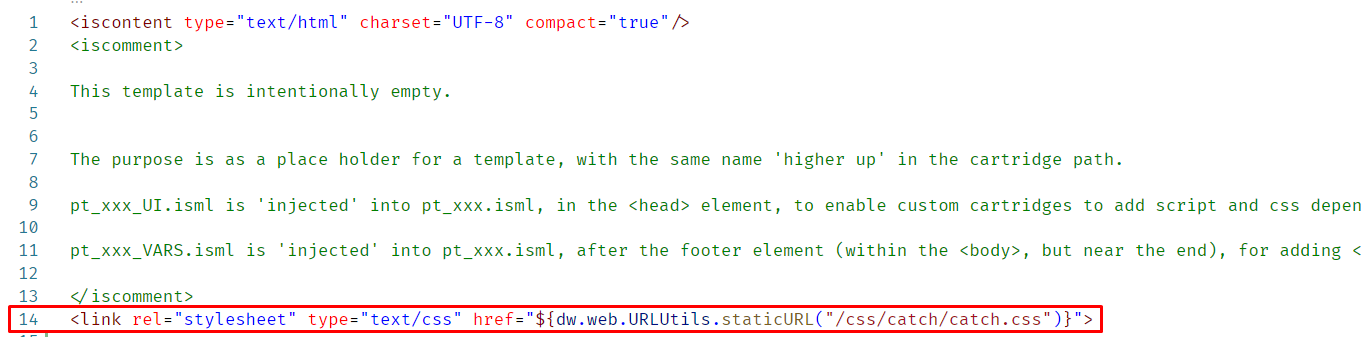
<script src="${dw.web.URLUtils.staticURL("/js/catch/catch-callout.js")}"></script>



1. YOUR\_CARTRIDGE\_core/cartridge/templates/default/checkout/summary/pt\_cart\_UI.isml

Add Catch styles to the end of the file:

<link rel="stylesheet" type="text/css" href="${dw.web.URLUtils.staticURL("/css/catch/catch.css")}">



1. YOUR\_CARTRIDGE\_core/cartridge/templates/default/checkout/summary/summary.isml

Put the following rows in line 222:

<iscomment>Start Catch Error view section</iscomment>

<div id="catch-error"></div>

<iscomment>End Catch Error view section</iscomment>



Wrap the <button> tag of the ‘PlaceOrder’ in ‘if’ statement in line 220:

<isif condition="${pdict.Basket.paymentInstrument.paymentMethod === 'Catch'}">

<isinclude template="catch/catchpay"/>

<iselse/>

**… ‘placeOrder' <button> tag**

</isif>



1. YOUR\_CARTRIDGE\_core/cartridge/templates/default/components/header/lheader.isml

Put the following row to the begin of the file:

<iscomment>Catch code start</iscomment>

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

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

<span class="catch-data" data-key="<isprint value="${catchHelper.getPreference('publicKey')}" encoding="htmldoublequote"/>"></span>

</isif>

<iscomment>Catch code end</iscomment>

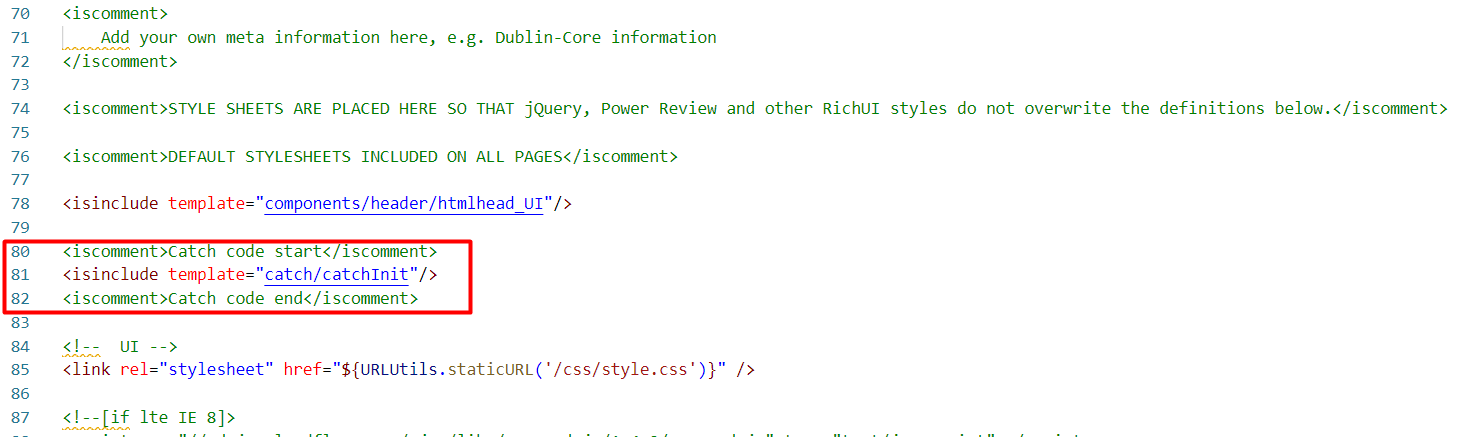




1. YOUR\_CARTRIDGE\_core/cartridge/templates/default/components/header/htmlhead.isml

Put the following row in line 80:

<isinclude template="catch/catchInit" />



1. YOUR\_CARTRIDGE\_core/cartridge/templates/default/components/header/htmlhead\_UI.isml

Put the following row in line 11:

<link rel="stylesheet" type="text/css" href="${dw.web.URLUtils.staticURL("/css/catch/catch.css")}">



1. YOUR\_CARTRIDGE\_core/cartridge/templates/default/components/order/orderdetails.isml

Put the following rows in line 222:

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

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

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

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

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

</isif>

</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'

&& !pdict.orderstatus}">

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

</isif>

1. YOUR\_CARTRIDGE\_core/cartridge/templates/default/components/order/orderdetotals.isml

Put the following rows in line 14:

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

<isset name="rewardsAdjustment" value="${LineItemCtnr.getPriceAdjustmentByPromotionID(constants.rewardsAdjustmentID)}" scope="page"/>

Put the following rows in line 34:

if (rewardsAdjustment) {

// exclude Catch rewards adjustment amount from order discount amount

orderDiscount = orderDiscount.add(rewardsAdjustment.getPrice());

}



Put the following rows in line 130:

<iscomment>Catch rewards</iscomment>

<isif condition="${rewardsAdjustment}">

<tr class="order-sales-tax">

<td>${Resource.msg('label.order.rewards','catch', null)}</td>

<td><isprint value="${rewardsAdjustment.getPrice()}"/></td>

</tr>

</isif>



1. YOUR\_CARTRIDGE\_core/cartridge/templates/default/components/order/orderdetotalsemail.isml

Put the following rows in line 14:

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

<isset name="rewardsAdjustment" value="${LineItemCtnr.getPriceAdjustmentByPromotionID(constants.rewardsAdjustmentID)}" scope="page"/>

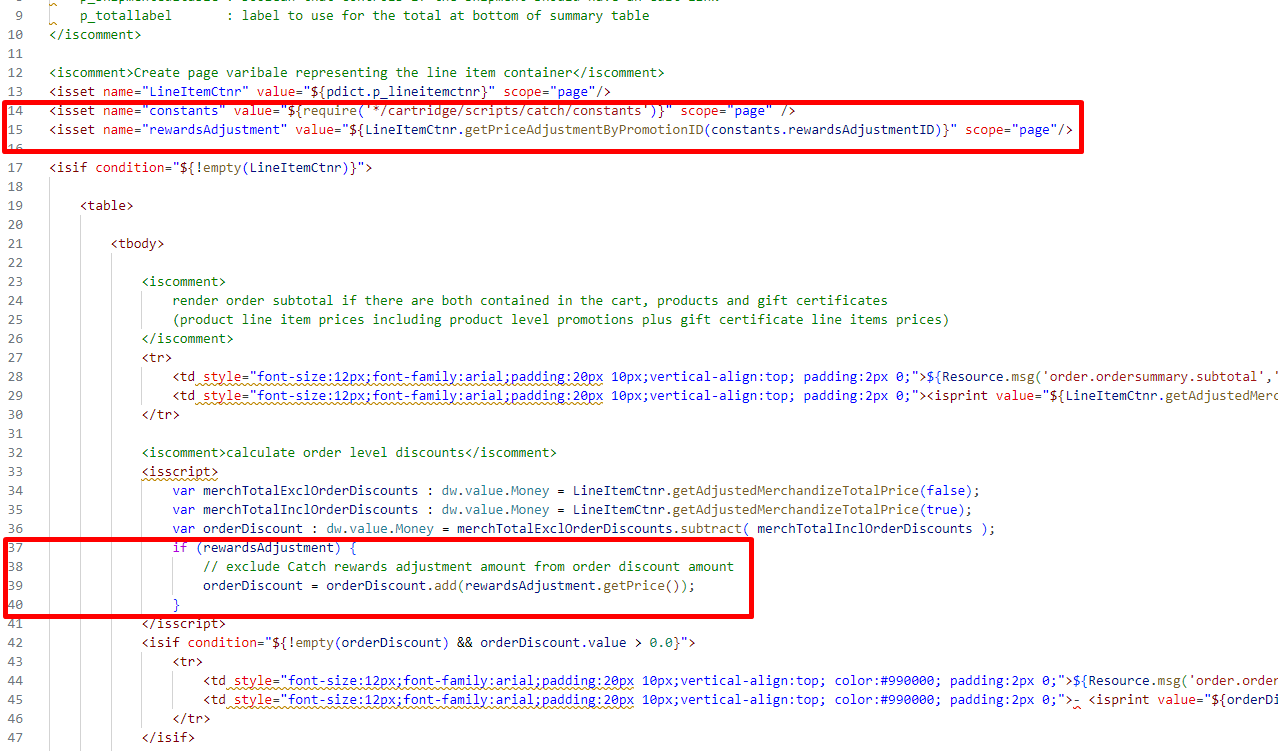
Put the following rows in line 37:

if (rewardsAdjustment) {

// exclude Catch rewards adjustment amount from order discount amount

orderDiscount = orderDiscount.add(rewardsAdjustment.getPrice());

}



Put the following rows in line 130:

<iscomment>Catch rewards</iscomment>

<isif condition="${rewardsAdjustment}">

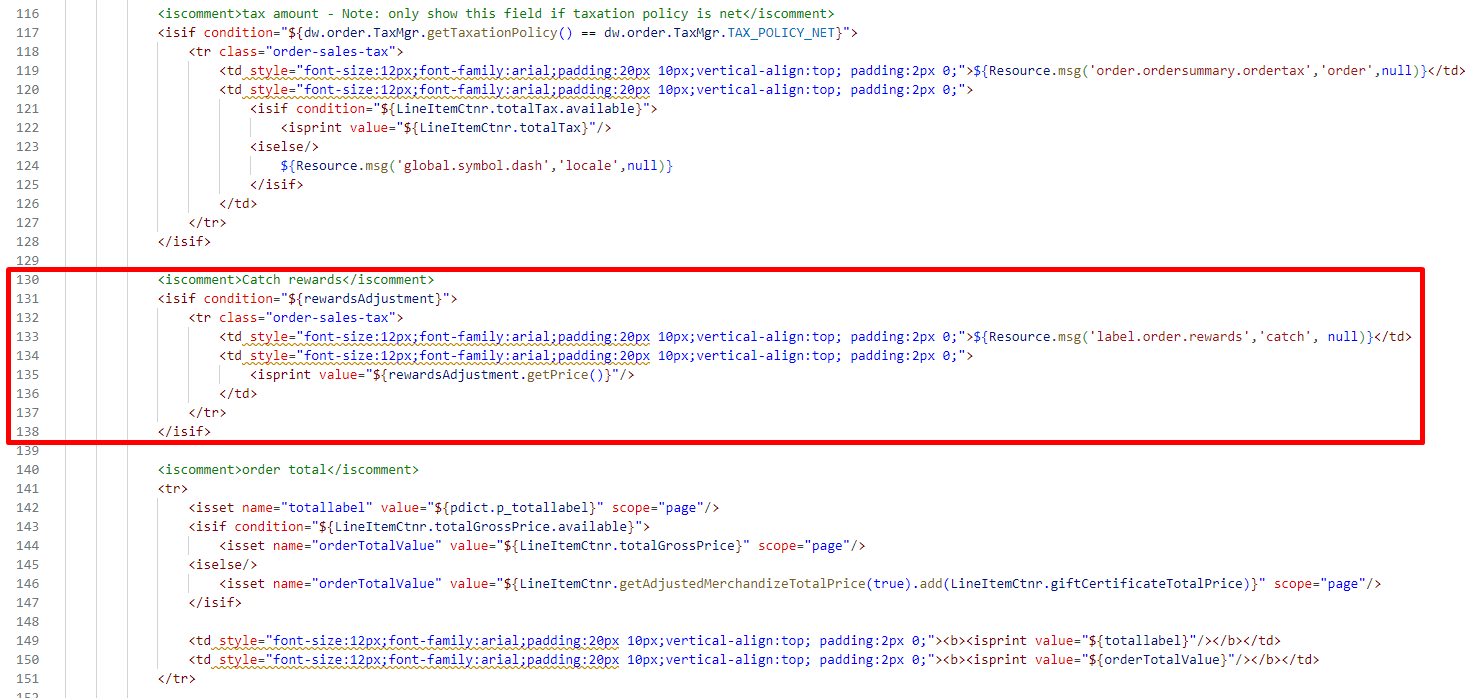
<tr class="order-sales-tax">

<td>${Resource.msg('label.order.rewards','catch', null)}</td>

<td><isprint value="${rewardsAdjustment.getPrice()}"/></td>

</tr>

</isif>



1. YOUR\_CARTRIDGE\_core/cartridge/templates/default/email/orderconfirmation.isml

Put the following rows in line 44:

<iscomment>Catch Code Start</iscomment>

<isif condition="${!pdict.Order.getPaymentInstruments('Catch').empty}">

<tr>

<th style="background:#cccccc;padding:5px 20px;font-size:12px;font-family:arial;text-align:left;">${Resource.msg('msg.confirmation.catch.mail','catch',null)}</th>

</tr>

</isif>

<iscomment>Catch Code End</iscomment>



1. YOUR\_CARTRIDGE\_core/cartridge/templates/default/product/productcontent.isml

Put the following row in line78:

<iscomment> Catch Code </iscomment>

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

<iscomment> Catch Code </iscomment>



1. YOUR\_CARTRIDGE\_core/cartridge/templates/default/product/producttopcontentPS.isml

Put the following rows in line 122:

<iscomment> Catch Code </iscomment>

<isif condition="${pdict.isBundle}">

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

</isif>

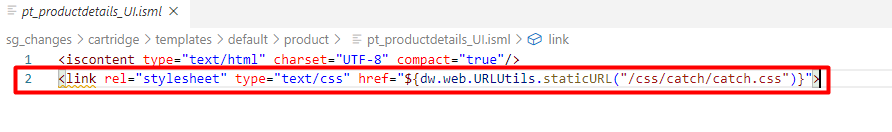
<iscomment> Catch Code </iscomment>



1. YOUR\_CARTRIDGE\_core/cartridge/templates/default/product/pt\_productdetails\_UI.isml

Put the following row to the end if the file:

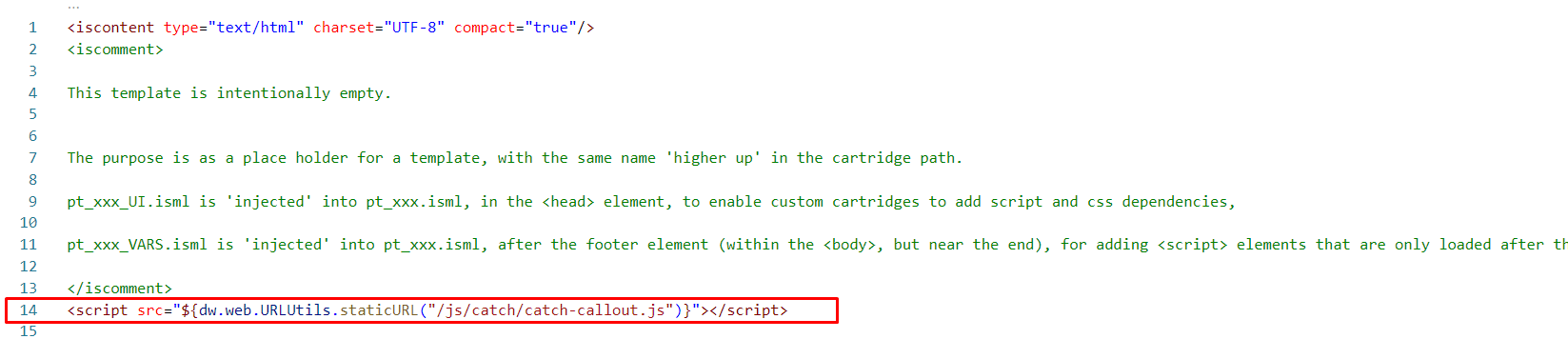
<link rel="stylesheet" type="text/css" href="${dw.web.URLUtils.staticURL("/css/catch/catch.css")}">



1. YOUR\_CARTRIDGE\_core/cartridge/templates/default/product/pt\_productdetails\_VARS.isml

Put the following row to the end if the file:

<script src="${dw.web.URLUtils.staticURL("/js/catch/catch-callout.js")}"></script>



1. YOUR\_CARTRIDGE\_core/cartridge/templates/default/product/components/productsetproduct.isml

Put the following rows in line 24:

<iscomment> Catch Code </iscomment>

<isif condition="${pdict.isSet}">

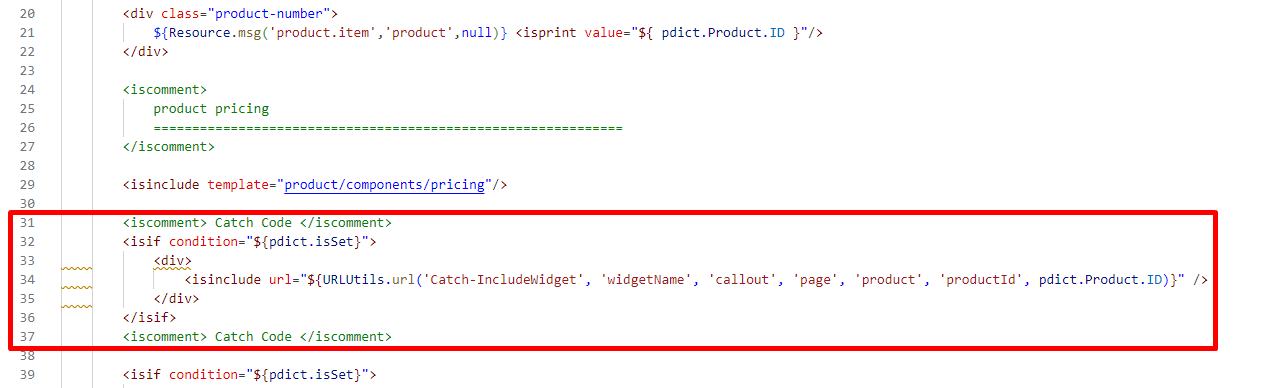
<div>

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

</div>

</isif>

<iscomment> Catch Code </iscomment>



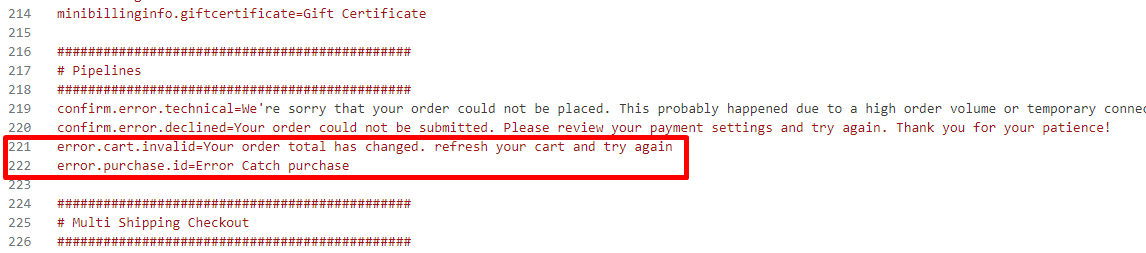
### Perform changes in resources

1. YOUR\_CARTRIDGE\_core/cartridge/templates/resources/checkout.properties

Put the following row in line 221:

error.cart.invalid=Your order total has changed. refresh your cart and try again

error.purchase.id=Error Catch purchase



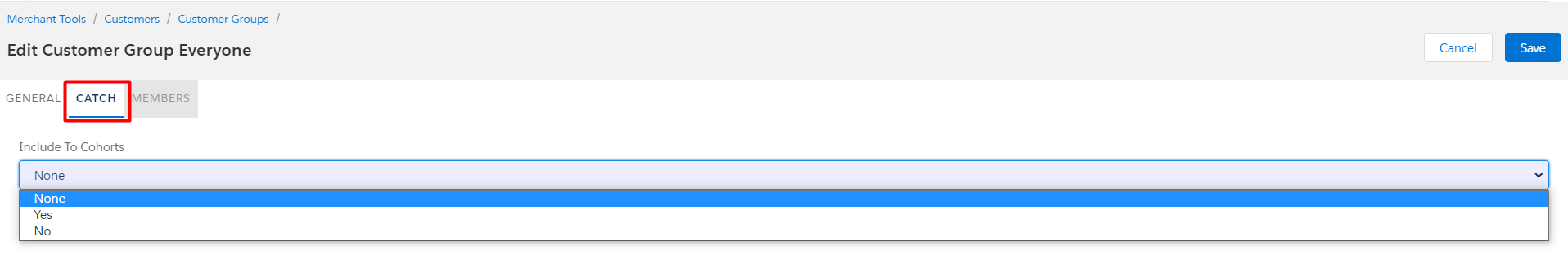
## Configuration

### Customer Groups

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

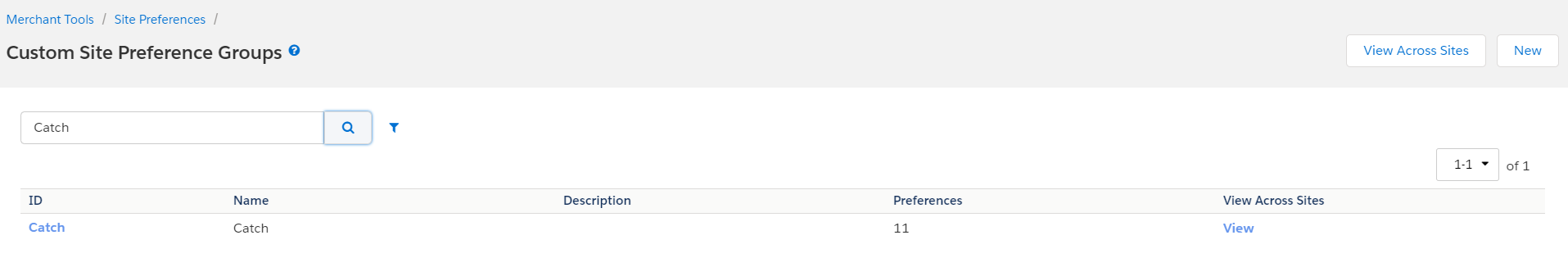
If the customer belongs to a particular Customer Group and you want to send this data to Catch in order to tailor Catch to each consumer’s context set “Include To Cohorts” attribute value to “Yes”.

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

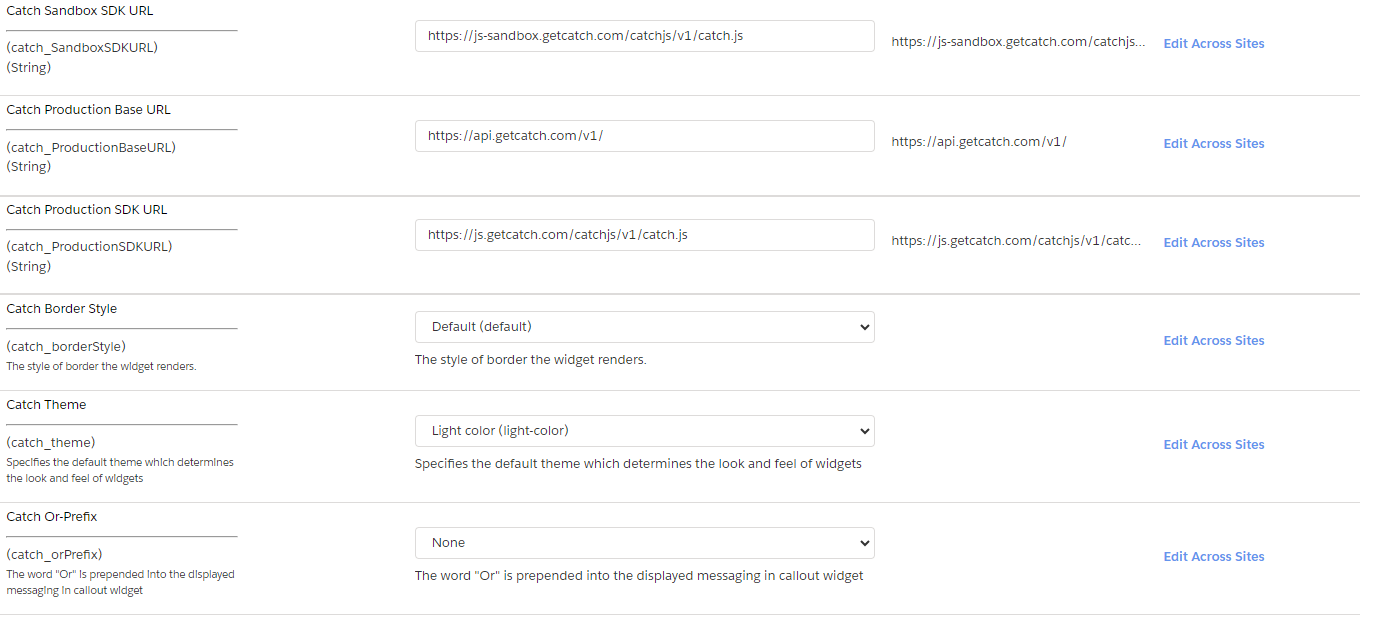


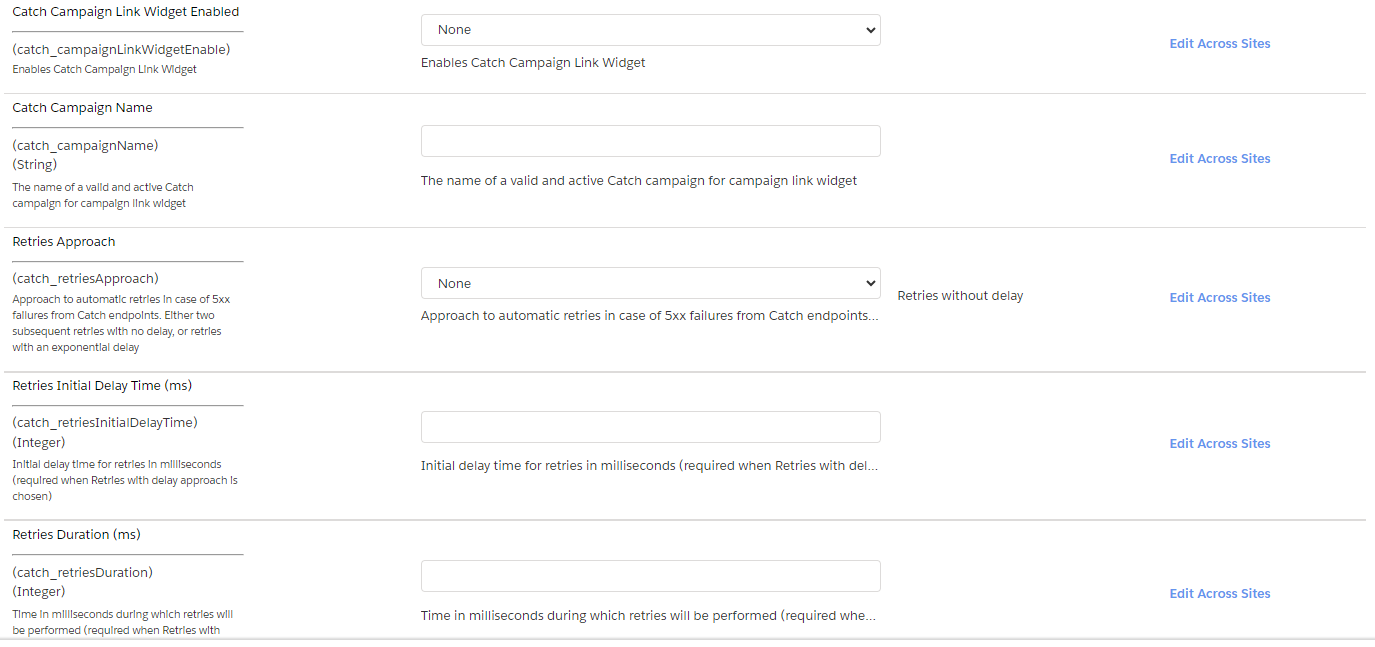
### Custom Site Preferences

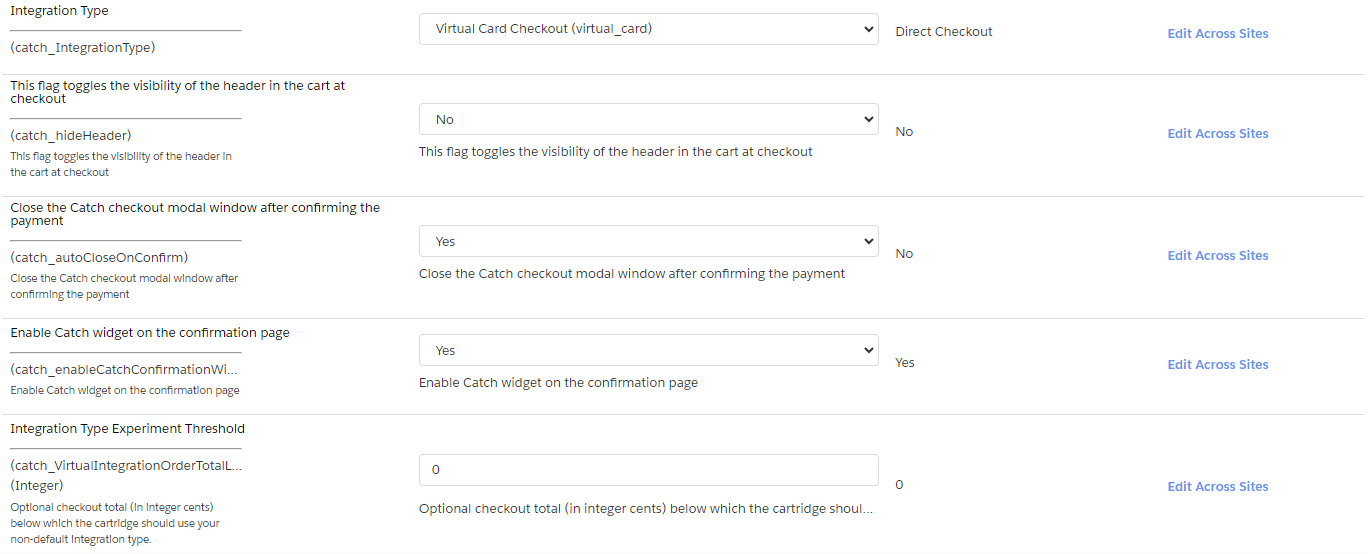
After import is successfully done, you should see Site Preferences in BM > Merchant Tools > Site Preferences > Custom Preferences.



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 | 2022-01-01 | 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 |
|  |  |  |