Pennies

Integration Guide

**Document Revision History**

| **Revision** | **Author** | **Date** | **Status and Description** |
| --- | --- | --- | --- |
| 16.1.0 | Lakshmi Menon | 03-06-2016 | Draft version internally reviewed |
| 16.1.1 | Shekhar Nair | 11-08-2016 | Changes after pennies certification |
| 17.1.0 | Lakshmi Menon | 14-09-2017 | Added section Job Details |
| 17.1.1 | Lakshmi Menon | 05-10-2017 | Added details on configuring new hooks order cart calculation |
| 18.1.0 | Rekha Mohan Lal | 11-09-2018 | Deprecated ServiceRegistry is changed to LocalServiceRegistry for API calls |
| 18.1.1 | Simi George | 15-02-2019 | Added missing integration details in order summary/order history and order review pages |
| 20.1.0 | Soumya Raj M | 12-02-2020 | Pennies donation banner to be displayed only if merchant Id and API key is not empty.  Code change in  Controller: COShipping.js  Function: updateShippingMethodList |
| 21.1.3 | Wirebox | 10-05-2022 | Updated compatibility 22.6 |

Contents

[Overview 4](#_Toc44402753)

[Features 4](#_Toc44402754)

[Limitations, Constraints 4](#_Toc44402755)

[Compatibility 4](#_Toc44402756)

[Summary 5](#_Toc44402757)

[Pennies Site Preferences 5](#_Toc44402758)

[Business Manager Configurations 5](#_Toc44402759)

[Storefront Cartridge Changes – General 6](#_Toc44402760)

[Storefront cartridge changes – JS/ISML 6](#_Toc44402761)

[Storefront cartridge changes 16](#_Toc44402762)

[Custom Tags 22](#_Toc44402763)

[Job Details 22](#_Toc44402764)

[Operations, Maintenance 23](#_Toc44402765)

# **Overview**

The pennies cartridge implementation provides the feature of integrating the Pennies Charity with other clients. The donations can be made to pennies through this cartridge.

# **Features**

* Ability to toggle the integration on/off on the storefront.
* Ability to configure the merchant id and access token through site preferences.
* Pennies donation to be displayed only if session currency is GBP.
* Pennies donation banner to be displayed only if merchant Id and API key is not empty
* Donation amount to be added to basket/order in such a way that it’s not included in the Demandware Revenue model.
* Donation amount to be calculated based on basket total
* Remove donation from basket, if basket total changes after adding the donation to basket.
* Remove donation from basket, if payment fails.

# **Limitations, Constraints**

Pennies cartridge currently supports only UK currencies. So, pennies donation can be enabled only for the sites which has UK currencies enabled.

# **Compatibility**

Available since Commerce Cloud Platform Release 18.10, Site Genesis 103.0.0.

The cartridge is available for installations on storefronts that support both SFRA and Site Genesis implementations.

# **Summary**

|  |  |
| --- | --- |
| **Name** | **Purpose** |
| cartridges | Contains Pennies integration cartridges |
| cartridges\int\_pennies\_controllers | This cartridge should only be used by clients who have their storefront developed using controllers. |
| cartridges\int\_pennies\_core | This cartridge contains the code for the storefront changes, Integration framework and Service framework (Pennies API call). |
| site-template | Contains custom attributes and settings required for the integration |
| documentation | Contains this document “Pennies Integration Document” |

# **Pennies Site Preferences**

Custom preferences have been added to a new attribute group “**Pennies**”.

|  |  |
| --- | --- |
| **Name** | **Purpose** |
| Pennies Donation Integration Enabled | Indicates if the pennies integration has been enabled on the site.  The client can switch off/on the pennies integration feature. |
| Pennies Access Token | Access token provided by Pennies |
| Pennies Merchant ID | Merchant ID Provided by Pennies |

# **Business Manager Configurations**

1. Please import and include the following cartridges in the site cartridge path as well as business manager cartridge path

|  |  |
| --- | --- |
| Site Cartridge Path | *int\_pennies\_controllers:int\_pennies\_core* |
| BM Cartridge Path | *int\_pennies\_core* |

1. Through site import/export please import the files in **site\_template**

Note: *Please note that the cartridges need to be imported first before importing the site\_template as the Job Schedules import will fail if the cartridge int\_pennies\_core isn’t uploaded to BM.*

1. Configure the site preference values under

**Merchant Tools > Site Preferences > Custom Site Preferences > Pennies**

1. Configure the job configurations for the following jobs as per need

**Generate Pennies Donation Report**

**Post Pennies Donation Details**

1. Navigate to [**Administration**](https://tryzens28-alliance-prtnr-eu03-dw.demandware.net/on/demandware.store/Sites-Site/default/ViewApplication-ExpandMenuGroup?MenuGroupID=AdministrationMenu&OverviewPage=SiteNavigationBar-ShowAdministrationOverview)**>**[**Operations**](https://tryzens28-alliance-prtnr-eu03-dw.demandware.net/on/demandware.store/Sites-Site/default/SiteNavigationBar-ShowMenuitemOverview?CurrentMenuItemId=operations)**>  Services** and confirm if the below mentioned services are configured. Please note do not change the service names.

[**pennies.donation.http.service**](https://tryzens28-alliance-prtnr-eu03-dw.demandware.net/on/demandware.store/Sites-Site/default/Service-DisplayDetails?ServiceUUID=86672c2f2a05c7fda4b544d368)

[**pennies.calculation.http.service**](https://tryzens28-alliance-prtnr-eu03-dw.demandware.net/on/demandware.store/Sites-Site/default/Service-DisplayDetails?ServiceUUID=b86e142fda384c2b8b22ede8d5)

# **Storefront Cartridge Changes – General**

**Hooks.json changes**

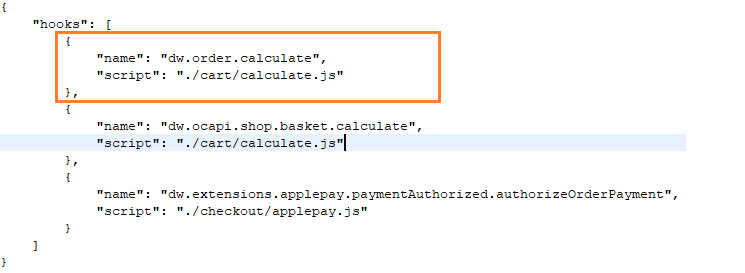
1. Open hooks.json in the storefront core cartridge and add the following entry if not already present

{

"name": "dw.order.calculate",

"script": "./cart/calculate.js"

}



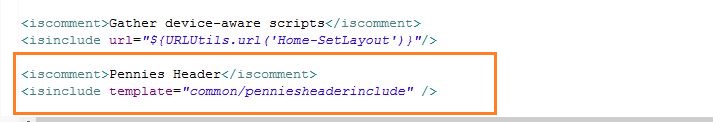
# **Storefront cartridge changes – JS/ISML**

**Header changes**

1. Open **htmlhead.isml** in the storefront cartridge.
2. Add the following lines of code to the end of the file

<iscomment>Pennies Header</iscomment>

<isinclude template=*"common/penniesheaderinclude"* />

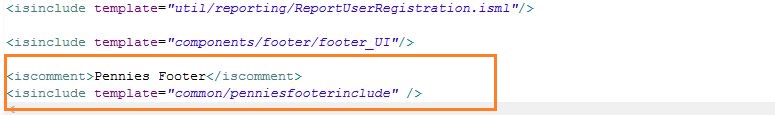


**Footer Changes**

1. Open **footer.isml** in the storefront cartridge.
2. Add the following lines of code to the end of the file

<iscomment>Pennies Footer</iscomment>

<isinclude template="common/penniesfooterinclude" />



**Display Pennies Inform Banner**

1. Open the isml where you would like to display the pennies inform panel banner
2. Add the following code at the beginning of the isml

<isinclude template="util/penniesmodules" />

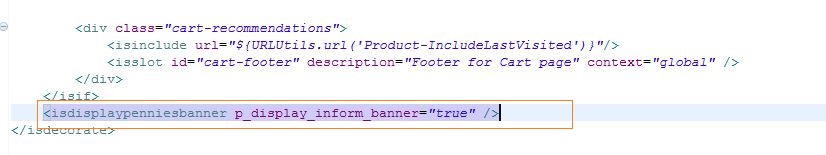
Below is the screenshot we have added in the cart.isml.



1. Add the following line of code where you would like to display the banner in the same isml

<isdisplaypenniesbanner p\_display\_inform\_banner="true" />

Below is the screenshot we have added in the cart.isml.

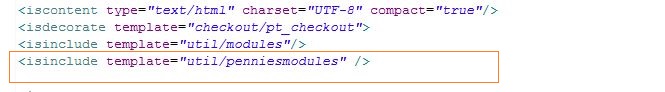


**Display Pennies Invite To Donate/Thank You Banner**

1. The Pennies Invite To Donate banner gets replaced with the Pennies Thank You banner and hence there is just 1 line of code that needs to be added.
2. Open the isml where you would like to display the Pennies Invite To Donate/Thank you banner
3. Add the following code at the beginning of the isml

<isinclude template="util/penniesmodules" />

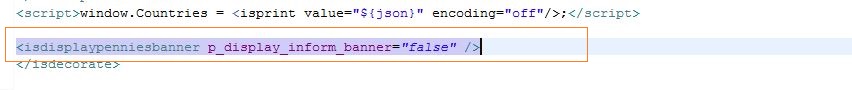
Below is the screenshot we have added for singleshipping.isml.



1. Add the following line of code where you would like to display the banner in the same isml

<isdisplaypenniesbanner p\_display\_inform\_banner="false" />

Below is the screenshot we have added for sigleshipping.isml.



**Order Summary/Order Confirmation/Order Confirmation Email Changes – Display Shipping Amount**

1. The below changes are based on Site Genesis templates and should be modified according to the client’s implementation of Order Summary.
2. When customer adds a donation to the basket/order, the shipping total gets incremented by the donation amount.
3. To display the shipping total without the donation amount in order summary, the following lines of code should be added at the beginning of the isml

<isinclude template=*"util/penniesmodules"* />

1. To display the shipping total without the donation amount in order summary, the following lines of code can be used where the shipping total is displayed in checkout order summary.

We have modified the ordertotals.isml and ordertotalsemail.isml with the corresponding changes.

**ordertotals.isml**

We have two changes in this file

**Change1:**

<td>

<isif condition=*"${LineItemCtnr.shippingTotalPrice.available}"*>

<isprint value=*"${Shipment.shippingTotalPrice}"*/>

<iselse/>

${Resource.msg('global.symbol.dash','locale',null)}

</isif>

</td>

Replace the above code with the below one.

<td>

<isif condition=*"${ dw.system.Site.getCurrent().getCustomPreferenceValue('penniesDonationIntegrationEnabled') }"* >

<isdisplayshippingpriceexcldonation p\_lineitemctnr=*"${LineItemCtnr}"* />

<iselseif condition=*"${LineItemCtnr.shippingTotalPrice.available}"* >

<isprint value=*"${Shipment.shippingTotalPrice}"*/>

<iselse/>

${Resource.msg('global.symbol.dash','locale',null)}

</isif>

</td>



**Change2:**

<td>

<isif condition=*"${LineItemCtnr.shippingTotalPrice.available}"*>

<isprint value=*"${LineItemCtnr.shippingTotalPrice}"*/>

<iselse/>

${Resource.msg('global.symbol.dash','locale',null)}

</isif>

</td>

Replace the above code with the below one

<td>

<isif condition=*"${ dw.system.Site.getCurrent().getCustomPreferenceValue('penniesDonationIntegrationEnabled') }"* >

<isdisplayshippingpriceexcldonation p\_lineitemctnr=*"${LineItemCtnr}"* />

<iselseif condition=*"${LineItemCtnr.shippingTotalPrice.available}"* >

<isprint value=*"${LineItemCtnr.shippingTotalPrice}"*/>

<iselse/>

${Resource.msg('global.symbol.dash','locale',null)}

</isif>

</td>



**ordertotalsemail.isml**

We have two changes in this file

**Change1:**

<td style="font-size:*12px*;font-family:*arial*;padding:*20px 10px*;vertical-align:*top*; padding:*2px 0*;">

<isif condition=*"${LineItemCtnr.shippingTotalPrice.available}"*>

<isprint value=*"${Shipment.shippingTotalPrice}"*/>

<iselse/>

${Resource.msg('global.symbol.dash','locale',null)}

</isif>

</td>

Replace the above code with the below one

<td style="font-size:*12px*;font-family:*arial*;padding:*20px 10px*;vertical-align:*top*; padding:*2px 0*;">

<isif condition=*"${ dw.system.Site.getCurrent().getCustomPreferenceValue('penniesDonationIntegrationEnabled') }"* >

<isdisplayshippingpriceexcldonation p\_lineitemctnr=*"${LineItemCtnr}"* />

<iselseif condition=*"${LineItemCtnr.shippingTotalPrice.available}"* >

<isprint value=*"${Shipment.shippingTotalPrice}"*/>

<iselse/>

${Resource.msg('global.symbol.dash','locale',null)}

</isif>

</td>



**Change2:**

<td style="font-size:*12px*;font-family:*arial*;padding:*20px 10px*;vertical-align:*top*; padding:*2px 0*;">

<isif condition=*"${LineItemCtnr.shippingTotalPrice.available}"*>

<isprint value=*"${LineItemCtnr.shippingTotalPrice}"*/>

<iselse/>

${Resource.msg('global.symbol.dash','locale',null)}

</isif>

</td>

Replace the above code with the below one.

<td style="font-size:*12px*;font-family:*arial*;padding:*20px 10px*;vertical-align:*top*; padding:*2px 0*;">

<isif condition=*"${ dw.system.Site.getCurrent().getCustomPreferenceValue('penniesDonationIntegrationEnabled') }"* >

<isdisplayshippingpriceexcldonation p\_lineitemctnr=*"${LineItemCtnr}"* />

<iselseif condition=*"${LineItemCtnr.shippingTotalPrice.available}"* >

<isprint value=*"${LineItemCtnr.shippingTotalPrice}"*/>

<iselse/>

${Resource.msg('global.symbol.dash','locale',null)}

</isif>

</td>



Pass the basket object as the parameter ***LineItemCtnr*** if the order has not been placed and pass Order object as the parameter ***LineItemCtnr*** if the order has been placed.

**Order Summary/Order Confirmation/Order Confirmation Email Changes – Display Donation Amount**

1. The below changes are based on Site Genesis templates and should be modified according to the client’s implementation of Order Summary.
2. When customer adds a donation to the basket/order, the donation amount needs to be displayed as a different section in the order summary.
3. To display the donation amount in order summary, the following lines of code should be added at the beginning of the isml (minisummary.isml)

<isinclude template=*"util/penniesmodules"* />

1. To display the donation amount in order confirmation page, the following lines of code should be added at the beginning of the isml (confirmation.isml)

<isinclude template=*"util/penniesmodules"* />

1. To display the donation amount in order review page, the following lines of code should be added at the beginning of the isml (summary.isml)

<isinclude template=*"util/penniesmodules"* />

1. To display the donation amount in order details page of order history, the following lines of code should be added at the beginning of the isml (orderhistory/orderdetails.isml)

<isinclude template=*"util/penniesmodules"* />

1. To display the donation amount in order summary, the following lines of code can be used where the checkout order summary is.

We have placed the following code in ordertotals.isml and ordertotalsemail.isml.

<iscomment>Pennies donation details</iscomment>

<isif condition=*"${ dw.system.Site.getCurrent().getCustomPreferenceValue('penniesDonationIntegrationEnabled') }"* >

<isdisplaypenniesdonationamount

p\_lineitemctnr=*"${LineItemCtnr}"* />

</isif>

**ordertotals.isml**



**ordertotalsemail.isml**

****

Pass the basket object as the parameter ***p\_lineitemctnr*** if the order has not been placed and pass Order object as the parameter ***p\_lineitemctnr*** if the order has been placed.

1. Clients can modify ***displaypenniesdonationamount.isml*** which is the rendering template for the tag ***<isdisplaypenniesdonationamount>***  as per their implementation of order summary.

**Order Summary Changes – Refresh order summary after donation is added/removed**

1. The below changes are based on Site Genesis templates and should be modified according to the client’s implementation of Order Summary.
2. When customer adds/removes a donation to/from the basket and ajax call is fired. On completion of this ajax call the order summary section needs to be updated.
3. The order summary is refreshed using the function **updateSummary()** in **int\_pennies\_core/cartridge/static/default/js/libPennies.js** . The html element ids specified in this function must be updated as per client’s implementation of the order summary.

**Refresh Pennies Banner – After Basket total changes in Basket/Shipping/Billing pages**

1. The Pennies donation amount if added to basket, must be removed from the basket if there is a change in the basket total. Scenarios where basket total changes without a page refresh for e.g. change in shipping method, the pennies thank you banner must be replaced with pennies invite to donate banner.
2. To achieve this within storefront cartridge navigate to

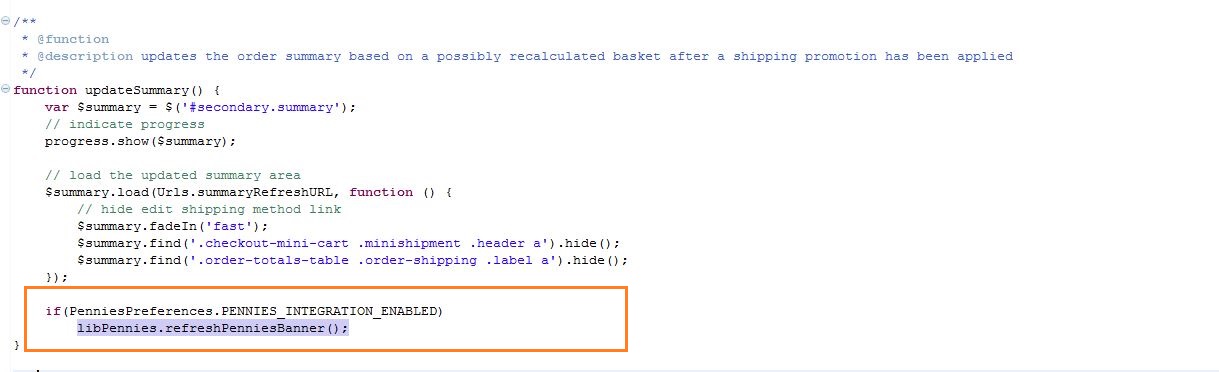
* For clients using gulp version of app.js navigate to **cartridge/js/pages/checkout/shipping.js**
* For clients using older version of app.js navigate to

**cartridge/static/default/js/app.js**

add the following lines of code at the end of the function **updateSummary()**

**if**(PenniesPreferences.PENNIES\_INTEGRATION\_ENABLED)

libPennies.refreshPenniesBanner();



# **Storefront cartridge changes**

**Shipping Method Calculation Related Changes**

1. In SiteGenesis, when the list of shipping methods are generated to be displayed on the shipping page, if the donation has been added to basket the donation amount gets added to each shipping method cost that’s displayed on the shipping page. The below mentioned code changes fix this issue.
2. The below mentioned changes are to be done in

*Controller : COShipping.js*

*Function : updateShippingMethodList()*

1. Add the highlighted code as shown below. The below shown code is based on the Site Genesis controller cartridge.

*function updateShippingMethodList() {*

*var i, cart, address, applicableShippingMethods, shippingCosts, currentShippingMethod, method, TransientAddress;*

*TransientAddress = app.getModel('TransientAddress');*

*cart = app.getModel('Cart').get();*

*if (cart) {*

*address = new TransientAddress();*

*address.countryCode = request.httpParameterMap.countryCode.stringValue;*

*address.stateCode = request.httpParameterMap.stateCode.stringValue;*

*address.postalCode = request.httpParameterMap.postalCode.stringValue;*

*address.city = request.httpParameterMap.city.stringValue;*

*address.address1 = request.httpParameterMap.address1.stringValue;*

*address.address2 = request.httpParameterMap.address2.stringValue;*

*applicableShippingMethods = cart.getApplicableShippingMethods(address);*

*shippingCosts = new HashMap();*

*currentShippingMethod = cart.getDefaultShipment().getShippingMethod() || ShippingMgr.getDefaultShippingMethod();*

*//Pennies Changes : START - Skipping pennies monitor basket when looping through applicable shipping methods*

*session.privacy.skipPenniesMonitorBasket= true;*

*//Pennies Changes : END - Skipping pennies monitor basket when looping through applicable shipping methods*

*// Transaction controls are for fine tuning the performance of the data base interactions when calculating shipping methods*

*Transaction.begin();*

*for (i = 0; i < applicableShippingMethods.length; i += 1) {*

*method = applicableShippingMethods[i];*

*cart.updateShipmentShippingMethod(cart.getDefaultShipment().getID(), method.getID(), method, applicableShippingMethods);*

*cart.calculate();*

*shippingCosts.put(method.getID(), cart.preCalculateShipping(method));*

*}*

*Transaction.rollback();*

*//Pennies Changes : START - Resetting skipMonitorBasket value in request to calculate value*

*session.privacy.skipPenniesMonitorBasket= false;*

*//Pennies Changes : END - Resetting skipMonitorBasket value in request to calculate value*

*Transaction.wrap(function () {*

*cart.updateShipmentShippingMethod(cart.getDefaultShipment().getID(), currentShippingMethod.getID(), currentShippingMethod, applicableShippingMethods);*

*cart.calculate();*

*});*

*session.forms.singleshipping.shippingAddress.shippingMethodID.value = cart.getDefaultShipment().getShippingMethodID();*

*//Pennies Changes : START - Update the shipping costs to remove the donation amount*

*var updatedShippingCosts = require("int\_pennies\_core/cartridge/scripts/common/libPennies").updateShippingCosts(shippingCosts, cart.object);*

*shippingCosts = updatedShippingCosts.ShippingCosts;*

*//Pennies Changes : END - Update the shipping costs to remove the donation amount*

*app.getView({*

*Basket: cart.object,*

*ApplicableShippingMethods: applicableShippingMethods,*

*ShippingCosts: shippingCosts*

*}).render('checkout/shipping/shippingmethods');*

*} else {*

*app.getController('Cart').Show();*

*}*

*}*

**Remove pennies donation from basket when payment fails**

1. The pennies donation needs to be removed from the basket if the payment fails.
2. This integration varies from client to client and needs to be integrated as per the site checkout design.
3. The below code needs to be added in place where the basket object is available

*// Pennies Changes : START - Remove donation from basket*

*if ('error' in result && result.error) {*

*dw.system.Transaction.wrap(function() {*

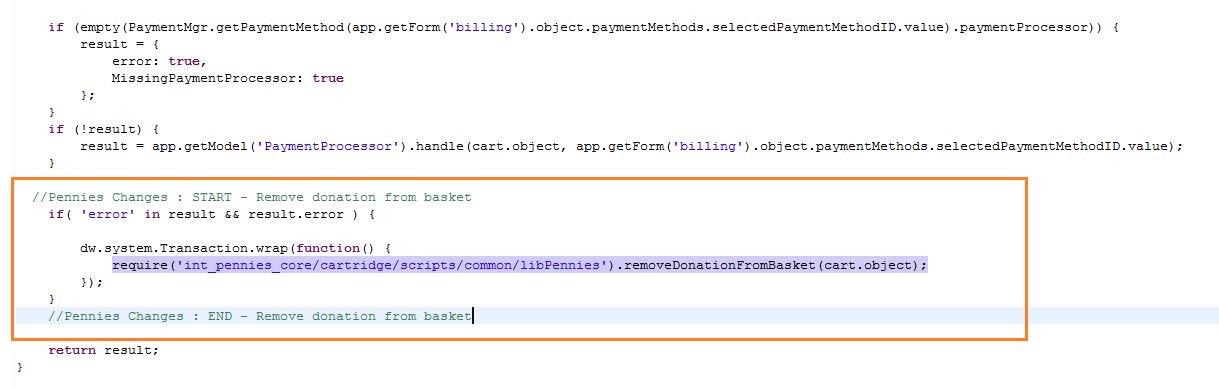
*require('int\_pennies\_core/cartridge/scripts/common/libPennies').removeDonationFromBasket(cart.object);*

*});*

*}*

*// Pennies Changes : END - Remove donation from basket*

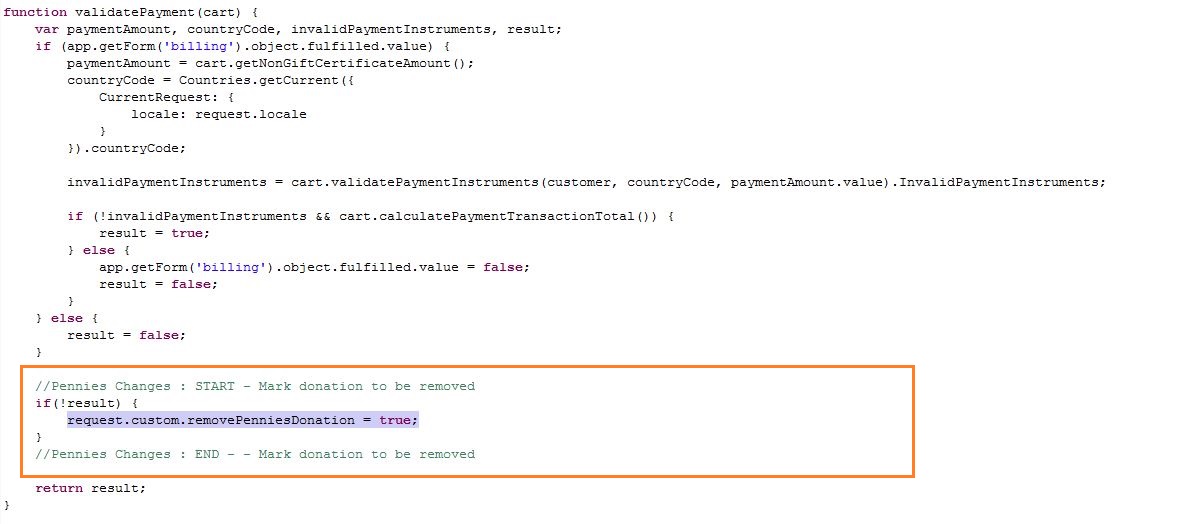
Place the above code in handlePaymentSelection() function in COBilling.js

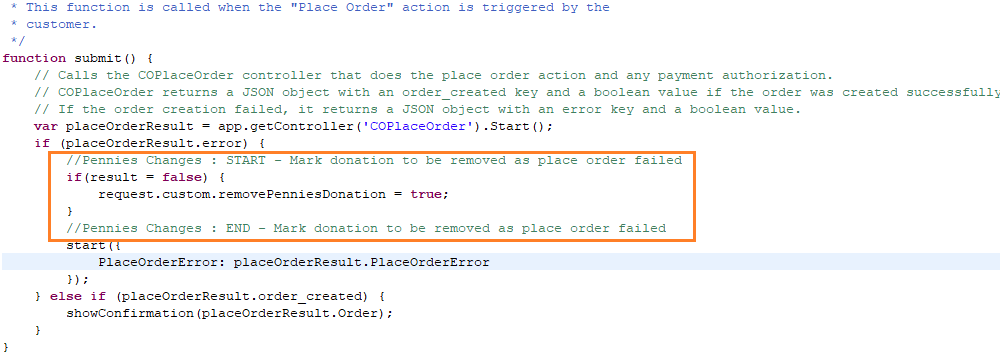


1. The below code needs to be added at places where the Basket object is not available in pipeline dictionary, for e.g in PlaceOrder-Start after CreateOrder pipelet, the basket object gets removed from pipeline dictionary and if the payment fails after that, then the below code needs to be added.

*request.custom.removePenniesDonation = true;*

We have added the above code in validatePayment() function in COBilling.js and submit() function in COSummary.js





1. Below are some of the controllers where the above changes can be added when payment validation fails
   1. COBilling.js : function validatePayment(cart)
   2. COBilling.js : function handlePaymentSelection(cart)
   3. COSummary.js : function submit() [*after COPlaceOrder-Start is called and*

*payment fails]*

* 1. Payment authorisation controllers

**Updating the donation details on the Order on Order Confirmation**

1. When an order is confirmed we need to update the following donation related details on the order

* Pennies hash key
* Pennies charity details
* Pennies donation status

1. The below mentioned changes should be made in

*Controller : COSummary.js*

*Function : showConfirmation(order)*

Add the highlighted code to the function

*function showConfirmation(order) {*

*//Pennies Changes : START - Update pennies donation details on order*

*var libPennies = require('int\_pennies\_core/cartridge/scripts/common/libPennies');*

*libPennies.updatePenniesDonationDetailsOnOrder(order);*

*//Pennies Changes : END - Update pennies donation details on order*

*if (!customer.authenticated) {*

*// Initializes the account creation form for guest checkouts by populating the first and last name with the*

*// used billing address.*

*var customerForm = app.getForm('profile.customer');*

*customerForm.setValue('firstname', order.billingAddress.firstName);*

*customerForm.setValue('lastname', order.billingAddress.lastName);*

*customerForm.setValue('email', order.customerEmail);*

*}*

*app.getForm('profile.login.passwordconfirm').clear();*

*app.getForm('profile.login.password').clear();*

*var pageMeta = require('~/cartridge/scripts/meta');*

*pageMeta.update({pageTitle: Resource.msg('confirmation.meta.pagetitle', 'checkout', 'SiteGenesis Checkout Confirmation')});*

*app.getView({*

*Order: order,*

*ContinueURL: URLUtils.https('Account-RegistrationForm') // needed by registration form after anonymous checkouts*

*}).render('checkout/confirmation/confirmation');*

*}*

**Updating donation details on order when the payment is captured**

1. When the payment for the order is captured, the pennies donation status should be set as ‘Ready for export’ if a donation has been added to the order. The donation details are sent to Pennies when an order donation status has been marked as ‘Ready for export’.
2. Insert the below code right after the payment has been captured successfully

*//Pennies Changes : START - Mark donation ready for export*

*libPennies.markDonationReadyForExport(order);*

*//Pennies Changes : END - Mark donation ready for export*

Below is an example showing changes in the showConfirmation(order) function inCOSummary.js . This must be placed appropriately in the code when a payment is captured.



**Monitor Basket for Basket Total changes after donation has been added to basket**

1. After a donation has been added to the basket, if the basket total changes then the donation must be removed from the basket.
2. The following code must be added to

*Controller : CartModel.js*

*Function : calculate()*

Add the highlighted code as shown below

calculate: function () {

dw.system.HookMgr.callHook('dw.ocapi.shop.basket.calculate', 'calculate', this.object);

//Pennies Changes : START - Check if basket total has changed since last donation

var libPennies = require('int\_pennies\_core/cartridge/scripts/common/libPennies');

libPennies.monitorBasket(this.object);

//Pennies Changes : END - Check if basket total has changed since last donation

}

# **Custom Tags**

The custom tags have been configured in int\_pennies\_core/cartridge/templates/default/util/penniesmodules.isml

**<isdisplaypenniesbanner>**

This custom tag displays the appropriate pennies donation banner based on the values in the session and the basket total.

Attributes

p\_display\_inform\_banner : Set to true if only inform panel banner should be displayed.

**<isdisplayshippingpriceexcldonation>**

This custom tag will display the actual shipping price excluding the pennies donation price. This tag can be used in the order confirmation page.

Attributes

p\_lineitemctnr **:** Theline item container(basket/order) should be passed to this template.

**<isdisplaypenniesdonationamount >**

This custom tag displays the pennies donation amount if it’s added to the basket/order. This tag can be used in the basket/order summary section.

Attributes

p\_lineitemctnr **:** Theline item container(basket/order) should be passed to this template.

# **Job Details**

The cartridge provides new job schedules to execute the Pennies related jobs.

**Generate Pennies Donation Report**

This job generates the pennies donation report. The report contains only orders whose donation details have been exported to Pennies.

A script-module-step is used to execute the job schedule.

Script module : int\_pennies\_core/cartridge/scripts/job/PenniesJobAdapter.js

Configurable parameters on step

|  |  |  |  |
| --- | --- | --- | --- |
| Parameter | Mandatory | Type | Description |
| ReportStartDate | N | Date | This parameter is indicating the start date when the report has to be generated based on date range |
| ReportEndDate | N | Date | This parameter is indicating the end date when the report has to be generated based on date range |
| FileNamePrefix | Y | String | File name prefix for the generated report |
| IMPEXFolderPath | Y | String | Folder path relative to IMPEX where the report should be generated |

**Post Pennies Donation Details**

This job posts the donation details of orders whose donation status has been marked as ‘Ready for export’ using the Pennies Post Donation API.

A script-module-step is used to execute the job schedule.

Script module: int\_pennies\_core/cartridge/scripts/job/PenniesJobAdapter.js

# Operations, Maintenance

**Failover / Recovery Process**

In case the Pennies service is unavailable, the user will not see donate to Pennies option during checkout.

The service availability can be tracked in SFCC using the Service Status.

**Support**

In case of problems with the integration, missing features, etc. please contact your Pennies account manager.