



**Demandware Implementation Guide**

|  |
| --- |
| **Client Success** |
| **3/9/2017** |
| **Version 17.2**  **DISCLAIMER:** This is an internal document of Kount Inc. Distribution to third parties is unauthorized. Kount Inc. believes this information to be accurate as of the date of publication but makes no guarantees with regard to the information or its accuracy. All information is subject to change without notice. All company and product names used herein are trademarks of their respective owners. |

Copyright © 2012-2017

Kount Inc.

All Rights Reserved

Table of Contents

[Overview 5](#_30j0zll)

[Summary 5](#_1fob9te)

[Requirements 6](#_3znysh7)

[Functional Overview 6](#_tyjcwt)

[Limitations / Constraints 6](#_1t3h5sf)

[Compatibility 6](#_2s8eyo1)

[Privacy, Payment 6](#_3rdcrjn)

[Kount Environments 7](#_lnxbz9)

[Integration Certification 8](#_1ksv4uv)

[Logging 8](#_44sinio)

[Demandware Workflow 9](#_2jxsxqh)

[Implementation Setup 10](#_3whwml4)

[Configuration 10](#_qsh70q)

[Step 1 – Site Import 11](#_1pxezwc)

[Step 2 - Create Services 13](#_2p2csry)

[Step 3 - Kount Site Preferences 14](#_3o7alnk)

[Step 4 - Assign Cartridge to Site 17](#_41mghml)

[Step 5 - Modify Storefront Code (Custom Coding Requirements) 17](#_2grqrue)

Step 6 -Modify Controllers/Pipelines 19

[Step 7 - Configure the Event Notification Service (Optional) 26](#_nmf14n)

[External Interfaces (Communication between Kount and Demandware) 27](#_46r0co2)

[User Defined Fields Setup and Configuration 28](#_2lwamvv)

[Configuration of User Defined Fields (UDFs) within Kount 28](#_3l18frh)

[UDF Settings within Demandware Custom Preferences 29](#_4k668n3)

[Modifying UDF map within Demandware 29](#_1egqt2p)

[Error Logging and Notifications 31](#_3ygebqi)

[Appendix A: Upgrading/Updating the Link Cartridge 32](#_sqyw64)

[Appendix B: Demandware Order Post-Authorization Workflow 33](#_3cqmetx)

[Kount Review/Escalate 34](#_1rvwp1q)

[Kount Approved 35](#_2r0uhxc)

[Kount Decline 36](#_28h4qwu)

[Appendix C: Test Cases 37](#_3q5sasy)

[Verifying the Functionality of the Cartridge: Test Cases 37](#_25b2l0r)

[Part I. Controllers 37](#_kgcv8k)

[Requirements for Test Cases: 37](#_34g0dwd)

Part I. Controllers 37

[Part 1.1. Controllers(pre authorization) 38](#_1jlao46)

[Part 1.2.: Controllers(post authorisation) 47](#_1opuj5n)

Part II. Controllers 49

[Appendix D: Manual Integration Steps 52](#_2250f4o)

[Appendix E: Errors Description 58](#_haapch)

# Overview

## Summary

Kount is a leading innovator of solutions for fraud and risk management. Kount's “decision engine” platform is ideal for managing fraud in online/telephone channels that process payments and onboard new customers. Kount is committed to offering an end-to-end, single-source solution focused on speed, simplicity and ease of use...with a no-compromise approach to providing real-time accuracy in results.

The Kount Demandware cartridge will provide rapid integration for Demandware implementations. The Kount cartridge is a self-contained cartridge that can easily integrate into any project. This cartridge can be configured in the Business Manager and contains all elements necessary to perform a successful best practices implementation of Kount.

Kount aggregates and evaluates data from three primary sources, the Data Collector (DC), the Risk Inquiry Service (RIS), and the Kount Persona technology. From these three sources Kount provides a risk score and a response based upon merchant administered rules.

The Data Collector gathers information from a customer’s device by redirecting the device browser momentarily to Kount then back to the merchant. This passive analysis obfuscates Kount’s interaction with the customer and does not affect the customer’s purchasing experience.

The Risk Inquiry Service evaluates the data provided by the Data Collector and the order-form data submitted to the merchant from the customer to create a fraud score. Merchant specified rules are assessed for each transaction during this evaluation process.

The Kount Persona is comprised of linked data points across the breadth of Kount merchants which provide behavioral analytics related to a transaction.

The [Demandware Workflow Diagram](#_3cqmetx) provides an overview of how a transaction flows through Demandware when the Kount Demandware Link Cartridge is enabled.

The Kount Demandware Link Cartridge is written by a third party development firm, support and point releases are ongoing. If additional Link Cartridges or customizations have been added to your instance of Demandware, conflicts may occur and may result in additional support and/or maintenance fees outside the Kount standard integration.

## Requirements

1. All steps from UX Studio Installation & Sandbox Setup Guide (from Demandware) have been completed
2. A Demandware Development Resource: The Integration and installation process includes deployment of a generic cartridge and modification of storefront code & controllers.

## Functional Overview

Kount Link Cartridge includes:

1. RIS (Risk Inquiry System) HTTPS Post
2. Device Data Collector implementation within checkout process
3. Supported payment types:
   1. *Credit card type*
   2. *PayPal*
   3. *Gift Certificate*
4. ENS (Event Notification System) update to Demandware instance
5. UDF (User Defined Field) Support
6. Device Data Exclusion Array (Support for Phone Orders)
7. Multiple Websites
8. Email Notifications for Errors, ENS and Risk Change Events
9. API/RIS Key (in lieu of older Certificate/Key configuration)

## Limitations / Constraints

* No AVS or CVV information has been passed
* Only listed payment methods will be sent to Kount
* Only Post Authorization configuration

## Compatibility

Cartridge designed for Demandware API version 17.2 and Kount version 0630. (A previous version of the Kount Link Cartridge can be obtained for 15.9 version of Demandware if needed.)

## Privacy, Payment

Kount supports different payment types and depends on chosen by the customer payment type (payment tokens are required). If chosen payment method not supported by Kount, then a value of NONE should be passed.

*The integration uses customer profile data, and transfers it to Kount. Hashed credit card data is sent to Kount system. The following credit card data is sent:*

* *Hashed Credit Card Number (using KHASH - Salted Irreversible Hash; PCI Level 1 Compliant)*
* *Customer Name on Order*
* *Credit Card Type*

# Kount Environments

Kount has separate environments for testing and production. The initial integration will first take place in the Kount TEST environment. Boarding documents containing the information for the TEST environment are contained in the “Welcome Email” provided to a merchant when they begin the boarding process with Kount.

The test environment is **not** engineered to support load testing; it is designed primarily to verify connectivity and proper data submission. Many features such as order linking, scoring, device location, and persona related information are disabled or limited in the test environment.

* Test credit cards can be passed into the test environment but will fail in the production environment.
* Port 443/HTTPS is required for submission and receipt of data in both the test and production environments.
* API Keys are required to authenticate to Kount. Each environment requires a separate API key.
  + API keys are created within the AWC (Agent Web Console). After logging in, browse to the **Admin** tab -> **API Keys** -> **Create API Key**.
  + Check both the **RIS and API checkboxes** when creating the API key.
  + The API keys must reside on each server that is posting to Kount.



**NOTE: API Keys are specific to each environment; Keys created in the TEST environment will not work in the PROD environment and vice versa.**

# Integration Certification

Upon verification that the correct data is being passed for both the Data Collector and Risk Inquiry Service the merchant will be issued a Certification Letter along with an additional boarding document providing the Production environment information.

Any customized data created in the test environment will have to be re-created in the production environment, this includes, users, rules, site IDs, user defined fields and API keys.

The test environment will continue to be available to the merchant for testing purposes but should not be used with production traffic from the merchant.

# Logging

To facilitate troubleshooting ensure that logging is enabled in the Demandware Link Cartridge. The Kount Link Cartridge will not interfere with default checkout flow of a site. If an error occurs within the Kount Link Cartridge or if the Kount Service cannot be reached errors are written into separate log files.

Logs are located **Administration > Site Development > Development Setup**.

Navigate to the Log Files section. The Log file naming convention is:  
 **custom-Kount-blade0-4-appserver**-<date stamp>.**log**

Notifications can also be enabled to deliver email messages to specific addresses if an error were to occur. Example of the email body for a Notification Email:

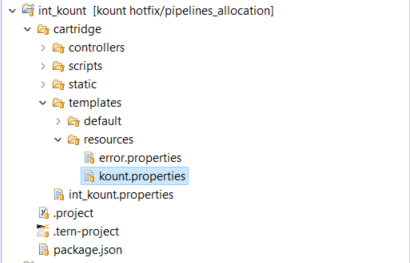


# Demandware Workflow

Within the Demandware workflow, if an order is Declined via a Kount Rule there is a generic Decline message that displays on the checkout page when the customer is attempting to place their order. The message that is displayed can be customized within Demandware.

For customizing Decline Pop-Up open:

int\_kount/cartridge/templates/resources/kount.properties



and change value of “kount.DECLINED” line



# Implementation Setup

The **int\_kount** cartridge is required for the integration. Please obtain the cartridge from your Kount Technical Account Manager, if you are not sure who to contact please email [support@kount.com](mailto:support@kount.com).

## Configuration

There are 6 basic configuration steps:

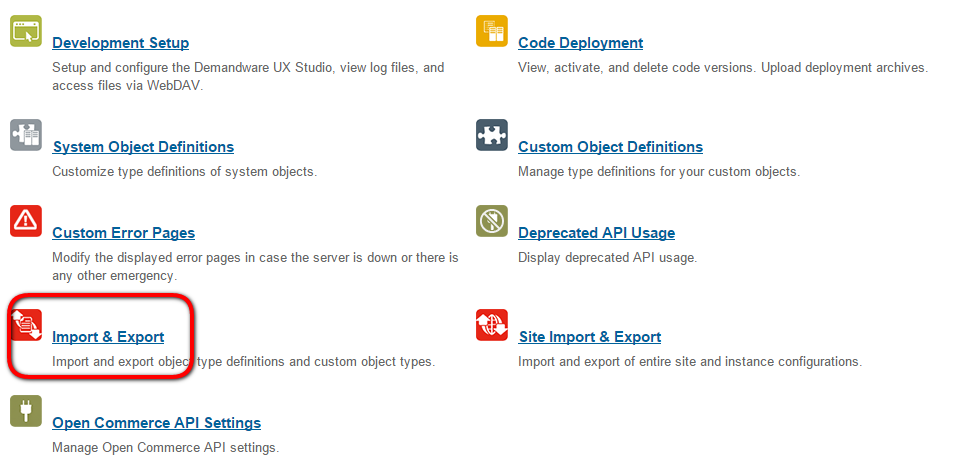
1. Import kount\_metadata.xml file
2. Create Services in DW Business Manager
3. Kount Site Preferences
4. Create API key in Kount Admin Panel
5. Assign the Cartridge to a Site
6. Modify Storefront Code and Controllers
7. (Optional) Configure the Event Notification Service

*Note: Manual integration instructions for step 1 can be found* here*.*

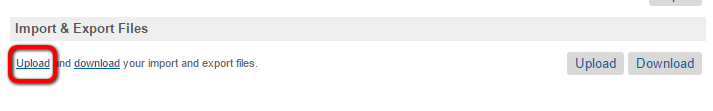
Below are the files and services involved in the Kount Link Cartridge integration.

|  |  |
| --- | --- |
| Cartridge name | int\_kount |
| Controllers | K.js  K\_ENS.js |
| Scripts | assign/CheckKountStatus.js  assign/PostRIS.js  assign/PreRIS.js  kount/KountConstants.js  kount/KountEventHub.js  kount/KountUtils.js  kount/LibKount.js  kount/PostRiskInqueryService.js  kount/UpdateCustomAttribute.js  kount/UpdateOrder.js  kount/CheckProductInventory.js  kount/UpdateOrderExportStatus.js  init/initKount.js |
| CSS | kount.css |
| Templates | kount/confirmationENS.isml  kount/iframe.isml  kount/logo.isml  mail/risk\_change.isml  mail/errornotification.isml |
| Static resources | resources/kount.properties  resources/error.properties |
| Metadata | kount\_metadata.xml |

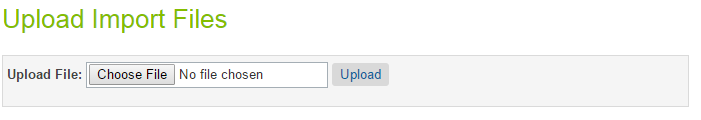
### Step 1 – Site Import

Use the kount\_metadata.xml file to create the custom System Object definitions and configure the Kount Site Preferences. After Logging into Demandware Business Manager. Select **Administration** > **Site Development** > **Import & Export**

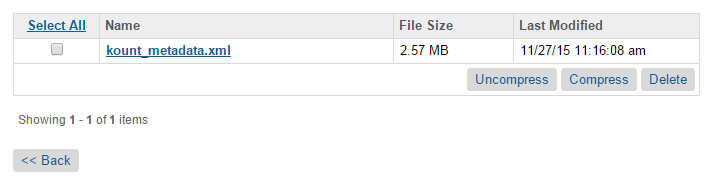
Select the **Upload** button to upload the kount\_metadata.xml file.



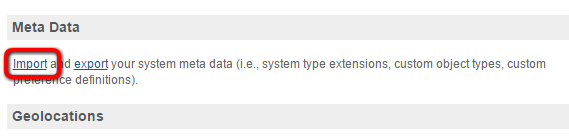
Select the **Choose File** button and navigate to the **kount\_metadata.xml** file. Select the **Upload** button.



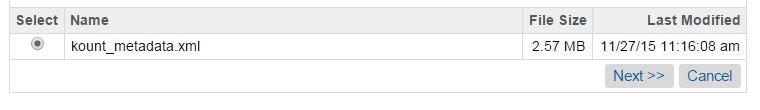
Notice that now the kount\_metadata.xml file is listed within the Manage Import Files page.

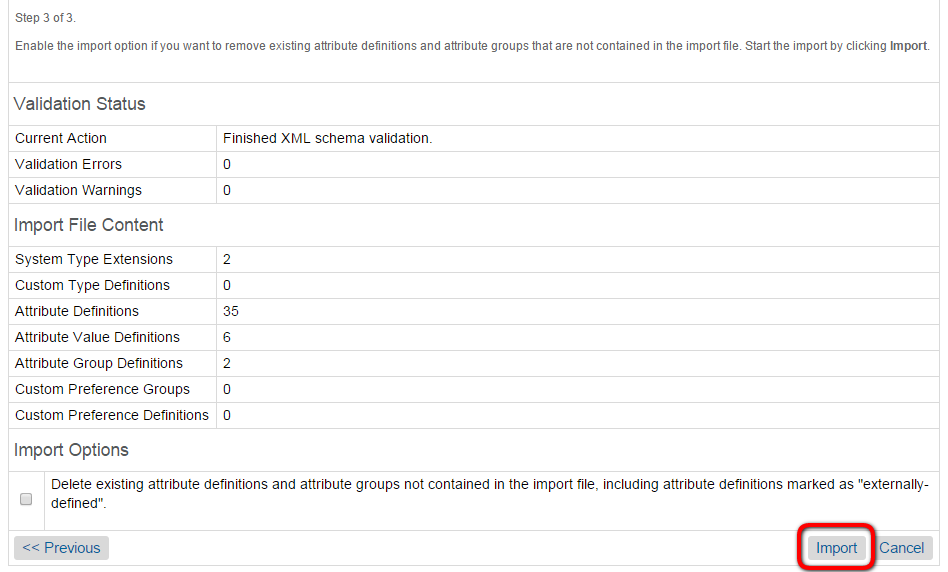


Navigate back to the Import & Export page and select the Import button.

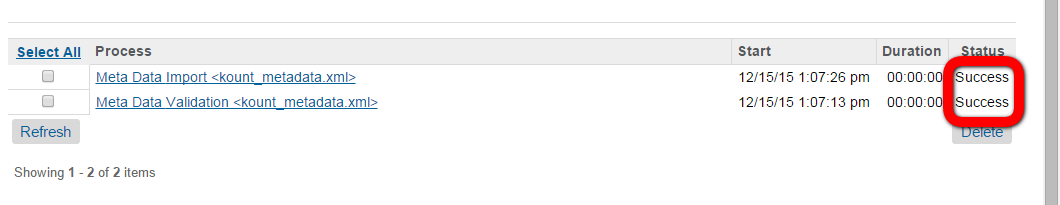


Select the radio button next to the kount\_metadata.xml file and choose the Next button. An XML Validation will run.

Select the Import button. 



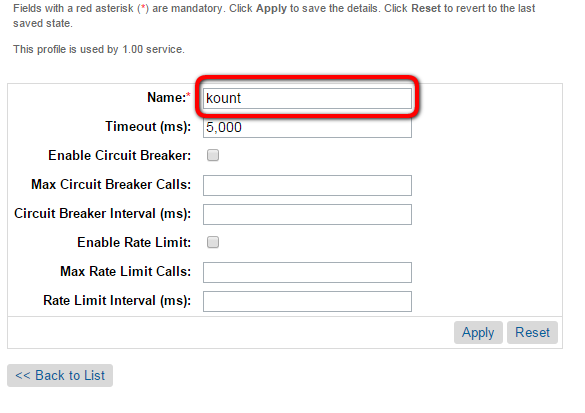
After the import has finished a “Success” status will display.



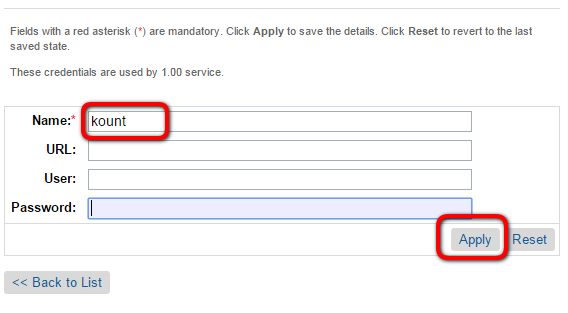
*If a Status of Success does not display, check the DW Sandbox Setup (Site Genesis) and attempt the import again.*

### Step 2 - Create Services

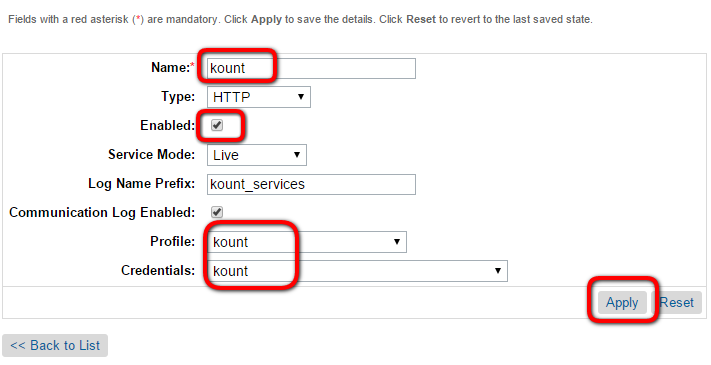
Log into the Business Manager and navigate to [**Administration**](https://500friends01-tech-prtnr-na05-dw.demandware.net/on/demandware.store/Sites-Site/default/ViewApplication-ExpandMenuGroup?MenuGroupID=AdministrationMenu&OverviewPage=SiteNavigationBar-ShowAdministrationOverview)> **Operations** > **Services**. Navigate to the **Profiles** tab and choose the **New** button to create new Profile, name it “kount”.



Navigate to the **Credentials** tab, create a new credentials instance by choosing the **New** button, name it “kount”.   
*NOTE: The password field should be left blank.*



Navigate to the **Services** tab, create a new service by choosing the **New** button, name it “kount”. Fill out the additional fields if needed.



### Step 3 - Kount Site Preferences

The Kount Site Preferences will need to be filled into the fields within this page. The script file will display the default values to the right of the screen, but does not populate the fields automatically. Verify the Site Preferences by navigating to **Merchant Tools** -> **Site Preferences** -> **Custom Preferences**. Select **Kount**.

The Custom Site Preferences page will display. All Sandbox Boarding information needed for this page will be provided by your Kount Technical Account Manager.

Let’s look at the settings page in sections.



* This is where the API/RIS Key that was created earlier in this document is entered. Please use [this link](#_26in1rg) if you have questions about API/RIS Key creation. Please note the Key will need RIS permissions to function properly.
* Enable Event Notification Service (ENS) toggles on or off. The ENS service communicates status changes in Kount to Demandware and updates them within the order. There are additional steps to enabling and configuring ENS. Please use [this link](#_vx1227) or check the index of this document.
* If you would like email notifications to be sent when ENS events are posted enter an email address in the “Kount ENS Email list” field.
* The **Array of Internal IP Addresses** field is designed for internal IP Addresses that should ignore device data, so for instance if your organization accepts phone orders you will want to enter the internal IP Addresses of the agents who accept those orders so their devices are not attached to the independent orders being taken over the phone.



* Website is typically DEFAULT, although website values can be passed. The corresponding Website values will need to be created inside the AWC (Agent Web Console). To do so, navigate to the **Fraud Control** tab > **Websites**.
* Merchant ID or MID value will be provided by your Technical Account Manager.

*Note: Kount does not provide email notifications for any events. The Demandware Link Cartridge has the ability to provide email notifications for various features within Demandware, this is not a feature provided by Kount but rather Demandware.*

* The “Kount ERROR Notification Email” is a list of email addresses that you would like any errors or warnings sent to that have occurred within Demandware concerning the Kount Link Cartridge. See the Error Logging section of this document for more information about the log files.
* The next 6 fields are various ENS email notifications. If any of these events are triggered an email will be distributed to anyone listed in Email field in the “Kount ENS Email List”.



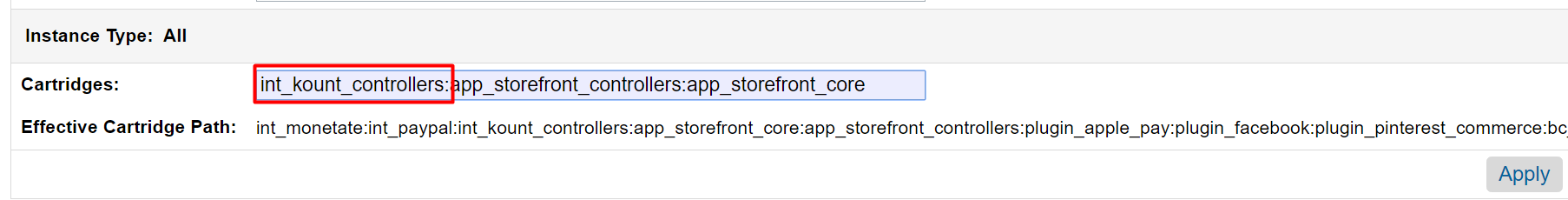
* UDF (User Defined Fields) can be established in the field. There are additional steps to enabling and configuring UDF values please use this link for instructions.
* Enable Kount – Enables the Kount service to run against orders being passed into the environment.
* Authorization type: Kount allows two different order workflow types as PRE- Authorization and POST-Authorization.
* Mode: Test mode for Test Environment production for Production Environment
* Kount uses your storefront controller cartridge, please fill this field with name of this cartridge. For example this cartridge can be named like: app\_storefront\_controllers or storefront\_controllers e.g. You can get name in the Administration > Sites > Manage Sites > <NAME\_OF\_YOUR\_SITE> - Settings -> Cartridges input  
    
  Example: app\_storefront\_controllers.

### Step 4 - Assign Cartridge to Site

The Cartridge will need to be assigned to the Merchant’s website & Business Manager.

Navigate to **Administration** -> **Sites** -> **Manage Sites**. Select **Kount** off the list of Sites, then choose the **Settings** tab.

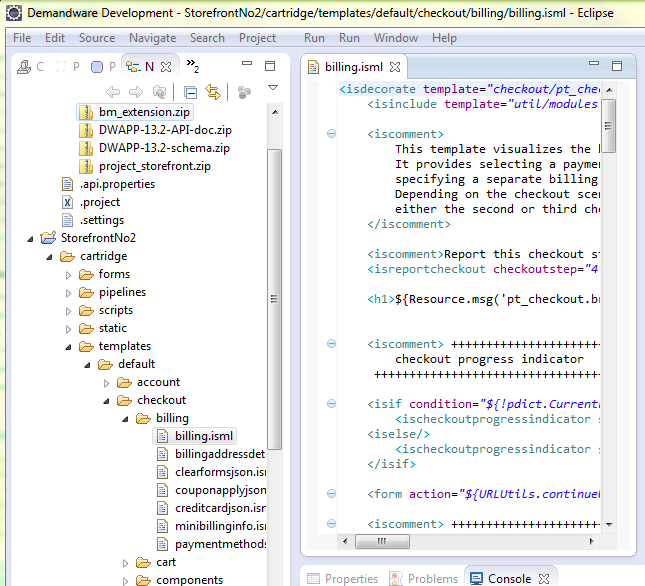
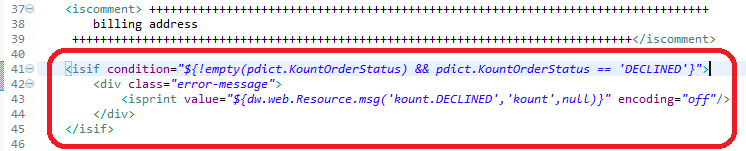
Enter the following text into the **Cartridges** field. **int\_kount:** Be sure to include the colon. Choose the **Apply** button.



### Step 5 - Modify Storefront Code (Custom Coding Requirements)

There are portions of the Kount cartridge implementation that require additions to the internal coding & controllers of the Demandware storefront code. Please consult with your Demandware development resources to complete this portion of the Kount integration.

***NOTE: If your changes are not visible, you may need to reset the code version. To rollback changes to a previous working version. Navigate to Administration > Site Development > Code Deployment. Activate the code version you wish to roll back to.***

* Navigate in Eclipse to your storefront cartridge. Other editors may be used, however the following steps are documented for use with Eclipse. Please reference the Demandware Test Drive documentation for further information.
* Open the Template folder, navigate to checkout/billing/billing.isml (as shown above) and find the billing address line that displayed in the screenshot below.

*NOTE: You can change error message by changing variable in the: int\_kount/cartridge/templates/resources/kount.properties*

* Insert these additional lines of code (as shown in the example) and **save the file.**

<isif condition=*"${!empty(pdict.KountOrderStatus) && pdict.KountOrderStatus =='DECLINED'}"*>

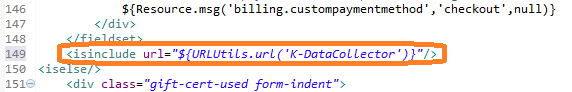
<div class=*"error-message"*>

<isprint value=*"${dw.web.Resource.msg('kount.DECLINED','kount',null)}"* encoding=*"off"*/>

</div>

</isif>

* Remaining within the Template folder, navigate to checkout/billing/paymentmethods.isml and find line displayed in the screenshot below. (Tip: Turn on Line No.’s to locate.)



* Insert the additional line of code (as shown in the example) and **save the file**.

<isinclude url*="${URLUtils.url('K-DataCollector')}"/*>

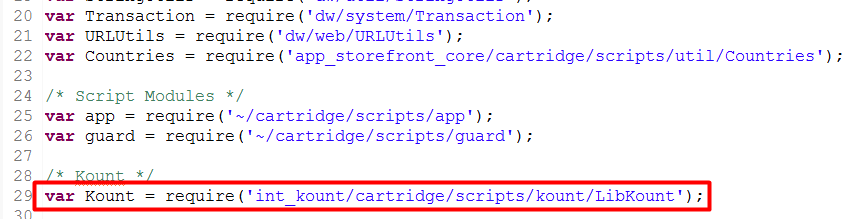
### Step 6 -Modify Controllers/Pipelines

*Note: If you’re using Pipelines instead of Controllers for your storefront, please skip “Controllers” section and go to “Pipelines” section*

Controllers:

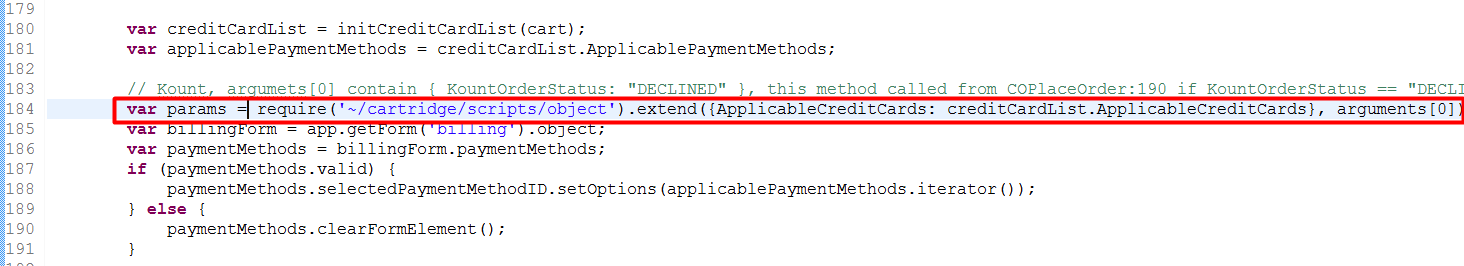
* Open <YOUR\_CONTROLLERS\_CARTRIDGE>/cartridge/controllers/COBilling.js
* Find line:
  + var guard = require('~/cartridge/scripts/guard');
* Insert the below:

var Kount = require('int\_kount/cartridge/scripts/kount/LibKount');



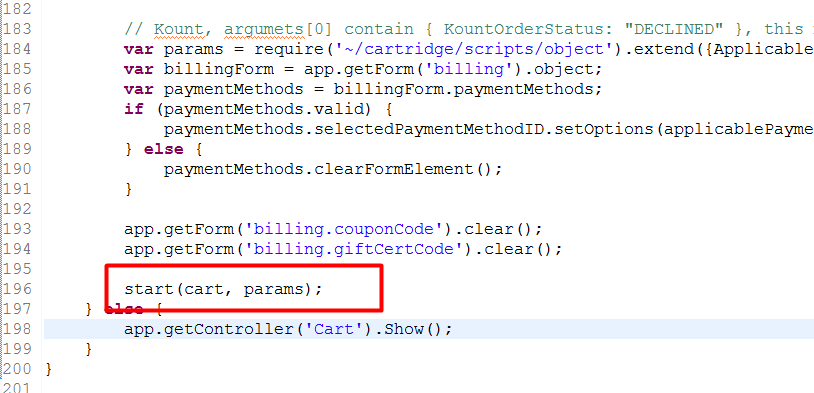
* Near line 180, find:
  + var billingForm = app.getForm('billing').object;
* Add line above:

var params = require('~/cartridge/scripts/object').extend({ApplicableCreditCards: creditCardList.ApplicableCreditCards}, arguments[0]);



* In the same function find (near line 191):
  + start(cart, {ApplicableCreditCards: creditCardList.ApplicableCreditCards});
* And replace the line with:

start(cart, params);



* In the same file (near line 431) find:
  + if (!result) {result = app.getModel('PaymentProcessor').handle(cart.object, app.getForm('billing').object.paymentMethods.selectedPaymentMethodID.value);}
* Insert this code above:

var RISresult = Kount.preRiskCall(cart, publicStart);

if(RISresult && RISresult.KountOrderStatus == 'DECLINED') {

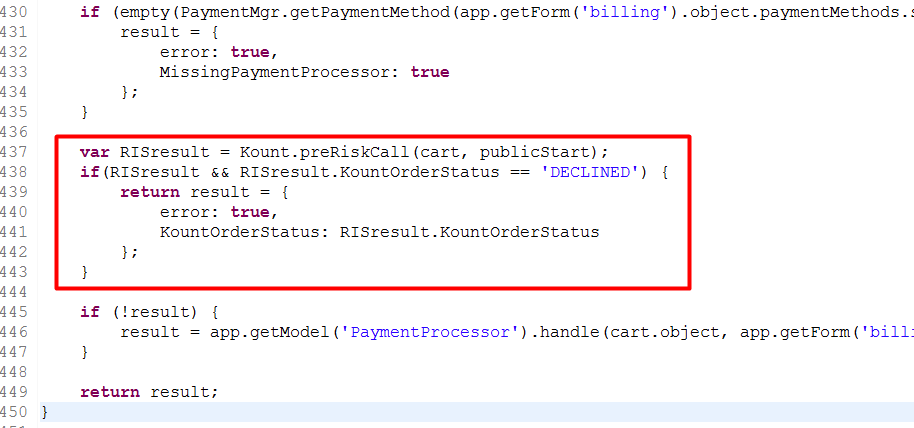
return result = {

error: true,

KountOrderStatus: RISresult.KountOrderStatus

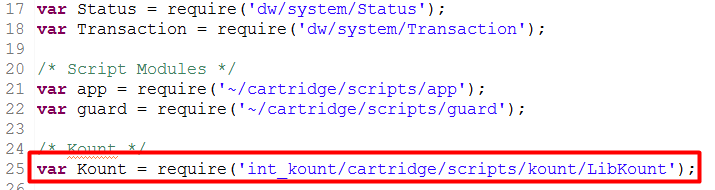
};

}



* Now open <YOUR\_CONTROLLERS\_CATRIDGE>/cartridge/controllers/COPlaceOrder.js
* Find line:
  + var guard = require('~/cartridge/scripts/guard');
* Insert the below:

var Kount = require('int\_kount/cartridge/scripts/kount/LibKount');

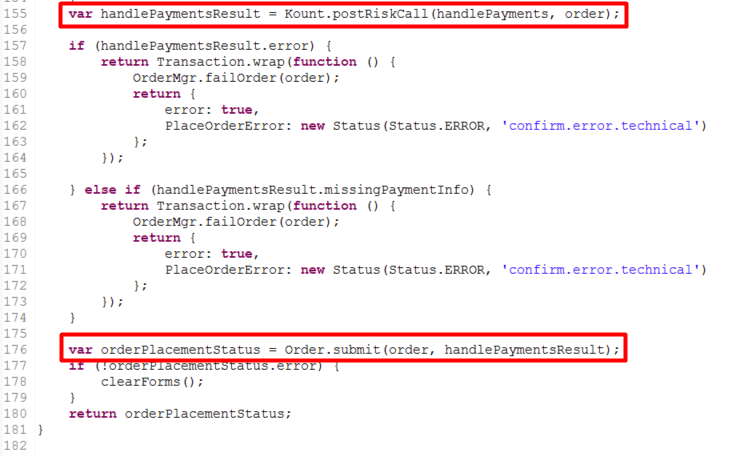


* Near line 155 find:
  + var handlePaymentsResult = handlePayments(order);
* Replace this line with:

var handlePaymentsResult = Kount.postRiskCall(handlePayments, order);

* Near line 176, find:
  + var orderPlacementStatus = Order.submit(order);
* Replace this line with:

var orderPlacementStatus = Order.submit(order, handlePaymentsResult);



* Now open <YOUR\_CONTROLLERS\_CATRIDGE>/cartridge/scripts/models/OrderModel.js
* Find line:

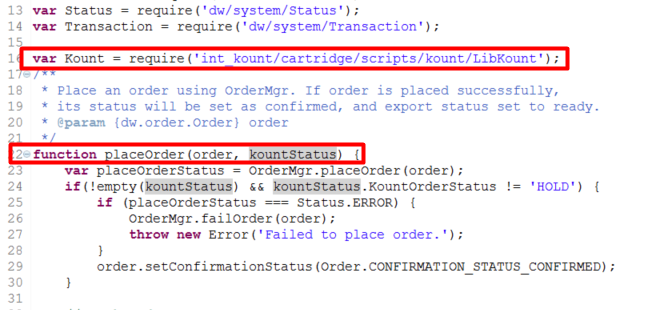
var Transaction = require('dw/system/Transaction');

* Insert the below:

var Kount = require('int\_kount/cartridge/scripts/kount/LibKount');

* Near line 22, find:
  + function placeOrder(order) {
* Replace this line with:

function placeOrder(order, kountStatus) {



* Replace all code from function **placeOrder** by this code:

var placeOrderStatus = OrderMgr.placeOrder(order);

if(!empty(kountStatus) && kountStatus.KountOrderStatus != 'HOLD') {

if (placeOrderStatus === Status.ERROR) {

OrderMgr.failOrder(order);

throw new Error('Failed to place order.');

}

order.setConfirmationStatus(Order.CONFIRMATION\_STATUS\_CONFIRMED);

}

// Mark order as EXPORT\_STATUS\_READY.

Transaction.wrap(function () {

if(kountStatus && kountStaatus.KountOrderStatus == 'APPROVED' && order.getExportStatus().getDisplayValue() != 'EXPORTED' ) {

order.setExportStatus(Order.EXPORT\_STATUS\_READY);

order.setConfirmationStatus(Order.CONFIRMATION\_STATUS\_CONFIRMED);

// Why DECLINED is here? Because if respond status from Kount is DECLINED we skip checkout and return user on the Billing page

} else if(kountStatus && (kountStatus.KountOrderStatus == 'DECLINED' || kountStatus.KountOrderStatus == 'HOLD')) {

order.setExportStatus(Order.EXPORT\_STATUS\_NOTEXPORTED);

order.setConfirmationStatus(Order.CONFIRMATION\_STATUS\_NOTCONFIRMED);

} else {

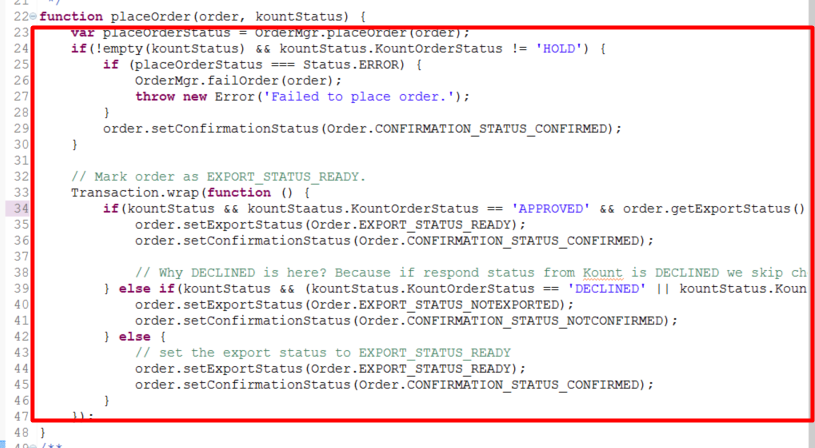
// set the export status to EXPORT\_STATUS\_READY

order.setExportStatus(Order.EXPORT\_STATUS\_READY);

order.setConfirmationStatus(Order.CONFIRMATION\_STATUS\_CONFIRMED);

}

});



* Near line 83, find:

OrderModel.submit = function (order) {

* Replace this code with:

OrderModel.submit = function (order, kountStatus) {

* Near line 88, find:

placeOrder(order);

* Replace this code with:

placeOrder(order, kountStatus);

* Near line 94, find:

order.getGiftCertificateLineItems().toArray().map(function (lineItem) {

return GiftCertificate.createGiftCertificateFromLineItem(lineItem, order.getOrderNo());

}).forEach(GiftCertificate.sendGiftCertificateEmail);

* Replace this code with:

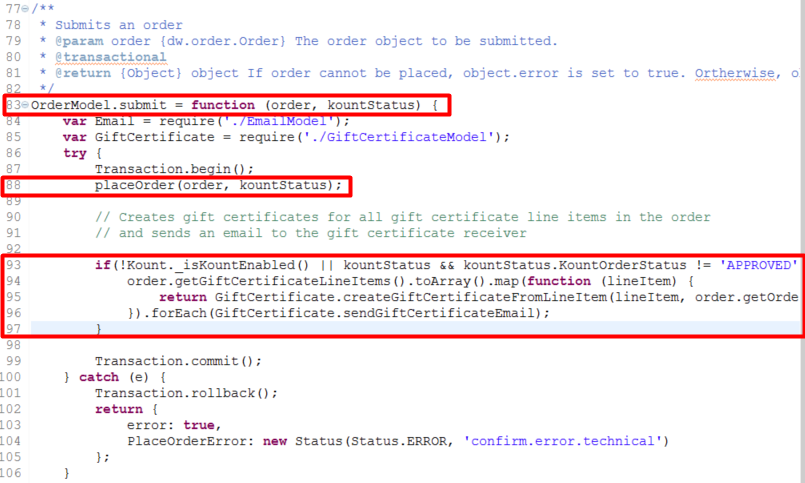
if(!Kount.\_isKountEnabled() || kountStatus && kountStatus.KountOrderStatus != 'APPROVED') {

order.getGiftCertificateLineItems().toArray().map(function (lineItem) {

return GiftCertificate.createGiftCertificateFromLineItem(lineItem, order.getOrderNo());

}).forEach(GiftCertificate.sendGiftCertificateEmail);

}



* Near line 108, find:

Email.sendMail({

template: 'mail/orderconfirmation',

recipient: order.getCustomerEmail(),

subject: Resource.msg('order.orderconfirmation-email.001', 'order', null),

context: {

Order: order

}

});

* Replace this code with:

if(!Kount.\_isKountEnabled() || kountStatus && kountStatus.KountOrderStatus == "APPROVED") {

Email.sendMail({

template: 'mail/orderconfirmation',

recipient: order.getCustomerEmail(),

subject: Resource.msg('order.orderconfirmation-email.001', 'order', null),

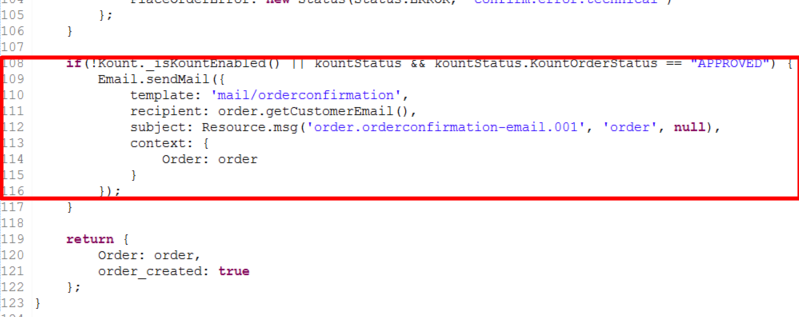
context: {

Order: order

}

});

}

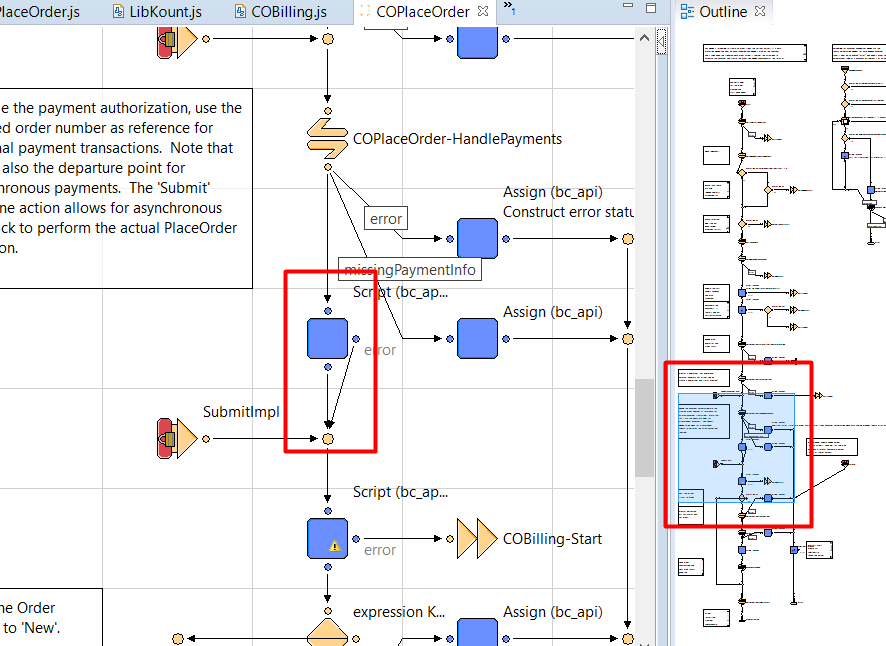


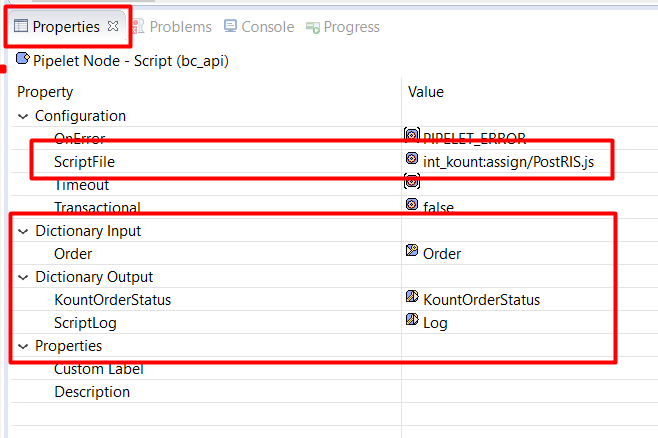
### **Pipelines:**

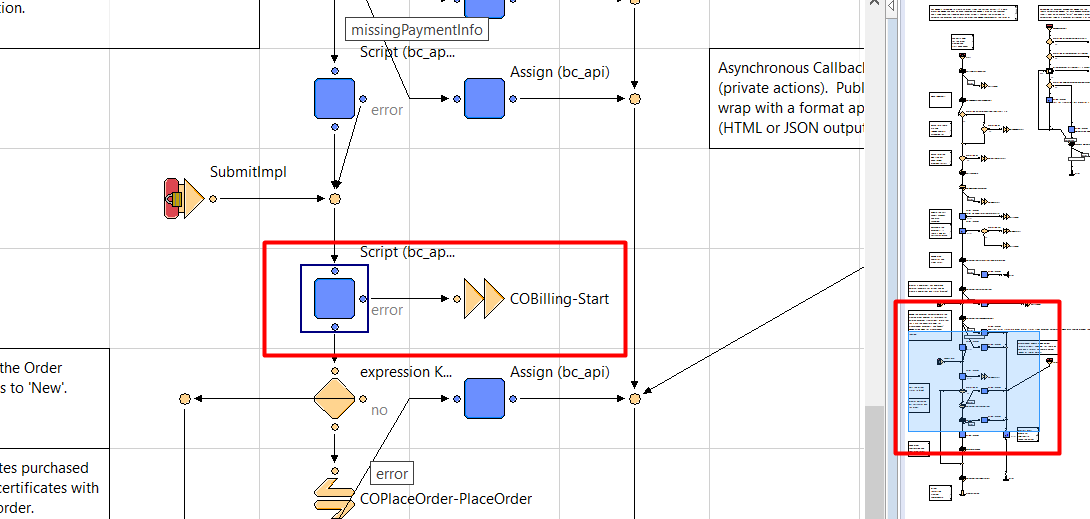
### 

* Open the Pipeline folder, navigate to COPlaceOrder.xml, Start Node with name “Start”
* To create the Post Risk Inquiry Service(RIS) post we must insert Script Node

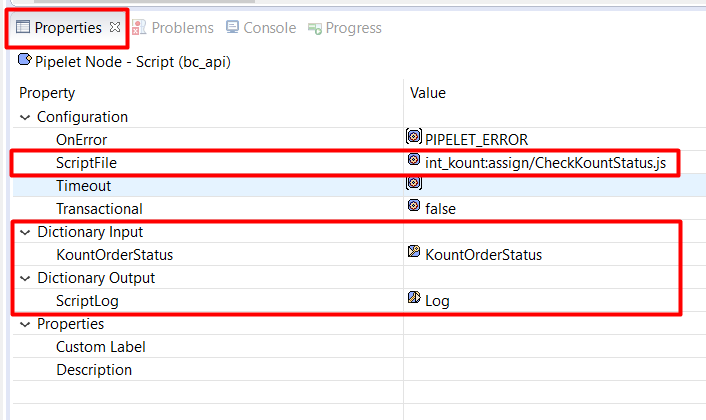
**NOTE: All script nodes for this guide are located at the: int\_kount/cartridge/scripts/assign/**



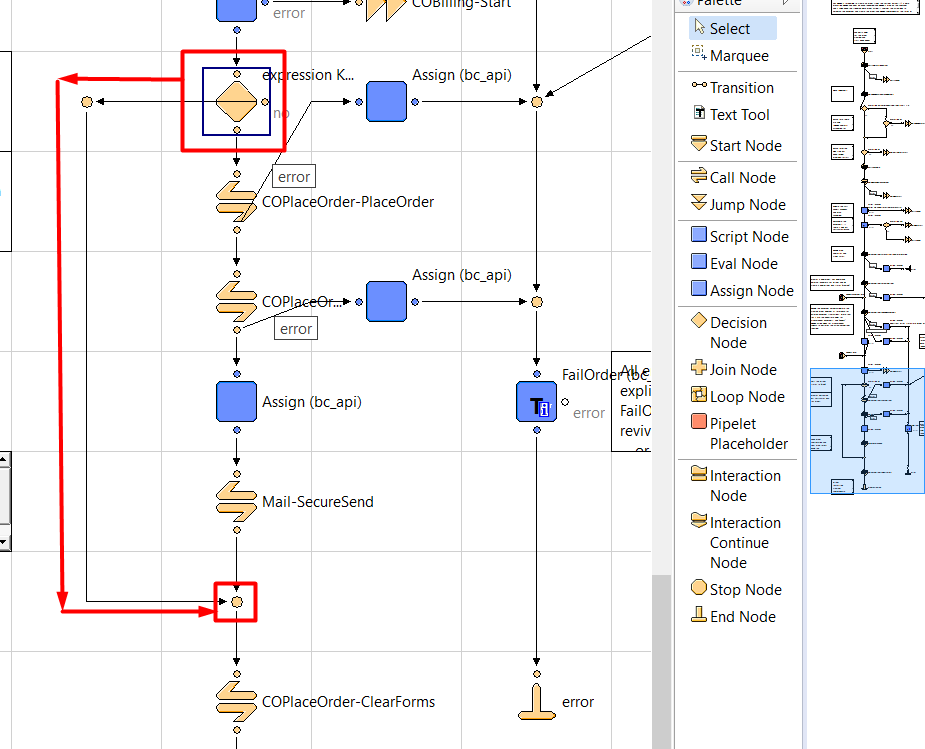
* After placing, click on this node, fill all Properties filed like:
* Below, after Join Node, insert Script node and Jump Node like:



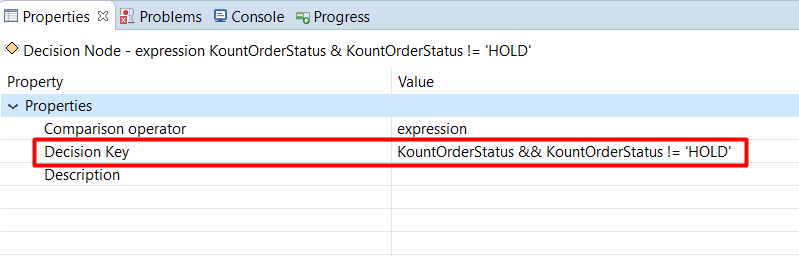
* Jump node must be linked to the COBilling-Start pipeline. Script node must be configured like:



* Below, between Script Node and Call Node (COPlaceOrder-PlaceOrder) need to insert Decision Node and link “Decision No” with Join node between **Mail-SecureSend Jump Node** and **COPlaceOrder-ClearForms Jump None**



* Configuration for Decision Node:



* For update ExportOrderStatus, update script SetOrderStatus.ds with code below and setup input parameter KountOrderStatus(see code below):

SetOrderStatus.ds :

/\*\*

\* Set the order export status to EXPORT\_STATUS\_READY

\*

\* @input Order : dw.order.Order The order.

\* @input KountOrderStatus : String

\*/

**importPackage**( dw.order );

**function** execute( pdict : PipelineDictionary ) : Number

{

// read pipeline dictionary input parameter

**var** order : Order = pdict.Order;

// get Kount order status result

**var** kountOrderStatus : String = pdict.KountOrderStatus;

// Special case if Kount order status is HOLD

**if**( !empty( kountOrderStatus ) && kountOrderStatus == 'APPROVED' && order.getExportStatus().getDisplayValue() != 'EXPORTED' ) {

order.setExportStatus(Order.EXPORT\_STATUS\_READY);

} **else** **if**( !empty( kountOrderStatus ) && ( kountOrderStatus == 'DECLINED' || kountOrderStatus == 'HOLD' ) ) {

order.setExportStatus(Order.EXPORT\_STATUS\_NOTEXPORTED);

} **else** {

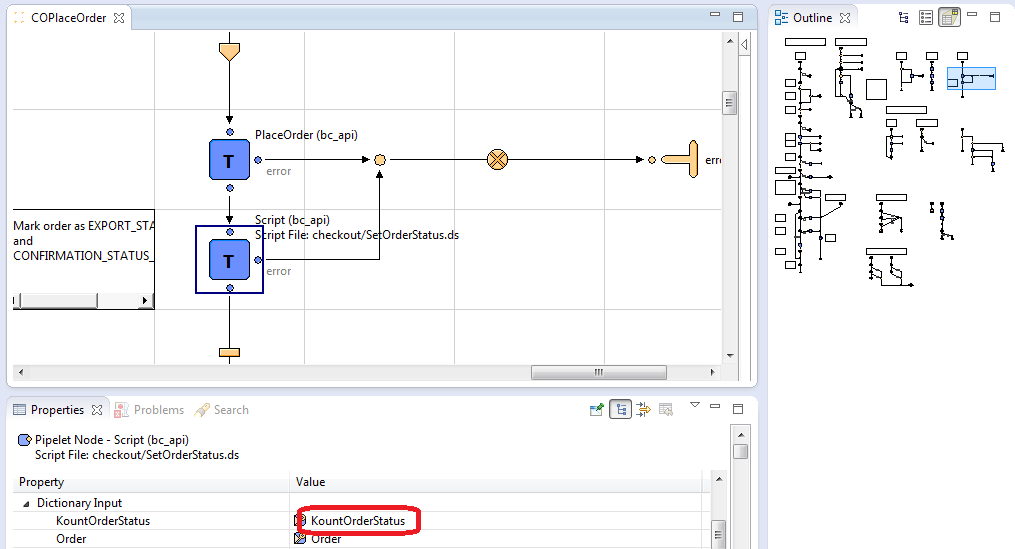
// set the export status to EXPORT\_STATUS\_READY

order.setExportStatus(Order.EXPORT\_STATUS\_READY);

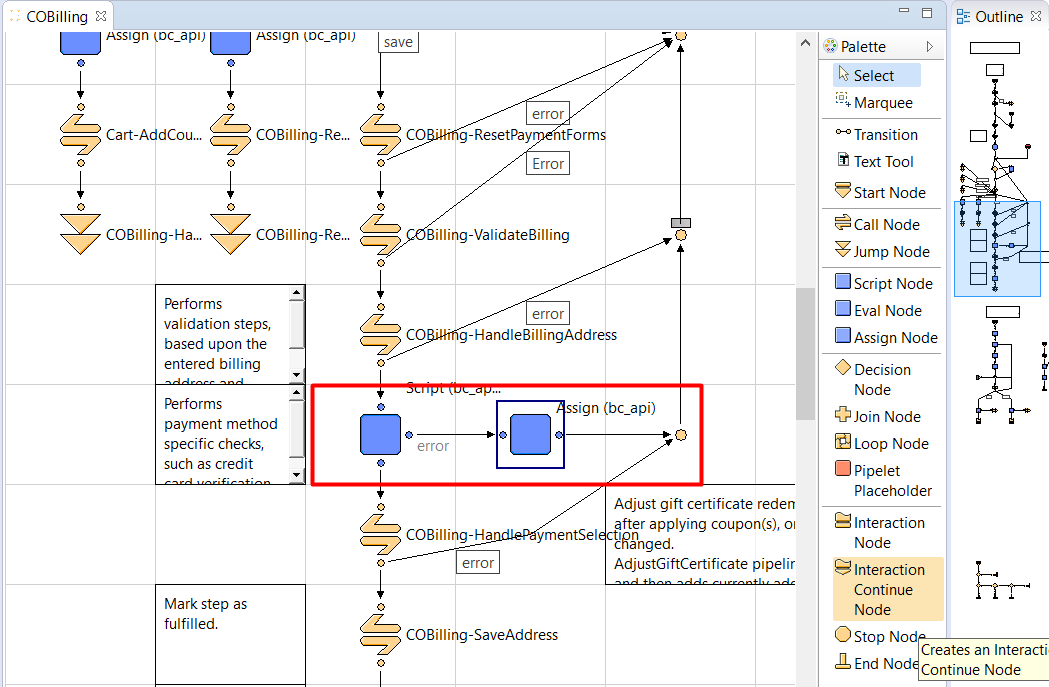
}

**return** PIPELET\_NEXT;

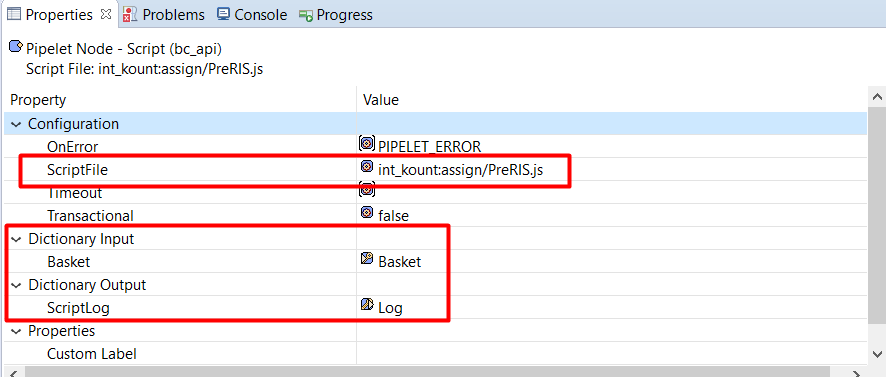
}



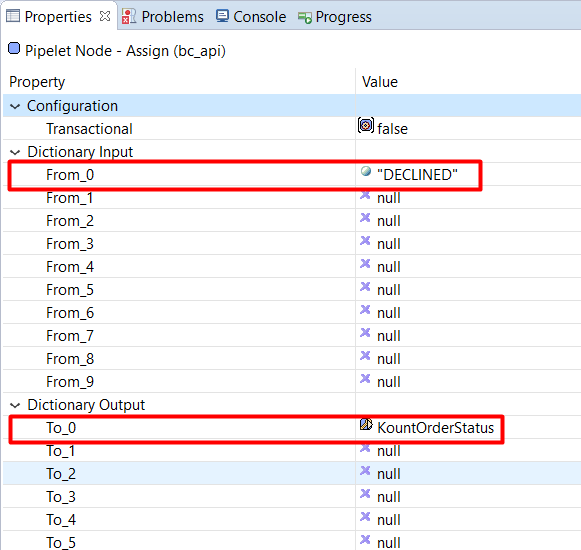
* Open the Pipeline folder, navigate to COBilling.xml, Start Node with name “Start”
* Between **COBilling-HandleBillingAddress** pipeline and **COBilling-HandlePaymentSelection** pipeline insert Script Node and Assign Node. Link Assign node with “error” flow.



* Configuration for Script Node:



* Configuration for Assign Node:



### Step 7 - Configure the Event Notification Service (Optional)

To configure the Event Notification Service (ENS) a Merchant URL will need to be set within the Kount Agent Web Console (AWC) and will need to be enabled within the Site Preferences in Demandware (use this link for the location of this setting). All events are sent to the ENS URL as an XML post(s).

The cartridge does not typically require adding IP or port to whitelist. Communication works through port 443 which is not blocked by Demandware.

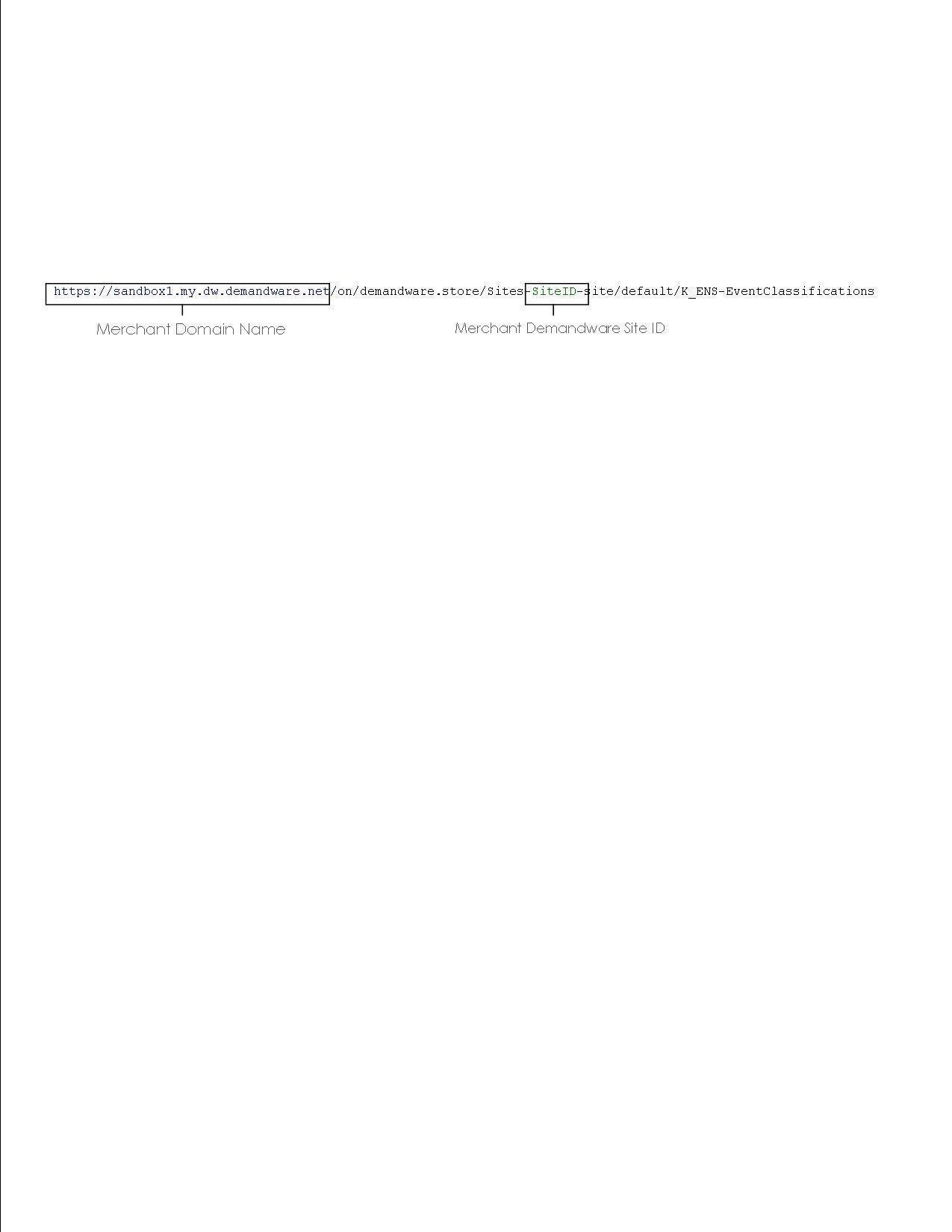
If needed here is the list of IP Addresses that will need to be whitelisted on your server in order to receive the XML posts from Kount:

209.81.12.0/24

64.128.91.0/24

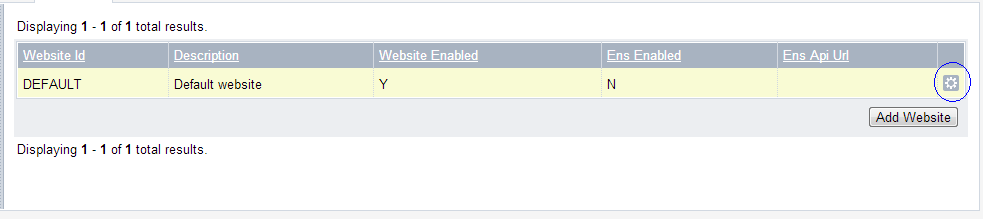
64.128.87.0/24

Below are the steps to setup the ENS (Event Notification System).

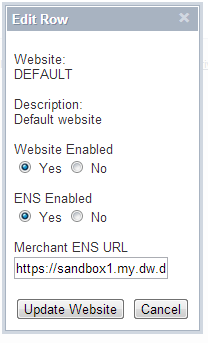
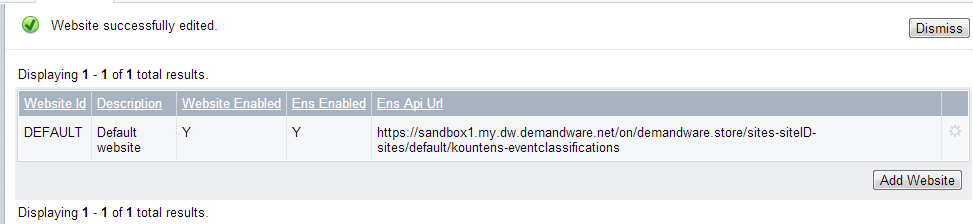
1. Identify your unique Merchant URL.  
     
   Here is an example of an ENS URL:

The rest of the URL are static values. *Note: The “default” value may be different if the merchant is using another language aside from English.*

1. Set the ENS URL within Kount. Navigate to the **Fraud Control** tab > **Websites**. Select the GEAR, (circled in the image below) and choose **Edit**.



1. The “Edit Row” dialog will display. Choose the **ENS Enabled** radio button. Enter the ENS URL (that is unique to your merchant account) within the **Merchant ENS URL** field. Select the **Update Website** button.



Notice the confirmation in the upper left stating the Website was successfully edited.

### External Interfaces (Communication between Kount and Demandware)

The cartridge uses Event Notification System (ENS) for synchronization with the Kount.

**Callback Controllers:**

Notifications from Kount are sent to the cartridge as a series of events formatted in XML. Handlers for these events are implemented as controllers described below. One of them; (K\_ENS-EventClassifications) is an event sorter; it is using the configuration described in [Step 6](#_2u6wntf).

* K\_ENS- EventClassifications – This is the event sorter determining classification of the event.
* K\_ENS- WorkflowStatusEdit, WorkflowReevaluate, RiskChangeScor, RiskChangeReply, RiskChangeVelo, RiskChangeVmax, RiskChangeGeox, RiskChangeNetw, RiskChangeReas - These are the different event handlers.

# User Defined Fields Setup and Configuration

The syntax must be formatted to match the values shown in the graphic in order to map correctly within Kount. The System Object Names within Demandware that can be accessed via UDFs are as follows:

|  |  |  |
| --- | --- | --- |
| Object Name | Label Name | Example Use |
| Order | order | Date|order.date |
| Shipping Address | shippingaddress | State|shippingaddress.state |
| Billing Address | billingaddress | City|billingaddress.city |
| \*Customer Profile | profile | DOB|profile.dob |

*\*Note: Customer Profile Information will be exported in case order was placed by registered customer. System Object Definitions and their attributes can be found within* ***Administration*** *>* ***Site Development*** *>* ***System Object Definitions***

NOTE: The **Amount** UDF Type is not supported in Demandware at this time. If you wish to support custom objects within Demandware please see the additional section regarding the UDF Map - Modifying the UDF map.

## Configuration of User Defined Fields (UDFs) within Kount

*(This is an optional portion of the integration.)*Kount provides a way for merchants to include additional information related to their business that may not be a standard field in Kount by creating UDFs. UDFs should be first setup in Kount admin panel. Navigate to the **Fraud Control** tab > **User Defined Fields**.



UDF field has type Number by default. In order to change the type, choose the appropriate value using the Type dropdown, (alpha-numeric is the only value that can contribute to the VIP List).

When creating UDFs there may be a few minutes delay from the time of creation to the display within the AWC.



## UDF Settings within Demandware Custom Preferences

To pass information into UDFs, navigate within the DW Business Manager to **Site Preferences** > **Custom Preferences**. Choose Kount and scroll down to the **Kount UDF fields**.





## Modifying UDF map within Demandware

To support other Demandware objects an additional mapping should be done in method – **getUDFObjectMap** in script **LibKount.js**



Adding new object to the map should be added in following manner:

UDFMap.put(<label of the object>, {

"meta" : <DW object>.describe(),

"object" : <DW object>

});

# Error Logging and Notifications

The Kount Link Cartridge will not interfere with default checkout flow of a site. If an error occurs within the Kount Link Cartridge or if the Kount Service cannot be reached errors are written into separate log files. Here is an example of the Log file naming convention: custom-Kount-blade0-4-appserver-<date stamp>.log

Notifications can also be enabled to deliver email messages to specific addresses if an error were to occur. This link will navigate to the area in the Site Preferences where this can be set.



Example of the email body for a Notification Email:



# Appendix A: Upgrading/Updating the Link Cartridge

If you have a previous version of Kount Demandware Link Cartridge integrated, you will need to perform the following steps to update/upgrade to the latest Link Cartridge.

1. Delete the instance of int\_kount from eclipse studio. Save all changes and clear the project by navigating to Project > Clean
2. Import the latest kount\_metadata.xml file provided with the latest instance of the Link Cartridge. Specific instructions are [here](#_3as4poj).
3. Import new version of cartridge to your eclipse studio (int\_kount)
4. If you were using 14.x version of the Demandware Link Cartridge, (or older), you will need to use the new Demandware Service Framework. Instructions for adding this service is located [here](#_49x2ik5).
5. If you were using 14.x version (or older) of the Kount Demandware Link Cartridge you will need to *update to the new Kount RIS/API Key to authenticate the RIS HTTPS submission to Kount, (instead of self-signed certificate methodology).* 
   1. Remove the old certificate from Business manager. Navigate to **Administration** > **Operations** > **Private Keys** choose the checkbox next to the certificate and select the **Delete** button.



1. Changes are made to the coding and controllers with each version of the Demandware Link Cartridge, please review ALL [Custom Coding Requirements](#_vx1227) and update them into the storefront code. Please refer to the release notes for specific changes from version to version of the cartridge

# [Appendix B: Demandware Order Post-Authorization Workflow](#_3cqmetx)

*This appendix contains examples of the workflow within a default installation of the Kount Link Cartridge & the expected workflow diagram.*



## Kount Review/Escalate

This is an example of an order that has triggered a Review or Escalate rule action in Kount.





## Kount Approved

This is an example of an order that has not triggered a rule or was manually Approved in Kount.





## Kount Decline

This is an example of an order that triggered a Decline rule action or was manually Declined in Kount.





# Appendix C: Test Cases

## Verifying the Functionality of the Cartridge: Test Cases

### Controllers/Pipelines

* 1.1. pre authorization
* 1.2. post authorization

### Requirements for Test Cases:

* Link to sandbox/test server and Business Manager credentials
* Access to the Kount sandbox/test environment ([https://awc.test.kount.net](https://awc.test.kount.net/))
* Browser utility installed (like Firebug or another tool), to monitor actions within html code source.

#### Prerequisites:

* Login to BM and open Kount
* Go to Administration > Sites > Manage Sites > Kount - Settings
* Fill **Cartridges** field **int\_paypal:int\_kount:app\_storefront\_core:app\_storefront\_controllers**
* Apply changes

##### TestCase 1: Verify all fields and choices are present within the Business Manager (BM) after installing the Kount Link Cartridge

1. Login to BM, choose the site to verify > Site Preferences > Custom Preferences > Kount.
2. Verify the following components are available and set:

- Enable Kount toggle

- Authorization type selector has Pre and Post values.

- Kount UDF fields

- Mode selector has Test and Production values

- Array of Internal IP Addresses to exclude from Data Collection (text area)

- Website ID field

- Merchant ID field

- Kount ERROR Notification (Email field(s))

- Kount ENS Email list (Email field(s))

- Enable Event Notification Service toggle

- RISK\_CHANGE\_NETW dropdown

- RISK\_CHANGE\_GEOX dropdown

- RISK\_CHANGE\_VMAX dropdown

- RISK\_CHANGE\_SCOR dropdown

- RISK\_CHANGE\_VELO dropdown

- RISK\_CHANGE\_REPLY dropdown

- Kount API Key

1. Verify the following components are absent:

- AWC link

- DC Server URL

- Kount RIS Server URL field

##### TestCase 2: Verify the Attributes values within the Order Attributes tab within the Business Manager (BM).

1. Login to BM, choose the site to verify > Ordering > Orders
2. Choose the Find button
3. Open Attributes tab
4. Verify the **Kount Order Status** displays with one of the following values.

- Approved

- Hold

- Declined

1. Verify the following fields display with values.

- Kount Order Transaction ID

- Kount Order GEOX

- Kount Order NETW

- Kount Order SCOR

- Kount Order VELO

- Kount Order VMAX

- Kount Order BROWSER  
- Kount Order OS   
- Kount Order IP\_ORG   
- Kount Order CARDS   
- Kount Order EMAILS  
- Kount Order DEVICES  
- Kount Order COUNTRY  
- Kount Order REPLY

### Part 1.1. Pre authorization

#### Prerequisites:

* + Login to BM and choose the site to verify > Site Preferences > Custom Preferences > Kount
  + Verify the mode selector is set to Test
  + Verify Authorization type selector is set to Pre
  + Save changes.

##### Test Case 3: Testing the Functionality of the iFrame

1. Open Store Front instance of Demandware and place items in the cart.
2. Start Checkout as Guest.
3. Enter a shipping address, select shipping method and click Continue button
4. **Scroll down Billing page**
5. Verify that in the bottom left corner of the page there is a pixel (little blue rectangle)
6. Right Click on pixel and choose **Inspect Element**
7. Verify that pixel is defined as a **div** with class **iframe\_test** and is contained inside the iframe component
8. Using prompt in the Prerequisites set: Set mode selector to production
9. Open the Billing page again and verify that pixel is not visible (the little blue rectangle is absent)
10. Open the page sources (e.g. using Firebug) and search for the iframe
11. Verify that iframe component is still there and div around it is without the test class

##### TestCase 4: Verify internal orders do not collect Device Data via the iFrame

**Prerequisites:**

* + Login to BM and choose the site to verify > Site Preferences > Custom Preferences > Kount
  + Enter your current public IP Address to Array of Internal IP Addresses to exclude from Data Collection
  + Save changes

1. Open Store Front instance of Demandware and place items in the cart
2. Start Checkout as Guest
3. Enter a shipping address, select shipping method and click Continue button
4. **Scroll down Billing page**
5. Verify that in the bottom left corner of the page there is NOT a pixel (little blue rectangle)
6. Open the page sources and search for iframe
7. Verify that there is no available iframe component on page (there will be a disabled one, with id="\_atssh")
8. Select or enter a billing address, enter payment data and click Continue button, submit the order
9. Open the order within the BM (Login to BM, choose the site to verify > Ordering > Orders)
10. Navigate to the Attributes tab and verify that **Kount Order COUNTRY** value is “No data available”

##### TestCase 5: Session ID values for both iFrame collection and HTTPS RIS post are matching

1. Open Store Front instance of Demandware and place items in the cart
2. Start Checkout as Guest
3. Enter a shipping address, select shipping method and click Continue button
4. **Scroll down Billing page**
5. Verify that in the bottom left corner of the page there is a pixel (little blue rectangle)
6. Right Click on pixel choose Inspect Element
7. Verify the iFrame component **src** parameter and collect the value (within Notepad, screenshot, etc.) after “s=”  
   an example would look like this: s=45TaxQ66Yp5Bhn4I\_EvN2pY0bc1qMiaa
8. Select/Enter a billing address, enter payment data and click Continue button. Submit the Order
9. Login to Kount and open Reports > Order Search.

10. Using Start Date, find your report and click “details” link.

11. Verify that Session ID, which is in Transaction Summary  is as in iframe.

##### Test Case 6: Send data to Kount. Check that guest user and order data sent correctly to Kount.

**Steps:**

1. Open Store Front.

2. Choose some product, add it to Cart and start Checkout as Guest.

3. On Shipping page enter a shipping address, select shipping method and Continue checkout.

4. Select or enter a billing address, enter payment data and click Continue button.

5. Submit Order.

6. Login to Kount and open Reports > Order Search.

7. Using Start Date, find your report and click “details” link.

8. Verify that next data sent correctly:

- Transactional Summary section: Date, Order Num

- Customer: Created, Name, Email, Ship Email, Cust. ID (empty)

- Payment: Total, Type

- Device: Location, IP Address (to find out you address can use e.g. can use [http://www.myipaddress.com](http://www.myipaddress.com/))

- Addresses: Billing Address, Shipping Address

- Phone Numbers: Billing Phone, Shipping Phone

- Shopping Cart

- Bank Information

##### TestCase 7: Send data to Kount. Check that registered user and order data sent correctly to Kount.

**Steps:**

1. Open Store Front.

2. Login as registered user (or register a new one).

3. Choose some product, add it to Cart and start Checkout.

4. On Shipping page enter a shipping address, select shipping method and Continue checkout.

5. Select or enter a billing address, enter payment data and click Continue button.

6. Submit Order.

7. Login to Kount and open Reports > Order Search.

8. Using Start Date, find your report and click “details” link.

9. Verify that next data sent correctly:

- Transactional Summary section: Date, Order Num

- Customer: Created, Name, Email, Ship Email, Cust. ID

- Payment: Total, Type

- Device: Location, IP Address (to find out you address can use e.g. can use [http://www.myipaddress.com](http://www.myipaddress.com/))

- Addresses: Billing Address, Shipping Address

- Phone Numbers: Billing Phone, Shipping Phone

- Shopping Cart

- Bank Information

##### TestCase 8.: Decline Order. Check that declined order return user to billing page with proper message.

**Prerequisites:**

Set Rules on Kount that will decline orders :

- Login to Kount

- Open Fraud Control -> Rule Sets

- Copy Rule Set and make all rules action to Declined.

- Open Rule Set and click Activate Rule Set.

So each order which come to Kount will become Declined.

**Steps:**

1. Open Store Front.

2. Login as registered user (or register a new one).

3. Choose some product, add it to Cart and start Checkout.

4. On Shipping page enter a shipping address, select shipping method and Continue checkout.

5. Select or enter a billing address, enter payment data and click Continue button.

6. Submit Order.

7. Verify that you redirected to billing page and “We were unable to process your transaction at this time. Please contact customer support.” notification message appeared.

##### TestCase 9: Workflow Reevaluate. Check that order status changed in OMS (BM), when it changed on Kount.

**Prerequisites:**

Need to enable ENS:

- Login to BM and open Kount > Site Preferences > Custom Preferences > Kount.

- Check Enable Event Notification Service checkbox.

- Apply changes.

**Steps:**

1. Open Store Front.

2. Login as registered user (or register a new one).

3. Choose some product, add it to Cart and start Checkout.

4. On Shipping page enter a shipping address, select shipping method and Continue checkout.

5. Select or enter a billing address, enter payment data and click Continue button.

6. Submit Order.

7. Login to Kount and open Reports > Order Search.

8. Using Start Date, find your report and click “details” link.

9. Verify order status and Transaction ID.

10. Login to BM and open Kount > Ordering > Orders > Find and open your order.

11. Open Attributes tab and verify that order status, Transaction ID have the same values as on Kount.

12. Again open order on Kount and change it status using form in the bottom of page.

13. Open order in BM and verify that order status changed.

##### TestCase 10: Risk Change Score. Check that order risk evaluation changed OMS (BM), when it changed on Kount.

**Prerequisites:**

Need to enable ENS:

- Login to BM and open Kount > Site Preferences > Custom Preferences > Kount.

- Check Enable Event Notification Service checkbox.

- Apply changes.

**Steps:**

1. Open Store Front.

2. Login as registered user (or register a new one).

3. Choose some product, add it to Cart and start Checkout.

4. On Shipping page enter a shipping address, select shipping method and Continue checkout.

5. Select or enter a billing address, enter payment data and click Continue button.

6. Submit Order.

7. Login to Kount and open Reports > Order Search

8. Using Start Date, find your report

9. Click “details” link.

10. Verify data in Current Risk Evaluation block.

11. Login to BM and open Kount > Ordering > Orders > Find and open your order.

12. Open Attributes tab and verify that order status, Transaction ID have the same values as on Kount.

13. Make the XML posts for the RISK EVENT changes and the Reevaluate change events.

14. Open order in BM and verify that order RISK EVENT attribute changed.

##### TestCase 11. Additional Payment Types verification

1. Open Store Front.

2. Choose some product, add it to Cart and start Checkout as Guest.

3. On Shipping page enter a shipping address, select shipping method and Continue checkout.

4. Select or enter a billing address, enter different payment method, fill correct data (important for PayPal) and click Continue button.

5. Submit Order.

6. Login to Kount and open Reports > Order Search.

7. Using Start Date, find your report and click “details” link.

8. Verify that next data sent correctly:

- In case PayPal used – on Kount should be PayPal

- In case Credit Card – on Kount should be Credit Card

- In case Gift card used – on Kount should be Gift card

- In case Gift card + Credit Card used – on Kount should be Credit Card

- In case Gift card + Pay Pal used – on Kount should be PayPal

##### TestCase 12. User Defined Fields verification

1. Login to BM and open Kount > Site Preferences > Custom Preferences > Kount

2. Setup UDF fields in Demandware instance. Fill 'Kount UDF fields' field with 3 values:  
- for Order DW object (e.g. TESTUDF1|order.Order Attribute Definition ID)  
- for OrderAddress DW object (e.g. TESTUDF2|shippingaddress.OrderAddress Attribute Definition ID)  
- for Profile DW object (e.g. TESTUDF3|profile.Profile Attribute Definition ID)

3. In Kount admin set 3 UDF fields (same as for #2)

4. Open Store Front.

5. Choose some product, add it to Cart and start Checkout.

6. On Shipping page enter a shipping address, select shipping method and Continue checkout.

7. Select or enter a billing address, enter payment data and click Continue button.

8. Submit Order.

9. In Kount admin verify UDF fields values for placed order.

10. Login to BM and open Kount > Site Preferences > Custom Preferences > Kount

11. Change UDF to have not correct values.

- for Order DW object (e.g. TESTUDF1|xxx.Order Attribute Definition ID)

12. Open Store Front.

13. Choose some product, add it to Cart and start Checkout.

14. On Shipping page enter a shipping address, select shipping method and Continue checkout.

15. Select or enter a billing address, enter payment data and click Continue button.

16. Submit Order.

17 Data for incorrect UDF is not sent to Kount.

18. Error is occured, Log file contains appropriate info about error in custom - <kount> - <hostname> - appserver - <creation date of the file in GMT>.log

19. Login to BM and open Kount > Site Preferences > Custom Preferences > Kount

20. Change UDF to have correct values.

- for Order DW object (e.g. TESTUDF1|order.Order Attribute Definition ID)

21. In Kount admin panel delete TESTUDF1.

22. Open Store Front.

23. Choose some product, add it to Cart and start Checkout.

24. On Shipping page enter a shipping address, select shipping method and Continue checkout.

25. Select or enter a billing address, enter payment data and click Continue button.

26. Submit Order.

27. Data for TESTUDF1 is not sent to Kount.

28. Error is occured, Log file contains appropriate info about error in custom - <kount> - <hostname> - appserver - <creation date of the file in GMT>.log

##### TestCase 13. Errors logging and email notification verification

1. Login to BM and open Kount > Site Preferences > Custom Preferences > Kount

2. Fill 'Kount Notification Email' field with an email or more.

3. Сreate a test situation when Kount service is unavailable

* *Go to* [*Administration*](https://kount01-tech-prtnr-na03-dw.demandware.net/on/demandware.store/Sites-Site/default/ViewApplication-ExpandMenuGroup?MenuGroupID=AdministrationMenu&OverviewPage=SiteNavigationBar-ShowAdministrationOverview&csrf_token=kBJD104QocRCeu6uqcMm60V52-zv7FHSxKewykHDCDE2oBWX3U5DY17hhWr41BbEEXuPxpLsEvagrFcEa-595I5UOVxwMpIpW4We0u8SgyVPi0JCqZdq74J9gVOuVzuNaO-CwdTkMPkj0--jmHsB_wGrQ7zHgl9ntp5xOJjCBnS-Z4glxBc) *>* [*Operations*](https://kount01-tech-prtnr-na03-dw.demandware.net/on/demandware.store/Sites-Site/default/SiteNavigationBar-ShowMenuitemOverview?CurrentMenuItemId=operations&csrf_token=kBJD104QocRCeu6uqcMm60V52-zv7FHSxKewykHDCDE2oBWX3U5DY17hhWr41BbEEXuPxpLsEvagrFcEa-595I5UOVxwMpIpW4We0u8SgyVPi0JCqZdq74J9gVOuVzuNaO-CwdTkMPkj0--jmHsB_wGrQ7zHgl9ntp5xOJjCBnS-Z4glxBc) *>* [*Services*](https://kount01-tech-prtnr-na03-dw.demandware.net/on/demandware.store/Sites-Site/default/Service-DisplayAll?csrf_token=kBJD104QocRCeu6uqcMm60V52-zv7FHSxKewykHDCDE2oBWX3U5DY17hhWr41BbEEXuPxpLsEvagrFcEa-595I5UOVxwMpIpW4We0u8SgyVPi0JCqZdq74J9gVOuVzuNaO-CwdTkMPkj0--jmHsB_wGrQ7zHgl9ntp5xOJjCBnS-Z4glxBc) *> kount - Details*
* *Uncheck Enable Checkbox*
* *Click Apply*

4. Open Store Front.

5. Choose some product, add it to Cart and start Checkout.

6. On Shipping page enter a shipping address, select shipping method and Continue checkout.

7. Select or enter a billing address, enter payment data and click Continue button.

8. Submit Order.

9. Login to BM and open Administration > Site Development > Development Setup > Open Logs folder

10. Verify separate log file availability: custom - <kount> - <hostname> - appserver - <creation date of the file in GMT>.log

11. Verify log file content has following format: [error time in GMT] <error type> <system description> <custom description>

12. Go to mail boxes .

13. Verify e-mail availability for each e-mail box. Recipient data will contain the following:

from - noreply@noreply.com

to - value from site preferences

cc - empty

bcc - empty

##### TestCase 14: Risk Change Score. Check that email with old and new Evaluation data receiving, when Risk and Reevaluation events changed.

#### Prerequisites:

Need to enable ENS:

- Login to BM and open Kount > Site Preferences > Custom Preferences > Kount.

- Check Enable Event Notification Service checkbox.

- Add your email to Kount Email list

- Check checkbox for all RISK\_CHANGE parameters

- Apply changes.

**Steps:**

1. Open Store Front.

2. Login as registered user (or register a new one).

3. Choose some product, add it to Cart and start Checkout.

4. On Shipping page enter a shipping address, select shipping method and Continue checkout.

5. Select or enter a billing address, enter payment data and click Continue button.

6. Submit Order.

7. Login to Kount and open Reports > Order Search

8. Using Start Date, find your report

9. Click “details” link.

10. Verify data in Current Risk Evaluation block.

11. Login to BM and open Kount > Ordering > Orders > Find and open your order.

12. Make the XML posts for the RISK EVENT changes and the Reevaluate change events.

13. Open your email box.

14. Verify availability of e-mail with old and new Risk Evaluation data.

##### TestCase 15.: Decline Order. Check that declined order returns user to billing page with proper message.

#### Prerequisites:

Set Rules on Kount that will decline orders :

- Login to Kount

- Open Fraud Control -> Rule Sets

- Copy Rule Set and make all rules action to Declined.

- Open Rule Set and click Activate Rule Set.

So each order which come to Kount will become Declined.

**Steps:**

1. Open Store Front.

2. Login as registered user (or register a new one).

3. Choose some product, add it to Cart and start Checkout.

4. On Shipping page enter a shipping address, select shipping method and Continue checkout.

5. Select or enter a billing address, enter payment data and click Continue button.

7. Verify that “We were unable to process your transaction at this time. Please contact customer support.” **notification message** appears.

##### TestCase 16.: Order in Review. Check that order in review is displayed correctly in Kount and DW BM.

**Prerequisites:**

Set Rules on Kount that will put orders to Review :

- Login to Kount

- Open Fraud Control -> Rule Sets

- Copy Rule Set and make all rules action to Review.

- Open Rule Set and click Activate Rule Set.

So each order which come to Kount will become Reviewed.

**Steps:**

1. Open Store Front.

2. Login as registered user (or register a new one).

3. Choose some product, add it to Cart and start Checkout.

4. On Shipping page enter a shipping address, select shipping method and Continue checkout.

5. Select or enter a billing address, enter payment data and click Continue button.

7. Open Kount and find the order. Order should have NO Order number.

8. Click on Details and check that all the data is present in Kount and Order status is **Review**

9. Go back to DW Storefront and click Submit Order button

10. On Kount Order number should be the same as on Confirmation page.

11. User receives **NO** confirmation e-mail

12. In BM Order status is **HOLD**

##### TestCase 17.: Approved Order. Check that approved order is displayed correctly in Kount and DW BM.

**Prerequisites:**

Set Rules on Kount that will put orders to Approved :

- Login to Kount

- Open Fraud Control -> Rule Sets

- Copy Rule Set and make all rules action to Approve.

- Open Rule Set and click Activate Rule Set.

So each order which come to Kount will become Approved.

**Steps:**

1. Open Store Front.

2. Login as registered user (or register a new one).

3. Choose some product, add it to Cart and start Checkout.

4. On Shipping page enter a shipping address, select shipping method and Continue checkout.

5. Select or enter a billing address, enter payment data and click Continue button.

7. Open Kount and find the order. Order should have **NO Order number**.

8. Click on Details and check that all the data is present in Kount and Order status is **Approved**.

9. Go back to DW Storefront and click Submit Order button

10. On Kount Order number should be the same as on Confirmation page.

11. User receives **confirmation e-mail**

12. In DW BM and Kount Order status is **Approved**

### Part 1.2. Post authorisation)

*All the Test Cases from ‘Part 1.1.: Controllers (pre authorization)’ should be used except Test Case 15, Test Case 16, Test Case 17, which should be replaced by following Test Cases:*

##### TestCase 18: Decline Order. Check that declined order return user to billing page with proper message.

**Prerequisites:**

Set Rules on Kount that will decline orders :

- Login to Kount

- Open Fraud Control -> Rule Sets

- Copy Rule Set and make all rules action to Declined.

- Open Rule Set and click Activate Rule Set.

So each order which come to Kount will become Declined.

**Steps:**

1. Open Store Front.

2. Login as registered user (or register a new one).

3. Choose some product, add it to Cart and start Checkout.

4. On Shipping page enter a shipping address, select shipping method and Continue checkout.

5. Select or enter a billing address, enter payment data and click Continue button.

6. Submit Order.

7. Verify that you redirected to billing page and “We were unable to process your transaction at this time. Please contact customer support.” **notification message** appeared.

##### TestCase 19: Order in Review. Check that order in review is displayed correctly in Kount and DW BM.

**Prerequisites:**

Set Rules on Kount that will put orders to Review :

- Login to Kount

- Open Fraud Control -> Rule Sets

- Copy Rule Set and make all rules action to Review.

- Open Rule Set and click Activate Rule Set.

So each order which come to Kount will become Reviewed.

**Steps:**

1. Open Store Front.

2. Login as registered user (or register a new one).

3. Choose some product, add it to Cart and start Checkout.

4. On Shipping page enter a shipping address, select shipping method and Continue checkout.

5. Select or enter a billing address, enter payment data and click Continue button.

6. Click Submit Order button

7. On Kount Order number should be the same as on Confirmation page.

8. User receives **NO** confirmation e-mail

9. In BM Order status is **HOLD**

10. Click on Details and check that all the data is present in Kount and Order status is **Review**

##### TestCase 20: Approved Order. Check that approved order is displayed correctly in Kount and DW BM.

**Prerequisites:**

Set Rules on Kount that will put orders to Approved :

- Login to Kount

- Open Fraud Control -> Rule Sets

- Copy Rule Set and make all rules action to Approve.

- Open Rule Set and click Activate Rule Set.

So each order which come to Kount will become Approved.

**Steps:**

1. Open Store Front.

2. Login as registered user (or register a new one).

3. Choose some product, add it to Cart and start Checkout.

4. On Shipping page enter a shipping address, select shipping method and Continue checkout.

5. Select or enter a billing address, enter payment data and click Continue button.

6. On Kount Order number should be the same as on Confirmation page.

7. User receives **confirmation e-mail**

8. In DW BM and Kount Order status is **Approved**

9. Click on **Details** and check that all the data is present in Kount.

# Appendix D: Manual Integration Steps

**Create System Object Definitions (manual integration method)**

1. Create the following system object definitions:



* Order
  + Kount Order Status
    - id: kount\_Status
    - display name: Kount Order Status
    - type: Enum of Strings
    - mandatory: false
    - externally managed: true
    - values:
      * value: APPROVED, display value: Approved
      * value: HOLD, display value: Hold
      * value: DECLINED, display value: Declined
      * value: CHANGE, display value: Change
      * value: TIME OUT, display value: Time Out
      * value: TIME OUT PAYMENT, display value: Time Out Payment Authorization
  + Kount Order NETW
    - id: kount\_NETW
    - display name: Kount Order NETW
    - type: String
    - mandatory: false
    - externally managed: true
  + Kount Order GEOX
    - id: kount\_GEOX
    - display name: Kount Order GEOX
    - type: String
    - mandatory: false
    - externally managed: true
  + Kount Order SCOR
    - id: kount\_SCOR
    - display name: Kount Order SCOR
    - type: String
    - mandatory: false
    - externally managed: true
  + Kount Order TRAN
    - id: kount\_TRAN
    - display name: Kount Order TRAN
    - type: String
    - mandatory: false
    - externally managed: true
  + Kount Order VELO
    - id: kount\_VELO
    - display name: Kount Order VELO
    - type: String
    - mandatory: false
    - externally managed: true
  + Kount Order VMAX
    - id: kount\_VMAX
    - display name: Kount Order VMAX
    - type: String
    - mandatory: false
    - externally managed: true
  + Kount Order Browser
    - id: kount\_BROWSER
    - display name: Kount Order BROWSER
    - type: String
    - mandatory: false
    - externally managed: true
  + Kount Order OS
    - id: kount\_OS
    - display name: Kount Order OS
    - type: String
    - mandatory: false
    - externally managed: true
  + Kount Order IP\_ORG
    - id: kount\_IP\_ORG
    - display name: Kount Order IP\_ORG
    - type: String
    - mandatory: false
    - externally managed: true
  + Kount Order Cards
    - id: kount\_CARDS
    - display name: Kount Order CARDS
    - type: String
    - mandatory: false
    - externally managed: true
  + Kount Order DEVICES
    - id: kount\_DEVICES
    - display name: Kount Order DEVICES
    - type: String
    - mandatory: false
    - externally managed: true
  + Kount Order COUNTRY
    - id: kount\_COUNTRY
    - display name: Kount Order COUNTRY
    - type: String
    - mandatory: false
    - externally managed: true
  + Kount Order EMAILS
    - id: kount\_EMAILS
    - display name: Kount Order EMAILS
    - type: String
    - mandatory: false
    - externally managed: true
  + Kount Order REASON CODE
    - id: kount\_REASON\_CODE
    - display name: Kount Order REASON CODE
    - type: String
    - mandatory: false
    - externally managed: true



* Site Preferences
  + Enable Kount
    - id: kount\_IsEnabled
    - display name: Enable Kount
    - type: Boolean
    - mandatory: false
  + Kount DC Server Url
    - id: kount\_DCUrl
    - display name: Kount DC Server Url
    - description: Kount Data Collector Server Url.
    - type: String
    - mandatory: false
    - default value: https://tst.kaptcha.com
  + Kount RIS Server URL
    - id: kount\_RISServerUrl
    - display name: Kount RIS Server URL
    - description: Kount RIS Server URL
    - type: String
    - mandatory: false
    - default value: https://risk.test.kount.net
  + Array of IP
    - id: kount\_IPFilter
    - display name: Array of IP
    - description: Allows to skip step for particular IP address(es). E.g. 192.168.0.1, 10.10.10.1, ...
    - type: Text
    - mandatory: false
  + Website ID
    - id: kount\_WebsiteId
    - display name: Website ID
    - type: String
    - mandatory: false
    - minimum value: 1
    - field length: 8
  + Merchant ID
    - id: kount\_MerchantID
    - display name: Merchant ID
    - description: Merchant Number
    - type: String
    - mandatory: false
  + Kount AWC link
    - id: kount\_AWCLink
    - display name: Kount AWC link
    - description: Copy and put this link to Browser, that would be go to Kount Agent Web Console
    - type: String
    - mandatory: false
    - default value: [https://awc.test.kount.net](https://awc.test.kount.net/)
  + Enable Event Notification Service
    - id: kount\_ENS
    - display name: Enable Event Notification Service
    - description: Enable/Disable ENS
    - type: Boolean
    - mandatory: false
  + Kount ENS Email list
    - id: kount\_EmailList
    - display name: Kount ENS Email list
    - description: Email list for ENS
    - type: Set of String
    - mandatory: false
  + Kount ERROR Notification Email
    - id: kount\_NotificationEmail
    - display name: Kount ERROR Notification Emai
    - description: In case errors entered email will be notified
    - type: Set of String
    - mandatory: false
  + RISK\_CHANGE\_GEOX
    - id: kount\_RISK\_CHANGE\_GEOX
    - display name: Kount Email list
    - description: Enable/Disable notification mail
    - type: Boolean
    - mandatory: false
  + RISK\_CHANGE\_NETW
    - id: kount\_RISK\_CHANGE\_NETW
    - display name: Kount Email list
    - description: Enable/Disable notification mail
    - type: Boolean
    - mandatory: false
  + RISK\_CHANGE\_REAS
    - id: kount\_RISK\_CHANGE\_REAS
    - display name: Kount Email list
    - description: Enable/Disable notification mail
    - type: Boolean
    - mandatory: false
  + RISK\_CHANGE\_REPLY
    - id: kount\_RISK\_CHANGE\_REPLY
    - display name: Kount Email list
    - description: Enable/Disable notification mail
    - type: Boolean
    - mandatory: false
  + RISK\_CHANGE\_SCOR
    - id: kount\_RISK\_CHANGE\_SCOR
    - display name: Kount Email list
    - description: Enable/Disable notification mail
    - type: Boolean
    - mandatory: false
  + RISK\_CHANGE\_VELO
    - id: kount\_RISK\_CHANGE\_VELO
    - display name: Kount Email list
    - description: Enable/Disable notification mail
    - type: Boolean
    - mandatory: false
  + RISK\_CHANGE\_VMAX
    - id: kount\_RISK\_CHANGE\_VMAX
    - display name: Kount Email list
    - description: Enable/Disable notification mail
    - type: Boolean
    - mandatory: false
  + Kount UDF fields id:
    - Id: kount\_UDF
    - display name: Kount UDF Fields
    - description: Kount UDF Fields
    - type: Set of String
    - mandatory: false
  + Kount API Key:
    - Id: kount\_APIKey
    - display name: kount API Key
    - type: String
    - mandatory: false

2. Create an attribute group for Kount site preferences and add fields.

* id: Kount
* display name: Kount
* attributes:
  + kount\_IsEnabled
  + kount\_DCUrl
  + kount\_RISServerUrl
  + kount\_IPFilter
  + kount\_WebsiteId
  + kount\_MerchantID
  + kount\_AWCLink
  + kount\_ENS
  + kount\_EmailList
  + kount\_RISK\_CHANGE\_NETW
  + kount\_RISK\_CHANGE\_GEOX
  + kount\_RISK\_CHANGE\_VMAX
  + kount\_RISK\_CHANGE\_SCOR
  + kount\_RISK\_CHANGE\_VELO
  + kount\_RISK\_CHANGE\_REPLY
  + kount\_UDF
  + kount\_NotificationEmail
  + kount\_APIKey

# Appendix E: Errors Description

|  |  |
| --- | --- |
| Error Message | Possible Causes/Solutions |
| Kount method|script - Update Orders; ERROR - KOUNT: UpdateCustomAttribute.js: Order not found | 1. Check API callback link at the [Kount](http://awc.test.kount.net/) - Fraud Control - [Websites](https://awc.test.kount.net/dmc/websites.html)  2. Make sure that orders in Kount Dashboard exist in BM |
| Kount method|script - EventClassifications; ERROR - KOUNT: K\_ENS.js: Error when parsing ENS xml | Ensure that you using latest version of Kount cartridge |
| Kount method|script - PostRISRequest; ERROR - The service is not enabled | Ensure that your Kount Service is enabled (BM - Administration - Operations - Services) |
| Kount method|script - Update Orders; ERROR - KOUNT: UpdateOrder.ds: kount\_REPLY custom field wasn't save | Please, check your error logs for details |