



Kount[®]

Magento User Guide

2017-12-11

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 ©2008-2017
Kount Inc.
All Rights Reserved.**

Overview	6
Magento Payment Workflow Diagram	7
Kount Environments	8
Integration Certification	9
Logging	9
Installation.....	10
Configuration	10
Kount Integration Options	11
Payment Workflow.....	12
Workflow.....	15
Pre-Authorization.....	15
Decline	15
Approve.....	15
Review & Escalate	15
Post-Authorization	16
Decline	16
Approve.....	16
Review & Escalate	16
Managing 'Review' Transactions.....	18
Inventory Management	19
Payment Refunds	20
Pre-Authorization.....	20
Post-Authorization	20
Orders Submitted by Customer Service Representatives	21
Configuring Multiple Sites.....	22
RIS Payment Types	23
Event Notification Service	24
Customizing the ENS Path Within the AWC	24
Questions & Answers.....	26
Q: Where do I get the Kount Magento extension?.....	26
Q: Where do I get the Merchant ID?.....	26
Q: How do I receive a login to the AWC?	26
Q: Should I send production traffic to the test environment to test with?	26
Q: How soon does the ENS notification interact with Magento?	26
Q: Why am I not seeing any log files in the path I specified?	26

Troubleshooting the Kount Magento Extension	27
Q: Was the API key in use from the correct Kount environment?.....	27
Q: Can the URLs be resolved from the Magento Server?	27
Q: I keep getting the following pop up when submitting orders.	27
Appendix A	28
Kount Magento Extension Installation Requirements	28
Appendix B	29
Kount Magento Installation	29
Appendix C	31
Configuring the Kount Integration Options	31
Account Information	33
Payment Review Workflow	35
Disable Payment Methods	36
Phone to Web – Admin Ordering.....	37
Event Notification Service	38
Customizing the ENS Path Within the AWC	38
Logging	40
Admin Ordering	40
Appendix D	41
Warning & Error Codes	41
Appendix E	45
Test Scenarios	45
Approve.....	45
Decline	45
Review	46
Optional Test Scenarios.....	47
Shipping/Billing Address Test Scenario.....	47
Shipping Phone/Billing Phone Test Scenario	48
Order Number Test Scenario.....	48
Appendix F.....	49
Integrating Kount with a Custom Payment Method	49
Using Standard Magento Payment Workflow.....	49
Using Non-Standard Magento Workflow	50
How Does the Kount Extension Work?	50
How to Disable Calling Workflow Methods on Default Events	51
Calling Workflow Methods on Custom Events	52

Overview

Kount aggregates and evaluates data from three primary sources:

- **Data Collector (DC):** Processed in the Iframe. This passive analysis obfuscates Kount’s interaction with the customer and does not affect the customer’s purchasing experience.
- **Risk Inquiry Service (RIS):** 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.

Special note for Magento 2: Magento 2 uses a technique called “transparent redirect” during the checkout process, which sends credit card information directly to the payment processor. Magento no longer has access to the credit card number, so that information will not be sent to Kount.

- **Kount Persona Technology:** Comprised of linked data points across the breadth of Kount merchants, which provide behavioral analytics related to a transaction.

From these three sources, Kount provides a risk score and a response based upon merchant-administered rules.

The Magento Workflow Diagram on page **Error! Bookmark not defined.** provides an overview of how a transaction flows through Magento once the Kount extension is installed.

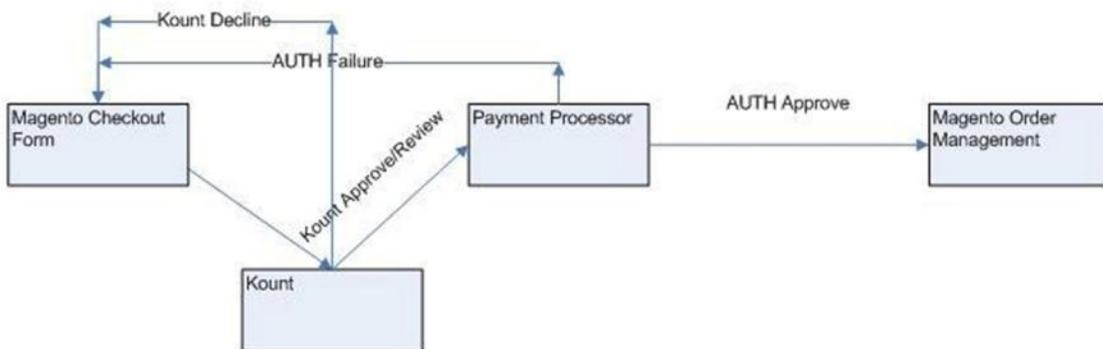
The Kount Magento extension was written and is supported by a third party development firm. Support and point releases are ongoing. If additional extensions or customizations have been added to your instance of Magento, conflicts may occur and may result in additional support and/or maintenance fees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

Magento Payment Workflow Diagram

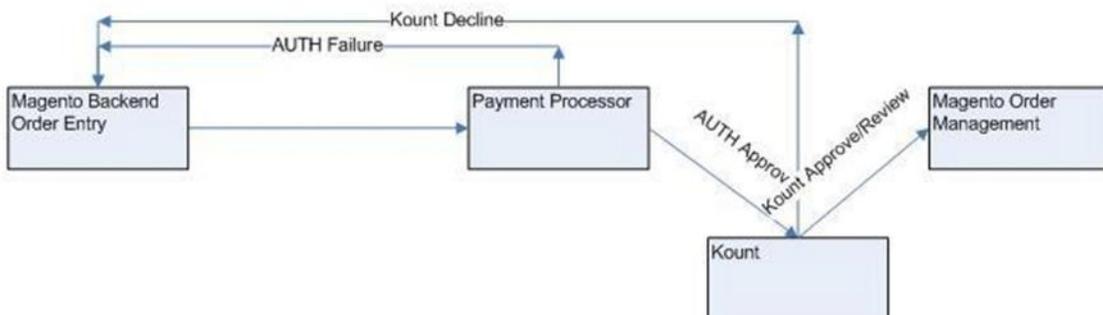
Magento Workflow Diagram Without Kount



Magento Workflow Diagram Pre-Authorization



Magento Workflow Diagram Post-Authorization



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 will be sent to the merchant via e-mail.

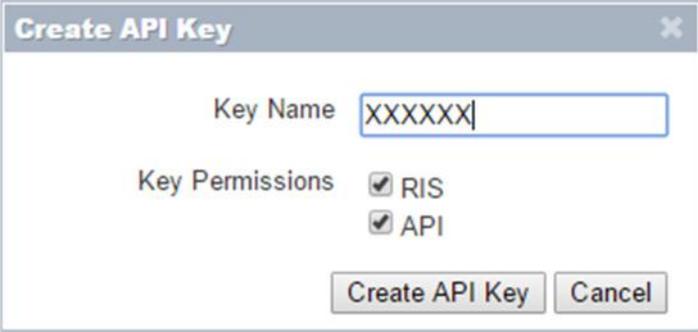
The TEST environment is **not** engineered to support load testing; it is designed primarily to verify connectivity and proper data submission. Many features such as order linking, scoring, device location, and persona-related information are disabled or limited in the TEST environment.

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. After logging in, browse to *Admin* → *API Keys* → *Create API Key*. Check both the RIS and API key checkboxes when creating the API key.



The screenshot shows a dialog box titled "Create API Key". It features a text input field for "Key Name" containing "XXXXXX|". Below this, under the heading "Key Permissions", there are two checked checkboxes labeled "RIS" and "API". At the bottom of the dialog, there are two buttons: "Create API Key" and "Cancel".

The API keys must reside on each server that is posting to Kount.

The API keys are specific to each environment; keys created in the TEST environment will not work in the PRODUCTION 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 logging is enabled using the procedure below. This setting can enable/disable logging for the Kount Extension.

Kount extension logging is also dependent on the Magento global logging settings.

Logging

Enable Kount SDK Logging [WEBSITE]

Log File [Download](#) [GLOBAL]

The [Download](#) link enables direct download access to the log files.

Installation

There are two recommended methods for installing the Kount Magento Extension:

1. Decompress the extension archive and place the files in the appropriate file structure within your Magento installation. For details, please see **Error! Reference source not found.** If additional support is required, this will be at the expense of the merchant.
2. Use the Composer package manager to define the Kount Magento Extension as a dependency in your project. Composer is a tool for dependency management in PHP. It allows you to declare the libraries your project depends on, and it will easily install and update them for you. This installation method is useful if you're a developer already using Composer in your project, or are interested in easily managing upgrades.

Installation requirements and instructions can be found in **Error! Reference source not found.** and **Error! Reference source not found.**

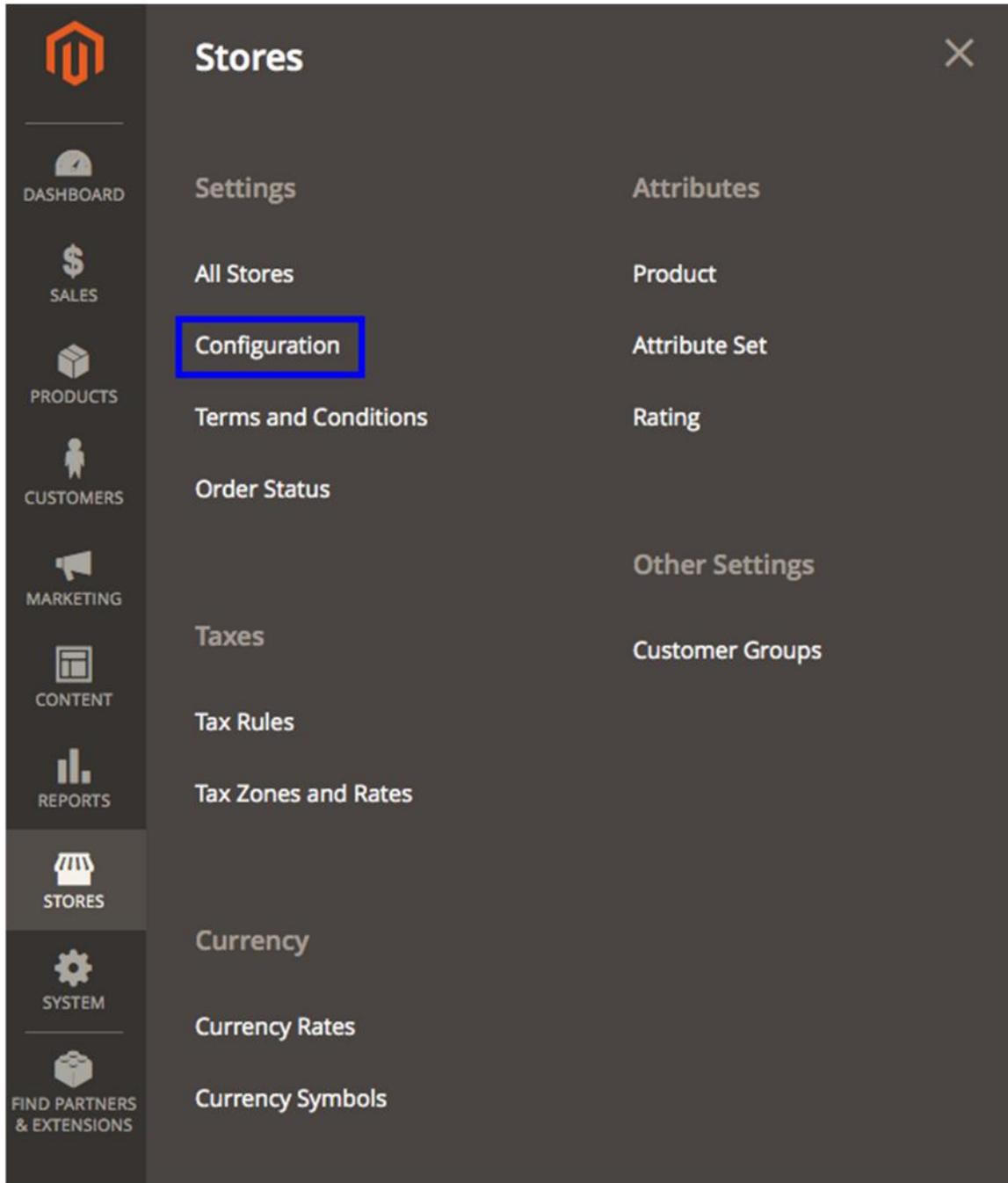
Configuration

The configuration of the Kount Magento Extension occurs within the Magento Admin Panel. Magento users must have the proper resource access within Magento to view and administer the Kount Magento Extension.

Once the Kount Magento Extension has been installed, follow the instructions below.

Kount Integration Options

Select **Stores** → **Configuration**



Select Stores → Configuration → Kount Integration

Account Information



Enabled	<input type="text" value="Yes"/>	[WEBSITE]
	<small>Enable or disable the module store wide.</small>	
Environment	<input type="text" value="TEST"/>	[WEBSITE]
	<small>Whether the extension server will run in test mode.</small>	
Merchant Number	<input type="text"/>	[WEBSITE]
	<small>Your 6 digit merchant number you wish to use from AWC.</small>	
Website ID	<input type="text"/>	[WEBSITE]
	<small>The Website ID from AWC, exactly as it appears.</small>	
API Key	<input type="text"/>	[WEBSITE]
	<small>API authentication token.</small>	

Payment Workflow

Option 1: Pre-Authorization Payment Review

Payment Review Workflow

Payment Review Workflow Mode	<input type="text" value="Pre-Authorization Payment Review"/>	[WEBSITE]
------------------------------	---	-----------

Option 2: Post-Authorization Payment Review

Payment Review Workflow

Payment Review Workflow Mode	<input type="text" value="Post-Authorization Payment Review"/>	[WEBSITE]
Decline Action	<input type="text" value="Cancel Order / Void Payment"/>	[WEBSITE]
	<small>Cancel should be used with payment method in Authorize Only mode. Refund should be used with payment method in Authorize and Capture mode.</small>	
Notify Kount RIS of Processor Decline	<input type="text" value="Yes"/>	[WEBSITE]
	<small>For Post Auth transactions only.</small>	
Decline Action	<input type="text" value="Hold Order / Decline Status"/>	[WEBSITE]
Decline Action	<input type="text" value="Refund / Credit Order"/>	[WEBSITE]

Ability to Disable Payment Methods:

Disable Payment Methods

Kount RIS inquiry is not performed for payment methods which are disabled here.

Disable Payment Method Codes [WEBSITE]

- Check / Money order (checkmo)
- Credit Card Direct Post (Authorize.net) (authorizenet_directpost)
- PayPal Express Checkout Payflow Edition (payflow_express)
- PayPal Credit (payflow_express_bml)
- Credit Card (paypal pro) (payflowpro)
- PayPal Billing Agreement (paypal_billing_agreement)
- Credit Card (Braintree) (braintree)
- No Payment Information Required (free)

Select payment methods to disable.

Notifications:

Event Notifications

Url [WEBSITE]

Additional IPs for Load Balancer [WEBSITE]

Additional IPs from which allowed ENS. Enter comma separated.

Logging:

Logging

Enable Kount SDK Logging [WEBSITE]

Log File [Download](#) [GLOBAL]

Admin Ordering:

Admin Ordering

Use Kount for Backend Ordering [WEBSITE]

Phone To Web Enable [WEBSITE]

Enable Phone-to-Web orders.

Ordering Settings:

When an order is marked for review, the Kount module's default state is set to automatically hide these orders for customers on the front-end of the Magento store.

If you wish to change the default, you may be able to do so within the Admin panel. Changing the answers to "Yes" will display the order to a customer after they have been marked for review.

Order Settings



Display on Review [global]

Orders in Kount review status display in customer's account.

Display Declined [global]

Orders in Kount declined status display in customer's account.

Workflow

As illustrated in the **Error! Reference source not found.**, Kount can be invoked prior to or after the payment processor is queried for authorization. In the Pre-Authorization scenario, all transactions are processed by Kount. In the Post-Authorization scenario, the merchant has the ability to choose whether or not to pass on orders that were declined by the payment processor. Based upon the rules created by the merchant, one of four actions will be returned from Kount: Decline, Approve, Escalate, or Review.

Pre-Authorization

Decline

When a transaction is evaluated by Kount and one or more merchant-created 'Decline' rule conditions are triggered, the order will not continue to the payment processor for authorization, and a failure message will be presented to the customer. This allows merchants to create 'Decline' rule conditions and automatically stop transactions from being submitted to the payment processor.

Similar to when a transaction is declined by the payment processor, when transactions are declined by Kount, no reference of that transaction will be found in the Magento Order Management panel. Since Kount is queried before the processor, declined transactions will be displayed in the Kount Agent Web Console.

Approve

When a transaction is evaluated by Kount and no merchant-created rule conditions for 'Decline', 'Review', or 'Escalate' are triggered, the order is flagged with a status of 'Approve' and will continue to the payment processor for authorization. The default status within Magento for a Kount 'Approve' order is 'Processing'.

Select Items ▾ 25 records found 20 ▾ per page 1 of 2 >

<input type="checkbox"/>	ID	Purchase Point	Purchase Date	Bill-to Name	Ship-to Name	Grand Total (Base)	Grand Total (Purchased)	Status	Action
<input type="checkbox"/>	00000066	Main Website Main Website Store Default Store View	Apr 25, 2016 5:28:41 PM	Joel Martindale	Joel Martindale	\$25.00	\$25.00	Processing	View
<input type="checkbox"/>	00000065	Main Website Main Website Store Default Store View	Apr 25, 2016 5:25:31 PM	Joel Martindale	Joel Martindale	\$25.00	\$25.00	Processing	View
<input type="checkbox"/>	00000064	Main Website Main Website Store Default Store View	Apr 25, 2016 5:20:11 PM	Joel Martindale	Joel Martindale	\$25.00	\$25.00	Processing	View

Review & Escalate

When a transaction is evaluated by Kount and one or more merchant-created rule conditions for 'Review' or 'Escalate' are triggered, the order is flagged as a status of 'Review' or 'Escalate' and will continue to the payment processor for authorization. The default status within Magento for a Kount 'Review' or 'Escalate'

order is 'Review'.

Select Items ▾ 9 records found 20 ▾ per page < 1 of 1 >

<input type="checkbox"/>	ID	Purchase Point	Purchase Date	Bill-to Name	Ship-to Name	Grand Total (Base)	Grand Total (Purchased)	Status	Action
<input type="checkbox"/>	00000059	Main Website Main Website Store Default Store View	Apr 22, 2016 1:07:20 PM	Joel Martindale	Joel Martindale	\$50.00	\$50.00	Review	View
<input type="checkbox"/>	00000058	Main Website Main Website Store Default Store View	Apr 22, 2016 11:12:51 AM	Joel Martindale	Joel Martindale	\$25.00	\$25.00	Review	View
<input type="checkbox"/>	00000057	Main Website Main Website Store Default Store View	Apr 22, 2016 11:10:40 AM	Joel Martindale	Joel Martindale	\$25.00	\$25.00	Review	View

Post-Authorization

Decline

Unlike the Pre-Authorization workflow, when a transaction is evaluated by Kount and one or more merchant-created “Decline” rule conditions are triggered, the customer is not presented with an error popup or notification that the transaction failed.

Within Magento, the inventory will be replaced and the order will be refunded. The order will be refunded if it was authorized and captured. If the order was only authorized during checkout, it will be canceled.

The third option is to change the order status to “On Hold”.

Approve

When a transaction is evaluated by Kount and no merchant-created rule conditions are triggered, the order is flagged with a status of ‘Approved’. The default status within Magento for a Kount ‘Approve’ order is ‘Processing’.

Select Items ▾ 25 records found 20 ▾ per page < 1 of 2 >

<input type="checkbox"/>	ID	Purchase Point	Purchase Date	Bill-to Name	Ship-to Name	Grand Total (Base)	Grand Total (Purchased)	Status	Action
<input type="checkbox"/>	00000066	Main Website Main Website Store Default Store View	Apr 25, 2016 5:28:41 PM	Joel Martindale	Joel Martindale	\$25.00	\$25.00	Processing	View
<input type="checkbox"/>	00000065	Main Website Main Website Store Default Store View	Apr 25, 2016 5:25:31 PM	Joel Martindale	Joel Martindale	\$25.00	\$25.00	Processing	View
<input type="checkbox"/>	00000064	Main Website Main Website Store Default Store View	Apr 25, 2016 5:20:11 PM	Joel Martindale	Joel Martindale	\$25.00	\$25.00	Processing	View

Review & Escalate

When a transaction is evaluated by Kount and one or more merchant-created rule conditions for ‘Review’ or ‘Escalate’ are triggered, the order is flagged with a status of ‘Review’ or ‘Escalate’, and will continue to the

payment processor for authorization. The default status within Magento for a Kount 'Review' or 'Escalate' order is 'Review'.

Select Items 9 records found 20 per page < 1 of 1 >

<input type="checkbox"/>	ID	Purchase Point	Purchase Date	Bill-to Name	Ship-to Name	Grand Total (Base)	Grand Total (Purchased)	Status	Action
<input type="checkbox"/>	000000059	Main Website Main Website Store Default Store View	Apr 22, 2016 1:07:20 PM	Joel Martindale	Joel Martindale	\$50.00	\$50.00	Review	View
<input type="checkbox"/>	000000058	Main Website Main Website Store Default Store View	Apr 22, 2016 11:12:51 AM	Joel Martindale	Joel Martindale	\$25.00	\$25.00	Review	View
<input type="checkbox"/>	000000057	Main Website Main Website Store Default Store View	Apr 22, 2016 11:10:40 AM	Joel Martindale	Joel Martindale	\$25.00	\$25.00	Review	View

Managing 'Review' Transactions

If the Event Notification Service has been enabled, transactions administered within the Kount Agent Web Console can interact with Magento (to enable the Event Notification Service see Appendix C 'Event Notifications').

When a transaction has been flagged by Kount for 'Review' or 'Escalate', the transaction will be placed in an 'On Hold' state within Magento and labeled with a status of 'Review'. Upon reviewing the transaction details within the Kount Agent Web Console, each transaction can be manually approved or declined.

If the Event Notification Service is configured, when changing the status of an order either approving or declining, an event will be sent from Kount to the Magento server and the order will be removed from the 'Review' status and placed back into the status prior to being placed in the 'On Hold' state.

Manual Approve

When a transaction is manually approved within the Kount Agent Web Console, an event will be sent from Kount to Magento indicating an approved status. Within the Magento order management framework, the transaction will be removed from the 'Review' status and will be returned to the status within Magento prior to being placed 'On Hold', which is normally 'Processing'.

Manual Decline

When assigning a transaction to a 'Decline' status within the Kount Agent Web Console, an event will be sent to Magento releasing the order from the 'Review' status resulting in a 'Closed' status, and will attempt to cancel the authorization and refund the charge.

Transactions flagged for Manual Review:

Select Items 20 per page 1 of 1

<input type="checkbox"/>	ID	Purchase Point	Purchase Date	Bill-to Name	Ship-to Name	Grand Total (Base)	Grand Total (Purchased)	Status	Action
<input type="checkbox"/>	00000059	Main Website Main Website Store Default Store View	Apr 22, 2016 1:07:20 PM	Joel Martindale	Joel Martindale	\$50.00	\$50.00	Review	View
<input type="checkbox"/>	00000058	Main Website Main Website Store Default Store View	Apr 22, 2016 11:12:51 AM	Joel Martindale	Joel Martindale	\$25.00	\$25.00	Review	View
<input type="checkbox"/>	00000057	Main Website Main Website Store Default Store View	Apr 22, 2016 11:10:40 AM	Joel Martindale	Joel Martindale	\$25.00	\$25.00	Review	View

Transactions Manually Approved, Manually Declined, still in Review:

<input type="checkbox"/>	ID	Purchase Point	Purchase Date	Bill-to Name	Ship-to Name	Grand Total (Base)	Grand Total (Purchased)	Status	Action
<input type="checkbox"/>	00000059	Main Website Main Website Store Default Store View	Apr 22, 2016 1:07:20 PM	Joel Martindale	Joel Martindale	\$50.00	\$50.00	Review	View
<input type="checkbox"/>	00000058	Main Website Main Website Store Default Store View	Apr 22, 2016 11:12:51 AM	Joel Martindale	Joel Martindale	\$25.00	\$25.00	Review	View
<input type="checkbox"/>	00000057	Main Website Main Website Store Default Store View	Apr 22, 2016 11:10:40 AM	Joel Martindale	Joel Martindale	\$25.00	\$25.00	Review	View

Inventory Management

When a transaction is 'Declined' by either Kount or the payment processor, no inventory is allocated from within Magento. For orders that are flagged as 'Review', inventory is reserved for that item. If a transaction is being held and subsequently manually declined, the inventory item will be returned and inventory incremented. Otherwise if the transaction being held for review is manually approved the inventory will remain decremented.

Payment Refunds

Pre-Authorization

Normally there are two options regarding credit card payment authorization: Authorization and capture of funds together, or capturing funds at a later time.

If the “online” refund is not possible (typically when no payment has yet been made on the order and no invoice created), the extension will attempt to cancel the order.

If an “online” refund is not possible or not successful, and it is not possible to cancel an order. The order will be set to the ‘Kount Decline’ order status and placed in the ‘On Hold’ order state.

Post-Authorization

When Post-Authorization has been chosen within the Kount Integration Options, a decline action must be specified.

For authorization, only the “Cancel Order / Void Payment” options should be selected. In the case when authorization and capture happens at the same time, the “Refund / Credit Order” option should be selected. Depending on the payment method used on the order, this will create a credit memo and refund or void the payment record.

Orders Submitted by Customer Service Representatives

There are two methods when submitting orders by customer service representatives. The first method is to create an order within the Magento Admin Console and the second is to submit the order as if a customer through the site's checkout page.

If orders are submitted through the Magento Admin Console and need to be evaluated by Kount, configure the Kount integration option for Admin Ordering to be enabled.

Admin Ordering

Use Kount for Backend Ordering	<input type="text" value="Yes"/>	[WEBSITE]
Phone To Web Enable	<input type="text" value="Yes"/> <small>Enable Phone-to-Web orders.</small>	[WEBSITE]
IP Address Exclude	<input type="text"/> <small>Current IP: 76.232.220.180 The IP Addresses you wish to exclude for Phone-to-Web orders. Enter comma separated.</small>	[WEBSITE]

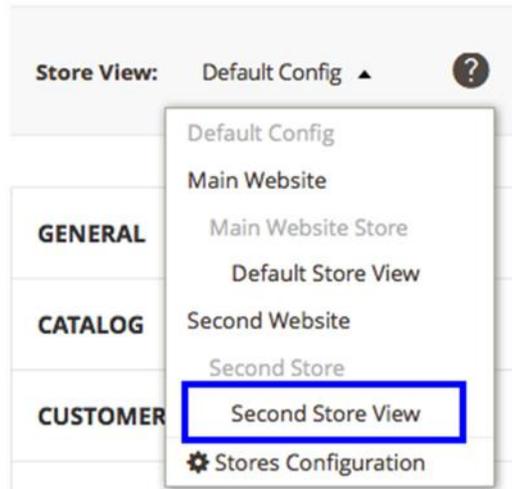
When orders are submitted through the customer checkout page, the Phone To Web option must be configured.

The IP address(es) of the service center must be added to the configuration. Once the IP address(es) have been added, all device information related to the IP address will no longer be associated with the transaction. This will ensure that order linking based on the device will not occur.

Configuring Multiple Sites

If multiple sites are administered within a single Magento instance, each site can be administered independently by Kount. This can be accomplished by creating additional SITE IDs in the Kount Agent Web Console and configuring the Magento Kount Integration Options.

Browse to the additional site configuration within Magento:



In the Website ID field, uncheck the **Use Default** checkbox and input the new site ID that was created in the Kount AWC.

Account Information

Enabled	Yes	<input checked="" type="checkbox"/> Use Default	[WEBSITE]
Enable or disable the module store wide.			
Environment	TEST	<input checked="" type="checkbox"/> Use Default	[WEBSITE]
Whether the extension server will run in test mode.			
Merchant Number	123456	<input checked="" type="checkbox"/> Use Default	[WEBSITE]
Your 6 digit merchant number you wish to use from AWC.			
Website ID	NEWSITE	<input type="checkbox"/> Use Default	[WEBSITE]
The Website ID from AWC, exactly as it appears.			
API Key		<input checked="" type="checkbox"/> Use Default	[WEBSITE]
API authentication token.			

Click the **Save Config** button.

RIS Payment Types

Kount is integrated with Magento with the following payment types supported natively within the Kount Magento Extension that provides cross-merchant linking with the Kount network:

- Braintree
- Authorize.net Direct Post
- PayPal Express Checkout
- PayPal Payflow Link
- PayPal Payflow Pro
- PayPal Payments Standard
- PayPal Payments Pro
- PayPal Payments Advanced
- eWay (Magento Enterprise Edition)

Event Notification Service

Kount provides the ability to manually approve or decline transactions within the Kount Agent Web Console, then have that action post back to the Magento instance. This interaction is dependent upon the **Event Notification Service** (ENS) URL being configured within the Kount AWC.

A load balancer is part of the network infrastructure and can be configured via Magento Admin Console. Within the Kount extension, security measures have been implemented to protect against unauthorized submissions. Only ENS events being sent from Kount will be accepted.

To begin, browse to the Event Notifications section of the Kount Integration Options menu within the Magento Admin Panel. A portion of the URL string that is displayed will be used for the configuration.

Event Notifications

Url	<input type="text" value="http://ee.kount.swarmingstage.com/kount/ens/"/>	[WEBSITE]
Additional IPs for Load Balancer	<input type="text"/>	[WEBSITE]

Additional IPs from which allowed ENS. Enter comma separated.

Customizing the ENS Path Within the AWC

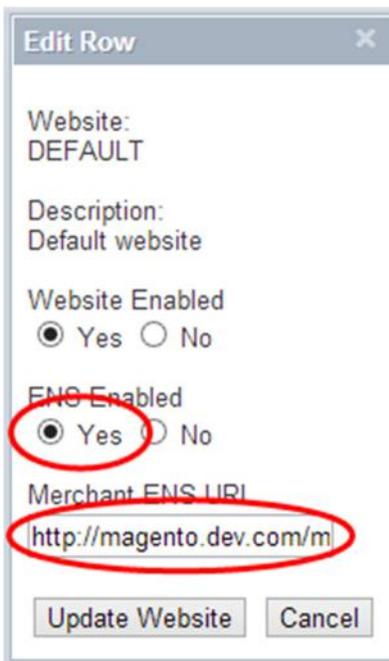
Log into the AWC and browse to the Fraud Control →Websites menu:



If multiple sites are being hosted, choose the appropriate site and click the gearbox and edit link.



Choose the ENS Enabled radio button and then paste in the portion of the URL that was selected from the Event Notifications.



Click the 'Update Website' button.

When adding or editing the ENS URL, there can be a five-minute delay before the URL will take effect. It may also be necessary to whitelist specific IP addresses in order for notifications to be delivered to the server. Contact a Client Success Manager to obtain the ENS IP addresses.

Questions & Answers

Q: Where do I get the Kount Magento extension?

Once the initial kick-off call has concluded, the extension can be downloaded in the Kount Support Resource Center, along with the TEST environment boarding information.

Q: Where do I get the Merchant ID?

A Sandbox Boarding Information document will be sent following the initial kick-off call with the Merchant ID and URLs associated with the DC and RIS processes. A separate document for production will be sent with the production service URLs once the test transaction has been certified.

Q: How do I receive a login to the AWC?

Once a user has been created in the AWC an automated email will be sent requesting a password creation.

Q: Should I send production traffic to the test environment to test with?

Production traffic should not be sent to the test environment due to the possibility of skewed scoring from the test orders.

Q: How soon does the ENS notification interact with Magento?

The ENS events are queued within Kount and then pushed out to the merchant. There can be up to a 30 second delay for the information to be updated between systems.

Q: Why am I not seeing any log files in the path I specified?

Check your configuration file and make sure that logging is enabled in Magento.

Troubleshooting the Kount Magento Extension

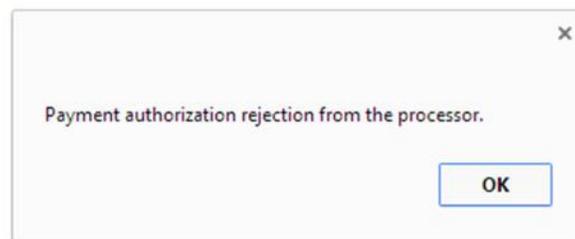
Q: Was the API key in use from the correct Kount environment?

The TEST and PROD environments are completely isolated. It is impossible to utilize the correct API key between the two environments.

Q: Can the URLs be resolved from the Magento Server?

From the command line hosting the Magento server, can the various URLs provided by Kount be resolved?

Q: I keep getting the following pop up when submitting orders.



Check in the kount.log file where the warnings or errors will be displayed.

The default log location for the Kount extension is found in the following path:

`/var/www/magento/var/log`

Log file names: kount.log, exception.log, system.log

Appendix A

Kount Magento Extension Installation Requirements

For more info on Magento 2 system requirements, visit: <http://devdocs.magento.com/guides/v2.0/install-gde/system-requirements.html>

Appendix B

Kount Magento Installation

Please ensure to follow each step precisely as missing a step can cause unexpected results and or issues in your Magento installation. If you encounter an issue that causes your installation to fail or you're seeing error messages display, you will need to reinstall the module and start from the first step again.

Installation via Coping Files

1. It is recommended to enable the cache at System > Cache Management before installing any extension.
2. Open the extension ZIP file you downloaded from the Magento Marketplace and extract it to a directory on your computer using a tool like WinRar, WinZIP or similar.
3. Using the files extracted, copy the folder `swarming_Kount` into `/app/code/Swarming/Kount` in the Magento project.
4. Using Command Line, Terminal or otherwise use the following commands from the root of Magento project:

```
php bin/magento module:enable Swarming_Kount
php bin/magento module:enable Swarming_KountEnterprise (for enterprise version)
php bin/magento setup:upgrade
```

Please make sure to follow the extension specific setup steps now (as applicable) and you're done with installation.

Installation via Composer:

1. It is recommended to enable the cache at System > Cache Management before installing any extension.
2. Download the extension zip archive and save it somewhere outside of your Magento project
3. Navigate to your Magento project root
4. Run the following composer command:

```
composer config repositories.artifacts artifact </full/path/to/folder/with/extension/archive>
the command will add the following code into the composer.json file in repositories node. If the node does not exist, it will be created.
```

```
"artifacts": {
    "type": "artifact",
    "url": "/full/path/to/folder/with/extension/archive>"
}
```

5. Run composer command:
`composer require swarming/Kount`
6. Enable the module and upgrade the database by running the following commands:
`php bin/magento module:enable Swarming_Kount`
`php bin/magento setup:upgrade`

Please make sure to follow the extension specific setup steps now (as applicable) and you're done with installation.

Troubleshooting

- Module isn't showing up in the admin panel / does not work as intended: Try manually deleting content of the `/var/cache/`, `/var/generation/` and `/var/di/ (/generated/ for Magento 2.2)` directories. Please remember that these directories are located in the root of your Magento installation.
- Module continues being hidden or inactive from your Magento installation: Double check all files are in their correct respective locations per the file structure of the module and the official file structure of Magento. It is imperative that you do not forget to upload any single file as this could cause errors and/or unexpected behavior that you may be experiencing.
- Module continues to remain inactive after the previous two attempts: Make sure your Magento installation has the correct file permissions for each uploaded file. Remember that the files *must* be readable by the web server to run properly.
- Receiving errors that 'Class XYZ' could not be found: Reinstall your module, making sure that you have triple checked all the correct files have been uploaded.
- Receiving blank pages, 50x errors or "Exception printing is disabled": Enable "developer mode" in Magento in order to see the real error messages that are displaying.

If you continue to receive errors even after attempting all of the following steps please contact Swarming Technology.

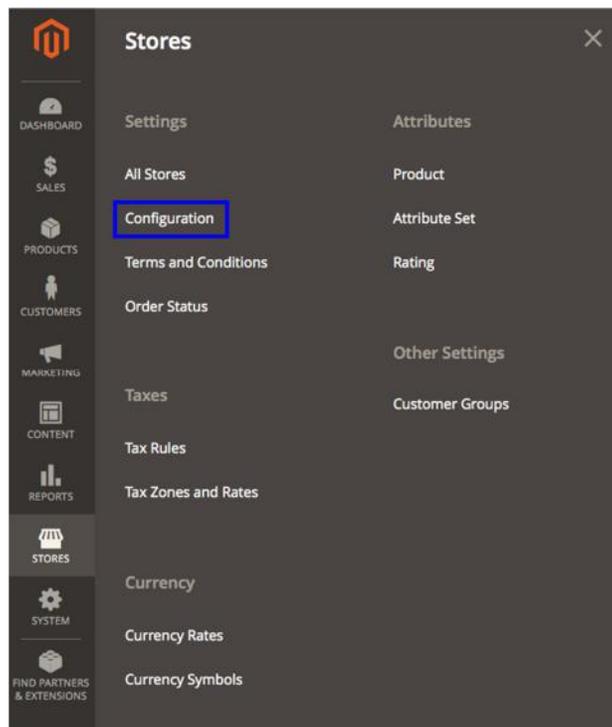
Appendix C

Configuring the Kount Integration Options

Log into the Magento Admin Panel



Configure Kount Integration Options by browsing to Stores →Configure Kount Integration Options by browsing to Stores - Configuration



Click the “Kount Integration Options” link.

The screenshot shows a configuration interface with a sidebar on the left and a main content area on the right. The sidebar contains a list of categories: GENERAL, CATALOG, CUSTOMERS, SALES, SWARMING, SERVICES, and ADVANCED. The 'SWARMING' category is expanded, and the 'Kount Integration' option is highlighted with a blue border. The main content area displays the 'Account Information' section with the following fields:

- Enabled:** Yes (dropdown), [WEBSITE]. Description: Enable or disable the module store wide.
- Environment:** TEST (dropdown), [WEBSITE]. Description: Whether the extension server will run in test mode.
- Merchant Number:** 900410 (text input), [WEBSITE]. Description: Your 6 digit merchant number you wish to use from AWC.
- Website ID:** DEMO.EE (text input), [WEBSITE]. Description: The Website ID from AWC, exactly as it appears.
- API Key:** eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpc3MiOiI5MDA0 (text input), [WEBSITE]. Description: API authentication token.

In the Kount Integration Options page, click on the arrows to expand each available option.

Configuration

Search, Notifications (2), User: swarming

The screenshot shows the configuration page with the 'SWARMING' category expanded. The 'Kount Integration' option is highlighted. The main content area displays a list of expandable options:

- Account Information
- Payment Review Workflow
- Disable Payment Methods
- Event Notifications
- Logging
- Admin Ordering

Each option has a downward arrow icon on the right side, indicating it can be expanded. A 'Save Config' button is visible in the top right corner.

Account Information

Account Information



Enabled	<input type="text" value="Yes"/>	[WEBSITE]
	Enable or disable the module store wide.	
Environment	<input type="text" value="TEST"/>	[WEBSITE]
	Whether the extension server will run in test mode.	
Merchant Number	<input type="text"/>	[WEBSITE]
	Your 6 digit merchant number you wish to use from AWC.	
Website ID	<input type="text"/>	[WEBSITE]
	The Website ID from AWC, exactly as it appears.	
API Key	<input type="text"/>	[WEBSITE]
	API authentication token.	

Enabled – Toggle the Kount extension from being Enabled or Disabled in the Magento Order Management workflow.

Merchant Number – The merchant number issued by Kount. This merchant number is the same regardless if it is in the Kount Test or Production environment.

Website ID – On the initial install this field is set as “Default”. This allows multiple sites within the Magento instance to be managed by Kount. Different rules and searching features can be created for different sites.

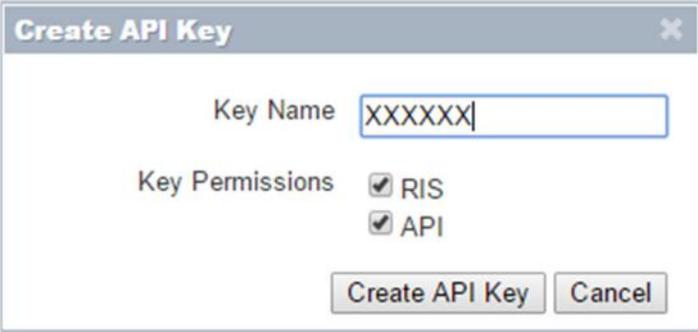
Environment – This setting determines to which Kount environment the Magento instance will be posting data.

API Key – The API key is used for authentication and is created and visible within the Kount Agent Web Console by browsing to *Admin* → *API Keys*.

The API key functionality has replaced the previous method that used X509 certificates for authentication.

API keys are created within the AWC. After logging in, browse to *Admin* → *API Keys* → *Create API Key*.

Check both the RIS and API key checkboxes when creating the API key.



The image shows a dialog box titled "Create API Key". It contains a text input field for "Key Name" with the value "XXXXXX". Below this, there are two checked checkboxes under the heading "Key Permissions": "RIS" and "API". At the bottom of the dialog are two buttons: "Create API Key" and "Cancel".

Payment Review Workflow

Payment Review Workflow

Payment Review Workflow Mode	<input type="text" value="Post-Authorization Payment Review"/>	[WEBSITE]
Decline Action	<input type="text" value="Cancel Order / Void Payment"/>	[WEBSITE]
	<small>Cancel should be used with payment method in Authorize Only mode. Refund should be used with payment method in Authorize and Capture mode.</small>	
Notify Kount RIS of Processor Decline	<input type="text" value="Yes"/>	[WEBSITE]
	<small>For Post Auth transactions only.</small>	

Payment Review Workflow Mode – Specify when Kount will be queried.

- Pre-Authorization Payment Review – Query Kount before the payment gateway.
- Post-Authorization Payment Review – Query Kount after the payment gateway. For Post-Authorization transactions. If a decline rule is triggered there is not a notification that is presented to the customer upon checkout.

Decline Action – The Decline Action options are specific to Post-Authorization transactions. Based on the authorization strategy a merchant has chosen - whether to authorize and capture later or authorize and capture together - different decline actions will be required to configure.

- Hold Order / Decline Status – This will place the order in a hold state but will set the status to decline within Magento. This will allow merchants to release the order if the transaction was a false decline without making the customer resubmit the transaction.
- Cancel Order / Void Payment – Choose this to cancel orders when only authorization happens at checkout and capture is separate.
- Refund / Credit Order – Select this to cancel orders where the transaction is authorized and captured at the same time.

Disable Payment Methods

Disable Payment Methods

Kount RIS inquiry is not performed for payment methods which are disabled here.

Disable Payment Method Codes

None
Check / Money order (checkmo)
Credit Card Direct Post (Authorize.net) (authorizenet_direct)
PayPal Express Checkout Payflow Edition (payflow_express)
PayPal Credit (payflow_express_bml)
Credit Card (paypal pro) (payflowpro)
PayPal Billing Agreement (paypal_billing_agreement)
Credit Card (Braintree) (braintree)
No Payment Information Required (free)

[WEBSITE]

Select payment methods to disable.

Disable Payment Method Codes – If any of these payment methods are selected, there will not be a query made out to Kount for analysis. This functionality provides merchants the ability to whitelist payment methods that they do not want to be evaluated by Kount.

Phone to Web – Admin Ordering

When a merchant submits phone orders via the same web page interface as a customer, the data regarding the merchant's device is being sent to Kount, **not** the customer's device data. This will cause order linking to occur and in time will elevate the score of all orders associated with the persona.

The device information must therefore be stopped from being sent to Kount. To block the device information of customer service representatives from being sent to Kount, the Phone To Web configuration must be enabled and the IP addresses associated with the customer service network(s) must be added.

Admin Ordering

Use Kount for Backend Ordering	<input type="text" value="No"/>	[WEBSITE]
Phone To Web Enable	<input type="text" value="Yes"/>	[WEBSITE]
IP Address Exclude	<input type="text"/>	[WEBSITE]

Current IP: 178.151.245.71
The IP Addresses you wish to exclude for Phone-to-Web orders.
Enter comma separated.

Event Notification Service

Kount provides the ability to manually approve or decline transactions within the Kount Agent Web Console and have that action post back to the Magento instance. This interaction is dependent upon the Event Notification Service URL being configured within the Kount AWC.

To begin, browse to the Event Notifications section of the Kount Integration Options menu within the Magento admin panel. A portion of the URL string that is displayed will be used for the configuration.

Event Notifications

Url	<input type="text" value="http://ee.kount.swarmingstage.com/kount/ens/"/>	[WEBSITE]
Additional IPs for Load Balancer	<input type="text"/>	[WEBSITE]

Additional IPs from which allowed ENS. Enter comma separated.

Customizing the ENS Path Within the AWC

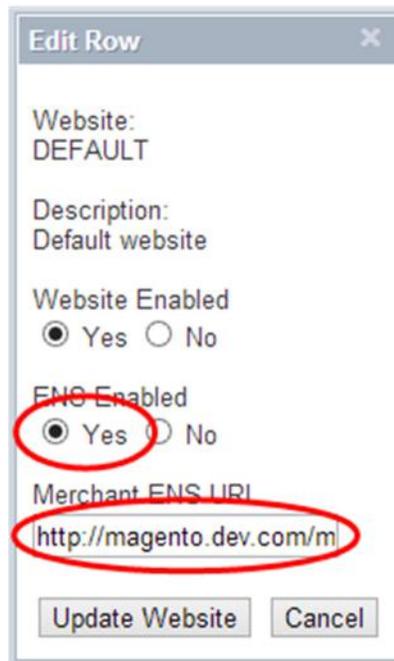
Log into the AWC and browse to the *Fraud Control* → *Websites* menu.



If multiple sites are being hosted, choose the appropriate site and click the gearbox and Edit link.



Choose the ENS Enabled radio button and then paste in the portion of the URL that was selected from the Event Notifications.



Click the Update Website button.

When adding or editing the ENS URL, there can be a five-minute delay before the URL will take effect.

Logging

The logging functionality will enable or disable logging Kount information to the server.

Logging

Enable Kount SDK Logging [WEBSITE]

Log File [Download](#) [GLOBAL]

Admin Ordering

Admin Ordering will pass orders being submitted by the Customer Service Reps through the backend to Kount.

Admin Ordering

Use Kount for Backend Ordering [WEBSITE]

Phone To Web Enable [WEBSITE]

Enable Phone-to-Web orders.

IP Address Exclude [WEBSITE]

Current IP: 178.151.245.71

The IP Addresses you wish to exclude for Phone-to-Web orders.
Enter comma separated.

Appendix D

Warning & Error Codes

Key	Warning/Error Label	Brief Description
201	MISSING_VERS	Missing version of Kount, this is built into SDK but must be supplied by merchant if not using the SDK
202	MISSING_MODE	The mode type for post is missing Refer to Risk Inquiry Service Modes section of this document
203	MISSING_MERC	The six digit Kount Merchant ID was not sent
204	MISSING_SESS	The unique session ID was not sent
205	MISSING_TRAN	Kount transaction ID number
211	MISSING_CURR	The currency was missing in the RIS submission
212	MISSING_TOTL	The total amount was missing
221	MISSING_EMAL	The email address was missing
222	MISSING_ANID	For MODE=P RIS inquiries, the caller ID is missing
223	MISSING_SITE	The website identifier that was created in the Agent Web Console ('DEFAULT' is the default website ID) is missing
231	MISSING_PTYP	The payment type is missing. Refer to the RIS Payment Types section of this document for details
232	MISSING_CARD	The credit card information is missing
233	MISSING_MICR	Missing Magnetic Ink Character Recognition string
234	MISSING_PYPL	The PayPal Payer ID is missing
235	MISSING_PTOK	The payment token is missing. Refer to the RIS Payment Types section of this document for details
241	MISSING_IPAD	The IP address is missing
251	MISSING_MACK	The merchant acknowledgement is missing
261	MISSING_POST	The RIS query submitted to Kount contained no data
271	MISSING_PROD_TYPE	The shopping cart data array attribute is missing. Refer to the Shopping Cart Data section of this document for details
272	MISSING_PROD_ITEM	The shopping cart data array attribute is missing. Refer to the Shopping Cart Data section of this document for details
273	MISSING_PROD_DESC	The shopping cart data array attribute is missing. Refer to the Shopping Cart Data section of this document for details
274	MISSING_PROD_QUANT	The shopping cart data array attribute is missing. Refer to the Shopping Cart Data section of this document for details

Key	Warning/Error Label	Brief Description
275	MISSING_PROD_PRICE	The shopping cart data array attribute is missing. Refer to the Shopping Cart Data section of this document for details
301	BAD_VERS	The version of Kount supplied by merchant does not fit the four integer parameter
302	BAD_MODE	The mode type is invalid. Refer to the RIS Inquiry Service Modes section of this document for details
303	BAD_MERC	The six digit Kount Merchant ID is malformed or wrong
304	BAD_SESS	The unique session ID is invalid. Refer to the Data Collector Requirements and Section creation code example sections of this document for details
305	BAD_TRAN	Kount transaction ID number is malformed
311	BAD_CURR	The currency was wrong in the RIS submission
312	BAD_TOTL	The total amount is wrong. TOTL is the whole number amount charged to customer
321	BAD_EMAL	The email address does not meet required format or is greater than 64 characters in length
322	BAD_ANID	For MODE=P RIS inquiries the caller ID is malformed
323	BAD_SITE	The website identifier that was created in the Agent Web Console ('DEFAULT' is the default website ID) does not match what was created in the AWC
324	BAD_FRMT	The specified format is wrong. Format options are key value pairs, XML, JSON, YAML
331	BAD_PTYT	The payment type is wrong. Refer to the RIS Payment Types section of this document for details
332	BAD_CARD	The credit card information is malformed or wrong, test cards do not work in the production environment
333	BAD_MICR	Malformed or improper Magnetic Ink Character Recognition string. Refer to the RIS Payment Types section of this document for details
334	BAD_PYPL	The PayPal Payer ID is malformed or corrupt. Refer to the RIS Payment Types section of this document for details
335	BAD_GOOG	Malformed or improper Google Checkout Account ID string. Refer to the RIS Payment Types section of this document for details

Key	Warning/Error Label	Brief Description
336	BAD_BLMML	Malformed or improper Bill Me Later account number. Refer to the RIS Payment Types section of this document for details
337	BAD_PENC	The encryption method specified is wrong
338	BAD_GDMP	The GreenDot payment token is not a valid payment token
339	BAD_HASH	When payment type equals 'CARD', [PTYP=CARD] and payment encryption type equals 'KHASH', [PENC=KHASH] the value must be 20 characters in length.
340	BAD_MASK	Invalid or excessive characters in the PTOK field
341	BAD_IPAD	The IP address does not match specifications
342	BAD_GIFT	The Gift Card payment token is invalid due to invalid characters, null, or exceeding character length
351	BAD_MACK	The merchant acknowledgement must be 'Y' or 'N'
362	BAD_CART	There is a discrepancy in the shopping cart key count and the number of items actually being sent in the cart
371	BAD_PROD_TYPE	The shopping cart data array attribute is missing. Refer to the Shopping Cart Data section of this document for details
372	BAD_PROD_ITEM	The shopping cart data array attribute is corrupt or missing. Refer to the Shopping Cart Data section of this document for details
373	BAD_PROD_DESC	The shopping cart data array attribute is corrupt or missing. Refer to the Shopping Cart Data section of this document for details
374	BAD_PROD_QUANT	The shopping cart data array attribute is corrupt or missing. Refer to the Shopping Cart Data section of this document for details
375	BAD_PROD_PRICE	The shopping cart data array attribute is corrupt or missing. Refer to the Shopping Cart Data section of this document for details
399	BAD_OPTN	A UDF has been mistyped or does not exist in the Agent Web Console
401	EXTRA_DATA	RIS keys submitted by merchant were not part of SDK
404	UNNECESSARY_PTOK	When PTYP equals NONE and a PTOK is submitted
413	REQUEST_ENTITY_TOO_LARGE	The RIS Post to Kount exceeded the 4K limit. Refer to the Shopping Cart Data section for further details.
501	UNAUTH_REQ	Error regarding certificate – Using test certificate in prod
502	UNAUTH_MERC	Invalid Merchant ID has been entered
601	SYS_ERR	Unspecified system error – Contact Merchant Services

Key	Warning/Error Label	Brief Description
602	SYS_NOPROCESS	Kount will not process particular transaction
701	NO_HDR	No header found with merchantId=[XXXXX], sessionId=[htot2kk5khpamo45f777q455], trans=[122347] This error occurs when a RIS request goes to the database and there is no data available in the reply. The Update post had an invalid transaction ID#. Check all required fields for update post and confirm they are being passed correctly.

For the most up-to-date information on error codes, see:

https://support.kount.com/Developer_Resources/Getting_Started/Kount_Complete_Technical_Specification_Guide_r._January_2016#RIS_Warning_and_Error_Codes

Appendix E

Test Scenarios

Approve

1. Add JohnDoeApprove@Acme.com email address to the VIP Approve list in the Kount Test environment.
2. Place a test order using any item, via the front end of the test website.
 - a) When filling out the customer information, please use JohnDoeApprove@Acme.com as the email address.
 - b) Fill in necessary Test Credit Card information.
3. Confirm that after the order has been placed that the proper customer experience message is displaying.
4. Confirm that appears as approved in the Agent Web Console.

Decline

1. Add JohnDoeDecline@Acme.com email address to the VIP Decline list in the Kount Test environment.
2. Place a test order using any item, via the front end of the test website.
 - a) When filling out the customer information, please use JohnDoeDecline@Acme.com as the email address.
 - b) Fill in necessary Test Credit Card information.
3. Confirm that after the order has been placed that the proper customer experience message is displaying.
4. Confirm that appears as declined in the Agent Web Console.

Review

1. Add JohnDoeReview@Acme.com email address to the VIP Decline list in the Kount Test environment.
2. Place a test order using any item, via the front end of the test website.
 - a) When filling out the customer information, please use JohnDoeReview@Acme.com as the email address.
 - b) Fill in necessary Test Credit Card information.
3. Confirm that after the order has been placed that the proper customer experience message is displaying.
4. Confirm that appears as declined in the Agent Web Console.

Note: *Merchants are responsible for checking interaction with their OMS to validate expected behavior.*

Optional Test Scenarios

Shipping/Billing Address Test Scenario

1. Add JohnDoeReview@Acme.com email address to the VIP Review list in the Kount Test environment.
2. Place a test order using any item, via the front end of the test website.
 - a) When filling out the customer information, please use JohnDoeReview@Acme.com as the email address.
 - b) Enter 1234 Main Street, Any Town, ID, USA 83705 for the Shipping Address.
 - c) Enter 5678 Oak Street, Any Town, ID 83705 for the Billing Address.
 - d) Fill in necessary Test Credit Card information.
3. Confirm that after the order has been placed that the proper customer experience message is displaying.
4. Confirm that the order is held as review in the Agent Web Console.
5. Confirm the Shipping Address passed correctly.
6. Confirm the Billing Address passed correctly.

Shipping Phone/Billing Phone Test Scenario

1. Add JohnDoeReview@Acme.com email address to the VIP Review list in the Kount Test environment.
2. Place a test order using any item, via the front end of the test website.
 - a) When filling out the customer information, please use JohnDoeReview@Acme.com as the email address.
 - b) Enter 123-456-7890 for the Shipping Phone Number.
 - c) Enter 102-345-6789 for the Billing Phone Number.
 - d) Fill in necessary Test Credit Card information.
3. Confirm that after the order has been placed that the proper customer experience message is displaying.
4. Confirm that the order is held as review in the Agent Web Console.
5. Confirm the Shipping Phone passed correctly.
6. Confirm the Billing Phone passed correctly.

Order Number Test Scenario

1. Add JohnDoeReview@Acme.com email address to the VIP Review list in the Kount Test environment.
2. Place a test order using any item, via the front end of the test website.
 - a) When filling out the customer information, please use JohnDoeReview@Acme.com as the email address.
3. Confirm that after the order has been placed that the proper customer experience message is displaying.
4. Confirm that the order is held as review in the Agent Web Console.
5. Confirm that the order number has populated correctly.

Note: Merchants are responsible for validating proper process management and testing interaction with their OMS to validate expected behavior.

Appendix F

Integrating Kount with a Custom Payment Method

Note: this documentation is meant for developers. You should be familiar with extending Magento functionality in order to integrate Kount with a custom payment method.

Using Standard Magento Payment Workflow

Kount works with all default Magento payment methods out of the box. The Kount extension collects all necessary data except payment (credit card) data. By default, it sets a “no payment” flag.

To send additional payment data to Kount, you need to add a custom RIS payment builder. There are two payment builders available:

- **Inquiry builder:** used for initial requests to Kount in both workflows (Pre-Authorization and Post-Authorization).
- **Update builder:** used for updating a Kount account after a successful or failed order is placed. Should only be used with Pre-Authorization workflow.

Both RIS payment builders must implement the `\Swarming\Kount\Model\Ris\Base\Builder\PaymentInterface` interface. It has one method, `process()`, which expects the RIS request and order payment objects as parameters.

The second step is an addition of these builders into RIS inquiry and update builders via dependency injection (DI).

```
<virtualType name="Swarming\Kount\Model\Ris\Inquiry\Builder\Payment"
type="Swarming\Kount\Model\Ris\Base\Builder\Payment">
    <arguments>
        <argument name="payments" xsi:type="array">
            <item name="custom_payment_code" xsi:type="string">Custom\Inquiry\Payment\Builder</item>
        </argument>
    </arguments>
</virtualType>
<virtualType name="Swarming\Kount\Model\Ris\Update\Builder\Payment"
type="Swarming\Kount\Model\Ris\Base\Builder\Payment">
    <arguments>
        <argument name="payments" xsi:type="array">
            <item name="custom_payment_code" xsi:type="string">Custom\Update\Payment\Builder</item>
        </argument>
    </arguments>
</virtualType>
```

Payment information which could be added to the RIS request:

```
$request->setCardPayment() // card number
$request->setExpirationYear(); // expiration year
$request->setExpirationMonth(); //expiration month
$request->setAvst(); // AVS street address reply from payment
$request->setAvsz(); // AVS zip code reply from payment
$request->setCvvr(); //CVV/CVC/CVV2 reply from payment
```

Using Non-Standard Magento Workflow

Examples of non-standard workflow include Authorize.Net DirectPost and PayPal Express Checkout.

How Does the Kount Extension Work?

The extension defines a workflow interface, implemented for Pre-Authorization and Post-Authorization workflows.

```
interface WorkflowInterface
{
    /**
     * @param \Magento\Sales\Model\Order\Payment $payment
     * @throws LocalizedException
     */
    public function start(Payment $payment);

    /**
     * @param \Magento\Sales\Model\Order $order
     * @throws LocalizedException
     */
    public function failure(Order $order);

    /**
     * @param \Magento\Sales\Model\Order $order
     * @throws LocalizedException
     */
    public function success(Order $order);
}
```

The **start** method is called on the sales_order_payment_place_start event.

The **success** method is called on the checkout_submit_all_after event.

The **failure** method is called on the sales_model_service_quote_submit_failure event.

How to Disable Calling Workflow Methods on Default Events

It could be necessary to disable calling workflow methods on the standard events, and instead call them during other events.

Each observer for these events has a dependency on `\Swarming\Kount\Model\Observer\ConditionInterface`. The default condition allows to execute an observer, and it's possible to configure conditions for each payment method via dependency injection. An example of this is done for `Authorize.Net DirectPost` on the `checkout_submit_all_after` event:

```
<virtualType name="Swarming\Kount\Model\Observer\SubmitAllAfter\Condition"
type="Swarming\Kount\Model\Observer\Condition">
  <arguments>
    <argument name="conditions" xsi:type="array">
      <item name="authorizenet_directpost"
xsi:type="string">Swarming\Kount\Model\Observer\Condition\Negative</item>
      </argument>
    </arguments>
  </virtualType>
```

This disables the workflow method call on the `checkout_submit_all_after` event.

There are two already-implemented conditions in the extension, `Negative` and `Positive`, which could be used for enabling/disabling the calling of methods on events. If you wish to use any custom logic to determine the positive or negative conditional, simply create your own implementation of `\Swarming\Kount\Model\Observer\ConditionInterface`.

Calling Workflow Methods on Custom Events

An example of calling workflow methods on custom events can be seen in the `\Swarming\Kount\Plugin\PayPal\Controller\Payflow\ReturnUrl` class.

You will need to add three dependencies to the custom observer:

```
/**
 * @var \Swarming\Kount\Helper\Workflow
 */
protected $helperWorkflow;

/**
 * @var \Swarming\Kount\Model\Config\Workflow
 */
protected $configWorkflow;

/**
 * @var \Swarming\Kount\Model\WorkflowFactory
 */
protected $workflowFactory;
```

And use them as such:

```
if (!$this->helperWorkflow->isProcessable($order)) {
    return $result;
}

$workFlowType = $this->configWorkflow->getWorkflowMode($order->getStoreId());
$workFlow = $this->workflowFactory->create($workFlowType);

$workFlow->success($order); // or $workFlow->failure($order);
```

The **success()** and **failure()** methods require a `\Magento\Sales\Model\Order` `$order` object as a parameter. The **start()** method requires a `\Magento\Sales\Model\Order\Payment` `$payment` object as a parameter.

Keep in mind that the custom event must call the workflow's methods for a specific payment method only. It's possible to reuse conditional logic with a default false condition, as it's done for \Swarming\Kount\Observer\PaymentPlaceEnd observer (customization for Authorize.Net DirectPost):

```
<virtualType name="Swarming\Kount\Model\Observer\PaymentPlaceEnd\Condition"
type="Swarming\Kount\Model\Observer\Condition">
  <arguments>
    <argument name="default" xsi:type="boolean">false</argument>
    <argument name="conditions" xsi:type="array">
      <item name="authorizenet_directpost"
xsi:type="string">Swarming\Kount\Observer\PaymentPlaceEnd\Condition\AuthorizeDirectPost</item>
      </argument>
    </arguments>
  </virtualType>
```



Kount[®]

Copyright © 2008-2016

By Kount Inc.

All Rights Reserved