ACI Integration Guide

**Document Revision History**

| **Revision** | **Author** | **Date** | **Status and Description** |
| --- | --- | --- | --- |
| 19.1.0 | Ganga Sukumaran | 08-05-2018 | Initial version |
| 21.1.0 | Aneesh Sudhakaran | 01-12-2020 | * Updated as part of recertification changes * 3DSecure changes |
| 21.2.0 | Sourabh Jacob | 24-06-2021 | * Replaced deprecated job step with custom job step * Updated cartridge to work with SFRA v6 |
| 24.0.0 | Brinsil Saju | 01-04-2024 | * Upgraded the ACI cartridge to support SFRA version 6.3.0 * Updated ACI node modules to their latest supported version |

Contents

[1 Overview 5](#_Toc58259545)

[2 Features 5](#_Toc58259546)

[3 Compatibility 6](#_Toc58259547)

[4 Installation 6](#_Toc58259548)

[4.1 Cartridges 6](#_Toc58259549)

[4.2 BM Imports 6](#_Toc58259550)

[4.2.1 Cartridge path 6](#_Toc58259551)

[4.2.2 Metadata Import 6](#_Toc58259552)

[4.2.3 Services Import 6](#_Toc58259553)

[4.2.4 Payment Processor 7](#_Toc58259554)

[4.2.5 Jobs Import 7](#_Toc58259555)

[4.2.6 Content Library Import 7](#_Toc58259556)

[5 Configurations 8](#_Toc58259557)

[5.1 Site Preferences 8](#_Toc58259558)

[5.1.1 ACI 8](#_Toc58259559)

[5.1.2 ACI Merchant Configurations 9](#_Toc58259560)

[5.2 Configure Shipping Method Custom Attribute 9](#_Toc58259561)

[5.3 Configuring Payment Methods 10](#_Toc58259562)

[5.3.1 Configuring Credit Cards 10](#_Toc58259563)

[5.3.2 Configure PayPal 11](#_Toc58259564)

[5.3.3 Configuring additional payment methods 12](#_Toc58259565)

[5.3.4 Available payment methods 12](#_Toc58259566)

[6 Cartridge Extensions 14](#_Toc58259567)

[6.1 Sending additional parameters to ACI 14](#_Toc58259568)

[6.1.1 Custom Parameters collected from browser 14](#_Toc58259569)

[6.1.2 Custom Risk Parameters 14](#_Toc58259570)

[6.1.3 Additional custom parameters 14](#_Toc58259571)

[6.2 Backoffice operations 15](#_Toc58259572)

[6.2.1 Capture 15](#_Toc58259573)

[6.2.2 Refund 16](#_Toc58259574)

[6.2.3 Reversal 16](#_Toc58259575)

[6.3 Widget Customization 17](#_Toc58259576)

[6.3.1 Extending Java script functionality 17](#_Toc58259577)

[6.3.2 Modify widget styling 17](#_Toc58259578)

[7 Operations, Maintenance 18](#_Toc58259579)

[7.1 Availability 18](#_Toc58259580)

[7.2 Failover / Recovery Process 18](#_Toc58259581)

[7.3 Support 18](#_Toc58259582)

[8 Known Issues 18](#_Toc58259583)

[8.1 Version Compatibility issues 18](#_Toc58259584)

[9 Appendix 19](#_Toc58259585)

[9.1 System Extensions 19](#_Toc58259586)

[9.1.1 Order 19](#_Toc58259587)

[9.1.2 PaymentMethod 19](#_Toc58259588)

[9.1.3 PaymentTransaction 19](#_Toc58259589)

[9.1.4 ShippingMethod 19](#_Toc58259590)

[9.1.5 CustomerPaymentInstrument 20](#_Toc58259591)

# Overview

The ACI payment cartridge enables a commerce cloud store to integrate with ACI payment service. This document provides detailed instructions on installing ACI payment cartridge onto a commerce cloud store.

The integration is based on the SFRA demo store, on version 6.3.0, provided by SFCC.

The integration consists of an archive with contents as described in the below table.

|  |  |
| --- | --- |
| **Name** | **Purpose** |
| cartridges | Contains ACI integration cartridges |
| cartridges\int\_aci\_sfra | This cartridge contains the SFRA specific changes required for ACI integration |
| cartridges\int\_aci\_core | This cartridge contains the API calls for ACI integration |
| Site-template | Contains custom attributes and settings required for the integration |
| Documentation | Contains this document “ACI Integration Guide” |

# Features

The integration described in this document supports the following features:

* Hosted payment page checkout for payment methods offered by ACI using the ACI CopyAndPay widget
* Ability to manage payment methods offered by ACI via BM
* Supports both synchronous and asynchronous payment methods
* Ability to capture payment immediately during order placement or to authorise payment first and capture payment later
* Code plugins for refund / capture / reversal operations
* Ability to save cards to account and use them later during checkout
* Allows business users to modify the ACI payment widget styling
* Allows business users to modify the javascript functionality that is provided with the cartridge
* Ability to specify custom risk parameters to be sent to ACI
* Ability to hold orders which needs manual review in pending status and release or cancel them later based on review

# Compatibility

Available since Commerce Cloud Platform Release 24.4, SFRA 6.3.0. The cartridge was tested against compatibility mode 22.7

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

# Installation

## Cartridges

Cartridges relevant for SFRA version are int\_aci\_sfra and int\_aci\_core

## BM Imports

### Cartridge path

Add the cartridges shown below to the beginning of the cartridge path

|  |  |
| --- | --- |
| Site Cartridge Path | int\_aci\_sfra:int\_aci\_core |
| BM Cartridge Path | int\_aci\_core |

### Metadata Import

Locate the folder site-template in the installation package and perform following steps.

* Review the contents in the site-template folder.
* Change the site ID under path ***site-template\sites*** to your site ID.
* Change the library ID under path ***site-template\libraries*** to your library ID
* Archive the folder to ‘site-template.zip’ and import the file via Site Import & Export

### Services Import

A new service, “http.aciworldwide.checkout”, has been created as part of the integration.

Import File: ***site-template\services.xml***

Please set the urls as shown below based on the instance where the cartridge is being integrated

|  |  |
| --- | --- |
| **Environment** | **URL** |
| Non-production instances | <https://test.oppwa.com/v1> |
| Production instances | <https://oppwa.com/> |

### Payment Processor

A new payment processor named ACI has been created as part of the cartridge.

### Jobs Import

Following jobs have been developed as part of cartridge.

|  |  |
| --- | --- |
| **Job Name** | **Job Description** |
| ACI Pending Transactions | The job will periodically check the status of all pending transactions in ACI and update the order export status or cancel the order depending on the transaction status in ACI.  *NOTE: Please change the site-id to the correct site id before importing. As this is scheduled job, please check, and update the frequency as per merchant requirements.* |

Import File: ***site-template\jobs.xml***

### Content Library Import

*NOTE: Please change the library id in the file to the correct library id before importing.*

# Configurations

## Site Preferences

Two site preference groups have been created for this cartridge

### ACI

|  |  |
| --- | --- |
| **Name** | **Purpose** |
| ACI Enabled/Disabled | Checking a flag for ACI payment method is enabled or Disabled |
| ACI API Version | Version of ACI API. Defaulted to v1. This value needs to be changed when there is a version change in ACI API |
| ACI Bearer Token | Authentication token provided by ACI |
| Allow customers to save card during checkout? | Checking this flag will allow customers to save cards during checkout |
| Capture on order confirmation? | Determines if the payment needs to be captured during order placement or to only pre-auth a payment during order placement. |
| ACI Entity ID | Entity ID for ACI authentication |
| ACI ForceResultCode | Set this value to force ACI to return specific error codes for example, 800.100.100. This is only for testing purposes and should not be set in LIVE environment |
| ACI Transaction Mode | Set this value to "INTERNAL" or EXTERNAL in all non-production instances and to "LIVE" in production instance. |
| Summary page in checkout flow? | Not applicable for SFRA version. This flag is used in SiteGenesis version |
| ACI Widget URL | ACI CopyPay Widget URL, for example - <https://test.oppwa.com/v1> for test environment |
| ACI Notification Emails | Notification emails in case of failure while reversing transaction in place order flow. Multiple emails IDs will be comma separated |
| ACS transaction ID | This field contains a universally unique transaction identifier assigned by the ACS to identify a single transaction. |
| Authentication Type | The type of authentication that was requested by the ACS. |
| 3DS Requestor Challenge Indicator | Indicates whether a challenge is requested for this transaction. |
| ACS Challenge mandate indicator | Indication of if a challenge is required for the transaction to be authorized due to local/regional mandates or other variable. |
| Directory Server transaction ID | Transaction ID assigned by the directory server |
| Exemption flag | Flags the transaction as exemption during authorization. Possible values: |
| Transaction status reason | Provides information on why the Transaction Status field has the specified value. |
| 3D Secure version | Version of 3D (examples: 1.0.2, 2.1.0, 2.2.0 ) |

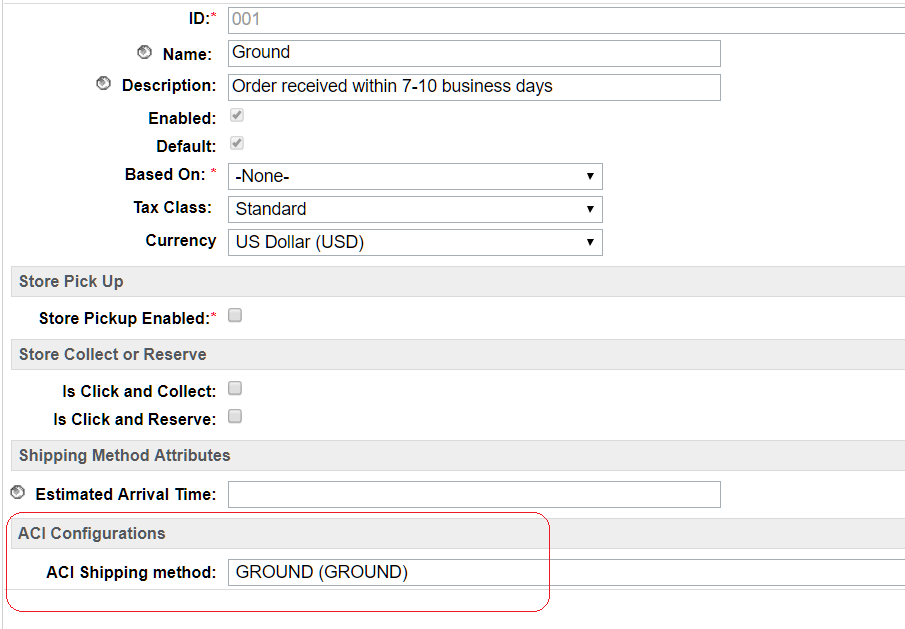
### ACI Merchant Configurations

Merchant configurations can be set to override the information that is shown on the cardholder statement. Setting this is not mandatory and it needs to be set only if the details about the client needs to be overridden in cardholder statement.

|  |  |
| --- | --- |
| **Name** | **Purpose** |
| ACI Merchant Country | The country of the merchant. This can be configured to override the merchant country shown in the card holder statement in ACI |
| ACI Merchant City | The city of the merchant. This can be configured to override the merchant city of the card holder statement in ACI |
| ACI Merchant Category code | The merchants's category code. This can be configured to override the merchant category code shown in the card holder statement in ACI |
| ACI Merchant Name | Name of the merchant. This can be configured to override the merchant name of the card holder statement in ACI |
| ACI Merchant Phone | The merchants's phone number. This can be configured to override the merchant phone number shown in the card holder statement in ACI |
| ACI Merchant PostCode | The postal code or zip code of the merchant. This can be configured to override the merchant postcard of the card holder statement in ACI |
| ACI Merchant State | The county, state or region of the merchant. This field can be used to override the information that is shown on the cardholder statement |
| ACI Merchant Street | The building number and/or street name of the merchant. This can be configured to override the merchant address of the card holder statement in ACI |

## Configure Shipping Method Custom Attribute

A custom attribute, ACI\_ShippingMethod, of type “Enum of Strings” is added to Shipping Method object. Each of the shipping methods configured in the site needs to be mapped to one of the values in the list as this is required in the API call to ACI as mentioned in ACI documentation - <https://docs.oppwa.com/reference/parameters#shipping-address>

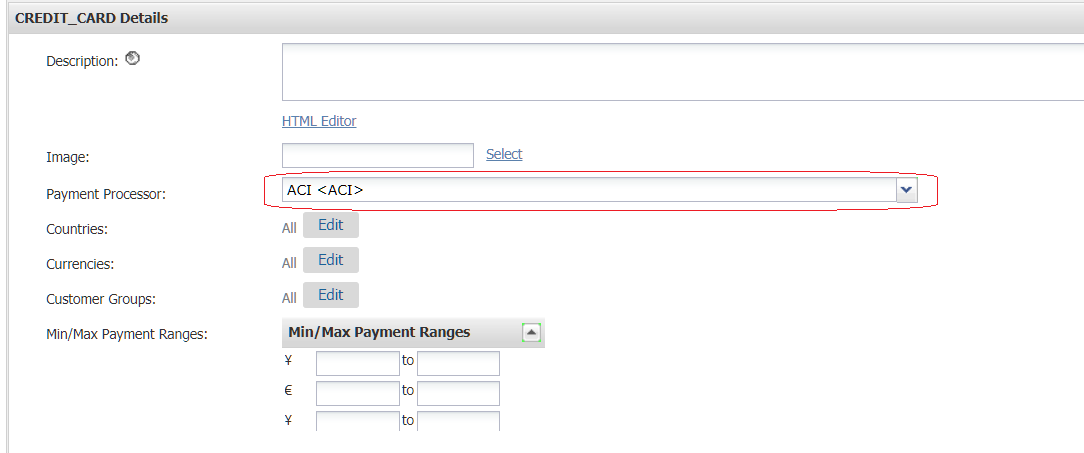


## Configuring Payment Methods

### Configuring Credit Cards

Please find below the steps to configure CREDIT\_CARD payment method for ACI.

* Change the payment processor of the payment method to ACI

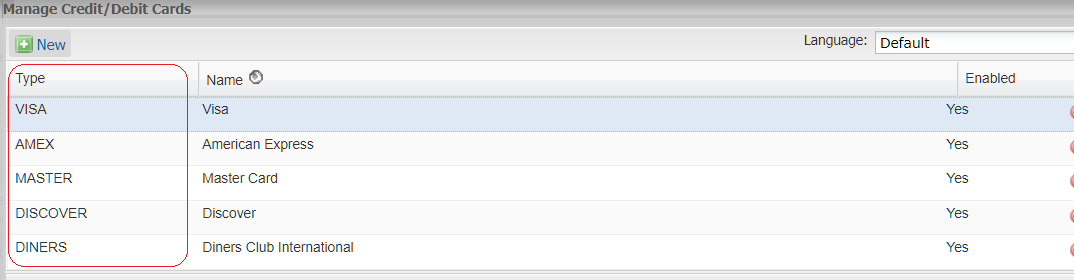


#### Configure card types

The card type of credit or debit cards must match the Brand ID specified in the ACI documentation - <https://docs.oppwa.com/tutorials/integration-guide>

Please follow below steps to configure the card types supported by ACI.

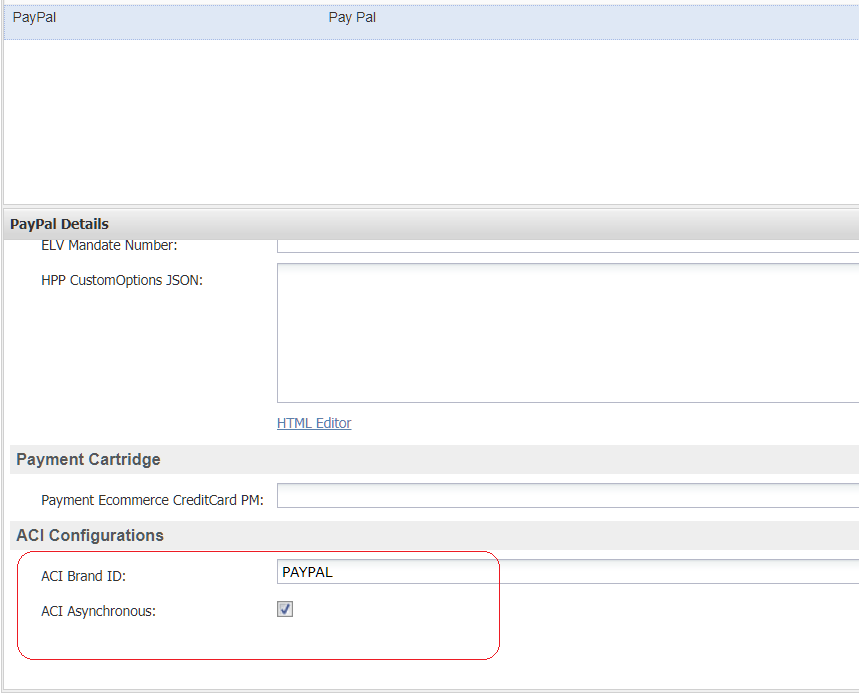
* Update the card type of existing cards to the Brand ID specified in ACI documentation - <https://docs.oppwa.com/tutorials/integration-guide>
* While adding new cards, ensure the card type matches the Brand ID required by ACI
* All asynchronous card types must be configured as separate individual payment methods



### Configure PayPal

Please follow the below steps to configure PayPal payment method to integrate with ACI

* Change the payment processor of the payment method to ACI
* Update the ACI Brand ID to PAYPAL
* Check the ACI Asynchronous flag to mark the payment method as asynchronous



### Configuring additional payment methods

Please follow the below steps to configure additional payment methods offered by ACI

* Change the payment processor of the payment method to ACI
* Update the ACI Brand ID to the brand id specified in ACI documentation. Please refer <https://docs.oppwa.com/reference/brands-reference> for more details.
* Update the ACI Asynchronous flag, if the payment method is asynchronous. Please refer <https://docs.oppwa.com/reference/brands-reference> for more details.

### Available payment methods

Please follow the below steps to configure additional payment methods offered by ACI

* Credit Card
* PayPal

Please follow the below documentation to configure additional payment methods offered by ACI.

ACI documentation - <https://docs.oppwa.com/tutorials/integration-guide>

# Cartridge Extensions

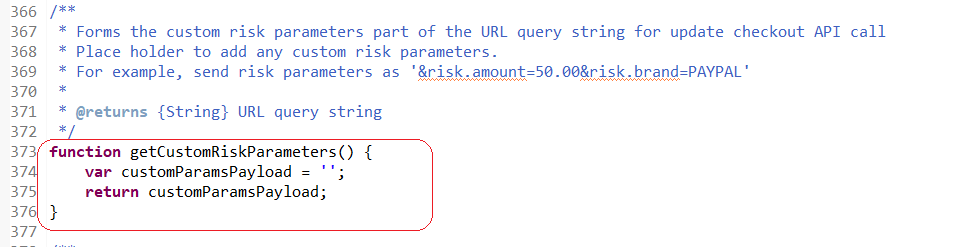
## Sending additional parameters to ACI

### Custom Parameters collected from browser

Any custom parameters collected from browser must be prefixed with “SHOPPER\_” before passing to ACI. For example, to send a custom parameter “taxType”, it must be prefixed as “SHOPPER\_taxType” and send in the form as “customParameters[SHOPPER\_taxType]” to ACI

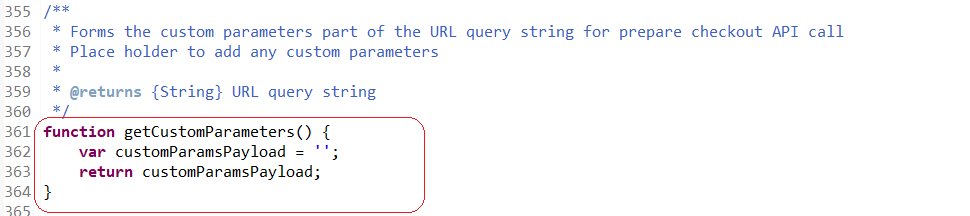
### Custom Risk Parameters

Script ACIServiceWrapper.js in path int\_aci\_core/cartridge/scripts/aci wraps all the changes required to make a call to the ACI APIs. Function getCustomRiskParameters () can be modified to send any additional risk parameters configured by merchants to ACI.



### Additional custom parameters

Function getCustomParameters() in script ACIServiceWrapper.js can be modified to send any additional parameters to ACI.



## Backoffice operations

The cartridge ‘**int\_ACI\_SFRA\_test’** has implementation for the following API methods – Capture, Refund and Reversals. They are optional and requires custom integration. The implementation is available in script ***TransactionHelper.js*** in path ***int\_aci\_core/cartridge/scripts/aci/***

When site cartridge path has int\_ACI\_SFRA\_test  configured within it, then the back office test page will be accessible via controller ACITransactionSimulator-Test. But if site cartridge path does not have int\_ACI\_SFRA\_test configured within it, then the back office test page will not be accessible via controller ACITransactionSimulator-Test. An OOB error page would be thrown. Also when site cartridge path has int\_ACI\_SFRA\_test  configured within it, and the SFCC instance in Production, then the back office test page will not be accessible via controller ACITransactionSimulator-Test. Customer will be redirected to Error page.

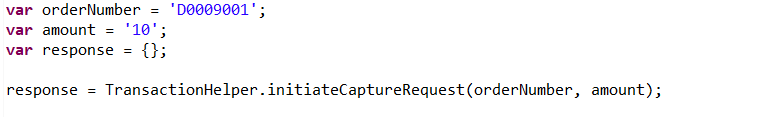
### Capture

The implementation supports full capture and partial captures. The function ***TransactionHelper.initiateCaptureRequest*** has the code required to handle capture requests. The function accepts the order number and the amount to be captured as arguments.

The function returns the below error codes:

|  |  |  |
| --- | --- | --- |
| **Error Code** | **Reason** | **Handling Error** |
| UNABLE\_TO\_FIND\_ORDER | Returned when not able to find an order corresponding to the order number passed in the argument. | Retry the operation with correct order number |
| ACI\_CAPTURE\_ERROR | Returned in one of the following scenarios:   * Invalid Capture Amount – amount to be captured exceeds order total plus already captured amount * ACI Service unavailable * ACI Service returned error response | Check the error log and ACI service logs and take appropriate action. |
| UNABLE\_TO\_FIND\_TRANSACTION | Returned when there is no transaction ID associated with the order | Manually capture the payment via ACI console |
| GENERAL\_CAPTURE\_ERROR | Returned in case of any exceptions other than above errors | Check the error log and ACI service logs and take appropriate action. |

Sample code to invoke capture request:



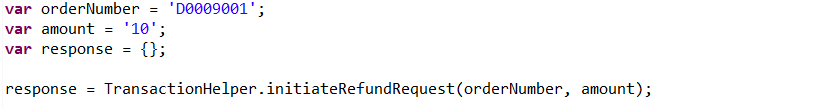
### Refund

The implementation supports full refund and partial refunds. The function ***TransactionHelper.initiateRefundRequest*** has the code required to handle refund requests. The function accepts the order number and the amount to be refunded as arguments.

The function returns the below error codes:

|  |  |  |
| --- | --- | --- |
| **Error Code** | **Reason** | **Handling Error** |
| UNABLE\_TO\_FIND\_ORDER | Returned when not able to find an order corresponding to the order number passed in the argument. | Retry the operation with correct order number |
| ACI\_REFUND\_ERROR | Returned in one of the following scenarios:   * Invalid refund amount – amount to be refunded exceeds order total plus already refunded amount * ACI Service unavailable * ACI Service returned error response | Check the error log and ACI service logs and take appropriate action. |
| UNABLE\_TO\_FIND\_TRANSACTION | Returned when there is no transaction ID associated with the order | Manually refund the payment via ACI console |
| GENERAL\_REFUND\_ERROR | Returned in case of any exceptions other than above errors | Check the error log and ACI service logs and take appropriate action. |

Sample code to invoke refund request:



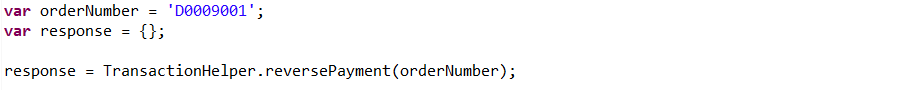
### Reversal

The implementation supports full reversal. The function ***TransactionHelper.reversePayment*** has the code required to handle reversal requests. The function accepts the order number as argument.

The function returns the below error codes:

|  |  |  |
| --- | --- | --- |
| **Error Code** | **Reason** | **Handling Error** |
| UNABLE\_TO\_FIND\_ORDER | Returned when not able to find an order corresponding to the order number passed in the argument. | Retry the operation with correct order number |
| ACI\_REVERSAL\_ERROR | Returned in one of the following scenarios:   * ACI Service unavailable * ACI Service returned error response | Check the error log and ACI service logs and take appropriate action. |
| UNABLE\_TO\_FIND\_TRANSACTION | Returned when there is no transaction ID associated with the order | Manually reverse the payment via ACI console |
| GENERAL\_REVERSAL\_ERROR | Returned in case of any exceptions other than above errors | Check the error log and ACI service logs and take appropriate action. |

Sample code to invoke reversal request:



## Widget Customization

### Extending Java script functionality

Script ***aciwidget.js*** in path ***LINK\_aci\cartridge\client\default\js\aci\aciwidget.js*** contains the java script functionality required for COPYPAY payment widget. The script can be modified if required to add or change any widget specific script changes explained in <https://docs.oppwa.com/tutorials/integration-guide/advanced-options>

### Modify widget styling

SCSS file, ***aciwidget-style.scss***, in path ***LINK\_aci\cartridge\client\default\scss\aciwidget-style.scss*** can be modified to make any changes to the widget styling as explained in <https://docs.oppwa.com/tutorials/integration-guide/customisation#cssreference>

# Operations, Maintenance

## Availability

In case of problems with the connection to ACI Payments please contact the ACI Payments support department.

Please supply as much information as possible (Merchant account, time, order number, ACI Widget ID etc.).

## Failover / Recovery Process

In case the ACI service is unavailable, the user will not be able to checkout using ACI payment methods.

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

## Support

In case of problems with the integration, missing features, etc. please contact the ACI Payments support department or your ACI account manager.

# Known Issues

## Version Compatibility issues

The cartridge was tested on versions 3.30 and version 4.0.0 and was found to have compatibility issues in version 3.3.0. For versions 3.3.0 or below the files in folder LINK\_aci must be replaced with the below files.

* LINK\_aci\cartridge\templates\default\checkout\billing\paymentOptions.isml



* LINK\_aci\cartridge\controllers\CheckoutServices.js



# Appendix

## System Extensions

In addition to the changes mentioned above, the below system extensions were done as part of the cartridge.

### Order

|  |  |
| --- | --- |
| **Name** | **Purpose** |
| ACI Payment Response | Saves the response returned as part of ACI Payment Status API call |
| Is ACI payment in pending status | Marks the order as pending if the transaction response from ACI is in pending status or requires manual review |

### PaymentMethod

|  |  |
| --- | --- |
| **Name** | **Purpose** |
| ACI Brand ID | Indicates the brand id of the ACI payment method. Must match the value on ACI documentation. Please refer https://docs.oppwa.com/tutorials/integration-guide for more details |
| ACI Asynchronous | Indicates if the payment method is synchronous or asynchronous |

### PaymentTransaction

|  |  |
| --- | --- |
| **Name** | **Purpose** |
| ACI Transaction Status Flow | Holds the transaction status flow for ACI payment transactions |

### ShippingMethod

|  |  |
| --- | --- |
| **Name** | **Purpose** |
| ACI Shipping method | Parameter to be passed in the ACI PrepareCheckout request |

The following provides the list of possible values for the custom attribute under Shipping Method as mentioned in ACI documentation in <https://docs.oppwa.com/reference/parameters#shipping-address> :

* LOWEST\_COST
* CARRIER\_DESIGNATED\_BY\_CUSTOMER
* ELECTRONIC\_DELIVERY
* GROUND
* INTERNATIONAL
* MILITARY
* NEXT\_DAY\_OVERNIGHT
* OTHER
* STORE\_PICKUP
* SAME\_DAY\_SERVICE
* TWO\_DAY\_SERVICE
* THREE\_DAY\_SERVICE

### CustomerPaymentInstrument

|  |  |
| --- | --- |
| **Name** | **Purpose** |
| ACI Registration ID | Registration ID to be used for saving customer cards to account |