**Stripe Integration Guide**

Version 16.1.0

****



Table of Contents

[1. Summary 1-3](#_Toc280189026)

[2. Component Overview 2-4](#_Toc280189027)

[2.1 Functional Overview 2-4](#_Toc280189028)

[2.2 Use Cases 2-4](#_Toc280189029)

[2.3 Limitations, Constraints 2-4](#_Toc280189030)

[2.4 Compatibility 2-4](#_Toc280189031)

[2.5 Privacy, Payment 2-4](#_Toc280189032)

[3. Implementation Guide 3-5](#_Toc280189033)

[3.1 Setup 3-5](#_Toc280189034)

[3.2 Configuration 3-5](#_Toc280189035)

[3.3 Custom Code 3-5](#_Toc280189036)

[3.4 External Interfaces 3-5](#_Toc280189037)

[3.5 Testing 3-5](#_Toc280189038)

[4. Operations, Maintenance 4-6](#_Toc280189039)

[4.1 Data Storage 4-6](#_Toc280189040)

[4.2 Availability 4-6](#_Toc280189041)

[4.3 Support 4-6](#_Toc280189042)

[5. User Guide 5-7](#_Toc280189043)

[5.1 Roles, Responsibilities 5-7](#_Toc280189044)

[5.2 Business Manager 5-7](#_Toc280189045)

[5.3 Storefront Functionality 5-7](#_Toc280189046)

[6. Known Issues 6-8](#_Toc280189047)

[7. Release History 7-9](#_Toc280189048)

# Summary

* <CHAPTER NOT TO EXCEED ONE PAGE>
* <WHAT THE COMPONENT DOES, E.G. INTEGRATION WITH A 3RD PARTY PAYMENT GATEWAY>
* <3RD PARTY PRODUCTS AND CONTRACTUAL REQUIREMENTS, E.G. WHAT PRODUCT AND PRODUCT VERSION THE COMPONENT INTEGRATES WITH, IF CUSTOMER IS REQUIRED TO CONTRACT WITH THE 3RD PARTY OR IF THEIR SERVICES ARE RESOLD BY DEMANDWARE, ANY ADDITIONAL COSTS THE CUSTOMER WILL BE CHARGED BY 3RD PARTY>
* <WHAT THE INTEGRATION ENCOMPASSES, E.G. DEPLOYMENT OF A GENERIC CARTRIDGE, MODIFICATION OF STOREFRONT CODE, NEW BUSINESS MANAGER MODULE, CERTIFICATION PROCESS>

The Stripe LINK Cartridge facilitates integration between a Demandware Storefront with Stripe Payment Services; including Stripe.js Credit Card tokenization, Payments, and Stripe’s RELAY services for supporting embedded eCommerce solutions on social channels.

Contracting with Stripe is required for Live Production deployment of the cartridge. Though the cartridge can be installed and tested using a freely available Stripe Test account at <https://dashboard.stripe.com>. Please contact your Stripe Implementation Consultant for help with taking your Stripe account Live.

Stripe LINK Cartridge integration requires the following steps:

* **Import System Object Metadata** for managing Stripe configurations and supporting data points
* **Import Services** for enabling external interfaces with Stripe API
* **Import and customize OCAPI settings** for enabling RELAY services
* **Schedule Job Process** for managing Product Feed
* **Apply Custom Code** to the DWRE Storefront
* **Configure the Storefront Settings** to connect to Stripe services and enable LINK cartridge integration to the storefront

# Component Overview

## Functional Overview

<BUSINESS/FUNCTIONAL OVERVIEW OF COMPONENT, INCLUDING FLOW CHARTS, DIAGRAMS WHERE APPLICABLE>

### *Stripe Tokenization and Charges*

Stripe Tokenization modifies default Demandware Credit Card collection and processing with Stripe.js technologies to tokenize the credit card data and process payments using the tokenized data. During the checkout process, the cartridge will tokenize any new cards entered by customers. This data is transformed into a Stripe Customer object and associated Source. At the point of purchase, the tokenized data is used to generate a Stripe Charge to validate the payment transaction. Registered Customers can manage (add, delete) Credit Cards to their storefront-connected Stripe Customer Account for re-use in subsequent storefront purchases.

### *Stripe RELAY*

Stripe RELAY LINK cartridge updates enable Stripe RELAY to interact with the Storefront to manage the checkout process from an external system. This includes allowing Stripe RELAY to create/modify/delete basket items, submit payments, and complete orders from social-embedded interfaces. For example, allowing customers to purchase goods from Twitter. Additionally, Stripe RELAY cartridge functionality includes the creation of a Google Product Feed that can be used to import into the Stripe Account.

## Use Cases

<USE CASES COVERED BY THE COMPONENT>

### *Stripe.js Tokenization*

When customers enter credit card information on the storefront, the card information is tokenized via Stripe.js via client > Stripe interactions. Unmasked credit card data is therefore never sent to the Demandware servers.

### Stripe Charges

System will create a Stripe Charge (authorize or capture, based on Business Manager configuration) based on a successfully created and submitted Basket. All Stripe Charges are created against a Stripe Customer ID and associated Source.

### AVS Auto-Fail Transactions

Site administrators

### OCAPI-based purchases (Stripe RELAY)

Customer can purchase Demandware-based products from Stripe RELAY enabled third party interfaces, including embedded products on Twitter.

## Limitations, Constraints

<WHICH PARTS OF THE 3RD PARTY OFFERING ARE NOT SUPPORTED>

<WHICH DEMANDWARE FEATURES OR CONFIGURATIONS ARE NOT SUPPORTED, E.G. MULTIPLE LOCALES, NON-US SHIPPING ADDRESSES, ORDERS WITH MORE THAN ONE SHIPPING ADDRESS, CATALOG SIZE LARGER THAN 500K>

<IF YOUR INTEGRATION REQUIRES A CUSTOM CATALOG FEED, DESCRIBE ANY LIMITATIONS ON PRODUCT ATTRIBUTES, SUCH AS NON-SUPPORTED HTML ATTRIBUTES, 0.00$ PRICES, ONLY ORDERABLE PRODUCTS EXCLUDING BACKORDER>

Stripe offers a number of standard services that are not supported by the cartridge. These include Subscriptions, Plans, and Coupons support.

The Google Product Feed support included in the cartridge only supports a limited number of product attributes, including variation attributes for Size and Color. Additional attribute support requires customizing the product feed output.

## Compatibility

<AVAILABLE SINCE DEMANDWARE X.Y.Z>

<DEMANDWARE VERSION REFERENCED FOR INTEGRATION INTO STOREFRONT (REFERENCE APPLICATION) AND USED IN SCREENSHOTS>

Available since Demandware Platform Release 16.8, Site Genesis 103.0.0

The cartridge is available for installations on storefronts that support both Pipeline and Controller SiteGenesis implemenations.

## Privacy, Payment

<INDICATE IF CUSTOMER PROFILE DATA IS BEING ACCESSED, IF CREDIT CARD DATA IS BEING PROCESSED OR STORED WITHIN DEMANDWARE>

No unmasked credit card data is stored within Demandware. The cartridge tokenizes all payment data via direct client > Stripe communications and obscures any sensitive CCD before it arrives on the Demandware servers. Similarly, all credit card data that is retrieved by Demandware from the Stripe servers is also masked and/or tokenized.

# Implementation Guide

## Setup

<LIST CARTRIDGES THAT ARE PART OF THE COMPONENT AND ENTRY POINTS/PARAMETERS (API) IN DETAIL>

<HIGHLIGHT WHAT CARTRIDGES NEED TO BE DEPLOYED TO PRODUCTION, WHAT ARE FOR TESTING, WHAT SHOW CUSTOM CODE EXAMPLES (E.G. FOR STOREFRONT INTEGRATION)>

The Stripe LINK Cartridge contains several cartridges that are required for full functionality. Additionally, Pipeline and Controller support is broken out into two separate cartridges, thereby facilitating the installation and use of one or the other models.

Import all three cartridges into UX studio and associate them with the Server Connection.

### *Cartridges*

#### int\_stripe

int\_stripe contains all the script logic, templates, resource properties, and front-end code required for the base Stripe storefront integration.

* js
  + pages/checkout/billing\_sample.js
  + pages/account\_sample.js
  + stripe.js
* scripts
  + hooks/afterPostPaymentInstrument.js
  + hooks/afterSetShippingAddress.js
  + hooks/authorize.js
  + hooks/authorizeCreditCard.js
  + payment/processor/STRIPE\_CREDIT.js
  + service/stripe.ds
  + service/stripeInit.ds
  + util/Resource\_sample.ds
  + hooks.json
  + stripeHelper.ds
* scss
  + \_stripe.scss
* static
  + stripe/jquery.payment.js
* templates
  + account/payment/makefdefault.isml
  + account/payment/paymentinstrumentdetails\_sample.isml
  + account/payment/paymentinstrumentlist\_sample.isml
  + checkout/billing/papymentmethods\_sample.isml
  + checkout/billing/stripe\_creditcard\_fields.isml
  + checkout/billing/stripe\_paymentmethods.isml
  + checkout/billing/stripecreditcardjson.isml
  + feed/displayproductfeed.isml
  + stripe/footerinclude.isml
  + resources/checkout.properties

#### int\_stripe\_controllers

int\_stripe\_controllers contains all the logic for managing the Stripe integration controller logic via Demandware’s Controllers model. int\_stripe\_controllers should be used in favor of int\_stripe\_pipelines if the storefront will be using the latest version of SiteGenesis.

* controllers
  + COBilling\_sample.js
  + COPlaceOrder\_sample.js
  + PaymentInstruments\_sample.js
  + Stripe.js

#### int\_stripe\_pipelines

int\_stripe\_pipelines contains all the logic for managing the Stripe integration controller logic via Demandware’s Pipelines controller model. int\_stripe\_pipelines can be used in case the integration requires the use of Pipelines for any reason.

Additional note: int\_stripe\_pipelines is required for the custom Product Feed job. The Demandware Platform, as of version 16.8, requires Pipeline code for custom scheduled jobs.

* pipelines
  + COBilling\_sample.xml
  + COPlaceOrder\_sample.xml
  + PaymentInstruments\_sample.xml
  + STRIPE\_CREDIT.xml
  + Stripe.xml
* scripts
  + feed/GenerateProductFeed.ds
  + pipelinescripts/AddCard.ds
  + pipelinescripts/AuthorizePayment.ds
  + pipelinescripts/DeleteCard.ds
  + pipelinescripts/DisplayProductFeeds.ds
  + pipelinescripts/FetchCards.ds
  + pipelinescripts/GetCustomerCreditCard.ds
  + pipelinescripts/IsStripeEnabled.ds
  + pipelinescripts/MakeDefault.ds
  + pipelinescripts/RefundCharge.ds

## Configuration

*<CONFIGURATION STEPS INCLUDING ASSIGNMENT OF CARTRIDGES TO SITE, CUSTOM PREFERENCES, SET UP OF JOB SCHEDULES, CONFIGURATION DIFFERENCES FOR DIFFERENT INSTANCE TYPES, IMPORT OF METADATA>*

### Assign Cartridges to Site(s)

#### Site Cartridge Assignment

1. Navigate to Administration > Sites > Manage Sites
2. Click on the Site Name for the Storefront Site that will gain Stripe Functionality
3. Select the “Settings” tab
4. Add “*int\_stripe*” to the **Cartridges:** path, separating each cartridge in the list with “:”
   1. For example, “*app\_storefront\_controllers:app\_storefront\_core:int\_stripe\_controllers:int\_stripe*”
5. Add either the “*int\_stripe\_controllers*” OR “*int\_stripe\_pipelines*” to the cartridge path
   1. Note that if both cartridges are added to the cartridge path then “*int\_stripe\_controllers*” code is executed when Stripe URLs are requested
6. Repeat steps 2 – 5 for each Storefront Site where Stripe will be implemented

#### Business Manager Cartridge Assignment

Stripe cartridges need to be assigned to the Business Manager Site only if the implementation makes use of the Product Feed for use with Stripe RELAY functionality. Note also that, at the time of this writing, the Pipelines cartridge – “int\_stripe\_pipelines” – is required for the custom Job Feed schedule rather than “int\_stripe\_controllers”.

1. Navigate to Administration > Sites > Manage Sites
2. Click on the Business Manager Site > “Manage the Business Manager site.” Link
3. Add “*int\_stripe:in\_stripe\_pipelines*” to the **Cartridges:** path, separating each cartridge in the list with “:”

### Import System Object Definitions

1. Login to Business Manager and navigate to **Administration > Site Development > Import & Export**
2. Navigate to **Import & Export > Import & Export Files** and click “Upload”
3. In the **Upload Import Files > Upload File** section, select *stripe\_metadata.xml* file from the metadata/ folder of the LINK Cartridge and click “Upload”
4. Return to the **Administration > Site Development > Import & Export** page
5. Click on “Import” in the **Import & Export > Meta Data** section
6. Select the radio button next to the *stripe\_metadata.xml* file and click “Next >>”
7. Once the XML validation completes, click “Import”
8. After the import has completed, a Success status will display in the Status section

### Import Services

1. Login to Business Manager and navigate to **Administration > Operations > Import & Export**
2. Navigate to **Import & Export > Import & Export Files** and click “Upload”
3. In the **Upload Import Files > Upload File** section, select *stripe\_services.xml* file from the metadata/ folder of the LINK Cartridge and click “Upload”
4. Return to the **Administration > Operations > Import & Export** page
5. Click on “Import” in the **Import & Export > Services** section
6. Select the radio button next to the *stripe\_services.xml* file and click “Next >>”
7. Once the XML validation completes, click “Import”
8. After the import has completed, a Success status will display in the Status section

## Custom Code

<IF STOREFRONT APPLICATION NEEDS TO BE MODIFIED, EXPLAIN STEPS RELATIVE TO REFERENCE APPLICATION>

<IF ADDITIONAL CUSTOM CODE NEEDS TO BE WRITTEN, EXPLAIN IN DETAIL AND PROVIDE EXAMPLE.>

*<ESTIMATED INTEGRATION EFFORTS>*

## External Interfaces

<INTERFACES TO EXTERNAL SERVICES, E.G HTTP CLIENT, WEBSERVICE CALLS, DESCRIPTION OF REQUESTS AND RESPONSES>

## Testing

*<SAMPLE DATA, TEST ACCOUNT, TEST CASES, REFERENCE TO CERTIFICATION REQUIREMENTS>*

# Operations, Maintenance

## Data Storage

*<DESCRIBE ANY DATA THAT WILL BE STORED WITHIN DEMANDWARE, E.G. CUSTOM OBJECTS, INCLUDING DURATION AND CLEANUP JOBS IF APPLICABLE>*

<IS THERE A SEPARATE DATA STORAGE OUTSIDE OF DEMANDWARE, SPECIFY LOCATION AND DURATION IF APPLICABLE>

## Availability

<EXPECTED AVAILABILITY /UPTIME OF ANY EXTERNAL SERVICE, INTERFACES>

<FALLBACK SOLUTION, BEHAVIOR IF EXTERNAL SERVICES ARE NOT AVAILABLE, IMPACT ON CUSTOMER STOREFRONT>

<ANY EXISTING UTILITIES THAT HELP TO DETECT AVAILABILITY/UPTIME OF EXTERNAL SERVICE, E.G. WEBSERVICE CALL, GOMEZ PING>

<ESTIMATED PERFORMANCE METRICS FOR PEAK BUSINESS HOURS IF AVAILABLE>

<NOTIFICATION PROCESS IF EXTERNAL SERVICES, INTERFACES ARE NOT RESPONDING, E.G. HOTLINE /SUPPORT PHONE NUMBER>

## Support

<CONTACT PERSON IN CASE DEFECT FIXES OR IMPROVEMENTS FOR COMPONENT ARE REQUIRED>

# User Guide

## Roles, Responsibilities

*<LIST RECURRING TASKS THAT NEED TO BE FULFILLED BY CUSTOMER, MERCHANT TO RUN THE INTEGRATION, E.G. MANUAL FEED OF CATALOG DATA INTO 3RD PARTY SERVICE, IF APPLICABLE>*

There are no recurring tasks required by the merchant. Once configurations and job schedules are set up, the functionality runs on demand.

## Business Manager

<UI SCREENSHOTS AND DESCRIBTION OF FUNCTIONALITY>

<DESCRIBE NEW BUSINESS MANAGER MODULES AND CONFIGURATION OPTIONS IF APPLICABLE>

## Storefront Functionality

*<DESCRIBE NEW STOREFRONT FUNCTIONALITY>*

### Credit Card Tokenization

Stripe.js credit card tokenization requires the inclusion of javascript on the payment forms, both during Checkout > Billing as well as My Account > Saved Payment Instruments. Additionally, the credit card ‘type’ form field are

### Saved Credit Cards

When an authenticated customer selects a saved credit card on the Checkout > Billing page, they will see a list of their Stripe-saved Sources as radio buttons rather than the default SiteGenesis <select/> options.

# Known Issues

*<LIST KNOWN ISSUES AND WORKAROUNDS>*

The LINK Cartridge has no known issues.

# Release History

<RELEASE HISTORY OF THE COMPONENT; THIS DOCUMENT IS PART OF THE COMPONENT AND DOES NOT HAVE ITS OWN RELEASE/VERSION NUMBER.>

*<PLEASE USE THE FOLLOWING NAMING SCHEMA: CHANGES TO THE 1ST DIGIT CONTAIN INCOMPATIBILITIES, CHANGES TO THE 2ND DIGIT PROVIDE NEW MINOR FEATURES WITHOUT CAUSING INCOMPATIBILITY, A CHANGE TO THE 3RD DIGIT PROVIDES BUG FIXES WITHOUT INTRODUCING NEW FEATURES>*

|  |  |  |
| --- | --- | --- |
| **Version** | **Date** | **Changes** |
| 16.1.0 | <DATE> | Initial release |