**Worldpay Integration**

Version 19.1.0



****

[1. SUMMARY 5](#_Toc517950198)

**2.** **[Changelog](#_Change_Log)………………………………………………………………………………………………………………………………………….6**

[3. COMPONENT OVERVIEW 7](#_Toc517950199)

[3.1 FUNCTIONAL OVERVIEW 7](#_Toc517950200)

[3.1.1 Payment Methods 7](#_Toc517950201)

[3.1.2 Order Notifications 8](#_Toc517950202)

[3.1.3 Order Inquiry job 11](#_Toc517950203)

[3.1.4 Job Failure Mailer Service 12](#_Toc517950204)

[3.1.5 APM Lookup Service 13](#_Toc517950205)

[3.1.6 Multiple Merchant ID Support 13](#_Toc517950206)

[3.1.7 Brazil Integration 14](#_Toc517950207)

[3.1.8 Order CleanUp Batch Job 14](#_Toc517950208)

[3.1.9 Initiate Cancel Order Job 16](#_Toc517950209)

[3.1.10 Capture Service 18](#_Toc517950210)

[3.2 LIMITATIONS, CONSTRAINTS 19](#_Toc517950211)

[3.3 COMPATIBILITY 21](#_Toc517950212)

[4. IMPLEMENTATION GUIDE 22](#_Toc517950213)

[4.1 SETUP 22](#_Toc517950215)

[4.2 INSTALLATION 24](#_Toc517950216)

[4.2.1 Adding the Cartridges in Salesforce Commerce Cloud Studio 24](#_Toc517950217)

[4.2.2 IP Address 24](#_Toc517950218)

[4.2.3 Activating the Cartridges in Business Manager 24](#_Toc517950219)

[4.3 CONFIGURATION 25](#_Toc517950220)

[4.3.1 Importing Meta Data 25](#_Toc517950221)

[4.3.2 Setting Worldpay Custom Site Preferences 28](#_Toc517950222)

[4.3.3 Create Worldpay Payment Processor 31](#_Toc517950223)

[4.3.4 Importing Worldpay Payment Methods 32](#_Toc517950224)

[4.3.5 Importing Job 34](#_Toc517950225)

[4.3.6 Importing Content Asset 35](#_Toc517950226)

[4.3.7 Importing Service 36](#_Toc517950227)

[4.3.8 IDEAL APM Integration 37](#_Toc517950228)

[4.3.9 BMultiple Merchant ID Support Integration 38](#_Toc517950229)

[4.3.10 Look and Feel Customization for Worldpay Redirect Pages 38](#_Toc517950230)

[4.3.11 Hosted Payment Pages iframe or lightbox for Worldpay Redirect Payment Method 39](#_Toc517950231)

[Country Currency Mapping 40](#_Toc517950232)

[4.3.12 40](#_Toc517950233)

[4.3.13 Country Currency and Locale Mapping for Klarna 41](#_Toc517950234)

[4.4 CUSTOM CODE 41](#_Toc517950235)

[4.4.1 Error Messaging 41](#_Toc517950236)

[4.4.2 Client Side Encryption 41](#_Toc517950237)

[4.4.3 Form File Changes 42](#_Toc517950238)

[4.5 CUSTOM CODE FOR CONTROLLERS 42](#_Toc517950239)

[4.5.1 Generic Section 42](#_Toc517950240)

[4.5.2 APM Lookup Support 66](#_Toc517950250)

[4.5.3 Brazil Integration 72](#_Toc517950251)

[4.5.4 Street Number Implementation 73](#_Toc517950252)

[4.5.5 Error Messaging 74](#_Toc517950253)

[4.5.6 Client Side Encryption 74](#_Toc517950254)

[4.5.7 Form File Changes 75](#_Toc517950255)

[4.5.8 Update shippingaddress.xml and billingaddress.xml 75](#_Toc517950256)

[4.5.9 Update the paymentinstrumentdetails.isml 75](#_Toc517950257)

[4.5.10 Update the locale.properties and countries.json 76](#_Toc517950258)

[4.6 EXTERNAL INTERFACES 76](#_Toc517950259)

[4.6.1 CHINA UNION PAY 82](#_Toc517950260)

[4.6.2 SOFORT 84](#_Toc517950261)

[4.6.3 ALIPAY 85](#_Toc517950262)

[4.6.4 YANDEX 86](#_Toc517950263)

[4.6.5 Googlepay 86](#_Toc517950263)

[4.6.6 QIWI 87](#_Toc517950264)

[4.6.7 BOLETO 89](#_Toc517950265)

[4.6.8 MISTER CASH 90](#_Toc517950266)

[4.6.9 PAYPAL 91](#_Toc517950267)

[4.6.10 eNETS 93](#_Toc517950268)

[4.6.11 iDEAL 94](#_Toc517950269)

[4.6.12 GiroPay 95](#_Toc517950270)

[4.6.13 Konbini 97](#_Toc517950271)

[4.6.14 Poli 100](#_Toc517950272)

[4.6.15 SEPA-DD 104](#_Toc517950273)

[4.6.16 Przeleway24 107](#_Toc517950274)

[4.6.17 CashU 108](#_Toc517950275)

[4.6.18 KLARNA 110](#_Toc517950276)

[5. PRODUCTION SETUP 123](#_Toc517950277)

[5.1 PRODUCTION SERVICE SETUP 123](#_Toc517950279)

[6. OPERATIONS, MAINTENANCE 124](#_Toc517950280)

[6.1 DATA STORAGE 124](#_Toc517950282)

[6.1.1 Order Level Attributes 124](#_Toc517950283)

[6.1.2 Order Payment Instrument Attributes 126](#_Toc517950284)

[6.1.3 Order Notifica2tion Custom Object 126](#_Toc517950285)

[6.2 AVAILABILITY 126](#_Toc517950286)

[6.3 SUPPORT 127](#_Toc517950287)

[7. USER GUIDE 128](#_Toc517950288)

[7.1 ROLES, RESPONSIBILITIES 128](#_Toc517950290)

[7.2 BUSINESS MANAGER 128](#_Toc517950291)

[7.3 STOREFRONT FUNCTIONALITY 128](#_Toc517950292)

[8. TESTING 129](#_Toc517950293)

[8.1 TEST CASES 129](#_Toc517950295)

[8.2 TEST DATA 129](#_Toc517950296)

[9. KNOWN ISSUES 133](#_Toc517950297)

[10. RELEASE HISTORY 134](#_Toc517950298)

[APPENDIX A: APM/Card mapping keys 135](#_Toc517950299)

[APPENDIX B: IDEAL Bank List 137](#_Toc517950300)

[APPENDIX C: Error Codes and Error Messages for Transaction Notification 137](#_Toc517950301)

[APPENDIX D: Order Notification and SFCC Order status mapping 138](#_Toc517950302)

[APPENDIX E: Error Codes and Error Messages 139](#_Toc517950303)

[APPENDIX F:Enabling and disabling worldpay Integration 140](#_Toc517950303)

# SUMMARY

This document provides technical overview and implementation details for each Worldpay service integrated within SFCC platform with MobileFirst Reference Application version. The Worldpay cartridges (link\_worldpay and link\_worldpay\_core) extends the functionality of reference eCommerce Storefront, enabling synchronous and asynchronous access to Worldpay payment transaction services listed below.

* Payment Methods:

1. Sofort
2. IDeal
3. Qiwi
4. Alipay
5. Mistercash
6. Yandex
7. Boleto
8. Paypal
9. China Union Pay
10. Enets
11. Giropay
12. Poli
13. Poli NZ
14. Konbini
15. Przelewy24
16. CashU
17. Sepa DD
18. Klarna
19. 3D/Non 3D Credit Cards (XML Direct with client side encryption and tokenzation features)
20. 3D/Non 3D Credit Cards (XML Redirect with tokenzation and hosted payment pages lightbox/iframe feature)
21. WeChat Pay
22. MOTO
23. Google Pay

* Order Notification Job
* Order Enquiry Job
* Order Cancel Or Refund Job for mac Issue identified orders
* Order CleanUp Batch Job
* APM Look Up Service
* Multiple Merchant ID support
* Statement Narrative support
* Brazil Integration

# Change Log

|  |  |  |
| --- | --- | --- |
| **Release Version** | **Date** | **Description** |
| 19.1.0 | Mar 20, 2019 | * Google Pay * Stored Credentials * Release 18.3.0 Feedback |
| 18.3.0 | Feb 18, 2019 | * WeChat implementation * Alternative payment method * Merchant token * MOTO payment method |
| 18.2.0 | Jul 16, 2018 | * Complete SFRA implementation |

# COMPONENT OVERVIEW



## FUNCTIONAL OVERVIEW

### Payment Methods

All the supported Payment Methods are broadly categorized into XML DIRECT and XML REDIRECT Model based upon the manner in which they are integrated.



#### XML Direct Request

Merchants who collect and store their shoppers' payment details on their own platform can use the XML Direct model as an effective payment-processing gateway. With this model, the merchant collects both order and payment details and then communicates the relevant payment details on a per order basis with Worldpay, for processing. The XML Direct model only allows for using a select number of on-line payment methods, where no consumer interaction is involved. In view of the cost involved in establishing appropriate security measures this model only applies for merchants with established high transaction volumes. Find below that list of supported Payment Methods that are XML DIRECT based:

1. Sofort
2. iDEAL
3. Qiwi
4. Alipay
5. Mistercash
6. Yandex
7. Boleto
8. Paypal
9. GiroPay
10. Poli
11. Poli NZ
12. Konbini
13. Przelewy24
14. CashU
15. SEPA DD
16. Credit Card- Direct (3D/Non 3D)
17. Klarna
18. WeChat Pay
19. MOTO
20. Google Pay

#### XML Redirect Request

Merchants can also redirect the users to Worldpay Hosted page to collect payment details. It allows for real-time processing of payments and ensures a maximum number of up-to-date payment methods. The Hosted Payment Page service is secure, provides Worldpay with required information to perform active fraud risk assessment and, is the fastest way to get up and running with on-line payments. Find below the list of supported Payment Methods that are XML REDIRECT based

1. China Union Pay
2. Enets
3. Credit Card – Redirect (3D/Non 3D)
4. Klarna-Hosted Payment Pages

**\*\* Refer to the Brazil Integration section for more details on Latin American Credit Cards and BOLETO**.

### Order Notifications

The link\_worldpay\_core cartridge provides a service to update all the SFCC Order status according to status changes of individual payments in Worldpay. (PUSH mechanism)

**Worldpay-Notify** is the secure route function exposed to Worldpay to receive Order Notifications i.e. a HTTPS message whenever a payment status changed at Worldpay. Following URL (**https://mechantsandbox/Worldpay-Notify**) has been exposed to Worldpay to send the notifications. SFCC will send an HTTP 200 [OK] response to Worldpay to acknowledge the receipt of notification. Corresponding to each Order Notification received a custom object is created with the Order Number and Order Notification XML captured. A SFCC Job is then run (manually or scheduled), to process those custom objects and update the Orders corresponding to each Order Notification. (Refer Section [2.1.2.1](#_Order_Notification_Job) for more job details)

If the Worldpay system does not receive an [OK] in reply to an HTTPS order notification it waits for approximately an hour to deliver the notification again. Then it will resend the order notification approximately every five minutes for a week until it is acknowledged. Worldpay places order notifications in a queue and the first notification in the queue will be resent until it is acknowledged. If acknowledged, or if after a week still no [OK] is returned, the retry mechanism stops sending the message and proceeds with the next order notification from the queue.

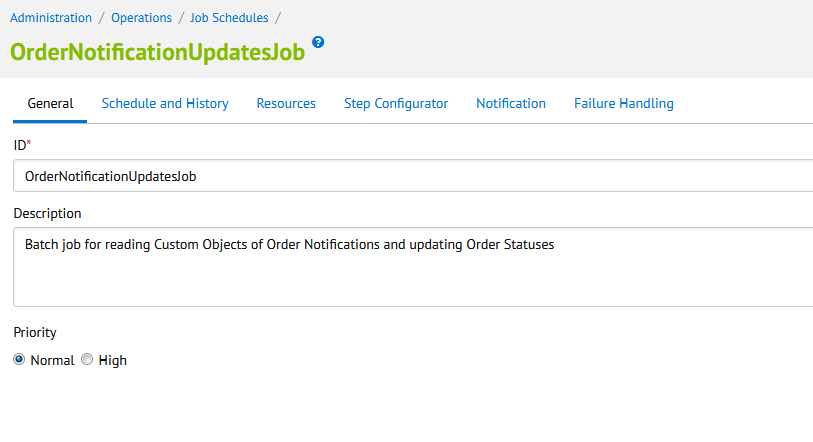
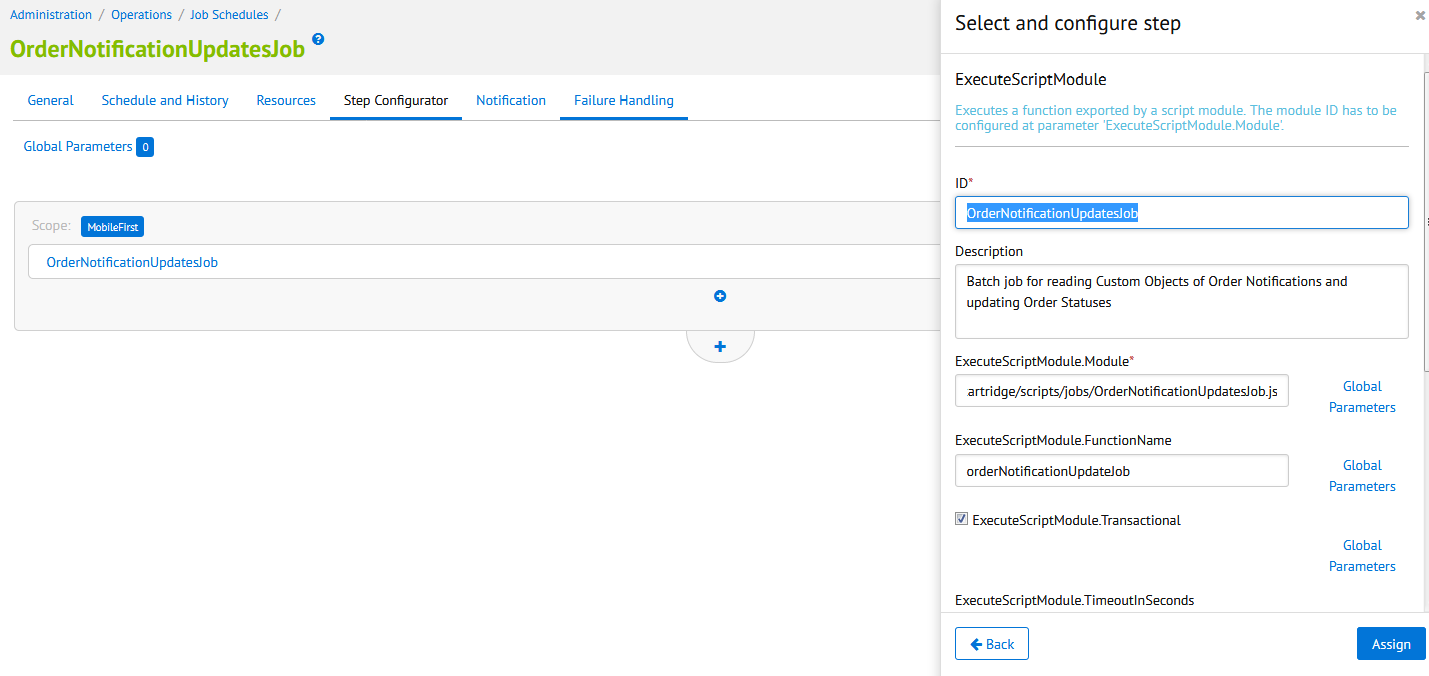
This URL (**https://mechantsandbox/Worldpay-Notify**) needs to be updated in Worldpay Admin console by the merchant to receive the Order Notifications. The order notifications are captured from Worldpay Payment Service when a payment reaches one of the following statuses:

* AUTHORISED
* CANCELLED
* CAPTURED
* EXPIRED
* SENT\_FOR\_REFUND
* REFUSED
* SETTLED
* INFORMATION\_REQUESTED
* CHARGED\_BACK
* POST AUTH CANCELLED

#### Order Notification job

The OrderNotificationUpdatesJob is a batch job which can be run manually or scheduled at a specific interval of time -i.e., 2 mins, 5 mins. Each time the job runs, it will pick all the custom objects created because of the prior received Order Notifications, sorted by the creation time. The job read the custom objects one by one, will update the statuses in Business Manager (Order Status/Payment Status/Confirmation Status) depending on the notification status in xml string and update the token details in order/customer card and finally remove the custom object post processing from SFCC.

Refer below Step configuration section in job to execute script and their functions

|  |  |
| --- | --- |
| Fields | Fields data |
| ID | OrderNotificationUpdatesJob |
| Description | Batch job for reading Custom Objects of Order Notifications and updating Order Statuses |
| ExecuteScriptModule.Module | link\_worldpay\_core/cartridge/scripts/jobs/OrderNotificationUpdatesJob.js |
| ExecuteScriptModule.FunctionName | orderNotificationUpdateJob |
| ExecuteScriptModule.Transactional | Enabled |

#### Error Handling for Order Notification Job

Please refer to[Appendix C](#_APPENDIX_C:_Error) for error codes and error messages for Order Notifications

#### Transaction payment status handling for Order Notification Job

Please refer to [Appendix D](#_APPENDIX_D:_Order) for mapping of different status on receipt of different order notification from WP into BM.

#### Notification update Service

[The Cartridge provides feature to expose the history of the Payment Transaction change (Order Notifications) for a particular Order in SFCC business manager.](https:// /WorldPay-GetNotificationUpdates  is%20the%20pipeline%20exposed%20to%20get%20the%20History%20of%20Notifications%20received%20by%20particular%20Order. )A JSON object is returned by the notifications updates service. This service is provided in 2 flavors, depending upon the need to fetch the entire status history or just the latest status. If order no or status history is not found an error JSON is returned.

**[Worldpay-GetNotificationUpdates](https://merchantSandbox/WorldPay-GetNotificationUpdates  is the pipeline exposed to get the History of Notifications received by particular Order.)** [is the route exposed to get the History of Notifications received by particular Order.](https://merchantSandbox/WorldPay-GetNotificationUpdates  is the pipeline exposed to get the History of Notifications received by particular Order.)

This service takes 2 parameters as Http Parameters,

1. orderNo
2. allupdates

If ‘allupdates’ is true, all notifications received for the order will be returned as JSON response.

If ‘allupdates’ is false, only the latest notifications received for the order will be returned as JSON response.

All values for ‘allupdates’ i.e. true, TRUE, True will be considered as true, any value other than these will be considered as false.

Example for the JSON response when ‘allupdates’ = true is as below:

**RESPONSE:**

{"statusList":

[{"status":"AUTHORISED:Mon Nov 10 2014 08:11:24 GMT-0000 (GMT)"},

{"status":"CANCELLED:Mon Nov 11 2014 08:11:24 GMT-0000 (GMT)"}

]

}

Example for the JSON response when ‘allupdates’ = false is as below:

**RESPONSE:**

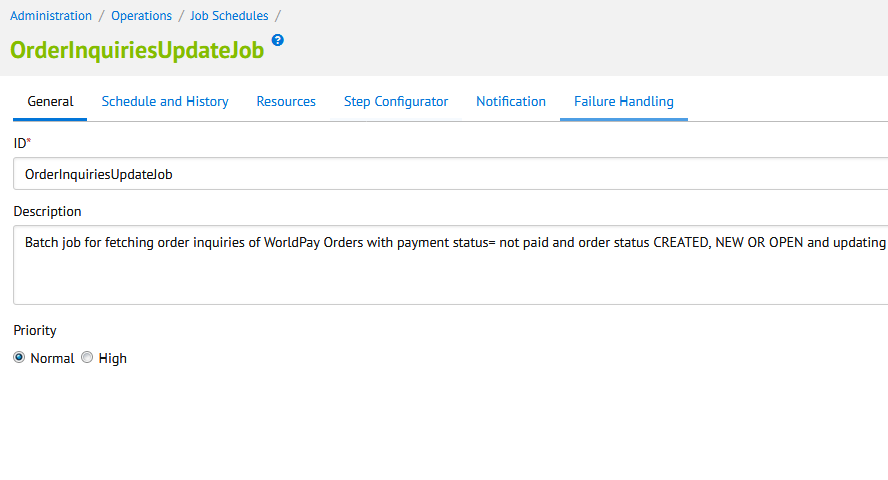
{"latestStatus":

[{"Status":"AUTHORISED:Mon Nov 10 2014 08:11:24 GMT-0000 (GMT)"}]

}

### Order Inquiry job

The OrderInquiryUpdatesJob is a batch job which can be run manually or scheduled at a specific interval of time -i.e., 2 mins, 5 mins. Each time the job runs, it will pick all the orders with status as CREATED/ NEW/OPEN and payment status as NOT PAID and order placed at least 15 mins back (configurable custom preference in business manager) by the creation time. The job iterates the orders and make order inquiry service call, will update the statuses in Business Manager (Order Status/Payment Status/Confirmation Status) depending on the response status in xml response of order enquiry and update the token details in order/customer card



Refer below Step configuration section in job to execute script and their functions

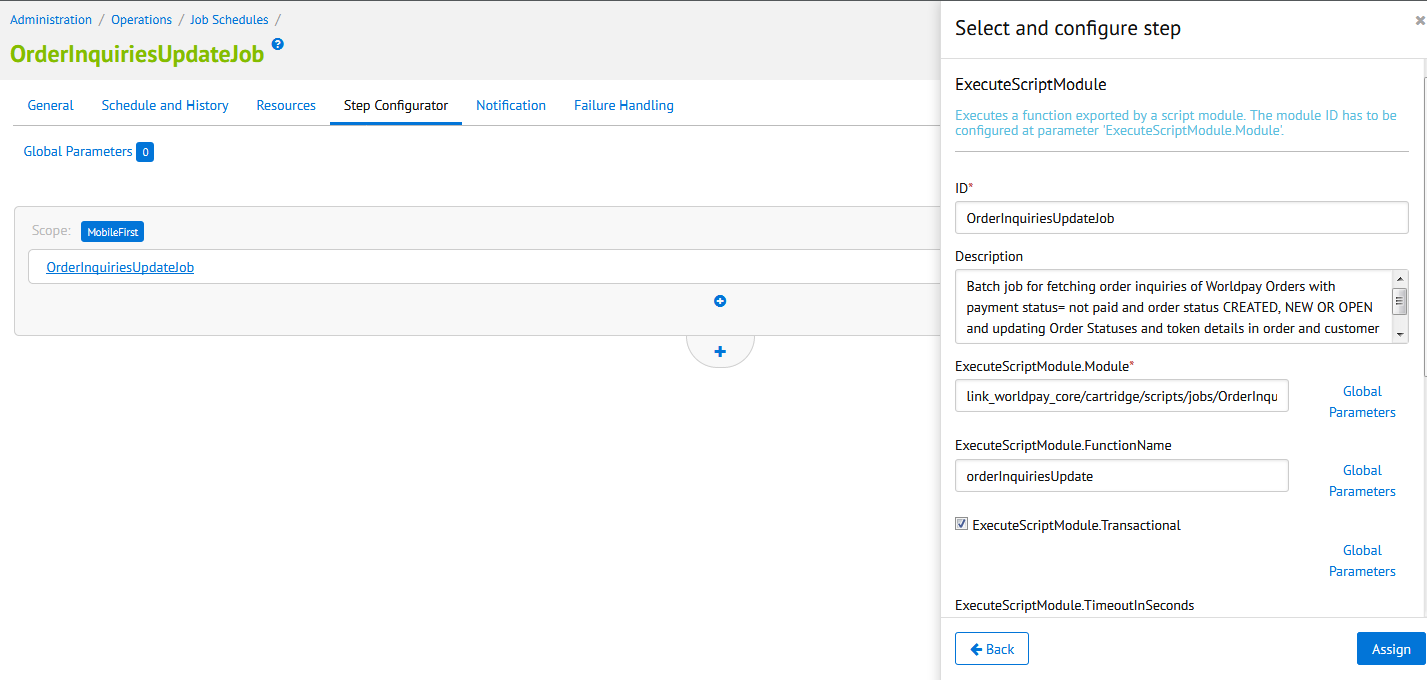


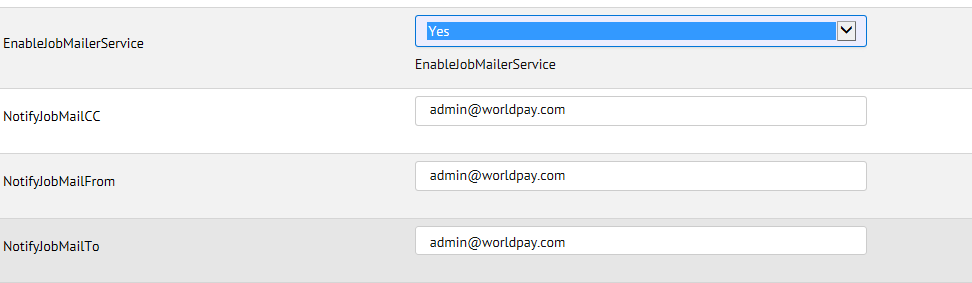
Table for Reference:

|  |  |
| --- | --- |
| Fields | Fields data |
| ID | OrderInquiriesUpdateJob |
| Description | Batch job for fetching order inquiries of Worldpay Orders with payment status= not paid and order status CREATED, NEW OR OPEN and updating Order Statuses and token details in order and customer saved payment instruments |
| ExecuteScriptModule.Module | link\_worldpay\_core/cartridge/scripts/jobs/OrderInquiriesUpdateJob.js |
| ExecuteScriptModule.FunctionName | orderInquiriesUpdate |
| ExecuteScriptModule.Transactional | Enabled |

### Job Failure Mailer Service

The cartridge also provides a feature to send Emails to the registered Emails in case of failure of any Order Notification update. On receipt of any Order Notification a custom object is created in SFCC with the Order Number and the Order Notification XML, if for any reason the XML captured is incorrect of some error occurs while the execution of the job an email is triggered.

To enable job mailer service to notify BM admin for any failure happens while processing the Order Notification Batch Job, the site preference EnableJobMailerService needs to be select **Yes** from the dropdown. Also, the below site preferences needs to be set



### APM Lookup Service

To enable APM look up service, the site preference **EnableAPMLookUpService** needs to be selected Yes from dropdown.

|  |  |
| --- | --- |
|  |  |
| **EnableAPMLookUpService** when enabled | 1. This service works for merchant code configured in site preference, which return APMs which are enabled for respective merchant code. 2. All active Payments methods which have different merchant code configured in payment method section will be displayed 3. All active payment methods other than Worldpay processor will be displayed |
| **EnableAPMLookUpService** when disabled | 1. Loookup service will not be executed 2. All active payment methods will be displayed |

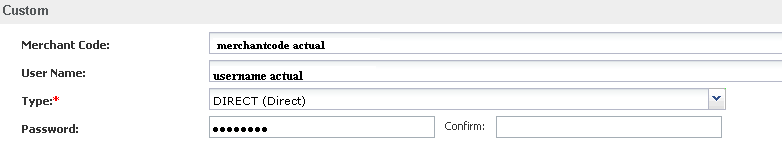
[Note: If merchant supports multiple merchant Ids through payment methods in that case APM Look service may not work so, it should not be enabled in site preference.]

### Multiple Merchant ID Support

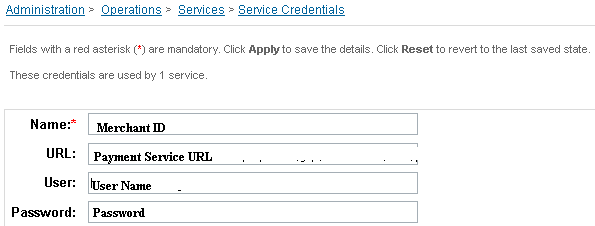
Worldpay cartridge also supports a mechanism to provide separate Merchant ID’s at APM level. This is primarily required for Merchants that need to configure different Merchant ID for financial purpose or if a specific APM is of a different version (to support both V1 and V3 APM’s on a single Merchant Site).

This is achieved, by setting a different MID and its details (Username & password) in the Payment Methods section in Business Manager. If merchant ID, username and password present at payment method level then would get preference compare to merchant ID in site preference, username and password from service credentials. If they are not configured at payment method level then merchant ID (in site preference), username and password provided in service credential will be used in request.

Payment Method Merchant ID Support Settings



Service Credential Settings when service merchant settings to be picked up

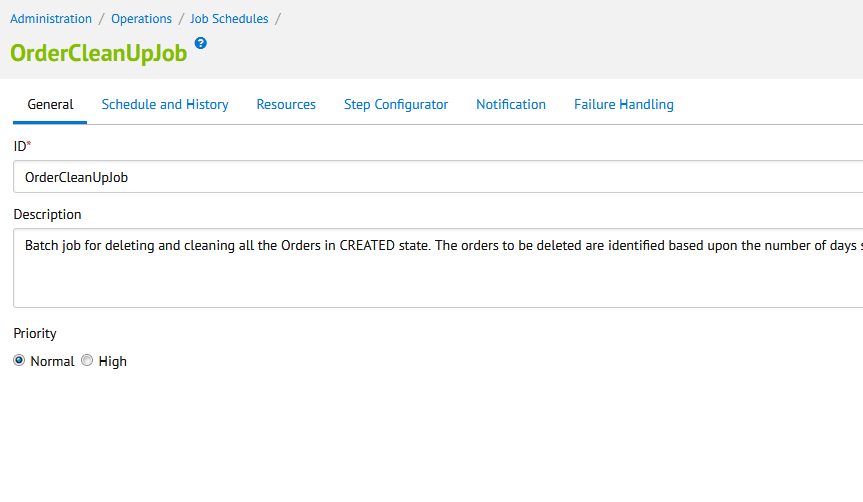


### Brazil Integration

The Worldpay cartridge provides support to Credit Cards and supported APM’s (BOLETO) in Brazil. The implementation ensures that all the additional information (Example: CPF/CNJP and Installment Number) that are required by the payment service provider in Brazil are also captured.

### Order CleanUp Batch Job

The Worldpay cartridge provides support to a batch job that will mark all the orders with status as **CREATED** and aged by a defined time interval. The time interval field is Business Manager configurable and also configurable at Payment Method level. A Batch Job configured in Business Manager shall run (as per the set schedule) and check the orders with order status **CREATED** which are old to a defined interval and will mark those orders in **FAILED** status in demandware system. This ensures that all the inventory transactions and coupon redemptions associated with the Order are rolled back. Job can be Enabled/Disabled or from Business Manager itself. Email will be sent to BM configured recipients once the job run with Job run status.



Refer below Step configuration section in order to execute script and their functions

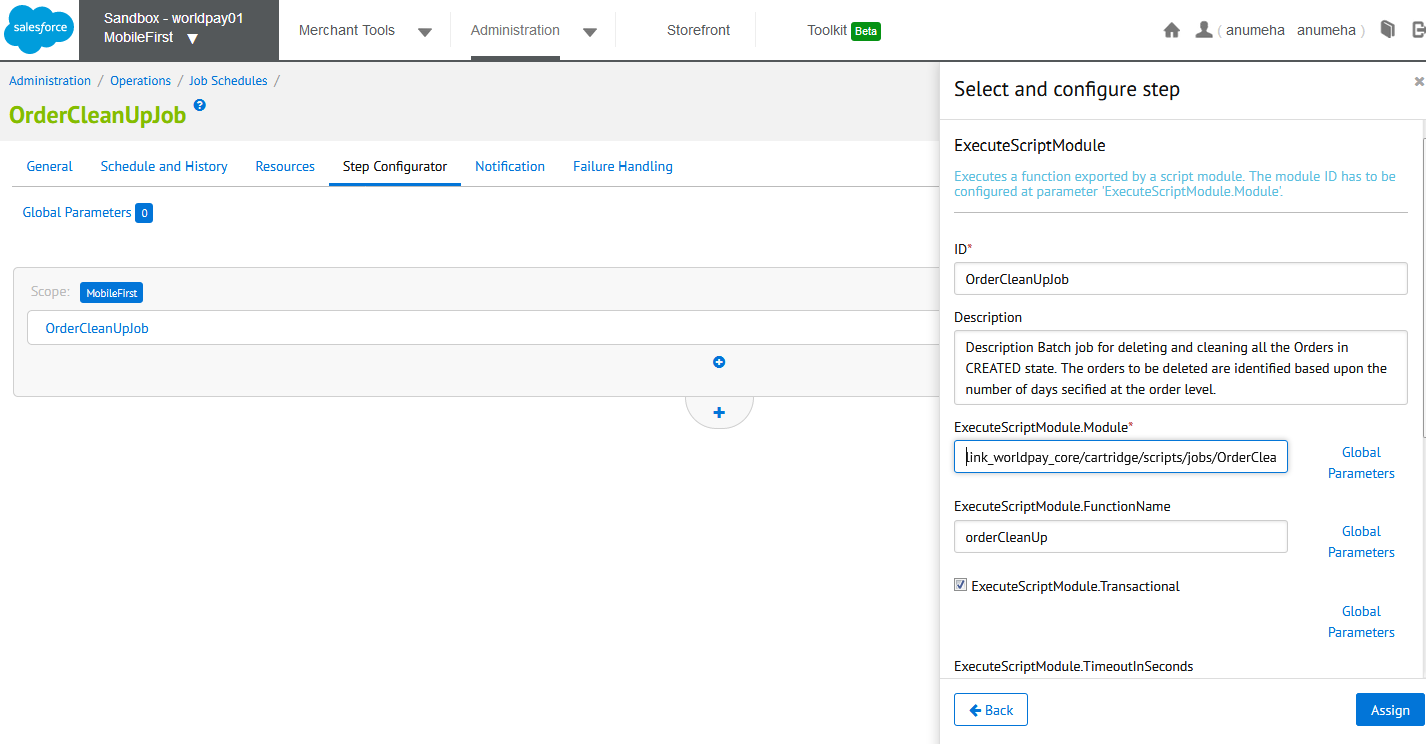
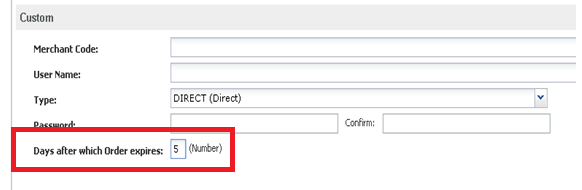


Table for Reference:

|  |  |
| --- | --- |
| Attributes Fields | Attributes |
| ID | OrderCleanUpJob |
| Description | Description Batch job for deleting and cleaning all the Orders in CREATED state. The orders to be deleted are identified based upon the number of days secified at the order level. |
| ExecuteScriptModule.Module | link\_worldpay\_core/cartridge/scripts/jobs/OrderCleanUpJob.js |
| ExecuteScriptModule.FunctionName | orderCleanUp |
| ExecuteScriptModule.Transactional | Enabled |

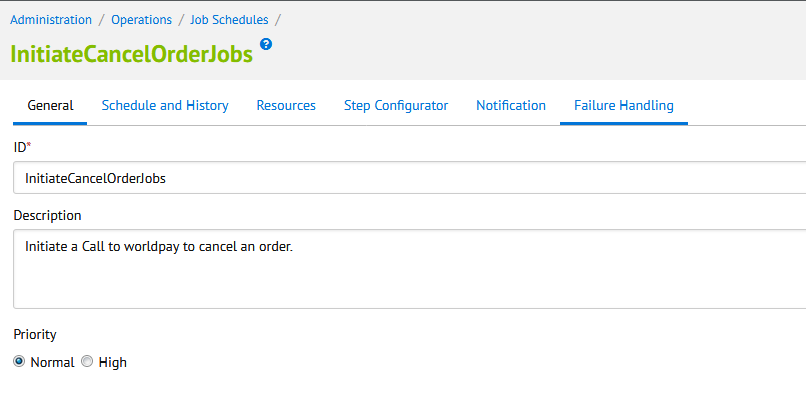
Note: Please ensure before running batch job every payment method in site has attribute “Days after which order expires” value configured.

Business Manager Configuration field:



### Initiate Cancel Order Job

The Worldpay cartridge provides support to a batch job that cancels or refunds FAILED orders with the attribute Worldpay MAC Missing. The time interval for selecting orders defined in the custom preference Worldpay Order Inquiry Lag Time. A Batch Job configured in Business Manager shall run (as per the set schedule) and check orders with order status FAILED, with Worldpay MAC Missing which are old to a defined interval of time and will send a request to Worldpay System to either cancel or refund.  Job can be Enable/Disable from Business Manager itself. Email will be sent to BM configured recipients after job run with Job run status.



Refer below Step configuration section in order to execute script and their functions

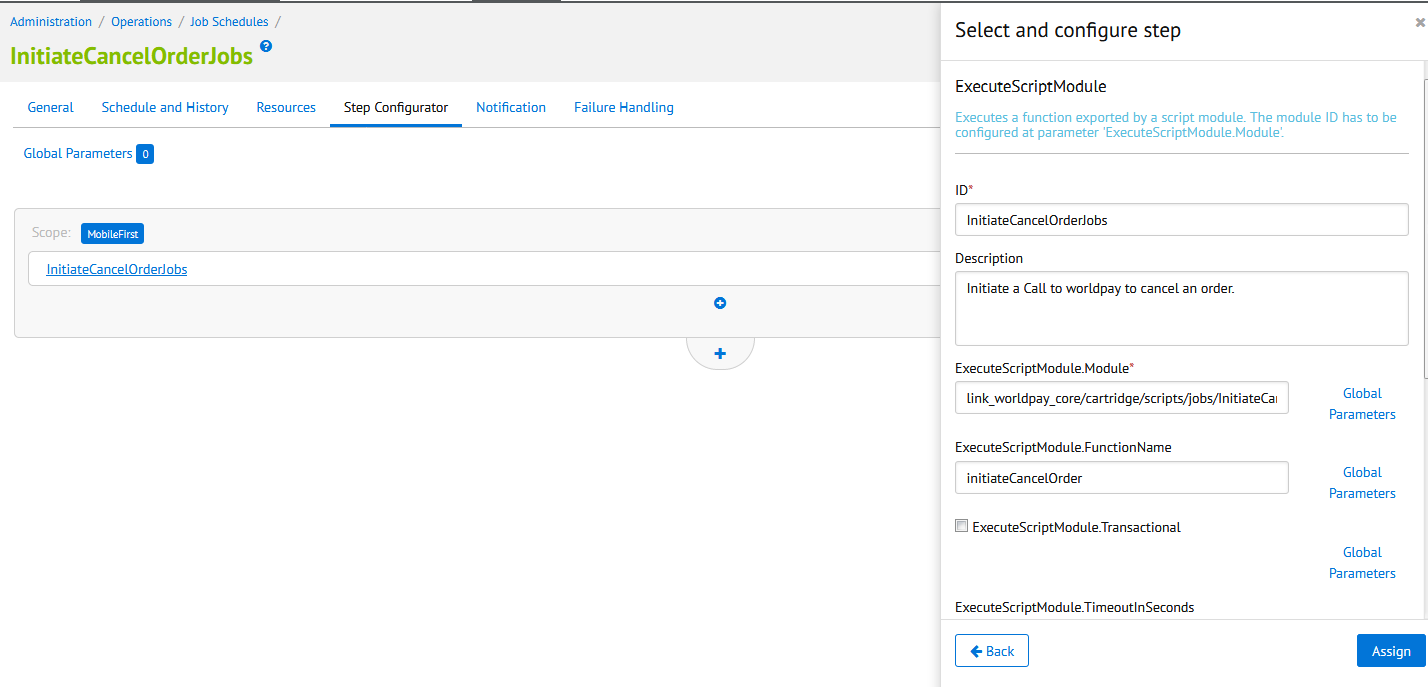


Table for Reference:

|  |  |
| --- | --- |
| Fields | Fields data |
| ID | InitiateCancelOrderJobs |
| Description | Initiate a Call to worldpay to cancel an order. |
| ExecuteScriptModule.Module | link\_worldpay\_core/cartridge/scripts/jobs/InitiateCancelOrderJob.js |
| ExecuteScriptModule.FunctionName | initiateCancelOrder |
| ExecuteScriptModule.Transactional | Enabled |

### Capture Service

Capturing a payment can be done with tracking ID and without tracking ID for partial payment and full payment. A tracking ID is the shipment ID identifier that merchant should use to track the delivery of your order.

#### Capture Service with Tracking ID Request

<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE paymentService PUBLIC "-//WorldPay//DTD WorldPay PaymentService v1//EN" "http://dtd.worldpay.com/paymentService\_v1.dtd">

<paymentService merchantCode="YOUR\_MERCHANT\_CODE" version="1.4">

  <modify>

    <orderModification orderCode="jsxml0000000000">

      <capture>

        <amount value="10000" currencyCode="GBP" exponent="2" debitCreditIndicator="credit"/>

        <shipping>

          <shippingInfo trackingId="AAA0000000"/>

          <shippingInfo trackingId="BBB0000000"/>

          <shippingInfo trackingId="XXX0000000"/>

        </shipping>

      </capture>

    </orderModification>

  </modify>

</paymentService>

#### Capture Service with Tracking ID Response

<?xml version="1.0" encoding="UTF-8"?> <!DOCTYPE paymentService PUBLIC "-//WorldPay//DTD WorldPay PaymentService v1//EN" "http://dtd.worldpay.com/paymentService\_v1.dtd"> <paymentService version="1.4" merchantCode="SAPIENTNITROECOM"><reply><ok><captureReceived orderCode="T00028020000"><amount value="10000" currencyCode="GBP" exponent="2" debitCreditIndicator="credit"/></captureReceived></ok></reply></paymentService>

#### Capture Service without Tracking ID Request

<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE paymentService PUBLIC "-//WorldPay//DTD WorldPay PaymentService v1//EN" "http://dtd.worldpay.com/paymentService\_v1.dtd">

<paymentService merchantCode="YOUR\_MERCHANT\_CODE" version="1.4">

  <modify>

    <orderModification orderCode="jsxml0000000000">

      <capture>

        <amount value="10000" currencyCode="GBP" exponent="2" debitCreditIndicator="credit"/>

        <shipping>

          <shippingInfo trackingId=""/>

        </shipping>

      </capture>

    </orderModification>

  </modify>

</paymentService>

#### Capture Service without Tracking ID Response

<?xml version="1.0" encoding="UTF-8"?> <!DOCTYPE paymentService PUBLIC "-//WorldPay//DTD WorldPay PaymentService v1//EN" "http://dtd.worldpay.com/paymentService\_v1.dtd"> <paymentService version="1.4" merchantCode="SAPIENTNITROECOM"><reply><ok><captureReceived orderCode="T00028020000"><amount value="10000" currencyCode="GBP" exponent="2" debitCreditIndicator="credit"/></captureReceived></ok></reply></paymentService>

## LIMITATIONS, CONSTRAINTS

This section details the limitations, constraints and the best practice for merchants planning to integrate the Worldpay cartridge with SFCC Storefront.

1. At the moment, it is not possible to detect against session timeouts while the customer is entering payment details in the Worldpay forms. If there is a delay longer than the session timeout, basket contents are lost and order is placed and kept in Created state.
2. As the placement of the street number in addresses varies between different countries, the integration does not provide a parsing mechanism for the street number part as this would be very application specific. In order to avoid future errors, the parsing of street numbers must be taken care of during the integration installation. If this step is ignored, the customer will have to supply the street number at the Worldpay payment page.
3. The cartridge supports both LookUp service and Multiple MID support service. However, only one of them should be enabled at a time.
4. At the moment, the cartridge supports only one generic message that is displayed on the Billing Page in case of a Failure response from APM’s or Credit Card redirect method. This is majorly because we do not have any identifier or error code that is returned.
5. The web service Timeout that has been set in service profile is common across all the web service calls that we make to Worldpay be it Look up call, APM Order XML call, Authorization call etc.
6. ThankYou page is displayed for both Successful and Pending Open Scenario; Merchant shall take care to display a message to the user on the Thank You page with a suitable disclaimer that demarcates both the scenarios.
7. SFCC Site Genesis has a default behavior to show an error message on Place Order page in case of a previously failed transaction. This can be removed by commenting/removing the code from summary.isml

<div class="error-form">${Resource.msg(pdict.PlaceOrderError.code,'checkout',null)}</div>

1. The Maximum and Minimum supported amount per APM should be configured at the SFCC level in the Payment Method configuration section
2. Merchants shall ensure that Overnight Shipping and One Day shipping methods should not be displayed for Delayed APMs or for the ones that have KIOS payments
3. Merchant Shall take care of implementing the validations on the additional field that will be supported at the Merchant’s site (Example: CPF filed in-case of Boleto, Swift Code/Bank Code in-case of GiroPay), while implementing Credit Card Direct the merchant should ensure that the Credit Card Form is reset in case of any error.
4. Cartridge shall be responsible to update the order status and export status only on receipt of following notifications/enquiry AUTHORISED, CAPTURED. Rest all other notification/enquiry status are read by the Job and Transaction history flag is updated with the notification/enquiry status value no further action is taken by the cartridge
5. Merchants shall take care of implementing credit card fields refresh functionality in case of user changes the card Type after filling the card information for previous selection to have better shopper experience
6. At the moment, it is not possible to run client side encryption for saved credit card where card number displayed as masked even client side encryption feature is turned on in merchant Worldpay account and enabled corresponding setting in Merchant site custom preference.
7. Merchant must ensure customer sequence number not exceeding 64 characters length and may contain alphanumeric or underscore.
8. Tokenization would be applicable for registered user only provided tokenization feature is turned on in merchant Worldpay account and enabled corresponding setting in Merchant site custom preference.
9. At the moment, it is not possible to validate mac secret identifier into response [failure /pending] URL's for all Redirect APMs and Redirect Credit Card orders.
10. Notification feature is restricted to only allowed IP address provided by Worldpay, and corresponding setting to be configured in Merchant site custom preference.
11. Merchants shall take care of implementing Order Settlement and refunds.
12. Merchant shall take care of enable/disable site preference for user to provide CVV on billing page as security measure in tokenization service for saved cards. Also need to enable CVV check service from Worldpay to work properly.
13. Merchant must ensure SFCC Service Framework configurations are setup and configured as per the contract with Worldpay on service availability.
14. Merchant shall take care for CVV field validation as enable/disable, CVV in BM configurations will only hide/show CVV field on billing page and no CVV validation is performed on Worldpay.
15. Jobs Templates can be modified as per merchant needs. These Jobs templates are prepared for reference.
16. The images are available in static folder for the cards. They can further be modified if required.
17. In case of iframe and lightbox for Airplus, Denkort and Laser card, Payment Method is displayed with message “Please contact worldpay”
18. Credit Card Payment method must be enabled because of some code restrictions based on the MFRA Code.
19. For Credit Redirect, credit card must be available in the system.
20. Gifts certificates are out of scopes.
21. There are two LINT issues in the code related to @ character.

## COMPATIBILITY

This cartridge is integrated and tested with MobileFirst MFRA code base not released globally yet of SFCC.

# IMPLEMENTATION GUIDE



## SETUP

This section describes the controller structure and name of files in Worldpay cartridges.

**CARTRIDGE STRUCTURE**

Controllers:

Controllers are present in the “link\_worldpay” cartridge.

|  |  |
| --- | --- |
| **CONTROLLER NAME** | **DESCRIPTION** |
| Worldpay | Main MFRA controller that acts as the external interface. |
| WorldpayHelper | Helper controller that supports and performs additional tasks required to complete |
| CheckoutServices | MFRA controller that supports and executes the Checkout process. |
| COPlaceOrder | MFRA controller that supports and executes the Checkout process. |

* package.json – initialize hook or service

Scripts:

**link\_worldpay\_core**

* common/CreateRequestHelper.js
* common/PaymentInstrumentUtils.js
* common/Utils.js
* common/WorldpayConstants.js
* common/worldPayHelper.js
* init/httpServiceInit.js
* jobs/InitiateCancelOrderJob.js
* jobs/OrderCleanUpJob.js
* jobs/OrderInquiriesUpdateJob.js
* jobs/OrderNotificationUpdatesJob.js
* jobs/WorldpayJobs.js
* lib/LibCreateRequest.js
* object/ResponseData.js
* object/WorldpayPreferences.js
* order/UpdateOrderStatus.js
* order/UpdateTransactionStatus.js
* pipelets/CheckWorldpayOrder.js
* pipelets/GenerateErrorMessageForJob.js
* pipelets/GetCreatedOrderDetails.js
* pipelets/GetPaymentCardToken.js
* pipelets/InitiateCancelOrder.js
* pipelets/ReadNotifyCustomObject.js
* pipelets/RemoveNotifyCustomObject.js
* pipelets/SaveCustomerCreditCard.js
* pipelets/SendWorldpayOrderInquiriesRequest.js
* pipelets/WriteToNotifyLog.js
* service/ServiceFacade.js
* hooks.json

**link\_worldpay**

* hooks/payment/processor/worldpay.js
* hooks.json

Templates:

**link\_worldpay\_core**

* default/allstatusjson.isml
* default/emailtemplateforjob.isml
* default/errorjson.isml
* default/http\_500.isml
* default/lateststatusjson.isml
* default/notifyResponsejson.isml
* default/worldpayredirect.isml

**link\_worldpay**

* default/account/payment/editAddPayment.isml
* default/checkout/billing/paymentOptions/billingUserContent.isml
* default/checkout/billing/paymentOptions/creditCardContent.isml
* default/checkout/billing/paymentOptions/creditCardTab.isml
* default/checkout/billing/paymentOptions/elvContent.isml
* default/checkout/billing/paymentOptions/giropayContent.isml
* default/checkout/billing/paymentOptions/idealContent.isml
* default/checkout/billing/paymentOptions/paymentOptionsContent.isml
* default/checkout/billing/paymentOptions/paymentOptionsSummary.isml
* default/checkout/billing/paymentOptions/paymentOptionsTabs.isml
* default/checkout/billing/paymentOptions/redirectContent.isml
* default/checkout/billing/paymentOptions/redirectSummary.isml
* default/checkout/billing/paymentOptions/redirectTab.isml
* default/checkout/billing/creditCardForm.isml
* default/checkout/billing/paymentOptions.isml
* default/checkout/confirmation/confirmationEmail.isml
* default/checkout/checkout.isml
* default/checkout/worldpaysummary.isml

Properties:

**link\_worldpay\_core**

* resources/creditCard.properties
* resources/forms.properties
* resources/worldpayerror.properties
* resources/worldpay.properties

## INSTALLATION

### Adding the Cartridges in Salesforce Commerce Cloud Studio

Inside mfra\_worldpay folder just before cartridges directory, open package.json and ensure the base path mentioned is correctly resolved (please provide the correct path to app\_storefront\_base), open the command prompt and run ‘npm install’ (assuming node.js (8.9.4 and above) is installed). Run the “npm run compile:js” followed by “npm run compile:scss”. Create the dw.json file at the location and Upload the code to the environment using the command ‘npm run uploadCartridge’. Please go to the link\_worldpay:link\_worldpay\_core cartridge and upload the code by configuring the various parameters in the sample dw.json.

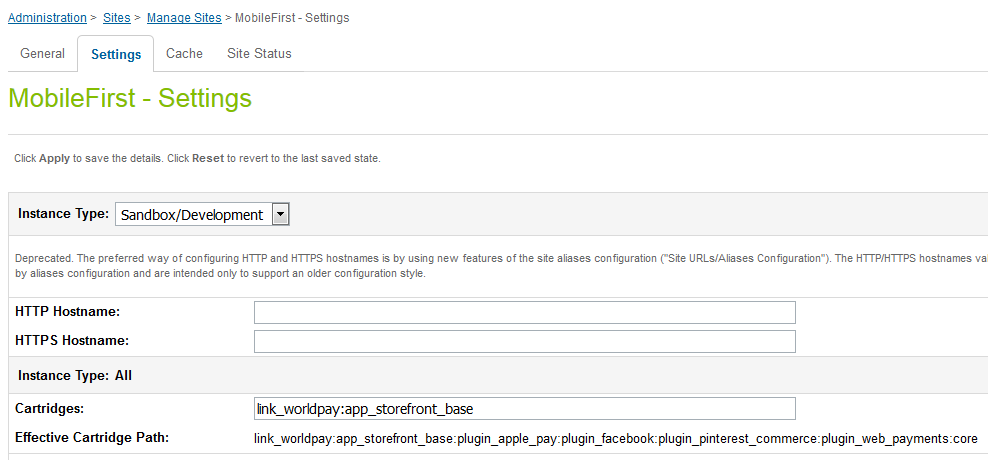
### IP Address

Please provide outgoing IP of your client in order to white list your IP otherwise some of the attempts to make payment will be marked as fraud.

### Activating the Cartridges in Business Manager

Before the Worldpay functionality can become available to Reference Architecture, the cartridges have to be added to the cartridge path of the Site in question. In order to do this, follow the following instructions:

1. Log into Business Manager
2. Navigate to *Administration 🡪 Sites 🡪 Manage Sites*.
3. Click on the site name and on the next page go to the *Settings* tab.
4. In the textbox *Cartridges* add “**link\_worldpay:link\_worldpay\_core:app\_storefront\_base**”.
5. Click *Apply*.
6. To activate the cartridge for the Sandbox instances repeat steps **4** and **5** after selecting the appropriate instance from the *Instance Type* dropdown menu.
7. Repeat steps **3** to **6** for each site that is to use Worldpay.
8. To run the Job in worldpay cartridge,Navigate to *Administration* 🡪 *Sites* 🡪 *Manage Sites*.
9. Go to “Manage the **Business Manager** site”
10. In the textbox *Cartridges* add to the start: “**link\_worldpay:link\_worldpay\_core:app\_storefront\_base**”.



## CONFIGURATION

This chapter will guide the user to configure the cartridge in Business Manager.

### Importing Meta Data

All import files can be found in the import folder within cartridge installation pack. To import all necessary Worldpay settings, log in to the Business Manager and navigate to *Administration* 🡪 *Site Development* 🡪 *Import & Export*. Now upload the worldpay\_metadata.xml file using the upload button and, finally go back and use the import button to import the file. After a successful import, you will be able to configure the custom site preferences according to your Worldpay account data. Also, verify the **Order level attributes** in BM (Site -> Ordering -> Order) open any order and navigate to the Attributes tab. It should be as below: -

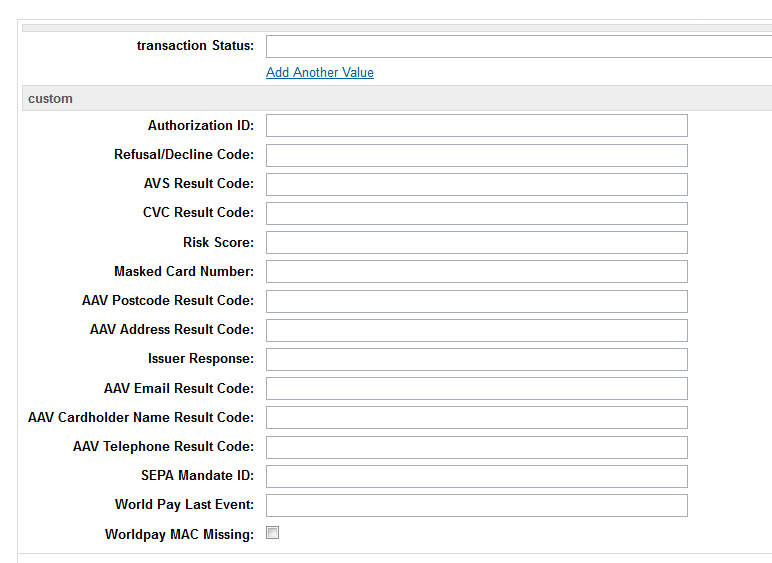
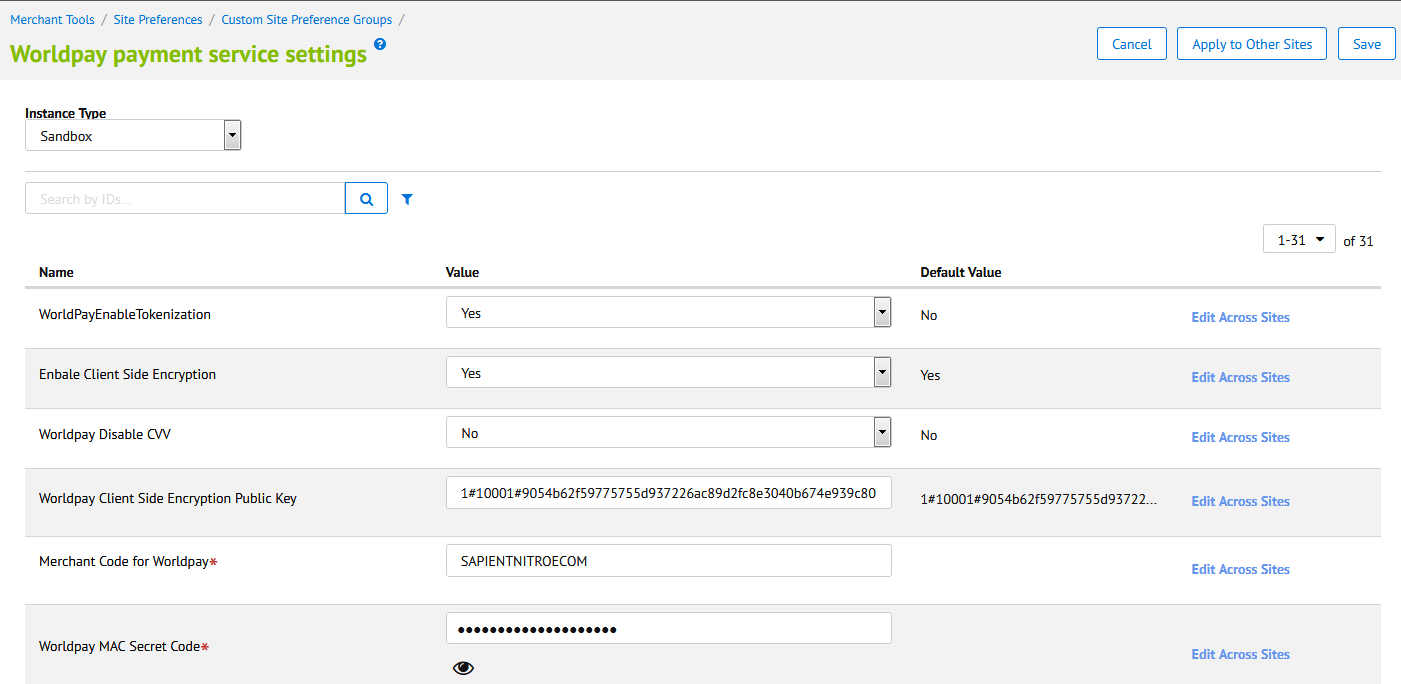
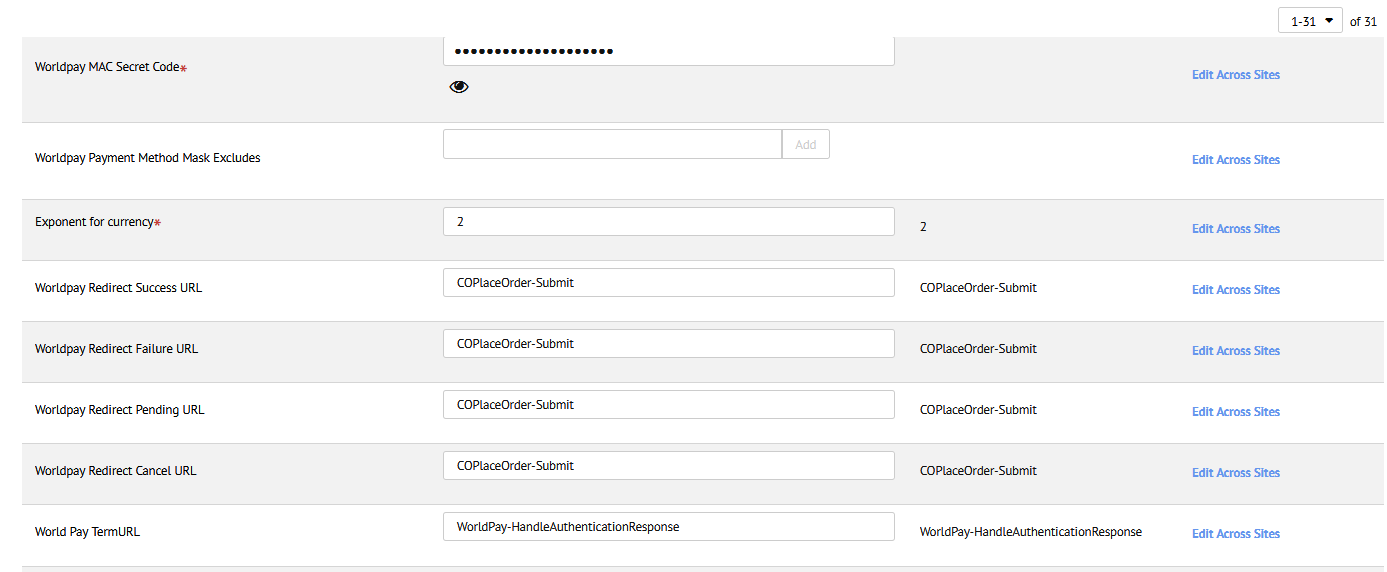
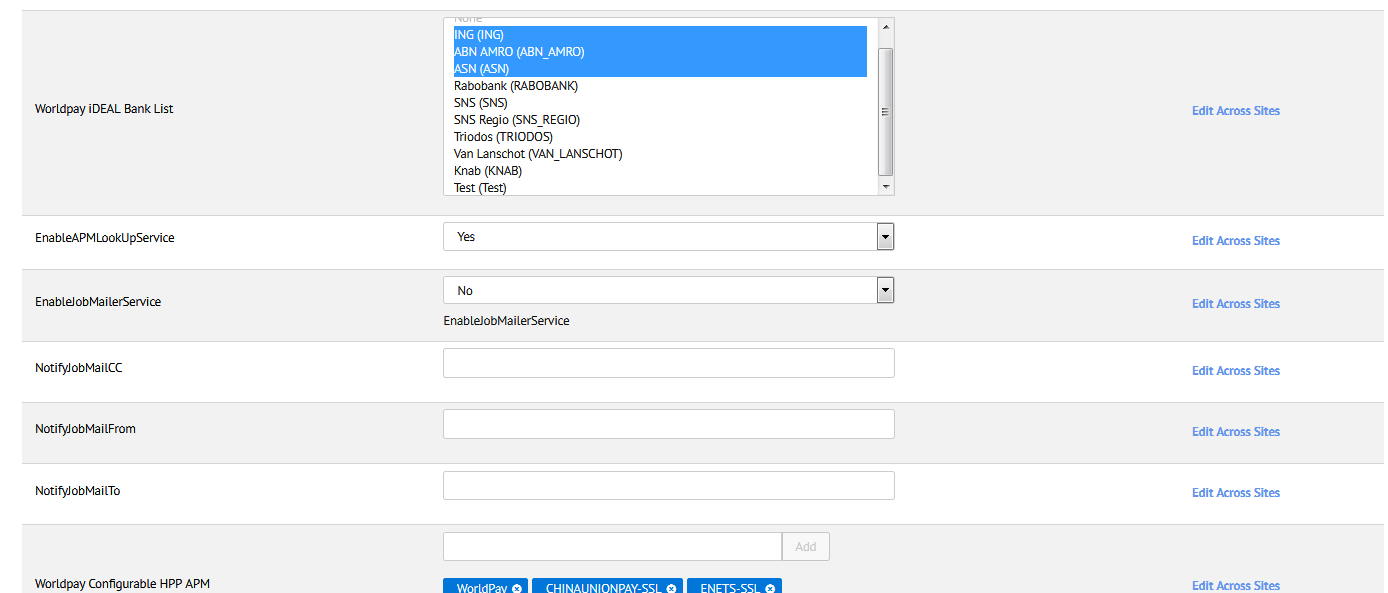


Fig: 2.1

Custom Preferences Screenshot







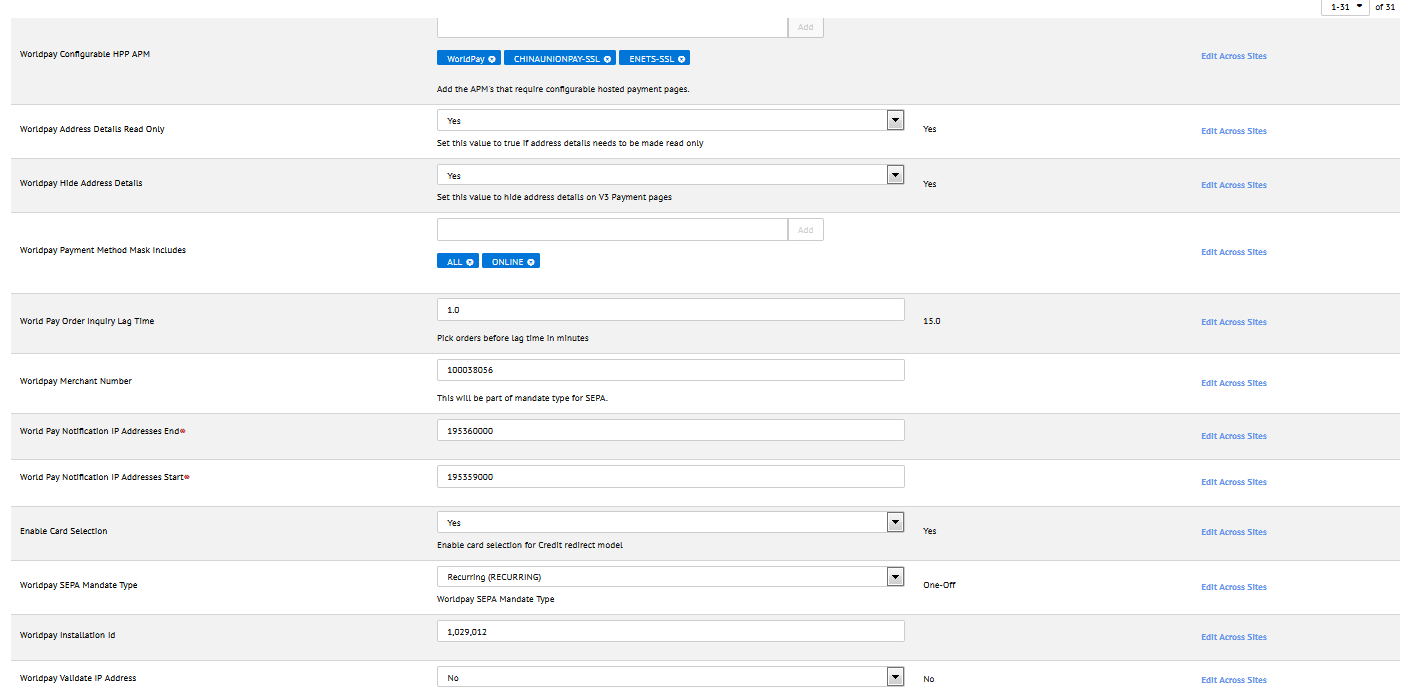


Fig: 2.2

### Setting Worldpay Custom Site Preferences

In Business Manager, navigate to the *MoblieFirst Site🡪* Merchant Tools🡪 *Site Preferences* 🡪 *Custom Preferences*. A custom site preference group with the ID Worldpay is available. Please select it and edit the attributes according to your Worldpay account data and the data shown in figure 2.2. In addition to the ones supplied by Worldpay fill out the following properties with the values listed against them:

* Worldpay Redirect SuccessURL: COPlaceOrder-Submit
* Worldpay Redirect PendingURL: COPlaceOrder-Submit
* Worldpay Redirect FailureURL: COPlaceOrder-Submit
* Worldpay Redirect CancelURL: COPlaceOrder-Submit
* Worldpay TermURL: Worldpay-HandleAuthenticationResponse

\*\* The fields highlighted in blue in the below table are mandatory fields

|  |  |  |
| --- | --- | --- |
| **Site Preference** | **Description** | **Default Values** |
| **Merchant Code for Worldpay** | This field represents the specific Merchant code value. | Merchant Specific - Provided by Worldpay |
| **Worldpay MAC Secret Code** | The field represents the MAC secret code that configurable in Worldpay Merchant Interface. | Merchant Specific |
| **Worldpay Payment Method Mask Includes** | This field represents all the Cards and APM’s supported and will be shown on Hosted Payment Page. | Merchant Specific - Add the Cards /APMs supported on Hosted Payment Pages.  (Refer to Appendix A for values) |
| **Worldpay Payment Method Mask Excludes** | This field represents all the Cards and APM’s that are needs to be excluded and will not be shown on Hosted Payment Page. | Merchant Specific |
| **Exponent for currency** | This field represents the decimal places supported in the Order Amount. | Merchant Specific  (Default Value : 2) |
| **Worldpay Redirect Success URL** | This field represents the Success URL, that the user will be redirected to incase of a successful transaction in APM’s or Credit Card Redirect. | Default Value : COPlaceOrder-Submit |
| **Worldpay Redirect Failure URL** | This field represents the Failure URL, that the user will be redirected to incase of a Failure transaction in APM’s or Credit Card Redirect. | Default Value : COPlaceOrder-Submit |
| **Worldpay Redirect Pending URL** | This field represents the Pending URL, that the user will be redirected to incase of a Pending or Pending-Open transaction in APM’s or Credit Card Redirect. | Default Value : COPlaceOrder-Submit |
| **Worldpay Redirect Cancel URL** | This field represents the Cancel URL, that the user will be redirected to incase of a cancelled transaction in APM’s or Credit Card Redirect. | Default Value : COPlaceOrder-Submit |
| **Worldpay TermURL** | This field represents the Term URL on which the user will be redirected in case of 3D secure. | Worldpay-HandleAuthenticationResponse |
| **Worldpay iDEAL Bank List** | This field represents the Bank’s supported for iDEAL APM. The merchant can select the Banks that needs to be displayed on the Merchant Site. | Merchant Specific - Select the ones to be supported. |
| **Worldpay Installation Id** | This field represents the installation ID provided by Worldpay, it will determine the look and feel of the Hosted Payment Pages. | Merchant Specific - Provided by Worldpay |
| **Worldpay Address Details Read Only** | This field is set to true if address details needs to be made read only | Merchant Specific  (Default Value : true) |
| **Worldpay Hide Address Details** | This field is set to true to hide address details on V3 Payment pages | Merchant Specific  (Default Value : true) |
| **EnableAPMLookUpService** | This field can be used to Turn On/Off the Look Up service, based upon the Merchant requirement. | Merchant Specific |
| **EnableJobMailerService** | This field can be used to Turn On/Off the Email Service. The Emails are triggered in-case the cartridge logic fails to handle the Order Notification/Enquiry and update its status in Order transaction field. | Merchant Specific |
| **NotifyJobMailCC** | This field represents the Email address of the people to kept in CC. | Merchant Specific: Multiple Emails can be added separated by semicolon. |
| **NotifyJobMailFrom** | This field represents the Email address of the people to be in From. | Merchant Specific: Multiple Emails can be added separated by semicolon. |
| **NotifyJobMailTo** | This field represents the Email address of the people to be in To. | Merchant Specific: Multiple Emails can be added separated by semicolon. |
| **WorldpayEnable**  **Tokenization** | This field is set to true if merchant wants to enable tokenization | Merchant Specific(Default value : false) |
| **Worldpay Disable CVV** | This field is set to true if merchant want to disable CVV check for Saved Card with token and new card.  We are suppressing the cvv field in the request of the order placed with saved card having token and new card. | Merchant Specific(Default value : false) |
| **Enable Client Side Encryption** | This field is set to true if merchant wants to enable Client Side Encryption | Merchant Specific(Default value : false) |
| **Worldpay Client Side Encryption Public Key** | This field represents the public key provided by Worldpay, used to calculate mac value of an order | Merchant Specific |
| **Worldpay Order Inquiry Lag Time** | This field represents the maximum time(in minutes) for which orders to be look-up in Worldpay for Order Enquiry or order cancel/refund Job | Merchant Specific(Default value : 15) |
| **Worldpay Configurable HPP APM** | This field represents the APM’s for which the site supports configurable Hosted Payment Pages (Look & Feel configuration). | Merchant Specific - Add the APM's that support configurable Hosted Payment Pages.  Ensure that the **Worldpay** **InstallationID** set if some value is added to this Preference. |
| **Enable Card Selection** | This field represents the Merchants choice to enable /disable Card selection on the Site. | Merchant Specific: Enable /Disable Card selection on the Site. |
| **Worldpay Merchant Number** | This field represents a Merchant specific number provided by Worldpay; this is sent as a part of request for SEPA. | Merchant Specific - Provided by Worldpay |
| **Worldpay SEPA Mandate Type** | This field represents the Mandate type to be sent in the SEPA request. (Default value ONE-OFF) | Merchant Specific – This represents the mandate type to be set for SEPA request. The merchant can choose to set it either as ONE-OFF or RECURRING. |
| **Worldpay Notification IP Addresses End** | This field represents the Worldpay IP Address range which send notification, last IP Address in range | Merchant Specific - Provided by Worldpay |
| **Worldpay Notification IP Addresses Start** | This field represents the Worldpay IP Address range which send notification, first IP Address in range | Merchant Specific - Provided by Worldpay |
| **Worldpay validate IP Address** | This field is set to true if merchant want to validate IP address | Merchant Specific(Default value : false) |
| **CaptureServiceTrackingID** | This field is set to true if merchant want to Validates the order id and token upon match it will proceed for capture service initiation with tracking ID | Merchant Specific(Default value : false) |
| **Enable Stored Credentials** | This field is set to true to comply with scheme mandates for storing (and subsequently using) card payment credentials for cardholder and merchant-initiated transactions. | Default value : false |

### Create Worldpay Payment Processor

Before you can import any actual payment method for Worldpay, you need to create the payment processor to handle this payment method. In order to do this, follow the instructions below:

1. In Business Manager navigate to *MobileFirst Site🡪Merchant Tools* *🡪 Ordering 🡪 Payment Processors*.
2. Click on the new button to create a new Payment Processor.
3. Enter ID: **Worldpay** and then click the apply button.

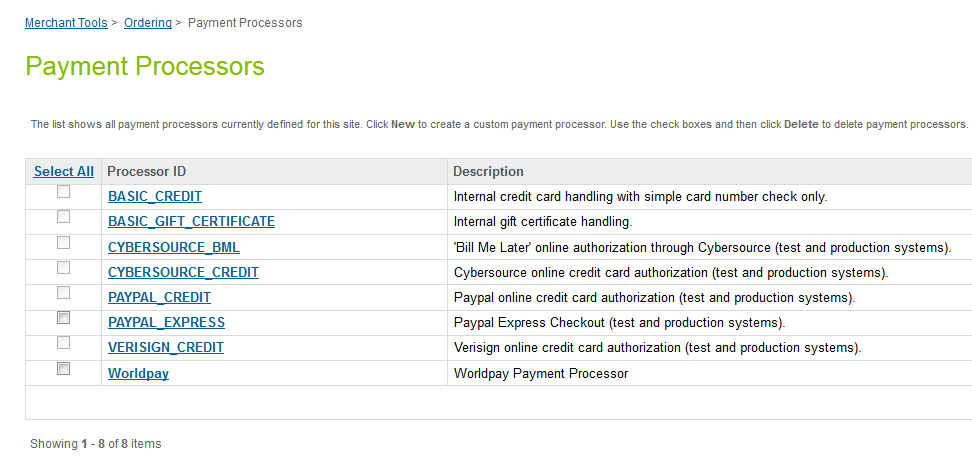


Fig: 2.3

### Importing Worldpay Payment Methods

In the integration package, a payment method definition is provided in the worldpay\_paymentmethods .xml file. All import files can be found in the import folder within the cartridge installation pack. To import this file, follow the instructions below:

1. In Business Manager navigate to *MobileFirst Site🡪* Merchant Tools*🡪 Ordering 🡪 Import & Export*.
2. Click on the upload button and in the next screen select the file by clicking the browse button. Notice that the file is in the import subdirectory. After you have selected the file click on the upload button.
3. After the file has been uploaded click on the back button. In the Payment Methods area, click on the Import button.
4. Select the file you just uploaded and click on the Next button.
5. After the file is successfully validated click on the Next button.
6. Make sure REPLACE is selected and click on the Import button.
7. Navigate to MobileFirst Site🡪 Merchant Tools 🡪 Ordering🡪 Payment Methods.
8. Enable the Payment Method under **Ordering -> Payment Methods section** that needs to be activated. Repeat these steps for all the Payment Methods that are required to be supported on the site. (Refer to Fig: 2.4)

Set below custom attributes of Payment method as provided by merchant: (Refer to Fig: 2.5)

|  |  |
| --- | --- |
| **Custom Attribute** | **Description** |
| Merchant Code | This field represents the hex code provided by merchant |
| User Name | This field represents the username provided by merchant |
| Type | Select the Payment Method Type to DIRECT/REDIRECT as per the one that needs to be supported on the merchant site. (Refer to Fig: 2.5) |
| Password | This field represents the password provided by merchant |
| Days after which Order expires | Configure the number of days after which the Order should be cleaned up |
| SEPA mandate number | This will be part of mandate type for SEPA. |
| HPP CustomOptions JSON | This would only be populated in case of payment method as “Credit Card Redirect” where merchant defines JSON which determines the type as iframe/lightbox in which Hosted Payment Page appears on Merchant site.  Default Value is blank represent Worldpay redirection page appears |
| Worldpay Statement Narrative |  |

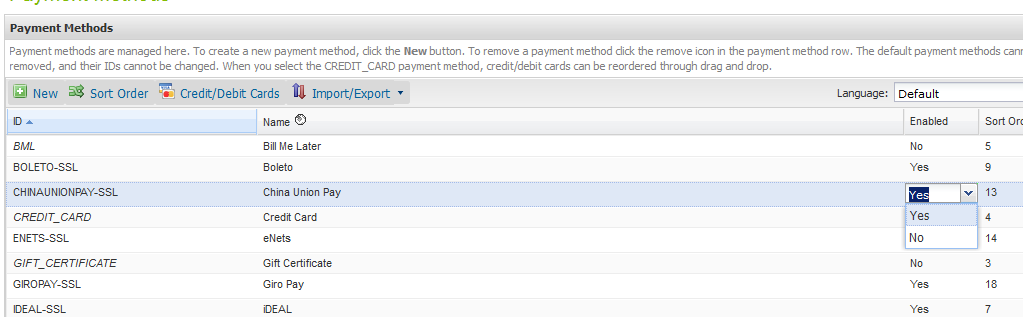
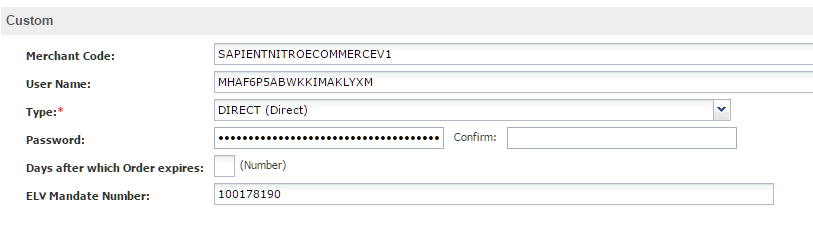


Fig: 2.4

Fig: 2.5

* Once the payment methods are added and enabled successfully, update credit or debit card details as provided by merchant under Payment method-> Credit/Debit Cards section.Enable the credit/debit card under **Credit/Debit Cards Section** that needs to be activated. Ensure custom attribute named “Worldpay Card Type” is populated for all active cards with corresponding Worldpay type like for VISA it is “VISA-SSL”. Repeat these steps for all the cards that are required to be supported on the site. (Refer to Fig: 2.6)

Set below card details of Payment method as provided by merchant. All these fields are optional (Refer to Fig: 2.6)

|  |  |
| --- | --- |
| **Card Attribute** | **Description** |
| Security Code Length | This field represents maximum length of security code for card type |
| Card Number Verification | This field represents maximum length of CVV for card type |
| Card Number Length | This field represents maximum length of card number(range between 13-16) for card type |
| Checksum Verification | This field is set to True if merchant wants to verify the card details |
| Worldpay Card Type | This field represents the card type in Worldpay like for VISA it is “VISA-SSL” |

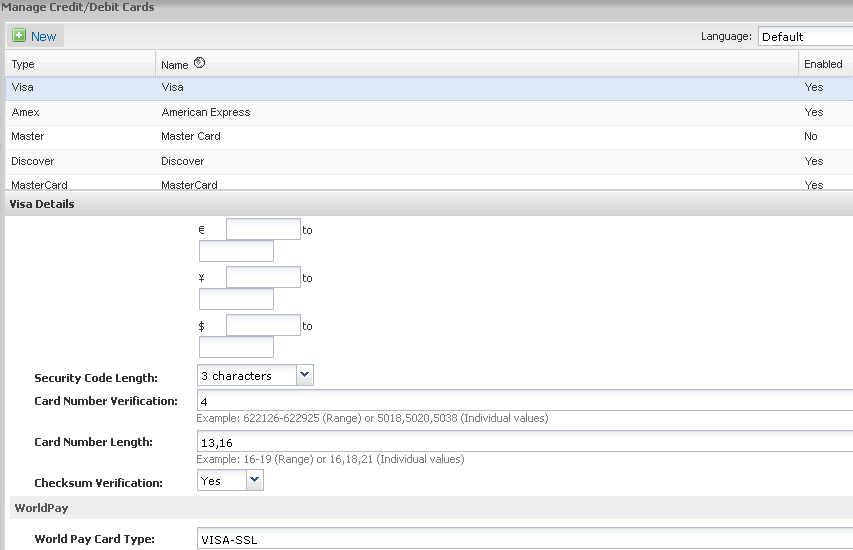
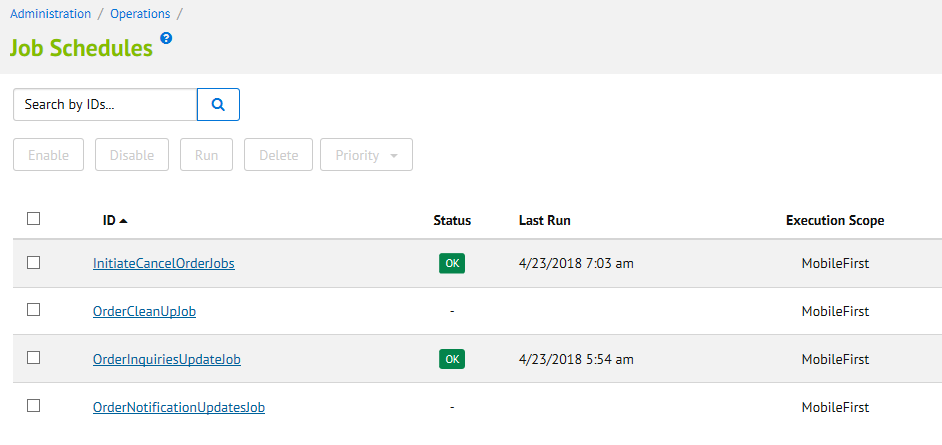


Fig: 2.6

### Importing Job

In the integration package, a job definition is provided in the worldpay\_jobs.xml file. All import files are placed in the import folder within the cartridge installation pack. To import this file, follow the instructions below:

1. In Business Manager navigate to *Administration🡪* *Operations🡪*  *Import & Export*.
2. Click on the upload button and in the next screen select the file by clicking the browse button. Notice that the file is in the import subdirectory. After you have selected the file click on the upload button.
3. After the file has been uploaded click on the back button. In the Job Schedules area, click on the Import button.
4. Select the file you just uploaded and click on the Next button.
5. After the file is successfully validated click on the Next button.
6. Make sure REPLACE is selected and click on the Import button.
7. Associate this job to respective merchant site.



Enable the jobs to be active, apply the configurations setting of scheduling into General and Sites sections one by one.

### Importing Content Asset

All import files are available in the import folder within the cartridge installation pack.

Note: Ensure content assets XML has appropriate library Id of the site where XML to be imported

1. In Business Manager, navigate to *MobileFirst Site🡪 Merchant Tools* *🡪 Content 🡪 Import & Export*.
2. Click on the upload button and in the next screen select file worldpay\_contentassets.xml by clicking the browse button. Notice that the file is in the import subdirectory. After you have selected the file click on the upload button.
3. After the file has been uploaded click on the back button. In the Content area, click on the Import button.
4. Select the file you just uploaded and click on the Next button.
5. After the file is successfully validated click on the Next button.
6. Select REPLACE and click on the Import button.
7. Navigate to *MobileFirst Site🡪Merchant Tools* *🡪 Content🡪Content Asset.*
8. Content Assets with IDs “worldpayhelper” and “worldpay-elv-consent” are added.

### Importing Service

To import the service file from integration package, follow the instructions below:

1. In Business Manager, navigate to Administration 🡪 Operations🡪 Import & Export.
2. Click on the upload button and in the next screen select the file worldpay\_service.xml by clicking the browse button. Notice that the file is in the import subdirectory. After you have selected the file click on the upload button.
3. After the file has been uploaded click on the back button. In the Import/export Service area, click on the Import button.
4. Select the file you just uploaded and click on the Next button.
5. After the file is successfully validated click on the Next button.
6. Select MERGE and click on the Import button.

Once the service object is created successfully, fill out the following properties under Services🡪Service Profile🡪 Worldpay Profile with the values listed against them:

|  |  |
| --- | --- |
| **Worldpay Profile** | **Description** |
| Name | Default “worldpayprofile”. Merchant can give any name as per need |
| Timeout (ms) | Default 30000 Timeout value in milli sec as per Merchant contract with Worldpay for webservice timeout |
| Circuit Breaker | Circuit break settings when connection not established for long time |
| Rate limits | Rate Limits after which service resumed back once circuit breaks |

Fill out the following properties under Worldpay credentials with the values listed against them:

All above service credentials are used by one Worldpay service only. (Refer Fig. 2.9)

|  |  |
| --- | --- |
| **Worldpay Credentials** | **Description** |
| Name | Merchant Hex Code provided by Worldpay |
| URL | This field represents the Web service URL that is invoked while calling Worldpay service. |
| User | This field represents the User Name provided specifically to the above mentioned Merchant Code. |
| Password | This field represents the Password provided specifically to the above mentioned Merchant Code. |

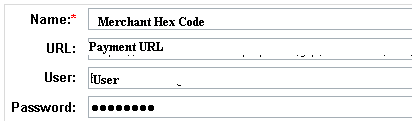


Fig 2.9

### IDEAL APM Integration

Bank configurations in Business Manager:

All the supported Bank List values are preconfigured. However, Modify/Delete the list if required. Follow the below steps to add additional banks:

1. Log into Business Manager
2. Navigate to *Administration 🡪 Site Development 🡪 System Object Definition 🡪 SitePreferences*
3. Open the tab Attribute Definition and find “*WorldpayIdealBankList*“
4. Add the required *Bank ID* and *Bank Name* in the *Value* and *Display Value* correspondingly.

(For more details refer Appendix B)

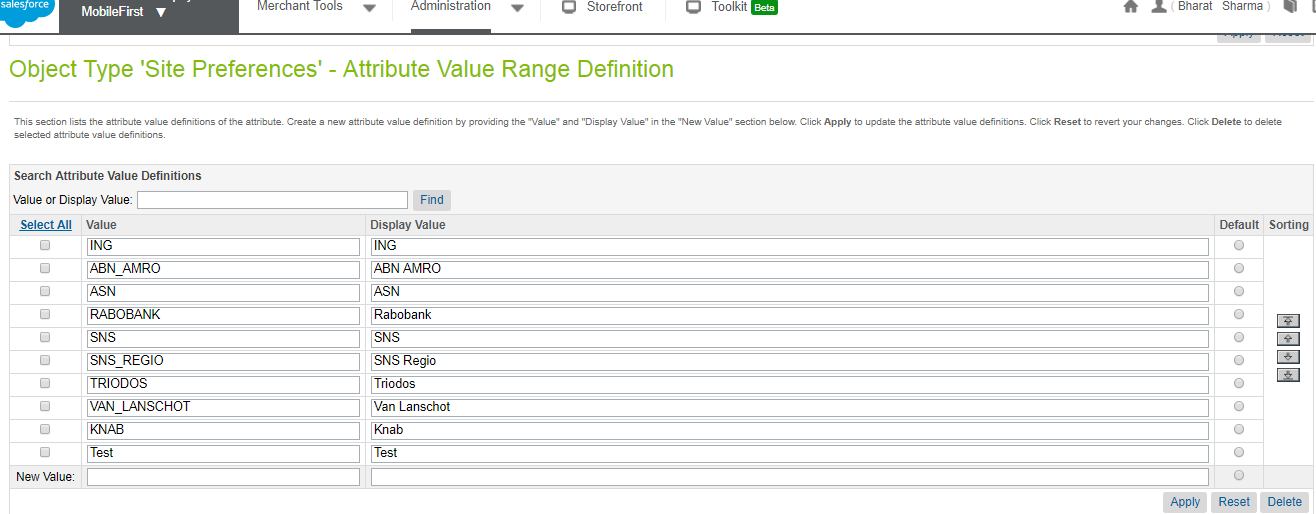


Fig: 2.6

Select the specific banks using the below steps:

1. Log into Business Manager
2. From the top bar select the required site for which configurations are to be done.
3. Navigate to MobileFirst🡪Merchant Tools🡪 *Site Preferences 🡪 Custom Preferences 🡪 Worldpay*
4. Select the banks to be supported. (Multiple banks can be selected from the drop down)

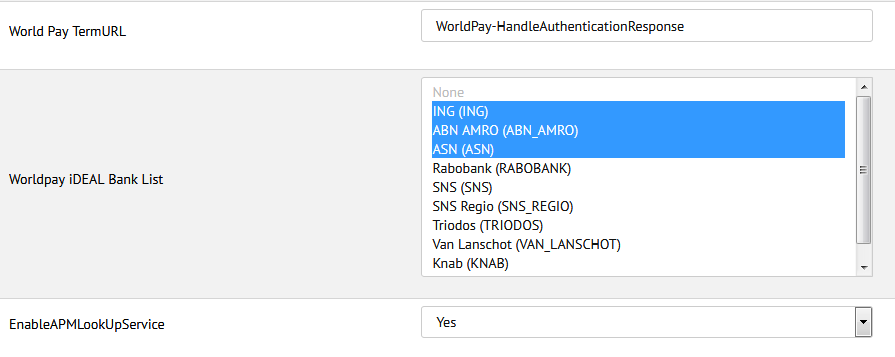


Fig: 2.7

**Note:** Please ensure list of bank should be updated; please contact Worldpay to get the latest list of Bank.

### BMultiple Merchant ID Support Integration

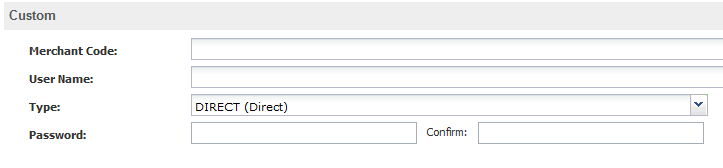
After importing metadata provided with Worldpay cartridge, additional fields (merchant ID, username, and password) will be created in payment methods in BM.

1. Log into Business Manager
2. From the top bar select the required site for which configurations are to be done
3. Navigate to MobileFirst Site🡪Merchant Tools 🡪 Ordering🡪 Payment Methods.
4. Input the required details for the below field:

Merchant ID

Username

Password



### Look and Feel Customization for Worldpay Redirect Pages

Configurations that needs to be done in Business Manager to support look and feel customization:

1) Log into Business Manager

2) From the top bar select the required site for which configurations are to be done.

3) Navigate to SitePreferences 🡪 Custom Preferences 🡪 Worldpay

4) Input the required details for the below field:

Worldpay Configurable HPP APM

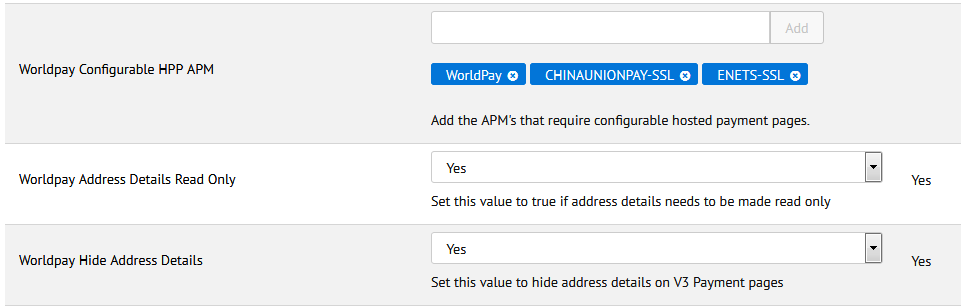
Worldpay Installation Id

Worldpay Address Details Read Only

Worldpay Hide Address Details

\*\* Ensure that the Payment Method ID (refer to APPENDIX A – KEY column), which needs to be supported for customizable is added to the **Worldpay Configurable HPP APM** sitepreference.

\*\*Please contact the Worldpay contact person for values of the above details.



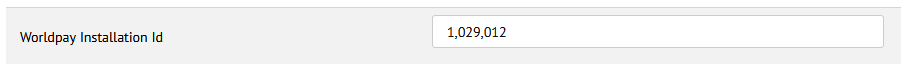


Fig 2.9

### Hosted Payment Pages iframe or lightbox for Worldpay Redirect Payment Method

Configurations that needs to be done in Business Manager to support look and feel customization:

1) Log into Business Manager

2) Navigate to MobileFirst 🡪 Merchant Tools *🡪 Ordering 🡪 Payment Methods*

3) Select “Credit Card Redirect” payment method and add the following data under worldpay HPP CustomOptions JSON attribute.

4) Write JSON specifically type as iframe or lightbox. Click on Apply to enable the Iframe/Light box on HPP.

Example:

{

“type”:”iframe”

}

OR

{

                "type": "lightbox",

                 "lightboxMaskOpacity": 50,

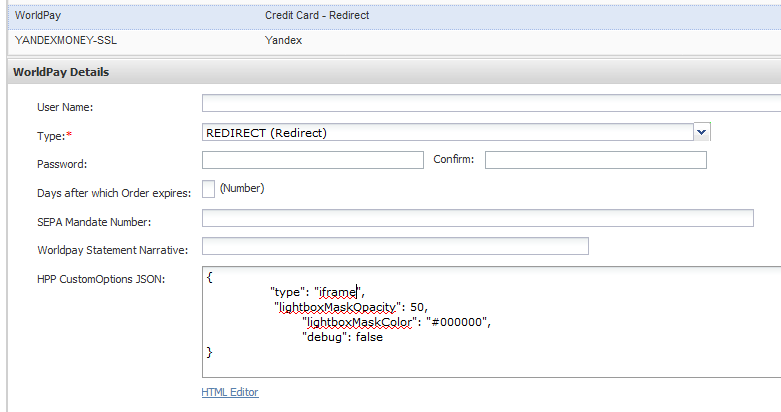
                  "lightboxMaskColor": "#000000",

               "debug": false

}

For lightbox JSON settings below two attributes can be specified with merchant specific values.

* lightboxMaskOpacity
* lightboxMaskColor



### Country Currency Mapping

|  |  |
| --- | --- |
| **Country** | **Currency** |
| United States | US Dollars |
| UNITED KINGDOM | Pound Sterling |
| GERMANY | Euro |
| SINGAPORE | Singapore Dollar |
| JAPAN | Japanese Yen |
| SWEDEN | Yuan Renminbi |
| CHINA | Euro |
| FRANCE | Euro |
| RUSSIAN FEDERATION | Russian Ruble |
| BELGIUM | Euro |
| ITALY | Euro |
| CANADA | Canadian Dollar |
| AUSTRALIA | Australian Dollar |

### Country Currency and Locale Mapping for Klarna

|  |  |  |  |
| --- | --- | --- | --- |
| purchase\_country | purchase\_currency | locale | extra locale |
| AT | Euro | de-at | en-at |
| FI | Euro | fi-fi | en-fi , sv-fi |
| DE | Euro | de-de | en-de |
| NL | Euro | nl-nl | en-nl |
| NO | NOK | nb-no | en-no |
| SE | SEK | sv-se | en-se |
| GB | GBP | gb-gb | en-sb |

* Ensure the appropriate locale ,currencies and countries setting are available in business manager.

## CUSTOM CODE

**Pre-Requisite:** Make sure the cartridge of site MFRA “link\_worldpay:link\_worldpay\_core:app\_storefront\_base” is specified in Site Settings path under Manage Sites > Merchant Site

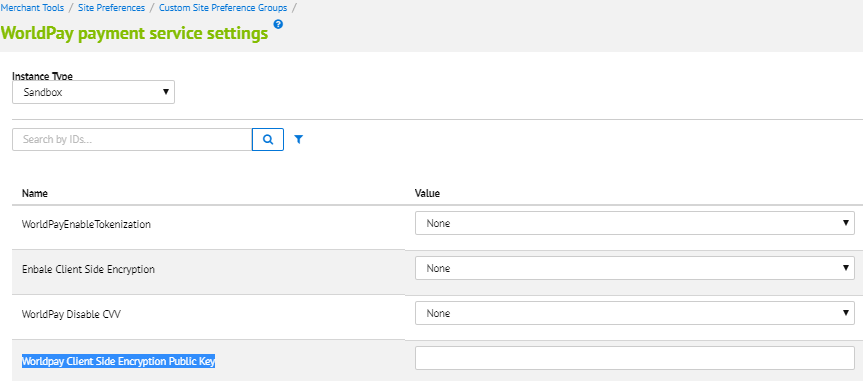
This section covers only MFRA changes required in int\_worldpay\_controllers to support controllers to work.

### Error Messaging

In cartridge all the error messaging is achieved using properties file (worldpayerror.properties). These messages can be configured while installation of the cartridge and can be modified based upon the merchants business requirement. Using properties file for messaging gives the Merchant the advantage for supporting multi lingual site.

### Client Side Encryption

Please get the client side encryption from the worldpay merchant interface in order to encrypt the Card PII data. This will prevent the exposure of Card PII over the network. Download the public key of Client side encryption from worldpay interface and configure it in the site preference (‘Worldpay Client Side Encryption Public Key’) available in business manager (Mobile First Merchant Tools>Site Preferences>Custom Site Preference Groups).



### Form File Changes

Disclaimer: shipping and billing address in forms must have country code in UPPERCASE as Worldpay mandatory requirement. We have delivered only default locale form files. If merchant require any specific locale they can copy the required form to respective locales.

Purpose: Worldpay XML request must have country code in UPPERCASE.

## CUSTOM CODE FOR CONTROLLERS

Pre-Requisite: Make sure the controller cartridge of site site-genesis is (say, e.g. app\_sitegenesis\_controllers) and “int\_worldpay\_controllers” are specified in Site Settings path under Manage Sites > Merchant Site

This section covers only controller changes required in int\_worldpay\_controllers to support controllers to work.

### Generic Section

**Replacing the references of SiteGenesiscontroller in WorldPay Cartridge**

Replace reference of storefront cartridge name in Global variables (say, app\_storefront\_controllers) with sitegenesis controller cartridge name (say, app\_sitegenesis\_controllers)

This modification is required in two controller files “WorldPay.ds” and “WorldPayHelper.ds”



Replace reference of storefront cartridge name in ISML with sitegenesis core cartridge name (say, app\_sitegenesis\_core)

This modification is required in ISML file “worldpaypaymentmethods.isml”, code specified below

var currentCountry = require('app\_sitegenesis\_core/cartridge/scripts/util/Countries').getCurrent(pdict);

#### Controller – COBilling.js

##### Update “returnToForm” Function

1. Include script module at the top of script include section

Include global variable below the var guard

|  |
| --- |
| var WorldpayHelper = require('link\_worldpay\_controllers/cartridge/scripts/WorldpayHelper'); |

1. Replace params section by below code snippet under the function returnToForm to handle error cases

|  |
| --- |
| function returnToForm(cart, params) {  var pageMeta = require('~/cartridge/scripts/meta');  // if the payment method is set to gift certificate get the gift certificate code from the form  if (!empty(cart.getPaymentInstrument()) && cart.getPaymentInstrument().getPaymentMethod() === PaymentInstrument.METHOD\_GIFT\_CERTIFICATE) {  app.getForm('billing').copyFrom({  giftCertCode: cart.getPaymentInstrument().getGiftCertificateCode()  });  }  pageMeta.update({  pageTitle: Resource.msg('billing.meta.pagetitle', 'checkout', 'SiteGenesis Checkout')  });  if (params && params.errorMessage) {  app.getView(require('~/cartridge/scripts/object').extend(params, {  Basket : cart.object,  ContinueURL : URLUtils.https('COBilling-Billing'),  errorMessage : params.errorMessage  })).render('checkout/billing/billing');  } else if (!empty(params)) {  app.getView(require('~/cartridge/scripts/object').extend(params, {  Basket: cart.object,  ContinueURL: URLUtils.https('COBilling-Billing')  })).render('checkout/billing/billing');  } else {  app.getView({  Basket: cart.object,  ContinueURL: URLUtils.https('COBilling-Billing')  }).render('checkout/billing/billing');  }  } |

##### Update “initCreditCardList” function

* Update below code snippet under initCreditCardList to add worldpay available credit cards

|  |
| --- |
| function initCreditCardList(cart) {  var paymentAmount = cart.getNonGiftCertificateAmount();  var countryCode;  var applicablePaymentMethods;  var applicablePaymentCards;  var applicableCreditCards;  countryCode = cart.object.billingAddress && cart.object.billingAddress.countryCode.value ? cart.object.billingAddress.countryCode.value : Countries.getCurrent({  CurrentRequest: {  locale: request.locale  }  }).countryCode;    applicablePaymentMethods = PaymentMgr.getApplicablePaymentMethods(customer, countryCode, paymentAmount.value);  applicablePaymentCards = PaymentMgr.getPaymentMethod(PaymentInstrument.METHOD\_CREDIT\_CARD).getApplicablePaymentCards(customer, countryCode, paymentAmount.value);  app.getForm('billing').object.paymentMethods.creditCard.type.setOptions(applicablePaymentCards.iterator());  applicableCreditCards = null;  if (customer.authenticated) {  var profile = app.getModel('Profile').get();  if (profile) {  applicableCreditCards = profile.validateWalletPaymentInstruments(countryCode, paymentAmount.getValue()).ValidPaymentInstruments;  }  applicableCreditCards = WorldpayHelper.AvailableCreditCards(applicableCreditCards);  }  return {  ApplicablePaymentMethods: applicablePaymentMethods,  ApplicableCreditCards: applicableCreditCards  };  } |

##### Update “publicStart” function

* Update the method to handle error messages

|  |
| --- |
| function publicStart(errorMessage) {  var cart = app.getModel('Cart').get();  if (cart) {  // Initializes all forms of the billing page including: - address form - email address - coupon form  initAddressForm(cart);  initEmailAddress(cart);  var creditCardList = initCreditCardList(cart);  var applicablePaymentMethods = creditCardList.ApplicablePaymentMethods;  var billingForm = app.getForm('billing').object;  var paymentMethods = billingForm.paymentMethods;  if (paymentMethods.valid) {  paymentMethods.selectedPaymentMethodID.setOptions(applicablePaymentMethods.iterator());  } else {  paymentMethods.clearFormElement();  }  app.getForm('billing.couponCode').clear();  app.getForm('billing.giftCertCode').clear();  if(errorMessage && errorMessage.errorMessage){  start(cart, require('~/cartridge/scripts/object').extend(errorMessage, {ApplicableCreditCards: creditCardList.ApplicableCreditCards}));  }  else{  start(cart, {ApplicableCreditCards: creditCardList.ApplicableCreditCards});  }  } else {  app.getController('Cart').Show();  }  } |

* Also update creditcardlist under local methods

exports.InitCreditCardList=initCreditCardList; and exports.HandlePaymentSelection = handlePaymentSelection;

##### Update “validatePayment” function

* Update the method with the countrycode conditions.

|  |
| --- |
| function validatePayment(cart) {  var paymentAmount, countryCode, invalidPaymentInstruments, result;  if (app.getForm('billing').object.fulfilled.value) {  paymentAmount = cart.getNonGiftCertificateAmount();  countryCode = cart.object.billingAddress && cart.object.billingAddress.countryCode ? cart.object.billingAddress.countryCode : Countries.getCurrent({  CurrentRequest: {  locale: request.locale  }  }).countryCode;  invalidPaymentInstruments = cart.validatePaymentInstruments(customer, countryCode, paymentAmount.value).InvalidPaymentInstruments;  if (!invalidPaymentInstruments && cart.calculatePaymentTransactionTotal()) {  result = true;  } else {  app.getForm('billing').object.fulfilled.value = false;  result = false;  }  } else {  result = false;  }  return result;  } |

##### Update “saveCreditCard” function

* Update the method to update worldpay card details. This will ensure that duplicate card won’t be saved from My Account and order placement

|  |
| --- |
| function saveCreditCard() {  var i, creditCards, GetCustomerPaymentInstrumentsResult, newCreditCard;  if (customer.authenticated && app.getForm('billing').object.paymentMethods.creditCard.saveCard.value  && app.getForm('billing').object.paymentMethods.selectedPaymentMethodID.value != Resource.msg('orderdetails.paymentmethod.worldpay','worldpay',null)) {  // var creditCards = customer.getProfile().getWallet().getPaymentInstruments(PaymentInstrument.METHOD\_CREDIT\_CARD);  var ArrayList = require('dw/util/ArrayList');  var Pipelet = require('dw/system/Pipelet');  creditCards = new ArrayList();  GetCustomerPaymentInstrumentsResult = new Pipelet('GetCustomerPaymentInstruments').execute({  Customer: customer,  PaymentMethod: PaymentInstrument.METHOD\_CREDIT\_CARD  });  if (GetCustomerPaymentInstrumentsResult.result !== PIPELET\_ERROR) {  creditCards = GetCustomerPaymentInstrumentsResult.PaymentInstruments;  }  Transaction.wrap(function () {  newCreditCard = customer.getProfile().getWallet().createPaymentInstrument(PaymentInstrument.METHOD\_CREDIT\_CARD);  // copy the credit card details to the payment instrument  newCreditCard.setCreditCardHolder(app.getForm('billing').object.paymentMethods.creditCard.owner.value);  newCreditCard.setCreditCardNumber(app.getForm('billing').object.paymentMethods.creditCard.number.value);  newCreditCard.setCreditCardExpirationMonth(app.getForm('billing').object.paymentMethods.creditCard.expiration.month.value);  newCreditCard.setCreditCardExpirationYear(app.getForm('billing').object.paymentMethods.creditCard.expiration.year.value);  newCreditCard.setCreditCardType(app.getForm('billing').object.paymentMethods.creditCard.type.value);  for (i = 0; i < creditCards.length; i++) {  var creditcard = creditCards[i];  if (creditcard.getCreditCardExpirationMonth() === newCreditCard.getCreditCardExpirationMonth()  && creditcard.getCreditCardExpirationYear() === newCreditCard.getCreditCardExpirationYear()  && creditcard.creditCardNumber.substring(creditcard.creditCardNumber.length - 4).equals(newCreditCard.creditCardNumber.substring(newCreditCard.creditCardNumber.length - 4))  ) {  WorldpayHelper.ReplaceToken(newCreditCard, creditcard);  customer.getProfile().getWallet().removePaymentInstrument(creditcard);  }  }  });  }  return true;  } |

#### Controller – COPlaceOrder.js

##### Update “API Include” module

* Include script module at the top of script include section

|  |
| --- |
| var PaymentProcessor = app.getModel('PaymentProcessor');  var Worldpay = require('link\_worldpay\_controllers/cartridge/controllers/Worldpay'); |

##### Update “handlePayments” method

* Add below section after authorizationResult received to handle the authorize result for Redirection and 3D and error message and konbini APM

|  |
| --- |
| function handlePayments(order) {  if (order.getTotalNetPrice() !== 0.00) {  var paymentInstruments = order.getPaymentInstruments();  if (paymentInstruments.length === 0) {  return {  missingPaymentInfo: true  };  }  /\*\*  \* Sets the transaction ID for the payment instrument.  \*/  var handlePaymentTransaction = function () {  paymentInstrument.getPaymentTransaction().setTransactionID(order.getOrderNo());  };  for (var i = 0; i < paymentInstruments.length; i++) {  var paymentInstrument = paymentInstruments[i];  if (PaymentMgr.getPaymentMethod(paymentInstrument.getPaymentMethod()).getPaymentProcessor() === null) {  Transaction.wrap(handlePaymentTransaction);  } else {  var authorizationResult = PaymentProcessor.authorize(order, paymentInstrument);  if(authorizationResult.returnToPage){  return {  returnToPage :true,  order : order,  klarnasnippet: authorizationResult.klarnasnippet  };  } else if(authorizationResult.redirect || authorizationResult.worldpayredirect){  return {  redirect :true,  redirectUrl : authorizationResult.redirectUrl  };  } else if(authorizationResult.is3D){  return {  is3D : true,  redirectUrl : authorizationResult.redirectUrl,  paRequest : authorizationResult.paRequest,  termUrl : authorizationResult.termUrl,  orderNo : order.orderNo  };  }  else if (authorizationResult.not\_supported || authorizationResult.error) {  return {  error : true,  errorCode : authorizationResult.errorCode,  errorMessage : authorizationResult.errorMessage  };  }else if (authorizationResult.authorized && !empty(authorizationResult.isPaymentKonbini) && authorizationResult.isPaymentKonbini) {  return {  isPaymentKonbini : true,  WorldpayRedirectURL : authorizationResult.redirectUrl  };  }  }  }  }  return {};  } |

##### Update “start” method

* Replace the initial code that return to Cart.Show() if cart does not exist with the code below by adding if condition
* Move SaveCreditCard functionality of COBilling and place the code just before submitImpl function at the end of function else condition for above cart and handle the APM’s redirections by replacing the code with below

|  |
| --- |
| function start() {  var cart = Cart.get();  if (!cart) {  app.getController('Cart').Show();  return {};  }  var COShipping = app.getController('COShipping');  // Clean shipments.  COShipping.PrepareShipments(cart);  // Make sure there is a valid shipping address, accounting for gift certificates that do not have one.  if (cart.getProductLineItems().size() > 0 && cart.getDefaultShipment().getShippingAddress() === null) {  COShipping.Start();  return {};  }  // Make sure the billing step is fulfilled, otherwise restart checkout.  if (!session.forms.billing.fulfilled.value) {  app.getController('COCustomer').Start();  return {};  }  Transaction.wrap(function () {  cart.calculate();  });  var COBilling = app.getController('COBilling');  Transaction.wrap(function () {  if (!COBilling.ValidatePayment(cart)) {  COBilling.Start();  return {};  }  });  // Recalculate the payments. If there is only gift certificates, make sure it covers the order total, if not  // back to billing page.  Transaction.wrap(function () {  if (!cart.calculatePaymentTransactionTotal()) {  COBilling.Start();  return {};  }  });  // Handle used addresses and credit cards.  var saveCCResult = COBilling.SaveCreditCard();  if (!saveCCResult) {  return {  error: true,  PlaceOrderError: new Status(Status.ERROR, 'confirm.error.technical')  };  }  // Creates a new order. This will internally ReserveInventoryForOrder and will create a new Order with status  // 'Created'.  var order = cart.createOrder();  if (!order) {  app.getController('Cart').Show();  return {};  }  var handlePaymentsResult = handlePayments(order);  if (handlePaymentsResult.error) {  Transaction.wrap(function(){  OrderMgr.failOrder(order);  });  app.getController('COBilling').Start({'errorMessage' : handlePaymentsResult.errorMessage});  return {error : false};  } else if(handlePaymentsResult.is3D) {  return {  is3D : true,  redirectUrl : handlePaymentsResult.redirectUrl,  paRequest : handlePaymentsResult.paRequest,  termUrl : handlePaymentsResult.termUrl,  orderNo :handlePaymentsResult.orderNo  };  } else if(handlePaymentsResult.returnToPage) {  app.getView({  Order : handlePaymentsResult.order,  klarnasnippet:handlePaymentsResult.klarnasnippet  }).render('checkout/summary/summary');  return {};  } else if(handlePaymentsResult.redirect) {  return {redirect :true,  redirectUrl:handlePaymentsResult.redirectUrl  };  } else if (handlePaymentsResult.missingPaymentInfo) {  Transaction.wrap(function(){  OrderMgr.failOrder(order);  });  app.getController('COBilling').Start({'errorMessage' : handlePaymentsResult.errorMessage});  return {error : true};  } else if (handlePaymentsResult.isPaymentKonbini) {  return {  Order : order,  order\_created : true,  WorldpayRedirectURL : handlePaymentsResult.WorldpayRedirectURL  };  }    // Handle used addresses and credit cards.  var saveCCResult = COBilling.SaveCreditCard();  if (!saveCCResult) {  return {  error: true,  PlaceOrderError: new Status(Status.ERROR, 'confirm.error.technical')  };  }  return submitImpl(order);  } |

##### Update “submit” method

* Replace the existing asynchronous callback for order placement submission flow submit() function with the code below:

|  |
| --- |
| function submit() {    var isWorldpayPaymentProcessor = false;  var WorldpayHelper = require('link\_worldpay\_controllers/cartridge/scripts/WorldpayHelper');  var order = WorldpayHelper.GetOrder().Order;  var paymentMethod:String;  if (order!=null) {  //Check payment processor as worldpay  var paymentInstruments = order.getPaymentInstruments();  if (paymentInstruments.length > 0) {  for (var i = 0; i < paymentInstruments.length; i++) {  var paymentInstrument = paymentInstruments[i];  var paymentProcessor = PaymentMgr.getPaymentMethod(paymentInstrument.getPaymentMethod()).getPaymentProcessor();  if (paymentProcessor != null && paymentProcessor.getID().equals("Worldpay")) {  isWorldpayPaymentProcessor = true;  paymentMethod = paymentInstrument.paymentMethod;  break;  } } }  } else {  app.getController('Cart').Show();  return {success : false}; }  if (isWorldpayPaymentProcessor) {  /\*var Order = require('dw/order/Order');  var paymentStatus = request.querystring.paymentStatus;  Logger.getLogger('worldpay').debug(req.querystring.order\_id + ' orderid COPlaceOrder paymentStatus ' + paymentStatus);\*/  if(paymentMethod.equals(dw.order.OrderPaymentInstrument.METHOD\_CREDIT\_CARD)){  if (submitImpl(order).error) {  app.getController('COSummary').Start();  return;  }  app.getController('COSummary').ShowConfirmation(order);  return;  }  var redirectionResult = Worldpay.HandleRedirection(order);  if (redirectionResult.success) {  if (submitImpl(order).error) {  app.getController('COSummary').Start();  return;  }  app.getController('COSummary').ShowConfirmation(order, redirectionResult.klarnasnippet);  return;  }  return;  } else {  var Logger = require('dw/system/Logger');  Logger.getLogger("worldpay").debug("AuthorizeOrder.ds : not worldpay order anuj order" );  if (submitImpl(order.object).error) {  app.getController('COSummary').Start();  return;  }  app.getController('COSummary').ShowConfirmation(order);  return;  }  } |

##### Update “submitImpl” method

* Add submitImpl method in module export section at the end
* exports.SubmitImpl = guard.ensure(['https'], submitImpl);
* Add “submitImpl “function with the code below

|  |
| --- |
| function submitImpl(order) {  var orderPlacementStatus = Transaction.wrap(function () {  if (OrderMgr.placeOrder(order) === Status.ERROR) {  OrderMgr.failOrder(order);  app.getController('COBilling').Start();  return false;  }  order.setConfirmationStatus(order.CONFIRMATION\_STATUS\_CONFIRMED);  return true;  });  if (orderPlacementStatus === Status.ERROR) {  return {error: true};  }  // Creates purchased gift certificates with this order.  /\*if (!createGiftCertificates(order)) {  Transaction.wrap(function () {  OrderMgr.failOrder(order);  });  app.getController('COBilling').Start();  return {error: true};  }\*/  // Send order confirmation and clear used forms within the checkout process.  var Email = app.getModel('Email');  var Resource = require('dw/web/Resource');  Email.get('mail/orderconfirmation', order.getCustomerEmail())  .setSubject((Resource.msg('order.orderconfirmation-email.001', 'order', null) + ' ' + order.getOrderNo()).toString())  .send({  Order: order  });  // Mark order as EXPORT\_STATUS\_READY.  Transaction.wrap(function () {  order.setExportStatus(dw.order.Order.EXPORT\_STATUS\_READY);  order.setConfirmationStatus(dw.order.Order.CONFIRMATION\_STATUS\_CONFIRMED);  });  // Clears all forms used in the checkout process.  session.forms.singleshipping.clearFormElement();  session.forms.multishipping.clearFormElement();  session.forms.billing.clearFormElement();  return {  Order: order,  order\_created: true  };  } |

#### Controller – COSummary.js

##### Update “API Include” module

* Include script module at the top of script include section

|  |
| --- |
| var Cart = app.getModel('Cart');  var ISML = require('dw/template/ISML');  var OrderMgr = require('dw/order/OrderMgr');  var Order = app.getModel('Order'); |

##### Update “submit” method

* Replace the existing submit method with the code

|  |
| --- |
| function submit() {  // Calls the COPlaceOrder controller that does the place order action and any payment authorization.  // COPlaceOrder returns a JSON object with an order\_created key and a boolean value if the order was created successfully.  // If the order creation failed, it returns a JSON object with an error key and a boolean value.  var placeOrderResult = app.getController('COPlaceOrder').Start();  if (placeOrderResult.error) {  start();  return;  }else if(placeOrderResult.redirect){  app.getView({  ContinueURL: placeOrderResult.redirectUrl,  Location :placeOrderResult.redirectUrl  }).render('util/redirect');  return;  }else if(placeOrderResult.is3D){  var OrderObj: Order = OrderMgr.getOrder(placeOrderResult.orderNo);  var Resource = require('dw/web/Resource');  var error3D1 = Resource.msg('worldpay.redirect.error1','worldpayerror',null);  var error3D2 = Resource.msg('worldpay.redirect.error2','worldpayerror',null);  var message3D = Resource.msg('worldpay.3dsecure.message','worldpay',null);    ISML.renderTemplate('worldpayissuerredirect',{  ContinueURL: placeOrderResult.redirectUrl,  IssuerURL:placeOrderResult.redirectUrl,  PaRequest:placeOrderResult.paRequest,  TermURL:placeOrderResult.termUrl,  Order:OrderObj,  error3D1: error3D1,  error3D2: error3D2,  message3D: message3D  });  return;  }  else if (placeOrderResult.order\_created) {  if (placeOrderResult.WorldpayRedirectURL) {  showConfirmation(placeOrderResult.Order, placeOrderResult.WorldpayRedirectURL);  } else {  showConfirmation(placeOrderResult.Order);  }  return;  }  } |

##### Update “showConfirmation” method

* Add another parameter named “worldpayRedirectURL” to function “showConfirmation”and also pass the same to view. Code snippet specified below

|  |
| --- |
| function showConfirmation(order,worldPayRedirectURL) {  if (!customer.authenticated) {  // Initializes the account creation form for guest checkouts by populating the first and last name with the  // used billing address.  var customerForm = app.getForm('profile.customer');  customerForm.setValue('firstname', order.billingAddress.firstName);  customerForm.setValue('lastname', order.billingAddress.lastName);  customerForm.setValue('email', order.customerEmail);  customerForm.setValue('orderNo', order.orderNo);  }  app.getForm('profile.login.passwordconfirm').clear();  app.getForm('profile.login.password').clear();  var pageMeta = require('~/cartridge/scripts/meta');  pageMeta.update({pageTitle: Resource.msg('confirmation.meta.pagetitle', 'checkout', 'SiteGenesis Checkout Confirmation')});  app.getView({  Order: order,  WorldpayRedirectURL : worldPayRedirectURL,  ContinueURL: URLUtils.https('Account-RegistrationForm') // needed by registration form after anonymous checkouts  }).render('checkout/confirmation/confirmation');  } |

##### Add “submitOrder” method

* Add new flow “COSummary-SubmitOrder” which gets invoked when user click “Place order” button on order review page for “Credit Card Redirect” payment method implementation where merchant desire to implement Hosted Payment Pages rendering as IFRAME or LIGHTBOX.
* Add below new code snippet under COSummary and export the method at the end

exports.SubmitOrder = guard.ensure(['https'], submitOrder);

|  |
| --- |
| function submitOrder() {  var cart = Cart.get();  if (cart) {  submit();  return;  } else if (!empty(session.privacy.order\_id)) {  app.getView({  Order : Order.get(session.privacy.order\_id).object  }).render('checkout/summary/summary');  return;  } else {  app.getController('Cart').Show();  return {};  }  } |

#### Controller – PaymentInstruments.js

##### Update “create” method

* Modify create function by adding below code to stop adding duplicate cards in worldPay integration

|  |
| --- |
| if (isDuplicateCard) {  wallet.removePaymentInstrument(oldCard);  }    var WorldpayHelper = require('link\_worldpay\_controllers/cartridge/scripts/WorldpayHelper');  WorldpayHelper.ReplacePaymentInstrument(paymentInstrument, paymentInstruments);  Transaction.commit(); |

#### ISML Template changes

##### Update “summary.isml”

* Remove below code snippet in summary.isml

|  |
| --- |
| <isif condition="${pdict.PlaceOrderError != null}">  <div class="error-form">${Resource.msg(pdict.PlaceOrderError.code,'checkout',null)}</div> |

* Also replace “pdict.Basket” with “LineCntr” in file.
* Replace <isreportcheckout tag and <ischeckoutprogressindicator tag sections in summary.isml on the summary page with below code snippet. This code snippet determine if the page is summary page or summary page with IFRAME/Lightbox

|  |
| --- |
| <isif condition="${!empty(pdict.Basket)}">  <isset name="LineCntr" value="${pdict.Basket}" scope="page"/>  <iselseif condition="${!empty(pdict.Order)}">  <isset name="LineCntr" value="${pdict.Order}" scope="page"/>  </isif>  <isset name="summaryaction" value="${URLUtils.https('COSummary-Submit')}" scope="page" />  <script src="${URLUtils.staticURL('/lib/jquery/jquery-1.11.1.min.js')}" type="text/javascript"></script>  <isif condition="${!empty(LineCntr.getPaymentInstruments())}">  <isloop items="${LineCntr.getPaymentInstruments()}" var="paymentInstr" status="loopstate">  <isif condition="${dw.order.PaymentMgr.getPaymentMethod(paymentInstr.paymentMethod).ID=='Worldpay'}">  <isscript>  var isValidCustomOptionsHPP = require('link\_worldpay\_core/cartridge/scripts/common/Utils').isValidCustomOptionsHPP(dw.order.PaymentMgr.getPaymentMethod(paymentInstr.paymentMethod));  if (!empty(isValidCustomOptionsHPP) && empty(pdict.Basket)) {  var CustomOptionsHPPJSON = require('link\_worldpay\_core/cartridge/scripts/common/Utils').getCustomOptionsHPP(dw.order.PaymentMgr.getPaymentMethod(paymentInstr.paymentMethod),  paymentInstr.custom.worldpayRedirectURL, LineCntr.getOrderNo(), LineCntr.getOrderToken(), paymentInstr.custom.preferedCard);  }  </isscript>  </isif>  </isloop>  </isif>  <isif condition="${typeof isValidCustomOptionsHPP !== 'undefined' && !empty(isValidCustomOptionsHPP)}">  <isset name="summaryaction" value="${URLUtils.https('COSummary-SubmitOrder')}" scope="page" />  </isif>  <isif condition="${!empty(pdict.Basket)}">  <isreportcheckout checkoutstep="${5}" checkoutname="${'OrderSummary'}"/>  <isif condition="${!pdict.CurrentForms.multishipping.entered.value}">  <ischeckoutprogressindicator step="3" multishipping="false" rendershipping="${LineCntr.productLineItems.size() == 0 ? 'false' : 'true'}"/>  <iselse/>  <ischeckoutprogressindicator step="4" multishipping="true" rendershipping="${LineCntr.productLineItems.size() == 0 ? 'false' : 'true'}"/>  </isif>  </isif> |

* Replace form action “ <isset name="summaryaction" value="${summaryaction}" scope="page" />” with “<isset name="summaryaction" value="${URLUtils.https('COSummary-Submit')}" scope="page" />” and wrap the form tag having Place Order button inside if condition where basket not empty as <isif condition="${!empty(pdict.Basket)}"></isif>.
* At end of file before closing </isdecorate> tag insert below line as

|  |
| --- |
| <div class="order-summary-footer">  <div class="place-order-totals">  <isordertotals p\_lineitemctnr="${LineCntr}" p\_showshipmentinfo="${false}" p\_shipmenteditable="${false}" p\_totallabel="${Resource.msg('summary.ordertotal','checkout',null)}"/>  </div>  <isif condition="${!empty(pdict.Basket)}">  <form action="${summaryaction}" method="post" class="submit-order">  <fieldset>  <div class="form-row">  <a class="back-to-cart" href="${URLUtils.url('Cart-Show')}">  <isprint value="${Resource.msg('summary.editcart','checkout',null)}" encoding="off" />  </a>  <isif condition="${pdict.Basket.paymentInstruments[0].getPaymentMethod().equals('KLARNA-SSL') || pdict.Basket.paymentInstruments[0].getPaymentMethod().equals('Worldpay')} ">  <button class="button-fancy-large" type="submit" name="submit" value="${Resource.msg('global.submitorderklarna','locale',null)}">  ${Resource.msg('global.submitorderklarna','locale',null)}  </button>  <iselse>  <button class="button-fancy-large" type="submit" name="submit" value="${Resource.msg('global.submitorder','locale',null)}">  ${Resource.msg('global.submitorder','locale',null)}  </button>  </isif>  </div>  <input type="hidden" name="${dw.web.CSRFProtection.getTokenName()}" value="${dw.web.CSRFProtection.generateToken()}"/>    </fieldset>  </form>  </isif>  </div>  <isinclude template="worldpaysummary"/>  </isdecorate> |

##### Update “minisummary.isml”

* Replace “pdict.Basket” with “lineCtnr” in file.
* Add below code snippet in minisummary.isml after <isinclude> tag of modules inclusion. This code snippet determine basket or order whichever available taken as lineCtntr and it determine if the page is summary page or summary page with IFRAME/lightbox

|  |
| --- |
| <isif condition="${!empty(pdict.Basket)}">  <isset name="lineCtnr" value="${pdict.Basket}" scope="page"/>  <iselseif condition="${!empty(pdict.Order)}">  <isset name="lineCtnr" value="${pdict.Order}" scope="page"/>  </isif> |

* Replace <isif condition="${checkoutstep <= 5}"> with <isif condition="${checkoutstep <= 6}">
* Wrap editURL <a> tag with non-empty basket if condition

|  |
| --- |
| <isset name="editUrl" value="${URLUtils.url('Cart-Show')}" scope="page"/>  <isif condition="${lineCtnr.productLineItems.size() == 0 && lineCtnr.giftCertificateLineItems.size() == 1}">  <isset name="editUrl" value="${URLUtils.url('GiftCert-Edit','GiftCertificateLineItemID', lineCtnr.giftCertificateLineItems[0].UUID)}" scope="page"/>  </isif>  ${Resource.msg('summary.title','checkout',null)} <isif condition="${!empty(pdict.Basket)}"><a class="section-header-note" href="${editUrl}">${Resource.msg('global.edit','locale',null)}</a></isif>  </h3>    <div class="checkout-mini-cart">  <isif condition="${checkoutstep != 5 && checkoutstep != 6}">  <isminilineitems p\_lineitemctnr="${lineCtnr}"/>  </isif> |

* Replace line <isif condition="${checkoutstep != 5}"> inside div checkout-mini-cartwith <isif condition= <isif condition="${checkoutstep != 5 && checkoutstep != 6}">

<isminilineitems p\_lineitemctnr="${lineCtnr}"/>

* Replace <isif condition="${checkoutstep > 3}"> with <isif condition="${checkoutstep == 6}">

|  |
| --- |
| <iscomment>render the order totals</iscomment>  <div class="checkout-order-totals">  <iscomment><isif condition="${checkoutstep == 6}">  <isordertotals p\_lineitemctnr="${lineCtnr}" p\_showshipmentinfo="${true}" p\_shipmenteditable="${false}" p\_totallabel="${Resource.msg('global.ordertotal','locale',null)}"/></iscomment>  <isif condition="${checkoutstep > 3}">  <isordertotals p\_lineitemctnr="${lineCtnr}" p\_showshipmentinfo="${true}" p\_shipmenteditable="${true}" p\_totallabel="${Resource.msg('global.ordertotal','locale',null)}"/>  <iselse/>  <isordertotals p\_lineitemctnr="${lineCtnr}" p\_showshipmentinfo="${false}" p\_shipmenteditable="${false}" p\_totallabel="${Resource.msg('global.estimatedtotal','locale',null)}"/>  </isif>  </div> |

##### Update “minishipments.isml”

* Replace “Shipments” variable set via <isset tag, with below code snippet in minishipments.isml. This code snippet determine basket or order whichever available taken as lineCtntr and it determine if the page is summary page or summary page with IFRAME/lightbox

|  |
| --- |
| <isif condition="${!empty(pdict.Basket)}">  <isset name="lineCtnr" value="${pdict.Basket}" scope="page"/>  <isset name="Shipments" value="${pdict.Basket.shipments}" scope="page"/>  <iselseif condition="${!empty(pdict.Order)}">  <isset name="lineCtnr" value="${pdict.Order}" scope="page"/>  <isset name="Shipments" value="${pdict.Order.shipments}" scope="page"/>  </isif> |

* Replace “pdict.Basket” with “lineCtnr” inside <isif condition="${!empty(Shipments)}"> condition i.e. <isif condition="${shipment.productLineItems.length <= 0 || shipment.custom.shipmentType == null && shipment.UUID==lineCtnr.defaultShipment.UUID && !empty(shipment.shippingAddress) && empty(shipment.shippingAddress.address1)}">

And “ <isif condition="${Shipments.size() > 1 && pdict.Basket.productLineItems.size() > 0}"><div class="name">${Resource.msgf('multishippingshipments.shipment','checkout',null, shipmentCount)}</div></isif> with “<isif condition="${Shipments.size() > 1 && lineCtnr.productLineItems.size() > 0}"><div class="name">${Resource.msgf('multishippingshipments.shipment','checkout',null, shipmentCount)}</div></isif>”

* Wrap edit link <a> tag with non-empty basket if condition everywhere in file

|  |
| --- |
| <h3 class="section-header">  <isif condition="${shipment.giftCertificateLineItems.size() > 0}">  ${Resource.msg('minishipments.shipping','checkout',null)} <span>${Resource.msg('minishipments.giftcertdelivery','checkout',null)}</span>  <iselseif condition="${shipment.custom.shipmentType == 'instore'}"/>  <isset name="editUrl" value="${URLUtils.https('Cart-Show')}" scope="page"/>  <isif condition="${!empty(pdict.Basket)}"> <a href="${editUrl}" class="section-header-note">${Resource.msg('global.edit','locale',null)}</a></isif>  ${Resource.msg('cart.store.instorepickup','checkout',null)}  <iselseif condition="${shipment.shippingAddress != null && lineCtnr.productLineItems.size() > 0}"/>  <isif condition="${!empty(pdict.Basket)}"> <a href="${editUrl}" class="section-header-note">${Resource.msg('global.edit','locale',null)}</a></isif>  ${Resource.msg('minishipments.shippingaddress','checkout',null)}  </isif>  </h3> |

##### Update “minibillinginfo.isml”

* Replace “billingAddress” and “paymentInstruments” variables set via <isset tag, with below code snippet in minibillinginfo.isml. This code snippet determine basket or order whichever available taken as lineCtntr and it determine if the page is summary page or summary page with IFRAME/lightbox

|  |
| --- |
| <isif condition="${!empty(pdict.Basket)}">  <isset name="lineCtnr" value="${pdict.Basket}" scope="page"/>  <isset name="billingAddress" value="${pdict.Basket.billingAddress}" scope="page"/>  <isset name="paymentInstruments" value="${pdict.Basket.paymentInstruments}" scope="page"/>  <iselseif condition="${!empty(pdict.Order)}">  <isset name="lineCtnr" value="${pdict.Order}" scope="page"/>  <isset name="billingAddress" value="${pdict.Order.billingAddress}" scope="page"/>  <isset name="paymentInstruments" value="${pdict.Order.paymentInstruments}" scope="page"/>  </isif> |

* Replace ”<a href="${URLUtils.https('COBilling-Start')}" class="section-header-note">${Resource.msg('global.edit','locale',null)}</a>” with if condition everywhere in file. <isif condition="${!empty(pdict.Basket)}"> <a href="${URLUtils.https('COBilling-Start')}" class="section-header-note">${Resource.msg('global.edit','locale',null)}</a></isif>
* Add a selectedPaymentMethodMiniSection as follows

<div class="selectedPaymentMethodMiniSection" data-paymentmethodid="${paymentInstr.paymentMethod}"><isprint value="${dw.order.PaymentMgr.getPaymentMethod(paymentInstr.paymentMethod).name}" /></div>

##### Update “PaymentMethods.isml”

* Add the below code snippet to paymentMethods.isml, after legend block for displaying the Error Messages in case of a failed transaction.

|  |
| --- |
| <isif condition="${pdict.errorMessage != null}">  <div class="error-form">${pdict.errorMessage}</div>  </isif> |

**\*\* Merchant shall take responsibility to add this section at the top of the Page of above the Payment Method selection section.**

* Replace all payment method specific code snippet like credit card block, Bill me later paypal etc till the closing of </fieldset> with the <div class="payment-method-container"></div> from paymentMethods.isml as we have already taken this section inside worldpay cartridge in file named “worldpaypaymentmethods.isml”

|  |
| --- |
| **<iscomment>**  **Credit card block**  **--------------------------------------------------------------**  **</iscomment>**  <iscomment>display select box with stored credit cards if customer is authenticated</iscomment>  <isif condition="${pdict.CurrentCustomer.authenticated && !empty(pdict.ApplicableCreditCards)}">  <div class="form-row">  <label class="label">${Resource.msg('billing.selectcreditcard','checkout',null)}</label>  <div class="field-wrapper">  <select name="${pdict.CurrentForms.billing.paymentMethods.creditCardList.htmlName}" id="creditCardList" class="input-select">  <option value="" selected="selected">${Resource.msg('billing.creditcardlistselect','checkout',null)}</option>  <isloop items="${pdict.ApplicableCreditCards}" var="creditCardInstr">  <option value="${creditCardInstr.UUID}">(<isprint value="${creditCardInstr.creditCardType}"/>) <isprint value="${creditCardInstr.maskedCreditCardNumber}"/> - ${Resource.msg('billing.creditcardlistexp','checkout',null)} <isprint value="${creditCardInstr.creditCardExpirationMonth}" formatter="00" />.<isprint value="${creditCardInstr.creditCardExpirationYear}" formatter="0000" /></option>  </isloop>  </select>  </div>  </div>  <div class="form-row form-row-button">  <button id="credit-card-select-go" name="${pdict.CurrentForms.billing.creditCardSelect.htmlName}" type="submit" value="Go" class="simple-submit">Select</button>  </div>  <iscomment>  <isloop items="${pdict.ApplicableCreditCards}" var="creditCardInstr">  <a href="${URLUtils.https('COBilling-UpdateCreditCardSelection', 'creditCardUUID', creditCardInstr.UUID)}">  (<isprint value="${creditCardInstr.creditCardType}"/>)  <isprint value="${creditCardInstr.maskedCreditCardNumber}"/>  - ${Resource.msg('billing.creditcardlistexp','checkout',null)}  <isprint value="${creditCardInstr.creditCardExpirationMonth}" formatter="00" />  .<isprint value="${creditCardInstr.creditCardExpirationYear}" formatter="0000" />  </a>  </isloop>  </iscomment>  </isif>  <isinputfield formfield="${pdict.CurrentForms.billing.paymentMethods.creditCard.owner}" type="input"/>  <isinputfield formfield="${pdict.CurrentForms.billing.paymentMethods.creditCard.type}" type="select"/>  <isinputfield formfield="${pdict.CurrentForms.billing.paymentMethods.creditCard.number}" type="input" dynamicname="true"/>  <div class="form-row required">  <label>  <span class="required-indicator">${Resource.msg('billing.requiredindicator','checkout',null)}</span>  <span>${Resource.msg('billing.creditcardlistexpdate', 'checkout', null)}</span>  </label>  <isscript>  var currentCountry = require('~/cartridge/scripts/util/Countries').getCurrent(pdict);  </isscript>  <isdynamicform formobject="${pdict.CurrentForms.billing.paymentMethods.creditCard.expiration}" formdata="${currentCountry.dynamicForms.expirationInfo}"/>  </div>  <isscript>  var help = {  label: Resource.msg('billing.linkcvn', 'checkout', null),  cid: 'checkout-security-code'  };  </isscript>  <isinputfield formfield="${pdict.CurrentForms.billing.paymentMethods.creditCard.cvn}" type="input" rowclass="cvn" dynamicname="true" help="${help}"/>  <isif condition="${pdict.CurrentCustomer.authenticated}">  <isinputfield formfield="${pdict.CurrentForms.billing.paymentMethods.creditCard.saveCard}" type="checkbox"/>  </isif>    </div>  <div class="payment-method <isif condition="${!empty(pdict.selectedPaymentID) && pdict.selectedPaymentID=='PayPal'}">payment-method-expanded</isif>" data-method="Custom">  <!-- Your custom payment method implementation goes here. -->  ${Resource.msg('billing.custompaymentmethod','checkout',null)}  </div> |

##### Update “creditcardjson.isml”

* Add below code snippet to creditcardjson.isml file

|  |
| --- |
| <isscript>  var cc = {  maskedNumber:pdict.SelectedCreditCard.maskedCreditCardNumber,  holder:pdict.SelectedCreditCard.creditCardHolder,  type:pdict.SelectedCreditCard.creditCardType,  expirationMonth:pdict.SelectedCreditCard.creditCardExpirationMonth,  expirationYear:pdict.SelectedCreditCard.creditCardExpirationYear,  showCvvField:dw.system.Site.getCurrent().getCustomPreferenceValue('WorldpayEnableTokenization') && !empty(pdict.SelectedCreditCard.creditCardToken) && dw.system.Site.getCurrent().getCustomPreferenceValue('WorldpayDisableCVV')  }  var json = JSON.stringify(cc);  </isscript> |

##### Update “ReportCheckout.isml”

Add below code snippet to ReportCheckout.isml file for tracking

* Replace “pdict.Basket” with “LineCntr” at two places and check for LineCntr for null

|  |
| --- |
| 'BasketID', LineCntr != null ? LineCntr.UUID:null, |

* Add below code snippet after <isset> variable declared in file ReportCheckout.isml. This will help in ensuring checkout tracking of new page displayed for IFRAME/lightbox in redirect flow

|  |
| --- |
| <isset name="checkoutstep" value="${pdict.checkoutstep}" scope="page"/>  <isset name="checkoutname" value="${pdict.checkoutname}" scope="page"/>  <isset name="LineCntr" value="${pdict.Basket}" scope="page"/>  <isif condition="${empty(pdict.Basket) && !empty(pdict.Order)}">  <isset name="LineCntr" value="${pdict.Order}" scope="page"/>  </isif> |

##### Update “style.css/\_checkout.scss”

* Update style.css or ideally in file \_checkout.scss in order to make lightbox center align on page add below css in file \_checkout.scss which gets compiled to style.css

|  |
| --- |
| /\* lightbox center align \*/  div#custom-html #wp-cl-lightbox {margin: 0 !important; left: 8.5%; top: 25%;}  div#wp-cl { text-align: center;}  /\*ipad\*/  @media screen and (min-width:768px) and (max-width: 959px)  {  div#custom-html #wp-cl-lightbox {left: 20%; top: 20%;position: fixed; width:60% !important;}  } |

#### APM specific changes

##### For Credit Card Redirect

###### Update “billing.xml”

* Add below code snippet in billing.xml Form inside form group “paymentMethods”

|  |
| --- |
| <group formid="paymentMethods">  <field formid="cards" type="string" mandatory="false" binding="cards"></field> |

##### For IDEAL Forms

###### Update “billing.xml”

* Add the below code snippet in billing.xml within the group with formid=”paymentMethods"

|  |
| --- |
| <group formid="paymentMethods">  <include formid="idealFields" name="ideal"/> |

##### For GiroPay Forms

###### Update “billing.xml”

* Add the below code snippet in billing.xml within the group with formid= “paymentMethods" (Screenshot attached for better understanding)

|  |
| --- |
| <group formid="paymentMethods">  <include formid="giropayFields" name="giropay"/> |

##### For SEPA DD Forms

###### Update “billing.xml”

* Add the below code snippet in billing.xml within the group with formid= “paymentMethods" (Screenshot attached for better understanding)

|  |
| --- |
| <group formid="paymentMethods">  <include formid="elvFields" name="elv"/> |



##### For Konbini

###### Update “orderconfirmation.isml”

* Add below code snippet before ThankYou section <table> tag starts in orderconfirmation.isml and after corresponding </table> tag add closing </isif> tag

|  |
| --- |
| <td colspan="2" style="font-size:12px;font-family:arial;padding:20px 10px;vertical-align:top;">  <isinclude template="worldpayconfirmation"/>  <table style="background:#ffffff;border:1px solid #999999;width:680px;">  <tr>  <th style="background:#cccccc;padding:5px 20px;font-size:12px;font-family:arial;text-align:left;">${Resource.msg('confirmation.thankyou','checkout',null)}</th>  </tr>  <tr> |

###### Update “confirmation.isml”

* Add below code section for Konbini payment method before thank you message

|  |
| --- |
| <div class="confirmation <isif condition="${!pdict.CurrentCustomer.authenticated}">create-account</isif>">  <isinclude template="worldpayconfirmation"/>  <div class="confirmation-message">  <h1>${Resource.msg('confirmation.thankyou','checkout',null)}</h1> |

#### Property file changes

##### Update “forms.properties”

* Update the country supported in the form.properties files.

|  |
| --- |
| country.ru=Russia  country.pl=Poland  country.eg=Egypt |

* Update the years in the form.properties files.

|  |
| --- |
| year.2037=2037  year.2036=2036  year.2035=2035  year.2034=2034  year.2033=2033  year.2032=2032  year.2031=2031  year.2030=2030  year.2029=2029  year.2028=2028  year.2027=2027  year.2026=2026  year.2025=2025  year.2024=2024  year.2023=2023  year.2022=2022 |

* Update the form.properties files for all the relevant countries.

##### Update “Resources.ds”

* Add below code under Resource helper URL section

|  |
| --- |
| rateLimiterReset : URLUtils.url('RateLimiter-HideCaptcha').toString(),  csrffailed : URLUtils.url('CSRF-Failed').toString(),  apmLookUp : URLUtils.https('Worldpay-APMLookupService').toString(),  loadPaymentMethods : URLUtils.https('Worldpay-LoadPaymentMethods').toString() |

* Replace CHECK\_TLS with the below code.

|  |
| --- |
| ResourceHelper.getPreferences = function(pageContext) {  var cookieHintAsset = ContentMgr.getContent('cookie\_hint');  var consentTrackingHintAsset = ContentMgr.getContent('consent\_tracking\_hint');  return {  LISTING\_INFINITE\_SCROLL: (Site.getCurrent().getCustomPreferenceValue('enableInfiniteScroll') ? true : false),  LISTING\_REFINE\_SORT: true,  STORE\_PICKUP: Site.getCurrent().getCustomPreferenceValue('enableStorePickUp'),  COOKIE\_HINT: (cookieHintAsset && cookieHintAsset.online) || false,  CONSENT\_TRACKING\_HINT: (consentTrackingHintAsset && consentTrackingHintAsset.online) || false,  CHECK\_TLS: Site.getCurrent().getCustomPreferenceValue('checkTLS'),  WP\_ENABLE\_CLIENT\_SIDE\_ENCRYPTION : (dw.system.Site.getCurrent().getCustomPreferenceValue('WorldpayEnableClientSideEncryption') ? true : false),  WP\_CSE\_PUBLIC\_KEY : dw.system.Site.getCurrent().getCustomPreferenceValue('WorldpayClientSideEncryptionPublicKey'),  WP\_DISABLE\_CVV : dw.system.Site.getCurrent().getCustomPreferenceValue('WorldpayDisableCVV')    }; |

* Update the ConsentTrackingHintAsset with the below code snippet

|  |
| --- |
| ResourceHelper.getPreferences = function(pageContext) {  var cookieHintAsset = ContentMgr.getContent('cookie\_hint');  var consentTrackingHintAsset = ContentMgr.getContent('consent\_tracking\_hint'); |

### APM Lookup Support

#### billing.js (compiled into app.js)

* Replace existing **function “setCCFields”** with below code snippet.

|  |
| --- |
| function setCCFields(data) {  var $selectPaymentMethod = $('.payment-method-options');  var selectedPaymentMethod = $selectPaymentMethod.find(':checked').val();  var $creditCard = $('[data-method="'+selectedPaymentMethod+'"]');  $creditCard.find('input[name$="creditCard\_owner"]').val(data.holder).trigger('change');  $creditCard.find('select[name$="\_type"]').val(data.type).trigger('change');  $creditCard.find('input[name\*="\_creditCard\_number"]').val(data.maskedNumber).trigger('change');  $creditCard.find('[name$="\_month"]').val(data.expirationMonth).trigger('change');  $creditCard.find('[name$="\_year"]').val(data.expirationYear).trigger('change');  if(selectedPaymentMethod && selectedPaymentMethod == 'Worldpay'){  // hide CVV  if (window.SitePreferences.WP\_DISABLE\_CVV) {  $creditCard.find('input[name\*="\_cvn"]').removeClass('required');  $creditCard.find('div.cvn').hide();  } else {  $creditCard.find('input[name\*="\_cvn"]').addClass('required');  $creditCard.find('div.cvn').show();  }  $creditCard.find('input[name$="creditCard\_owner"]').prop('readonly', true);  $creditCard.find('select[name$="\_type"]').attr("disabled", true);  $creditCard.find('input[name\*="\_creditCard\_number"]').prop('readonly', true);  $creditCard.find('[name$="\_month"]').attr("disabled", true);  $creditCard.find('[name$="\_year"]').attr("disabled", true);  $creditCard.find('input[name$="\_saveCard"]').val('false');  $creditCard.find('input[name$="\_saveCard"]').removeProp('checked');  if ($('select[id$="\_country"]').val() == "BR") {  $("#Payment\_Brazil").show();  }  }else{ // show CVV  $creditCard.find('input[name\*="\_cvn"]').addClass('required');  $creditCard.find('input[name$="\_cvn"]').val('').trigger('change');  }  } |

* Replace existing function “**updatePaymentMethod**” with below code snippet

|  |
| --- |
| function updatePaymentMethod(paymentMethodID) {  $.ajax({  url:Urls.loadPaymentMethods,  type:"Post",  data:{"selectedPaymentMethodId":paymentMethodID},  success:function(response){  $(".payment-method-container").html(response);  if($('.payment-method-options').find(':checked').val()=='Worldpay' && $('select[id$="\_country"]', $form).val() == "BR"){  $("#Payment\_Brazil").hide();  }  // select credit card from list  $('.creditCardList').on('change', function () {  var cardUUID = $(this).val();  if (!cardUUID) {  $("#clearpaymentform").trigger('click');  return;  }  // remove server side error  $('.required.error').removeClass('error');  $('.error-message').remove();  var selectedPaymentMethod = $('.payment-method-options').find(':checked').val();  if(selectedPaymentMethod=='Worldpay'){  if(cardUUID!=""){  $(".card-block").show();  //$(".save-card-block").hide();  }else{  $(".card-block").hide();  $(".save-card-block").show();  }  }  populateCreditCardForm(cardUUID);  var $creditCardSection = $('[data-method="Worldpay"]');  $creditCardSection.find('[name$="\_cards"]').val('');  var $form = $('.address');  var countryCode = $('select[id$="\_country"]', $form).val();  if(paymentMethodID == "Worldpay" && countryCode =="BR"){  $("#Payment\_Brazil").show();  $creditCardSection.find('input[name$="creditCard\_cpf"]').removeClass('required');  $creditCardSection.find('input[name$="creditCard\_installments"]').removeClass('required');  } else if(paymentMethodID == "CREDIT\_CARD" && countryCode =="BR"){  var $creditCardSection = $('[data-method="CREDIT\_CARD"]');  $creditCardSection.find('input[name$="creditCard\_cpf"]').addClass('required');  $creditCardSection.find('input[name$="creditCard\_installments"]').removeClass('required');  }  });  if(paymentMethodID=='Worldpay'){  var $creditCard = $('[data-method="Worldpay"]');  $creditCard.find('input,select').val('');  $creditCard.find('input[name$="creditCard\_owner"]').removeClass('required');  $creditCard.find('select[name$="\_type"]').removeClass('required');  $creditCard.find('input[name\*="\_creditCard\_number"]').removeClass('required');  $creditCard.find('[name$="\_month"]').removeClass('required');  $creditCard.find('[name$="\_year"]').removeClass('required');  $creditCard.find('input[name$="\_cvn"]').removeClass('required');  $('.creditCardList').val('');  $creditCard.find('input[name$="\_saveCard"]').val('false');  $creditCard.find('input[name$="\_saveCard"]').removeProp('checked');  $(".card-block").hide();  $(".save-card-block").show();  }  var $form = $('.address');  var countryCode = $('select[id$="\_country"]', $form).val();  fieldForBrazil(paymentMethodID,countryCode);    // select prefered credit card from list  $('#preferedCreditCardList').on('change', function (e) {  e.preventDefault();  var $creditCard = $('[data-method="Worldpay"]');  $creditCard.find('input[name$="creditCard\_owner"]').removeClass('required');  $creditCard.find('input[name$="creditCard\_owner"]').val('');  $creditCard.find('select[name$="\_type"]').removeClass('required');  $creditCard.find('input[name\*="\_creditCard\_number"]').removeClass('required');  $creditCard.find('input[name\*="\_creditCard\_number"]').val('');  $creditCard.find('[name$="\_month"]').removeClass('required');  $creditCard.find('[name$="\_year"]').removeClass('required');  $creditCard.find('input[name$="\_cvn"]').removeClass('required');  $('.creditCardList').val('');  $creditCard.find('input[name$="\_saveCard"]').val('true');  $creditCard.find('input[name$="\_saveCard"]').prop('checked',true);  $(".card-block").hide();  $(".save-card-block").show();  $("#Payment\_Brazil").hide();  });  $("#clearpaymentform").on('click',function(e){  e.preventDefault();  var $creditCard = $('#PaymentMethod\_Worldpay');  if ($('input#is-Worldpay').prop('checked')) {  $creditCard.find('input,select').val('');  $creditCard.find('input[name$="creditCard\_owner"]').removeClass('required');  $creditCard.find('select[name$="\_type"]').removeClass('required');  $creditCard.find('input[name\*="\_creditCard\_number"]').removeClass('required');  $creditCard.find('[name$="\_month"]').removeClass('required');  $creditCard.find('[name$="\_year"]').removeClass('required');  $creditCard.find('input[name$="\_cvn"]').removeClass('required');  $('.creditCardList').val('');  $creditCard.find('input[name$="\_saveCard"]').val('true');  $creditCard.find('input[name$="\_saveCard"]').prop('checked',true);  $(".card-block").hide();  $(".save-card-block").show();  $("#Payment\_Brazil").hide();  } else {  $creditCard = $('[data-method="CREDIT\_CARD"]');  $creditCard.find('input,select').val('');  $creditCard.find('input[name$="creditCard\_owner"]').addClass('required');  $creditCard.find('select[name$="\_type"]').addClass('required');  $creditCard.find('input[name\*="\_creditCard\_number"]').addClass('required');  $creditCard.find('[name$="\_month"]').addClass('required');  $creditCard.find('[name$="\_year"]').addClass('required');  $creditCard.find('input[name$="\_cvn"]').addClass('required');  $('.creditCardList').val('');  $creditCard.find('input[name$="\_saveCard"]').val('true');  $creditCard.find('input[name$="\_saveCard"]').prop('checked',true);  $creditCard.find('div.cvn').show();  $(".card-block").show();  $(".save-card-block").show();  if (countryCode === 'BR') {  $("#Payment\_Brazil").show();  }  }  });  }  });  $('input[name$="\_selectedPaymentMethodID"]').removeAttr('checked');  $('input[value=' + paymentMethodID + ']').prop('checked', 'checked');  //formPrepare.validateForm();  } |

* Add the below function **populateApmMethods** after function updatePaymentMethod.

|  |
| --- |
| function populateApmMethods() {  // load APM details  var countryCode = $('select[id$="\_country"]').val();  if(countryCode == undefined || countryCode ==''){  return; }  var url = Urls.apmLookUp + '?billingCountry=' + countryCode;    $.ajax({  url: url,  method: "POST",  dataType:"html"  }).done(function(data){  var result = new Array();  var foundSelectedPaymentMethod = false;  var selectedPaymentMethod = 'CREDIT\_CARD';  var paymentMethods= $(".payment-method-options");  $(paymentMethods).replaceWith(data);  if ($('.selectedPaymentMethodMiniSection').length>0) {  var datapaymentmethodid = $('.selectedPaymentMethodMiniSection').attr('data-paymentmethodid');  if ($('.payment-method-options input[value='+datapaymentmethodid+']')) {  foundSelectedPaymentMethod = true;  $('.payment-method-options input[value='+datapaymentmethodid+']').prop('checked', 'checked');  selectedPaymentMethod = datapaymentmethodid;  }  }  if (!foundSelectedPaymentMethod) {  $('input[value=CREDIT\_CARD]').prop('checked', 'checked');  }  updatePaymentMethod(selectedPaymentMethod);  });  } |

* **Remove below function creditCardList'->onChange** below inside the exports.init

$('#creditCardList').on('change', function ()

|  |
| --- |
| // select credit card from list  $('#creditCardList').on('change', function () {  var cardUUID = $(this).val();  if (!cardUUID) {return;}  populateCreditCardForm(cardUUID);  // remove server side error  $('.required.error').removeClass('error');  $('.error-message').remove(); |

* **Add new function** **checkoutForm->onsubmit** after removal of above function. This handles the client Side Encryption.

|  |
| --- |
| $checkoutForm.on('submit',function(e){  if($('.payment-method-options').find(':checked').val()=='CREDIT\_CARD'){  var $creditCard = $('[data-method="CREDIT\_CARD"]');  $creditCard.find('[name$="\_encryptedData"]').val('');  var data = {  cvc: $creditCard.find('input[name\*="\_cvn"]').val(),  cardHolderName: $creditCard.find('input[name$="creditCard\_owner"]').val(),  cardNumber: $creditCard.find('input[name\*="\_creditCard\_number"]').val(),  expiryMonth: $creditCard.find('[name$="\_month"]').val() < 10 ? '0'+$creditCard.find('[name$="\_month"]').val():$creditCard.find('[name$="\_month"]').val(),  expiryYear: $creditCard.find('[name$="\_year"]').val(),  }  Worldpay.setPublicKey(window.SitePreferences.WP\_CSE\_PUBLIC\_KEY);  var encryptedData = Worldpay.encrypt(data, function(e){  console.log("Worldpay Client Side Encryption validation error "+e);  });  if (encryptedData) {  $creditCard.find('[name$="\_encryptedData"]').val(encryptedData);  }  } else if($('.payment-method-options').find(':checked').val()=='Worldpay'){  var $creditCard = $('[data-method="Worldpay"]');  $creditCard.find('select[name$="\_type"]').removeAttr("disabled");  $creditCard.find('[name$="\_month"]').removeAttr("disabled");  $creditCard.find('[name$="\_year"]').removeAttr("disabled");  }  }); |

* Replace the code line “var selectedPaymentMethod = $selectPaymentMethod.find(':checked').val(); “ with “var selectedPaymentMethod = $('.payment-method-options').find(':checked').val();” in exports init function.
* **Make a call to new function** **populateApmMethods** **on country change** inside function exports.init and below formPrepare.init({

|  |
| --- |
| var selectedPaymentMethod = $('.payment-method-options').find(':checked').val();  formPrepare.init({  formSelector: 'form[id$="billing"]',  continueSelector: '[name$="billing\_save"]'  });  var $countrySelector= $('select[name$="\_country"]');  $countrySelector.on('change',function(e){  populateApmMethods();  selectedPaymentMethod = $('.payment-method-options').find(':checked').val();  var $form = $('.address');  var countryCode = $('select[id$="\_country"]', $form).val();  fieldForBrazil((selectedPaymentMethod) ? selectedPaymentMethod : 'CREDIT\_CARD',countryCode);  //updatePaymentMethod((selectedPaymentMethod) ? selectedPaymentMethod : 'CREDIT\_CARD');  });  populateApmMethods();  selectedPaymentMethod = $('.payment-method-options').find(':checked').val();    // default payment method to 'CREDIT\_CARD'  //updatePaymentMethod((selectedPaymentMethod) ? selectedPaymentMethod : 'CREDIT\_CARD');  $checkoutForm.on('click', 'input[type="radio"][name$="\_selectedPaymentMethodID"]', function(e){  updatePaymentMethod($(this).val());  }); |

### Brazil Integration

#### Update “creditcard.xml “

* Add below code snippet in **creditcard.xml** inside the form tag

|  |
| --- |
| <!-- field for CPF incase of Brazil -->  <field formid="cpf" label="label.cpf" type="string" max-length="25" binding="CPFNumber" missing-error="creditcard.numbermissingerror" value-error="creditcard.numbervalueerror"/>    <!-- field for Installments incase of brazil -->  <field formid="installments" label="label.installments" type="string" max-length="2" binding="Installments" missing-error="creditcard.numbermissingerror" value-error="creditcard.numbervalueerror"/>  <field formid="cards" type="string" mandatory="false" binding="cards"></field> |

\*\*The merchant need to ensure and provide validation and the max limit of the installments they want to support.

#### Update “billing.js” (compiled into app.js)

Add below code snippet to billing.js for Brazil integration:

* Add below function in billing.js after populateCreditCardForm function and call this method on change of country code:

|  |
| --- |
| function fieldForBrazil(paymentMethodID,countryCode){  if(countryCode =="BR"){  if(paymentMethodID == "BOLETO-SSL"){  $("#Payment\_Brazil").show();  var $fieldsSection = $("#Payment\_Brazil");  $fieldsSection.find('input[name$="creditCard\_cpf"]').addClass('required');  $fieldsSection.find('input[name$="creditCard\_cpf"]').parents('.form-row').find('label').prepend('<span class="required-indicator">â€¢ </span>');  } else if(paymentMethodID == "CREDIT\_CARD"){  $("#Payment\_Brazil").show();  var $fieldsSection = $("#Payment\_Brazil");  $fieldsSection.find('input[name$="creditCard\_cpf"]').addClass('required');  $fieldsSection.find('input[name$="creditCard\_cpf"]').parents('.form-row').find('label').prepend('<span class="required-indicator">â€¢ </span>');  $fieldsSection.find('input[name$="creditCard\_installments"]').removeClass('required');  } else if(paymentMethodID == "Worldpay"){  var $fieldsSection = $("#Payment\_Brazil");  $fieldsSection.find('input[name$="creditCard\_cpf"]').removeClass('required');  $fieldsSection.find('input[name$="creditCard\_installments"]').removeClass('required');  } else {  $("#Payment\_Brazil").hide();  }  }else{  $("#Payment\_Brazil").hide();  }  } |

### Street Number Implementation

In order to supply the street number (in Worldpay documentation also referred as house number) for the redirect payment model, it is required to implement an appropriate parsing algorithm in the CreateRequestHelper.ds script file. Line number 83 and 145 contains the appropriate attribute that needs to be configured. The street number can appear for example in different configurations:

In front of the street name (USA, Canada)

After the street name (Europe)

As an extra field

Therefore, the appropriate parsing logic has to be implemented during installation and according to the used scenario. If this step is not taken, the customer will have to resupply the street number at the Worldpay payment page. For more information, please refer to the WorldPay documentation.

### Error Messaging

In cartridge all the error messaging is achieved using properties file (worldpayerror.properties). These messages can be configured while installation of the cartridge and can be modified based upon the merchants business requirement. Using properties file for messaging gives the Merchant the advantage for supporting multi lingual site.

### Client Side Encryption

#### Update “pt\_checkout.isml”

Add below code snippet to pt\_checkout.isml for Client Side Encryption before </head> tag closing.

|  |
| --- |
| <iscomment>Add template-specific header information here.</iscomment>    <isif condition="${dw.system.Site.getCurrent().getCustomPreferenceValue('WorldpayEnableClientSideEncryption')}">  <script src="https://payments.worldpay.com/resources/cse/js/worldpay-cse-1.0.1.min.js"></script>  </isif> |

#### Update “creditcard.xml”

* Add the below code snippet in creditcard.xml inside form tag

|  |
| --- |
| <!-- field for credit card encrypted data -->  <field formid="encryptedData" type="string" mandatory="false" /> |

Make CVN field as non mandatory

|  |
| --- |
| <!-- field for credit card security code -->  <field formid="cvn" label="creditcard.cvnlabel" type="string" mandatory="true" masked="0"  missing-error="creditcard.cvnmissingerror" value-error="creditcard.cvnrangeerror"/> |

* Add the below snippet in creditcard.xml

|  |
| --- |
| <field formid="cards" type="string" mandatory="false" binding="cards"></field> |

* Add the below snippet to increase the size of the card

|  |
| --- |
| <!-- field for credit card number -->  <field formid="number" label="creditcard.number" type="string" mandatory="true" masked="4" max-length="23"  description="creditcard.numberexample" binding="creditCardNumber"  missing-error="creditcard.numbermissingerror" value-error="creditcard.numbervalueerror"/> |

* Add the below snippet to increase the expiration years.

|  |
| --- |
| <field formid="year" label="resource.year" type="integer" mandatory="true" binding="creditCardExpirationYear"  missing-error="creditcard.yearmissingerror">  <options>  <option optionid="2018" label="year.2018" value="2018"/>  <option optionid="2019" label="year.2019" value="2019"/>  <option optionid="2020" label="year.2020" value="2020"/>  <option optionid="2021" label="year.2021" value="2021"/>  <option optionid="2022" label="year.2022" value="2022"/>  <option optionid="2023" label="year.2023" value="2023"/>  <option optionid="2024" label="year.2024" value="2024"/>  <option optionid="2025" label="year.2025" value="2025"/>  <option optionid="2026" label="year.2026" value="2026"/>  <option optionid="2027" label="year.2027" value="2027"/>  <option optionid="2028" label="year.2028" value="2028"/>  <option optionid="2029" label="year.2029" value="2029"/>  <option optionid="2030" label="year.2030" value="2030"/>  <option optionid="2031" label="year.2031" value="2031"/>  <option optionid="2032" label="year.2032" value="2032"/>  <option optionid="2033" label="year.2033" value="2033"/>  <option optionid="2034" label="year.2034" value="2034"/>  <option optionid="2035" label="year.2035" value="2035"/>  <option optionid="2036" label="year.2036" value="2036"/>  <option optionid="2037" label="year.2037" value="2037"/>  </options> |

* Update the credit.xml files for all the respected countries.

#### Update “Resource.ds”

* Add below code under Resource helper preference section

|  |
| --- |
| CHECK\_TLS: Site.getCurrent().getCustomPreferenceValue('checkTLS'),  WP\_ENABLE\_CLIENT\_SIDE\_ENCRYPTION : (dw.system.Site.getCurrent().getCustomPreferenceValue('WorldpayEnableClientSideEncryption') ? true : false),  WP\_CSE\_PUBLIC\_KEY : dw.system.Site.getCurrent().getCustomPreferenceValue('WorldpayClientSideEncryptionPublicKey'),  WP\_DISABLE\_CVV : dw.system.Site.getCurrent().getCustomPreferenceValue('WorldpayDisableCVV') |

### Form File Changes

Disclaimer: shipping and billing address in forms must have country code in UPPERCASE as WorldPay mandatory requirement.

Purpose: WorldPay XML request must have contry code in UPPERCASE.

### Update shippingaddress.xml and billingaddress.xml

* Update the shipping and billing address with the relevant countries.
* All the countries files should also be updated for all the relevant countries.

### Update the paymentinstrumentdetails.isml

* Update the maximum length and size.

|  |
| --- |
| <isscript>  var ownerAttributes = {  maxlength: 40,  size: 40  };  var numberAttributes = {  maxlength: 23,  size: 23  };  </isscript> |

### Update the locale.properties and countries.json

* In locale.properties files update the submit order for the Klarna.
* Add the respective countries configuration in the countries.json file.

Note: There can be some existing app.js error coming from default siteGenesis. You need to check in console for app.js. That might affect credit card form.

### Stored Credentials (Not available in Controller)

A stored credential is Visa or Mastercard information (such as a card number) stored by a merchant or its agent, a [Payment Facilitator](http://support.worldpay.com/support/kb/gg/corporate-gateway-guide/content/industryschemeextras/paymentfacilitatorrules.htm) (PF) or a [Staged Digital Wallet Operator](http://support.worldpay.com/support/kb/gg/corporate-gateway-guide/content/staged-digital-wallet-operator/staged-digital-wallet-operator.htm) (SDWO) **to process future transactions**.

## What to do when storing credentials

**Include the storedCredentials element**

To indicate that you're storing payment details, include <storedCredentials> with the usage value of "FIRST" in your authorisation request:

...

  <storedCredentials usage="FIRST"/>

</paymentDetails>

**Store the transactionIdentifier from the response**

In the response, you get a <transactionIdentifier> (a child of <schemeResponse>), as shown below. This must be stored and (where applicable) supplied with subsequent use of these card details.

**Note:**The <transactionIdentifier> may contain up to 56 characters, and may include spaces, even at the beginning or end. Preserve all of these characters.

...

  <schemeResponse>

    <transactionIdentifier>1234567890</transactionIdentifier>

  </schemeResponse>

</payment>

Note: There are two ways to use stored credentials, depending on whether the **cardholder** or **merchant** initiates a transaction.

### Cardholder initiates

If the cardholder initiates a transaction with their stored details, you must include <storedCredentials> with the usage value of "USED":

...

  <storedCredentials usage="USED">

  </storedCredentials>

</paymentDetails>

### Merchant initiates

If the merchant initiates a transaction with the stored details (such as with a recurring payment), **you must include**:

* <storedCredentials> with:
  + the usage value of "USED", and
  + the appropriate merchantInitiatedReason
* <schemeTransactionIdentifier>, which is the <transactionIdentifier> from either:

Note: The Merchant initiated reasons can be INSTALMENT, RECURRING, UNSCHEDULED

To change the type of reason we have to change it from the code base in the following files:

1. The types are mentioned in the properties file, named worldpay.properties

worldpay.storedcred.mi.INSTALMENT= INSTALMENT

worldpay.storedcred.mi.RECURRING= RECURRING

worldpay.storedcred.mi.UNSCHEDULED = UNSCHEDULED

1. To change its type we have to change in file named LibCreateRequest.js

Find this code:

**if** (!(orderObj.createdBy.equals(WorldpayConstants.CUSTOMERORDER)) && session.isUserAuthenticated()) { //eslint-disable-line

storedCredentials = **new** XML('<storedCredentials usage="USED" merchantInitiatedReason="' + Resource.msg('worldpay.storedcred.mi.UNSCHEDULED', 'worldpay', **null**) + '"></storedCredentials>'); //eslint-disable-line

}

In this part of code change inside Resource.msg.

For eg, for installment + Resource.msg('worldpay.storedcred.mi. INSTALMENT, 'worldpay', **null**)

And so on,

* + The "FIRST" response message
  + The previous "USED" response message

...

  <storedCredentials usage="USED" merchantInitiatedReason="UNSCHEDULED">

    <schemeTransactionIdentifier>1234567890</schemeTransactionIdentifier> <!-- <transactionIdentifier> response from FIRST or USED request-->

  </storedCredentials>

</paymentDetails>

## Example Direct 'first' request

Select

<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE paymentService PUBLIC "-//WorldPay//DTD WorldPay PaymentService v1//EN"

"http://dtd.worldpay.com/paymentService\_v1.dtd">

<paymentService version="1.4" merchantCode="YOUR\_MERCHANT\_CODE">

  <submit>

    <order orderCode="YOUR\_ORDER\_CODE">

      <description>YOUR DESCRIPTION</description>

      <amount currencyCode="GBP" exponent="2" value="5000"/>

      <paymentDetails>

        <CARD-SSL>

          <cardNumber>4444333322221111</cardNumber>

          <expiryDate>

            <date month="01" year="2020"/>

          </expiryDate>

          <cardHolderName>A Shopper</cardHolderName>

          <cardAddress>

            <address>

              <address1>47A</address1>

              <address2>Queensbridge Road</address2>

              <address3>Suburbia</address3>

              <postalCode>CB94BQ</postalCode>

              <city>Cambridge</city>

              <state>Cambridgeshire</state>

              <countryCode>GB</countryCode>

            </address>

          </cardAddress>

        </CARD-SSL>

        <storedCredentials usage="FIRST"/> <!--Indicates that you are storing credentials-->

        <session shopperIPAddress="123.123.123.123" id="0215ui8ib1"/>

      </paymentDetails>

      <shopper>

        <shopperEmailAddress>ashopper@myprovider.com</shopperEmailAddress>

        <browser>

          <acceptHeader>text/html</acceptHeader>

          <userAgentHeader>Mozilla/5.0 ...</userAgentHeader>

        </browser>

      </shopper>

    </order>

  </submit>

</paymentService>

### Response to 'first' request

Your response includes the <schemeResponse> parameter, within which you get the <transactionIdentifier>. Where required, use the value returned in <transactionIdentifier> as the <schemeTransactionIdentifier> in your request to use credentials:

<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE paymentService PUBLIC "-//WorldPay//DTD WorldPay PaymentService v1//EN"

  "http://dtd.worldpay.com/paymentService\_v1.dtd">

<paymentService version="1.4" merchantCode="ExampleCode1">

  <reply>

    <orderStatus orderCode="ExampleOrder1">

      <payment>

        <paymentMethod>VISA-SSL</paymentMethod>

        <amount value="5000" currencyCode="GBP" exponent="2" debitCreditIndicator="credit"/>

        <lastEvent>AUTHORISED</lastEvent>

        <AuthorisationId id="666"/>

        <balance accountType="IN\_PROCESS\_AUTHORISED">

          <amount value="5000" currencyCode="GBP" exponent="2" debitCreditIndicator="credit"/>

        </balance>

        <cardNumber>4444\*\*\*\*\*\*\*\*1111</cardNumber>

        <riskScore value="0"/>

        <schemeResponse>

          <transactionIdentifier>20005060720116005061</transactionIdentifier> <!--Store this for later use-->

        </schemeResponse>

      </payment>

    </orderStatus>

  </reply>

</paymentService>

## Example Direct 'used' request

Select

<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE paymentService PUBLIC "-//WorldPay//DTD WorldPay PaymentService v1//EN"

"http://dtd.worldpay.com/paymentService\_v1.dtd">

<paymentService version="1.4" merchantCode="YOUR\_MERCHANT\_CODE">

  <submit>

    <order orderCode="YOUR\_ORDER\_CODE">

      <description>YOUR DESCRIPTION</description>

      <amount currencyCode="GBP" exponent="2" value="5000"/>

      <paymentDetails>

        <CARD-SSL>

          <cardNumber>4444333322221111</cardNumber>

          <expiryDate>

            <date month="01" year="2020"/>

          </expiryDate>

          <cardHolderName>A Shopper</cardHolderName>

          <cardAddress>

            <address>

              <address1>47A</address1>

              <address2>Queensbridge Road</address2>

              <address3>Suburbia</address3>

              <postalCode>CB94BQ</postalCode>

              <city>Cambridge</city>

              <state>Cambridgeshire</state>

              <countryCode>GB</countryCode>

            </address>

          </cardAddress>

        </CARD-SSL>

        <storedCredentials usage="USED" merchantInitiatedReason="UNSCHEDULED">

          <schemeTransactionIdentifier>20005060720116005061</schemeTransactionIdentifier> <!-- Only include this if it's a merchant-initiated transaction-->

        </storedCredentials>

        <session shopperIPAddress="123.123.123.123" id="0215ui8ib1"/>

      </paymentDetails>

      <shopper>

        <shopperEmailAddress>ashopper@myprovider.com</shopperEmailAddress>

        <browser>

          <acceptHeader>text/html</acceptHeader>

          <userAgentHeader>Mozilla/5.0 ...</userAgentHeader>

        </browser>

      </shopper>

    </order>

  </submit>

</paymentService>

### Response to 'used' request

In transactions, you receive another <transactionIdentifier>. We are using the first time received transactionIdentifier value.

<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE paymentService PUBLIC "-//WorldPay//DTD WorldPay PaymentService v1//EN"

"http://dtd.worldpay.com/paymentService\_v1.dtd">

<paymentService version="1.4" merchantCode="MYMERCHANT">

  <reply>

    <orderStatus orderCode="ExampleOrder1"> <!--The orderCode you supplied in the order-->

      <payment>

        <paymentMethod>VISA-SSL</paymentMethod>

        <amount value="5000" currencyCode="GBP" exponent="2" debitCreditIndicator="credit"/>

        <lastEvent>AUTHORISED</lastEvent>

        <AuthorisationId id="666"/>

        <balance accountType="IN\_PROCESS\_AUTHORISED">

          <amount value="5000" currencyCode="GBP" exponent="2" debitCreditIndicator="credit"/>

        </balance>

        <cardNumber>4444\*\*\*\*\*\*\*\*1111</cardNumber>

        <riskScore value="0"/>

        <schemeResponse>

          <transactionIdentifier>94443862893748229682</transactionIdentifier>

        </schemeResponse>

      </payment>

    </orderStatus>

  </reply>

</paymentService>

## EXTERNAL INTERFACES

Current version of cartridge support integration with APM’s [as per the list shared in above sections], however below URL can be referred to know more about the other possible APM’ supported by Worldpay. <http://support.worldpay.com/support/kb/gg/alternativepayments/content/apmguide.htm>

### CHINA UNION PAY

China UnionPay (also known as UnionPay or CUP) is a payment method where shoppers with UnionPay cards can make online e-commerce payments. UnionPay issue UnionPay cards in China.

|  |  |
| --- | --- |
| **Property** | **Description/Value** |
| **Payment Type** | Debit / Credit Card payment |
| **Countries** | China |
| **Acceptance Currencies** | AED, BND, BHD, CNY, GBP, EUR, USD, AUD, CAD, CHF, GEL, HKD, IDR, INR, JPY, KRW, KZT, LAK, MOP, MYR, OMR, NPR, NZD, PHP, PKR, RUB, SGD, THB, TRY, TWD, UAH, VND. |
| **Mode** | REDIRECT |

#### Request Attribute Mapping

<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE paymentService PUBLIC "-//Worldpay/DTD Worldpay PaymentService v1//EN"

"http://dtd.worldpay.com/paymentService\_v1.dtd">

<paymentService version="1.4" merchantCode="SAPIENTNITROECOM">

<submit>

<order orderCode="Example\_UnionPay\_Order\_Code">

<description>Test order</description>

<amount value="100" currencyCode="CNY" exponent="2"/>

<orderContent>

<![CDATA[ ]]>

</orderContent>

<paymentMethodMask>

<include code="ALL"/>

</paymentMethodMask>

<shopper>

<shopperEmailAddress>shopper@worldpay.com</shopperEmailAddress>

</shopper>

<statementNarrative>Merchants Statement Narrative</statementNarrative>

</order>

</submit>

</paymentService>

#### Response Attribute Mapping

<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE paymentService PUBLIC "-//Worldpay//DTD Worldpay PaymentService v1//EN"

"http://dtd.worldpay.com/paymentService\_v1.dtd">

<paymentService version="1.4" merchantCode="SAPIENTNITROECOM">

<reply>

<orderStatus orderCode="Example\_UnionPay\_Order\_Code">

<reference id="3015836999">https://secure-test.worldpay.com/jsp/shopper/SelectPaymentMethod.jsp?OrderKey=SAPIENTNITROECOM%5EExample\_UnionPay\_Order\_Code</reference>

</orderStatus>

</reply>

</paymentService>

### SOFORT

SOFORT Banking (SOFORT Überweisung) is a real-time bank transfer payment method that shoppers based in Austria, Belgium, France, Germany and Switzerland can use to transfer funds directly to merchants from their bank accounts.

|  |  |
| --- | --- |
| **Property** | **Description/Value** |
| **Payment Type** | Debit / Credit Card payment |
| **Countries** | Austria, Belgium , Germany and Switzerland |
| **Acceptance Currencies** | EUR (Austria, Belgium, Germany)  CHF (Switzerland) |
| **Mode** | DIRECT |

#### Request Attribute Mapping

<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE paymentService PUBLIC '-//Worldpay//DTD Worldpay PaymentService v1//EN' 'http://dtd.worldpay.com/paymentService\_v1.dtd'>

<paymentService merchantCode="SAPIENTNITROECOM" version="1.4">

<submit>

<order orderCode="Example\_SOFORT\_Banking\_Order\_Code\_0">

<description>Description here</description>

<amount currencyCode="EUR" value="100" exponent="2" />

<orderContent>

<![CDATA[ ]]>

</orderContent>

<paymentDetails>

<SOFORT-SSL shopperCountryCode="DE">

<successURL>http://www.worldpay.com/?successURL</successURL>

<cancelURL>http://www.worldpay.com/?cancelURL</cancelURL>

<pendingURL>http://www.worldpay.com/?pendingURL</pendingURL>

</SOFORT-SSL>

</paymentDetails>

<shopper>

<shopperEmailAddress>shopper@worldpay.com</shopperEmailAddress>

</shopper>

<statementNarrative>SN12345</statementNarrative>

</order>

</submit>

</paymentService>

#### Response Attribute Mapping

<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE paymentService PUBLIC "-//Worldpay//DTD Worldpay PaymentService v1//EN"

"http://dtd.worldpay.com/paymentService\_v1.dtd">

<paymentService version="1.4" merchantCode="SAPIENTNITROECOM">

<reply>

<orderStatus orderCode="Example\_SOFORT\_Banking\_Order\_Code\_0">

<reference id="3015838844">https://secure-test.worldpay.com/jsp/test/shopper/APMSTSimulator.jsp?customerRef=3013339841&amp;paymentMethod=SOFORT</reference>

</orderStatus>

</reply>

</paymentService>

### ALIPAY

Alipay is a popular payment method that shoppers based in China use to make payments online. After logging into their Alipay account, shoppers use one of the options below to complete their payment:

* eWallet
* Debit card
* Credit Card
* Bank Transfer
* Payment URL

|  |  |
| --- | --- |
| **Property** | **Description/Value** |
| **Payment Type** | eWallet |
| **Countries** | China |
| **Acceptance Currencies** | CNY, GBP, EUR, USD, AUD, CAD, HKD, SGD. |
| **Mode** | DIRECT |

#### Request Attribute Mapping

<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE paymentService PUBLIC '-//Bibit//DTD Bibit PaymentService v1//EN' 'http://dtd.worldpay.com/paymentService\_v1.dtd'>

<paymentService merchantCode="SAPIENTNITROECOM" version="1.4">

<submit>

<order orderCode="Example\_Alipay\_Order\_Code\_0">

<description>Description here</description>

<amount currencyCode="CNY" value="1000" exponent="2" />

<orderContent>

<![CDATA[ ]]>

</orderContent>

<paymentDetails>

<ALIPAY-SSL shopperCountryCode="CN">

<successURL>http://www.worldpay.com/?successURL</successURL>

<cancelURL>http://www.worldpay.com/?cancelURL</cancelURL>

<pendingURL>http://www.worldpay.com/?pendingURL</pendingURL>

</ALIPAY-SSL>

</paymentDetails>

<shopper>

<shopperEmailAddress>shopper@worldpay.com</shopperEmailAddress>

</shopper>

<statementNarrative>MERCHANT STATEMENT NARRATIVE</statementNarrative>

</order>

</submit>

</paymentService>

#### Response Attribute Mapping

<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE paymentService PUBLIC "-//Worldpay//DTD Worldpay PaymentService v1//EN"

"http://dtd.worldpay.com/paymentService\_v1.dtd">

<paymentService version="1.4" merchantCode="SAPIENTNITROECOM">

<reply>

<orderStatus orderCode="Example\_Alipay\_Order\_Code\_0">

<reference id="3015839286">https://secure-test.worldpay.com/jsp/test/shopper/APMSTSimulator.jsp?customerRef=3013340267&amp;paymentMethod=ALIPAY</reference>

</orderStatus>

</reply>

</paymentService>

### YANDEX

Yandex.Money is primarily an eWallet that shoppers based in the countries in the [Payment Method Properties table](http://support.worldpay.com/support/kb/gg/alternativepayments/apm063.html#Payment_Method_Properties) below use to make real-time online payments. There is an additional option available to shoppers wishing to pay by a credit or debit card linked to their Yandex.Money account.

|  |  |
| --- | --- |
| **Property** | **Description/Value** |
| **Payment Type** | eWallet/credit and debit card |
| **Countries** | Armenia, Azerbaijan, Belarus, Georgia, Kazakhstan, Kyrgyzstan, Moldova, Russia, Tajikistan, Turkmenistan, Ukraine and Uzbekistan |
| **Acceptance Currencies** | RUB and USD |
| **Mode** | DIRECT |

#### Request Attribute Mapping

<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE paymentService PUBLIC '-//Bibit//DTD Bibit PaymentService v1//EN' 'http://dtd.worldpay.com/paymentService\_v1.dtd'>

<paymentService merchantCode="SAPIENTNITROECOM" version="1.4">

<submit>

<order orderCode="Example\_Yandex\_Order\_Code\_0">

<description>Description here</description>

<amount currencyCode="RUB" value="5000" exponent="2" />

<orderContent>

<![CDATA[ ]]>

</orderContent>

<paymentDetails>

<YANDEXMONEY-SSL shopperCountryCode="RU">

<successURL>http://www.worldpay.com/?successURL</successURL>

<cancelURL>http://www.worldpay.com/?cancelURL</cancelURL>

<pendingURL>http://www.worldpay.com/?pendingURL</pendingURL>

</YANDEXMONEY-SSL>

</paymentDetails>

<shopper>

<shopperEmailAddress>shopper@worldpay.com</shopperEmailAddress>

</shopper>

</order>

</submit>

</paymentService>

#### Response Attribute Mapping

<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE paymentService PUBLIC "-//Worldpay//DTD Worldpay PaymentService v1//EN"

"http://dtd.worldpay.com/paymentService\_v1.dtd">

<paymentService version="1.4" merchantCode="SAPIENTNITROECOM">

<reply>

<orderStatus orderCode="Example\_Yandex\_Order\_Code\_0">

<reference id="3015839577">https://secure-test.worldpay.com/jsp/test/shopper/APMSTSimulator.jsp?customerRef=3013340527&amp;paymentMethod=YANDEXMONEY</reference>

</orderStatus>

</reply>

</paymentService>

### QIWI

Qiwi is a payment method that shoppers based in Russia and Kazakhstan can use to make payments. Shoppers who pay with Qiwi can complete their payment using one of the following:

* Their Qiwi eWallet, split into purses for different currencies
* A payment kiosk
* A credit/debit card linked to their Qiwi account
* The prepay balance on the shopper’s mobile phone
* A Webmoney account linked to their Qiwi account

|  |  |
| --- | --- |
| **Property** | **Description/Value** |
| **Payment Type** | eWallet / Postpay via kiosk |
| **Countries** | Russia and Kazakhstan |
| **Acceptance Currencies** | RUB and USD |
| **Mode** | DIRECT |

#### Request Attribute Mapping

<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE paymentService PUBLIC "-//Bibit//DTD Bibit PaymentService v1//EN" "http://dtd.worldpay.com/paymentService\_v1.dtd">

<paymentService merchantCode="SAPIENTNITROECOM" version="1.4">

<submit>

<order orderCode="Example\_Qiwi\_Order\_Code\_0">

<description>Description here</description>

<amount currencyCode="RUB" value="5000" exponent="2"/>

<orderContent>

<![CDATA[ ]]>

</orderContent>

<paymentDetails>

<QIWI-SSL shopperCountryCode="RU">

<successURL>http://www.worldpay.com/?successURL</successURL>

<cancelURL>http://www.worldpay.com/?cancelURL</cancelURL>

<pendingURL>http://www.worldpay.com/?pendingURL</pendingURL>

</QIWI-SSL>

</paymentDetails>

<shopper>

<shopperEmailAddress>shopper@worldpay.com</shopperEmailAddress>

</shopper>

<shippingAddress>

<address>

<address1>Tverskaya Ulitsa 3453</address1>

<postalCode>74564</postalCode>

<city>Moscow</city>

<countryCode>RU</countryCode>

<telephoneNumber>7987654321</telephoneNumber>

</address>

</shippingAddress>

<statementNarrative>MERCHANT STATEMENT NARRATIVE</statementNarrative>

</order>

</submit>

</paymentService>

#### Response Attribute Mapping

<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE paymentService PUBLIC "-//Worldpay//DTD Worldpay PaymentService v1//EN"

"http://dtd.worldpay.com/paymentService\_v1.dtd">

<paymentService version="1.4" merchantCode="SAPIENTNITROECOM">

<reply>

<orderStatus orderCode="Example\_Qiwi\_Order\_Code\_0">

<reference id="3015840491">https://secure-test.worldpay.com/jsp/test/shopper/APMSTSimulator.jsp?customerRef=3013341249&amp;paymentMethod=QIWI</reference>

</orderStatus>

</reply>

</paymentService>

### BOLETO

Boleto Bancário is a post-payment service that allows a shopper to pay for online services over the counter at any supporting outlet in Brazil, or through internet banking. Boleto Bancário is a popular solution for e-commerce payments in Brazil, where a significant proportion of the population does not have access to a bank account.

|  |  |
| --- | --- |
| **Property** | **Description/Value** |
| **Payment Type** | Post-pay voucher |
| **Countries** | Brazil |
| **Acceptance Currencies** | BRL |
| **Mode** | DIRECT |

#### Request Attribute Mapping

<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE paymentService PUBLIC "-//Worldpay/DTD Worldpay PaymentService v1//EN" "http://dtd.worldpay.com/paymentService\_v1.dtd">

<paymentService version="1.4" merchantCode="SAPIENTNITROECOM">

<submit>

<order orderCode="Example\_Boleto\_Order\_32">

<description>test order</description>

<amount value="2000" currencyCode="BRL" exponent="2"/>

<orderContent>

<![CDATA[]]>

</orderContent>

<paymentDetails>

<BOLETO-SSL shopperCountryCode="BR">

<cpf>263.946.533-30</cpf>

<successURL>http://www.worldpay.com/successURL</successURL>

<cancelURL>http://www.worldpay.com/cancelURL</cancelURL>

<pendingURL>http://www.worldpay.com/pendingURL</pendingURL>

</BOLETO-SSL>

</paymentDetails>

<shopper>

<shopperEmailAddress>shopper@worldpay.com</shopperEmailAddress>

</shopper>

<shippingAddress>

<address>

<firstName>Joe</firstName>

<lastName>Bloggs</lastName>

<address1>Rua Castilho 34</address1>

<address2>Santo André</address2>

<address3></address3>

<postalCode>04642-000</postalCode>

<city>São Paulo</city>

<state>São Paulo</state>

<countryCode>BR</countryCode>

<telephoneNumber>0123456789</telephoneNumber>

</address>

</shippingAddress>

</order>

</submit>

</paymentService>

#### Response Attribute Mapping

<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE paymentService PUBLIC "-//Worldpay//DTD Worldpay PaymentService v1//EN"

"http://dtd.worldpay.com/paymentService\_v1.dtd">

<paymentService version="1.4" merchantCode="SAPIENTNITROECOM">

<reply>

<orderStatus orderCode="Example\_Boleto\_Order\_32">

<reference id="3015902434">https://secure-test.worldpay.com/jsp/test/shopper/APMSTSimulator.jsp?customerRef=3013397009&amp;paymentMethod=BOLETO</reference>

</orderStatus>

</reply>

</paymentService>

### MISTER CASH

Mister Cash/Bancontact is a debit card service available in Belgium. Shoppers can use this service to make real-time card payments for products and services.

|  |  |
| --- | --- |
| **Property** | **Description/Value** |
| **Payment Type** | Debit Card payment |
| **Countries** | Belgium |
| **Acceptance Currencies** | EUR |
| **Mode** | DIRECT |

#### Request Attribute Mapping

<?xml version='1.0' ?>

<!DOCTYPE paymentService PUBLIC '-//Worldpay//DTD Worldpay PaymentService v1//EN' 'dtd/paymentService\_v1.dtd'>

<paymentService merchantCode='SAPIENTNITROECOM' version='1.4'>

<submit>

<order orderCode='ORDERCODE\_Mister\_Cash'>

<description>Description here</description>

<amount currencyCode='EUR' value='100' exponent='2'/>

<orderContent>Order Content Here</orderContent>

<paymentDetails>

<MISTERCASH-SSL shopperCountryCode='BE'>

<successURL>http://www.worldpay.com/successURL</successURL>

<cancelURL>http://www.worldpay.com/cancelURL</cancelURL>

<pendingURL>http://www.worldpay.com/pendingURL</pendingURL>

</MISTERCASH-SSL>

</paymentDetails>

<shopper>

<shopperEmailAddress>shopper@worldpay.com</shopperEmailAddress>

</shopper>

<shippingAddress>

<address>

<firstName>Pierre</firstName>

<lastName>Bloggs</lastName>

<address1>Shopperstreet</address1>

<postalCode>1234</postalCode>

<city>Shoppercity</city>

<countryCode>BE</countryCode>

</address>

</shippingAddress>

</order>

</submit>

</paymentService>

#### Response Attribute Mapping

<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE paymentService PUBLIC "-//Worldpay//DTD Worldpay PaymentService v1//EN"

"http://dtd.worldpay.com/paymentService\_v1.dtd">

<paymentService version="1.4" merchantCode="SAPIENTNITROECOM">

<reply>

<orderStatus orderCode="ORDERCODE\_Mister\_Cash">

<reference id="3015902737">https://secure-test.worldpay.com/jsp/test/shopper/APMSTSimulator.jsp?customerRef=3013397287&amp;paymentMethod=MISTERCASH</reference>

</orderStatus>

</reply>

</paymentService>

### PAYPAL

PayPal is an American international [e-commerce](http://en.wikipedia.org/wiki/E-commerce) business allowing payments and money transfers to be made through the Internet.[Online money transfers](http://en.wikipedia.org/wiki/Online_money_transfers) serve as electronic alternatives to paying with traditional paper methods, such as [cheques](http://en.wikipedia.org/wiki/Cheques) and [money orders](http://en.wikipedia.org/wiki/Money_order).

|  |  |
| --- | --- |
| **Property** | **Description/Value** |
| **Payment Type** | Debit Card payment |
| **Countries** | Belgium,All |
| **Acceptance Currencies** | GBP,USD and EUR |
| **Mode** | DIRECT |

#### Request Attribute Mapping

<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE paymentService PUBLIC "-//Bibit//DTD Bibit PaymentService v1//EN" "http://dtd.bibit.com/paymentService\_v1.dtd">

<paymentService version="1.2" merchantCode="SAPIENTNITROECOM">

<submit>

<order orderCode="TG20071016-002">

<description>IT Products</description>

<amount value="8933" currencyCode="EUR" exponent="2"/>

<orderContent>The order content</orderContent>

<paymentDetails>

<PAYPAL-EXPRESS>

<successURL>https://www.merchant.com/paymentProcessor.html?order=TG20071016-

002&amp;status=success</successURL>

<failureURL>https://www.merchant.com/paymentProcessor.html?order=TG20071016-

002&amp;status=failure</failureURL>

<cancelURL><https://www.merchant.com/shoppingCart.html?order=TG20071016->

002</cancelURL>

</PAYPAL-EXPRESS>

</paymentDetails>

</order>

</submit>

</paymentService>

#### Response Attribute Mapping

<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE paymentService PUBLIC "-//Worldpay//DTD Worldpay PaymentService v1//EN"

"http://dtd.worldpay.com/paymentService\_v1.dtd">

<paymentService version="1.4" merchantCode="SAPIENTNITROECOM">

<reply>

<orderStatus orderCode="TG20071016-002">

<reference id="3015903693">https://secure-test.worldpay.com/jsp/simulators/shopper/PayPal\_dummy\_redirect.jsp?returnURL=https%3A%2F%2Fsecure-test.worldpay.com%2Fshopper%2Fpaypal%2FdirectPayment.html%3Ftoken%3DTOKEN1412764560062&amp;cancelURL=https%3A%2F%2Fwww.merchant.com%2FshoppingCart.html%3Forder%3DTG20071016-002&amp;subject=DUMMY&amp;merchant=SAPIENTNITROECOM&amp;order=TG20071016-002&amp;amount=89.33&amp;currency=EUR&amp;description=IT Products</reference>

</orderStatus>

</reply>

</paymentService>

### eNETS

eNETS is a popular real-time bank transfer payment method, available in Singapore. Shoppers can make payments online by logging on to their own bank account. The eNETS payment method supports major browsers and mobile platforms.

|  |  |
| --- | --- |
| **Property** | **Description/Value** |
| **Payment Type** | Debit Card payment |
| **Countries** | Singapore |
| **Acceptance Currencies** | SGD |
| **Mode** | REDIRECT |

#### Request Attribute Mapping

<?xml version="1.0"?><!DOCTYPE paymentService PUBLIC "-//RBS Worldpay/DTD RBS WorldpayPaymentService v1//EN" "http://dtd.wp3.rbsworldpay.com/paymentService\_v1.dtd">

<paymentService merchantCode="SAPIENTNITROECOM" version="1.4">

<submit>

<order orderCode="Sample\_Enets\_0">

<description>Sample\_Enets\_9</description>

<amount currencyCode="SGD" exponent="2" value="1"/>

<orderContent>&lt;center&gt;&lt;table border="1"&gt;&lt;tr&gt;&lt;th&gt;ProductID&lt;/th&gt;&lt;th&gt;Product Name&lt;/th&gt;&lt;th&gt;Quantity&lt;/th&gt;&lt;th&gt;Price&lt;/th&gt;&lt;/tr&gt;&lt;tr&gt;&lt;td&gt;taylor-dress-1951113\_01&lt;/td&gt;&lt;td&gt;TAYLOR DRESS&lt;/td&gt;&lt;td&gt;1&lt;/td&gt;&lt;td&gt;58&lt;/td&gt;&lt;/tr&gt;&lt;tr&gt;&lt;td&gt;taylor-dress-1951113\_02&lt;/td&gt;&lt;td&gt;TAYLOR DRESS&lt;/td&gt;&lt;td&gt;1&lt;/td&gt;&lt;td&gt;60&lt;/td&gt;&lt;/tr&gt;&lt;tr&gt;&lt;td colspan="4"&gt;Your payment will be handled by RBS Worldpay Payments Services&lt;br/&gt;This name may appear on your bank statement&lt;br/&gt;http://www.rbsworldpay.com&lt;/td&gt;&lt;/tr&gt;&lt;/table&gt;&lt;/center&gt;</orderContent>

<paymentMethodMask>

<include code="ENETS-SSL"/>

</paymentMethodMask>

<shopper>

<shopperEmailAddress>manav@gmail.com</shopperEmailAddress>

</shopper>

<shippingAddress>

<address>

<firstName>Manav</firstName>

<lastName>Gulati</lastName>

<street>test</street>

<postalCode>10005</postalCode>

<city>NY</city>

<countryCode>CN</countryCode>

<telephoneNumber>9999999999</telephoneNumber>

</address>

</shippingAddress>

</order>

</submit>

</paymentService>

#### Response Attribute Mapping

<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE paymentService PUBLIC "-//Worldpay//DTD Worldpay PaymentService v1//EN"

"http://dtd.worldpay.com/paymentService\_v1.dtd">

<paymentService version="1.4" merchantCode="SAPIENTNITROECOM">

<reply>

<orderStatus orderCode="Sample\_Enets\_0">

<reference id="3015959319">https://secure-test.worldpay.com/jsp/shopper/SelectPaymentMethod.jsp?OrderKey=SAPIENTNITROECOM%5ESample\_Enets\_0</reference>

</orderStatus>

</reply>

</paymentService>

### iDEAL

iDEAL is a method of payment that enables [consumer](http://www.ideal.nl/en/glossary/#consumer) to pay online through their own bank. In addition to webshops, other online organizations that are not part of the e-commerce market also offer iDEAL.

|  |  |
| --- | --- |
| **Property** | **Description/Value** |
| **Countries** | Netherlands |
| **Acceptance Currencies** | EUR |
| **Mode** | DIRECT |

|  |  |
| --- | --- |
| **Bank Names** | **Bank ID’s** |
| ING | ING |
| ABN | ABN\_AMRO |
| ASN | ASN |
| Rabobank | RABOBANK |
| SNS | SNS |
| SNS Regio | SNS\_REGIO |
| Triodos | TRIODOS |
| Van Lanschot | VAN\_LANSCHOT |
| Knab | KNAB |

#### Request Attribute Mapping

<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE paymentService PUBLIC "-//Worldpay//DTD Worldpay PaymentService v1//EN"

"http://dtd.worldpay.com/paymentService\_v1.dtd">

<paymentService version="1.4" merchantCode="SAPIENTNITROECOM">

<submit>

<order orderCode="TG20090909-123">

<description>IT Products</description>

<amount value="2002" currencyCode="EUR" exponent="2"/>

<orderContent>The order content</orderContent>

<paymentDetails>

<IDEAL-SSL shopperBankCode="ING">

<successURL>https://www.merchant.com/success?order=1249480932689</successURL>

<failureURL>https://www.merchant.com/failure?order=1249480932689</failureURL>

<cancelURL>https://www.merchant.com/cancel?order=1249480932689</cancelURL>

</IDEAL-SSL>

</paymentDetails>

</order>

</submit>

</paymentService>

#### Response Attribute Mapping

<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE paymentService PUBLIC "-//Worldpay//DTD Worldpay PaymentService v1//EN"

"http://dtd.worldpay.com/paymentService\_v1.dtd">

<paymentService version="1.4" merchantCode="SAPIENTNITROECOM">

<reply>

<orderStatus orderCode="TG20090909-123">

<reference id="3015960397">https://secure-test.worldpay.com/wcc/simulator?op=IDEAL-AuthInit&amp;trxId=126208818731246</reference>

</orderStatus>

</reply>

</paymentService>

### GiroPay

This is a popular German bank transfer payment method that shoppers can use to login to their bank account to make an online payment. Giropay payments are authorized immediately after the shopper submits their payment request. Shopper needs to enter the Bank Code value on payment page and Bank details will be sent in as a part of DIRECT XML request as b**ankCode** node.

|  |  |
| --- | --- |
| Property | Description/Value |
| Payment Type | Real Time Bank Transfer |
| Countries | Germany |
| Acceptance Currencies | EUR |
| Mode | DIRECT |
| Payment Method Mask | GIROPAY-SSL |

#### Request Attribute Mapping

<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE paymentService PUBLIC "-//Worldpay/DTD Worldpay PaymentService v1//EN" "http://dtd.worldpay.com/paymentService\_v1.dtd">

<paymentService version="1.4" merchantCode="MERCHANTCODE">

<submit>

<order orderCode="GiropayTestOrder">

<description>test order</description>

<amount value="100" currencyCode="EUR" exponent="2"/>

<orderContent><![CDATA[]]> </orderContent>

<paymentDetails>

<GIROPAY-SSL>

<successURL>http://www.worldpay.com/?success</successURL>

<failureURL>http:// www.worldpay.com/?failure</failureURL>

<cancelURL>http:// www.worldpay.com/?cancel</cancelURL>

<bankCode>12345678</bankCode>

</GIROPAY-SSL>

<session shopperIPAddress="127.0.0.1" id="ssn97414780"/>

</paymentDetails>

<shopper>

<shopperEmailAddress>shopper@worldpay.com</shopperEmailAddress>

<browser>

<acceptHeader>text/html</acceptHeader>

<userAgentHeader>Mozilla/5.0 ...</userAgentHeader>

</browser>

</shopper>

</order>

</submit>

</paymentService>

#### Response Attribute Mapping

<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE paymentService PUBLIC "-//Worldpay//DTD Worldpay PaymentService v1//EN"

"http://dtd.worldpay.com/paymentService\_v1.dtd">

<paymentService version="1.4" merchantCode="SAPIENTNITROECOMMERCEV1">

<reply>

<orderStatus orderCode="00004805">

<reference id="3019678016">https://secure-test.worldpay.com/jsp/test/shopper/APMSTSimulator.jsp?customerRef=3016933093&amp;paymentMethod=GIROPAY/reference>

</orderStatus>

</reply>

</paymentService>

### Konbini

Digital Garage offers a number of payment methods that shoppers in Japan use to shop online. These include:

* Offline cash payment at a convenience store (known as Konbini). The penetration of convenience stores is very high in Japan, so this offers shoppers a very convenient way of making payments.
* Offline electronic payment at an ATM or internet banking (known as *Pay-Easy*).
* Online electronic payment by internet banking.

Since most payment options are off-line, merchants should expect the system to create a payment with SHOPPER\_REDIRECTED status where it can remain for a number of days whilst the shopper makes their payment.

If the shopper pays immediately, the payment is in SHOPPER\_REDIRECTED status for a short period of time and then moves to AUTHORISED.

The following table provides more detail of the available payment options:

|  |  |  |
| --- | --- | --- |
| Options | Notes | Authorization |
| Digital Garage (Konbini) | The shopper places their order. At the end of the shopper journey the shopper views a unique payment reference, along with instructions on how to complete the payment.  The shopper makes a note of the unique payment reference and visits their local convenience store.  At the convenience store the shopper inputs their unique payment reference and phone number into a kiosk. The terminal prints a payment slip which the shopper takes to the store till and pays in cash.  A number of major convenience stores participate in the Konbini scheme, including 7Eleven,  Lawson, FamilyMart, Circle K Sunkus and Seicomart. Not all stores are automatically available to all merchants (please speak to your Worldpay support contact for more information). | Delayed |
| ATM Payment | The shopper places their order. At the end of the shopper journey the shopper is shown a unique payment reference, along with instructions on how to complete the payment. The shopper can either:  Pay at an ATM which is part of the Pay-Easy payment scheme  Pay at an ATM of another institution | Delayed |
| Internet Banking | The shopper places their order. At the end of the shopper journey the shopper is shown a unique payment reference, along with instructions on how to complete the payment. The shopper can either:  Pay at a bank which is part of the Pay-Easy payment scheme  Pay at a bank of another institution  The shopper then logs into their internet banking application to complete the payment. | Delayed |

|  |  |
| --- | --- |
| Property | Description Value |
| Payment Type | Post-Pay Voucher/Internet banking/ATM Payment |
| Countries | Japan |
| Acceptance Currencies | JPY |
| Supported Languages | JP |

#### Request Attribute Mapping

<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE paymentService PUBLIC "-//Bibit//DTD Bibit PaymentService v1//EN" "http://dtd.worldpay.com/paymentService\_v1.dtd">

<paymentService merchantCode="SAPIENTNITROECOMMERCEV1" version="1.4">

<submit>

<order orderCode="ORDERCODE">

<description>Description here</description>

<amount currencyCode="JPY" value="100" exponent="0"/>

<orderContent>

<![CDATA[ ]]>

</orderContent>

<paymentDetails>

<KONBINI-SSL shopperCountryCode="JP">

<successURL>http://www.worldpay.com/?successURL</successURL>

<cancelURL>http://www.worldpay.com/?cancelURL</cancelURL>

<pendingURL>http://www.worldpay.com/?pendingURL</pendingURL>

</KONBINI-SSL>

</paymentDetails>

<shopper>

<shopperEmailAddress>shopper@worldpay.com</shopperEmailAddress>

</shopper>

<shippingAddress>

<address>

<firstName>Doe</firstName>

<lastName>John</lastName>

<address1>Main Street 3453</address1>

<postalCode>74564</postalCode>

<city>Tokyo</city>

<countryCode>JP</countryCode>

<telephoneNumber>08012534568</telephoneNumber>

</address>

</shippingAddress>

</order>

</submit>

</paymentService>

#### Response Attribute Mapping

<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE paymentService PUBLIC "-//Worldpay//DTD Worldpay PaymentService v1//EN"

"http://dtd.worldpay.com/paymentService\_v1.dtd">

<paymentService version="1.4" merchantCode="SAPIENTNITROECOMMERCEV1">

<reply>

<orderStatus orderCode="ORDERCODE">

<reference id="3020978183">https://secure-test.worldpay.com/jsp/test/shopper/APMSTSimulator.jsp?customerRef=3018125177&amp;paymentMethod=KONBINI</reference>

</orderStatus>

</reply>

</paymentService>

### Poli

POLi is real-time bank transfer payment method that shoppers can use to make real-time payments directly from their online bank account.

To make a payment by using this method:

* On the payment page, the shopper selects the Poli payment method. The shopper is redirected to the Poli website.
* The shopper selects the bank from which the payment must be made. The shopper is redirected to the website of the bank.
* The shopper logs on to the bank and makes the payment.

***Payment method properties***

The properties for this payment method are described in the following table.

|  |  |
| --- | --- |
| Property | Description/Value |
| Payment Type | Real-time bank transfer |
| Countries | Australia and New Zealand |
| Default Language | En |
| Supported Languages | En |
| Acceptance Currency and Preferred Currency | * **Australia**: AUD * **New Zealand**: NZD |

#### Request Attribute Mapping

***Australia***

<?xml version='1.0' ?>

<!DOCTYPE paymentService PUBLIC '-//Worldpay//DTD Worldpay PaymentService v1//EN' 'dtd/paymentService\_v1.dtd'>

<paymentService merchantCode='SAPIENTNITROECOMMERCEV1' version='1.4'>

<submit>

<order orderCode='Example\_Poli\_order'>

<description>Description here</description>

<amount currencyCode='AUD' value='100' exponent='2'/>

<orderContent>Order Content Here</orderContent>

<paymentDetails>

<POLI-SSL shopperCountryCode='AU'>

<successURL>http://www.worldpay.com/successURL</successURL>

<cancelURL>http://www.worldpay.com/cancelURL</cancelURL>

<pendingURL>http://www.worldpay.com/pendingURL</pendingURL>

</POLI-SSL>

</paymentDetails>

<shopper>

<shopperEmailAddress>shopper@worldpay.com</shopperEmailAddress>

</shopper>

</order>

</submit>

</paymentService>

***New Zealand***

<?xml version='1.0' ?>

<!DOCTYPE paymentService PUBLIC '-//Worldpay//DTD Worldpay PaymentService v1//EN' 'dtd/paymentService\_v1.dtd'>

<paymentService merchantCode='SAPIENTNITROECOMMERCEV1' version='1.4'>

<submit>

<order orderCode='Example\_PoliNZ\_order'>

<description>Description here</description>

<amount currencyCode='NZD' value='100' exponent='2'/>

<orderContent>Order Content Here</orderContent>

<paymentDetails>

<POLINZ-SSL shopperCountryCode='NZ'>

<successURL>http://www.worldpay.com/successURL</successURL>

<cancelURL>http://www.worldpay.com/cancelURL</cancelURL>

<pendingURL>http://www.worldpay.com/pendingURL</pendingURL>

</POLINZ-SSL>

</paymentDetails>

<shopper>

<shopperEmailAddress>shopper@worldpay.com</shopperEmailAddress>

</shopper>

</order>

</submit>

</paymentService>

#### Response Attribute Mapping

***Australia***

<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE paymentService PUBLIC "-//Worldpay//DTD Worldpay PaymentService v1//EN"

"http://dtd.worldpay.com/paymentService\_v1.dtd">

<paymentService version="1.4" merchantCode="SAPIENTNITROECOMMERCEV1">

<reply>

<orderStatus orderCode="Example\_Poli\_order">

<reference id="3020978372">https://secure-test.worldpay.com/jsp/test/shopper/APMSTSimulator.jsp?customerRef=3018125342&amp;paymentMethod=POLI</reference>

</orderStatus>

</reply>

</paymentService>

***New Zealand***

<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE paymentService PUBLIC "-//Worldpay//DTD Worldpay PaymentService v1//EN"

"http://dtd.worldpay.com/paymentService\_v1.dtd">

<paymentService version="1.4" merchantCode="SAPIENTNITROECOMMERCEV1">

<reply>

<orderStatus orderCode="Example\_PoliNZ\_order">

<reference id="3020978547">https://secure-test.worldpay.com/jsp/test/shopper/APMSTSimulator.jsp?customerRef=3018125494&amp;paymentMethod=POLINZ</reference>

</orderStatus>

</reply>

</paymentService>

### SEPA-DD

SEPA-DD is an alternative payment method service, which provides option to make payment using IBAN or Bank account number. SEPA-DD provides normal as well recurring payment options.

#### Request Attribute Mapping

***ONE-OFF Order***

<?xml version="1.0"?><!DOCTYPE paymentService PUBLIC "-//Worldpay//DTD Worldpay PaymentService v1//EN" "<http://dtd.worldpay.com/paymentService_v1.dtd>"> <paymentService merchantCode="SAPIENTNITROECOM" version="1.4">

<submit>

<order orderCode="00008304">

<description>Merchant Order Number : 00008304</description>

<amount currencyCode="USD" exponent="2" value="11024"/>

<orderContent>Product Name Platinum V Neck Suit Dress price USD 99.00Your payment will be handled by Worldpay Payments Services.</orderContent>

<paymentDetails>

<SEPA\_DIRECT\_DEBIT-SSL>

<bankAccount-SEPA>

<iban>DE93100000000012345678</iban>

<accountHolderName>M.Mustermann</accountHolderName>

</bankAccount-SEPA>

</SEPA\_DIRECT\_DEBIT-SSL>

</paymentDetails>

<shopper>

<shopperEmailAddress>Mustermann@example.de</shopperEmailAddress>

<session id="00008304" shopperIPAddress="14.140.116.135"/>

</shopper>

<billingAddress>

<address>

<firstName>Mustermaan</firstName>

<lastName>M</lastName>

<address1>23</address1>

<postalCode>94109</postalCode>

<city>Berlin</city>

<state>CA</state>

<countryCode>DE</countryCode>

<telephoneNumber>1212121212</telephoneNumber>

</address>

</billingAddress>

<mandate>

<mandateType>ONE-OFF</mandateType>

<mandateId>M-100038056-201832512945355</mandateId>

</mandate>

</order>

</submit>

</paymentService>

***RECURRING ORDER***

<?xml version="1.0"?><!DOCTYPE paymentService PUBLIC "-//Worldpay//DTD Worldpay PaymentService v1//EN" "<http://dtd.worldpay.com/paymentService_v1.dtd>"> <paymentService merchantCode="SAPIENTNITROECOM" version="1.4">

<submit>

<order orderCode="00008305">

<description>Merchant Order Number : 00008305</description>

<amount currencyCode="USD" exponent="2" value="14384"/>

<orderContent>Product Name Platinum V Neck Suit Dress price USD 99.00Your payment will be handled by Worldpay Payments Services.</orderContent>

<paymentDetails>

<SEPA\_DIRECT\_DEBIT-SSL>

<bankAccount-SEPA>

<iban>DE93100000000012345678</iban>

<accountHolderName>M.Mustermann</accountHolderName>

</bankAccount-SEPA>

</SEPA\_DIRECT\_DEBIT-SSL>

</paymentDetails>

<shopper>

<shopperEmailAddress>Mustermann@example.de</shopperEmailAddress>

<session id="00008305" shopperIPAddress="14.140.116.135"/>

</shopper>

<billingAddress>

<address>

<firstName>Mustermaan</firstName>

<lastName>M</lastName>

<address1>23</address1>

<postalCode>94109</postalCode>

<city>Berlin</city>

<state>CA</state>

<countryCode>DE</countryCode>

<telephoneNumber>1212121212</telephoneNumber>

</address>

</billingAddress>

<mandate>

<mandateType>RECURRING</mandateType>

<mandateId>M-100038056-201832512180546</mandateId>

</mandate>

</order>

</submit>

</paymentService>

### Przeleway24

Przelewy24 is a real-time bank transfer, offline or pre-payment payment method that shoppers based in Poland can use to transfer funds directly to merchants from their bank accounts.

***Payment method properties***

|  |  |
| --- | --- |
| Property | Description/Value |
| Payment Type | Real-time bank transfer, offline bank transfer, pre-payment |
| Countries | Poland |
| Acceptance Currencies | PLN |
| Settlement to Worldpay Currency | PLN |
| Chargebacks (Yes/No) | Yes |
| Supported Languages | PL (Polish), DE (German), ES (Spanish), IT (Italian)  There is also an option to switch to EN (English) on the Przelewy24 payment page. |
| Direct Model (Yes/No) | Yes |

#### Request Attribute Mapping

<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE paymentService PUBLIC '-//Worldpay//DTD Worldpay PaymentService v1//EN' 'http://dtd.worldpay.com/paymentService\_v1.dtd'>

<paymentService merchantCode="SAPIENTNITROECOMMERCEV1" version="1.4">

<submit>

<order orderCode="Example\_Przelewy24\_Order\_Code">

<description>Description here</description>

<amount currencyCode="PLN" value="100" exponent="2" />

<orderContent>

<![CDATA[ ]]>

</orderContent>

<paymentDetails>

<PRZELEWY-SSL shopperCountryCode="PL">

<successURL>http://www.worldpay.com/?successURL</successURL>

<cancelURL>http://www.worldpay.com/?cancelURL</cancelURL>

<pendingURL>http://www.worldpay.com/?pendingURL</pendingURL>

</PRZELEWY-SSL>

</paymentDetails>

<shopper>

<shopperEmailAddress>shopper@worldpay.com</shopperEmailAddress>

</shopper>

</order>

</submit>

</paymentService>

#### Response Attribute Mapping

<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE paymentService PUBLIC "-//Worldpay//DTD Worldpay PaymentService v1//EN"

"http://dtd.worldpay.com/paymentService\_v1.dtd">

<paymentService version="1.4" merchantCode="SAPIENTNITROECOMMERCEV1">

<reply>

<orderStatus orderCode="Example\_Przelewy24\_Order\_Code">

<reference id="3020980234">https://secure-test.worldpay.com/jsp/test/shopper/APMSTSimulator.jsp?customerRef=3018127000&amp;paymentMethod=PRZELEWY</reference>

</orderStatus>

</reply>

</paymentService>

### CashU

CashU is a popular eWallet service that allows shoppers to make online payments in the Middle East, North Africa and many other countries including Russia and Canada. Shoppers can load the eWallet in a variety of ways such as Ukash voucher and cash over the counter. In some countries a CashU prepaid card and credit card are also available.

|  |  |
| --- | --- |
| Property | Description/Value |
| Payment Type | eWallet |
| Countries | Bahrain, Canada, Egypt, Jordan, Kuwait, Lebanon, Malaysia, Mauritania, Morocco, Oman, Qatar, Saudi Arabia, Tunisia, Turkey, United Arab Emirates, Israel, Russian Federation, Tanzania, Ukraine, Uruguay, Venezuela |
| Acceptance Currencies | USD |
| Mode | DIRECT |
| Payment Method Mask | CASHU-SSL |

#### Request Attribute Mapping

<?xml version="1.0"?><!DOCTYPE paymentService PUBLIC "-//Worldpay//DTD Worldpay PaymentService v1//EN" "http://dtd.worldpay.com/paymentService\_v1.dtd"><paymentService merchantCode="SAPIENTNITROECOMMERCEV1" version="1.4">

<submit>

<order orderCode="00004805">

<description>Merchant Order No 00004805</description>

<amount currencyCode="USD" exponent="2" value="29398"/>

<orderContent>&lt;center&gt;&lt;table border="1"&gt;&lt;tr&gt;&lt;th&gt;Product ID&lt;/th&gt;&lt;th&gt;Product Name&lt;/th&gt;&lt;th&gt;Quantity&lt;/th&gt;&lt;th&gt;Price&lt;/th&gt;&lt;/tr&gt;&lt;tr&gt;&lt;td&gt;750518703626&lt;/td&gt;&lt;td&gt;Black Flat Front Wool Suit&lt;/td&gt;&lt;td&gt;1&lt;/td&gt;&lt;td&gt;USD 269.99&lt;/td&gt;&lt;/tr&gt;&lt;tr&gt;&lt;td colspan="4"&gt;Your payment will be handled by Worldpay Payments Services&lt;br/&gt;This name may appear on your bank statement&lt;br/&gt;http://www.worldpay.com&lt;/td&gt;&lt;/tr&gt;&lt;/table&gt;&lt;/center&gt;</orderContent>

<paymentDetails>

<CASHU-SSL shopperCountryCode="BH">

<successURL>https://worldpay02-tech-prtnr-eu02-dw.demandware.net/on/demandware.store/Sites-SiteGenesis-Site/default/Worldpay-Success?orderNo=00004805</successURL>

<cancelURL>https://worldpay02-tech-prtnr-eu02-dw.demandware.net/on/demandware.store/Sites-SiteGenesis-Site/default/Worldpay-Cancel?orderNo=00004805&amp;apmName=CASHU-SSL</cancelURL>

<pendingURL>https://worldpay02-tech-prtnr-eu02-dw.demandware.net/on/demandware.store/Sites-SiteGenesis-Site/default/Worldpay-Pending?orderNo=00004805&amp;apmName=CASHU-SSL</pendingURL>

</CASHU-SSL>

</paymentDetails>

<shopper>

<shopperEmailAddress>usingh5@sapient.com</shopperEmailAddress>

</shopper>

</order>

</submit>

</paymentService>

#### Response Attribute Mapping

<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE paymentService PUBLIC "-//Worldpay//DTD Worldpay PaymentService v1//EN"

"http://dtd.worldpay.com/paymentService\_v1.dtd">

<paymentService version="1.4" merchantCode="SAPIENTNITROECOMMERCEV1">

<reply>

<orderStatus orderCode="00004805">

<reference id="3019678016">https://secure-test.worldpay.com/jsp/test/shopper/APMSTSimulator.jsp?customerRef=3016933093&amp;paymentMethod=CASHU</reference>

</orderStatus>

</reply>

</paymentService>

### KLARNA

Klarna is a 'buy now, pay later' deferred payment method that gives shoppers the chance to view and use product before they pay for it. Klarna provide three payments methods to the shopper, Pay later, Slice it and Pay now, each with further options to give maximum flexibility.

|  |  |
| --- | --- |
| Property | Description/Value |
| Payment Type | Invoice, Account, Planned Payments, Pay in X months |
| Countries | Invoice: Austria, Finland, Germany, Netherlands, Norway, Sweden, United Kingdom  Account: Finland, Germany, Norway, Sweden  Planned Payments: Finland, Germany, Norway, Sweden  Pay in X Month: Finland, Germany, Norway, Sweden |
| Acceptance Currencies | EUR,GBP,NOK,SEK |
| Mode | DIRECT and HPP |
| Payment Method Mask | KLARNA-SSL |

#### Request Attribute Mapping Direct

<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE paymentService PUBLIC "-//RBS WorldPay//DTD RBS WorldPay PaymentService v1//EN" "http://dtd.wp3.rbsworldpay.com/paymentService\_v1.dtd">

<paymentService version="1.4" merchantCode="$$MERCHANT\_CODE$$">

  <submit>

    <order orderCode="$$ORDER\_CODE$$">

      <description>jsredirect test order</description>

      <amount value="20000" currencyCode="GBP" exponent="2"/>

      <orderContent>

        <![CDATA[]]>

      </orderContent>

      <paymentDetails>

        <KLARNA-SSL>

          <purchaseCountry>gb</purchaseCountry>

          <shopperLocale>en-gb</shopperLocale>

          <merchantUrls>

            <checkoutURL>http://www.companyname.co.uk</checkoutURL>

            <confirmationURL>http://www.success.com</confirmationURL>

          </merchantUrls>

        </KLARNA-SSL>

      </paymentDetails>

      <shopper>

        <shopperEmailAddress>sp@worldpay.com</shopperEmailAddress>

      </shopper>

      <orderLines>

        <orderTaxAmount>4000</orderTaxAmount>

        <termsURL>http://www.terms.co.uk?test</termsURL>

        <lineItem>

          <physical/>

          <reference>1</reference>

          <name>Bananas</name>

          <quantity>10</quantity>

          <quantityUnit>bag</quantityUnit>

          <unitPrice>1000</unitPrice>

          <taxRate>2500</taxRate>

          <totalAmount>10000</totalAmount>

          <totalTaxAmount>2000</totalTaxAmount>

          <totalDiscountAmount>0</totalDiscountAmount>

        </lineItem>

        <lineItem>

          <physical/>

          <reference>2</reference>

          <name>Oranges</name>

          <quantity>1</quantity>

          <quantityUnit>bag</quantityUnit>

          <unitPrice>10000</unitPrice>

          <taxRate>2500</taxRate>

          <totalAmount>10000</totalAmount>

          <totalTaxAmount>2000</totalTaxAmount>

          <totalDiscountAmount>0</totalDiscountAmount>

        </lineItem>

      </orderLines>

    </order>

  </submit>

</paymentService>

#### Response Attribute Mapping Direct

<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE paymentService PUBLIC "-//WorldPay//DTD WorldPay PaymentService v1//EN" "https://secure.worldpay.com/dtd/paymentService\_v1.dtd">

  <paymentService version="1.4" merchantCode="YOUR\_MERCHANT\_CODE">

    <reply>

      <orderStatus orderCode="jsxml3857727180">

        <payment>

          <paymentMethod>KLARNA-SSL</paymentMethod>

          <amount value="20000" currencyCode="EUR" exponent="2" debitCreditIndicator="credit"/>

          <lastEvent>SHOPPER\_REDIRECTED</lastEvent>

        </payment>

      <reference id="be76920a-0fc3-5299-996f-af7476253837">

      <![CDATA[PGRpdiBpZD0ia2xhcm5hLWNoZWNrb3V0LWNvbnRhaW5lciIgc3R5bGU9Im92ZXJmbG93LXg6IGhpZGRlbjsiPgogICAgPHNjcmlwdCB0eXBlPSJ0ZXh0L2phdmFzY3JpcHQiPgogICAgLyogPCFbQ0RBVEFbICovCiAgICAgICAgKGZ1bmN0aW9uKHcsayxpLGQsbixjLGwscCl7CiAgICAgICAgICAgIHdba109d1trXXx8ZnVuY3Rpb24oKXsod1trXS5x>

#### Request Attribute Mapping Hosted Payment Pages

<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE paymentService PUBLIC "-//RBS WorldPay//DTD RBS WorldPay PaymentService v1//EN" "http://dtd.wp3.rbsworldpay.com/paymentService\_v1.dtd">

<paymentService version="1.4" merchantCode="$$MERCHANT\_CODE$$">

  <submit>

    <order orderCode="$$ORDER\_CODE$$" installationId="$$INSTALLATION\_ID$$">

      <description>jsredirect test order</description>

      <amount value="20000" currencyCode="GBP" exponent="2"/>

      <orderContent>

        <![CDATA[]]>

      </orderContent>

      <paymentMethodMask>

        <include code="ALL"/>

      </paymentMethodMask>

      <shopper>

        <shopperEmailAddress>sp@worldpay.com</shopperEmailAddress>

      </shopper>

      <shippingAddress>

        <address>

          <firstName>John</firstName>

          <lastName>Doe</lastName>

          <address1>Units 270 - 289</address1>

          <address2>The Science Park</address2>

          <address3>Milton Road</address3>

          <postalCode>CB4 0WE</postalCode>

          <city>Cambridge</city>

          <state>Cambridgeshire</state>

          <countryCode>GB</countryCode>

          <telephoneNumber>01223 123456</telephoneNumber>

        </address>

      </shippingAddress>

      <orderLines>

        <orderTaxAmount>4000</orderTaxAmount>

        <termsURL>http://www.terms.co.uk?test</termsURL>

        <lineItem>

          <physical/>

            <reference>1</reference>

            <name>Bananas</name>

            <quantity>10</quantity>

            <quantityUnit>bag</quantityUnit>

            <unitPrice>1000</unitPrice>

            <taxRate>2500</taxRate>

            <totalAmount>10000</totalAmount>

            <totalTaxAmount>2000</totalTaxAmount>

            <totalDiscountAmount>0</totalDiscountAmount>

        </lineItem>

        <lineItem>

          <physical/>

            <reference>2</reference>

            <name>Oranges</name>

            <quantity>1</quantity>

            <quantityUnit>bag</quantityUnit>

            <unitPrice>10000</unitPrice>

            <taxRate>2500</taxRate>

            <totalAmount>10000</totalAmount>

            <totalTaxAmount>2000</totalTaxAmount>

            <totalDiscountAmount>0</totalDiscountAmount>

        </lineItem>

      </orderLines>

    </order>

  </submit>

</paymentService>

#### Response Attribute Mapping Mapping Hosted Payment Pages

<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE paymentService PUBLIC "-//WorldPay//DTD WorldPay PaymentService v1//EN""http://dtd.worldpay.com/paymentService\_v1.dtd">

  <paymentService version="1.4" merchantCode="YOUR\_MERCHANT\_CODE">

    <reply>

      <orderStatus orderCode="jsxml340676070">

        <reference id="6052026061">

https://payments.worldpay.com/app/hpp/integration/wpg/corporate?OrderKey=YOUR\_MERCHANT\_CODE%5Ejsxml340676070&amp;Ticket=00148933873090202KHvnMUxIT69nb3TvO-85Vg&amp;source=sc5

        </reference>

      </orderStatus>

    </reply>

  </paymentService>

### MOTO (Not available in controller)

MOTO is the mobile or telephone order, where the customer service agent placing the order for customer.

In order to use moto payment your customer service center agent should have the ‘Login\_on\_behalf’, ‘Login\_Agent permission’.

1. Navigate to Administration > Organization > Roles and permission
2. Click the roles
3. Click functional permission
4. Select the site
5. Give ‘Login\_on\_behalf’, ‘Login\_Agent permission.

Moto payment can be handled in two ways

1). Customer service agent placing the order for the customer from business manager itself. In order to use this update the following settings in your business manager, just to disable the amount to be paid field in the credit card payment section.

1. Navigate Administration > Site development > Customer service center settings
2. Open Order payment Instrument in customizable area
3. Update the existing condtion "condition" : "data.payment\_method\_id=== 'CREDIT\_CARD'"

Search for "condition" : "data.country !== 'FR'",

After that add

"meta" : {

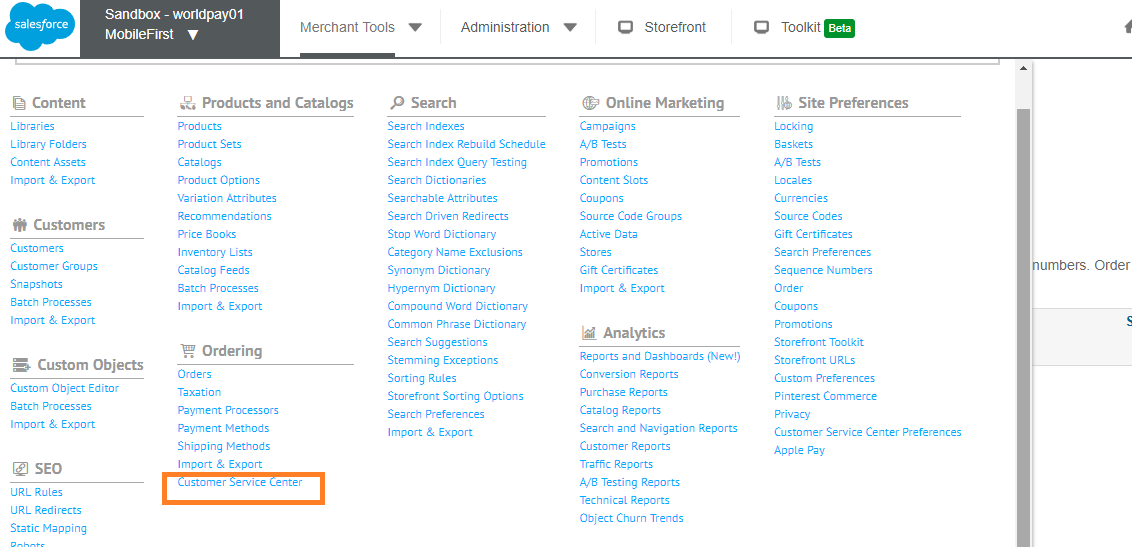
"disabled" : true

},

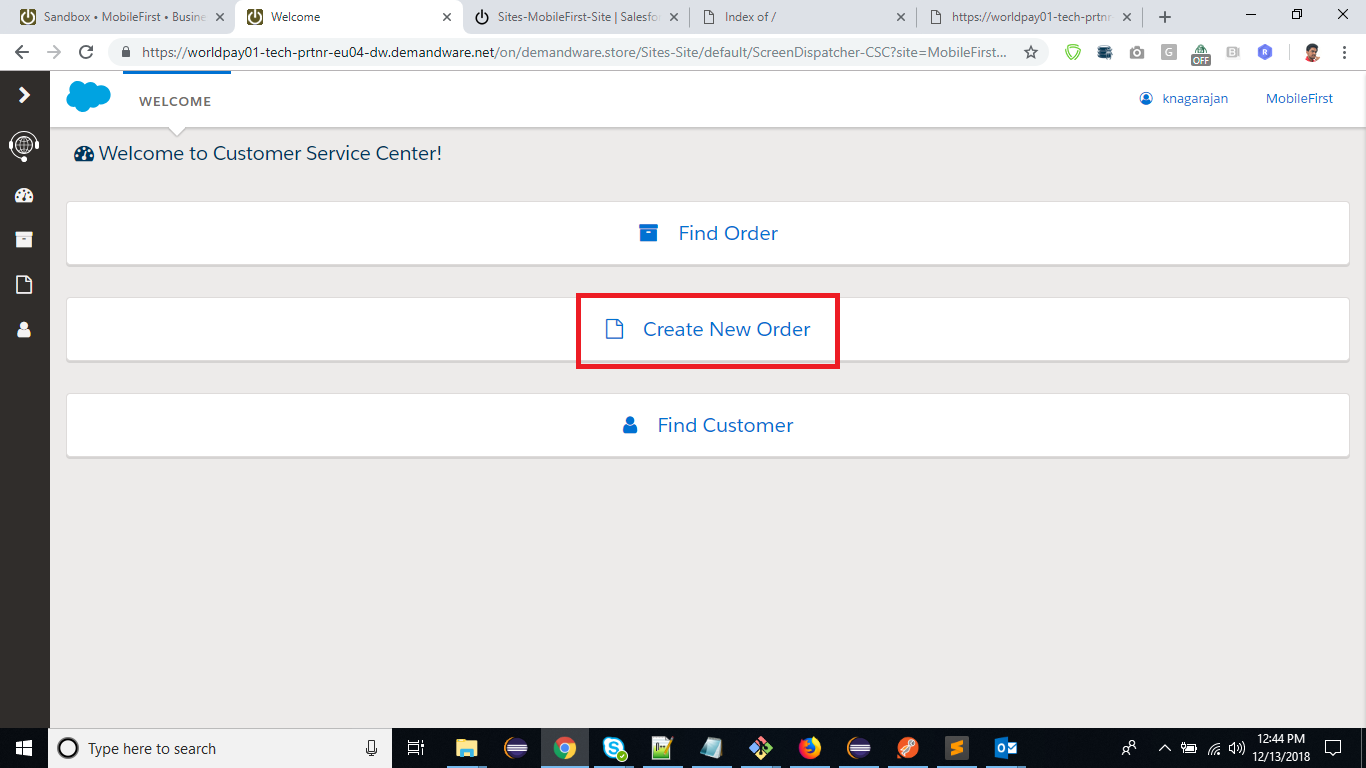
f. Since the customer service center customization works with OCAPI settings, make sure your site have the access to ‘POST /orders/{order\_no}/payment\_instruments’ in (Administration > Site Development > Open commerce settings)

Please refer the attached workflow .

Screen1

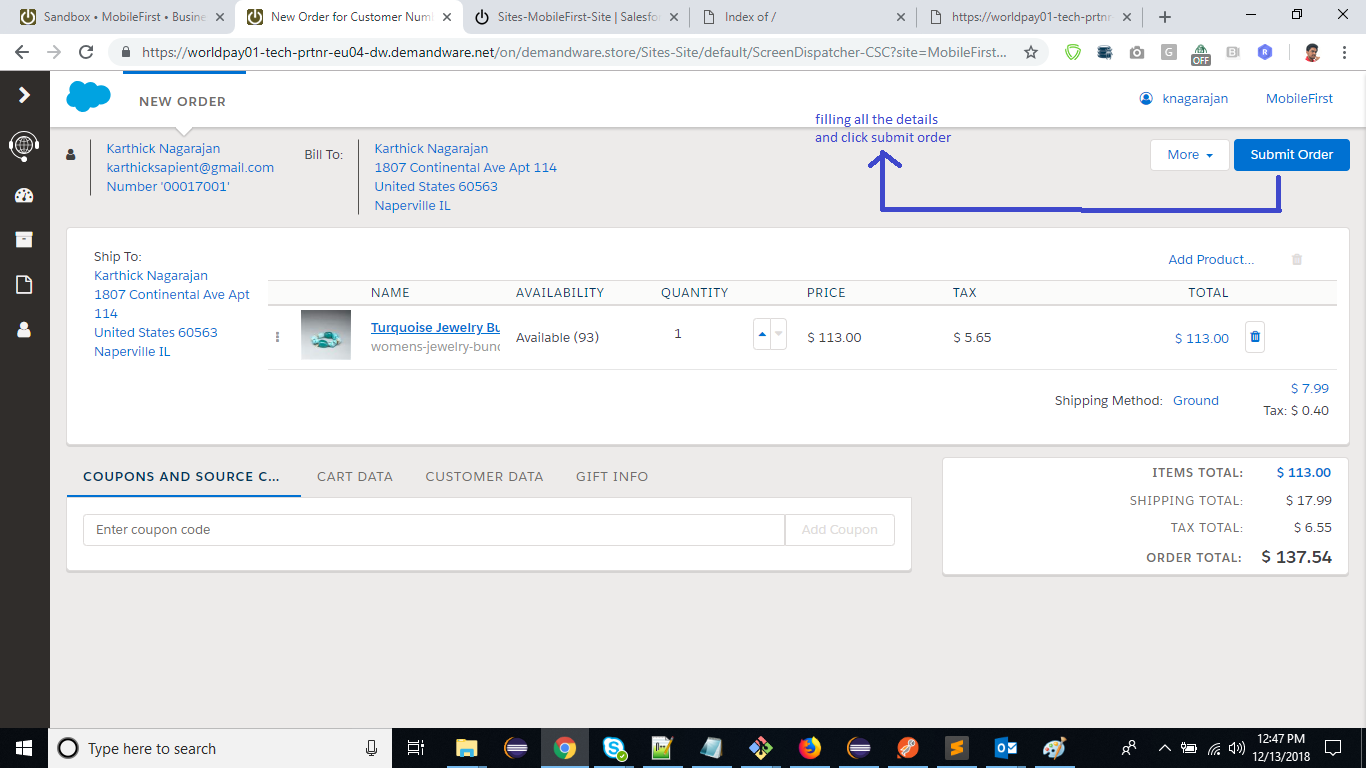


Screen 2



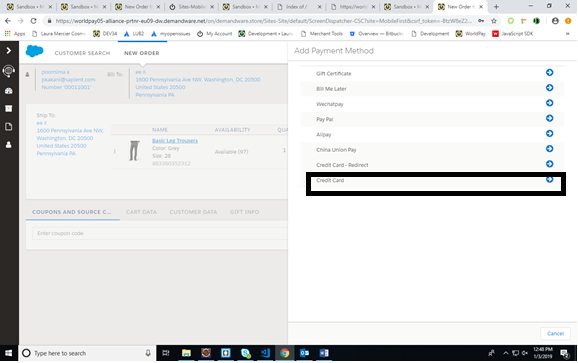
Screen 3

* Fill all the details of the customer and click submit order button



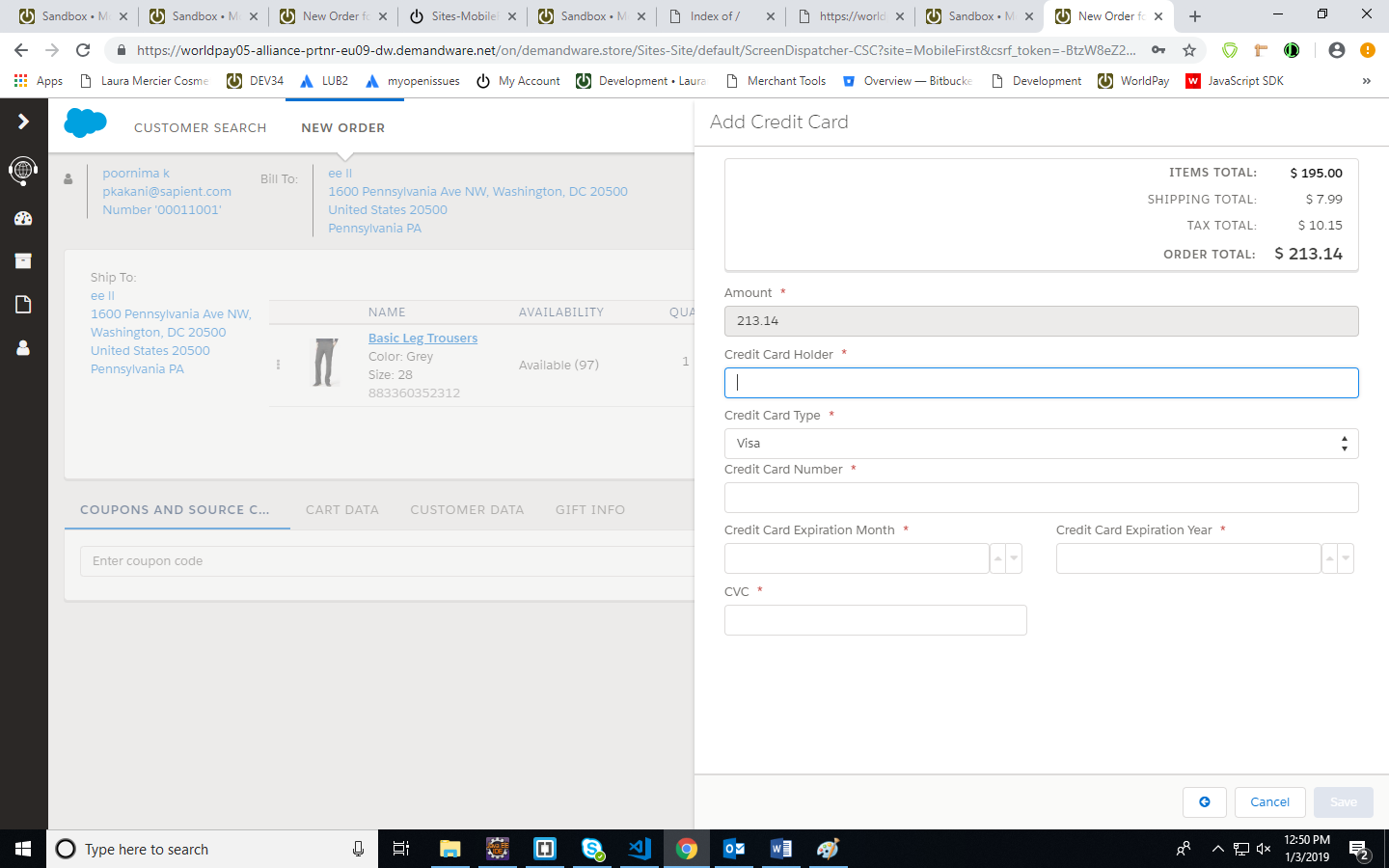
Screen 4

* Worldpay moto payment method listed in payment methods section



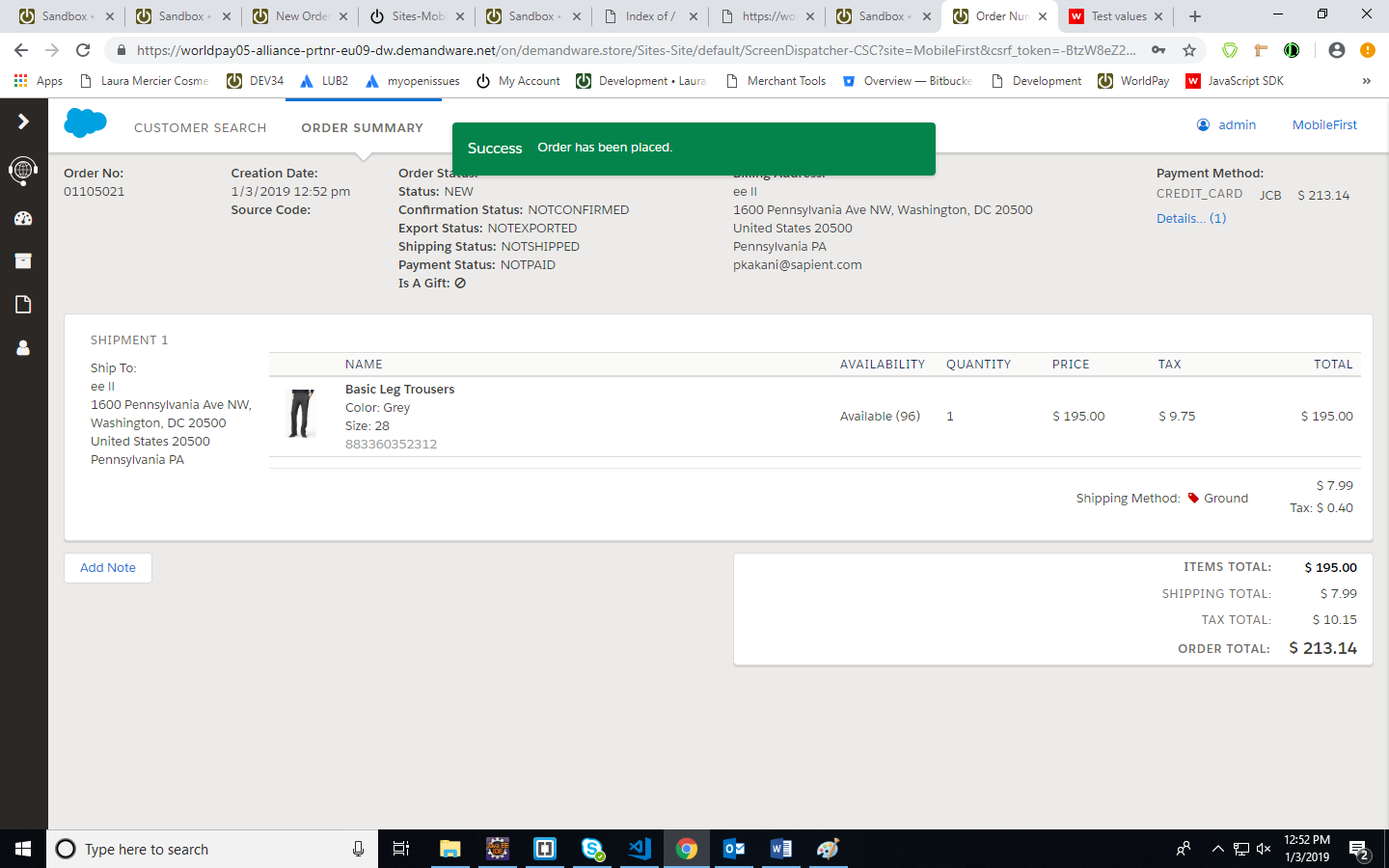
Screen 5

Filling all the credit card details of the customer and click save



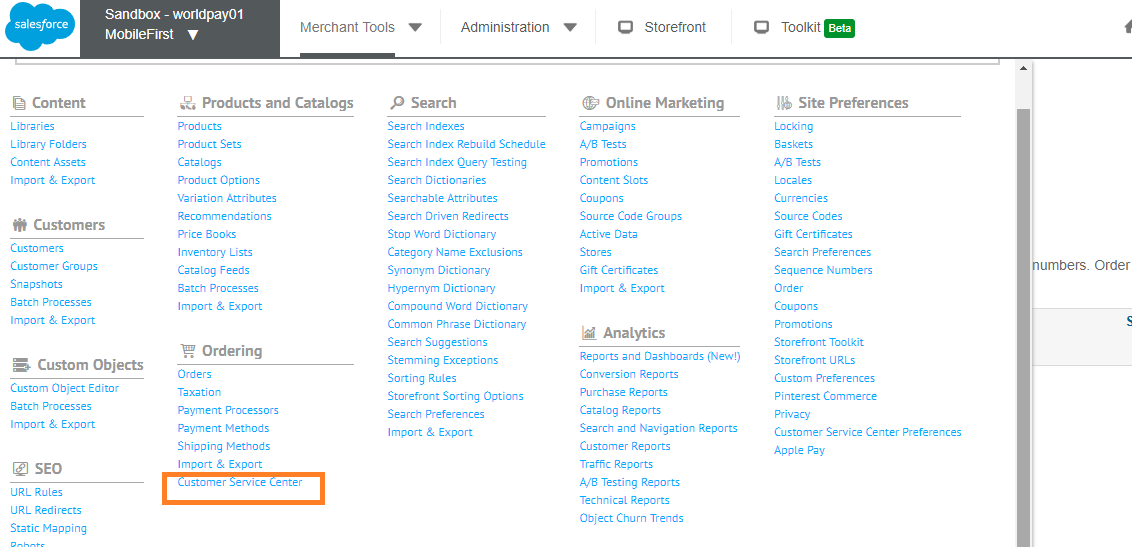
Screen 6

* Order placed successfully for the customer

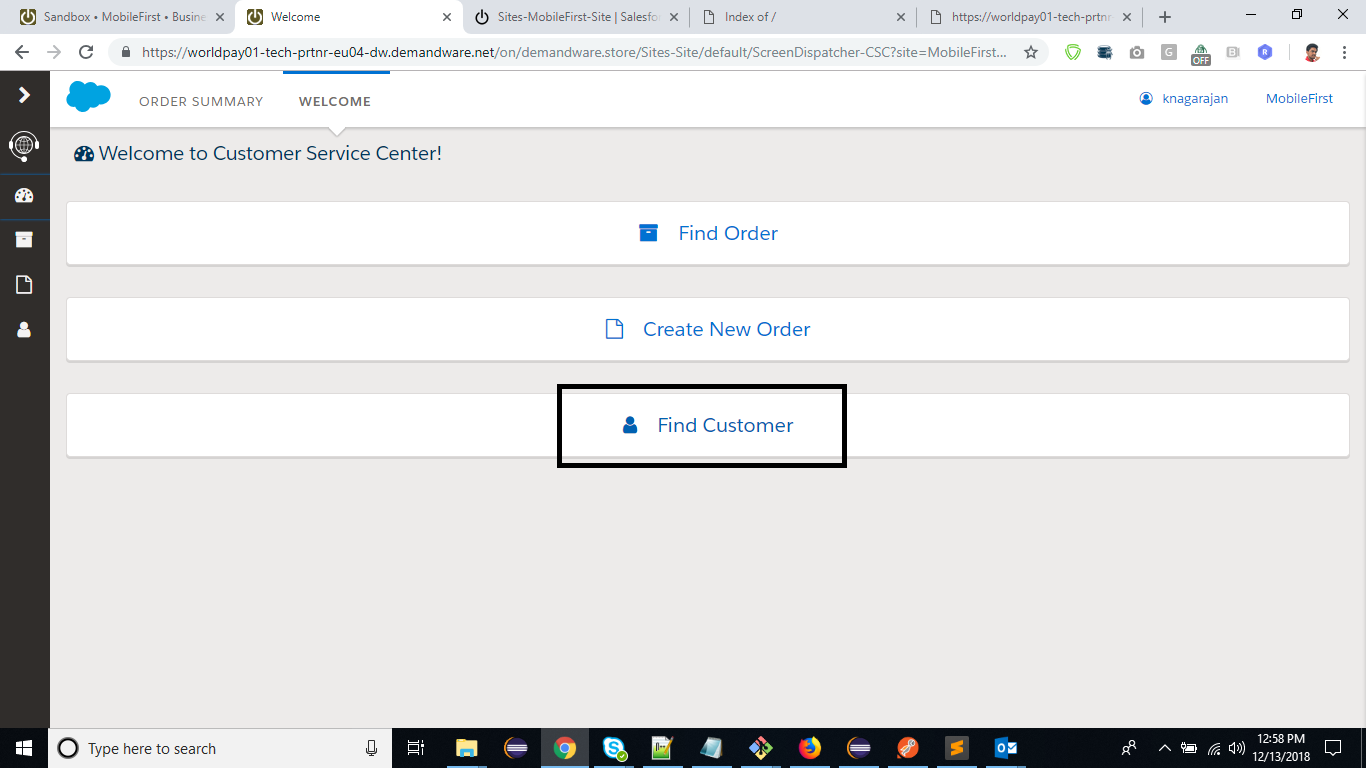


2). Customer service agent placing the order for the customer from storefront. Refer the attached workflow.

In order to enable storefront button in customer centre, update your storefront hostname and storefront path in (Merchant tools > site preferences > custom preferences)

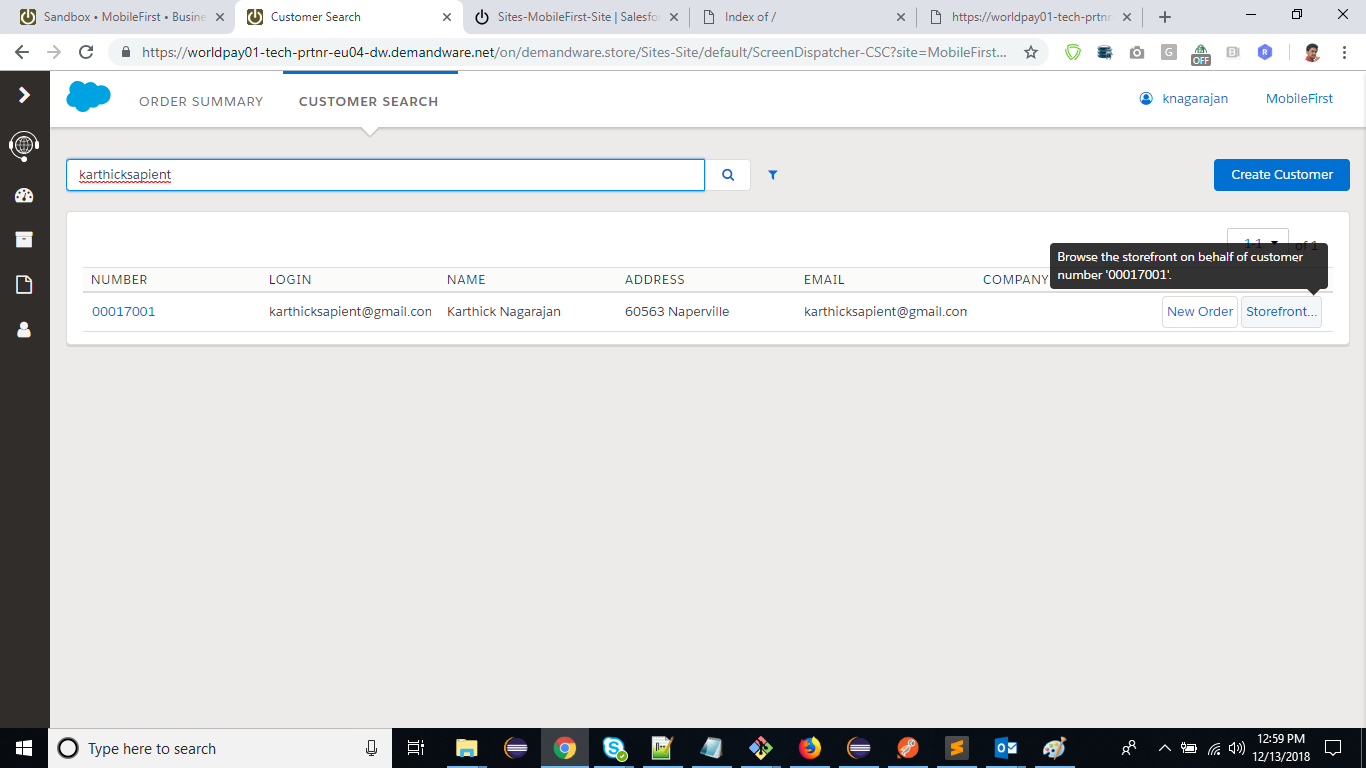
Screen 1

Screen 2



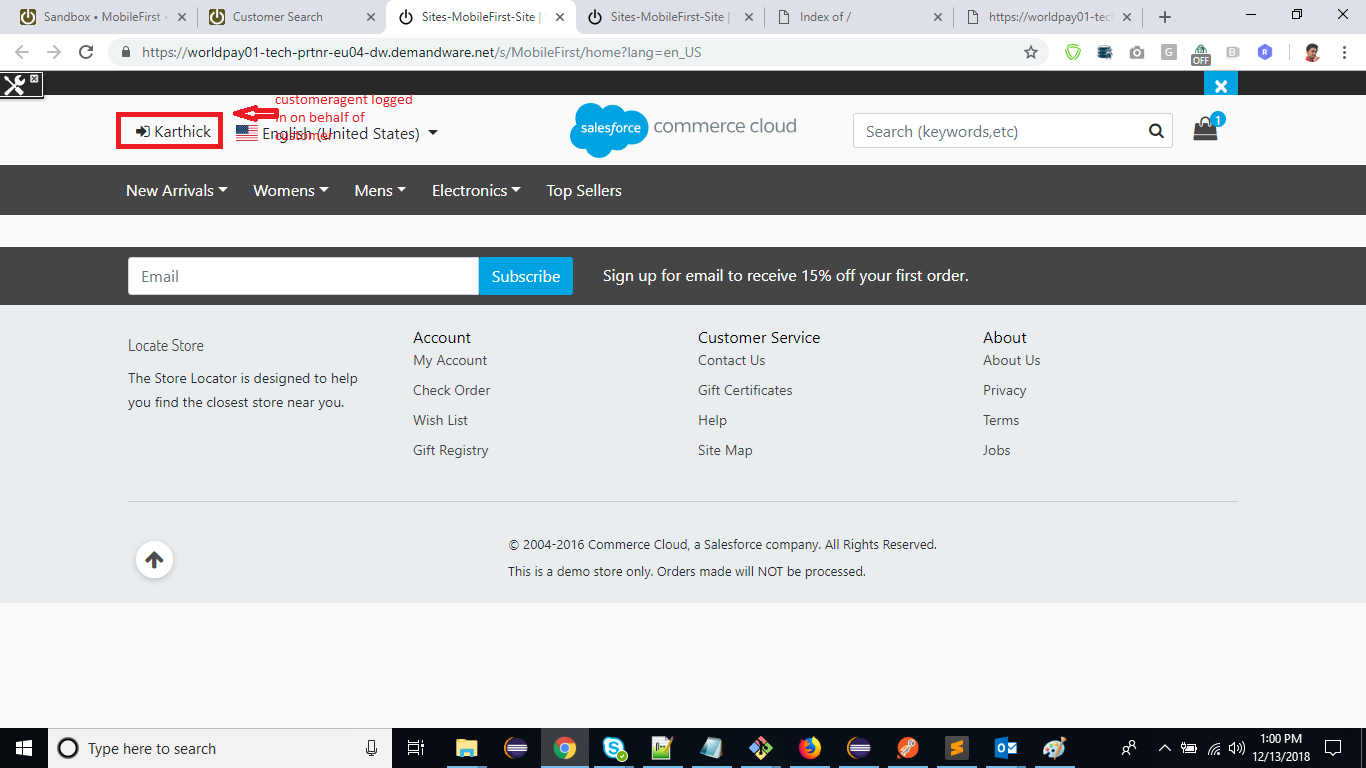
Screen 3

* Search the customer and click storefront



Screen 4

* Customer agent logged in on behalf of customer, here in the storefront checkout page they can’t see the moto as separate payment method. They can normally use credit card payment methods the request will go as moto order



Request and Response will be same as the credit card payment method, only the additional part in the xml is ‘<dynamicInteractionType type="MOTO"/>’

**Request**

<?xml version="1.0"?><!DOCTYPE paymentService PUBLIC "-//Worldpay/DTD Worldpay PaymentService v1//EN" "http://dtd.worldpay.com/paymentService\_v1.dtd"><paymentService merchantCode="SAPIENTNITROECOM" version="1.4">

<submit>

<order orderCode="T00048284">

<description>Merchant Order Number : T00048284</description>

<amount currencyCode="USD" exponent="2" value="3149"/>

<orderContent>&lt;center&gt;&lt;table border="1"&gt;&lt;tr&gt;&lt;th&gt;Product ID&lt;/th&gt;&lt;th&gt;Product Name&lt;/th&gt;&lt;th&gt;Quantity&lt;/th&gt;&lt;th&gt;Price&lt;/th&gt;&lt;/tr&gt;&lt;tr&gt;&lt;td&gt;793775064963&lt;/td&gt;&lt;td&gt;Striped Silk Tie&lt;/td&gt;&lt;td&gt;1&lt;/td&gt;&lt;td&gt;USD 29.99&lt;/td&gt;&lt;/tr&gt;&lt;tr&gt;&lt;td colspan="4"&gt;Your payment will be handled by Worldpay Payments Services&lt;br/&gt;This name may appear on your bank statement&lt;br/&gt;http://www.worldpay.com&lt;/td&gt;&lt;/tr&gt;&lt;/table&gt;&lt;/center&gt;</orderContent>

<paymentDetails>

<ECMC-SSL>

<cardNumber>5555555555554444</cardNumber>

<expiryDate>

<date month="4" year="2021"/>

</expiryDate>

<cardHolderName>KARTHICK NAGARAJ</cardHolderName>

<cardAddress>

<address>

<firstName>Karthick</firstName>

<lastName>Nagarajan</lastName>

<address1>1807 Continental Ave Apt 114</address1>

<address2/>

<postalCode>60563</postalCode>

<city>Naperville</city>

<state>IL</state>

<countryCode>US</countryCode>

<telephoneNumber>123456789</telephoneNumber>

</address>

</cardAddress>

</ECMC-SSL>

<session id="T00048284" shopperIPAddress="125.16.91.5"/>

</paymentDetails>

<shopper>

<shopperEmailAddress>karthicksapient@gmail.com</shopperEmailAddress>

<browser>

<acceptHeader>\*/\*</acceptHeader>

<userAgentHeader>Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/70.0.3538.110 Safari/537.36</userAgentHeader>

</browser>

</shopper>

<shippingAddress>

<address>

<firstName>Karthick</firstName>

<lastName>Nagarajan</lastName>

<address1>1807 Continental Ave Apt 114</address1>

<address2/>

<postalCode>60563</postalCode>

<city>Naperville</city>

<state>IL</state>

<countryCode>US</countryCode>

<telephoneNumber>2343422234</telephoneNumber>

</address>

</shippingAddress>

<dynamicInteractionType type="MOTO"/>

</order>

</submit>

</paymentService>

**Response**

<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE paymentService PUBLIC "-//WorldPay//DTD WorldPay PaymentService v1//EN"

"http://dtd.worldpay.com/paymentService\_v1.dtd">

<paymentService version="1.4" merchantCode="SAPIENTNITROECOM">

<reply>

<orderStatus orderCode="T00048284">

<payment>

<paymentMethod>ECMC-SSL</paymentMethod>

<paymentMethodDetail>

<card number="5555\*\*\*\*\*\*\*\*4444" hash="0x65bb11198e83d0f98592df932a1bd5a979fc7c25" type="creditcard">

<expiryDate>

<date month="04" year="2021"/>

</expiryDate>

</card>

</paymentMethodDetail>

<amount value="3149" currencyCode="USD" exponent="2" debitCreditIndicator="credit"/>

<lastEvent>AUTHORISED</lastEvent>

<AuthorisationId id="237758"/>

<CVCResultCode description="B"/>

<AVSResultCode description="E"/>

<cardHolderName>

<![CDATA[KARTHICK NAGARAJ]]>

</cardHolderName>

<issuerCountryCode>N/A</issuerCountryCode>

<balance accountType="IN\_PROCESS\_AUTHORISED">

<amount value="3149" currencyCode="USD" exponent="2" debitCreditIndicator="credit"/>

</balance>

<riskScore value="-5"/>

</payment>

</orderStatus>

</reply>

</paymentService>

### WECHATPAY

WeChat Pay is an eWallet payment method, incorporated into the WeChat app. It is one of the most popular payment methods in China, with over 600 million active users WeChat Pay is available to shoppers who hold a Chinese domestic bank account. When adding a bank card (credit or debit) to their WeChat wallet, shoppers must also provide their Chinese ID (resident identity card), name and phone number. These details are verified in real-time with their bank and must match exactly for successful verification. Without successful verification shoppers will not be able to use WeChat Pay for online purchases.

WeChat Pay is available in 10 currencies but shoppers will always have their wallet debited in CNY. WeChat Pay will perform any currency conversion immediately and show the CNY amount throughout the shopper journey. The exchange rate is obtained from their partner settlement banks (China Construction Bank or China CITIC Bank) at 10am each day (02:00 UTC), which is shown to the shopper on the payment success page.

Shoppers can link multiple bank cards to their wallet. They can opt to fund their payment from their wallet, provided they have a sufficient balance, or directly from their linked cards at time of payment.:

* eWallet

|  |  |
| --- | --- |
| **Property** | **Description/Value** |
| **Payment Type** | eWallet |
| **Countries** | China |
| **Acceptance Currencies** | CNY |
| **Mode** | DIRECT |

#### Request Attribute Mapping

<?xml version="1.0"?><!DOCTYPE paymentService PUBLIC "-//Worldpay/DTD Worldpay PaymentService v1//EN" "http://dtd.worldpay.com/paymentService\_v1.dtd"><paymentService merchantCode="SAPIENTNITROECOM" version="1.4">

<submit>

<order installationId="1029012" orderCode="01154011">

<description>Merchant Order Number : 01154011</description>

<amount currencyCode="USD" exponent="2" value="7349"/>

<orderContent>&lt;center&gt;&lt;table border="1"&gt;&lt;tr&gt;&lt;th&gt;Product ID&lt;/th&gt;&lt;th&gt;Product Name&lt;/th&gt;&lt;th&gt;Quantity&lt;/th&gt;&lt;th&gt;Price&lt;/th&gt;&lt;/tr&gt;&lt;tr&gt;&lt;td&gt;701644388317&lt;/td&gt;&lt;td&gt;Stripe Button Down Shirt.&lt;/td&gt;&lt;td&gt;1&lt;/td&gt;&lt;td&gt;USD 64.00&lt;/td&gt;&lt;/tr&gt;&lt;tr&gt;&lt;td colspan="4"&gt;Your payment will be handled by Worldpay Payments Services&lt;br/&gt;This name may appear on your bank statement&lt;br/&gt;http://www.worldpay.com&lt;/td&gt;&lt;/tr&gt;&lt;/table&gt;&lt;/center&gt;</orderContent>

<paymentDetails>

<WECHATPAY-SSL/>

</paymentDetails>

<shopper>

<shopperEmailAddress>admin@gmail.com</shopperEmailAddress>

<browser deviceType="0"/>

</shopper>

</order>

</submit>

</paymentService>

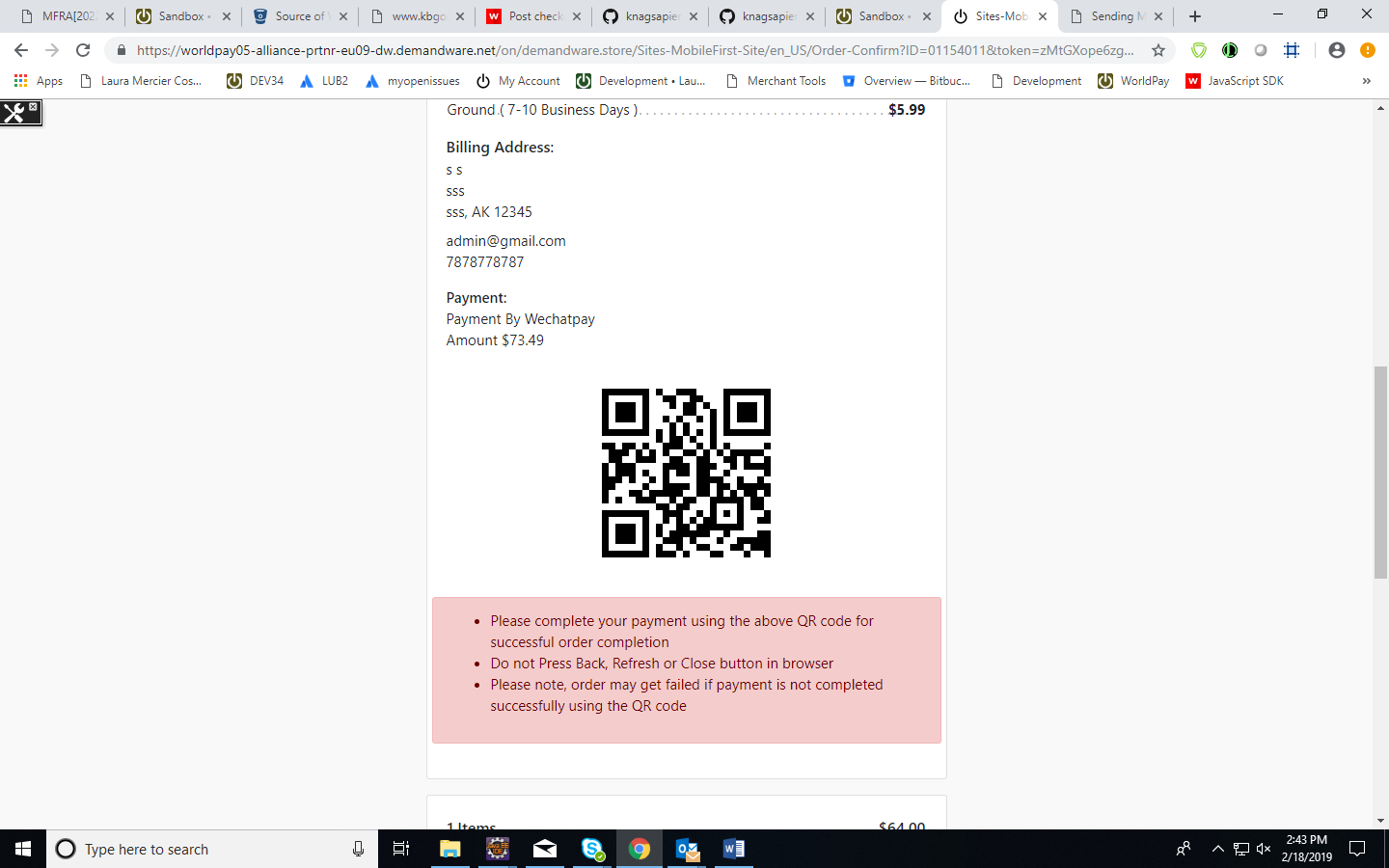
#### Response Attribute Mapping

<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE paymentService PUBLIC "-//WorldPay//DTD WorldPay PaymentService v1//EN"

"http://dtd.worldpay.com/paymentService\_v1.dtd">

<paymentService version="1.4" merchantCode="SAPIENTNITROECOM"><reply><orderStatus orderCode="01154011"><payment><paymentMethod>WECHATPAY-SSL</paymentMethod><amount value="7349" currencyCode="USD" exponent="2" debitCreditIndicator="credit"/><lastEvent>SENT\_FOR\_AUTHORISATION</lastEvent><riskScore value="-1"/></payment><qrCode><![CDATA[]]></qrCode></orderStatus></reply></paymentService>



NOTE: The order is automatically authorised after 15 seconds in the test environment.

For the Production environment, user needs to scan the QR code at their end from the wechat payment app to complete the order authorisation. You can refer the screenshot attached above.

### YANDEX

Yandex.Money is primarily an eWallet that shoppers based in the countries in the [Payment Method Properties table](http://support.worldpay.com/support/kb/gg/alternativepayments/apm063.html#Payment_Method_Properties) below use to make real-time online payments. There is an additional option available to shoppers wishing to pay by a credit or debit card linked to their Yandex.Money account.

|  |  |
| --- | --- |
| **Property** | **Description/Value** |
| **Payment Type** | eWallet/credit and debit card |
| **Countries** | Armenia, Azerbaijan, Belarus, Georgia, Kazakhstan, Kyrgyzstan, Moldova, Russia, Tajikistan, Turkmenistan, Ukraine and Uzbekistan |
| **Acceptance Currencies** | RUB and USD |
| **Mode** | DIRECT |

#### Request Attribute Mapping

<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE paymentService PUBLIC '-//Bibit//DTD Bibit PaymentService v1//EN' 'http://dtd.worldpay.com/paymentService\_v1.dtd'>

<paymentService merchantCode="SAPIENTNITROECOM" version="1.4">

<submit>

<order orderCode="Example\_Yandex\_Order\_Code\_0">

<description>Description here</description>

<amount currencyCode="RUB" value="5000" exponent="2" />

<orderContent>

<![CDATA[ ]]>

</orderContent>

<paymentDetails>

<YANDEXMONEY-SSL shopperCountryCode="RU">

<successURL>http://www.worldpay.com/?successURL</successURL>

<cancelURL>http://www.worldpay.com/?cancelURL</cancelURL>

<pendingURL>http://www.worldpay.com/?pendingURL</pendingURL>

</YANDEXMONEY-SSL>

</paymentDetails>

<shopper>

<shopperEmailAddress>shopper@worldpay.com</shopperEmailAddress>

</shopper>

</order>

</submit>

</paymentService>

#### Response Attribute Mapping

<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE paymentService PUBLIC "-//Worldpay//DTD Worldpay PaymentService v1//EN"

"http://dtd.worldpay.com/paymentService\_v1.dtd">

<paymentService version="1.4" merchantCode="SAPIENTNITROECOM">

<reply>

<orderStatus orderCode="Example\_Yandex\_Order\_Code\_0">

<reference id="3015839577">https://secure-test.worldpay.com/jsp/test/shopper/APMSTSimulator.jsp?customerRef=3013340527&amp;paymentMethod=YANDEXMONEY</reference>

</orderStatus>

</reply>

</paymentService>

### 4.6.20 Google Pay (Not available in Controller)

Update the site preferences named,

Google Pay Merchant ID and Google Pay Environment, for google pay integration.

|  |  |
| --- | --- |
| Example XML request Select  <?xml version="1.0" encoding="UTF-8"?>  <!DOCTYPE paymentService PUBLIC "-//WorldPay/DTD WorldPay PaymentService v1//EN"    "http://dtd.worldpay.com/paymentService\_v1.dtd">  <paymentService version="1.4" merchantCode="YOUR\_MERCHANT\_CODE">    <submit>      <order orderCode="YOUR\_ORDER\_CODE" shopperLanguageCode="en">        <description>GOOGLE\_PAY\_TEST\_ORDER</description>        <amount value="100" currencyCode="GBP" exponent="2"/>        <orderContent>          <![CDATA[]]>        </orderContent>        <paymentDetails>          <PAYWITHGOOGLE-SSL>            <protocolVersion>AAAAAA...</protocolVersion>            <signature>AAAAAA...</signature>            <signedMessage>AAAAAA...</signedMessage>          </PAYWITHGOOGLE-SSL>        </paymentDetails>        <shopper>          <shopperEmailAddress>shopper@worldpay.com</shopperEmailAddress>        </shopper>      </order>    </submit>  </paymentService> Example payload Here's an example of the payload within <PAYWITHGOOGLE-SSL>:  <PAYWITHGOOGLE-SSL>    <protocolVersion>ECv1</protocolVersion>    <signature>TJVA95OrM7E2cBab30RMHrHDcEfxjoYZgeFONFh7HgQ</signature>    <signedMessage>{"encryptedMessage":"qrnk5KIV1syX8xuaJYpjAxjvTmAfH4arrxK+5wT4XNsn+K/21QC5KyNyXVnaZsfz+FjYeZcZyTDsAXMGp01u9Psq3wp/9aJJyZ9PS9+G2sVJ+symppSvIHhPYxfcFLyvDxcU7J2ZUIU2o7ZAsHkc9eUL0zqcjTfRxxyladBQmP3mjy7OtIZbXodSr5vZZqKrRm0MqmPo/ISYor0gPmv+l+wRdLmCugdhBrg6mFGMuMBtnIjdtOZ+o4iSl4NqYo4BnI1py4GXG78FPHAlqU9Im+a9V6CtOsVG1eiLCn616fI8WorvPU46bDL7UApF/+vRF6lJviV9Ie9pUJHejO5Z17GPTfBuxO8ZAPWqcPY\u003d","ephemeralPublicKey":"BJ8xDzqkngvLUZcOAkqgkYpwGIHi2TzOXO6ZDJpKHPJCnJZ5WZOTO0HtNrTiOgiVc2gdZQ+TWmcy2Y1KnNsAGpg\u003d","tag":"5t5pipJgyhxJUg9XcymYcdcKmtBxQ71VvF2uhnempbs\u003d"}</signedMessage>  </PAYWITHGOOGLE-SSL> Response The response message is the same as our [standard XML responses](http://support.worldpay.com/support/kb/gg/corporate-gateway-guide/content/directintegration/paymentrequests.htm). |  |

# PRODUCTION SETUP



## PRODUCTION SERVICE SETUP

Steps for setting up the production service in business manager

1. In business manager navigate to “Administration > Operations > Services”
2. Click on “Credentials” Tab
3. Create new credentials by providing the merchant code, production URL, username and password
4. Now, navigate to “Merchant Tools > Site Preferences > Custom Preferences”
5. Inside “Worldpay” group “Merchant Code for Worldpay” as created in step 3. Also modify the other desired preferences on need basis
6. Worldpay Client Side Encryption Public Key
7. Worldpay MAC Secret Code
8. Worldpay Merchant Number
9. Worldpay Installation Id
10. In case if for separate merchant to be used at APM level, then service detail modification to be done in “Merchant Tools > Ordering > Payment Methods”

# OPERATIONS, MAINTENANCE



## DATA STORAGE

### Order Level Attributes

Some additional attributes are defined and stored in Order object. These are the custom attributes and will be sent as a part of Order XML, so that the same can be used by Order Management System for further processing.

|  |  |  |  |
| --- | --- | --- | --- |
| Sr. No. | Additional Custom Fields | Attribute Id | Description |
| 1 | Refusal/Decline Code | declineCode | This field capture the value of declined code from Worldpay response. |
| 2 | Risk Score | riskScore | This field capture the string sends by for Risk management or Risk Guardian. |
| 3 | Authorization ID | authID | This field captures the value of Authorization ID for Authorized Request. |
| 4 | Masked Card Number | cardNumer | This field captures the masked card number from Worldpay response. |
| 5 | CVC Result Code | cvcResultCode | This field capture CVV status code from response |
| 6 | AVS Result Code | avsResultCode | This field capture AVS status code from response |
| 7 | Transaction Status | transactionStatus | This field maintains the history of Order status. This is Array, so whenever order status will be changed transaction status will be updated by order notification. |
| 8 | AAV Postcode Result Code | aaVPostcodeResultCode | This field will be populate from Authorization response of AMEX card |
| 9 | AAV Address Result Code | aaVAddressResultCode | This field will be populate from Authorization response of AMEX card |
| 10 | Issuer Response | issuerResponse | Response from Issuer After 3D verification |
| 11 | AAV Email Result Code | aaVEmailResultCode | Email Result Code |
| 12 | AAV Telephone Result Code | aaVTelephoneResultCode | Telephone Result Code |
| 13 | SEPA Mandate ID | mandateID | Mandate ID created for SEPA transaction |
| 14 | Worldpay Last Event | WorldpayLastEvent | Last Event occurred on order at Worldpay |
| 15 | Worldpay MAC Missing | worldpayMACMissingVal | True when Mac error received in redirect Credit Card or APM response URL |
| 16 | AAV Cardholder Name Result Code | aaVCardholderNameResultCode | Card Holder Name Result Code |

### Order Payment Instrument Attributes

Apart from capturing the Order Attributes we also capture some Order Payment Instrument attributes. These are specific to the Payment Method and are captured in <payment> tag.

|  |  |  |
| --- | --- | --- |
| Sr. No. | Additional Custom Fields | Attribute Id |
| 1 | Bank | bank |
| 2 | Installments | installments  Note: Please contact Worldpay for further information on the max number of installments. |
| 3 | CPF | cpf |
| 4 | BankCode | bankCode |
| 5 | Worldpay Merchant ID | Merchant Hex Code whose order is placed |
| 6 | Worldpay Token Requested | Token requested value where selected for cards |

### Order Notification Custom Object

The attributes for custom Object ‘**OrderNotifyUpdates’** are defined in the below table. A new custom Object is created each time a notification is received with all these attributes and the custom object will be retained till the time it is deleted by the Notification job.

|  |  |  |
| --- | --- | --- |
| Sr. No. | Additional Custom Fields | Attribute Id |
| 1 | Order No | orderNo |
| 2 | Xml String | xmlString |
| 3 | Custom Object ID | ID |
| 4 | Time Stamp | timeStamp |

## 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

Worldpay to add content here

|  |  |  |
| --- | --- | --- |
| **Name** | **Email** | **Support Type** |
| Lochan Sim | Lochan.Sim@worldpay.com | Primary |
| Rhuta Patel | Rhuta.Patel@worldpay.com | Secondary |
| Jonathan Berry | jonathan.berry@worldpay.com | Secondary |

# USER GUIDE



## ROLES, RESPONSIBILITIES

Typically, most of the integration works is done by the backend developer. We expect that the person doing this integration is familiar with the web service, xml processing and has hands on experience with the SFCC platform

## BUSINESS MANAGER

***NA [As changes required for storefront has been already covered under setup section]***

## STOREFRONT FUNCTIONALITY

***NA [As changes required for storefront has been already covered under setup section]***

# TESTING



## TEST CASES

This Cartridge has gone through QA and UAT including live proving across all the payment methods. Please refer to the test cases attached here.





## TEST DATA

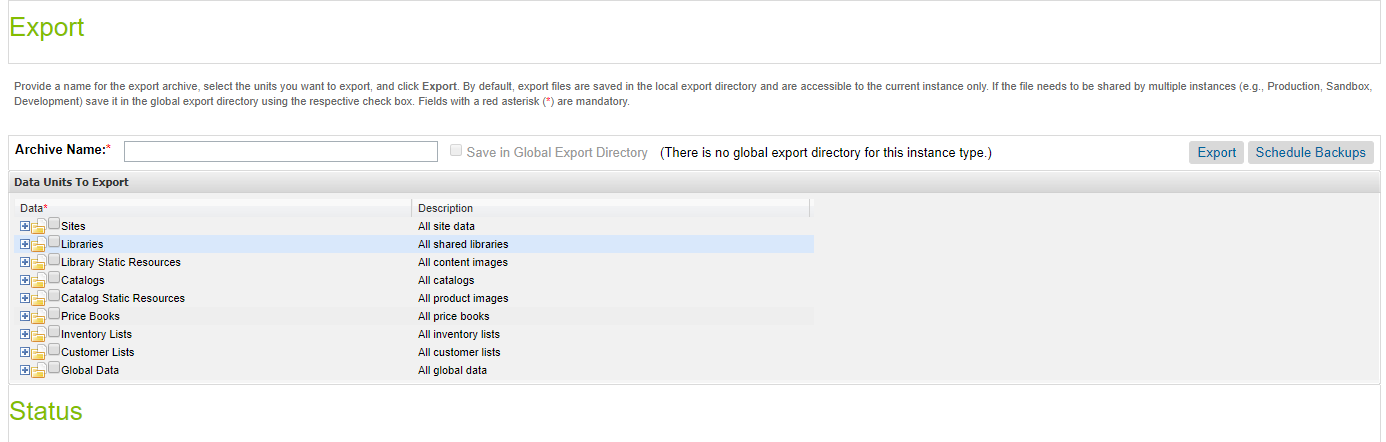
Refer to Appendix A for values of test card numbers.

## Automation test suit for functional testing

**Pre Requiste**

1. Redirect credit card payment should be in page format so remove any config did for iframe or light box in payment method.
2. We chat redirect and direct payment can be test one at a time. So while testing redirect mode make the line change around 22(describe => describe.skip) in wechatDirectPaymentMethod.js and vice versa on the file wechatRedirectPaymentMethod.js while testing the we chat direct mode.
3. Order confirmation page customer zipcode, customer country and customer phone number comparison can be done with the data configured in test/functional/mocks/customers.js. So we recommend to create one new customer for automation testing which having the address zip code ‘14304’ and country code ‘US’ and phone number ‘3333333333’. Save Visa card ‘4917610000000000’ for that customer.
4. Do Site export, the required xml files are: customer-list, catalogs, pricebooks, promotions

In Business Manager, Administration => Site Development => Site Import and Export



Select which one you want to take export of like Price Books, select all price books, give

Archive Name and click on export after the success status you can download that file/folder by clicking on download button.

1. Update the folder path and file name to be sync with the path given in the file

functional/mocks/testDataMgr/main.js.

1. For example

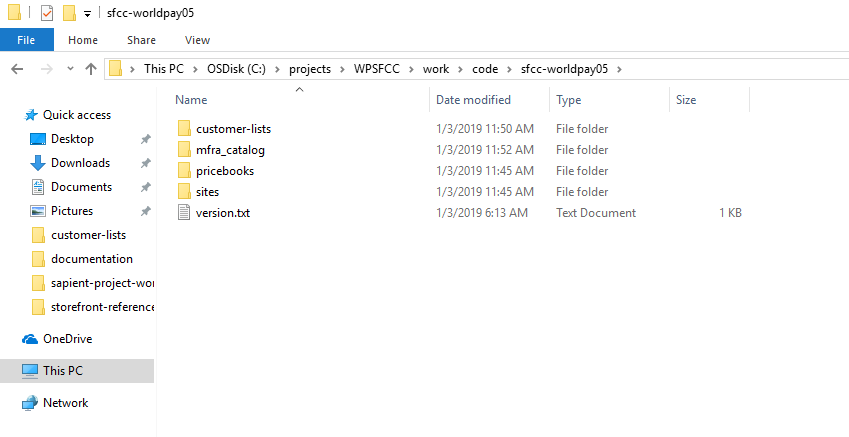
Customer xml path is mentioned in the file like this

files: [demoDataDir + '/customer-lists/customer.xml']

The same path should exist in the site export directory, if you site export direct contains the file path like this C:\projects\WPSFCC\work\code\sfcc\_worldpay\customer-lists\{siteid}.xml means you need to rename the file name and make the structure C:\projects\WPSFCC\work\code\sfcc\_worldpay\customer-lists\customer.xml like this.

So check the xml folder structure corresponding the path mentioned in the file.

Site export folder structure should like this



b. There are some issues in parsing customer-list, so in the customer lists customer xml file update the xml string ‘customer-list’ to ‘customers’. Or you can export all the customer directly from business manager and replace that xml in the ‘customer-lists’ folder.

1. In worldpay\_main.js, functional/mocks/testDataMgr/worldpay\_main, provide the required credentials which are existing in the site

a) loginUserEmail – existing user email(recommend user email in 3rd point)

b) loginPassword - existing user password

c) newUserPassword – new user registration password

d) storefrontPath – absolute folder path to the site export

e) resourcePath – absolute folder path to storefront reference architecture resource folder

f) 3 Variant Product ID like variantProduct1, variantProduct2, variantProduct3(example product ids which are having good inventory)

g) All credit card details scenario you want to check like creditCardAmex, creditCardDiscover, creditCardVisa, creditCardError, creditCardRefused, creditCard3D containing card name, card type, number, expiry date, cvv number.

Note:- Login user email should be available in the customer list xml file also.

1. If dw.json is not there in root folder, create one containing following :

{

"hostname": "worldpay03-tech-prtnr-eu04-dw.demandware.net",

"username": "USERNAME\_OF\_STOREFRONT",

"password": "PASSOWRD\_OF\_STOREFRONT",

"code-version": "CODE\_VERSION"

}

**Steps to configure and run the automation testing**

1. Run command npm install in root folder
2. Download selenium-standalone globally and start selenium server using command:-

npm install selenium-standalone@latest -g

selenium-standalone install

selenium-standalone start

1. To run functional testing, run command “selenium-standalone start” in one terminal and the command to run test “npm run test:functional” in another terminal. Functional automation testing will be started running each file individually in the folder ‘test/functional/checkout’

The test scenarios are,

* For Guest User :-

Direct Credit Card Payment Method – directcreditcardPaymentMethod.js

Redirect Credit Card Payment Method – redirectcreditcardPaymentMethod.js

Wechat Direct Payment Method – wechatDirectPaymentMethod.js

Wechat Redirect Payment Method – wechatRedirectPaymentMethod.js

Paypal Payment Method – paypalPaymentMethod.js

Refused Credit Card Payment Method – refusedCreditCardPaymentMethod.js

* For Registered User :-

Login and using Saved cards (User account should have saved cards for this)

Register Customer and Direct Credit Card Payment Method

Sepa Payment Method – sepaPaymentMethod.js

Alipay Payment Method – alipayPaymentMethod.js

3D Payment Method – 3dPaymentMethod.js

Mister cash Payment Method –mistercashPaymentMethod.js

Error Credit Card Payment Method – errorCreditCardPaymentMethod.js

To run single scenario/single file only change the specs line(around 88) in the file ‘test/functional/webdriver/wdio.conf.js’. For ex. To test only the savecard functionality change the line

specs: [

specs

],

To

specs: [

'test/functional/checkout/registeredUser/loginSavedCard.js'

],

Note:- You should be able to see count of testcases passing, failing and skipped ones. And also required order number in the console.

**Integration Testing**

1. Open terminal and run command “npm run test:integration” to run the test/integration files one by one and you will be able to see the required response using different payment options and for failed scenario, only error message is printed.

Note:- The response we get is of “CheckoutServices-PlaceOrder”.

1. The test scenarios are as follows:-

* 3DCreditCardPaymentMethod
* alipayPaymentMethod
* errorCreditCardPaymentMethod
* loginDirectCreditCardPaymentMethod
* loginSavedCardPaymentMethod
* mistercashPaymentMethod
* paypalPaymentMethod
* redirectCreditCardPaymentMethod
* sepaPaymentMethod
* wechatPaymentMethod

For integration testing :-

<https://medium.com/adobetech/how-to-combine-rest-api-calls-with-javascript-promises-in-node-js-or-openwhisk-d96cbc10f299>

**Unit Testing**

Open terminal and run command “npm run test” to run the test/unit files

# KNOWN ISSUES

* *Worldpay Notify URL and Worldpay notifyupdate URL are exposed as public; there is no security check available like MAC security check. As there is no such data sent from Worldpay to validate. To provide some level of security we have applied IP level validated to the URL. Merchants must ensure that they have applied proper security on these URL’s.*

# RELEASE HISTORY

|  |  |  |
| --- | --- | --- |
| **Version** | **Date** | **Changes** |
| 18.1.0 | 15-05-2018 | Initial Release –MFRA Compatible |

# APPENDIX A: APM/Card mapping keys

Supported Cards Mapping table:

|  |  |  |
| --- | --- | --- |
| **Card Name** | **Key Value** | **Test card number** |
| Airplus | AIRPLUS-SSL | 122000000000003 |
| American Express | AMEX-SSL | 34343434343434 |
| Dankort | [DANKORT-SSL](http://dtd.worldpay.com/v1/paymentService_v1.dtd.html#ElementDef161) | 5019717010103742 |
| Diners | DINERS-SSL | 36700102000000 |
| Discover card | DISCOVER-SSL | 6011000400000000 |
| JCB | JCB-SSL | 3528000700000000 |
| Laser | LASER-SSL | 630495060000000000 630490017740292441 |
| Maestro | MAESTRO-SSL | 6759649826438453, 6799990100000000019 |
| MasterCard | ECMC-SSL | 5555555555554444, 5454545454545454 |
| Visa | VISA-SSL | 4444333322221111,  4911830000000 |

Supported Payment Method Mapping table:

|  |  |
| --- | --- |
| **Payment Method Name** | **Key Value** |
| Konbini | KONBINI-SSL |
| Poli | POLI-SSL |
| Poli NZ | POLINZ-SSL |
| Przelewy24 | PRZELEWY-SSL |
| SEPA-DD | ELV-SSL |
| Alipay | ALIPAY-SSL |
| Boleto | BOLETO-SSL |
| CashU | CASHU-SSL |
| Credit Card – Direct | CREDIT\_CARD |
| Credit Card – Redirect | Worldpay |
| China Union Pay | CHINAUNIONPAY-SSL |
| ENETS | ENETS-SSL |
| Giropay | GIROPAY-SSL |
| IDEAL | IDEAL-SSL |
| Mistercash | MISTERCASH-SSL |
| Paypal | PAYPAL-EXPRESS |
| Sofort | SOFORT-SSL |
| Qiwi | QIWI-SSL |
| Yandex | YANDEXMONEY-SSL |

# APPENDIX B: IDEAL Bank List

|  |  |
| --- | --- |
| **Bank Name** | **Bank Code** |
| ABN | ABN\_AMRO |
| ASN | ASN |
| ING | ING |
| Knab | KNAB |
| Rabobank | RABOBANK |
| SNS | SNS |
| SNS Regio | SNS\_REGIO |
| Triodos | TRIODOS |
| Van Lanschot | VAN\_LANSCHOT |

# APPENDIX C: Error Codes and Error Messages for Transaction Notification

|  |  |
| --- | --- |
| **Master Error Code** | **Error Message** |
| 111 | XML Parse error has occurred |
| 112 | XML Corrupted, Could not find Order No. |
| 113 | XML in Custom Object Corrupted |
| 114 | Error occurred while deleting Custom Object |
| 115 | Error occurred while reading status history from Order Object |
| 116 | Error occurred while Updating Order Object |
| 117 | Error occurred while reading Custom Object |
| 118 | No Transaction History Available |
| 119 | Last status Not available as No Transaction History is Available |
| 120 | Wrong Order Number |

# APPENDIX D: Order Notification and SFCC Order status mapping

|  |  |
| --- | --- |
| **Order Notifications Received From Worldpay** | **Changes in Order object in Business Manager** |
| **AUTHORISED** | Order Status: NEW                  Payment Status: no change                  Export Status: READY FOR EXPORT                  Confirmation Status : CONFIRMED |
| **CANCELLED** | Order Status: FAILED/CANCELLED  Payment Status: NOT PAID                  Export Status: NOT EXPORTED  Confirmation Status : NOT CONFIRMED |
| **CAPTURED** | Order Status: COMPLETED                  Payment Status: PAID                  Export Status:  no change                  Confirmation Status :  no change |
| **SENT\_FOR \_REFUND** | Order Status:  no change                  Payment Status:  no change                  Export Status: no change                  Confirmation Status :  no change |
| **REFUSED** | Order Status: FAILED  Payment Status: NOT PAID                   Export Status: NOT EXPORTED  Confirmation Status : NOT CONFIRMED |
| **SETTLED** | Order Status: no change                  Payment Status: no change                  Export Status:  no change                  Confirmation Status :  no change |
| **INFORMATION\_REQUESTED** | Order Status:  no change                  Payment Status:  no change                  Export Status: no change                  Confirmation Status : no change |
| **CHARGED\_BACK** | Order Status:  no change                  Payment Status:  no change                  Export Status: no change                  Confirmation Status : no change |
| **EXPIRED** | Order Status:   FAILED                  Payment Status:   NOT PAID                  Export Status:   NOT EXPORTED           Confirmation Status :  NOT CONFIRMED |

# APPENDIX E: Error Codes and Error Messages

|  |  |
| --- | --- |
| **Master Error Code** | **Error Message** |
| worldpay.error.code1 | Internal error has occurred , please choose a different Payment Method or try again later. |
| worldpay.error.code2 | Parse error has occurred, please choose a different Payment Method or try again later. |
| worldpay.error.code3 | Invalid amount error, please choose a different Payment Method or try again later. |
| worldpay.error.code4 | Security error , please choose a different Payment Method or try again later. |
| worldpay.error.code5 | Invalid request, please choose a different Payment Method or try again later. |
| worldpay.error.code6 | Please choose a different Payment Method or try again later. |
| worldpay.error.code7 | Payment details in the order element are incorrect, please choose a different Payment Method or try again later. |
| worldpay.error.code8 | Submission error , please choose a different Payment Method or try again later. |
| worldpay.error.generalerror | Please choose a different Payment Method or try again later. |
| worldpay.error.cancelerror | Your transaction has been cancelled. |

# APPENDIX F: Enabling and Disabling worldpay integration

1. Go to Merchant tools > Ordering > Payment Methods

2. Disable all the worldpay specific payment methods

3. Also cross check one by one for the type of processor, the payment method is using (To check it, click on any of the payment method lets say, CREDIT\_CARD, and see the Payment Pocessor). If it is worldpay, change it to some othe desired processor( eg. For CREDIT\_CARD, BASIC\_CREDIT).

4. Enable the payment methods we want to use with the type of processor.

5. To enable it again change the processor type to worldapy.

6. Go to [Merchant Tools](https://worldpay02-tech-prtnr-eu04-dw.demandware.net/on/demandware.store/Sites-Site/default/ViewApplication-ExpandMenuGroup?MenuGroupID=SiteMenu&OverviewPage=SiteNavigationBar-ShowSiteOverview&csrf_token=J3jXHp4dwSBWpuaRTzOUTrs6vGohZn7ad2s3-pYNg6YDwoYT6ntqxJZdivSIfyCzzKwcICillPMHO-KHy-gvqE_nIve3vxlhtmUH1QW5V-uxdyw0UfzUHLmc3MFCB1kuhT-UwLvUA3pd27IKSC-A4qrF8ngGXYE3-077CND_5AalPq5yoY0) > [Site Preferences](https://worldpay02-tech-prtnr-eu04-dw.demandware.net/on/demandware.store/Sites-Site/default/SiteNavigationBar-ShowMenuitemOverview?CurrentMenuItemId=site-prefs&itemType=Site&csrf_token=J3jXHp4dwSBWpuaRTzOUTrs6vGohZn7ad2s3-pYNg6YDwoYT6ntqxJZdivSIfyCzzKwcICillPMHO-KHy-gvqE_nIve3vxlhtmUH1QW5V-uxdyw0UfzUHLmc3MFCB1kuhT-UwLvUA3pd27IKSC-A4qrF8ngGXYE3-077CND_5AalPq5yoY0) > custom Preferences > worldpay and disable ‘Enable Apm look up service’.