**WorldPay Integration**

Version 17.1.0



[1. SUMMARY 1-4](#_Toc479577978)

[2. COMPONENT OVERVIEW 2-5](#_Toc479577979)

[2.1 FUNCTIONAL OVERVIEW 2-5](#_Toc479577980)

[2.1.1 Payment Methods 2-5](#_Toc479577981)

[2.1.2 Order Notifications 2-6](#_Toc479577982)

[2.1.3 APM Lookup Service 2-13](#_Toc479577983)

[2.1.4 Multiple Merchant ID Support 2-13](#_Toc479577984)

[2.1.5 Brazil Integration 2-14](#_Toc479577985)

[2.1.6 Order CleanUp Batch Job 2-14](#_Toc479577986)

[2.1.7 Initiate Cancel Order Job 2-15](#_Toc479577987)

[2.2 USE CASES 2-17](#_Toc479577988)

[2.3 LIMITATIONS, CONSTRAINTS 2-17](#_Toc479577989)

[2.4 COMPATIBILITY 2-19](#_Toc479577990)

[3. IMPLEMENTATION GUIDE 3-20](#_Toc479577991)

[3.1 SETUP 3-20](#_Toc479577992)

[3.2 INSTALLATION 3-21](#_Toc479577993)

[3.2.1 Adding the Cartridges in Demandware Studio 3-21](#_Toc479577994)

[3.2.2 Activating the Cartridges in Business Manager 3-22](#_Toc479577995)

[3.3 CONFIGURATION 3-23](#_Toc479577996)

[3.3.1 Importing Meta Data 3-23](#_Toc479577997)

[3.3.2 Setting WorldPay Custom Site Preferences 3-24](#_Toc479577998)

[3.3.3 Create WorldPay Payment Processor 3-27](#_Toc479577999)

[3.3.4 Importing WorldPay Payment Methods 3-27](#_Toc479578000)

[3.3.5 Importing Job 3-30](#_Toc479578001)

[3.3.6 Importing Content Asset 3-30](#_Toc479578002)

[3.3.7 Importing Service 3-31](#_Toc479578003)

[3.3.8 IDEAL APM Integration 3-32](#_Toc479578004)

[3.3.9 Multiple Merchant ID Support Integration 3-33](#_Toc479578005)

[3.3.10 Look and Feel Customization for WorldPay Redirect Pages 3-33](#_Toc479578006)

[3.3.11 Hosted Payment Pages iframe or lightbox for WorldPay Redirect Payment Method 3-34](#_Toc479578007)

[3.4 CUSTOM CODE FOR CONTROLLERS 3-36](#_Toc479578008)

[3.4.1 Generic Section 3-36](#_Toc479578009)

[3.4.2 APM Lookup Support 3-55](#_Toc479578019)

[3.4.3 Brazil Integration 3-60](#_Toc479578020)

[3.4.4 Street Number Implementation 3-61](#_Toc479578021)

[3.4.5 Error Messaging 3-61](#_Toc479578022)

[3.4.6 Client Side Encryption 3-61](#_Toc479578023)

[3.4.7 Form File Changes 3-62](#_Toc479578024)

[3.5 EXTERNAL INTERFACES 3-63](#_Toc479578025)

[3.5.1 CHINA UNION PAY 3-63](#_Toc479578026)

[3.5.2 SOFORT 3-64](#_Toc479578027)

[3.5.3 ALIPAY 3-65](#_Toc479578028)

[3.5.4 YANDEX 3-66](#_Toc479578029)

[3.5.5 QIWI 3-68](#_Toc479578030)

[3.5.6 BOLETO 3-69](#_Toc479578031)

[3.5.7 MISTER CASH 3-71](#_Toc479578032)

[3.5.8 PAYPAL 3-72](#_Toc479578033)

[3.5.9 eNETS 3-73](#_Toc479578034)

[3.5.10 iDEAL 3-74](#_Toc479578035)

[3.5.11 GiroPay 3-76](#_Toc479578036)

[3.5.12 Konbini 3-77](#_Toc479578037)

[3.5.13 Poli 3-80](#_Toc479578038)

[3.5.14 Nordea 3-84](#_Toc479578039)

[3.5.15 ELV 3-86](#_Toc479578040)

[3.5.16 Przeleway24 3-88](#_Toc479578041)

[3.5.17 CashU 3-90](#_Toc479578042)

[4. OPERATIONS, MAINTENANCE 4-93](#_Toc479578043)

[4.1 DATA STORAGE 4-93](#_Toc479578044)

[4.1.1 Order Level Attributes 4-93](#_Toc479578045)

[4.1.2 Order Payment Instrument Attributes 4-94](#_Toc479578046)

[4.1.3 Order Notification Custom Object 4-94](#_Toc479578047)

[4.2 AVAILABILITY 4-95](#_Toc479578048)

[4.3 SUPPORT 4-95](#_Toc479578049)

[5. USER GUIDE 5-96](#_Toc479578050)

[5.1 ROLES, RESPONSIBILITIES 5-96](#_Toc479578051)

[5.2 BUSINESS MANAGER 5-96](#_Toc479578052)

[5.3 STOREFRONT FUNCTIONALITY 5-96](#_Toc479578053)

[6. TESTING 6-97](#_Toc479578054)

[6.1 TEST CASES 6-97](#_Toc479578055)

[6.2 TEST DATA 6-97](#_Toc479578056)

[7. KNOWN ISSUES 7-97](#_Toc479578057)

[8. RELEASE HISTORY 8-98](#_Toc479578058)

[APPENDIX A: APM/Card mapping keys 8-99](#_Toc479578059)

[APPENDIX B: IDEAL Bank List 8-101](#_Toc479578060)

[APPENDIX C: Error Codes and Error Messages 8-101](#_Toc479578061)

[APPENDIX D: Order Notification and Demandware Order status mapping 8-102](#_Toc479578062)

# SUMMARY

This document provides technical overview and implementation details for each WorldPay service integrated within Demandware platform with SiteGenesis controller version. The WorldPay cartridges (int\_worldpay and int\_worldpay\_controllers) extends the functionality of Demandware Storefront, enabling synchronous and asynchronous access to WorldPay eCommerce 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. Nordea
13. Poli
14. Konbini
15. Przelewy24
16. 3D/Non 3D Credit Cards (XML Direct with client side encryption and tokenzation features)
17. 3D/Non 3D Credit Cards (XML Redirect with tokenzation and hosted payment pages lightbox/iframe feature)

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

# 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. Konbini
12. Przelewy24
13. Credit Card- Direct (3D/Non 3D)

#### 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. Nordea
4. Credit Card – Redirect (3D/Non 3D)

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

### Order Notifications

The int\_worldpay or int\_worldpay\_controllers cartridge provides a service to update all the Demandware Order status according to status changes of individual payments in WorldPay.

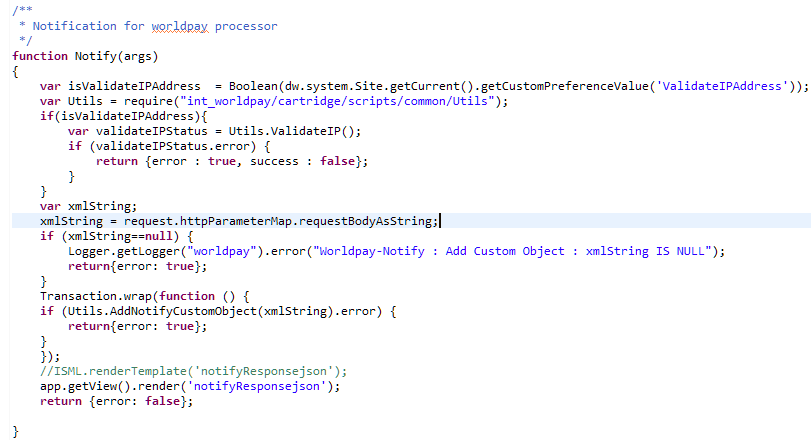
**WorldPay-Notify** is the secure controller function exposed to WorldPay to receive Order Notifications i.e. a HTTPS message whenever a payment status changed at WorldPay. Following URL (**https://mechantsandbox/WorldPay-Notify**) has been exposed to WorldPay to send the notifications. Demandware will 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 Demandware 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:

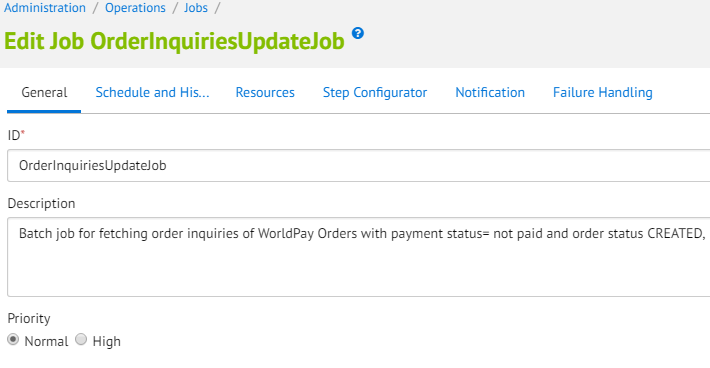
* AUTHORISED
* CANCELLED
* CAPTURED
* EXPIRED
* SENT\_FOR\_REFUND
* REFUSED
* SETTLED
* INFORMATION\_REQUESTED
* CHARGED\_BACK
* POST AUTH CANCELLED

Below is the screenshot for the Order Notification Controller

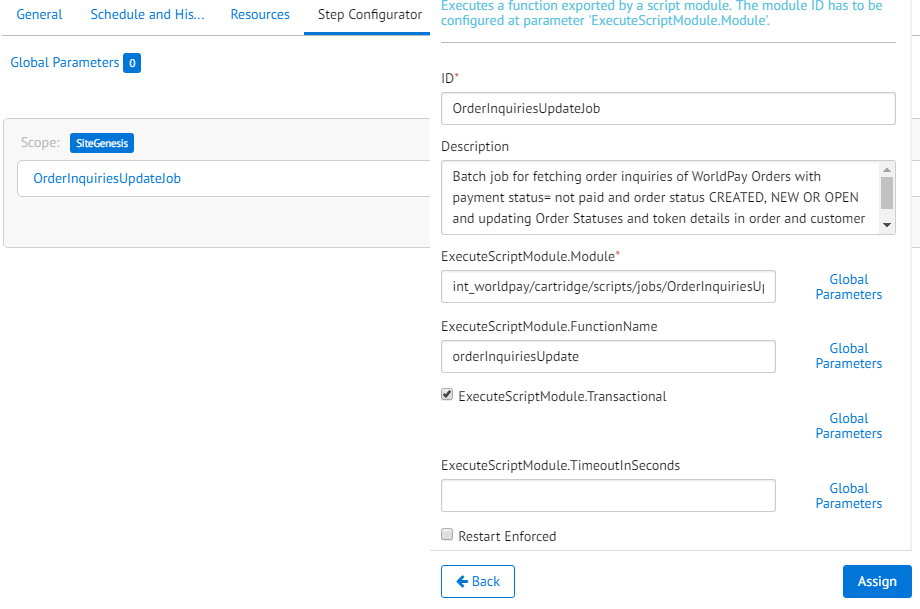


#### Order Enquiry job

The **OrderEnquiryUpdatesJob** 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 by the creation time. The job iterate the orders and make order enquiry 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

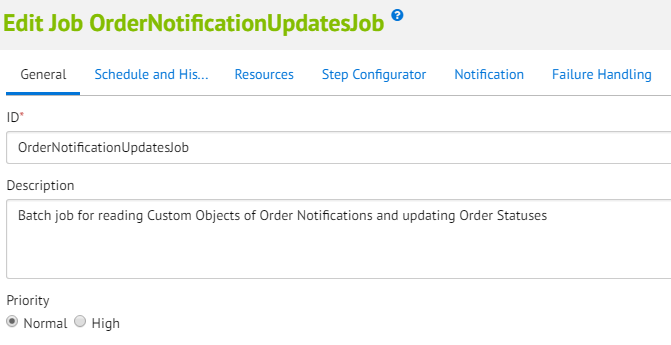


Refer below Step configuration section in order to execute script and their functions

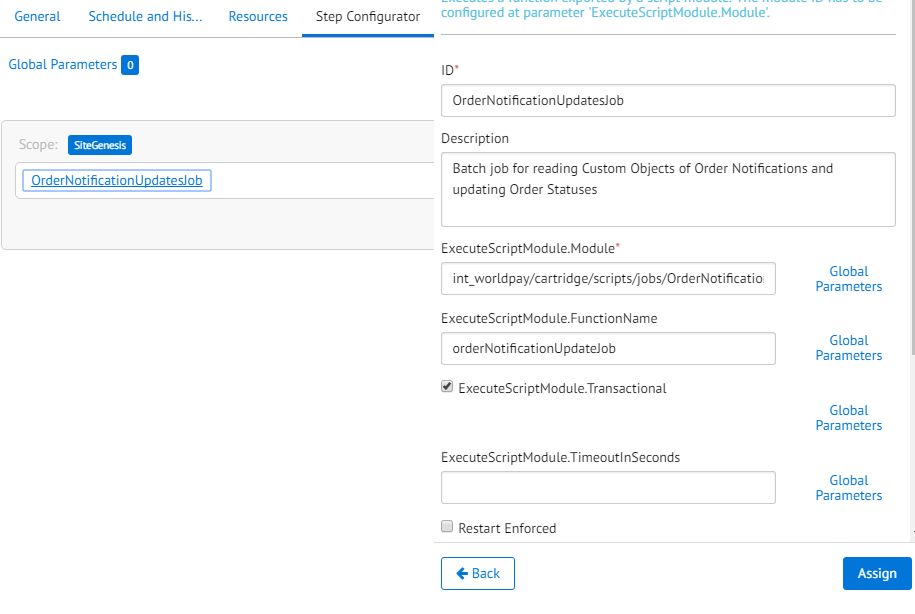


#### Order Notification job

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 Demandware.



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



Please refer to Appendix E for mapping of different status on receipt of different order notification from WP into BM.

#### 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 Demandware 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 set **true** by checking the checkbox. Also, the below site preferences needs to be set

|  |  |  |  |
| --- | --- | --- | --- |
| |  | | --- | | **EnableJobMailerService:** | | |  | | --- | |  | |
| |  | | --- | | **NotifyJobMailCC:** | | |  | | --- | |  | |
| |  | | --- | | **NotifyJobMailFrom:** | | |  | | --- | |  | |
| |  | | --- | | **NotifyJobMailTo:** | | |  | | --- | |  | |

#### Error Handling For Order Notification Job

Please refer toAppendix C for error codes and error messages for Order Notifications

#### Notification update Service

[The Cartridge provides the Merchant a service using which the merchant can fetch the history of the Payment Transaction change (Order Notifications) received by particular Order.](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 controller 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.)

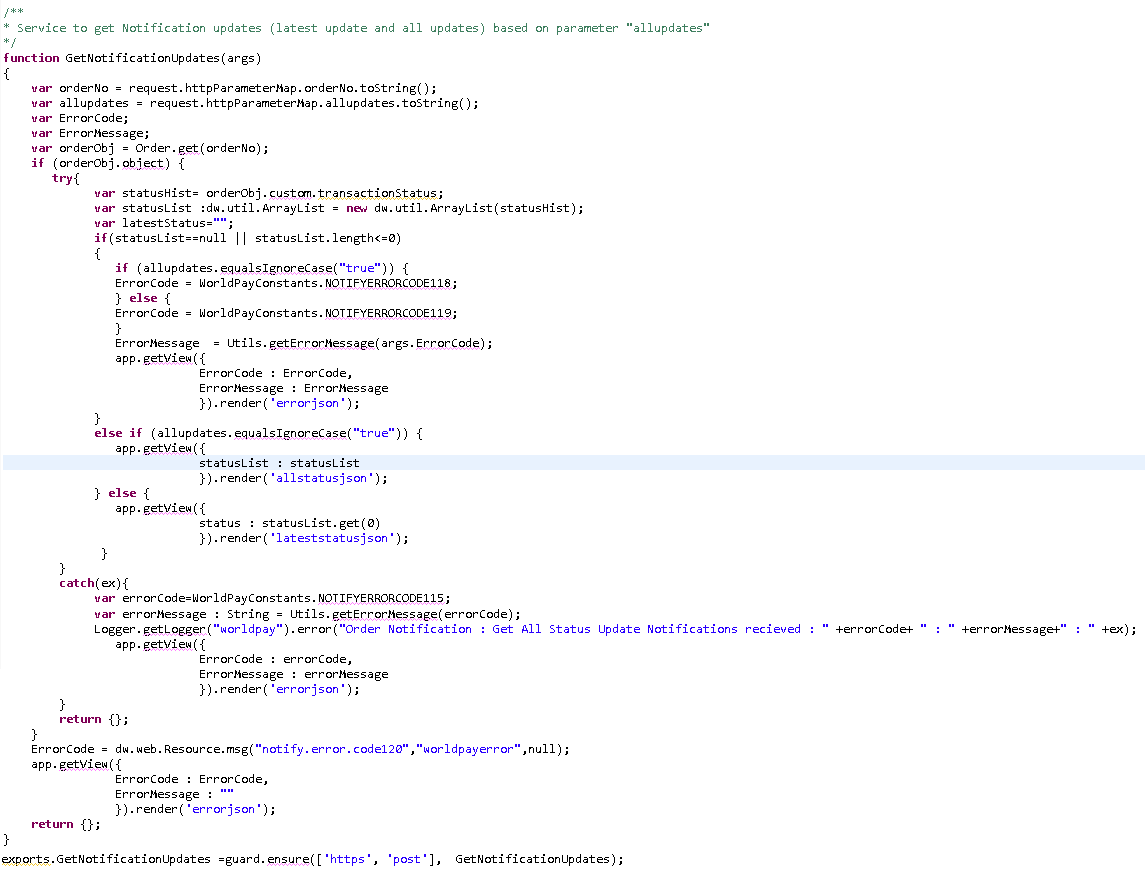
This service takes 2 parameters as Http Parameters,

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’ ie 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)"}]

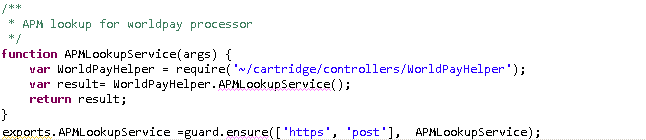
}

### APM Lookup Service

To enable APM look up service, the site preference **EnableAPMLookUpService** needs to be set to true.

[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. ]

|  |  |  |  |
| --- | --- | --- | --- |
| |  | | --- | | **EnableAPMLookUpService:** | | |  | | --- | |  | |
|  |  |

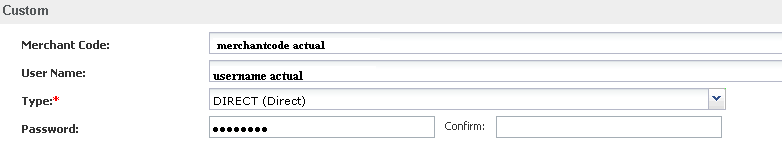


### Multiple Merchant ID Support

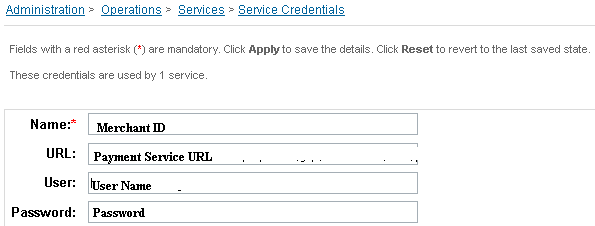
WorldPay cartridge also supports a mechanism to provide separate Merchant ID’s at APM level. This is primarily required for Merchants that need to configure different Merchant ID for financial purpose or if a specific APM is of a different version (to support both V1 and V3 APM’s on a single Merchant 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 over merchant ID in site preference, username and password from service credentials. If they are not configured at payment method level then merchant ID, 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

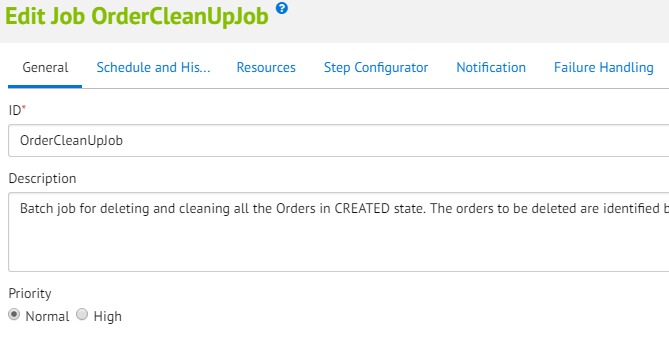


### Brazil Integration

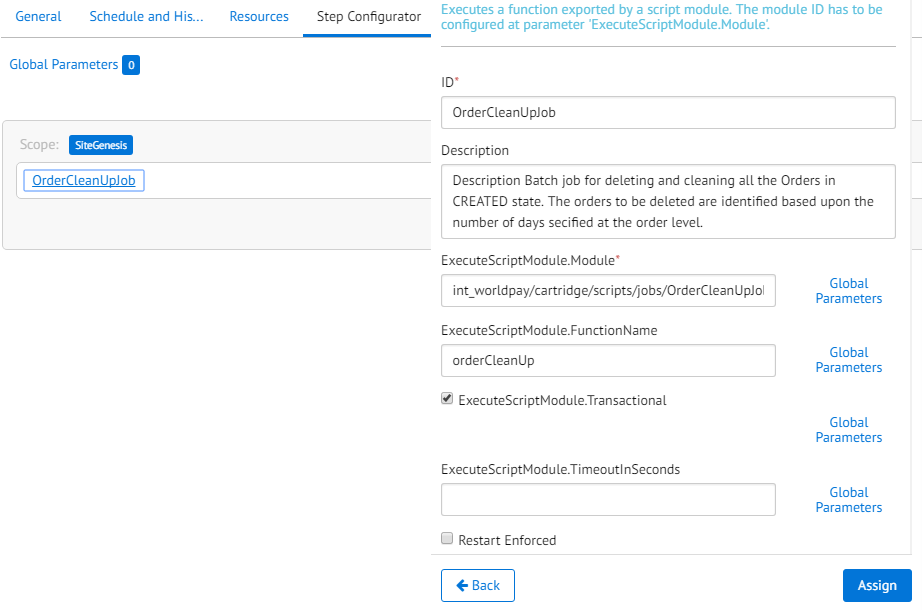
The WorldPay cartridge provides support to Credit Cards and supported APM’s (BOLETO) in Brazil. The implementation ensures that all the additional information (Example: CPF/CNJP or Installment Number) that are required by the payment service provider in Brazil are also captured. The implementation details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

### Order CleanUp Batch Job

The WorldPay cartridge provides support to a batch job that will mark all the orders which are in **CREATED** status and are aged to a defined time interval. The time interval field is Business Manager configurable and can be defined per 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 will be rolled back. Job can be Enable/Disable from Business Manager itself. A notification email shall be sent to BM configured recipients after job run with Job run status.

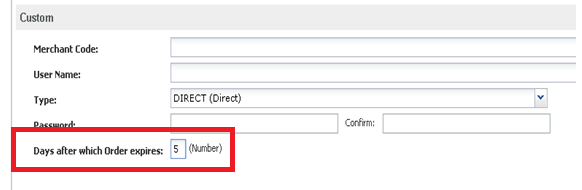


Refer below Step configuration section in order to execute script and their functions



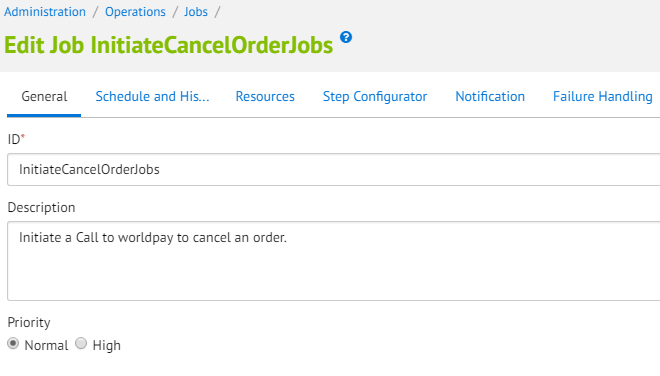
Note: Ensure before running batch job every payment method in site has attribute “Days after which order expires” value configured.

Business Manager Configuration field:

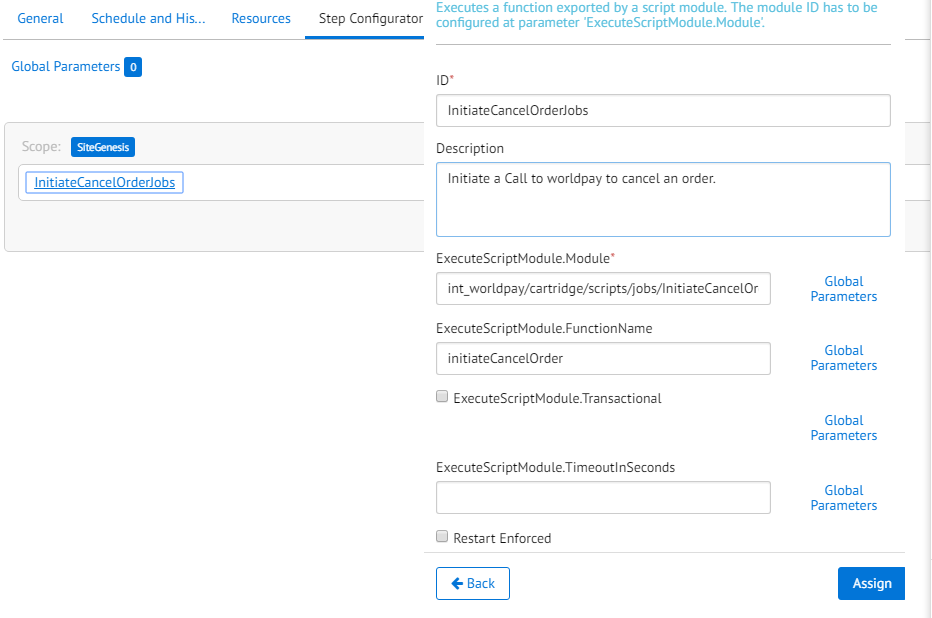


### Initiate Cancel Order Job

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 is defined in the custom preference World Pay 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 WorlPay 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. A notification email shall 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



## USE CASES

Please see the attached file for the use case scenarios.



## LIMITATIONS, CONSTRAINTS

This section details the limitations, constraints and the best practice for merchants planning to integrate the WorldPay cartridge with Demandware 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. Demandware 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 Demandware 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 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 are 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 Demandware 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

## COMPATIBILITY

This cartridge is integrated and tested with sitegenesis code base 17.3 and compatibility mode 16.2 of Demandware.

# IMPLEMENTATION GUIDE

## SETUP

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

**CARTRIDGE STRUCTURE**Controllers:

|  |  |
| --- | --- |
| **CONTROLLER NAME** | **DESCRIPTION** |
| WorldPay | Main controller that acts as the external interface. |
| WorldPayHelper | Helper controller that supports and performs additional tasks required to complete |

* package.json – initialize hook or service

Scripts:

* init/httpServiceInit.ds
* lib/LibCreateRequest.ds
* lib/ verifyMAC.ds
* checkout/CreatePaymentInstrument.ds
* checkout/ InvalidatePaymentCardFormElements.ds
* checkout/ RemovePaymentInstrument.ds
* checkout/UpdatePaymentInstrument.ds
* common/CreateRequestHelper.ds
* common/PaymentInstrumentUtils.ds
* common/Utils.ds
* common/WorldPayConstants.ds
* object/ResponseData.ds
* object/ WorldPayPreferences.ds
* order/AuthorizeOrder.ds
* order/UpdateorderStatus.ds
* order/UpdateTransactionStatus.ds
* pipelets/AddNotifyCustomObject.ds
* pipelets/APMLookUpService.ds
* pipelets/CheckWorldPayOrder.ds
* pipelets/GenarateErrorMessageForJob.ds
* pipelets/GetAllStatusUpdates.ds
* pipelets/GetCreatedOrderDetails.ds
* pipelets/GetLastestStatus.ds
* pipelets/GetOrderStatus.ds
* pipelets/GetPaymentCardTokends
* pipelets/ InitiateCancelOrder.ds
* pipelets/ReadNotifyCustomObject.ds
* pipelets/RemoveNotifyCustomObject.ds
* pipelets/SendCCAuthorizeRequest.ds
* pipelets/SendSecondAuthorizeRequest.ds
* pipelets/ SendWorldPayOrderInquiriesRequest.ds
* pipelets/ ValidateIPAddress.ds.
* pipelets/WriteToNotifyLog.
* Service/ServiceFacade
* jobs/ InitiateCancelOrderJob.ds
* jobs/ OrderCleanUpJob.ds
* jobs/ OrderInquiriesUpdateJob.ds
* jobs/ OrderNotificationUpdatesJob.ds
* jobs/ WorldPayJobs.ds
* hooks.json

Templates:

* default/allstatusjson.isml
* default/apmlookupjson.isml
* default/emailtemplateforjob.isml
* default/errorjson.isml
* default/lateststatusjson.isml
* default/notifyResponsejson.isml
* default/worldpayissuerredirect.isml
* default/worldpayorderdetails.isml
* default/worldpayredirect.isml
* default/worldpaypaymentmethods.isml
* default/worldpaysummary.isml

Properties:

* resources/worldpayerror.properties
* resources/worldpay.properties

Controllers:

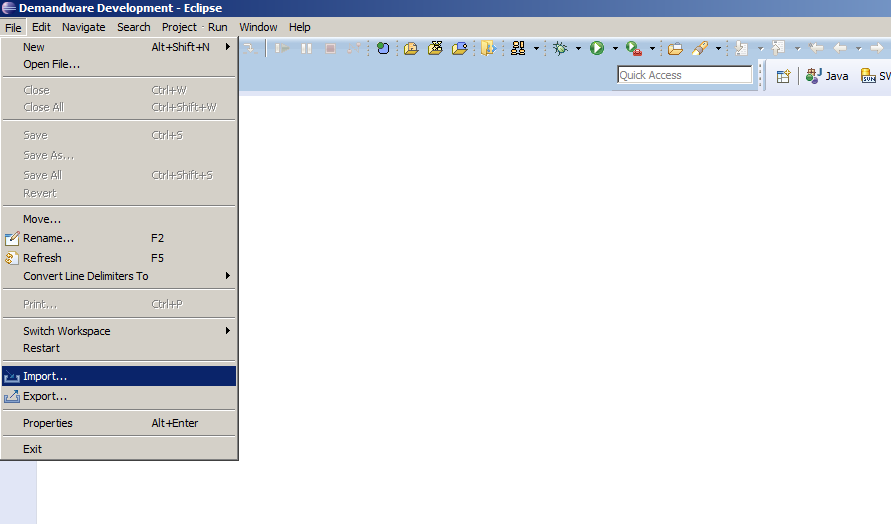
* WorldPay
* WorldPayHelper

## INSTALLATION

### Adding the Cartridges in Demandware Studio

To upload the cartridges into the Demandware server you first need to add the cartridges into Demandware studio. In order to do this, follow these instructions:

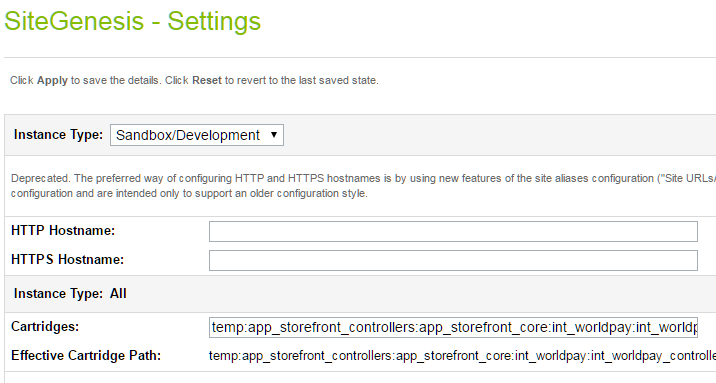
1. In Demandware UXStudio select in the menu *File 🡪 Import*.
2. In the import dialog select *General 🡪 Existing projects in the workspace* and click next
3. Ensure *Select archive file* is selected and select the compressed cartridge file by clicking on the *Browse* button.
4. Click *Finish* to import the cartridge.
5. Studio will now ask you if you want to link the cartridge to your active Demandware server connection. Click on y*es* or manually link the cartridge to your server by checking the project under project references in the server connection properties.



### Activating the Cartridges in Business Manager

Before the WorldPay functionality can become available to SiteGenesis, 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* append to the end “**:int\_worldpay**” and if controllers required then append “**:int\_worldpay\_controllers**” before Int\_worldpay cartridge inclusion
5. Click *Apply*.
6. To activate the cartridge for the Sandbox/Development/Production 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* append to the end: “**:int\_worldpay**”.



## 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: -

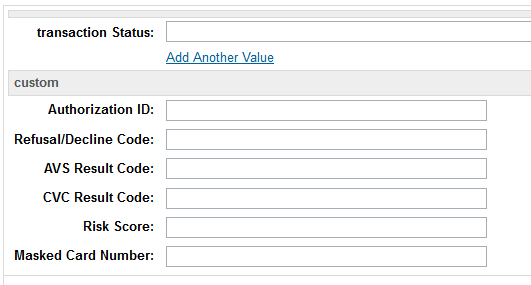


Fig: 2.1

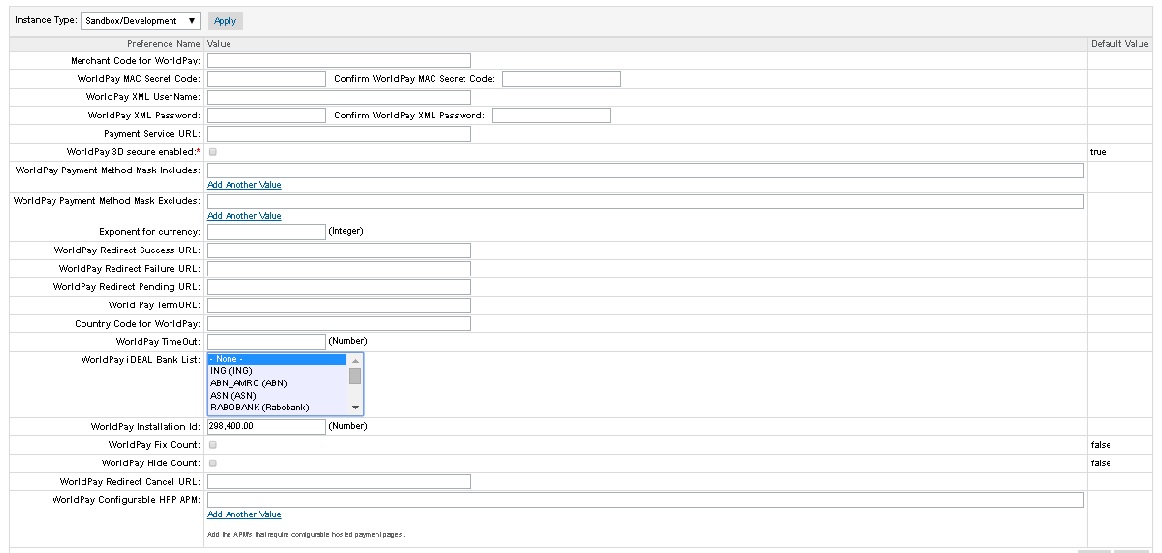


Fig: 2.2

### Setting WorldPay Custom Site Preferences

In Business Manager, navigate to the *SiteGenesis Site* 🡪 *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 12. In addition to the ones supplied by WorldPay fill out the following properties with the values listed against them:

* WorldPay SuccessURL: COPlaceOrder-Submit
* WorldPay PendingURL: COPlaceOrder-Submit
* WorldPay FailureURL: COPlaceOrder-Submit
* WorldPay CancelURL: COPlaceOrder-Submit
* WorldPay TermURL: WorldPay-HandleAuthenticationResponse

\*\* The fields highlighted in blue in the above 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 should also be configured in Worldpay Merchant Interface. | Merchant Specific |
| **WorldPay Payment Method Mask Includes** | This field represents all the Cards and APM’s that are supported and will be shown on Hosted Payment Page. | Merchant Specific - Add the Cards /APMs that are to be 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 |
| **World Pay 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 Merhcant 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 | 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 World Pay, used to calculate mac value of an order | Merchant Specific |
| **World Pay 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 ELV. | Merchant Specific - Provided by WorldPay |
| **WorldPay ELV Mandate Type** | This field represents the Mandate type to be sent in the ELV request. (Default value ONE-OFF) | Merchant Specific – This represents the mandate type to be set for ELV request. The merchant can choose to set it either as ONE-OFF or RECURRING. |
| **World Pay 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 |
| **World Pay 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) |

### 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 *SiteGenesis Site* *🡪 Ordering 🡪 Payment Processors*.
2. Click on the new button to create a new Payment Processor.
3. Enter as ID: **WorldPay** and then click the apply button.

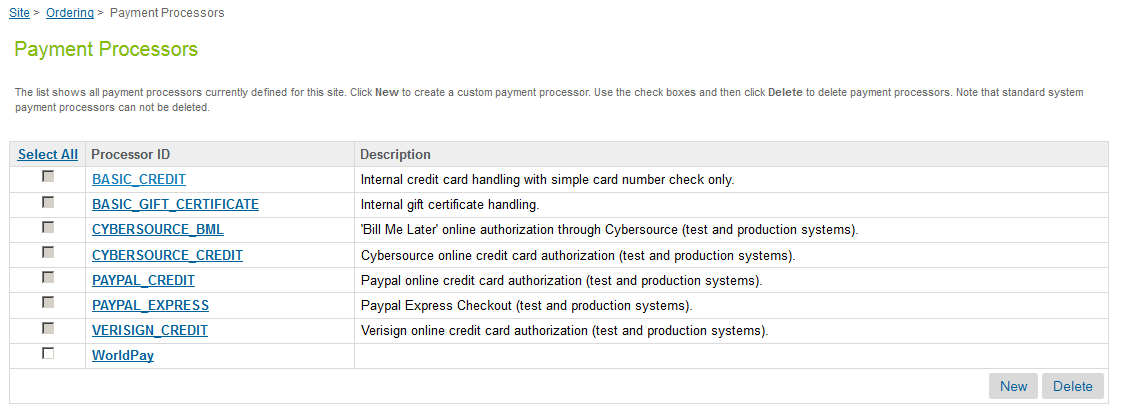


Fig: 2.3

### Importing WorldPay Payment Methods

In the integration package, a payment method definition is provided in the payment\_method.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 *SiteGenesis Site* *🡪 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 SiteGenesis Site 🡪 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 |
| ELV mandate number | This will be part of mandate type for ELV. |
| 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 |

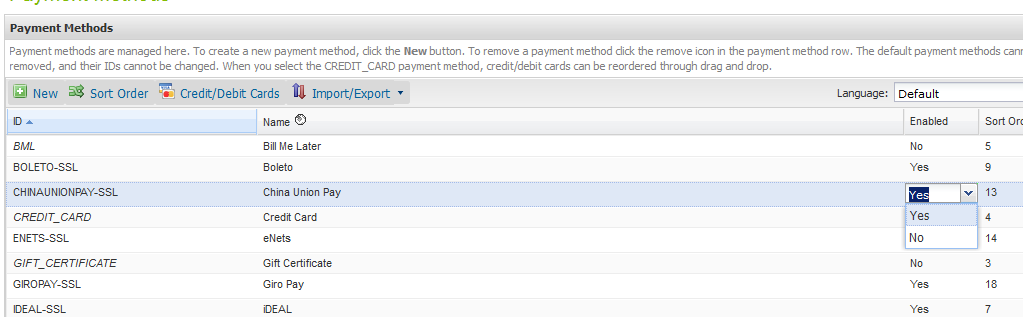
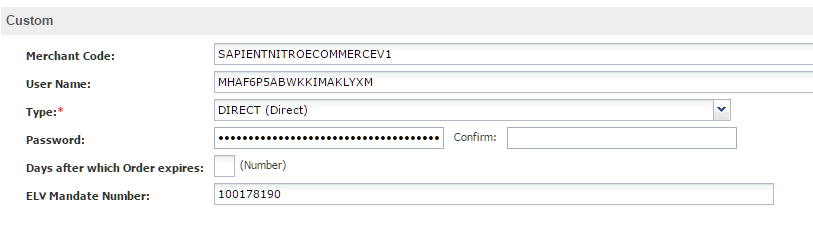


Fig: 2.4

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 |
| World Pay Card Type | This field represents the card type in WorldPay like for VISA it is “VISA-SSL” |

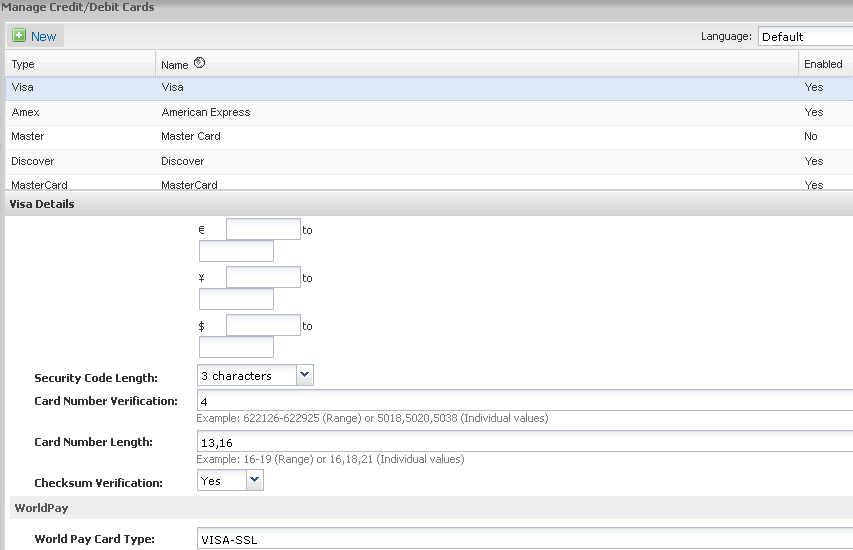
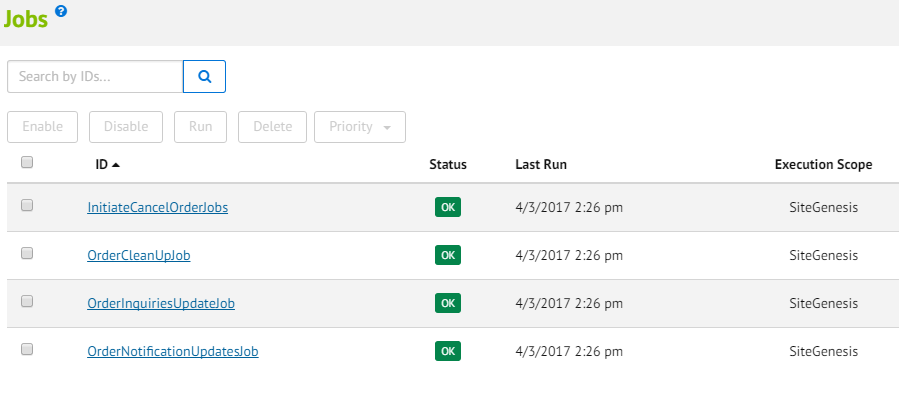


Fig: 2.6

### Importing Job

In the integration package, a job definition is provided in the worldpay\_jobs.xml file. All import files 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 *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 MERGE is selected and click on the Import button.



Enable the jobs that need to be activated, apply the configurations setting of scheduling into **General** and **Sites** sections one by one.

### Importing Content Asset

All import files can be found 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 *SiteGenesis Site* *🡪 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. Make sure REPLACE is selected and click on the Import button.
7. Navigate to *SiteGenesis Site* *🡪 Content🡪Content Asset.*
8. Confirm that the Content Asset with ID “worldpayhelper” and “worldpay-elv-consent” has been 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. Make sure MERGE is selected and click on the Import button.

Once the service is created successfully, fill out the following properties under World Pay Profile with the values listed against them:

|  |  |
| --- | --- |
| **World Pay 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 World Pay credentials with the values listed against them:

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

|  |  |
| --- | --- |
| **World Pay 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. |

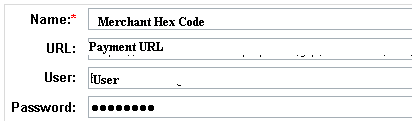


Fig 2.9

### IDEAL APM Integration

Bank configurations in Business Manager:

All the supported Bank List values are preconfigured. However, the list can be Modified/Deleted 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)

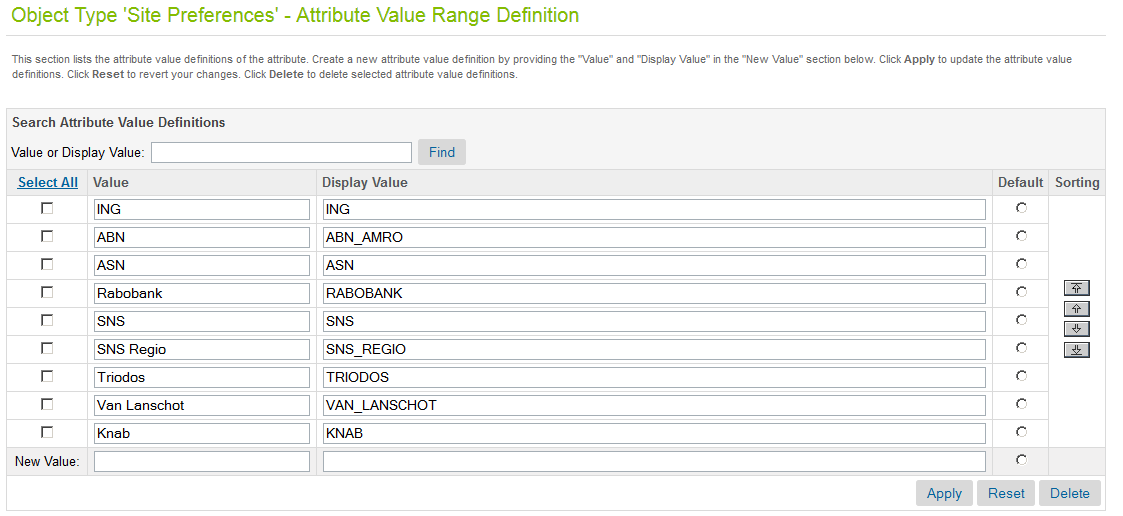


Fig: 2.6

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 *Site Preferences 🡪 Custom Preferences 🡪 WorldPay*
4. Select the banks that are to be supported. (Multiple banks can be selected from the drop down)

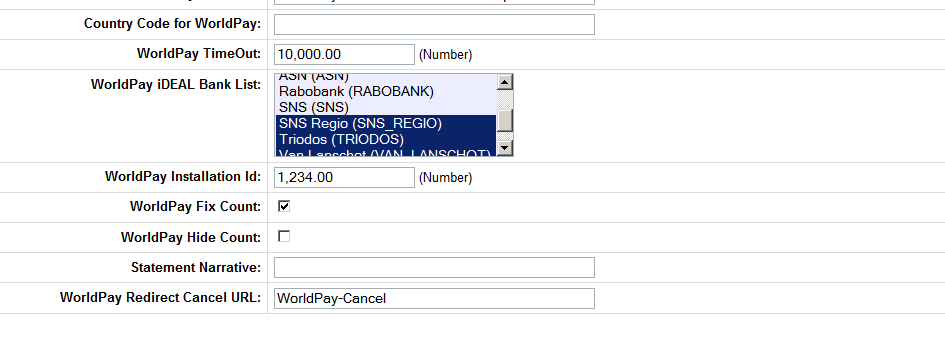


Fig: 2.7

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

### Multiple Merchant ID Support Integration

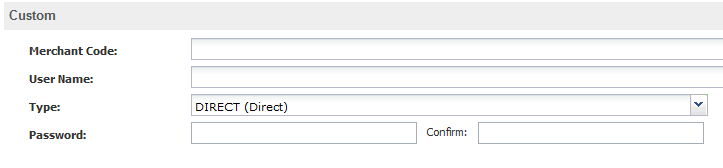
After importing metadata provided with world pay cartridge, additional fields (merchant ID, username, and password) will be created in payment methods in BM.

1. Log into Business Manager
2. From the top bar select the required site for which configurations are to be done
3. Navigate to SiteGenesis Site 🡪 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.

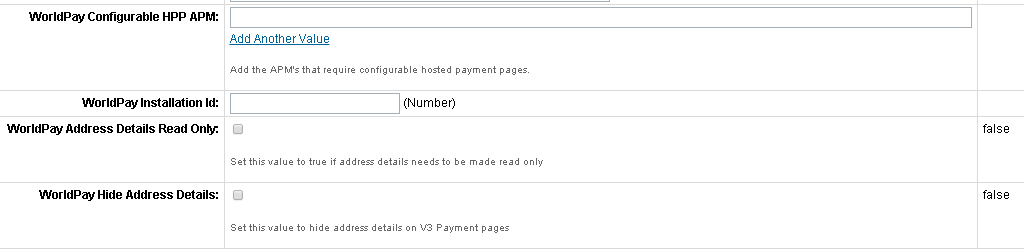


Fig 2.9

### 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 Merchant Tools *🡪 Ordering 🡪 Payment Methods*

3) Select “Credit Card Redirect” payment method and add the following data under wordlpay 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,

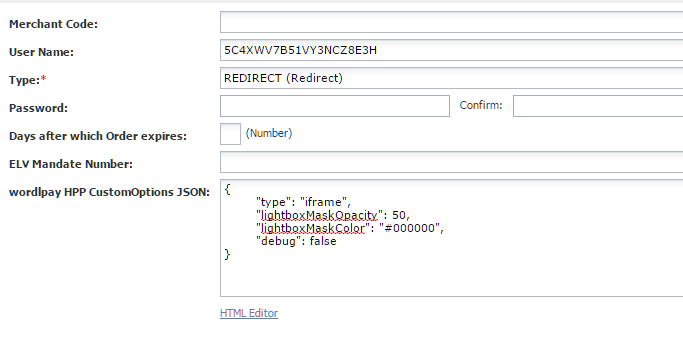
                  "lightboxMaskColor": "#000000",

               "debug": false

}

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

* lightboxMaskOpacity
* lightboxMaskColor



## 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

|  |
| --- |
| var WorldPayHelper = require('int\_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 = empty(cart)?(new dw.value.Money(0.0,dw.system.Site.getCurrent().getCurrencyCode())):cart.getNonGiftCertificateAmount();  var countryCode;  var applicablePaymentMethods;  var applicablePaymentCards;  var applicableCreditCards;  countryCode = 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  };  } |

* Also update creditcardlist under local methods

exports.InitCreditCardList=initCreditCardList;

##### 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){  var errormsg : dw.Object = new Object();  errormsg = errorMessage.errorMessage;  if (errormsg){  start(cart, require('~/cartridge/scripts/object').extend(errorMessage, {ApplicableCreditCards: creditCardList.ApplicableCreditCards}));  }  }  else{  start(cart, {ApplicableCreditCards: creditCardList.ApplicableCreditCards});  }      } else {  app.getController('Cart').Show();  }  } |

##### 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.maskedCreditCardNumber === newCreditCard.maskedCreditCardNumber && creditcard.creditCardType === newCreditCard.creditCardType) {  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('int\_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

|  |
| --- |
| var authorizationResult = PaymentProcessor.authorize(order, paymentInstrument);  if(authorizationResult.redirect){  return {  redirect :true,  redirectUrl : authorizationResult.redirectUrl  };  }else if(authorizationResult.returnToPage){  return {  returnToPage :true,  order : order  };  }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) {  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 {};  }  });  // 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 {};  } else {  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  }).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);  }  }else {  app.getController('Cart').Show();  return {};  }  } |

##### 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('int\_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) {  if(paymentMethod.equals(dw.order.OrderPaymentInstrument.METHOD\_CREDIT\_CARD)){  if (submitImpl(order.object).error) {  app.getController('COSummary').Start();  return; }  app.getController('COSummary').ShowConfirmation(order.object);  return; }  else if (WorldPay.HandleRedirection(order.object).success) {  if (submitImpl(order.object).error) {  app.getController('COSummary').Start();  return; }  app.getController('COSummary').ShowConfirmation(order.object);  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.object);  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() {  // Call the responsible controller which does the actual place order action and any payment authorization. The called  // controller must exit with a named end node "order\_created" if the order was created successfully or any other end  // node, if the order creation failed.  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);  ISML.renderTemplate('worldpayissuerredirect',{  ContinueURL: placeOrderResult.redirectUrl,  IssuerURL:placeOrderResult.redirectUrl,  PaRequest:placeOrderResult.paRequest,  TermURL:placeOrderResult.termUrl,  Order:OrderObj  });  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) {  …  ….  ..  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.custom.order\_id)) {  app.getView({  Order : Order.get(session.custom.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('int\_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>  </isif> |

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="${dw.order.PaymentMgr.getPaymentMethod(paymentInstr.paymentMethod).ID=='WorldPay'}">  <isscript>  var isValidCustomOptionsHPP = require('int\_worldpay/cartridge/scripts/common/Utils').isValidCustomOptionsHPP(dw.order.PaymentMgr.getPaymentMethod(paymentInstr.paymentMethod));  if (!empty(isValidCustomOptionsHPP) && empty(pdict.Basket)) {  var CustomOptionsHPPJSON = require('int\_worldpay/cartridge/scripts/common/Utils').getCustomOptionsHPP(dw.order.PaymentMgr.getPaymentMethod(paymentInstr.paymentMethod),  session.custom.worldpayRedirectURL, LineCntr.getOrderToken(),session.custom.preferedCard);  }  </isscript>  </isif>  </isloop>  </isif>  <isif condition="${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="${!pdict.CurrentForms.multishipping.entered.value}">  <ischeckoutprogressindicator step="3" multishipping="false" rendershipping="${pdict.Basket.productLineItems.size() == 0 ? 'false' : 'true'}"/>  <iselse/>  <ischeckoutprogressindicator step="4" multishipping="true" rendershipping="${pdict.Basket.productLineItems.size() == 0 ? 'false' : 'true'}"/>  </isif>  </isif> |

Replace form action "${URLUtils.https('COSummary-Submit')}" with "${summaryaction}" 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

|  |
| --- |
| <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>  <button class="button-fancy-large" type="submit" name="submit" value="${Resource.msg('global.submitorder','locale',null)}">  ${Resource.msg('global.submitorder','locale',null)}  </button>  </div>  <input type="hidden" name="${dw.web.CSRFProtection.getTokenName()}" value="${dw.web.CSRFProtection.generateToken()}"/>    </fieldset>  </form>  </isif>  </div>  <isinclude template="worldpaysummary"/> |

##### 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

|  |
| --- |
| <isif condition="${!empty(pdict.Basket)}><a class="section-header-note" href="${editUrl}">${Resource.msg('global.edit','locale',null)}</a></isif> |

* Replace line <isif condition="${checkoutstep != 5}"> inside div checkout-mini-cartwith <isif condition="${checkoutstep != 5 && checkoutstep != 6}">
* Replace <isif condition="${checkoutstep > 3}"> with

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

<isif condition="${checkoutstep == 6}">

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

<isif condition="${!empty(pdict.Basket)}”> <a href="${editUrl}" class="section-header-note">${Resource.msg('global.edit','locale',null)}</a></isif>

##### 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 “pdict.Basket” with “lineCtnr” inside <isif condition="${!empty(billingAddress)}">
* Wrap edit link <a> tag with non-empty basket 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>

##### 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.

|  |
| --- |
| </legend>  <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>**  **….**  **…..**  **…..**  **</fieldset>**  **<iselse/>** |

##### Update “creditcardjson.isml”

Add below code snippet to creditcardjson.isml file after expirationYear variable

|  |
| --- |
| expirationYear:pdict.SelectedCreditCard.creditCardExpirationYear,  showCvvField:dw.system.Site.getCurrent().getCustomPreferenceValue('WorldPayEnableTokenization') && !empty(pdict.SelectedCreditCard.creditCardToken) && dw.system.Site.getCurrent().getCustomPreferenceValue('WorldPayDisableCVV') |

##### 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="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" (Screenshot attached for better understanding)

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

##### 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">  <field formid="bankCode" label="paymentmethod.selectswiftcode" type="string" binding="bankCode" max-length="25" /> |







##### 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

|  |
| --- |
| <tr>  <td colspan="2" style="font-size:12px;font-family:arial;padding:20px 10px;vertical-align:top;">  <isif condition="${pdict.isPaymentKonbini}">  <h1>Click on the link below, to proceed with payment</h1>  <a href="${pdict.WorldPayRedirectURL}">Pay here</a>  <iselse>    <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;">${Resource.msg('confirmation.thankyou','checkout',null)}</th>  </tr>  <tr>  <td style="font-size:12px;font-family:arial;padding:20px 10px;vertical-align:top;">  <p>${Resource.msg('confirmation.message','checkout',null)}</p>  <p>${Resource.msg('confirmation.contact','checkout',null)}</p>  </td>  </tr>  <isif condition="${!empty(pdict.Order.custom.mandateID)}">  <tr>  <td style="${Resource.msg('email.tablecell','email',null)}" colspan="2">  <isprint value="${Resource.msg('forms.elv.mandateID','forms',null)}" /> : <isprint value="${pdict.Order.custom.mandateID}" />  </td>  </tr>  </isif>  </table>  </isif>  </td> |

###### Update “confirmation.isml”

Add below code section for Konbini payment method at the top and after closing of iscomment tag

|  |
| --- |
| <iscontent type="text/html" charset="UTF-8" compact="true"/>  <isdecorate template="checkout/pt\_orderconfirmation">  <isinclude template="util/modules"/>  <isinclude template="util/reporting/ReportOrder.isml"/>  <iscomment>  This template visualizes the order confirmation page. Note, that it  uses a different decorator template.  It displays the order related information, such as the order number,  creation date, payment information, order totals and shipments of  the order.  </iscomment>  <isset name="isPaymentKonbini" value="${false}" scope="page"/>  <isloop items="${pdict.Order.getPaymentInstruments()}" var="paymentInstr" status="piloopstate">  <isif condition="${paymentInstr.paymentMethod.equals('KONBINI-SSL')}">  <isset name="isPaymentKonbini" value="${true}" scope="page"/>  </isif>  </isloop>  <div class="confirmation <isif condition="${!pdict.CurrentCustomer.authenticated}">create-account</isif>">  <isif condition="${isPaymentKonbini}">  <h1>Click on the link below, to proceed with payment</h1>  <a href="${pdict.WorldPayRedirectURL}">Pay here</a>  <iselse>  <div class="confirmation-message">  <h1>${Resource.msg('confirmation.thankyou','checkout',null)}</h1>  <iscontentasset aid="confirmation-message" />  </div>  </isif>  <div class="order-confirmation-details">  <isorderdetails order="${pdict.Order}"/>  </div>  <isinclude template="checkout/confirmation/confirmationregister"/>  <div class="actions">  <a href="${URLUtils.http('Cart-ContinueShopping')}" class="continue">  ${Resource.msg('confirmation.returnshop','checkout',null)}  </a>  </div>  </div>  </isdecorate> |

#### Property file changes

##### Update “forms.properties”

Add below code snippet to **forms.properties**

|  |
| --- |
|  |
| creditcard.cpf=Tax Registration number  creditcard.installments=Number of Installments  paymentmethod.selectswiftcode=Enter Swift Code:  paymentmethod.selectbank=Choose your bank  paymentmethod.Iban=IBan number  paymentmethod.accountHolderName= Account Holder Name  paymentmethod.bankName= Bank Name  paymentmethod.bankLocation= Bank Location  paymentmehtod.payasorder= Save Details for Faster checkout using Pay as Order  forms.elv.creditoridentifier= Creditor Identifier  forms.elv.creditorname= Creditor Name  forms.elv.creditoraddress=Creditor Address  forms.elv.mandateID=Unique Mandate Reference |

##### Update “Resources.ds”

Add below code under Resource helper constant section

|  |
| --- |
| AVAIL\_STATUS\_NOT\_AVAILABLE : ProductAvailabilityModel.AVAILABILITY\_STATUS\_NOT\_AVAILABLE,  ENABLE\_APMLOOKUP : dw.system.Site.getCurrent().getCustomPreferenceValue('EnableAPMLookUpService') |

Add below code under Resource helper URL section

|  |
| --- |
| csrffailed : URLUtils.url('CSRF-Failed').toString(),  apmLookUp : URLUtils.https('WorldPay-APMLookupService').toString(),  loadPaymentMethods : URLUtils.https('WorldPay-LoadPaymentMethods').toString() |

### 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(data.showCvvField){  // hide CVV  $creditCard.find('input[name\*="\_cvn"]').removeClass('required');  $creditCard.find('div.cvn').hide();  }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);    // select credit card from list  $('.creditCardList').on('change', function () {  var cardUUID = $(this).val();  if (!cardUUID) {  var creditCardIndex = $(".creditCardList option:selected").index();  if(creditCardIndex == '0'){  if (SitePreferences.WP\_DISABLE\_CVV === true) {  $(".payment-method-container").find('div.cvn').show();  $(".payment-method-container").find('div.cvn').addClass('required');  $(".payment-method-container").find('div.cvn .input-text').addClass('required');  }    }  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"]').addClass('required');  $creditCardSection.find('input[name$="creditCard\_installments"]').addClass('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"]').addClass('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('true');  $(".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');  $(".card-block").hide();  $(".save-card-block").show();  $("#Payment\_Brazil").hide();  });  $("#clearpaymentform").on('click',function(e){  e.preventDefault();  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('true');  $(".card-block").hide();  $(".save-card-block").show();  $("#Payment\_Brazil").hide();  });  }  });  $('input[name$="\_selectedPaymentMethodID"]').removeAttr('checked');  $('input[value=' + paymentMethodID + ']').prop('checked', 'checked');  } |

* 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:"json"  }).done(function(data){  var result = new Array();  for(var i = 0; i < data.paymentMethod.length; i++){  result.push(data.paymentMethod[i].apm); }  var paymentMethods= $(".payment-method-options").find("input[type='radio']");  $(paymentMethods).each(function(index,value){  if($.inArray($(value).val(), result)==-1 && $(value).val() != "CREDIT\_CARD" && $(value).val()!= "WorldPay"){  $(value).parent().parent("div").hide();  }else{  $(value).parent().parent("div").show(); }});});  } |

* **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.

|  |
| --- |
| $checkoutForm.on('submit',function(e){  if($selectPaymentMethod.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);  } }  }); |

* **Make a call to new function** **populateApmMethods** **on country change** inside function exports.init and below formPrepare.init({

|  |
| --- |
| var $countrySelector= $('select[name$="\_country"]');  $countrySelector.on('change',function(e){  populateApmMethods();  var $form = $('.address');  var countryCode = $('select[id$="\_country"]', $form).val();  fieldForBrazil((selectedPaymentMethod) ? selectedPaymentMethod : 'CREDIT\_CARD',countryCode); });  if (Constants.ENABLE\_APMLOOKUP) {  populateApmMethods();  } |

### 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="creditcard.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="creditcard.installments" type="string" max-length="2" binding="Installments" missing-error="creditcard.numbermissingerror" value-error="creditcard.numbervalueerror"/> |

\*\*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\_installments"]').addClass('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 formid="cvn" label="creditcard.cvnlabel" type="string" mandatory="false" masked="0" |

#### 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.

## 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/apm.html#apm011.html

### 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** | CNY, GBP, EUR, USD, AED, ARS, AUD, AZM, BDT, BGN, BHD, BND, BRL, BSD, BWP, CAD, CDF, CHF, CLP, COP, CRC, CZK, DKK, DZD, ETB, FJD, GNF, HKD, HUF, IDR, ILS, INR, ISK, JOD, JPY, KES, KRW, KWD, KZT, LBP, LKR, MAD, MUR, MVR, MWK, MXN, MYR, NGN, NOK, NPR, NZD, OMR, PEN, PHP, PKR, PLN, RON, RUB, SAR, SCR, SEK, SGD, SLL, SYP, THB, TRY, TWD, TZS, UAH, UYU, VND, XAF, XOF, XPF, YER, ZAR, ZMK. |
| **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, France, Germany and Switzerland |
| **Acceptance Currencies** | EUR (Austria, Belgium, France, 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, CHF, DKK, HKD, NOK, SEK, 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** | en, fr, and nl |
| **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 |
| **Acceptance Currencies** | en, fr, and nl |
| **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 organisations 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 authorised 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>

### Nordea

Nordea is REDIRECT Method for Sweden and Finland.

**Payment method properties**

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

|  |  |
| --- | --- |
| Property | Description/Value |
| Countries | Sweden and Finland |
| Payment Method Mask | * **Sweden**: EBETALNING-SSL * **Finland**: SOLO-SSL |

#### Request Attribute Mapping

***Sweden***

<?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="SAPIENTNITROECOMV1" version="1.4">

<submit>

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

<description>Description here</description>

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

<orderContent>

<![CDATA[ ]]>

</orderContent>

<paymentDetails>

<EBETALNING-SSL shopperCountryCode="SWE">

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

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

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

</EBETALNING-SSL>

</paymentDetails>

<shopper>

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

</shopper>

<statementNarrative>MERCHANT STATEMENT NARRATIVE</statementNarrative>

</order>

</submit>

</paymentService>

***For Finland***

<?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="SAPIENTNITROECOMV1" version="1.4">

<submit>

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

<description>Description here</description>

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

<orderContent>

<![CDATA[ ]]>

</orderContent>

<paymentDetails>

<SOLO-SSL shopperCountryCode="FIN">

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

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

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

</SOLO-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="SAPIENTNITROECOMV1">

<reply>

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

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

</orderStatus>

</reply>

</paymentService>

### ELV

ELV is Germany based Direct alternative payment method service, which provides option to make payment using IBAN or Bank account number in Germany. ELV 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="SAPIENTNITROECOMMERCEV1" version="1.4">  <submit><order orderCode="1234576"><description>="elv order"</description><amount currencyCode="EUR" exponent="2" value="5000"/><paymentDetails><ELV-SSL><iban>DE93100000000012345678</iban>

<accountHolderName>M.Mustermann</accountHolderName><bankName>Deutsche Bank</bankName><bankLocation>Schwerin</bankLocation></ELV-SSL></paymentDetails><shopper>

<shopperEmailAddress>[Max.Mustermann@example.de</shopperEmailAddress](mailto:Max.Mustermann@example.de%3c/shopperEmailAddress)></shopper><billingAddress>

<address><address1>33 Potsdamer Straße</address1><postalCode>10785</postalCode>

<city>Berlin</city>

<countryCode>DE</countryCode>

</address>

</billingAddress>

<mandate>

<mandateType>ONE-OFF</mandateType>

<mandateId>M-SAPIENTNITROECOMMERCEV1-1234576</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="SAPIENTNITROECOMMERCEV1" version="1.4">  <submit><order orderCode="1234576"><description>="elv order"</description><amount currencyCode="EUR" exponent="2" value="5000"/><paymentDetails><ELV-SSL><iban>DE93100000000012345678</iban>

<accountHolderName>M.Mustermann</accountHolderName><bankName>Deutsche Bank</bankName><bankLocation>Schwerin</bankLocation></ELV-SSL></paymentDetails><shopper>

<shopperEmailAddress>[Max.Mustermann@example.de</shopperEmailAddress](mailto:Max.Mustermann@example.de%3c/shopperEmailAddress)></shopper><billingAddress>

<address><address1>33 Potsdamer Straße</address1><postalCode>10785</postalCode>

<city>Berlin</city>

<countryCode>DE</countryCode>

</address>

</billingAddress>

<mandate>

<mandateType>RECURRING</mandateType>

<mandateId>M-SAPIENTNITROECOMMERCEV1-1234576</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>

# 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 send 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 | ELV Mandate ID | mandateID | Mandate ID created for ELV transaction |
| 14 | World Pay 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 reponse URL |

### 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 |

# 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 Demandware 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.

# 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** |
| 15.2.0 | 02-01-2015 | Initial Release – SiteGenesis 2 compatible |
| 17.1.0 |  |  |

# 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  36148900647913 |
| Discover card | DISCOVER-SSL | 6011000400000000 |
| JCB | JCB-SSL | 3528000700000000 |
| Laser | LASER-SSL | 630495060000000000 630490017740292441 |
| Maestro | MAESTRO-SSL | 6759649826438453, 67999990100000000019 |
| MasterCard | ECMC-SSL | 5555555555554444, 5454545454545454 |
| Visa | VISA-SSL | 4444333322221111,  4911830000000 |

Supported Payment Method Mapping table:

|  |  |
| --- | --- |
| **Payment Method Name** | **Key Value** |
| 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

|  |  |
| --- | --- |
| **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 Demandware 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 |