PayDollar PayGate

Integration Guide version 3.67

(Leave Blank Intentionally)

Copyright Information

AsiaPay (HK) Limited
Unit 1701-05, 17/F, K. Wah Centre
191 Java Road
Hong Kong.
Telephone (852) 2538 8278
Fax: (852) 2545 3898
Web site: http://www.asiapay.com

Confidentiality and Usage

Revision History

Revision	Date	Revision Description
0.0	July 11,2001	First Draft issue
1.0	Apr 28, 2003	First issue (All in One Copy)
1.1	Oct 7, 2003	Add VBV Parameters for Server Side Integration
1.2	Nov 1, 2003	Modify VBV Parameters for Server Side Integration
1.3	Nov 6, 2003	Deleted Data Feed Retry Message
1.4	Nov 13, 2003	Add AuthId to Data Feed Output
1.5	Feb 20, 2004	Add JCB Card Type and Japanese Lang Parameter
1.6	Mar 9, 2004	All-in-One Version
1.7	May 6, 2004	Add JPY and TWD Currency Code
1.8	May 13, 2004	Add data feed flow comment
1.9	Aug. 20, 2004	Add AMEX and Security Code parameter
2.0	Sept 14, 2004	Add Payment Flow, 3D Transaction, PayPal Transaction and New Currencies
2.1	Oct 8, 2004	Modify the pop-up flow to be inline flow of 3D Transaction
2.2	Dec 6, 2004	Add errorUrl parameter to Direct Client Side Connection
2.3	Sep 15, 2005	Add ECI and Payer Authentication Status to datafeed section
2.5	Jan 25, 2006	Add data feed port requirement
2.6	Mar 30, 2006	Add new parameters to datafeed : - sourceIp - ipCountry
2.7	Jun 15, 2006	Add sample data feed page in ASP.
2.8	Jun 20, 2006	Modified Payment Method to support CUP.
2.9	Nov 27, 2006	Add Transaction Handling and FAQ Section
3.0	Aug 30, 2007	Add Alipay payment method
3.1	Jan 24, 2008	Language update – Supporting Thai
3.11	June, 25, 2008	Revise and updated content
3.2	Mar 25, 2009	Add China UnionPay (CHINAPAY) payment method
3.2.1	Apr 29, 2009	Add new parameter “payMethod” to Data Feed Output
3.2.2	May 14, 2009	Add APPENDIX
3.3	Nov 5, 2009	Add Tenpay and 99bill payment method
3.4	Dec 18, 2009	Add new optional parameter for connect to payment page. Origin Country and Destination Country
3.5	Jun 1, 2010	Add new parameter “cardIssuingCountry” to Data Feed Output
3.5.5	Jun 25, 2010	Add new parameter “channelType” to Data Feed Output
3.6	Jun 29,2010	Add new function “Multi-Currency Processing” and merge “Server Side Direct Connection” Spec into one spec. Move “Data Feed Handling” into a new section. Add mps fields in datafeed and merchant api.
3.7	Jul 27, 2010	Add Secure Hash Function. Add new optional parameter “amount” for Merchant API RequestRefund action

3.8	Dec 6,2010	Merchant can set Return Value Link (Data Feed) at merchant admin Language update – Supporting French, German, Russian and Spanish
3.9	Feb 16, 2011	Add new optional parameters on Client Post Through Browser for disable the print and retry function at payment result page.
3.10	Apr 1, 2011	Update the description of the parameter “payType” for integration. Add new parameters “mSchPayId”, “dSchPayId” to Data Feed Output.
3.11	Apr 20, 2011	Add MOP, PHP, THB, MYR, IDR and KRW Currency Code
3.12	Jun 1, 2011	Update the list of response code from PayDollar
3.13	Jul 12, 2011	Add SAR, NZD, AED, BND Currency Code Add MEPS payment method Add MemberPay Service integration parameters
3.14	Nov 2, 2011	Update Browser version compatibility Update IP Ranges Add “AMEX SafeKey” Update the wording of “3D-Secure” to “3-D Secure” Add datafeed return parameters (AlertCode and MerchantId)
3.15	Jan 30,2012	Update Debit payment for Thailand Add SCB , BAY, UOB, TMB , UOB , KTB , IBANKING, Bill Payment
3.16	Jun 6, 2012	Add Installment service
3.17	Jun 27, 2012	Update testing environment IP Ranges
3.18	Aug 3, 2012	Add Airline data support - Air Ticket Number
3.19	Oct 5, 2012	Add new optional parameters panFirst4, panLast4, accountHash and accountHashAlgo to Data Feed Out and Merchant API Query Add new payment method “UPOP”
3.20	Jan 8, 2013	Add customer billing information
3.21	Jan 28, 2013	Add payment gateway transaction time into datafeed and query function
3.22	Jun 13, 2013	Add VND Currency Code
3.23	Jun 21,2013	Add 99BILL, ALIPAY, CHINAPAY, PAYPAL, PPS, TENPAY, MEPS, OCTOPUS, ENETS, MYCLEAR, POLI, UPOP, UPOP-GNETE, UPOP-DNA, FUIOU, SCB, KRUNGSRIONLINE, KTB, UOB, TMB, IBANKING, BPM, GCash, BancNet, SMARTMONEY into direct client side connection method
3.24	Aug 22, 2013	Add M2U and CIMBCLICK
3.25	Nov 11, 2013	Add new optional parameter “□arame” to Data Feed output and Merchant API Query.
3.26	Nov 22, 2013	Add new currency INR
3.27	Feb 18, 2014	Add new language Vietnamese
3.28	Mar 19, 2014	Add OCTOPUS
3.29	May 29, 2014	Add “mpLatestStaticToken” and “mpMemberId” in DataFeed

3.30	Jun 06, 2014	Add “timeoutValue” to control Octopus timeout value
3.31	Jul 03, 2014	Add “hiddenAmount” and “expDateCheck”, for Client Post Through Browser
3.32	Oct 06, 2014	- Add Ewallet Integration Parameters in Direct Client Side Connection and Server Side Direct Connection - Add Ewallet information in Data Feed Response
3.33	Oct 31, 2014	Add WeChat payment method
3.34	Nov 3, 2014	Add Multi Currency Pricing and Integration Parameters for card data encryption
3.35	Nov 25, 2014	Update description of optional parameter “foreignAmount” on MCP Function
3.36	Dec 2, 2014	Add “promotionCode”, “promotionRuleCode”, “promotionOriginalAmt” on Query API
3.37	Dec 12, 2014	Add ONEPAY (ATM Card) Payment Method
3.38	Mar 19, 2015	Change the type of “Ord”
3.39	Jun 11, 2015	Add new parameters “rewardsRedeemTotalAmt”, “rewardsRedeemType”, “rewardsRedeemProgram”, “rewardsRedeemCode1”, “rewardsRedeemLabel1”, “rewardsRedeemAmt1”, “rewardsBalance1”, “rewardsRedeemCode2”, “rewardsRedeemLabel2”, “rewardsRedeemAmt2”, “rewardsBalance2”, “rewardsRedeemCode3”, “rewardsRedeemLabel3”, “rewardsRedeemAmt3”, “rewardsBalance3” to Data Feed Output and Query API Output
3.40	Aug 19, 2015	Add new parameters “netAmtAfterRewards” to Data Feed Output and Query API Output Add new parameters “netAmtAfterRewards “, “rewardsRedeemTotalAmt”, “rewardsRedeemType”, “rewardsRedeemProgram”, “rewardsRedeemCode1”, “rewardsRedeemLabel1”, “rewardsRedeemAmt1”, “rewardsBalance1”, “rewardsRedeemCode2”, “rewardsRedeemLabel2”, “rewardsRedeemAmt2”, “rewardsBalance2”, “rewardsRedeemCode3”, “rewardsRedeemLabel3”, “rewardsRedeemAmt3”, “rewardsBalance3” to Capture API Output
3.41	Sep 14, 2015	Add VISA Checkout and MasterPass Payment Method
3.42	Sep 16,2015	Add installOnly
3.43	Sep 23,2015	Revise output of “Settlement Report Request” Update description of Datafeed for VISA Checkout
3.44	Nov 12, 2015	Add VISA Checkout and MasterPass Payment Method
3.45	Dec 16,2015	Change the length of “5atched”
3.46	Jan 22, 2016	Add API function – Query payment request log
3.47	Jan 26, 2016	Add new parameter “payMode” for control which payment mode use of transaction
3.48	Jan 28, 2016	Updates on API function – Query payment request log Update compatible version of browsers
3.49	Apr 6, 2016	Updates Datafeed Updates API return value
3.50	Apr 28,2016	Updates API return value
3.51	May 23,2016	Add WELEND, update settlement report request

3.52	Sep 12, 2016	Change IP Address of UAT server
3.53	Feb 7, 2017	Add new integration parameter “orderPhoneNo”
3.54	Mar 14, 2017	Add installment provider datafeed output Add VA payment flow Add Credit Card with Installment flow
3.55	May 12, 2017	Add new API function “Cancel Payment” (Only for payment bank FASPAY)
3.56	Aug 28, 2017	Update example of source codes
3.57	Jun 18, 2019	Add MoMo Payment
3.58	July 15, 2019	Add Samsung Pay Payment
3.59	Sep 24, 2019	Add 3DS 2.0 related parts. Update the list of “Response Code From PayDollar” Add PaySDK iOS Add PaySDK Android
3.60	Oct 17, 2019	Flow chart Update. Update length for field named “threeDSVersion”
3.61	Jan 03, 2020	Added Wechat Payment in PaySDK iOS and PaySDK Android
3.62	Jul 27, 2020	Added New Payment Method and eWallet Code Updated SRC, PRC and ORD data type
3.63	Sep 2, 2020	Update PayDollar production IP range
3.64	Sep 29, 2020	Add new integration section “Optional Parameter for Shipping address” and “Optional Parameter for Customer Detail”
3.65	Jul 14, 2021	Update appUrlScheme, remark parameter details Update secureHash calculation Add new PRC code Add new payment method
3.66	Apr 06, 2022	Added deviceMode arameter Update deeplink, redirect parameter details Add new payment method Add panMasked for Query API
3.67	Jun 16, 2022	Added new payment method

Table of Contents

1 Overview

Introduction

PayDollar PayGate is a powerful web-based online payment services platform, which provides secure, multi-channel, multi-lingual and multi-currency payment services. It is used by many renowned companies and organizations in the region.

This technical specification document prescribes the constituent parts of specification for integration of an e-commerce web site with PayDollar e-commerce service, the on-line payment service by AsiaPay (HK) Limited, by subscribed merchants of the service. This document has been created to ensure that all technical specifications contain sufficient information to enable a merchant to design and modify the codes of an existing on-line shopping architecture or software to cater for the payment –enabling service. It also provides a checklist to enable the reviewers of specifications to conduct tests on the functionalities of the integration.

There is an extra parameter section under each connection method. That section is especially designed for 3DS 2.0 (Three Domain Secure 2.0). 3DS authentication is an additional security layer for card-not-present transactions. With 3DS 2.0, it is aimed to achieve a better, stronger fraud-detection intelligence.

PayDollar PayGate facilitates merchant to connect to our network with great flexibility. Merchant can choose one of the following integration methods, which will be described in detail in the document.

- Client Post through Browser (e.g. Shopping Cart)
- Direct Client Side Connection
- Server Side Direct Connection (e.g. IVR System, Mobile App)

Moreover, a list of merchant API functions will be also described in detail in the later section.

2 Connection method

2.1 Client Post Through Browser

It is the most popular connection method among merchants. The advantage of this connection method is simple and speedy. On the other hand, payment transaction flow is ready to use. Merchant can kick off the web site on-the-fly with just a small scale integration.

Scope and Compatibility

This connection is designed for merchants who have *Online Shopping Cart System*. The routine is HTML-based with Javascript and should be widely applicable to on-line shopping cart software and architecture, whose technical specifications and varieties are beyond the scope of this document. Compatibility with shopping cart software is yet to be exhaustively given and would not be included in the scope of this document.

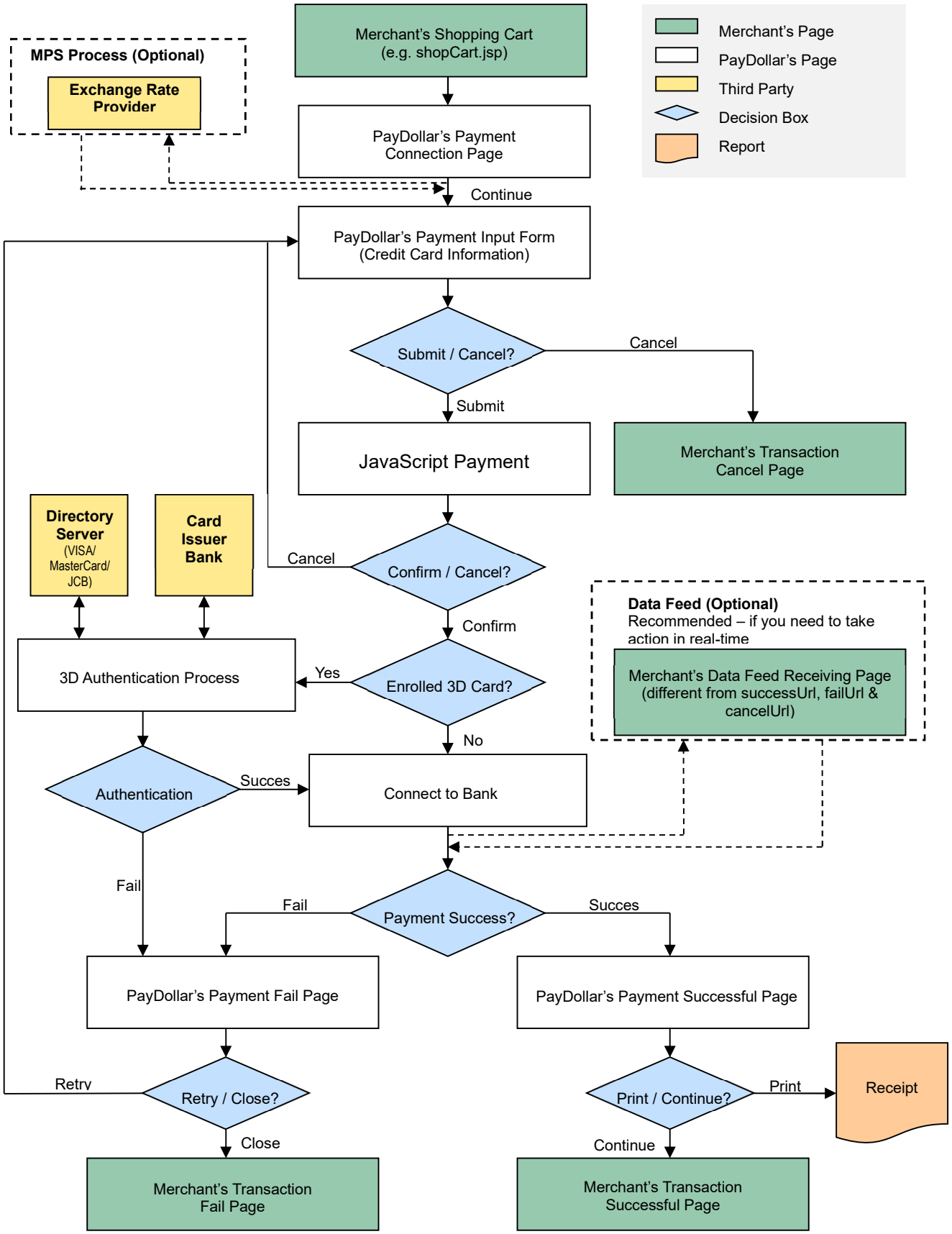
The compatible version of the software code is as follow:

Software Code	Version
HTML	4.0
Javascript	1.3

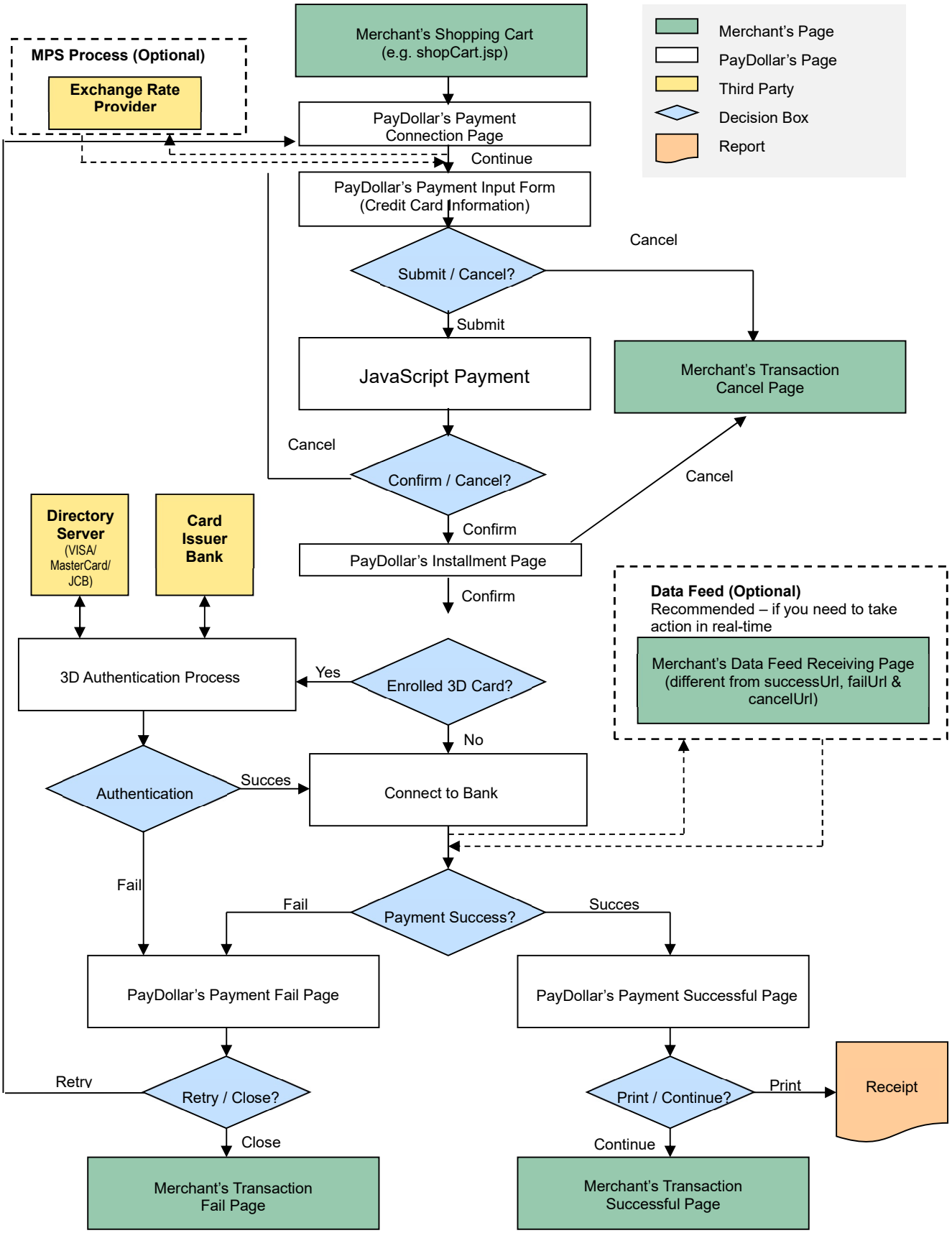
The version compatibility of the software code with popular browser software is as follow:

Browser	Version
Microsoft Internet Explorer	11 or above
Mozilla Firefox	27 or above
Google Chrome	30 or above
Safari	7 or above

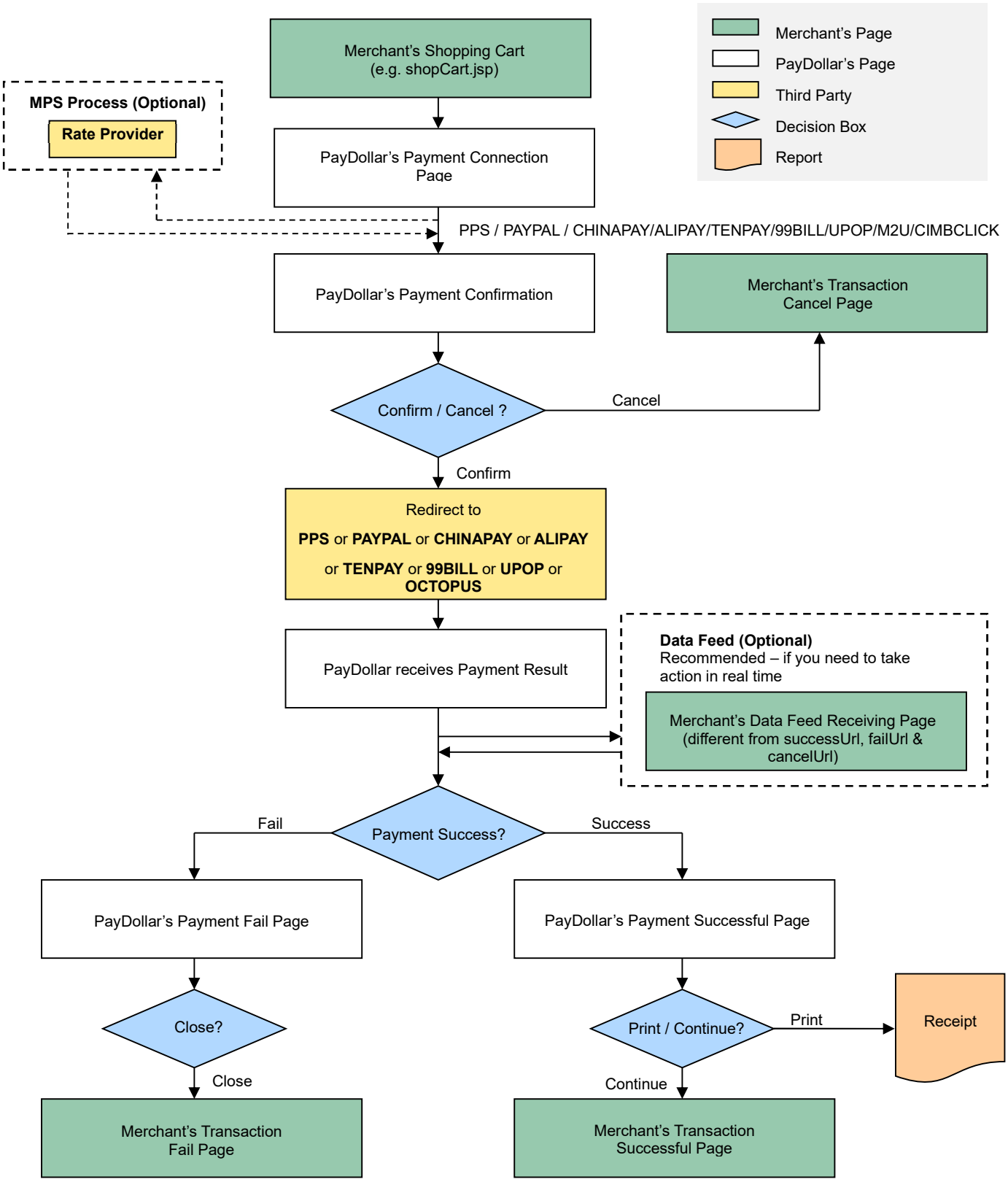
Credit Card Payment Flow



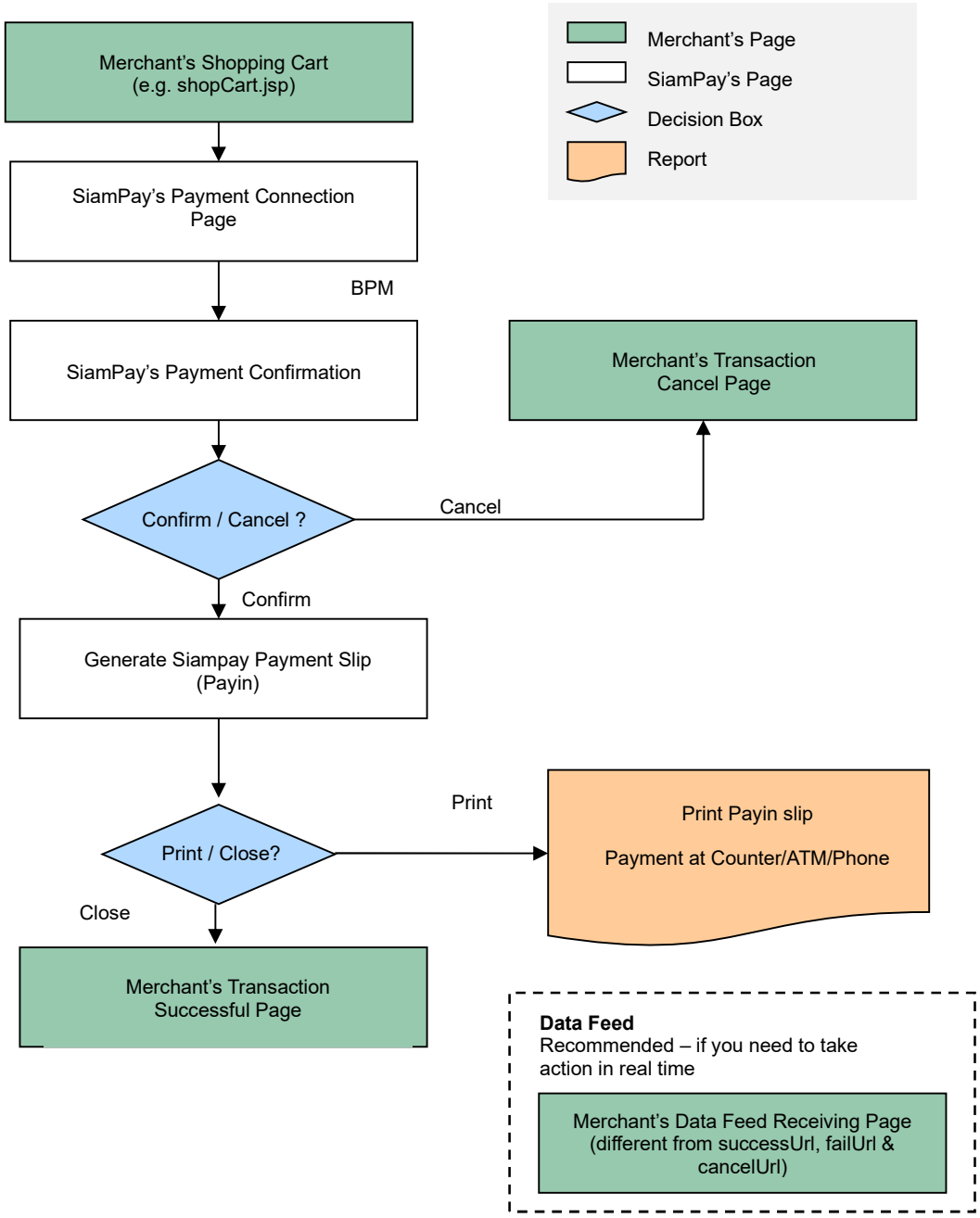
Credit Card with Installment Payment Flow



99BILL, ALIPAY, CHINAPAY, PAYPAL, PPS, TENPAY, SCB, BAY, KTB, UOB, TMB, BBL iBanking, UPOP, M2U, CIMBCLICK, OCTOPUS, WECHAT, ONEPAY Payment Flow

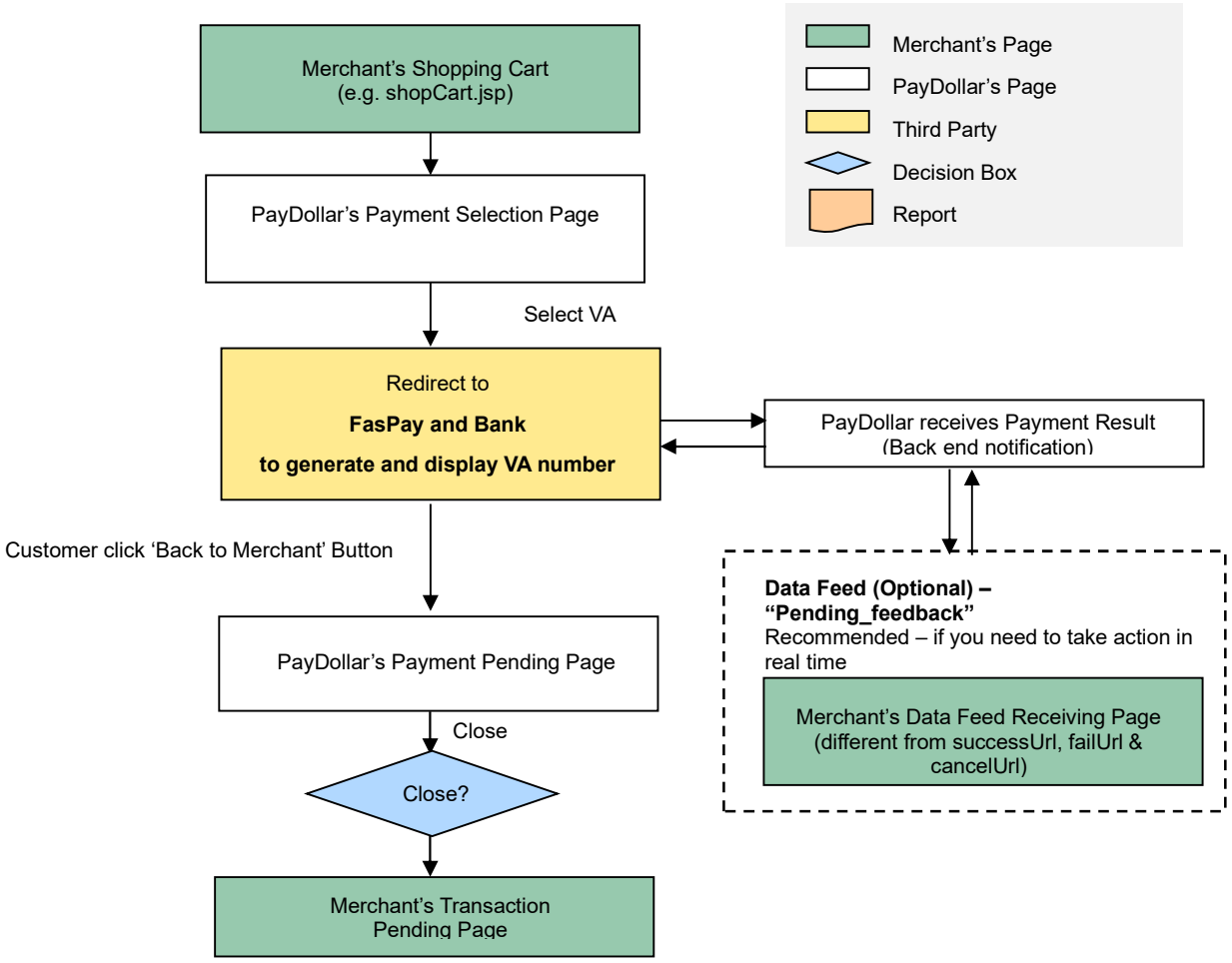


Bill Payment Flow (BPM)

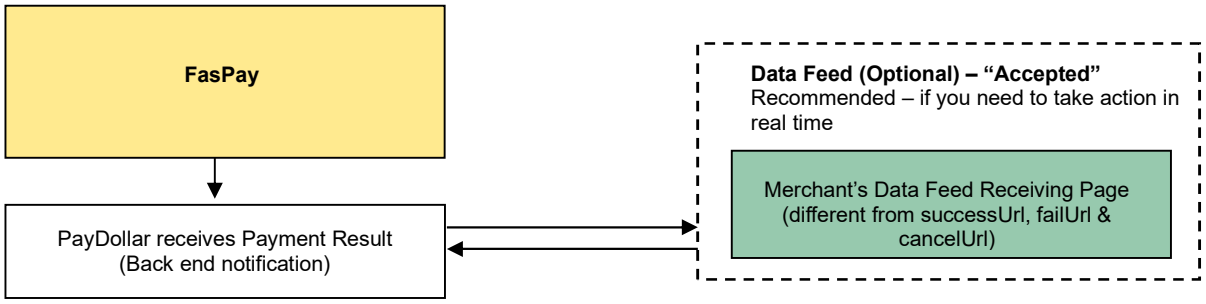


Permata, BII and BCA Virtual Account (VA) Payment Flow

Step 1: Customer generate Virtual Account online



Step 2: Customer pay to Virtual Account offline before expiry date



It is notable that the software codes of the payment routine, whose example is as given, should be embedded into the integration page, as in above, which should be able to generate the sum of purchase from the previous shopping practice of the user. The subsequent parts of the flow would be directed to system architecture in which the payment details are to be submitted by the user, and handled for settlement and clearance.

Definition of Parameters in the Integration Page

The following are the parameters for integration. PayDollar PayGate is case sensitive. Make sure the typeface is correct. When a transaction is finished, the system will return customer a payment message. Merchant can create static HTML pages to display the message. If merchant's web site supports data feed, the system can return payment message as shown in the following table.

Parameters	Data Type	Descriptions
Required Parameter (with UTF-8 Encoding) for connect to our payment page		
orderRef	Text (35)	Merchant's Order Reference Number
currCode	Text (3)	The currency of the payment: "344" – HKD "840" – USD "702" – SGD "156" – CNY (RMB) "392" – JPY "901" – TWD "036" – AUD "978" – EUR "826" – GBP "124" – CAD "446" – MOP "608" – PHP "764" – THB "458" – MYR "360" – IDR "410" – KRW "682" – SAR "554" – NZD "784" – AED "096" – BND "704" – VND "356" – INR Remark: mpsMode = SCP, the currCode value should be in foreign currency. mpsMode = MCP, the currCode value should be in base currency.
amount	Number (12,2)	The total amount you want to charge the customer for the provided currency. Remark: mpsMode = SCP, the amount should be calculated in foreign currency. mpsMode = MCP, the amount should be calculated in base currency.
lang	Text (1)	The language of the payment page "E" – English "C" – Traditional Chinese "X" – Simplified Chinese "J" – Japanese "T" – Thai "F" – French "G" – German "R" – Russian

		"S" – Spanish "V" – Vietnamese
cancelUrl	Text (300)	A Web page address you want us to redirect upon the transaction being cancelled by your customer **(For display purpose only. DO NOT use this URL to update your system. Please use DataFeed for this purpose.)**
failUrl	Text (300)	A Web page address you want us to redirect upon the transaction being rejected by us. **(For display purpose only. DO NOT use this URL to update your system. Please use DataFeed for this purpose.)**
successUrl	Text (300)	A Web page address you want us to redirect upon the transaction being accepted by us **(For display purpose only. DO NOT use this URL to update your system. Please use DataFeed for this purpose.)**
merchantId	Number	The merchant ID we provide to you
payType	Text(1) ("N", "H")	The payment type: "N" – Normal Payment (Sales) "H" – Hold Payment (Authorize only) For merchants who use authorize mode, please be reminded to perform the CAPTURE action as soon as the transaction is confirmed as valid. Once captured, the customer's credit card will be debited in coming bank settlement processing. If the merchant does not capture/reverse the authorized transaction over 14 days, the credit limit will be released to the cardholder after a time period which is subjected to card issuing bank Merchant may capture/reverse the authorized transaction in the merchant administration site > Operation > Transaction Detail. Remark: Hold Payment is not available for 99BILL, ALIPAY, CHIANPAY, PAYPAL, PPS, TENPAY, WECHAT, MEPS,OCTOPUS
payMethod	Text	The payment method: "ALL" – All the available payment method "CC" – Credit Card Payment "VISA" – Visa Payment "Master" – MasterCard Payment "JCB" – JCB Payment "AMEX" – AMEX Payment "Diners" – Diners Club Payment "PPS" – PayDollar PPS Payment "PAYPAL" – PayPal By PayDollar Payment

		“CHINAPAY” – China UnionPay By PayDollar Payment “ALIPAY” – Alipay By PayDollar Payment “ALIPAYHKONL” – Alipay HK By PayDollar Payment “TENPAY” – TENPAY BY PayDollar Payment “99BILL” – 99BILL BY PayDollar Payment “MEPS” – MEPS BY PayDollar Payment “SCB” –SCB (SCB Easy) BY PayDollar Payment “BPM” –Bill Payment BY PayDollar Payment “KTB” –Krung thai Bank (KTB Online) BY PayDollar Payment “UOB” –United Oversea bank BY PayDollar Payment “KRUNGSRIONLINE” –Bank of Ayudhya (KRUNGSRIONLINE) BY PayDollar Payment “TMB” –TMB Bank BY PayDollar Payment “IBANKING” –Bangkok Bank iBanking BY PayDollar Payment “UPOP” – UPOP BY PayDollar Payment “M2U” – M2U BY PayDollar Payment “CIMBCLICK” – CIMBCLICK BY PayDollar Payment “OCTOPUS” – OCTOPUS BY PayDollar Payment “PayMe” – PayMe Payment “WECHAT” – WeChat Pay BY PayDollar Payment (deprecated) “WECHATONL” – WeChat Pay BY PayDollar Payment “ONEPAY” – ONEPAY BY PayDollar Payment “WELEND” – WELEND BY PayDollar Payment “MOMOPAY” – MoMo eWallet Payment “LINEPAY” – LINE Pay Payment “PayMaya” – PayMaya Payment “BPI” – BPI Netbanking Payment “GCash” – GCash Payment “ENETS” – eNETS Payment “ENETSBANKING” – eNETS Net Banking Payment “ENETSQR” – eNETS QR Payment “FPS” – FPS Payment "QRIS" - QR Code Indonesian Standard Payment “DuitNow” – DuitNow Payment "OVO” – OVO Payment “DANA” – DANA Payment “KREDIVO” – Kredivo Payment “TouchnGo” – Touch n’ Go Payment

		"POLI" - POLi Payments "PAYID" - PayID Payment "humm" - humm Payment "ZIPPAY" - ZipPay Payment "HOOLAH" - hoolah Payment "ATOME" - Atome Payment “Pace” – Pace Payment “SHOPEEPAY” – ShopeePay Payment “TendoPay” – TendoPay Payment “Eximbay” – Eximbay Payment (Korea) “PAYU” – PayU Payment “TRUEMONEY” - TrueMoney Wallet Payment “JKOPAY” – JKOPAY Wallet Payment “VCO” – VISA Checkout Payment “MP” – MasterPass Payment *Device Wallet Payment:* “SAMSUNG” – Samsung Pay Payment “GOOGLE” – Google Pay™ Payment “APPLEPAY” – Apple Pay Payment
Optional Parameter for installment		
installment_service	Text (1) (“T”,”F”)	Installment service indicator
installment_period	Text (20)	In number of months *Not provide will Enable installment selection during the payment *Multiple period with separator “|” only for payment method “WELEND“
installOnly	Text (1) (“T”,”F”)	Control the mandatory of Installment payment
Optional Parameter for airline data		
airline_service	Text (1) (“T”,”F”)	Airline service indicator
airline_ticketNumber	Text (13)	Air ticket number
Optional Parameter for Multi Currency Pricing (MCP)		
mpsMode	Text(3)	The Multi-Currency Processing Service (MPS) Mode: “NIL” or not provide – Disable MPS (merchant not using MPS) “SCP” – Enable MPS with ‘Simple Currency Conversion’

		"DCC" – Enable MPS with 'Dynamic Currency Conversion' "MCP" – Enable MPS with 'Multi Currency Pricing' For merchant who applied MPS function
multiRateID	Text (12)	Unique Rate ID *Returned from API function "MCPMultiRateEnquiry"
mpsExRate	Number (10,4)	Exchange rate between the merchant's base currency and the foreign currency. *Returned from API function "MCPMultiRateEnquiry"
foreignCurrCode	Text (3)	Foreign currency code
foreignAmount	Number (12,2)	Calculated foreign amount ***The foreign amount should be converted from a base amount with minimum value HKD10 / MOP10 to avoid tolerance check failure.**
Optional Parameter for Promotion		
promotion	Text(1); ("T" or "F")	Enable / Disable the promotion
promotionCode	Text(8);	The promotion campaign code
promotionRuleCode	Text(8);	The promotion rule code
promotionOriginalAmt	Number (12,2)	The promotion original amount
Optional Parameter for connect to our payment page		
remark	Text (50)	A remark field for you to store additional data that will not show on the transaction web page
deviceMode	Text	Control the display mode for the payment page auto – PayDollar determine the display mode automatics by using the browser environment (Default) pc – Force to use pc layout mobile – Force to use mobile layout and payment flow
redirect	Number	Number of seconds auto-redirection to merchant's site takes place at PayDollar's Payment Success / Fail page 0 – Skip PayDollar's Payment result page and auto-redirect to merchant result page -1 – Disable auto-redirection for PayDollar's Payment result page -2 – Only skip PayDollar's Payment result page and auto-redirect to merchant result page for successful transaction Others positive number – Number of seconds for the auto-redirection
oriCountry	Number(3)	Origin Country Code Example: 344 – "HK"

		840 – “US”
destCountry	Number(3)	Destination Country Code Example: 344 – “HK” 840 – “US”
secureHash	Text (40)	Secure hash is used to authenticate the integrity of the transaction information and the identity of the merchant. It is calculated by hashing the combination of various transaction parameters and the Secure Hash Secret. *Applies to merchants who registered this function only. For more information, please refer to section 4.
print	Text(2) ; (“no”)	Disable the print function at payment result page.
failRetry	Text(2) ; (“no”)	Disable the retry function when the transaction is rejected
timeoutValue	Number	Timeout Value in second, only for OCTOPUS payment and ranged 180 and 7200
hiddenAmount	Text (1) (“T”,”F”)	Flag to control the display of amount. “T” for showing amount and “F” for not showing display of amount. *Enable to approved merchant only.
expDateCheck	Number	Value to control the expiry date checking. Eg. If “1” is submitted, the card expiry date must not be expired in one month. Only integer is allowed and minimum value is “1” and maximum value is “192”. *Enable to approved merchant only.
payMode	Text(2)	Payment Mode “EC” - Electronic Commerce (Default) “MO” - Mail Order “RC” - Recurring
orderPhoneNo	Text(30)	Phone Number of order
appUrlScheme	Text	iOS Universal Link or Android App Link URL for Web-2-App flow return to the merchant App from the payment App Available for the following payment method: “OCTOPUS” – iOS (Mobile flow) “ALIPAYHKONL” – iOS, Android (Mobile H5 flow) “WECHATONL” – iOS, Android (Mobile H5 flow)
deeplink	Text(1)	Flag for the Web-2-App flow “0”: (Default) Redirect to Payment Service Host directly “1”: Call Payment Service token with Mobile H5 directly “2”: Display summary page with button of Mobile H5 “3”: Display summary page with button of Mobile H5 and Redirect Login

		Available for “ALIPAYHKONL”
oaflag	Text (1) (“T”,”F”)	To enable to use JSAPI (Payment Service In-App Webview Payment) Available for “WECHATONL”
Optional Parameter for Shipping address (required to be eligible for PayPal Seller Protection)		
shippingAddress1	Text (240)	The first line of the shipping address. Must contain the full address
shippingAddress2	Text (240)	The second line of the shipping address. eg.suite or apartment number
shippingCity	Text (100)	The city of the shipping address
shippingState	Text (100)	The state of the shipping address
shippingPostalCode	Text (50)	The postal code of shipping address which is the zip code or equivalent
shippingCountryCode	Text (2)	The country code that identified the country or region
Optional Parameter for customer details (required to be eligible for PayPal Seller Protection)		
customer_firstName	Text (50)	Customer’s first name
customer_lastName	Text (50)	Customer’s last name
customer_email	Text (100)	Customer’s email address
customer_account_id	Text (100)	Unique identifier of the customer
customer_phone	Text (100)	Customer’s phone number
customer_country_code	Text (2)	Customer’s country code that identified the country or region eg.US
customer_create_date	Text (14)	Date after which no further payment shall be performed. Format: yyyyMMDDHHmmss e.g. “20200401122630”

Parameters for Supporting 3DS 2.0

The following are the parameters especially for 3DS 2.0. Merchant shall fulfill the parameters requirement to start using 3DS 2.0. The parameters are case sensitive and should be in UTF-8 encoding.

Parameters	Data Type	Mandatory	Descriptions
Basic Parameters			
threeDSTransType	Text (2)	No	Transaction type. Select the most suitable one. “01” – Goods/ Service Purchase ** Default value if threeDSTransType not provided.* “03” – Check Acceptance “10” – Account Funding “11” – Quasi-Cash Transaction “28” – Prepaid Activation and Load

threeDSCustomerEmail	Text (254)	No	The customer's email address * Strongly suggest to provide if available. Shall meet requirements of Section 3.4 of IETF RFC 5322.
threeDSMobilePhoneCountryCode	Text (3)	No	Customer's mobile phone number country code. Provide if available. If "threeDSMobilePhoneCountryCode" is provided, "threeDSMobilePhoneNumber" should also be provided. e.g. "852" – Hong Kong
threeDSMobilePhoneNumber	Text (15)	No	Customer's mobile phone number. Provide if available. If "threeDSMobilePhoneNumber" is provided, "threeDSMobilePhoneCountryCode" should also be provided.
threeDSHomePhoneCountryCode	Text (3)	No	Customer's home phone number country code. Provide if available. If "threeDSHomePhoneCountryCode" is provided, "threeDSHomePhoneNumber" should also be provided. e.g. "852" – Hong Kong
threeDSHomePhoneNumber	Text (15)	No	Customer's home phone number. Provide if available. If "threeDSHomePhoneNumber" is provided, "threeDSHomePhoneCountryCode" should also be provided.
threeDSWorkPhoneCountryCode	Text (3)	No	Customer's work phone number country code. Provide if available. If "threeDSWorkPhoneCountryCode" is provided, "threeDSWorkPhoneNumber" should also be provided. e.g. "852" – Hong Kong
threeDSWorkPhoneNumber	Text (15)	No	Customer's work phone number. Provide if available. If "threeDSWorkPhoneNumber" is provided, "threeDSWorkPhoneCountryCode" should also be provided.
threeDSIsFirstTimeItemOrder	Text (1)	No	Flag to show if the customer is re-ordering the

			item / product. i.e. purchasing the same item / product Provide only if the payment related to purchase item / product. "T" – It is the first time the customer purchases this item / product. "F" – It is the NOT the first time the customer purchases the item / product, it is re-ordering.
threeDSChallengePreference	Text(2)	No	Indicates whether a challenge is requested for this transaction. "01" – No preference ** Default value if threeDSChallengePreference not provided.* "02" – No challenge requested * "03" – Challenge requested (Merchant preference) "04" – Challenge requested (Mandate) "05" – No challenge requested (transactional risk analysis is already performed) * "06" – No challenge requested (Data share only)* "07" – No challenge requested (strong consumer authentication is already performed) * "08" – No challenge requested (utilise whitelist exemption if no challenge required) * "09" – Challenge requested (whitelist prompt requested if challenge required) **** If "No challenge requested" options are selected, the chargeback liability shift to merchant.**
Recurring / Installment Payment Related (Provide only if it is a recurring / installment payment)			
threeDSRecurringFrequency	Text (4)	Yes (Only for recurring / installment payment)	Minimum number of days between payment. e.g. "30" – 30 days between payments.
threeDSRecurringExpiry	Text (8)	Yes (Only for recurring /	Date after which no further payment shall be performed. Format: YYYYMMDD

		installment payment)	e.g. “20190401”
Billing Address Related (Provide only if billing address is available)			
threeDSBillingCountryCode	Text (3)	Yes (Only if billing address is available)	Payment cardholder billing country. Shall be the ISO 3166-1 numeric three-digit country code. e.g. “344” – Hong Kong, “840” – US
threeDSBillingState	Text (3)	No	Payment cardholder billing state Should be the country subdivision code defined in ISO 3166-2. Provide if available, as some countries do not have subdivision code. e.g. “PE” – Prince Edward Island of California
threeDSBillingCity	Text (50)	Yes (Only if billing address is available)	Payment cardholder billing city e.g. “Hong Kong”
threeDSBillingLine1	Text (50)	Yes (Only if billing address is available)	Payment cardholder billing address line 1
threeDSBillingLine2	Text (50)	No	Payment cardholder billing address line 2 Provide if available
threeDSBillingLine3	Text (50)	No	Payment cardholder billing address line 3 Provide if available
threeDSBillingPostalCode	Text (16)	No	Payment cardholder billing postal code Provide if available
Shipping / Delivery Related (Provide only if the payment requires shipping / delivery)			
threeDSDeliveryTime	Text (2)	No	The delivering time for this purchase. “01” – Electronic Delivery “02” – Same day shipping “03” – Overnight shipping “04” – Two-day or more shipping
threeDSDeliveryEmail	Text (254)	No	The email recipient of the purchased product. Provide only if the purchased product is using electronic delivery.
threeDSShippingDetails	Text (2)	No	Shipping details / type. Select the most suitable one.

			"01" – Ship to cardholder's billing address "02" – Ship to another verified address stored in merchant "03" – Ship to address that is different than the cardholder's billing address "04" – Ship to Store / Pick-up at local store (Store address shall be populated in shipping address fields) "05" – Digital goods (includes online services, electronic gift cards and redemption codes) "06" – Travel and Event tickets, not shipped "07" – Other (for example, Gaming, digital services not shipped, e-media subscriptions, etc.)
threeDSShippingCountryCode	Text (3)	No	Shipping / delivery country. Shall be the ISO 3166-1 numeric three-digit country code. e.g. "344" – Hong Kong, "840" – US
threeDSShippingState	Text (3)	No	Shipping / delivery state Should be the country subdivision code defined in ISO 3166-2 e.g. "PE" – Prince Edward Island of California
threeDSShippingCity	Text (50)	No	Shipping / delivery city. e.g. "Hong Kong"
threeDSShippingLine1	Text (50)	No	Shipping / delivery address line 1
threeDSShippingLine2	Text (50)	No	Shipping / delivery address line 2 Provide (if available).
threeDSShippingLine3	Text (50)	No	Shipping / delivery address line 3 Provide (if available).
threeDSShippingPostalCode	Text (16)	No	Shipping / delivery postal code Provide (if available).
threeDSIsAddrMatch	Text (1)	No	Flag for comparing the billing address and shipping address. Provide (if available). "T" – Shipping address matches billing address "F" – Shipping address NOT match billing address
Gift Card / Prepaid Card Purchase Related (Provide only if the purchase related to gift card / prepaid card)			

threeDSGiftCardAmount	Text (15)	No	Gift card / prepaid card purchase amount.
threeDSGiftCardCurr	Text (3)	No	Gift card / prepaid card purchase currency. ISO 4217 three-digit currency code. e.g. "344" – HKD
threeDSGiftCardCount	Text (2)	No	Number of gift card / prepaid card purchasing. e.g. "12" – Purchasing 12 gift cards
Pre-Order Purchase Related (Provide only if the payment is related to Pre-Order)			
threeDSPreOrderReason	Text (2)	No	The pre-order reasons. "01" – The pre-order is due to merchant availability, i.e. the purchasing product is not available due to the insufficient stock in the merchant. "02" – The pre-order is for future release product, i.e. the purchasing product is going to be released on future date.
threeDSPreOrderReadyDate	Text (8)	No	The expected ready date of the pre-order. Format: YYYYMMDD e.g. "20190401"
Account Info Related (Provide only if there is user account in merchant)			
threeDSAcctCreateDate	Text(8)	No	Date that the cardholder opened the account. Format: YYYYMMDD e.g. "20190401"
threeDSAcctAgeInd	Text(2)	No	Length of time that the cardholder has had the account. • 01 = No account (guest check-out) • 02 = Created during this transaction • 03 = Less than 30 days • 04 = 30−60 days • 05 = More than 60 days
threeDSAcctLastChangeDate	Text(8)	No	Date that the cardholder's account was last changed, including Billing or Shipping address, new payment account, or new user(s) added. Format: YYYYMMDD e.g. "20190401"
threeDSAcctLastChangeInd	Text(2)	No	Length of time since the cardholder's account information was last changed, including Billing or

			Shipping address, new payment account, or new user(s) added. • 01 = Changed during this transaction • 02 = Less than 30 days • 03 = 30−60 days • 04 = More than 60 days
threeDSAcctPwChangeDate	Text(8)	No	Date that cardholder's account had a password change or account reset. Format: YYYYMMDD e.g. "20190401"
threeDSAcctPwChangeInd	Text(2)	No	Indicates the length of time since the cardholder's account had a password change or account reset. • 01 = No change • 02 = Changed during this transaction • 03 = Less than 30 days • 04 = 30−60 days • 05 = More than 60 days
threeDSAcctPurchaseCount	Text(4)	No	Number of purchases with this cardholder account during the previous six months. e.g. "13" – Purchased 13 times during the previous six months
threeDSAcctCardProvisionAttempt	Text(3)	No	Number of Add Card attempts in the last 24 hours. e.g. "2" – Tried add card 2 times
threeDSAcctNumTransDay	Text(3)	No	Number of transactions (successful and abandoned) for this cardholder account across all payment accounts in the previous 24 hours. e.g. "2" – Processed 2 transactions
threeDSAcctNumTransYear	Text(3)	No	Number of transactions (successful and abandoned) for this cardholder account across all payment accounts in the previous year. e.g. "10" – Processed 10 transactions
threeDSAcctPaymentAcctDate	Text(8)	No	Date that the payment account was enrolled in the cardholder's account. Format: YYYYMMDD e.g. "20190401"
threeDSAcctPaymentAcctInd	Text(2)	No	Indicates the length of time that the payment

			account was enrolled in the cardholder's account. • 01 = No account (guest check-out) • 02 = During this transaction • 03 = Less than 30 days • 04 = 30−60 days • 05 = More than 60 days
threeDSAcctShippingAddrLastChangeDate	Text(8)	No	Date when the shipping address used for this transaction was first used. Format: YYYYMMDD e.g. "20190401"
threeDSAcctShippingAddrLastChangeInd	Text(2)	No	Indicates when the shipping address used for this transaction was first used • 01 = This transaction • 02 = Less than 30 days • 03 = 30−60 days • 04 = More than 60 days
threeDSAcctIsShippingAcctNameSame	Text(1)	No	Indicates if the Cardholder Name on the account is identical to the shipping Name used for this transaction. "T" – Account name identical to shipping name "F" – Account name different than shipping name
threeDSAcctIsSuspiciousAcct	Text(1)	No	Indicates whether has experienced suspicious activity (including previous fraud) on the cardholder account. "T" – Suspicious activity has been observed "F" – No suspicious activity has been observed
Account Authentication Info Related (Provide only if there is user account in merchant)			
threeDSAcctAuthMethod	Text(2)	No	Mechanism used by the Cardholder to authenticate. • 01 = No authentication occurred (i.e. cardholder "logged in" as guest) • 02 = Login to the cardholder account at the merchant system using merchant's own credentials • 03 = Login to the cardholder account at the merchant system using federated ID

			• 04 = Login to the cardholder account at the merchant system using issuer credentials • 05 = Login to the cardholder account at the merchant system using third-party authentication • 06 = Login to the cardholder account at the merchant system using FIDO Authenticator
threeDSAcctAuthTimestamp	Text(8)	No	Date and time in UTC of the cardholder authentication. Format: YYYYMMDD e.g. “20190401”
Pay Token Related (Provide only if the card info is de-tokenized from pay token)			
threeDSPayTokenInd	Text(1)	No	Indicates that the transaction was de-tokenized. * Only provide when de-tokenized happened “T” – The card info is de-tokenized from pay token

Redirect URL (successUrl, failUrl and cancelUrl) Output		
Ref	Text	Merchant’s Order Reference Number **(For display purpose only. DO NOT use this URL to update your system. Please use DataFeed for this purpose.)**

Example of Client Post Method (Source Code)

The following is an example of integration of shopping cart routine with the payment routine of PayDollar PayGate in HTML. It is noteworthy that the portion in bold typeface as follows is mandatory for successful integration.

In the following sample form, hidden fields are used to hold the values:

```
...
<form name="payFormCcard" method="post" action="
        https://test.paydollar.com/b2cDemo/eng/payment/payForm.jsp">
<input type="hidden" name="merchantId" value="1">
<input type="hidden" name="amount" value="3000" >
<input type="hidden" name="orderRef" value="000000000014">
<input type="hidden" name="currCode" value="344" >
<input type="hidden" name="mpsMode" value="NIL" >
<input type="hidden" name="successUrl"
        value="http://www.yourdomain.com/Success.html">
<input type="hidden" name="failUrl" value="http://www.yourdomain.com/Fail.html">
<input type="hidden" name="cancelUrl"
value="http://www.yourdomain.com/Cancel.html">
<input type="hidden" name="payType" value="N">
```

```
<input type="hidden" name="lang" value="E">
<input type="hidden" name="payMethod" value="CC">
<input type="hidden" name="secureHash"
        value="44f3760c201d3688440f62497736bfa2aadd1bc0">
<input type="submit" name="submit">
</form>
...
```

Kick Off

After the integration has been completed, it is ready to launch your e-commerce web to serve your customers. Please copy the following **TESTING URL** for client post method:

---

https://test.paydollar.com/b2cDemo/eng/payment/payForm.jsp

Please copy the following **PRODUCTION URL** for client post method:

---

https://www.paydollar.com/b2c2/eng/payment/payForm.jsp

2.2 Direct Client Side Connection

This method is used for the merchant if they want to capture the credit card information from their web page instead of using our standard payment page. The requirement of using this method is to install a SSL Certificate to your domain in order to protect your customers' credit card information.

Moreover, if the credit card used by the customer is an enrolled 3-D Secure card, the customer will be asked for providing a static password or one-time password to verify the payer identity. 3-D Secure is a credit card authorization program implemented by VISA with brand named "Verified By VISA", MasterCard with brand named "MasterCard SecureCode", JCB with brand named "J/Secure" and AMEX with brand named "AMEX SafeKey" to reduce fraudulent purchases by verifying purchaser identity during online transactions. PayDollar will assist to carry out this process and the customer will observe the 3D processing pages by our PayDollar shown as the later section.

As the 3D protocol is standardized for all brand types, including Verified By VISA, MasterCard SecureCode, JCB J/Secure and AMEX SafeKey. In this document, we use the case of Verified By VISA as an example to show the flow in detail.

For merchant who chooses this method of connection, 128-bit SSL sever certificate must be installed for data encryption. The system does not accept non-encrypted data.

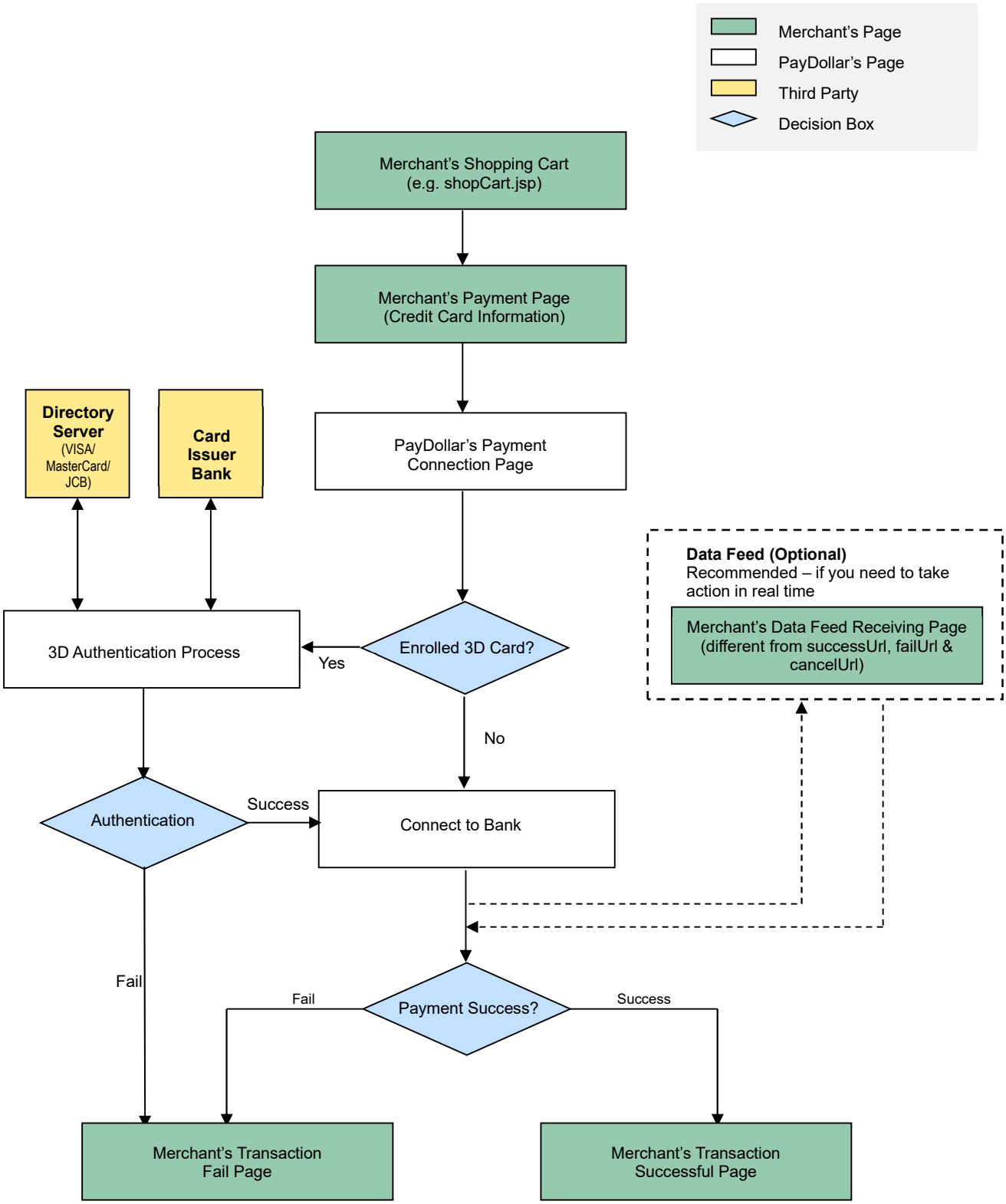
PayDollar use Extended Validation (EV) SSL Certificate to ensure your system function properly, please check your certificate store can recognize VeriSign intermediate CA certificate - Secure Site Pro/Managed PKI for SSL Premium with EV Certificates. If not, you are required to install the VeriSign intermediate CA certificate in your certificate store.

Please download the primary and secondary VeriSign EV SSL Intermediate CA certificates from the following link then import the 2 certificates into the keystore of your environment.

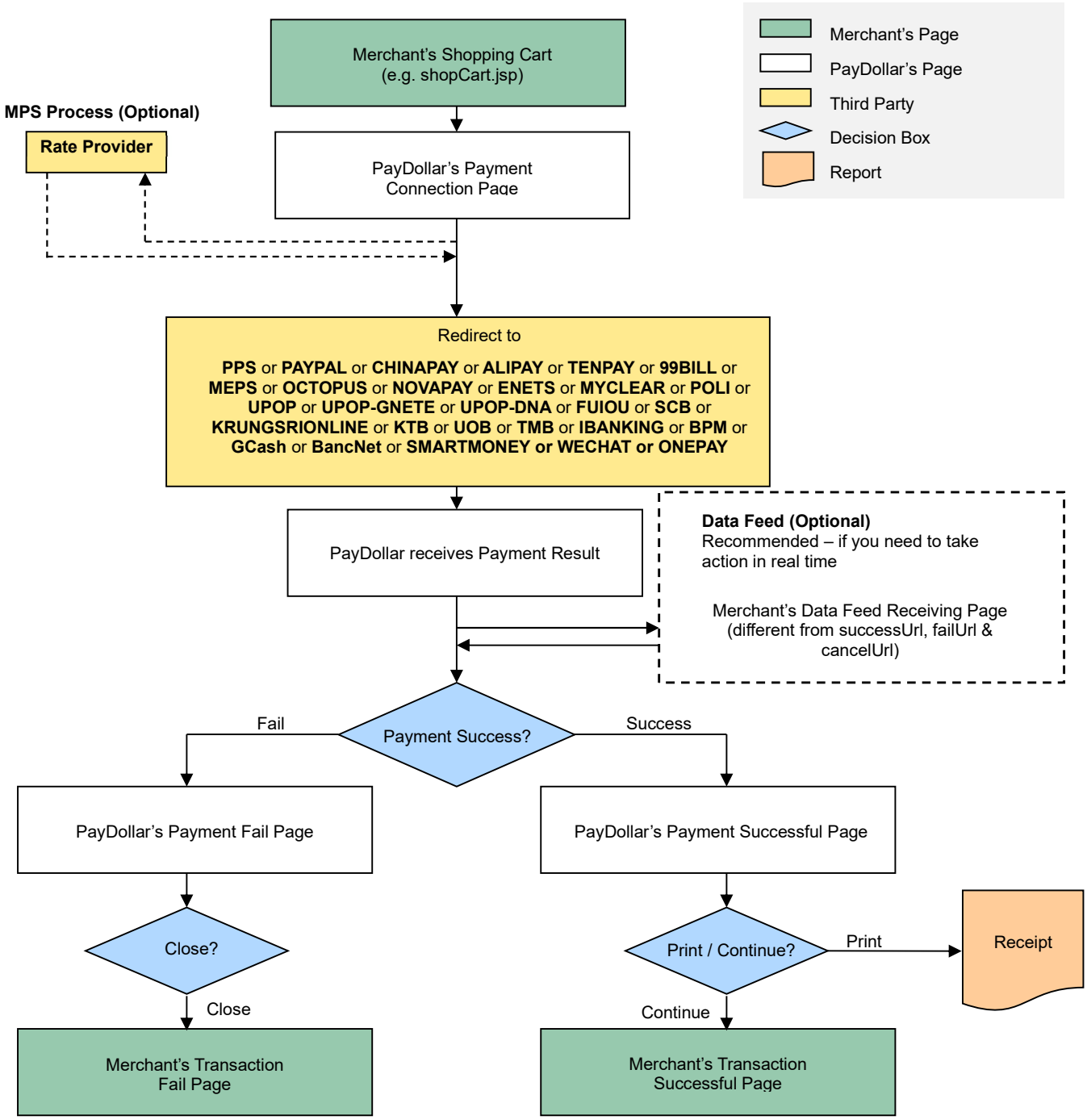
http://www.verisign.com/support/verisign-intermediate-ca/extended-validation-pro/index.html

(Please be reminded that you should choose the option "Issued After May 17th, 2009")

Credit and Debit Card Payment Flow for VISA, Master, Diners, JCB, AMEX



Other Payment Flow



Non-3D transaction

Your client's browser will be redirected from your site to our payment page and then we will redirect the page to your successful/fail page upon completed the transaction.

3D transaction

As 3D Authentication require your customers to enter the password of their cards, your clients' browser will be redirected to a 3D notification web page in order to notify your customers that they need to complete the 3D Authentication by entering the password in the pop-up window. Below are some sample pages for the case of Verified By VISA.

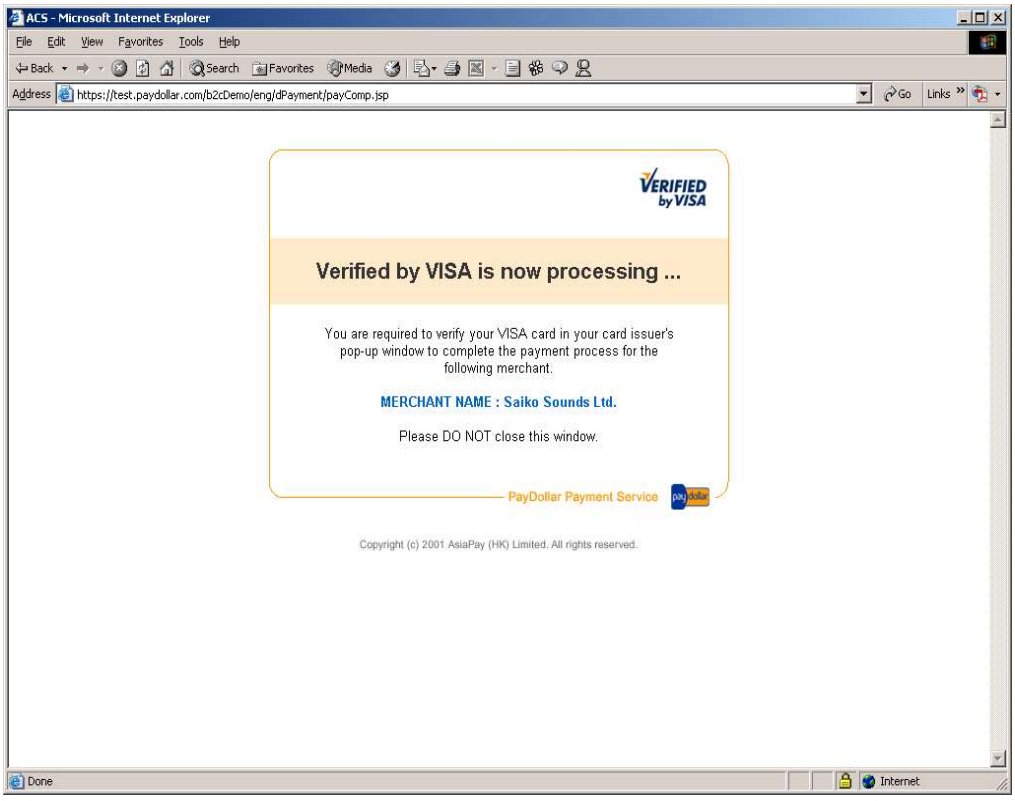


Figure 1.1 Sample notification page

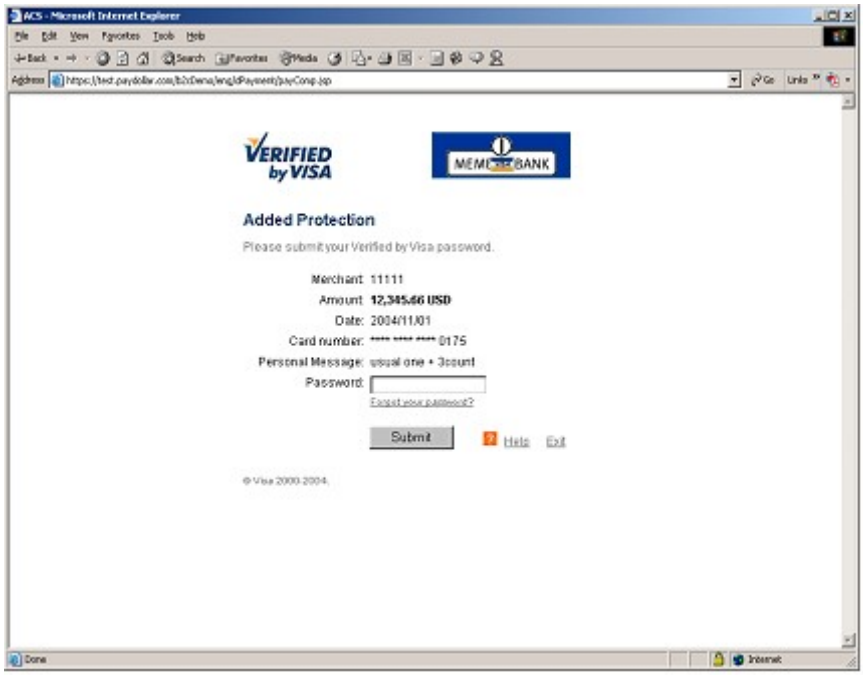


Figure 1.2 Sample issuing bank verification page

After the authentication process, the authentication result will forward to our system and the transaction process will be further continued by our acquiring bank according to the authentication result.

Integration Procedures

To connect to our system, you need to post the required parameters to our payment page URL and then get back the result by using data feed.

Definition of Parameters in the Integration Page

In the targeted page of integration, in which **sum of purchase** has been generated, the following fields (hidden or text) should be added:

Parameters	Data Type	Descriptions
Required Parameter (with UTF-8 Encoding) for connect to our payment page		
orderRef	Text (35)	Merchant's Order Reference Number
amount	Number (12,2)	The total amount you want to charge the customer (up to 2 decimal place) **Remark:** mpsMode = MCP, the amount should be calculated in base currency.
currCode	Text (3)	The currency of the payment: "344" – HKD "840" – USD "702" – SGD "156" – CNY (RMB) "392" – JPY "901" – TWD "036" – AUD "978" – EUR "826" – GBP "124" – CAD "446" – MOP "608" – PHP "764" – THB "458" – MYR "360" – IDR "410" – KRW "682" – SAR "554" – NZD "784" – AED "096" – BND "704" – VND "356" – INR **Remark:** mpsMode = MCP, the currCode value should be in base currency.
lang	Text (1)	The language of the payment page : "E" – English "C" – Traditional Chinese "X" – Simplified Chinese "J" – Japanese "T" – Thai "F" – French "G" – German "R" – Russian "S" – Spanish "V" – Vietnamese
merchantId	Number	The merchant ID we provide to you
pMethod	Text	The payment type ("VISA", "Master", "Diners", "JCB", "AMEX", PPS", "PAYPAL", "CHINAPAY", "ALIPAY", "TENPAY", "99BILL", "MEPS", "OCTOPUS", "NOVAPAY", "ENETS", "MYCLEAR", "POLI", "UPOP", "UPOP-GNETE" ,"UPOP-DNA",

		"FUIOU", "SCB", "KRUNGSRIONLINE", "KTB", "UOB", "TMB", "IBANKING", "BPM", "GCash", "BancNet", "SMARTMONEY, “M2U”, “CIMBCLICK”, “WECHAT”, “ONEPAY”, “VCO”, “WELEND”)
epMonth	Number(2)	Credit card expiry month (mandatory for card payment)
epYear	Number(4)	Credit card expiry year (mandatory for card payment)
cardNo	Text (16)	Credit card number (mandatory for card payment)
securityCode	Text (4)	Credit Card Verification Code (mandatory for card payment) - VISA: CVV2 (3-digit) - MasterCard: CVC2 (3-digit) - JCB: CAV2 (3-digit) - American Express: 4DBC (4-digit)
cardHolder	Text (20)	Credit card holder name (mandatory for card payment)
failUrl	Text (300)	A Web page address you want us to redirect upon the transaction being rejected by us **(For display purpose only. DO NOT use this URL to update your system. Please use DataFeed for this purpose.)**
successUrl	Text (300)	A Web page address you want us to redirect upon the transaction being accepted by us **(For display purpose only. DO NOT use this URL to update your system. Please use DataFeed for this purpose.)**
errorUrl	Text (300)	A Web page address you want us to redirect when unexpected error occur (e.g. parameter incorrect) **(For display purpose only. DO NOT use this URL to update your system. Please use DataFeed for this purpose.)**
payType	Text (1) (“N”,”H”)	The payment type: “N” – Normal Payment (Sales) “H” – Hold Payment (Authorize only) For merchants who use authorize mode, please be reminded to perform the CAPTURE action as soon as the transaction is confirmed as valid. Once captured, the customer's credit card will be debited in coming bank settlement processing. If the merchant does not capture/reverse the authorized transaction over 14 days, the credit limit will be released to the cardholder after a time period which is subjected to card issuing bank Merchant may capture/reverse the authorized transaction in the merchant administration site > Operation > Transaction Detail. Remark: Hold Payment is not available for PPS, PAYPAL, CHINAPAY, ALIPAY, TENPAY, 99BILL, MEPS, OCTOPUS, NOVAPAY, ENETS, MYCLEAR, POLI, UPOP, UPOP-GNETE, UPOP-DNA, FUIOU, SCB,

		KRUNGSRIONLINE, KTB, UOB, TMB, IBANKING, BPM, GCash, BancNet, SMARTMONEY, M2U, CIMBCLICK, WECHAT
Optional Parameter for installment		
installment_service	Text (1) (“T”,”F”)	Installment service indicator
installment_period	Text (20)	In number of months *Multiple period with separator “|” only for payment method “WELEND“
Optional Parameter for airline data		
airline_service	Text (1) (“T”,”F”)	Airline service indicator
airline_ticketNumber	Text (13)	Air ticket number
Optional Parameter for billing information		
billingFirstName	Text(60)	First name of customer
billingLastName	Text(60)	Last name of customer
billingStreet1	Text(40)	Address of customer
billingStreet2	Text(40)	Address of customer ,only mandatory if address exceed 40
billingCity	Text(50)	City
billingState	Text(2)	Mandatory if customer’s country is USA or Canada
billingPostalCode	Text(10)	Mandatory if customer’s country is USA or Canada
billingCountry	Text(2)	Eg.HK
billingEmail	Text(255)	Email address
custIPAddress	Text(15)	192.168.180.100
Optional Parameter for connect to our payment page		
remark	Text (50)	An additional remark field that will appear in the confirmation email and transaction detail report to help you to refer the order
oriCountry	Number(3)	Origin Country Code Example: 344 – “HK” 840 – “US”
destCountry	Number(3)	Destination Country Code Example: 344 – “HK” 840 – “US”
secureHash	Text (40)	Secure hash is used to authenticate the integrity of the transaction information and the identity of the merchant. It is calculated by hashing the combination of various transaction parameters and the Secure Hash Secret. *Applies to merchants who registered this function only. For more information, please refer to section 4.
timeoutValue	Number	Timeout Value in second, only for OCTOPUS payment and ranged 180 and 7200

payMode	Text(2)	Payment Mode "EC" - Electronic Commerce (Default) "MO" - Mail Order "RC" - Recurring
orderPhoneNo	Text(30)	Phone Number of order
Optional Parameter for Multi Currency Pricing (MCP)		
mpsMode	Text(3)	The Multi-Currency Processing Service (MPS) Mode: "MCP" – Enable MPS with 'Multi Currency Pricing' For merchant who applied MPS function
multiRateID	Text (12)	Unique Rate ID *Returned from API function "MCPMultiRateEnquiry"
mpsExRate	Number (10,4)	Exchange rate between the merchant's base currency and the foreign currency. *Returned from API function "MCPMultiRateEnquiry"
foreignCurrCode	Text (3)	Foreign currency code
foreignAmount	Number (12,2)	Calculated foreign amount ***The foreign amount should be converted from a base amount with minimum value HKD10 / MOP10 to avoid tolerance check failure.**
Optional Parameter for using third party EWallet		
eWalletService	Text ("T","F")	EWallet service indicator
eWalletBrand	Text (10)	The value of the "eWalletBrand": "MP" – MasterPass "ECO" – AMEX ExpressCheckout "VCO" – VISA Checkout
eWalletInd	Text	Only Text(3) for MasterPass EWallet, The Wallet Indicator is returned by the wallet platform and must be passed to the payment processor/acquirer in the financial authorization transaction.
vcoCallid	Text	Value of "callid" returned from VISA Checkout * Only for "VISA Checkout Direct Integration"
Optional Parameters for card data encryption		
encryptMode	Text	Encryption Mode: "WEB" – For online transaction
encryptAlgo	Text	Encrypted Algorithm: -"AES/CBC/PKCS5Padding" For "WEB"
encryptedData	Text	Other customized encrypted data (For partner only)

		For encrytedMode is "WEB", please follow this format cardNo|expDate|cardHolder|securityCode before encryption

Parameters for Supporting 3DS 2.0

The following are the parameters especially for 3DS 2.0. Merchant shall fulfill the parameters requirement to start using 3DS 2.0. The parameters are case sensitive and should be in UTF-8 encoding.

Parameters	Data Type	Mandatory	Descriptions
Basic Parameters			
threeDSTransType	Text (2)	No	Transaction type. Select the most suitable one. "01" – Goods/ Service Purchase ** Default value if threeDSTransType not provided.* "03" – Check Acceptance "10" – Account Funding "11" – Quasi-Cash Transaction "28" – Prepaid Activation and Load
threeDSCustomerEmail	Text (254)	No	The customer's email address * Strongly suggest to provide if available. Shall meet requirements of Section 3.4 of IETF RFC 5322.
threeDSMobilePhoneCountryCode	Text (3)	No	Customer's mobile phone number country code. Provide if available. If "threeDSMobilePhoneCountryCode" is provided, "threeDSMobilePhoneNumber" should also be provided. e.g. "852" – Hong Kong
threeDSMobilePhoneNumber	Text (15)	No	Customer's mobile phone number. Provide if available. If "threeDSMobilePhoneNumber" is provided, "threeDSMobilePhoneCountryCode" should also be provided.
threeDSHomePhoneCountryCode	Text (3)	No	Customer's home phone number country code. Provide if available. If "threeDSHomePhoneCountryCode" is provided, "threeDSHomePhoneNumber" should also be provided.

			e.g. “852” – Hong Kong
threeDSHomePhoneNumber	Text (15)	No	Customer’s home phone number. Provide if available. If “threeDSHomePhoneNumber” is provided, “threeDSHomePhoneCountryCode” should also be provided.
threeDSWorkPhoneCountryCode	Text (3)	No	Customer’s work phone number country code. Provide if available. If “threeDSWorkPhoneCountryCode” is provided, “threeDSWorkPhoneNumber” should also be provided. e.g. “852” – Hong Kong
threeDSWorkPhoneNumber	Text (15)	No	Customer’s work phone number. Provide if available. If “threeDSWorkPhoneNumber” is provided, “threeDSWorkPhoneCountryCode” should also be provided.
threeDSIsFirstTimeItemOrder	Text (1)	No	Flag to show if the customer is re-ordering the item / product. i.e. purchasing the same item / product Provide only if the payment related to purchase item / product. “T” – It is the first time the customer purchases this item / product. “F” – It is the NOT the first time the customer purchases the item / product, it is re-ordering.
threeDSChallengePreference	Text(2)	No	Indicates whether a challenge is requested for this transaction. “01” – No preference ** Default value if threeDSChallengePreference not provided.* “02” – No challenge requested * “03” – Challenge requested (Merchant preference) “04” – Challenge requested (Mandate) “05” – No challenge requested (transactional risk analysis is already performed) * “06” – No challenge requested (Data share only)*

			"07" – No challenge requested (strong consumer authentication is already performed) * "08" – No challenge requested (utilise whitelist exemption if no challenge required) * "09" – Challenge requested (whitelist prompt requested if challenge required) **** If "No challenge requested" options are selected, the chargeback liability shift to merchant.**
Recurring / Installment Payment Related (Provide only if it is a recurring / installment payment)			
threeDSRecurringFrequency	Text (4)	Yes (Only for recurring / installment payment)	Minimum number of days between payment. e.g. "30" – 30 days between payments.
threeDSRecurringExpiry	Text (8)	Yes (Only for recurring / installment payment)	Date after which no further payment shall be performed. Format: YYYYMMDD e.g. "20190401"
Billing Address Related (Provide only if billing address is available)			
threeDSBillingCountryCode	Text (3)	Yes (Only if billing address is available)	Payment cardholder billing country. Shall be the ISO 3166-1 numeric three-digit country code. e.g. "344" – Hong Kong, "840" – US
threeDSBillingState	Text (3)	No	Payment cardholder billing state Should be the country subdivision code defined in ISO 3166-2. Provide if available, as some countries do not have subdivision code. e.g. "PE" – Prince Edward Island of California
threeDSBillingCity	Text (50)	Yes (Only if billing address is available)	Payment cardholder billing city e.g. "Hong Kong"
threeDSBillingLine1	Text (50)	Yes (Only if billing address is	Payment cardholder billing address line 1

		available)	
threeDSBillingLine2	Text (50)	No	Payment cardholder billing address line 2 Provide if available
threeDSBillingLine3	Text (50)	No	Payment cardholder billing address line 3 Provide if available
threeDSBillingPostalCode	Text (16)	No	Payment cardholder billing postal code Provide if available
Shipping / Delivery Related (Provide only if the payment requires shipping / delivery)			
threeDSDeliveryTime	Text (2)	No	The delivering time for this purchase. "01" – Electronic Delivery "02" – Same day shipping "03" – Overnight shipping "04" – Two-day or more shipping
threeDSDeliveryEmail	Text (254)	No	The email recipient of the purchased product. Provide only if the purchased product is using electronic delivery.
threeDSShippingDetails	Text (2)	No	Shipping details / type. Select the most suitable one. "01" – Ship to cardholder's billing address "02" – Ship to another verified address stored in merchant "03" – Ship to address that is different than the cardholder's billing address "04" – Ship to Store / Pick-up at local store (Store address shall be populated in shipping address fields) "05" – Digital goods (includes online services, electronic gift cards and redemption codes) "06" – Travel and Event tickets, not shipped "07" – Other (for example, Gaming, digital services not shipped, e-media subscriptions, etc.)
threeDSShippingCountryCode	Text (3)	No	Shipping / delivery country. Shall be the ISO 3166-1 numeric three-digit country code. e.g. "344" – Hong Kong, "840" – US
threeDSShippingState	Text (3)	No	Shipping / delivery state Should be the country subdivision code defined

			in ISO 3166-2 e.g. “PE” – Prince Edward Island of California
threeDSShippingCity	Text (50)	No	Shipping / delivery city. e.g. “Hong Kong”
threeDSShippingLine1	Text (50)	No	Shipping / delivery address line 1
threeDSShippingLine2	Text (50)	No	Shipping / delivery address line 2 Provide (if available).
threeDSShippingLine3	Text (50)	No	Shipping / delivery address line 3 Provide (if available).
threeDSShippingPostalCode	Text (16)	No	Shipping / delivery postal code Provide (if available).
threeDSIsAddrMatch	Text (1)	No	Flag for comparing the billing address and shipping address. Provide (if available). “T” – Shipping address matches billing address “F” – Shipping address NOT match billing address
Gift Card / Prepaid Card Purchase Related (Provide only if the purchase related to gift card / prepaid card)			
threeDSGiftCardAmount	Text (15)	No	Gift card / prepaid card purchase amount.
threeDSGiftCardCurr	Text (3)	No	Gift card / prepaid card purchase currency. ISO 4217 three-digit currency code. e.g. “344” – HKD
threeDSGiftCardCount	Text (2)	No	Number of gift card / prepaid card purchasing. e.g. “12” – Purchasing 12 gift cards
Pre-Order Purchase Related (Provide only if the payment is related to Pre-Order)			
threeDSPreOrderReason	Text (2)	No	The pre-order reasons. “01” – The pre-order is due to merchant availability, i.e. the purchasing product is not available due to the insufficient stock in the merchant. “02” – The pre-order is for future release product, i.e. the purchasing product is going to be released on future date.
threeDSPreOrderReadyDate	Text (8)	No	The expected ready date of the pre-order. Format: YYYYMMDD e.g. “20190401”
Account Info Related (Provide only if there is user account in merchant)			

threeDSAcctCreateDate	Text(8)	No	Date that the cardholder opened the account. Format: YYYYMMDD e.g. “20190401”
threeDSAcctAgeInd	Text(2)	No	Length of time that the cardholder has had the account. • 01 = No account (guest check-out) • 02 = Created during this transaction • 03 = Less than 30 days • 04 = 30−60 days • 05 = More than 60 days
threeDSAcctLastChangeDate	Text(8)	No	Date that the cardholder’s account was last changed, including Billing or Shipping address, new payment account, or new user(s) added. Format: YYYYMMDD e.g. “20190401”
threeDSAcctLastChangeInd	Text(2)	No	Length of time since the cardholder’s account information was last changed, including Billing or Shipping address, new payment account, or new user(s) added. • 01 = Changed during this transaction • 02 = Less than 30 days • 03 = 30−60 days • 04 = More than 60 days
threeDSAcctPwChangeDate	Text(8)	No	Date that cardholder’s account had a password change or account reset. Format: YYYYMMDD e.g. “20190401”
threeDSAcctPwChangeInd	Text(2)	No	Indicates the length of time since the cardholder’s account had a password change or account reset. • 01 = No change • 02 = Changed during this transaction • 03 = Less than 30 days • 04 = 30−60 days • 05 = More than 60 days
threeDSAcctPurchaseCount	Text(4)	No	Number of purchases with this cardholder account during the previous six months.

			e.g. “13” – Purchased 13 times during the previous six months
threeDSAcctCardProvisionAttempt	Text(3)	No	Number of Add Card attempts in the last 24 hours. e.g. “2” – Tried add card 2 times
threeDSAcctNumTransDay	Text(3)	No	Number of transactions (successful and abandoned) for this cardholder account across all payment accounts in the previous 24 hours. e.g. “2” – Processed 2 transactions
threeDSAcctNumTransYear	Text(3)	No	Number of transactions (successful and abandoned) for this cardholder account across all payment accounts in the previous year. e.g. “10” – Processed 10 transactions
threeDSAcctPaymentAcctDate	Text(8)	No	Date that the payment account was enrolled in the cardholder’s account. Format: YYYYMMDD e.g. “20190401”
threeDSAcctPaymentAcctInd	Text(2)	No	Indicates the length of time that the payment account was enrolled in the cardholder’s account. • 01 = No account (guest check-out) • 02 = During this transaction • 03 = Less than 30 days • 04 = 30−60 days • 05 = More than 60 days
threeDSAcctShippingAddrLastChangeDate	Text(8)	No	Date when the shipping address used for this transaction was first used. Format: YYYYMMDD e.g. “20190401”
threeDSAcctShippingAddrLastChangeInd	Text(2)	No	Indicates when the shipping address used for this transaction was first used • 01 = This transaction • 02 = Less than 30 days • 03 = 30−60 days • 04 = More than 60 days
threeDSAcctIsShippingAcctNameSame	Text(1)	No	Indicates if the Cardholder Name on the account is identical to the shipping Name used for this transaction.

			"T" – Account name identical to shipping name "F" – Account name different than shipping name
threeDSAcctIsSuspiciousAcct	Text(1)	No	Indicates whether has experienced suspicious activity (including previous fraud) on the cardholder account. "T" – Suspicious activity has been observed "F" – No suspicious activity has been observed
Account Authentication Info Related (Provide only if there is user account in merchant)			
threeDSAcctAuthMethod	Text(2)	No	Mechanism used by the Cardholder to authenticate. • 01 = No authentication occurred (i.e. cardholder "logged in" as guest) • 02 = Login to the cardholder account at the merchant system using merchant's own credentials • 03 = Login to the cardholder account at the merchant system using federated ID • 04 = Login to the cardholder account at the merchant system using issuer credentials • 05 = Login to the cardholder account at the merchant system using third-party authentication • 06 = Login to the cardholder account at the merchant system using FIDO Authenticator
threeDSAcctAuthTimestamp	Text(8)	No	Date and time in UTC of the cardholder authentication. Format: YYYYMMDD e.g. "20190401"
Pay Token Related (Provide only if the card info is de-tokenized from pay token)			
threeDSPayTokenInd	Text(1)	No	Indicates that the transaction was de-tokenized. * Only provide when de-tokenized happened "T" – The card info is de-tokenized from pay token

Example of connecting to our gateway (Direct Client Side Connection)

As different type of programming language have different syntax. Therefore, the sample code below, is written in HTML code, the requirement is to form post all the required parameters to our secure API, highlighted in yellow.

Sample code:

```
. . .
<form name="payForm" method="post" action="https://test.paydollar.com/b2cDemo/eng/dPayment/payComp.jsp">
<input type="hidden" name="merchantId" value="1">
<input type="hidden" name="amount" value="3000" >
<input type="hidden" name="orderRef" value="000000000006">
<input type="hidden" name="currCode" value="344" >
<input type="hidden" name="pMethod" value="VISA" >
<input type="hidden" name="cardNo" value="4918914107195005" >
<input type="hidden" name="securityCode" value="123" >
<input type="hidden" name="cardHolder" value="Testing" >
<input type="hidden" name="epMonth" value="07" >
<input type="hidden" name="epYear" value="2015" >
<input type="hidden" name="payType" value="N" >
<input type="hidden" name="successUrl" value="http://www.yourwebsite.com/pSuccess.jsp">
<input type="hidden" name="failUrl" value="http://www.yourwebsite.com/pFail.jsp">
<input type="hidden" name="errorUrl" value="http://www.yourwebsite.com/pError.jsp">
<input type="hidden" name="lang" value="E">
<input type="hidden" name="secureHash" value="44f3760c201d3688440f62497736bfa2aadd1bc0">
<input type="submit" value="Pay Now">
</form>
. . .
```


Kick Off

After the integration has been completed, it is ready to launch your e-commerce web to serve your customers. Please copy the following **TESTING URL** for client post method:

---

https://test.paydollar.com/b2cDemo/eng/dPayment/payComp.jsp

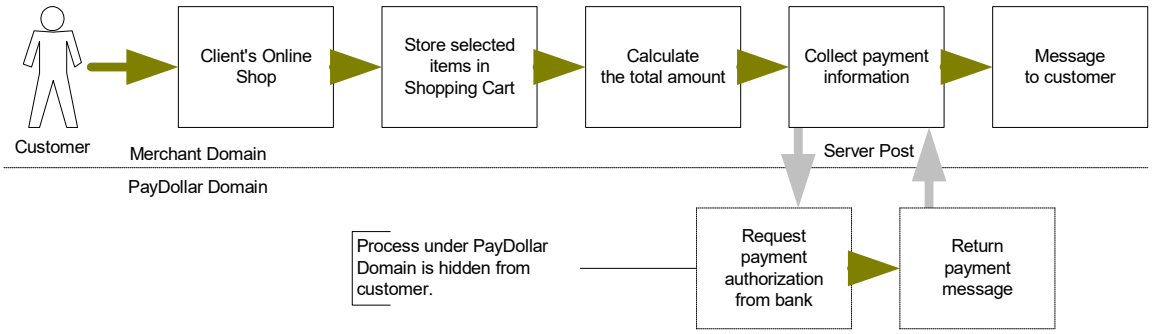
Please copy the following **PRODUCTION URL** for client post method:

---

https://www.paydollar.com/b2c2/eng/dPayment/payComp.jsp

2.3 Server Side Direct Connection

This connection method is for merchant to request payment authorization from bank directly through PayDollar PayGate system and subject to approval of acquiring bank. For example, merchant's IVR system or mobile application can directly integrate to us. And in this connection, merchants need to build their own payment information collection page to collect payment information, such as credit card number, expire data, holder's name and etc. Then, payment information has to be sent to a defined URL provided by the acquiring bank. Customer of the merchant, therefore, will not see any bank's payment page.



For merchant who chooses this method of connection, 128-bit SSL cert must be installed for data encryption. The system does not accept non-encrypted data.

PayDollar uses Extended Validation (EV) SSL Certificate. To ensure your system function properly, please check your certificate store can recognize VeriSign intermediate CA certificate - Secure Site Pro/Managed PKI for SSL Premium with EV Certificates. If not, you are required to install the VeriSign intermediate CA certificate in your certificate store.

Please download the primary and secondary VeriSign EV SSL Intermediate CA certificates from the following link then import the 2 certificates into the keystore of your environment.

http://www.verisign.com/support/verisign-intermediate-ca/extended-validation-pro/index.html

(Please be reminded that you should choose the option "Issued After May 17th, 2009")

We use Server Side Direct Connection for App 2 App integration and wallet payment direct integration. For the detail, please reference with "**Integration Guide (Wallet Direct Connection)**"

Definition of Parameters in the Integration Page

The following are the parameters for integration. PayDollar PayGate is case sensitive. Make sure the typeface is correct. When a transaction is finish, the system will return customer a payment message on the page created by merchant.

Parameters	Data Type	Descriptions
Required Parameter (with UTF-8 Encoding) for connect to our payment interface		
orderRef	Text (35)	Merchant's Order Reference Number
amount	Number (12,2)	Total amount your want to charge the customer [Up to 2 decimal place] **Remark:** mpsMode = MCP, the amount should be calculated in base currency.
currCode	Text (3)	The currency of the payment: "344" – HKD "840" – USD "702" – SGD "156" – CNY (RMB) "392" – JPY "901" – TWD "036" – AUD "978" – EUR "826" – GBP "124" – CAD "446" – MOP "608" – PHP "764" – THB "458" – MYR "360" – IDR "410" – KRW "682" – SAR "554" – NZD "784" – AED "096" – BND "704" – VND "356" – INR **Remark:** mpsMode = MCP, the currCode value should be in base currency.
lang	Text (1)	The language of the payment page : "E" - English
merchantId	Number	The merchant ID we provide to you
pMethod	Text	The payment card type "VCO" – VISA Checkout Card Payment: ("VISA", "Master", "Diners", "JCB", "AMEX") eWallet Payment: "ALIPAYAPP" – Alipay Global App "ALIPAYCNAPP" – Alipay China App "ALIPAYHKAPP" – Alipay HK App "WECHATAPP" – WeChat Pay App "WECHATONL" – WeChat Pay QR "FPS" – FPS App 2 App "OCTOPUS" – Octopus App 2 App / QR
epMonth	Number(2)	Credit card expiry month

epYear	Number(4)	Credit card expiry year
cardNo	Text (16)	Credit card number
cardHolder	Text (20)	Credit card holder name
securityCode	Text (4)	Credit Card Verification Code - VISA: CVV2 (3-digit) - MasterCard: CVC2 (3-digit) - JCB: CAV2 (3-digit) - American Express: 4DBC (4-digit)
payType	Text (1) (“N”,”H”)	The payment type: “N” – Normal Payment (Sales) “H” – Hold Payment (Authorize only) For merchants who use authorize mode, please be reminded to perform the CAPTURE action as soon as the transaction is confirmed as valid. Once captured, the customer's credit card will be debited in coming bank settlement processing. If the merchant does not capture/reverse the authorized transaction over 14 days, the credit limit will be released to the cardholder after a time period which is subjected to card issuing bank Merchant may capture/reverse the authorized transaction in the merchant administration site > Operation > Transaction Detail. Remark: Hold Payment is not available for PPS, PAYPAL, CHINAPAY, ALIPAY, TENPAY, 99BILL, MEPS, OCTOPUS, NOVAPAY, ENETS, MYCLEAR, POLI, UPOP, UPOP-GNETE, UPOP-DNA, FUIOU, SCB, KRUNGSRIONLINE, KTB, UOB, TMB, IBANKING, BPM, GCash, BancNet, SMARTMONEY, M2U, CIMBCLICK, WECHAT, ONEPAY
Optional Parameter for installment		
installment_service	Text (1) (“T”,”F”)	Installment service indicator
installment_period	Number	In number of months
Optional Parameter for airline data		
airline_service	Text (1) (“T”,”F”)	Airline service indicator
airline_ticketNumber	Text (13)	Air ticket number
Optional Parameter for billing information		
billingFirstName	Text(60)	First name of customer
billingLastName	Text(60)	Last name of customer
billingStreet1	Text(40)	Address of customer
billingStreet2	Text(40)	Address of customer ,only mandatory if address exceed 40

billingCity	Text(50)	City
billingState	Text(2)	Mandatory if customer's country is USA or Canada
billingPostalCode	Text(10)	Mandatory if customer's country is USA or Canada
billingCountry	Text(2)	Eg.HK
billingEmail	Text(255)	Email address
custIPAddress	Text(15)	192.168.180.100
Optional Parameter for connect to our payment interface		
remark	Text (50)	An additional remark field that will appear in the confirmation email and transaction detail report to help you to refer the order
secureHash	Text (40)	Secure hash is used to authenticate the integrity of the transaction information and the identity of the merchant. It is calculated by hashing the combination of various transaction parameters and the Secure Hash Secret. *Applies to merchants who registered this function only. For more information, please refer to section 4.
payMode	Text(2)	Payment Mode "EC" - Electronic Commerce (Default) "MO" - Mail Order "RC" - Recurring "RT" - Retail
orderPhoneNo	Text(30)	Phone Number of order
Parameter For 3D Transaction (Need to install MPI Server Software at Merchants' site)		
vbvTransaction	Text ("T","F")	3D Transaction (i.e. Verified By VISA, MasterCard SecureCode, J/Secure, AMEX SafeKey) "T" – True: The transaction has been gone through the VE/PA process **OR performed 3DS 2.0 Authentication.** "F" – False: Non-3D transaction without go through the VE/PA process [No need to send other 3D parameters except **vbvTransECI**] *Note: Bank's approval is required for non-3D transactions, please contact us for further details.
vbvCHReturnCode	Number	Verify Enrollment Return Code - Set to "0" if the *enrolled* value obtained in MPI VE Response Message is "Y" - Set to "1001" if the *enrolled* value obtained in MPI VE Response Message is "N"

		- Set to "-1" if the *enrolled* value is not available in MPI VE Response Message - **Keep empty / null if using 3DS 2.0.**
vbvPAReturnCode	Number	Payer Authentication Return Code - Set to "0" if the *status* value obtained in MPI VE Response Message is "Y" - Set to "1000" if the *status* value obtained in MPI VE Response Message is "A" - Set to "1003" if the *status* value obtained in MPI VE Response Message is "N" - Otherwise set to "-1" - **Keep empty / null if using 3DS 2.0.**
vbvTransTime	Text in format (YYYYMMDD HH:MM:SS)	Transaction Time, *time*, MPI PA Response Message. **If using 3DS 2.0, the receiving time of Authentication Response / Result Request.**
vbvTransAuth	Text (28)	Cardholder Authentication Verification Value, *CAVV,* value in MPI PA Response Message [Must be in Base64-Encoded format] **If using 3DS 2.0, the Authentication Value in Authentication Response / Result Request.**
vbvTransECI	Text (2)	Electronic Commerce Indicator, *ECI*, value in MPI PA Response Message For VISA & JCB card payment: - For Enrolled 3D VISA card [*enrolled* = Y] - Set it to the value of *ECI* obtained from MPI PA Response Message - Set it to "07" if *ECI* value is not available in MPI PA Response Message - For Not an Enrolled 3D VISA card [*enrolled* = N] - Set it to "06" - Otherwise, set it to "07" For MasterCard payment: - For Enrolled 3D MasterCard [*enrolled* = Y] - Set it to the value of *ECI* obtained from MPI PA Response Message - Set it to "00" if *ECI* value is not available in MPI PA Response Message - For Not an Enrolled 3D MasterCard [*enrolled* = N] - Set it to "01" - Otherwise, set it to "00

		For Other credit card payment, set it to "07" **If using 3DS 2.0, the ECI Value in Authentication Response / Result Request.**
vbvCAVVAlgo	Text	CAVV Algorithm, *cavvAlgorithm*, in MPI PA Response Message **Keep empty / null if using 3DS 2.0.**
vbvXID	Text(20)	Transaction Identifier, *xid*, in MPI PA Response Message [Must contain 20 characters] **Keep empty / null if using 3DS 2.0.**
vbvMerchantID	Text	Acquirer-defined Merchant Identifier, *merID*, in MPI PA Response Message **If using 3DS 2.0, the "Acquirer Merchant ID" of Authentication Request.**
vbvAcquirerBin	Text	Acquirer BIN, *acqBIN*, in MPI PA Response Message **If using 3DS 2.0, the "Acquirer BIN" of Authentication Request.**
vbvTransStatus	Text(1)	Transaction Status, *status*, in MPI PA Response Message - Set it to the value of *status* obtained from MPI PA Response Message - Set it to "U" if the *status* value is not available in the MPI PA Response Message - **If using 3DS 2.0, set it to the "Transaction Status" returned in Authentication Response / "Transaction Status" returned in Result Response.**
Optional Parameter for Multi Currency Pricing (MCP)		
mpsMode	Text(3)	The Multi-Currency Processing Service (MPS) Mode: "MCP" – Enable MPS with 'Multi Currency Pricing' For merchant who applied MPS function
multiRateID	Text (12)	Unique Rate ID *Returned from API function "MCPMultiRateEnquiry"
mpsExRate	Number (10,4)	Exchange rate between the merchant's base currency and the foreign currency. *Returned from API function "MCPMultiRateEnquiry"
foreignCurrCode	Text (3)	Foreign currency code
foreignAmount	Number (12,2)	Calculated foreign amount ***The foreign amount should be converted from a base amount with minimum value HKD10 / MOP10 to avoid tolerance check failure.**
Optional Parameter for using third party EWallet		
eWalletService	Text ("T","F")	EWallet service indicator

eWalletBrand	Text (10)	The value of the "eWalletBrand": "MP" – MasterPass "ECO" – AMEX ExpressCheckout "VCO" – VISA Checkout "SAMSUNG" – Samsung Pay "APPLEPAY" – Apple Pay "GOOGLE" – Google Pay™
eWalletPaymentData	Text	Retrieved EWallet data with Base64 Format * Used for Samsung Pay
eWalletInd	Text	Only Text(3) for MasterPass EWallet, The Wallet Indicator is returned by the wallet platform and must be passed to the payment processor/acquirer in the financial authorization transaction.
vcoCallid	Text	Value of "callid" returned from VISA Checkout * Only for "VISA Checkout Direct Integration"
Optional Parameters for card data encryption		
encryptMode	Text	Encryption Mode: "WEB" – For online transaction "POS" – For POS card present transaction
encryptAlgo	Text	Encrypted Algorithm: -"DEsede/CBC/PKCS5Padding" For "POS" -"AES/CBC/PKCS5Padding" For "WEB"
encryptedTrack1Data	Text	Encrypted card track 1 data Format: Initialization vector IV(16) + Encrypted Track 1 Data Note: The maximum record length for Track1 Data is 79 alphanumeric characters. Sample Track1 Data before encryption in plain text: "%B4111111111111111^MR TEST C ABCDE ^18091019681143300001 840 ?;"
encryptedTrack2Data	Text	Encrypted card track 2 data Format: Initialization vector IV(16) + Encrypted Track 2 Data Note: The maximum record length for Track2 Data is 40 alphanumeric characters. Sample Track2 Data before encryption in plain text: "; 4111111111111111=18091019681143384001?"
encryptedEmvData	Text	Encrypted card EMV data read through a chip on the EMV card Format: Initialization vector IV(16) + Encrypted EMV Data
encryptedData	Text	Other customized encrypted data (For partner only) For encrytedMode is "WEB", please follow this format cardNo|expDate|cardHolder|securityCode before encryption

Parameters for Supporting 3DS 2.0

The following are the parameters especially for 3DS 2.0. Merchant shall fulfill the parameters requirement to start using 3DS 2.0. The parameters are case sensitive and should be in UTF-8 encoding.

Parameters	Data Type	Mandatory	Descriptions
Basic Parameters			
threeDSVersion	Text (10)	No	Indicate the 3DS Version if 3D-Secure is processed in this payment. * Should be provided if "vbvTransaction" is "T". "1.0" – Using 3DS 1.0 "2.1.0" / "2.2.0" – Using 3DS 2.0 (depends on the actual message version used in 3DS 2.0 message)
threeDSCustomerIP	Text (45)	No	The customer incoming IP in IPv4 / IPv6 standard. * Suggest to provide for all transactions, if available. e.g. "1.12.123.255" / "2011:0db8:85a3:0101:0101:8a2e:0370:7334"
threeDSAresTransStatus	Text (1)	No	The "transStatus" returned in Authentication Response.
threeDSDsTransId	Text (36)	No (Mandatory if using 3DS 2.0)	If using 3DS 2.0, provide the DS Transaction ID, if available.
threeDSServerTransId	Text (36)	No	If using 3DS 2.0, provide the 3DS Server Transaction ID, if available.
threeDSAcsTransId	Text (36)	No	If using 3DS 2.0, provide the ACS Transaction ID, if available.
threeDSSdkTransId	Text (36)	No	If using 3DS 2.0, provide the SDK Transaction ID, if available.

Return Parameter		
src	Text	Return bank host status code
prc	Text	Return bank host status code
Ord	Text	Bank Reference Number
Holder	Text	The Holder Name of the Payment Account
successcode	Number	Transaction Status: 0 – Transaction succeeded 1 – Transaction Failure
Ref	Text	Merchant's Order Reference Number

PayRef	Number	PayDollar Payment Reference Number
Amt	Number (12,2)	Transaction Amount
Cur	Number (3)	Transaction Currency i.e. “344” - HKD
AuthId	Text	Approval Code
TxTime	Text (YYYY-MM-DD HH:MI:SS.0)	Transaction Time
errMsg	Text	Error Message
encryptedEmvResponse	Text	EMV Data Response for EMV transaction only. Initialization vector IV(16) + Encrypted EMV Response Data

All the return parameters will be concatenated as in html request format by separate with **&**.

Sample return string:

successcode=0&Ref=Test&PayRef=4780&Amt=1.0&Cur=344&prc=0&src=0&Ord=6697090&Holder=edward&AuthId=123456&TxTime=2003-10-07 17:48:02.0&errMsg=Transaction completed

Example of Source Code

As different type of programming language have different syntax, so we just propose the method to connect to our payment page. To connect, we suggest you to use server side posting:

Sample code for server post by using java:

```
// Set up the post data
String postData =
"merchantId=1&orderRef=test&amount=1&currCode=344&pMethod=VISA&epMonth=01&epYear=20
02&cardNo=4123412341234123&cardholder=Edward&remark=test";
// Post to payment page
strResult = ServerPost.post(postData,
        https://www.paydollar.com/b2c2/eng/directPay/payComp.jsp );
// Extract the payment status from strResult
...
// Finish
**********************************************************************
public class ServerPost
{
       static public String post( String ip_postData, String ip_pageUrl)
       {
              try
              {
                     String strResult = "";
                     URL url = new URL(ip_pageUrl);

                     URLConnection con = url.openConnection(); //from secure site
                     if(con instanceof com.sun.net.ssl.HttpsURLConnection){
                        ((com.sun.net.ssl.HttpsURLConnection)con).setSSLSocketFactory
                                    (SSLSocketFactory)SSLSocketFactory.getDefault());
                     }

                     con.setDoOutput(true);
                     con.setDoInput(true);
                     // Set request headers for content type and length
                     con.setRequestProperty(
                            "Content-type",
                            "application/x-www-form-urlencoded");
                     con.setRequestProperty(
                            "Content-length",
                            String.valueOf(ip_postData.length()));
                     // Issue the POST request
                     OutputStream outStream = con.getOutputStream();
                     outStream.write(ip_postData.getBytes());
                     outStream.flush();
                     // Read the response
                     InputStream inStream = con.getInputStream();
```

```
                    while (true)
                     {
                            int c = inStream.read();
                            if (c == -1)
                                   break;
                            strResult = strResult + String.valueOf((char)c);
                     }

                     inStream.close();
                     outStream.close();

                            return strResult;
                }
                catch (Exception e)
                {
                            System.out.print(e.toString());
                            return null;
                }
        }
}
```


Kick Off

After the integration has been completed, it is ready to launch your e-commerce web to serve your customers. Please copy the following **TESTING URL** for Direct Connect Server Post method:

---

https://test.paydollar.com/b2cDemo/eng/directPay/payComp.jsp

Please copy the following **PRODUCTION URL** for Direct Connect Server Post method:

---

https://www.paydollar.com/b2c2/eng/directPay/payComp.jsp

---

3 Data Feed handling

To use data feed function, merchant has to create a data feed page and inform PayDollar about the location of your page (e.g. http://www.yourdomain.com/datafeed.jsp). Merchant can enable or disable this function in the merchant administration site.

Definition of Parameters in the output of Data Feed

<table>
<tr><th>Parameters</th><th>Data Type</th><th>Descriptions</th></tr>
<tr><td colspan="3">Data Feed Output</td></tr>
<tr><td>src</td><td>Text</td><td>Return bank host status code (secondary). Please refer to Appendix A for detail.</td></tr>
<tr><td>prc</td><td>Text</td><td>Return bank host status code (primary). Please refer to Appendix A for detail.</td></tr>
<tr><td>Ord</td><td>Text (40)</td><td>Bank Reference Number</td></tr>
<tr><td>Holder</td><td>Text</td><td>The Holder Name of the Payment Account</td></tr>
<tr><td>successcode</td><td>Number</td><td>0- succeeded, 1- failure, Others - error</td></tr>
<tr><td>Ref</td><td>Text</td><td>Merchant's Order Reference Number</td></tr>
<tr><td>PayRef</td><td>Number</td><td>PayDollar Payment Reference Number</td></tr>
<tr><td>Amt</td><td>Number (12,2)</td><td>Transaction Amount</td></tr>
<tr><td>Cur</td><td>Text (3)</td><td>Transaction Currency i.e.
"344" – HKD "840" – USD "702" – SGD
"156" – CNY (RMB) "392" – JPY "901" – TWD
"036" – AUD "978" – EUR "826" – GBP
"124" – CAD "446" – MOP "608" – PHP
"764" – THB "458" – MYR "360" – IDR
"410" – KRW "682" – SAR "554" – NZD
"784" – AED "096" – BND "704" – VND
"356" – INR</td></tr>
<tr><td>remark</td><td>Text (200)</td><td>A remark field for you to store additional data that will not show on the transaction web page</td></tr>
<tr><td>AuthId</td><td>Text</td><td>Approval Code</td></tr>
<tr><td>eci</td><td>Text (2)</td><td>ECI value (for 3D enabled Merchants)
<table>
<tr><th colspan="2">VISA / JCB / AMEX</th></tr>
<tr><th>ECI Value</th><th>Definition</th></tr>
<tr><td>05</td><td>Both cardholder and card issuing bank are 3D enabled. 3D card authentication is successful</td></tr>
<tr><td>06</td><td>Either cardholder or card issuing bank is not 3D enrolled. 3D card authentication is unsuccessful, in sample situations as:</td></tr>
</table></td></tr>
</table>

		<table><tr><td></td><td>1. 3D cardholder not enrolled 2. Card issuing bank is not 3-D Secure ready</td></tr><tr><td>07</td><td>Authentication is unsuccessful or not attempted. The credit card is either a non-3D card or card issuing bank does not handle it as a 3D transaction</td></tr><tr><td colspan="2">MasterCard</td></tr><tr><td>ECI Value</td><td>Definition</td></tr><tr><td>00</td><td>Authentication is unsuccessful or not attempted. The credit card is either a non-3D card or card issuing bank does not handle it as a 3D transaction</td></tr><tr><td>01</td><td>Either cardholder or card issuing bank is not 3D enrolled. 3D card authentication is unsuccessful, in sample situations as: 1. 3D Cardholder not enrolled 2. Card issuing bank is not 3-D Secure ready</td></tr><tr><td>02</td><td>Both cardholder and card issuing bank are 3D enabled. 3D card authentication is successful</td></tr></table> Remark : Empty String will be sent when the transaction is rejected by PayDollar PayAlert.
payerAuth	Text (1)	Payer Authentication Status Y - Card is 3-D secure enrolled and authentication succeeds. N - Card is 3-D secure enrolled but authentication fails. P - 3-D Secure check is pending A - Card is not 3-D secure enrolled yet U - 3D-secure check is not processed. **If using 3DS 2.0, it is the "Transaction Status" returned in Authentication Response or "Transaction Status" returned in Result Response.** **Y - Authentication/ Account Verification Successful** **N - Not Authenticated /Account Not Verified; Transaction denied** **U - Authentication/ Account Verification Could Not Be Performed** **A - Attempts Processing Performed** **C - Challenge Required** **R - Authentication/ Account Verification Rejected**
sourceIp	Text (15)	IP address of payer
ipCountry	Text (3)	Country of payer (e.g. HK) - if country is on high risk country list, an asterisk will be shown (e.g. MY*)
payMethod	Text (10)	Payment method (e.g. VISA, Master, Diners, JCB, AMEX)
cardIssuingCountry	Text (3)	Card Issuing Country Code (e.g. HK) - if country is on high risk country list, an asterisk will be shown (e.g. MY*)

		- if the card issuing country of credit card is undefined, "- -" will be shown. Please refer to Appendix A "List of Country Code" for detail
channelType	Text (3)	Channel Type: SPC – Client Post Through Browser DPC – Direct Client Side Connection DPS – Server Side Direct Connection SCH – Schedule Payment DPL – Direct Payment Link Connection MOT – Moto Connection RTL – RetailPay Connection BPP – Batch Payment Process MOB – Mobile Payment Connection
secureHash	Text (40)	Secure hash is used to authenticate the integrity of the response information and the identity of PayDollar. It is calculated by hashing the combination of various response parameters and the Secure Hash Secret. *Applies to merchants who registered this function only. For more information, please refer to section 4.
AlertCode	Text (50)	The Alert Code e.g. R14 –IP Country not match with Issuing Country R 9 - In high risk country list
MerchantId	Number	The merchant Id of transaction
airline_ticketNumber	Text	Air Ticket Number
TxTime	Text (YYYY-MM-DD HH:MI:SS.0)	Transaction time
Additional output parameters for card information		
expMonth	Text (MM)	Expiry month of the card *Return to approved merchant only
expYear	Text (YYYY)	Expiry year of the card *Return to approved merchant only
panFull	Text	Encrypted PAN with AES256 encryption *Return to approved merchant only
panFirst6	Text (6)	First 6 digit of card *Return to approved merchant only
panFirst4	Text (4)	First 4 digit of card *Return to approved merchant only

panLast4	Text (4)	Last 4 digit of card *Return to approved merchant only
accountHash	Text	Hash value of card *Return to approved merchant only
accountHashAlgo	Text	Hash function of card: SHA-1 SHA-256 *Return to approved merchant only
Additional output parameters for transaction processing with MPS VAS		
mpsAmt	Number (12,2)	MPS Transaction Amount **Remark: For MPS Enabled only.**
mpsCur	Text (3)	MPS Transaction Currency **Remark: For MPS Enabled only.**
mpsForeignAmt	Number (12,2)	MPS Transaction Foreign Amount **Remark: For MPS Enabled only.**
mpsForeignCur	Text (3)	MPS Transaction Foreign Currency **Remark: For MPS Enabled only.**
mpsRate	Number (12,4)	MPS Exchange Rate: (Foreign / Base) e.g. USD / HKD = 7.77 **Remark: For MPS Enabled only.**
Additional output parameters for transaction processing with SchedulePay VAS		
mSchPayId	Number	The Master Schedule Payment Id *For Schedule payment transaction only
dSchPayId	Number	The Detail Schedule Payment Id *For Schedule payment transaction only
Additional output parameters for transaction processing with MemberPay VAS		
mpMemberId	Text	Member Id if merchant using memberPay
mpLatestStaticToken	Text	Encrypted static token if merchant using static token mode of memberPay
Additional output parameters for transaction processing through Third Party eWallet		
isEwallet	Text(1)	"T" –This is a transaction with Ewallet function
eWalletBrand	Text	ECO - AMEX Express checkout MP – MasterPass VCO – VISA Checkout
eWalletInd	Text	Only Text(3) for MasterPass EWallet, The Wallet Indicator is returned by the wallet platform and must be passed to the payment processor/acquirer in the financial authorization transaction.
Additional output parameters for transaction processed with Promopay VAS		

promotionCode	Text(8);	The promotion campaign code of transaction *For promotion transaction only
promotionRuleCode	Text(8);	The promotion rule code of transaction *For promotion transaction only
promotionOriginalAmt	Number (12,2)	The original amount of promotion *For promotion transaction only
Additional output parameters for installment transaction		
period	Number	The installment period (In number of months)
installmentProvider	Text	The issuing bank or provider of the installment "HASE" – Hang Seng Bank (Hong Kong) "BOCIHK" – Bank of China (Hong Kong) "BCA" - Bank Central Asia (Indonesia) "WELEND" – WeLend (Hong Kong)
firstPayAmt	Number	The transaction amount for first installment period * Return only if the installment bank providing this information
eachPayAmt	Number	The transaction amount for each installment period * Return only if the installment bank providing this information
lastPayAmt	Number	The transaction amount for last installment period * Return only if the installment bank providing this information
Additional output parameters for transaction processed with rewards program (e.g. Hang Seng Cash Dollar)		
netAmtAfterRewards	Number(12,2)	Total Net Amount after Redeem Rewards Program
rewardsRedeemTotalAmt	Number(12,2)	Total Rewards Redeem Amount
rewardsRedeemType	Text	Rewards Redeem Type "HASE" – Hang Seng Cash Dollar
rewardsRedeemProgram	Number	Number of redeem programs Return value is 1,2, or 3
rewardsRedeemCode1	Text	First Rewards Program Redeem Program Code * Return only when rewardsRedeemProgram return value is 1,2 or 3
rewardsRedeemLabel1	Text	First Rewards Program Redeem Program Label * Return only when rewardsRedeemProgram return value is 1,2 or 3
rewardsRedeemAmt1	Number(12,2)	First Rewards Program Redeem Amount * Return only when rewardsRedeemProgram return value is 1,2 or 3
rewardsBalance1	Text	First Rewards Program Balance Format Expired Rewards Balance Expired Date – Expired Balance Amount | Rewards Balance Expired Date- Rewards Balance Amount Example: 0115-20.00|0120-30.00 * Return only when rewardsRedeemProgram return value is 1,2 or 3

rewardsRedeemCode2	Text	Second Rewards Program Redeem Program Code * Return only when rewardsRedeemProgram return value is 2 or 3
rewardsRedeemLabel2	Text	Second Rewards Program Redeem Program Label * Return only when rewardsRedeemProgram return value is 2 or 3
rewardsRedeemAmt2	Number (12,2)	Second Rewards Program Redeem Amount * Return only when rewardsRedeemProgram return value is 2 or 3
rewardsBalance2	Text	Second Rewards Program Balance Format Expired Rewards Balance Expired Date – Expired Balance Amount | Rewards Balance Expired Date- Rewards Balance Amount Example: 0115-20.00|0120-30.00 * Return only when rewardsRedeemProgram return value is 2 or 3
rewardsRedeemCode3	Text	Third Rewards Program Redeem Program Code * Return only when rewardsRedeemProgram return value is 3
rewardsRedeemLabel3	Text	Third Rewards Program Redeem Program Label * Return only when rewardsRedeemProgram return value is 3
rewardsRedeemAmt3	Number(12,2)	Third Rewards Program Redeem Amount * Return only when rewardsRedeemProgram return value is 3
rewardsBalance3	Text	Third Rewards Program Balance Format Expired Rewards Balance Expired Date – Expired Balance Amount | Rewards Balance Expired Date- Rewards Balance Amount Example: 0115-20.00|0120-30.00 * Return only when rewardsRedeemProgram return value is 3
Additional output parameters for 3DS 2.0		
threeDSVersion	Text (10)	Indicate the 3DS Version if 3D-Secure is processed in this payment. “” – Non 3D “1.0” – Using 3DS 1.0 “2.1.0” / “2.2.0” – Using 3DS 2.0 (depends on the actual message version used in 3DS 2.0 message)

The data feed page must meet the following requirement:

- Print ‘**OK**’ in HTML when data captured (ACK message)
- **Make Sure to Print ‘OK’ for acknowledge to our system first then do the rest of your system process, if something wrong with your system process (i.e. download photo, ring tone problem) you can send a void request to our system, for more details please refer to our API guide and contact our technical staff.**

Please note that the system only supports either port 80 (HTTP) or 443 (HTTPS) for the data feed page location. And make sure the data feed page location is externally accessible, so that our server can call the data feed page.

* Since the system will read from the data feed page for the word ‘OK’ to determine whether the (data feed) message is delivered or not, if this word does not return successfully, the system will assume the data feed is lost.

Data Feed Setup

Merchant can enable or disable the data feed function in the merchant administration site > Profile > Payment Options.

Sample Data Feed Page

The following is a sample data feed page in JSP.

```
<%@ page language="java" %>
<%
     String successCode = request.getParameter("successcode");
     String payRef = request.getParameter("PayRef");
     String Ref = request.getParameter("Ref");

     // Print out 'OK' to notify us you have received the payment result
     out.print("OK");

     if ( successCode.equals("0") )
     {
          // Transaction Accepted
          // *** Add the Security Control here, to check the currency, amount with
the
          // *** merchant's order reference from your database, if the order exist
then
          // *** accepted otherwise rejected the transaction.

          //  Update your database for Transaction Accepted and send email or notify
your
          //   customer.
          ....

          // In case if your database or your system got problem, you can send a
          void transaction request. See API guide for more details
     }
     else
     {
          // Transaction Rejected
          // Update your database for Transaction Rejected

          .....
     }

%>
```

The following is a sample data feed page in ASP.

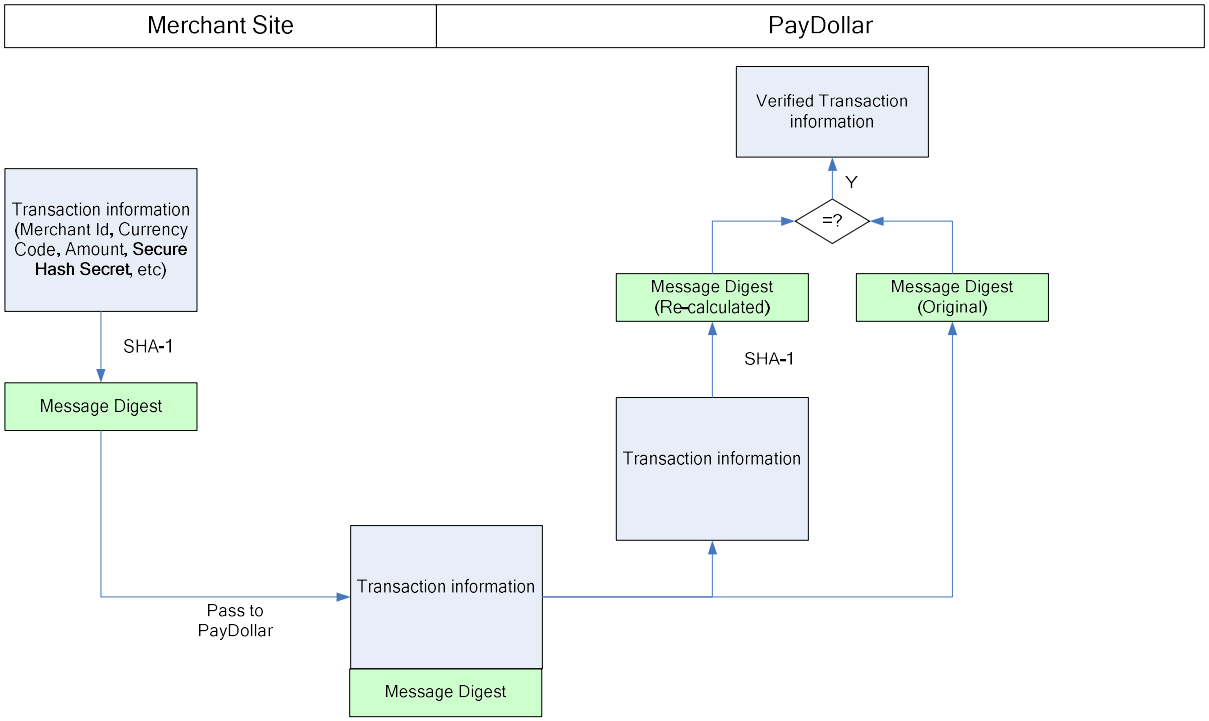
```
<%@ Language = "VBScript" %>
<%
     Dim successCode
     Dim payRef
     Dim Ref

     successCode = Request.Form("successcode")
     payRef = Request.Form("PayRef")
     Ref = Request.Form("Ref")

     ' Print out 'OK' to notify us you have received the payment result
     Response.write("OK")


     If successCode = "0" Then
          ' Transaction Accepted
          ' *** Add the Security Control here, to check the currency, amount with the
          ' *** merchant's order reference from your database, if the order exist then
          ' *** accepted otherwise rejected the transaction.

          ' Update your database for Transaction Accepted and send email or notify your
          ' customer.
          .....

          ' In case if your database or your system got problem, you can send a void
          ' transaction request. See API guide for more details
     Else
          ' Transaction Rejected
          ' Update your database for Transaction Rejected
          .....
     End If
%>
```

4 Transaction security by Secure Hash

Introduction

The purpose of Secure Hash is to enhance the transaction message communication security between merchant site and PayDollar. By employing this technology, the integrity of the information and the identity of the signatory can be authenticated with industry standard.

Basic flow of Secure Hash



1. For all transaction request send from merchant site to PayDollar, secure hash should be calculated and added to the transaction request to authenticate the integrity of the transaction information and the identity of the merchant.
2. The secure hash is calculated by hashing the following parameters using SHA-1 or SHA-256, a cryptographic hash function of industry standard.
   - Merchant ID
   - Merchant Reference Number
   - Currency Code
   - Amount
   - Payment Type
   - Secure Hash Secret – Assigned by PayDollar to merchant.
3. When the transaction request is received, PayDollar verifies the transaction by comparing the secure hash submitted by merchant and the secure hash re-calculated by other input parameters. If both values are the same, existing payment flow will follows. Or else, the payment request will be

dropped. PayDollar will send out an email to notify the merchant through the operation contact.

4. After the transaction is completed, PayDollar will send out datafeed to merchant site. A secure hash will also be calculated by hashing the following parameters using SHA-1 or SHA-256,
   - Src
   - Prc
   - Success Code
   - Merchant Reference Number
   - PayDollar Reference Number
   - Currency Code
   - Amount
   - Payer Authentication Status
   - Secure Hash Secret
5. After receiving the datafeed, merchant is suggested to verify the information by comparing the secure hash posted by PayDollar and the secure hash re-calculated by other response parameters. If the values doesn't matched, the datafeed may have been tampered within the redirection process and you are suggested to do further investigation before confirmation the order.

Client library provided by PayDollar

Client library is provided by PayDollar to facilitate the secure hash generation and verification process. It supports common programming language including Java, PHP, ASP, ASP.NET. The following function calls are supported.

Functions	Parameters	Descriptions
generateSecureHash	Input - Merchant ID - Merchant Reference Number - Currency Code - Amount - Payment Type - Secure Hash Secret Output - Secure Hash String	Create a secure hash using the input parameters and Secure Hash Secret. (The result secure hash should be included in the payment parameter send to PayDollar.)
verifyDatafeed	Input - Src - Prc - Success Code	Verify the parameters passed from PayDollar using input parameters and Secure Hash Secret.

	- Merchant Reference Number - PayDollar Reference Number - Currency Code - Amount - Payer Authentication Status - Secure Hash Secret - Secure Hash from PayDollar Output - True/False	(If the result of the output is true, it is verified that the result is sent from PayDollar and it is safe to trust the result.)

*Please login to PayDollar Merchant Administration Tools and download client library with sample code under Support → Developer Corner.

Generating and verifying Secure Hash manually

Merchant may also generate and verify secure hash manually, without using the client library provided by PayDollar. The following diagrams list out the exact algorithm,

Generate Secure Hash

1. Create the signing data string.
   Signing data string = Merchant ID + "|" + Merchant Reference + "|" + Currency Code + "|" + Amount + "|" + Payment Type + "|" + Secure Hash Secret
2. Secure Hash = SHA-1 or SHA-256 (Signing data string)

*SHA-1 is the original 160-bit hash function (default).

Example of Secure Hash Secret
gMAVIEGVpqHvxoNEqbrZRuBDFT1B0icW

Example of Signing data string
56100908|1280204670187|344|10|N|gMAVIEGVpqHvxoNEqbrZRuBDFT1B0icW

Example of Secure Hash
13068c0ef09139ea711d36bde16785a2d30b9a30

Verifying Secure Hash from PayDollar datafeed

1. Create the verify data string.

   Verify data string = Src + "|" + Prc + "|" + Success Code + "|" + Merchant Reference Number + "|" + PayDollar Reference Number + "|" + Currency Code + "|" + Amount + "|" + Payer Authentication Status + "|" + Secure Hash Secret
2. Verify Secure Hash = SHA-1 / SHA-256 (Verify data string)
3. Extract the secure hash from PayDollar datafeed.
4. Compare the output from step 2 and step 3. If they are equals, return True, else return False.

*SHA-1 is the original 160-bit hash function. (default)

Enable Secure Hash function of your merchant account

a) Please contact PayDollar Service Department (service@paydollar.com) to enable the Secure Hash function of your merchant account.

b) You may retrieve the Secure Hash Secret of the merchant account by accessing to the Merchant Administration Interface, "Profile" → "Payment Information". The Secure Hash Secret must be kept safely for the function to be effective.

c) The Secure Hash Secret will be changed every 2 years to enhance the level of security.

d) Once this function is enabled, a valid Secure Hash should be included in all transaction requests. All transaction without valid Secure Hash will be dropped by PayDollar.

e) You may download client library with sample code under,
   PayDollar Merchant Administration Tools → Support → Developer Corner.

5 Multi-Currency Processing Service

Introduction

PayDollar by Multi-Currency Processing Service (MPS) is an integrated e-payment transaction processing service that allows your online business of any size to securely accept real-time credit card payments from overseas cardholders and offer them the choice to pay for their goods and services in their billing currency, whilst merchants continue to be settled for transactions in their base currency.

Multi-Currency Processing Service (MPS) provides three different modes as below:

- Simple Currency Conversion (SCP)
- Multi-Currency Pricing (MCP)
- Dynamic Currency Conversion (DCC)

Multi-Currency Processing Service (MPS) facilitates merchant to connect to our network with great flexibility. Merchant can choose the following integration method.

- Client Post through Browser (e.g. Shopping Cart)

Definition:

"Foreign Currency" means those non-based currency for which the Program is available to merchant from time to time supported and advised by AsiaPay.

"Base Currency" means the currency in which the merchant is settled for payment transactions by its acquirer.

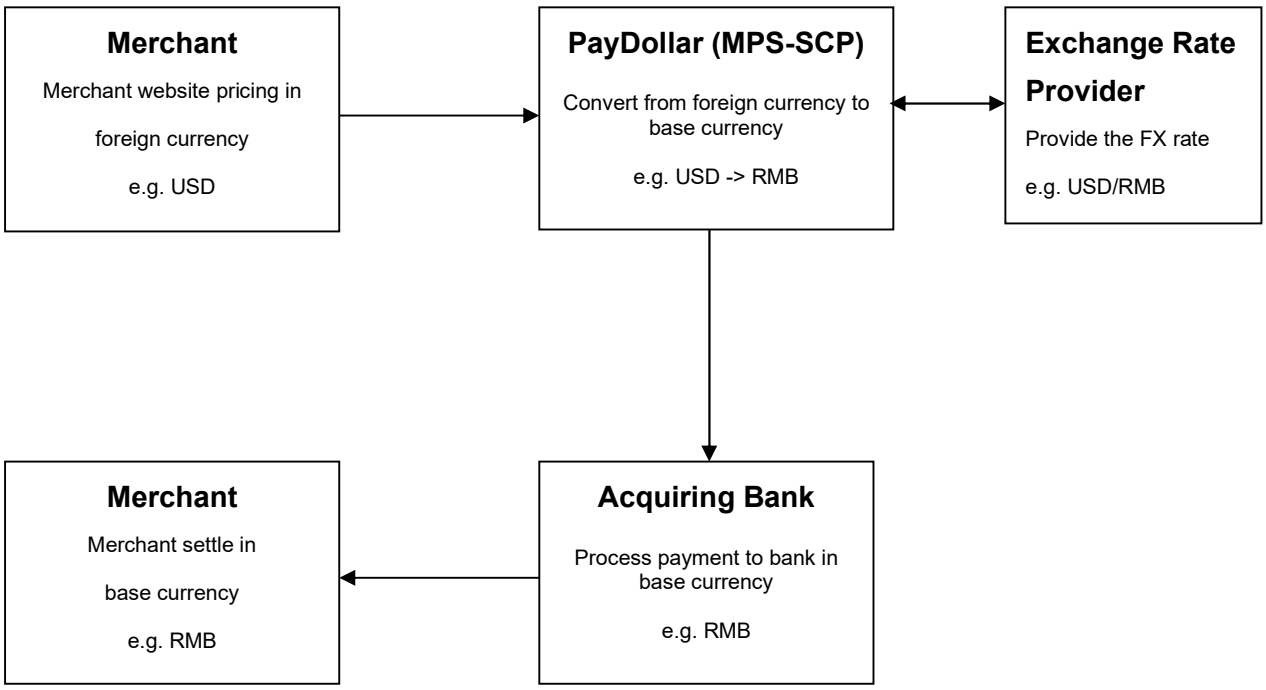
"Conversion Rate" means the foreign currency exchange rate derived by a recognized supplier.

Simple Currency Conversion (SCP)

Simple Currency Conversion (SCP) is a value added e-payment processing service that allows your online business to securely accept real-time credit card payments from overseas customers with **foreign currencies in pricing** while offering them to pay for goods and services in **your preferred currency.** And, your business can continue to collect settlement in base currency as usual.

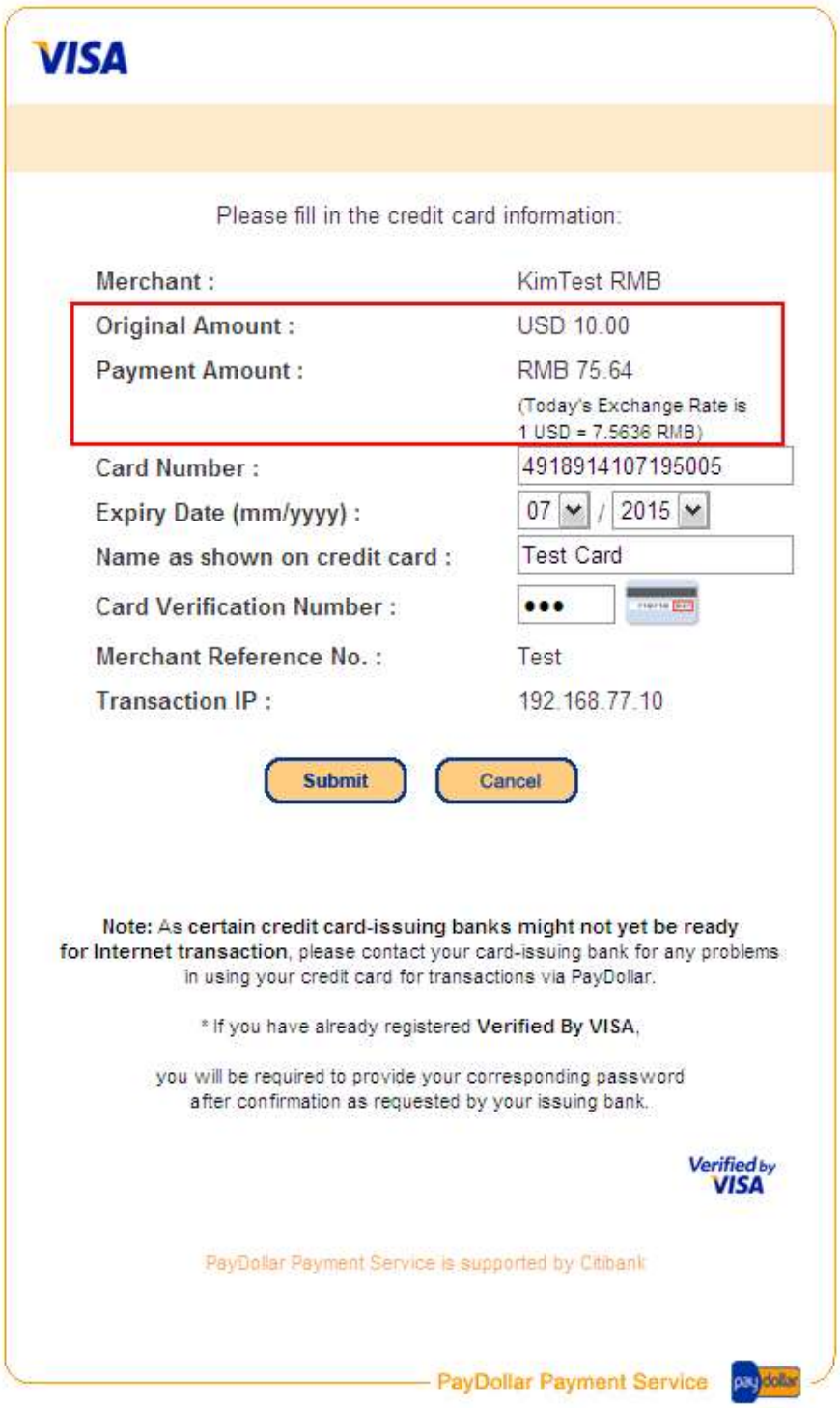
Simple Currency Conversion (SCP) will convert the foreign currency that posted by merchant to the base currency of merchant according to the conversion rate obtained from our exchange rate provider. After that, the foreign amount, base amount and the exchange rate will be shown on the Paydollar's payment input page.

Transaction Flow

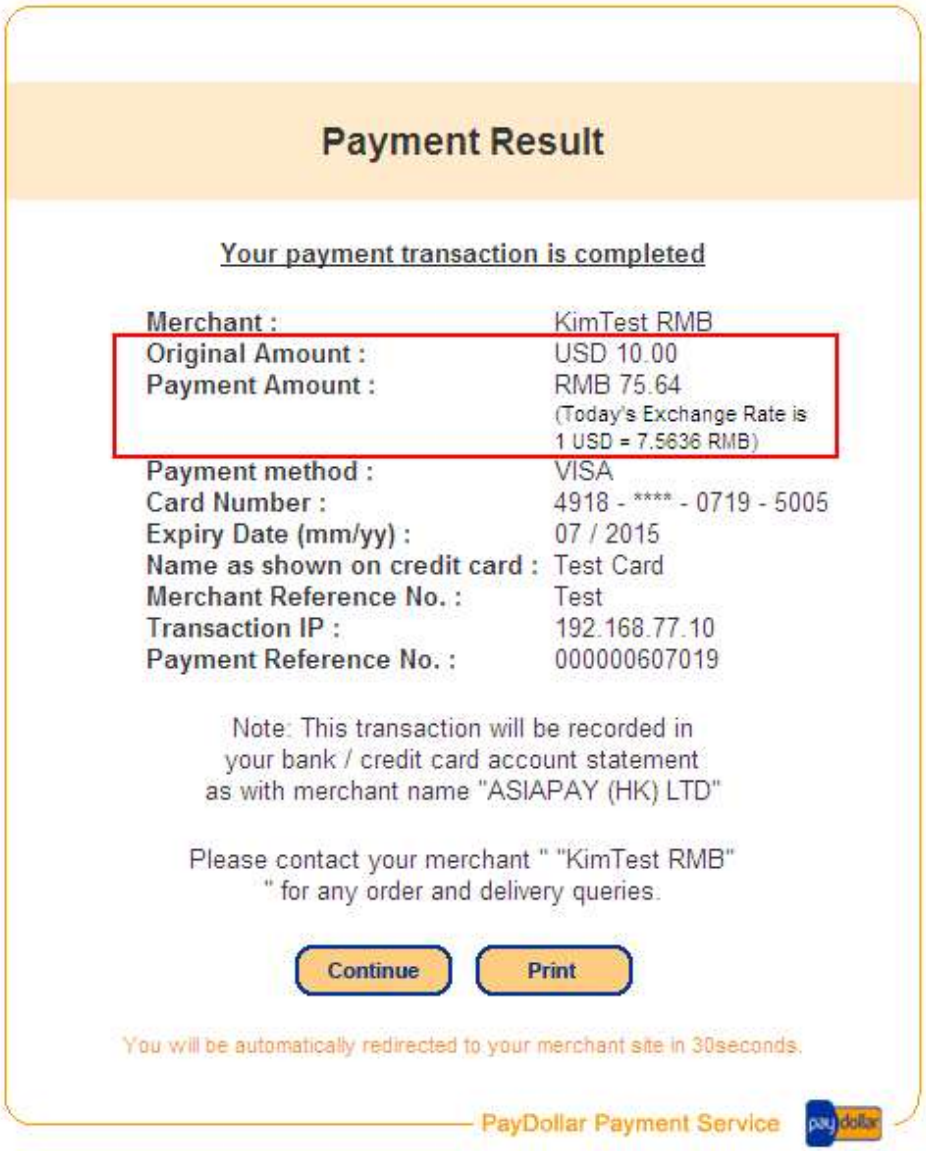


Simple Currency Conversion (SCP) Sample Transaction Screen

Payment Amount Conversion and Account Input Page:



Payment Result Page:

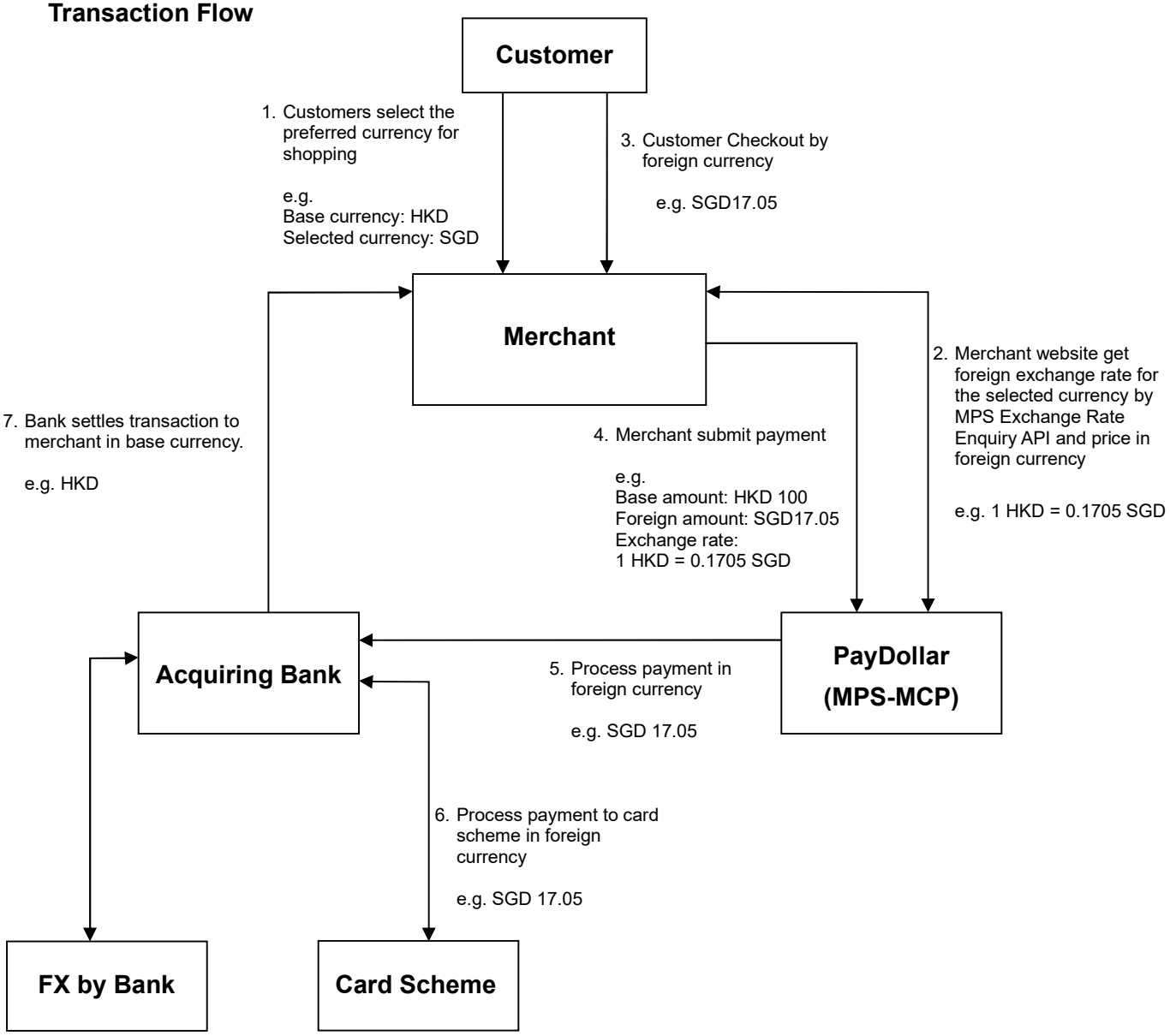


Multi Currency Pricing (MCP)

Multi-Currency Pricing (MCP) is a value added e-payment processing service that allows your online business to securely accept real-time credit card payments from overseas customers while offering them the choice to pay for goods and services in **merchant base currency** or **other worldwide currencies (e.g. USD)**. And, your business can continue to collect settlement in local currency as usual.

Multi-Currency Pricing (MCP) will translate the base currency that posted by merchant to the worldwide currency according to the conversion rate. After that, customer can select one of the currencies for payment.

Transaction Flow



MCP Calculation

PayDollar will provide an API to merchant to integrate with their web server. This MPS Exchange Rate Enquiry API is used for enquiry exchange rate and calculation of MCP foreign amount. The calculation of the MCP amounts will be based on the stored rates at PayDollar; the correct minor unit for that currency will provided.

Rounding

Rate response messages contain a margined foreign exchange rate with four decimal places. Merchant will use this rate to calculate the MCP amounts. Merchant will take account of the correct minor unit per currency when calculating the MCP amount. Round half up will be used for rounding.

.

- **Example: Two minor unit currencies**
  For exchange rate 1HKD=0.0960EUR
  HKD 376 = EUR 36.10 (Before Rounding, EUR 36.096)
  For exchange rate 1HKD=0.1394CAD
  HKD 345 = CAD 48.09 (Before Rounding, CAD 48.093)

- **Example: Zero Minor unit currency**
  For exchange rate 1HKD=132.3400KRW
  HKD 100 = KRW 13234

Rates Expiry

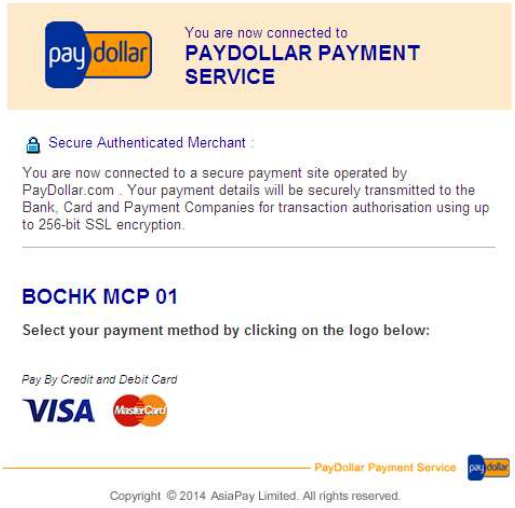
Rate response messages contain an expiry time. Rates are only valid prior to this expiry time. The transaction will be rejected by PayDollar if the MCP rates have expired. Therefore, merchant should make the rate enquiry every time in each new customer browsing session instead of storing the rate locally.

Tolerance Check

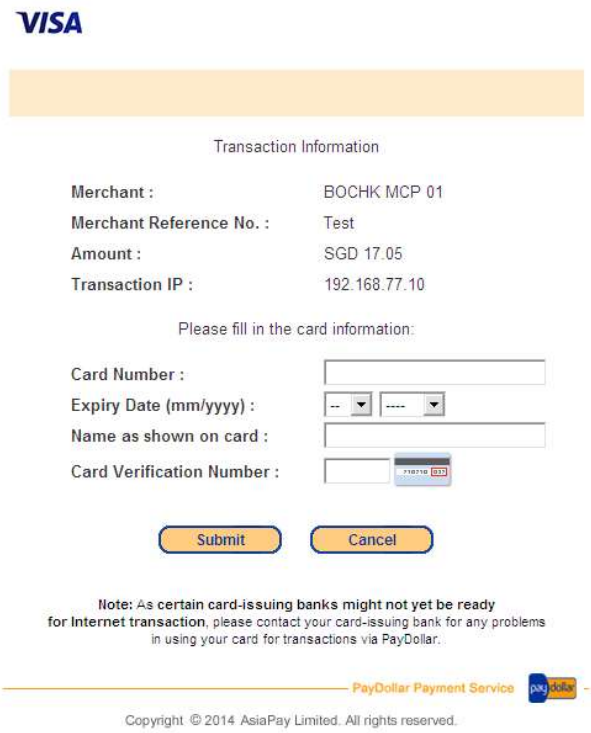
PayDollar has a rate tolerance check on every MCP transaction to make sure that all the MCP transaction only will be processed within an agreed tolerance level with bank of the latest margined exchange rates.The transaction will be rejected by PayDollar if the tolerance check fails.

Multi-Currency Pricing (MCP) Sample Transaction Screen

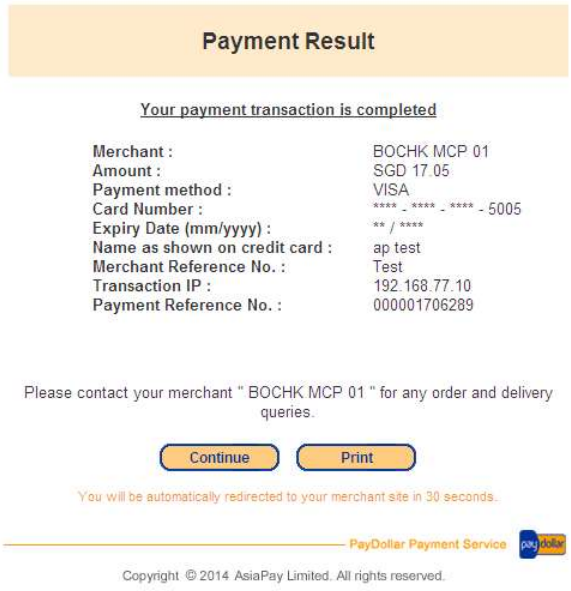
Payment Method Selection Page:



Payment Account Input Page:



Payment Result Page:

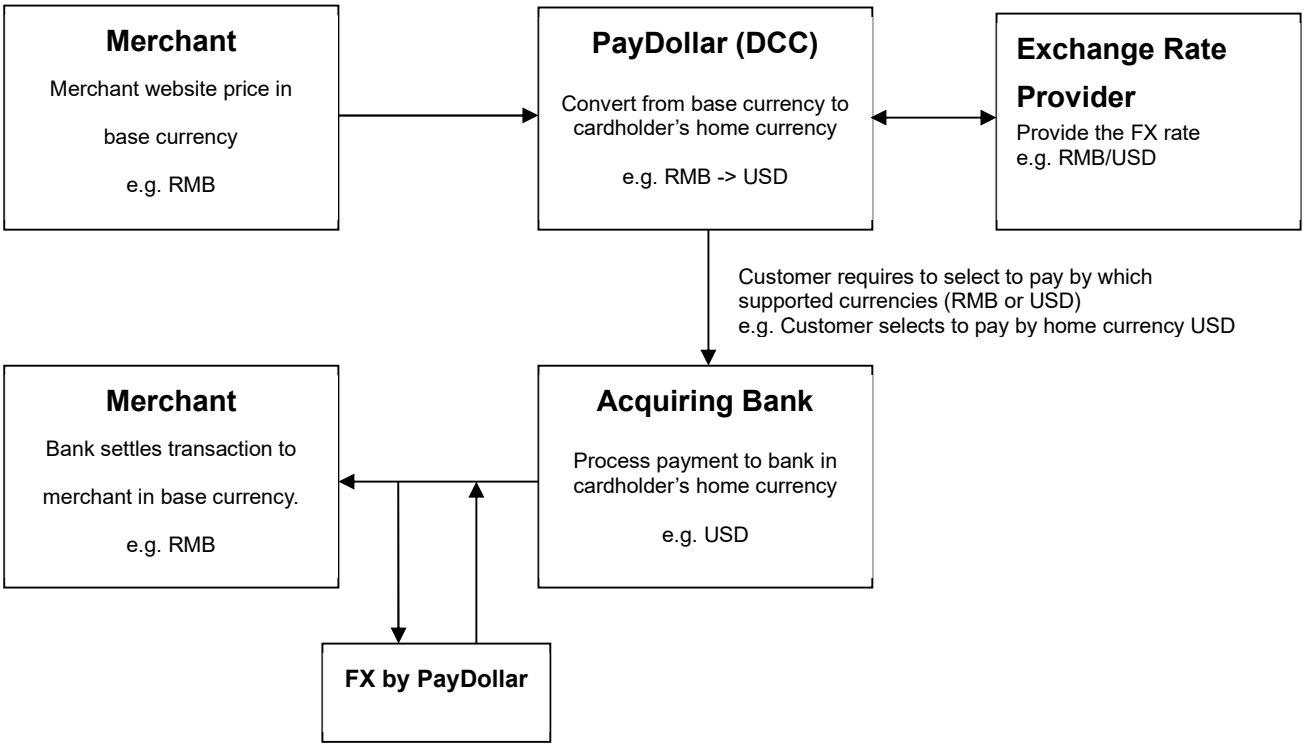


Dynamic Currency Conversion (DCC)

Dynamic Currency Conversion (DCC) is a value added e-payment processing service that allows your online business to securely accept real-time credit card payments from overseas customers while offering them the choice to pay for goods and services in the **merchant base currency** or **cardholder's home currency**. This payment process allows the merchant to show the value of the transaction in the cardholder's home currency.

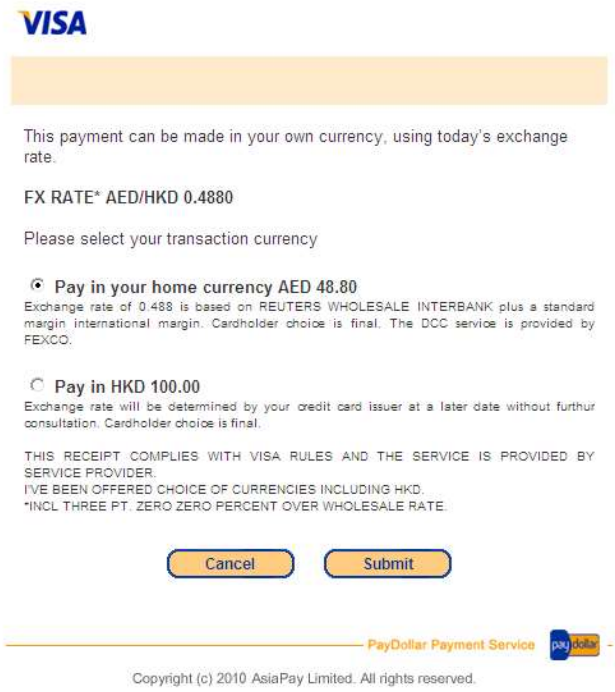
Dynamic Currency Conversion (DCC) will translate the base currency that posted by merchant to the cardholder's home currency according to the conversion rate. After that, customer can select one of the currencies for payment.

Transaction Flow

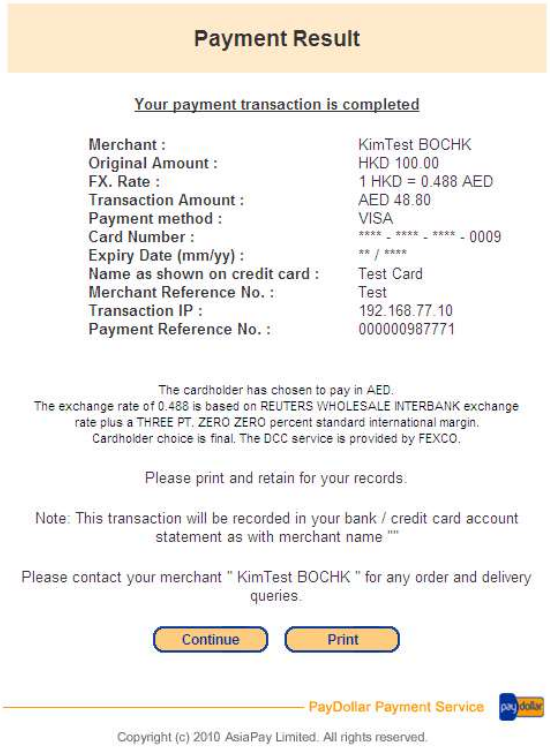


Dynamic Currency Conversion (DCC) Sample Transaction Screen

DCC Selection Page:



Payment Result Page:



Payment Result

Your payment transaction is completed

Merchant :	KimTest BOCHK
Original Amount :	HKD 100.00
FX. Rate :	1 HKD = 0.488 AED
Transaction Amount :	AED 48.80
Payment method :	VISA
Card Number :	**** - **** - **** - 0009
Expiry Date (mm/yy) :	** / ****
Name as shown on credit card :	Test Card
Merchant Reference No. :	Test
Transaction IP :	192.168.77.10
Payment Reference No. :	000000987771

The cardholder has chosen to pay in AED.
The exchange rate of 0.488 is based on REUTERS WHOLESALE INTERBANK exchange rate plus a THREE PT. ZERO ZERO percent standard international margin.
Cardholder choice is final. The DCC service is provided by FEXCO.

Please print and retain for your records.

Note: This transaction will be recorded in your bank / credit card account statement as with merchant name ""

Please contact your merchant " KimTest BOCHK " for any order and delivery queries.

Continue Print

PayDollar Payment Service

6 Functions of Merchant API

Introduction of API functions

There are totally four functions provided:

- Capture Authorized Payment
- Void Accepted Payment
- Request Refund Accepted Payment
- Query Payment Status

To connect to our system, you need to post the required parameters by HTML form posting to our merchant API web page and then get back the processing result from that page. You can implement it by server-side html post.

- URL of Testing Platform:
  https://test.paydollar.com/b2cDemo/eng/merchant/api/orderApi.jsp
- URL of Production Platform:
  https://www.paydollar.com/b2c2/eng/merchant/api/orderApi.jsp

Beside, a set of API login ID and password will be assigned to your merchant account for accessing this API function. And it can be obtained from us by sending a request email or directly contact us.

Capture Authorized Payment

The aim of this function is to capture the authorized payment.

Definition of Parameters in the Integration Page

Input /Return	Parameters (Required Fields are in Bold typeface) *Case Sensitive	Data Type	Expected Value	Descriptions
Input	**merchantId**	Number		The merchant ID we provide
	loginId	Text (30)		The loginId of merchant API
	password	Text (15)		The password of merchant API
	actionType		"Capture"	The action type
	payRef	Text (35)		Payment Reference Number
	amount	Number (12,2)		The amount you want to capture (must be less than or equal to the original amount)
Return	resultCode	Number	{"0","-1"}	0 - Request Successfully -1 – Request Failed
	orderStatus	Text(20)		The new order status after successfully request
	ref	Text		Merchant's Order Reference Number
	payRef	Number		PayDollar transaction reference
	amt	Number (12,2)		Transaction Amt
	cur	Number (3)		Transaction Currency "344" – HKD "840" – USD "156" – CNY (RMB) "392" – JPY "036" – AUD "978" – EUR "124" – CAD "446" – MOP "764" – THB "458" – MYR "410" – KRW "682" – SAR "784" – AED "096" – BND "356" – INR "702" – SGD "826" – GBP "901" – TWD "608" – PHP "360" – IDR "554" – NZD "704" – VND
	errMsg	Text		Error Message

Additional output parameters for transaction processed with rewards program (e.g. Hang Seng Cash Dollar)				
	netAmtAfterRewards	Number(12, 2)		Total Net Amount after Redeem Rewards Program
	rewardsRedeemTotalAmt	Number(12, 2)		Total Rewards Redeem Amount
	rewardsRedeemType	Text		Rewards Redeem Type "HASE" – Hang Seng Cash Dollar
	rewardsRedeemProgram	Number		Number of redeem programs Return value is 1,2, or 3
	rewardsRedeemCode1	Text		First Rewards Program Redeem Program Code * Return only when rewardsRedeemProgram return value is 1,2 or 3
	rewardsRedeemLabel1	Text		First Rewards Program Redeem Program Label * Return only when rewardsRedeemProgram return value is 1,2 or 3
	rewardsRedeemAmt1	Number(12, 2)		First Rewards Program Redeem Amount * Return only when rewardsRedeemProgram return value is 1,2 or 3
	rewardsBalance1	Text		First Rewards Program Balance Format Expired Rewards Balance Expired Date – Expired Balance Amount | Rewards Balance Expired Date- Rewards Balance Amount Example: 0115-20.00|0120-30.00 * Return only when rewardsRedeemProgram return value is 1,2 or 3
	rewardsRedeemCode2	Text		Second Rewards Program Redeem Program Code * Return only when rewardsRedeemProgram return value is 2

				or 3
	rewardsRedeemLabel2	Text		Second Rewards Program Redeem Program Label * Return only when rewardsRedeemProgram return value is 2 or 3
	rewardsRedeemAmt2	Number (12,2)		Second Rewards Program Redeem Amount * Return only when rewardsRedeemProgram return value is 2 or 3
	rewardsBalance2	Text		Second Rewards Program Balance Format Expired Rewards Balance Expired Date – Expired Balance Amount | Rewards Balance Expired Date- Rewards Balance Amount Example: 0115-20.00|0120-30.00 * Return only when rewardsRedeemProgram return value is 2 or 3
	rewardsRedeemCode3	Text		Third Rewards Program Redeem Program Code * Return only when rewardsRedeemProgram return value is 3
	rewardsRedeemLabel3	Text		Third Rewards Program Redeem Program Label * Return only when rewardsRedeemProgram return value is 3
	rewardsRedeemAmt3	Number(12, 2)		Third Rewards Program Redeem Amount * Return only when rewardsRedeemProgram return value is 3
	rewardsBalance3	Text		Third Rewards Program Balance Format Expired Rewards Balance Expired Date – Expired Balance Amount | Rewards Balance Expired Date- Rewards Balance Amount

				Example: 0115-20.00|0120-30.00 * Return only when rewardsRedeemProgram return value is 3

All the return parameters will be concatenated as in html request format by separate with **&**

Sample return string:

`resultCode=0&orderStatus=Accepted&ref=Test&payRef=4780&amt=1.0&cur=344&errMsg=Capture Successfully.`

Void Accepted Payment

The aim of this function is to void the accepted payment before settlement. It can be done only before our settlement time.

Definition of Parameters in the Integration Page

Input /Return	Parameters (Required Fields are in Bold typeface) *Case Sensitive	Data Type	Expected Value	Descriptions
Input	**merchantId**	Number		The merchant ID we provide to you
	loginId	Text (30)		The loginId of using merchant API
	password	Text (15)		The password of using merchant API
	actionType		"Void"	The action type
	payRef	Text (35)		Payment Reference Number
Return	resultCode	Number	{"0","-1"}	0 - Request Successfully -1 – Request Failed
	orderStatus	Text(20)		The new order status after successfully request
	ref	Text		Merchant's Order Reference Number
	payRef	Number		PayDollar transaction reference
	amt	Number (12,2)		Transaction Amt
	cur	Number (3)		Transaction Currency "344" – HKD "840" – USD "156" – CNY (RMB) "392" – JPY "036" – AUD "978" – EUR "124" – CAD "446" – MOP "764" – THB "458" – MYR "410" – KRW "682" – SAR "784" – AED "096" – BND "356" – INR "702" – SGD

				"826" – GBP	"901" – TWD
				"608" – PHP	"360" – IDR
				"554" – NZD	"704" – VND
	errMsg	Text		Error Message	

All the return parameters will be concatenated as in html request format by separate with **&**

Sample return string:

`resultCode=0&orderStatus=Voided&ref=Test&payRef=4780&amt=1.0&cur=344&errMsg=Void Successfully.`

Request Refund for Accepted Payment

The aim of this function is to request refund for accepted payment before 14 days.

Definition of Parameters in the Integration Page

Input /Return	Parameters (Required Fields are in Bold typeface) *Case Sensitive	Data Type	Expected Value	Descriptions
Input	**merchantId**	Number		The merchant ID we provide to you
	loginId	Text (30)		The loginId of using merchant API
	password	Text (15)		The password of using merchant API
	actionType		"RequestRefund"	The action type
	payRef	Text (35)		Payment Reference Number
(Optional Input)	**amount**	Number (12,2)		The amount you want to refund (must be less than or equal to the original amount)
Return	resultCode	Number	{"0","-1"}	0 - Request Successfully -1 – Request Failed
	orderStatus	Text(20)		The new order status after successfully request
	ref	Text		Merchant's Order Reference Number
	payRef	Number		PayDollar transaction reference
	amt	Number (12,2)		Transaction Amt
	cur	Number (3)		Transaction Currency "344" – HKD "840" – USD "156" – CNY (RMB) "392" – JPY "036" – AUD "978" – EUR

				"124" – CAD	"446" – MOP
				"764" – THB	"458" – MYR
				"410" – KRW	"682" – SAR
				"784" – AED	"096" – BND
				"356" – INR	"702" – SGD
				"826" – GBP	"901" – TWD
				"608" – PHP	"360" – IDR
				"554" – NZD	"704" – VND
	errMsg	Text		Error Message	

All the return parameters will be concatenated as in html request format by separate with **&**

Sample return string:

```
resultCode=0&orderStatus=RequestRefund&ref=Test&payRef=4780&amt=1.0&cur=344
&errMsg=Request successfully and we will process it later.
```

Query payment status

The aim of this function is to query the payment status on an order by either Merchant Reference Number or Payment Reference Number with XML

Definition of Parameters in the Integration Page

Input /Return	Parameters (Required Fields are in Bold typeface) *Case Sensitive	Data Type	Expected Value	Descriptions
Input	**merchantId**	Number		The merchant ID we provide to you
	loginId	Text (30)		The loginId of using merchant API
	password	Text (15)		The password of using merchant API
	actionType		"Query"	The action type
(Coditional)	**orderRef**	Text (35)		Merchant Reference Number
(Coditional)	**payRef**	Text (35)		Payment Reference Number
Return	orderStatus	Text(20)		The new order status after successfully request
	ref	Text		Merchant's Order Reference Number
	payRef	Number		PayDollar transaction reference
	mpsMode	Text(3)		The Multi – Currency Processing Service (MPS) Mode: "NIL" or not provide – Disable MPS (No currency

				conversion) “SCP” – Enable MPS with ‘Simple Currency Conversion’ “DCC” – Enable MPS with ‘Dynamic Currency Conversion’ “MCP” – Enable MPS with ‘Multi Currency Pricing’
	amt	Number (12,2)		Transaction Amt
	cur	Number (3)		Transaction Currency i.e. “344” – HKD “840” – USD “156” – CNY (RMB) “392” – JPY “036” – AUD “978” – EUR “124” – CAD “446” – MOP “764” – THB “458” – MYR “410” – KRW “682” – SAR “784” – AED “096” – BND “356” – INR “702” – SGD “826” – GBP “901” – TWD “608” – PHP “360” – IDR “554” – NZD “704” – VND
	prc	Text		Primary response code
	src	Text		Secondary response code
	ord	Text		Bank Reference Number
	holder	Text		The Holder Name of the Payment Account
	sourceIp	Text (15)		IP address of payer
	ipCountry	Text (3)		Country of payer (e.g. HK) - if country is on high risk country list, an asterisk will be shown (e.g. MY*)
	payMethod	Text (10)		Payment method (e.g. VISA, Master, JCB, AMEX)
	cardIssuingCountry	Text (3)		Card Issuing Country Code (e.g. HK) - if country is on high risk country list, an asterisk will be shown (e.g. MY*) - if the card issuing country of credit card is undefined, “- -“ will be shown. Please refer to Appendix A “List of Country Code” for detail
	airline_ticketNumber	Text		Air Ticket Number
	errMsg	Text		Error Message
	txTime	Text (YYYY-		transaction time

<table>
<tr><td></td><td></td><td>MM-DD HH:MI:SS.0)</td><td></td><td></td></tr>
<tr><td></td><td>successcode</td><td>Number</td><td></td><td>0- succeeded, 1- failure, Others – error
*Only return for order status “Accepted”, “Authorized”, “Rejected”</td></tr>
<tr><td></td><td>remark</td><td>Text (200)</td><td></td><td>A remark field for you to store additional data that will not show on the transaction web page</td></tr>
<tr><td></td><td>authId</td><td>Text</td><td></td><td>Approval Code</td></tr>
<tr><td></td><td>eci</td><td>Text (2)</td><td></td><td>ECI value (for 3D enabled Merchants)
<table>
<tr><td colspan="2">VISA / JCB / AMEX</td></tr>
<tr><td>ECI Value</td><td>Definition</td></tr>
<tr><td>05</td><td>Both cardholder and card issuing bank are 3D enabled. 3D card authentication is successful</td></tr>
<tr><td>06</td><td>Either cardholder or card issuing bank is not 3D enrolled. 3D card authentication is unsuccessful, in sample situations as:
1. 3D cardholder not enrolled
2. Card issuing bank is not 3-D Secure ready</td></tr>
<tr><td>07</td><td>Authentication is unsuccessful or not attempted. The credit card is either a non-3D card or card issuing bank does not handle it as a 3D transaction</td></tr>
<tr><td colspan="2">MasterCard</td></tr>
<tr><td>ECI Value</td><td>Definition</td></tr>
<tr><td>00</td><td>Authentication is unsuccessful or not attempted. The credit card is either a non-3D card or card issuing bank does not handle it as a 3D transaction</td></tr>
<tr><td>01</td><td>Either cardholder or card issuing bank is not 3D enrolled. 3D card authentication is unsuccessful, in sample situations as:
1. 3D Cardholder not enrolled
2. Card issuing bank is not 3-D Secure ready</td></tr>
<tr><td>02</td><td>Both cardholder and card issuing bank are 3D enabled. 3D card authentication is successful</td></tr>
</table>
Remark: String value “null” will be sent when the transaction is rejected by PayDollar PayAlert or the transaction doesn’t has ECI value.</td></tr>
<tr><td></td><td>payerAuth</td><td>Text (1)</td><td></td><td>Payer Authentication Status
Y - Card is 3-D secure enrolled and authentication succeeds.
N - Card is 3-D secure enrolled but authentication fails.
P - 3-D Secure check is pending
A - Card is not 3-D secure enrolled yet
U - 3D-secure check is not processed.</td></tr>
</table>

	channelType	Text (3)		Channel Type: SPC – Client Post Through Browser DPC – Direct Client Side Connection DPS – Server Side Direct Connection SCH – Schedule Payment DPL – Direct Payment Link Connection MOT – Moto Connection RTL – RetailPay Connection BPP – Batch Payment Process MOB – Mobile Payment Connection
	MerchantId	Number		The merchant Id of transaction
	alertCode	Text (50)		The Alert Code e.g. R14 –IP Country not match with Issuing Country R 9 - In high risk country list
Additional output parameters for card information				
	expMonth	Text (MM)		Expiry month of the card *Return to approved merchant only
	expYear	Text (YYYY)		Expiry year of the card *Return to approved merchant only
	panFirst6	Text (6)		First 6 digit of card *Return to approved merchant only
	panFull	Text		Encrypted PAN with AES256 encryption *Return to approved merchant only
	panMasked	Text		Full Masked PAN *Return to approved merchant only
	panFirst6	Text (6)		First 6 digit of card *Return to approved merchant only
	panFirst4	Text (4)		First 4 digit of card *Return to approved merchant only
	panLast4	Text (4)		Last 4 digit of card *Return to approved merchant only
	accountHash	Text		Hash value of card *Return to approved merchant only
	accountHashAlgo	Text		Hash function of card *Return to approved merchant only
Additional output parameters for transaction processed with MPS VAS				
	mpsAmt	Number		MPS Transaction Amount

		(12,2)		**Remark: For MPS Enable only.**
	mpsCur	Text (3)		MPS Transaction Currency **Remark: For MPS Enable only.**
	mpsForeignAmt	Number (12,2)		MPS Transaction Foreign Amount **Remark: For MPS Enable only.**
	mpsForeignCur	Text (3)		MPS Transaction Foreign Currency **Remark: For MPS Enable only.**
	mpsRate	Number (12,4)		MPS Exchange Rate: (Foreign / Base) e.g. USD / HKD = 7.77 **Remark: For MPS Enable only.**
Additional output parameters for transaction processed with SchedulePay VAS				
	mSchPayId	Number		The Master Schedule Payment Id *For Schedule payment transaction only
	dSchPayId	Number		The Detail Schedule Payment Id *For Schedule payment transaction only
Additional output parameters for transaction processed with MemberPay VAS				
	mpMemberId	Text		Member Id if merchant using memberPay
Additional output parameters for transaction processed with Third Party eWallet				
	isEwallet	Text(1)		"T" –This is a transaction with Ewallet function
	eWalletBrand	Text		ECO - AMEX Express checkout MP – MasterPass VCO – VISA Checkout
Additional output parameters for transaction processed with Promopay VAS				
	promotionCode	Text(8);		The promotion campaign code of transaction *For promotion transaction only
	promotionRuleCode	Text(8);		The promotion rule code of transaction *For promotion transaction only
	promotionOriginalAmt	Number (12,2)		The original amount of promotion *For promotion transaction only
Additional output parameters for transaction processed with Installment				
	installment_period	Number		The installment period (In number of months)
	installment_provider	Text		The issuing bank or provider of the installment "HASE" – Hang Seng Bank (Hong Kong) "BOCIHK" – Bank of China (Hong Kong) "BCA" - Bank Central Asia (Indonesia) "WELEND" – WeLend (Hong Kong)

	installment_firstPayAmt	Number		The transaction amount for first installment period * Return only if the installment bank providing this information
	installment_eachPayAmt	Number		The transaction amount for each installment period * Return only if the installment bank providing this information
	installment_lastPayAmt	Number		The transaction amount for last installment period * Return only if the installment bank providing this information
Additional output parameters for transaction processed with rewards program (e.g. Hang Seng Cash Dollar)				
	netAmtAfterRewards	Number(12, 2)		Total Net Amount after Redeem Rewards Program
	rewardsRedeemTotalAmt	Number(12, 2)		Total Rewards Redeem Amount
	rewardsRedeemType	Text		Rewards Redeem Type "HASE" – Hang Seng Cash Dollar
	rewardsRedeemProgram	Number		Number of redeem programs Return value is 1,2, or 3
	rewardsRedeemCode1	Text		First Rewards Program Redeem Program Code * Return only when rewardsRedeemProgram return value is 1,2 or 3
	rewardsRedeemLabel1	Text		First Rewards Program Redeem Program Label * Return only when rewardsRedeemProgram return value is 1,2 or 3
	rewardsRedeemAmt1	Number(12, 2)		First Rewards Program Redeem Amount * Return only when rewardsRedeemProgram return value is 1,2 or 3
	rewardsBalance1	Text		First Rewards Program Balance Format Expired Rewards Balance Expired Date – Expired Balance Amount | Rewards Balance Expired Date- Rewards Balance Amount Example: 0115-20.00|0120-30.00 * Return only when rewardsRedeemProgram return value is 1,2 or 3
	rewardsRedeemCode2	Text		Second Rewards Program Redeem Program Code * Return only when rewardsRedeemProgram return value is 2 or 3

	rewardsRedeemLabel2	Text		Second Rewards Program Redeem Program Label * Return only when rewardsRedeemProgram return value is 2 or 3
	rewardsRedeemAmt2	Number (12,2)		Second Rewards Program Redeem Amount * Return only when rewardsRedeemProgram return value is 2 or 3
	rewardsBalance2	Text		Second Rewards Program Balance Format Expired Rewards Balance Expired Date – Expired Balance Amount | Rewards Balance Expired Date- Rewards Balance Amount Example: 0115-20.00|0120-30.00 * Return only when rewardsRedeemProgram return value is 2 or 3
	rewardsRedeemCode3	Text		Third Rewards Program Redeem Program Code * Return only when rewardsRedeemProgram return value is 3
	rewardsRedeemLabel3	Text		Third Rewards Program Redeem Program Label * Return only when rewardsRedeemProgram return value is 3
	rewardsRedeemAmt3	Number(12, 2)		Third Rewards Program Redeem Amount * Return only when rewardsRedeemProgram return value is 3
	rewardsBalance3	Text		Third Rewards Program Balance Format Expired Rewards Balance Expired Date – Expired Balance Amount | Rewards Balance Expired Date- Rewards Balance Amount Example: 0115-20.00|0120-30.00 * Return only when rewardsRedeemProgram return value is 3

All the return parameters will be in XML format

Sample return XML file:

```
<records>
    <record>
        <orderStatus>Accepted</orderStatus>
        <ref>Test</ref>
```

```
<payRef>1390545</payRef>
<mpsMode>NIL</mpsMode>
<amt>1</amt>
<cur>344</cur>
<prc>0</prc>
<src>0</src>
<ord>00004295104</ord>
<holder>Holder Name</holder>
<authId>123456</authId>
<alertCode>R14</alertCode>
<remark>Test</remark>
<eci>05</eci>
<payerAuth>Y</payerAuth>
<sourceIp>202.65.133.23</sourceIp>
<ipCountry>HK</ipCountry>
<payMethod>VISA</payMethod>
<panFull>4918914107195005</panFull> or
<panMasked>491891******5005</panMasked> or
<panFirst4>4918</panFirst4>
<panLast4>5005</panLast4>
<expMonth>07</expMonth>
<expYear>2015</expYear>
<cardIssuingCountry>HK</cardIssuingCountry>
<channelType>SPN</channelType>
<txTime>2014-07-01 00:00:00.0</txTime>
<accountHash></accountHash>
<accountHashAlgo></accountHashAlgo>
<mpsAmt></mpsAmt>
<mpsCur></mpsCur>
<mpsForeignAmt></mpsForeignAmt>
<mpsForeignCur></mpsForeignCur>
<mpsRate></mpsRate>
<installment_period></installment_period>
<installment_provider></installment_provider>
<installment_firstPayAmt></installment_firstPayAmt>
<installment_eachPayAmt></installment_eachPayAmt>
<installment_lastPayAmt></installment_lastPayAmt>
<airline_ticketNumber></airline_ticketNumber>
```

```
        <successcode>0</successcode>
        <mSchPayId></mSchPayId>
        <dSchPayId></dSchPayId>
        <MerchantId>123456</MerchantId>
        <promotionCode></promotionCode>
        <promotionRuleCode></promotionRuleCode>
        <promotionOriginalAmt></promotionOriginalAmt>
        <mpMemberId></mpMemberId>
        <isEWallet>T</isEWallet>
        <eWalletBrand>EOC</eWalletBrand>
        <panFirst6></panFirst6>
        <errMsg>Query Successfully</errMsg>
    </record>

    <!-- more records … … … -->
</records>
```

Query Request Log

The aim of this function is to query the payment request log.

Definition of Parameters in the Integration Page

Input /Return	Parameters (Required Fields are in Bold typeface) *Case Sensitive	Data Type	Expected Value	Descriptions
Input	**merchantId**	Number		The merchant ID we provide to you
	loginId	Text (30)		The loginId of using merchant API
	password	Text (15)		The password of using merchant API
	actionType		"QueryRequestLog"	The action type
(Optional Input)	**orderRef**	Text (35)		Merchant Reference Number
(Optional Input)	**periodType**	Text (1)		Period Type: M – Query by month D – Query by Day
(Optional Input)	**queryDate**	Number(8)		Query date format: MMYYYY - periodType is M DDMMYYYY - periodType is D
Return	requestDate	Date		The request date of transaction
	sessionId	Number		The sessionID of transaction
	ref	Number(15)		Merchant Reference Number
	cur	Number (3)		Transaction Currency i.e. "344" – HKD "840" – USD "156" – CNY (RMB) "392" – JPY "036" – AUD "978" – EUR "124" – CAD "446" – MOP "764" – THB "458" – MYR "410" – KRW "682" – SAR "784" – AED "096" – BND "356" – INR "702" – SGD "826" – GBP "901" – TWD "608" – PHP "360" – IDR "554" – NZD "704" – VND

	amt	Number (12,2)		Transaction Amount
	payerIp	Text(15)		IP address of payer
	ipCountry	Text(3)		Country of payer (e.g. HK) - if country is on high risk country list, an asterisk will be shown (e.g. MY*)
	deviceBrowser	Text(50)		Device Browser
	deviceOS	Text(50)		Device Operation System
	deviceMoblie	Text(1)		Device Mobile
	channelType	Text(5)		Channel Type SPN - Client post through browser SPV - Client post through browser DPS - Direct Server Side DPC - Direct Client Side
	pageCode	Text(10)		pageCode of payment page CPF - Payment Selection Page CPF2 - Credit Card Input Page CPV - 3D Verify Page CPVR - 3D Verification Result Page CPC - Result Page DPC - Payment Processing DPR - Result Redirect Page CPFD - DCC Selection Page

All the return parameters will be in XML format

Sample return XML file:

```
<records>
    <record>
        <requestDate>2016-01-20 10:17:19</requestDate>
        <sessionId>1135335</sessionId>
        <ref>Test</ref>
        <cur>344</cur>
        <amt>1</amt>
        <payerIp>192.168.77.28</payerIp>
        <ipCountry/>
        <deviceBrowser>Chrome</deviceBrowser>
```

```
        <deviceOS/>
        <deviceMoblie>F</deviceMoblie>
        <channelType>SPC</channelType>
        <pageCode>CPF</pageCode>
    </record>

    <!-- more records … … … -->
</records>
```

Cancel Payment

The aim of this function is to cancel the payment that the status of transaction in “Pending_feedback”. This function only supports on the payment method(s) below.

Payment method(s) that support “Cancel” action:

BCA-KLIKPAY, BCA-VA, BRI-EPAY, CIMB-CLICKS, MANDIRI-CLICKPAY and PERMATA-VA

Definition of Parameters in the Integration Page

Input /Return	Parameters (Required Fields are in Bold typeface) *Case Sensitive	Data Type	Expected Value	Descriptions
Input	**merchantId**	Number		The merchant ID we provide to you
	loginId	Text (30)		The loginId of using merchant API
	password	Text (15)		The password of using merchant API
	actionType		“Cancel”	The action type
	payRef	Text (35)		Payment Reference Number
Return	resultCode	Number	{“0”,”-1”}	0 - Request Successfully -1 – Request Failed
	orderStatus	Text (20)		The new order status after successfully request
	ref	Text		Merchant’s Order Reference Number
	payRef	Number		PayDollar transaction reference
	amt	Number (12,2)		Transaction Amt
	cur	Number (3)		Transaction Currency “344” – HKD “840” – USD “156” – CNY (RMB) “392” – JPY “036” – AUD “978” – EUR “124” – CAD “446” – MOP “764” – THB “458” – MYR “410” – KRW “682” – SAR “784” – AED “096” – BND “356” – INR “702” – SGD “826” – GBP “901” – TWD “608” – PHP “360” – IDR “554” – NZD “704” – VND
	errMsg	Text		Error Message

All the return parameters will be concatenated as in html request format by separate with **&**

Sample return string:

resultCode=0&orderStatus=Rejected&ref=Test&payRef=4888&amt=1.0&cur=344&errMsg=Cancel Successfully.

Settlement report request

The aim of this function is to generate settlement report.

Definition of Parameters in the Integration Page

Input /Return	Parameters (Required Fields are in Bold typeface) *Case Sensitive	Data Type	Expected Value	Descriptions
Input	**merchantId**	Number		The merchant ID we provide to you
	loginId	Text (30)		The loginId of using merchant API
	password	Text (15)		The password of using merchant API
	startDate	Number(14)	DDMMYYYY hhmmss	Report Start Date
	endDate	Number(14)	DDMMYYYY hhmmss	Report End Date
(Optional Input)	**queryType**	Text(1)	O/S	Report Type: O - Use date/time of the authorization or sales transaction to generate report (default) S - Use date/time of the settlement transaction to generate report and only query transactions that settleflag is T
Return	authdate	Number(14)		The date/time of the authorization or sales transaction
	capturedate	Number(14)		The date/time of the captured (after authorize) transaction
	batchid	Number(15)		Settlement batch ID
	settledate	Number(14)		The date/time of the settlement transaction
	payref	Number (14)		Unique number in Payment platform
	merref	Text(30)		Merchant order reference number
	authid	Text(6)		Approval code
	cur	Number		Bank Return Status code 2
	amt	Number(12,2)		Transaction Amount
	orderstatus	Text		The Holder Name of the Payment Account
	terminal	Text(100)		Bank Terminal id
	bankmid	Text(100)		Bank Merchant id

	settleflag	boolean	{“T”,”F”}	Settled or not
	src	Text		Secondary response code
	prc	Text		Primary response code
	bank	Text		Eg.CITIBANK,PPS
	bankref	Text		Bank reference number
	traceno	Text		System trace number
	currency	Text		Order currency
	remark	Text		Order Remark
	errMsg	Text		Error Message
	originalamt	Number(12,2)		Original Transaction Amount

URL of Testing Platform:

https://test.paydollar.com/b2cDemo/GenTxnXML

URL of Production Platform:

https://www.paydollar.com/b2c2/GenTxnXML

All the return parameters will be in XML format

Sample return XML file:

```
<?xml version=”1.0” encoding=’UTF-8”?>
<reports>
<report>
    <authdate>24092005223000</authdate>
    <capturedate>24092005230000</capturedate>
    <batchid>123456</batchid>
    <!-- more parameter … … … -->
</report>

<!-- more reports … … … -->
</reports>
```

Sample source code of HTML server-side posting on Java

As different type of programming language have different syntax, so we just propose one method to connect to our merchant API page. To connect, we suggest you to use server side posting:

Sample code for serverpost by using java:

// SET UP THE POST DATA

```
String postData =
"merchantId=1&loginId=testing&password=pwd&payRef=123456&actionType=Capture&amount=
1&";
```

// POST TO PAYMENT PAGE

```
strResult = ServerPost.post(postData,
http://test.paydollar.com:8080/b2cDemo/eng/merchant/api/orderApi.jsp );
```

// EXTRACT THE PAYMENT STATUS FROM STRRESULT

.....

// FINISH

```
**************************************************************************
public class ServerPost
{
       static public String post( String ip_postData, String ip_pageUrl)
       {
              try
              {
                     String strResult = "";
                     URL url = new URL(ip_pageUrl);

                     URLConnection con = url.openConnection(); //from secure site
                     if(con instanceof com.sun.net.ssl.HttpsURLConnection){
                        ((com.sun.net.ssl.HttpsURLConnection)con).setSSLSocketFactory
                                   ((SSLSocketFactory)SSLSocketFactory.getDefault());
                     }

                     con.setDoOutput(true);
                     con.setDoInput(true);

                     // Set request headers for content type and length
                     con.setRequestProperty(
                            "Content-type",
                            "application/x-www-form-urlencoded");
                     con.setRequestProperty(
```

```
                        "Content-length",
                        String.valueOf(ip_postData.length()));

                    // Issue the POST request
                    OutputStream outStream = con.getOutputStream();
                    outStream.write(ip_postData.getBytes());
                    outStream.flush();

                    // Read the response
                    InputStream inStream = con.getInputStream();

                   while (true)
                    {
                        int c = inStream.read();
                        if (c == -1)
                            break;
                        strResult = strResult + String.valueOf((char)c);
                    }

                    inStream.close();
                    outStream.close();

                    return strResult;
                }
                catch (Exception e)
                {
                    System.out.print(e.toString());
                    return null;
                }
        }
}
```

7 Exceptional Transaction Handling

This section explains various scenarios of transactions, other than good and successful transactions that may occur.

A) Unsuccessful data feed

This may occur if

1. Data feed URL is wrongly set up; or
2. Connection between PayDollar and merchant server is lost; or
3. Server of either side cannot process data feed correctly.

Since the bank has already determined the transaction status, the transaction is completed. Merchant can confirm the status by

1. Log on to Merchant Administration and retrieve the corresponding transactions in Transaction Details Report; or
2. Query the transaction status by using Merchant API.

B) Unsuccessful redirection to successUrl / failUrl / cancelUrl

This may occur if

1. Wrong / invalid returned URLs are set in the integration; or
2. Connection between the customer and merchant server is lost; or
3. Customer's computer hangs / restarts / loses power.

Since the bank has already determined the transaction status, the transaction is completed. Merchants should educate the customer to contact the merchant and confirm the transaction status with them when such case happens.

C) Incomplete 3D authentication transactions by customer

This may occur if

1. The customer closes the browser when he / she is required to enter 3D authentication information at issuer bank webpage; or
2. The customer cannot access 3D authentication page of issuer bank due to various reasons, e.g. disabled cookies.

The transaction status remains "Pending_3D", and payer authentication status remains "P". In

PayDollar production, a schedule job is set up to change the status from “Pending_3D” to “Rejected” from time to time. The PRC / SRC pair is also updated to 3 / 9999. Data feed is also sent out in the schedule job for these unsuccessful transactions.

Sometimes customers may return to PayDollar payment page / merchant site by pressing the ‘Back’ button of the browser and try again. The same merchant reference number is used for these retry transactions. Thus merchants may receive multiple data feeds regarding transactions with the same merchant reference number, with one success transaction followed by failed transactions. Merchants can choose to ignore the fail transactions with the same merchant reference once a successful transaction has been processed.

D) Incomplete 99BILL / ALIPAY / CHINAPAY / PPS / TENPAY / SCB / BAY / KTB / UOB / TMB / BBL iBanking / UPOP / M2U / CIMBCLICK / WECHAT transactions by customer

This may occur if

1. The customer closes the browser when he / she is required to enter 99BILL / ALIPAY / CHINAPAY / PPS / TENPAY / WECHAT account information at respective site; or
2. The customer cannot access 99BILL / ALIPAY / CHINAPAY / PPS / TENPAY / WECHAT / SCB / BAY / KTB / UOB / TMB / BBL iBanking / UPOP page due to various reasons, e.g. disabled cookies or 99BILL / ALIPAY / CHINAPAY / PPS / TENPAY / SCB / BAY / KTB / UOB / TMB / BBL iBanking / UPOP host is down.

The transaction status remains “Pending”. In production environment a schedule job is set up in our servers to change the status from “Pending” to “Rejected” from time to time. Data feed is also sent out in the schedule job for these unsuccessful transactions.

Sometimes customers may return to PayDollar payment page by pressing the ‘Back’ button of the browser and try again. The same merchant reference number is used for these retry transactions.

Thus merchants may receive multiple data feeds regarding transactions with the same merchant reference number, with one success transaction followed by failed transactions. Merchants can choose to ignore the fail transactions with the same merchant reference once a successful transaction has been processed.

8 PaySDK iOS Mobile SDK

This section explains integration of PaySDK iOS SDK in merchant iOS mobile application. Merchant have to provide iOS App bundle ID and android application id to AsiaPay to apple PaySDK service. PaySDK will be sharing unique RSA Public key related to merchant.

A) SDK Integration Steps

SDK requirement

Swift Version: 4.0 and above

iOS Version: 11.0 and above

SDK Configuration

Add AP_PaySDK.framework file into your project by adding the dependencies in Build Phases / Link Binary With Libraries :

add the “AP_PaySDK” framework

or

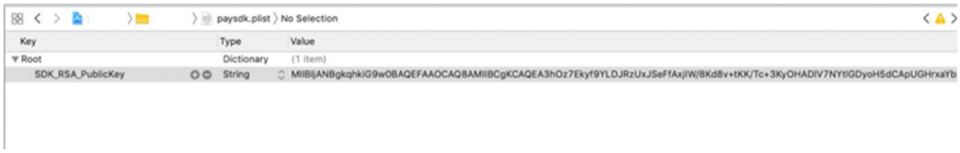
CocoaPods is a dependency manager for Cocoa projects. For usage and installation instructions, visit their website. To integrate PaySDK into your Xcode project using CocoaPods, specify it in your Podfile:

```
pod 'AP_PaySDK'
```

SDK Initialization

1. Create the paysdk.plist file in the app with the following attributes
   RSA Public Key(**SDK_RSA_Publickey**) :- Here you need to specify the RSA Public Key without header and footer. The value will be provided when the merchant apply the SDK service.

e.g



NOTE:Naming of property file and its attributes must be same.

2. Import sdk into the class

```
import AP_PaySDK
```

Note - Due to the limitation of App Transport Security(ATS) on http in iOS9.0, we need to add exception for `paydollar.com, xecure3d.com, pesopay.com, siampay.com, alipay.com, alipayobjects.com` in info.list

```
<key>NSAppTransportSecurity</key>
    <dict>
        <key>NSExceptionDomains</key>
        <dict>
            <key>paydollar.com</key>
            <dict>
                <key>NSIncludesSubdomains</key>
                <true/>
             <key>NSTemporaryExceptionAllowsInsecureHTTPLoads</key>
                <true/>
                <key>NSTemporaryExceptionMinimumTLSVersion</key>
                <string>TLSv1.0</string>
                <key>NSTemporaryExceptionRequiresForwardSecrecy</key>
                <false/>
            </dict>
          <key>xecure3d.com</key>
            <dict>
                <key>NSIncludesSubdomains</key>
                <true/>
                <key>NSTemporaryExceptionAllowsInsecureHTTPLoads</key>
                <true/>
                <key>NSTemporaryExceptionMinimumTLSVersion</key>
                <string>TLSv1.0</string>
                <key>NSTemporaryExceptionRequiresForwardSecrecy</key>
                <false/>
            </dict>
            <key>pesopay.com</key>
            <dict>
                <key>NSIncludesSubdomains</key>
                <true/>
                <key>NSTemporaryExceptionAllowsInsecureHTTPLoads</key>
                <true/>
                <key>NSTemporaryExceptionMinimumTLSVersion</key>
                <string>TLSv1.0</string>
```

```
            <key>NSTemporaryExceptionRequiresForwardSecrecy</key>
            <false/>
        </dict>
        <key>siampay.com</key>
        <dict>
            <key>NSIncludesSubdomains</key>
            <true/>
            <key>NSTemporaryExceptionAllowsInsecureHTTPLoads</key>
            <true/>
            <key>NSTemporaryExceptionMinimumTLSVersion</key>
            <string>TLSv1.0</string>
            <key>NSTemporaryExceptionRequiresForwardSecrecy</key>
            <false/>
        </dict>
        <key>alipay.com</key>
        <dict>
            <key>NSIncludesSubdomains</key>
            <true/>
            <key>NSTemporaryExceptionAllowsInsecureHTTPLoads</key>
            <true/>
            <key>NSTemporaryExceptionMinimumTLSVersion</key>
            <string>TLSv1.0</string>
            <key>NSTemporaryExceptionRequiresForwardSecrecy</key>
            <false/>
        </dict>
        <key>alipayobjects.com</key>
        <dict>
            <key>NSIncludesSubdomains</key>
            <true/>
            <key>NSTemporaryExceptionAllowsInsecureHTTPLoads</key>
            <true/>
            <key>NSTemporaryExceptionMinimumTLSVersion</key>
            <string>TLSv1.0</string>
            <key>NSTemporaryExceptionRequiresForwardSecrecy</key>
            <false/>
        </dict>
    </dict>
</dict>
```

or merchant can disable ATS by

```
<key>NSAppTransportSecurity</key>
 <dict>
      <key>NSAllowsArbitraryLoads</key><true/>
 </dict>
```

3. Add implementation of PaySDK protocol –

```
class ViewController: UIViewController , PaySDKDelegate
```

4. Instantiate PaySDK class

```
var paySDK = PaySDK.shared
```

5. Set paySDK delegate

```
paySDK.delegate = self
```

B) WebView Payment Call

Prepare Payment Call:

Initialize the PayData class and prepare the payment detail for the transaction.

```
paySDK.paymentDetails = PayData(
                            channelType: PayChannel.WEBVIEW,
                            payGate: PayGate.PAYDOLLAR,
                            envType: EnvType.SANDBOX,
                            merchantId: "1",
                            payType: payType.NORMAL_PAYMENT,
                            orderRef: "2018102409220001",
                            payMethod: "VISA",
                            lang: Language.ENGLISH,
                            currCode: Currency.HKD,
                            amount: 10,
                            remark: "",
                            extraData: [:])
```

Trigger the Payment:

Call the SDK to trigger the payment process. For WebView flow, it will display a In-App Webview for the payment.

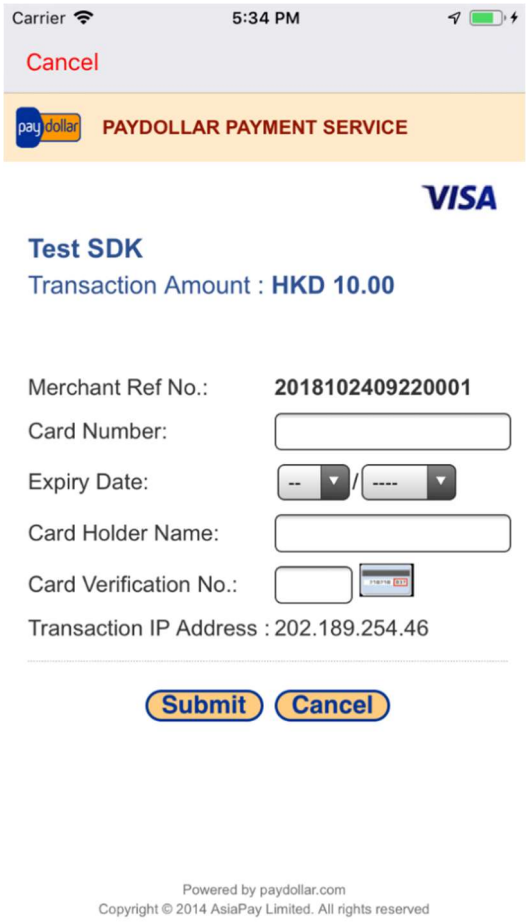
```
paySDK.process()
```

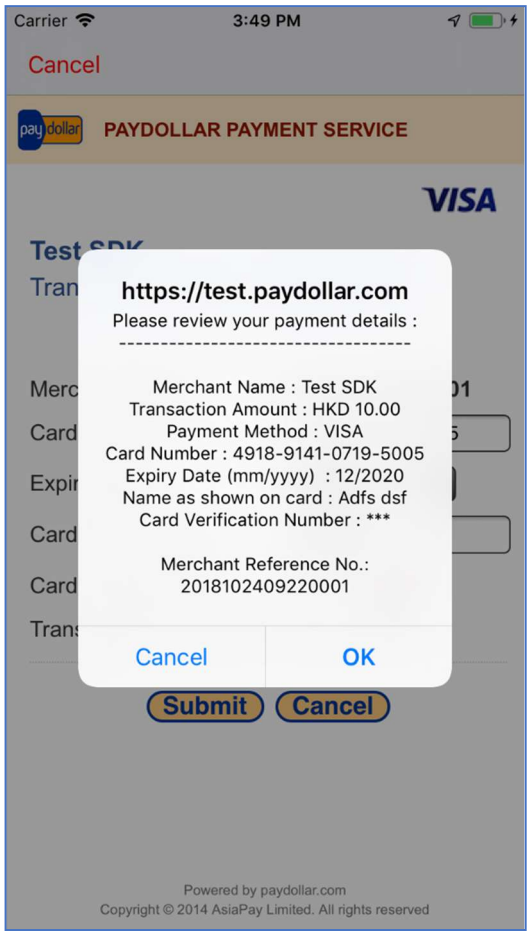
Prepare Response Handler:

Initialize a payment event handler to capture the payment response and result.

```
func paymentResult(result: PayResult) {


}
```

WebView Sample:





C) Direct Payment Call

Prepare Payment Call:

Initialize the PayData class and card details then prepare the payment detail for the transaction.

```
paySDK.paymentDetails = PayData(
                            channelType: PayChannel.DIRECT,
                            envType: EnvType.SANDBOX,
                            amount : "10",
```

```
                            payGate: PayGate.PAYDOLLAR,
                            currCode: currencyCode.HKD,
                            payType: payType.NORMAL_PAYMENT,
                            orderRef: "2018102409220001”,
                            payMethod: "VISA",
                            lang: Language.ENGLISH,
                            merchantId: "1",
                            remark: "",
                            extraData :[:])
paySDK. paymentDetails.cardDetails = CardDetails(
                            cardHolderName: “abc abc”,
                            cardNo: "4918914107195011”,
                            expMonth: “11”,
                            expYear: “2011”,
                            securityCode: “123”)
```

Trigger the Payment:

Call the SDK to trigger the payment process. For WebView flow, it will display a In-App Webview for the payment.

```
paySDK.process()
```

Prepare Response Handler:

Initialize a payment event handler to capture the payment response and result.

```
func paymentResult(result: PayResult) {

}
```

D) Installment Payment

Prepare Payment Call:

For Installment Payment, initialize the PayData class and prepare the payment detail for the transaction same as in with Webview and Direct Payment Call and add below parameters.

```
extraData:  [
         "installment_service " : "T",
```

```
        "installment_period": 3,
        "installOnly": "T"]
```

E) Schedule Pay

For Schedule Payment, initialize the PayData class and prepare the payment detail for the transaction same as in with Webview and Direct Payment Call and add below parameters.

```
extraData: [
        "appId" : "SP",
        "appRef" : "2018102409220001",
        "schType" : "Day",
        "schStatus" : "Active",
        "nSch" : "1",
        "sMonth" : "4",
        "sDay" : "26",
        "sYear" : "2019",
        "eMonth" : "",
        "eDay" : "",
        "eYear" : "",
        "name" : "Name",
        "email" : "kit@igears.com.hk"]
```

F) Member Pay

For New Member:

For Member Pay Payment, initialize the PayData class and prepare the payment detail for the transaction same as in with Webview and Direct Payment Call and add below parameters. Add parameter addNewMember. If merchant making payment for first time then value should be true

```
extraData: [
        "addNewMember": true,
        "memberPay_service": "T",
        "memberPay_memberId": "MPTEST_1",
        "memberId":"MPTEST_1”]
```

For Old Member:

For Member Pay Payment, initialize the PayData class and prepare the payment detail for the transaction same as in with Webview and Direct Payment Call and add below parameters. Add parameter addNewMember. If merchant making payment not for first time then value should be false and below will be extra parameters.

```
extraData: [
        "addNewMember": false,
        "memberPay_service": "T",
        "memberPay_memberId": "MPTEST_1",
        "memberId":"MPTEST_1",
        "token":"8de18f7b17fabe5bb682e792e319c0e411b528f58926"]
```

G) Promo Pay

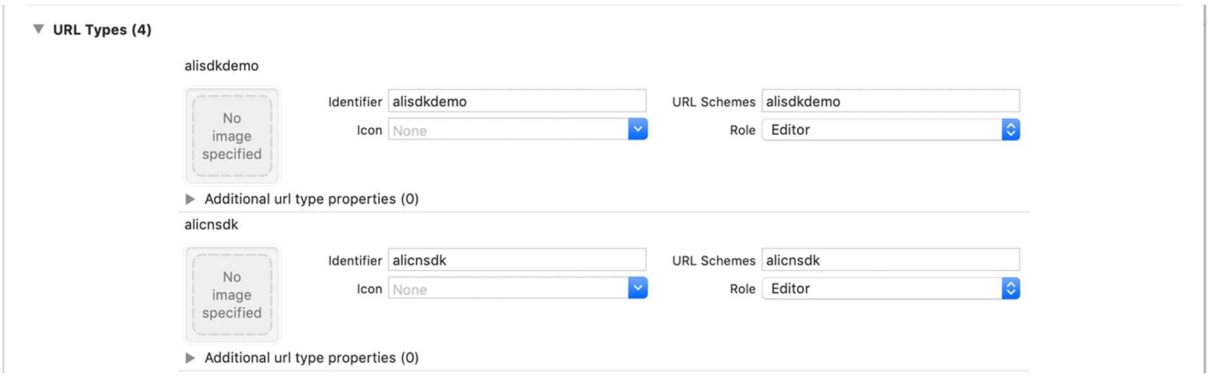
For Promo Pay Payment, initialize the PayData class and prepare the payment detail for the transaction same as in with Webview and Direct Payment Call and add below parameters.

```
extraData: [
        "promotion ": "T",
        "promotionCode ": "TEST1",
        "promotionRuleCode " : " TESTR25",
        "promotionOriginalAmt ":""]
```

H) Alipay Global Payment Call

Initialization Step:

1. Add URL Type in info .plist file



2. In AppDelegate file add

```
var paySDK = PaySDKClass.shared
```

And

```
func application(_ app: UIApplication, open url: URL, options:
[UIApplication.OpenURLOptionsKey : Any] = [:]) -> Bool {
            paySDK.processOrder(url: url)
            return true;
}
```

Prepare Payment Call:

Initialize the PayData class and prepare the payment detail for the transaction.

```
paySDK.paymentDetails = PayData(
                        channelType: PayChannel.DIRECT,
                        envType: EnvType.SANDBOX,
                        amount: 0.1,
                        payGate: PayGate.PAYDOLLAR,
                        currCode: currencyCode.HKD,
                        payType: payType.NORMAL_PAYMENT,
                        orderRef: "2018102409220001",
                        payMethod: "ALIPAYAPP",
                        lang: Language.ENGLISH,
                        merchantId: "1",
                        remark: "test",
                        extraData : [:])
```

Trigger the Payment:

Call the SDK to trigger the payment process.

```
paySDK.process()
```

If app is installed on device then get reponse in AppDelegate openurl function

```
func application(_ app: UIApplication, open url: URL, options:
[UIApplication.OpenURLOptionsKey : Any] = [:]) -> Bool {
        paySDK.processOrder(url: url)
        return true;
}
```

Prepare Response Handler:

Initialize a payment event handler to capture the payment response and result.

```
func paymentResult(result: PayResult) {
}
```

I) Alipay HK Payment Call

Initialization Step:

1. Add URL Type in info .plist file

2. In AppDelegate file add

```
var paySDK = PaySDKClass.shared
```

And

```
func application(_ app: UIApplication, open url: URL, options:
[UIApplication.OpenURLOptionsKey : Any] = [:]) -> Bool {
        paySDK.processOrder(url: url)
        return true;
}
```

Prepare Payment Call:

Initialize the PayData class and prepare the payment detail for the transaction.

```
paySDK.paymentDetails = PayData(
                        channelType: PayChannel.DIRECT,
                        envType: EnvType.SANDBOX,
                        amount: 0.1,
                        payGate: PayGate.PAYDOLLAR,
                        currCode: currencyCode.HKD,
                        payType: payType.NORMAL_PAYMENT,
                        orderRef: "2018102409220001",
                        payMethod: "ALIPAYHKAPP",
                        lang: Language.ENGLISH,
                        merchantId: "1",
                        remark: "test",
                        extraData : [:])
```

Trigger the Payment:

Call the SDK to trigger the payment process.

```
paySDK.process()
```

If app is installed on device then get reponse in AppDelegate openurl function

```
func application(_ app: UIApplication, open url: URL, options:
[UIApplication.OpenURLOptionsKey : Any] = [:]) -> Bool {
        paySDK.processOrder(url: url)
        return true;
}
```

Prepare Response Handler:

Initialize a payment event handler to capture the payment response and result.

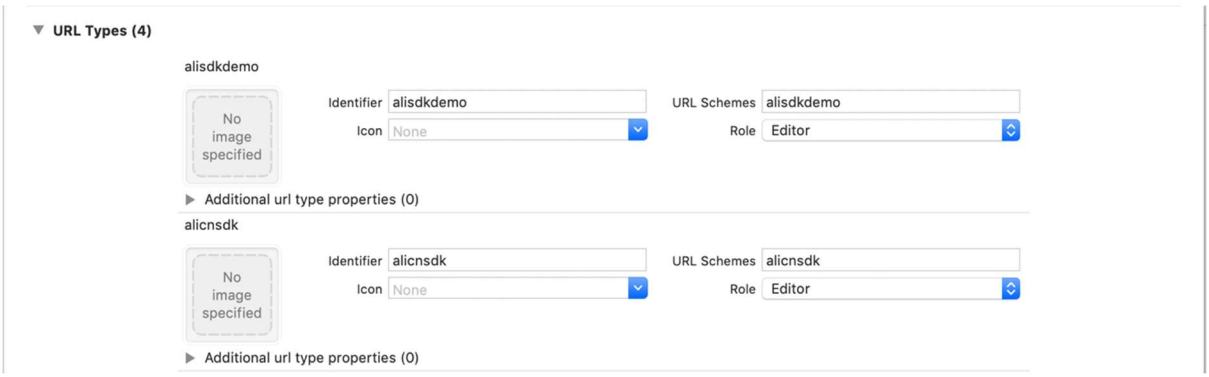
```
func paymentResult(result: PayResult) {
}
```

This method returns object of PayResult class.

J) Alipay China Payment Call

Initialization Step:

1. Add URL Type in info .plist file



2. In AppDelegate file add

```
var paySDK = PaySDKClass.shared
```

And

```
func application(_ app: UIApplication, open url: URL, options:
[UIApplication.OpenURLOptionsKey : Any] = [:]) -> Bool {
        paySDK.processOrder(url: url)
        return true
}
```

Prepare Payment Call:

Initialize the PayData class and prepare the payment detail for the transaction.

```
paySDK.paymentDetails = PayData(
                            channelType: PayChannel.DIRECT,
                            envType: EnvType.SANDBOX,
                            amount: 0.1,
                            payGate: PayGate.PAYDOLLAR,
                            currCode: currencyCode.HKD,
                            payType: payType.NORMAL_PAYMENT,
                            orderRef: "2018102409220001",
                            payMethod: "ALIPAYCNAPP",
                            lang: Language.ENGLISH,
                            merchantId: "1",
                            remark: "test",
                            extraData : [:])
```

Trigger the Payment:

Call the SDK to trigger the payment process.

```
paySDK.process()
```

If app is installed on device then get reponse in AppDelegate openurl function

```
func application(_ app: UIApplication, open url: URL, options:
[UIApplication.OpenURLOptionsKey : Any] = [:]) -> Bool {
        paySDK.processOrder(url: url)
        return true;
}
```

Prepare Response Handler:

Initialize a payment event handler to capture the payment response and result.

```
func paymentResult(result: PayResult) {
}
```

K) WeChat Pay Payment Call

Prepare Payment Call:

Initialize the PayData class and prepare the payment detail for the transaction.

```
paySDK.paymentDetails = PayData(
```

```
                            channelType: PayChannel.DIRECT,
                            envType: EnvType.SANDBOX,
                            amount: 0.1,
                            payGate: PayGate.PAYDOLLAR,
                            currCode: currencyCode.HKD,
                            payType: payType.NORMAL_PAYMENT,
                            orderRef: "2018102409220001",
                            payMethod: "WECHATAPP",
                            lang: Language.ENGLISH,
                            merchantId: "1",
                            remark: "test",
                            extraData : [:])
```

Trigger the Payment:

Call the SDK to trigger the payment process.

```
paySDK.process()
```

Prepare Response Handler:

Initialize a payment event handler to capture the payment response and result.

```
func paymentResult(result: PayResult) {
}
```

L) 3DS 2.0 Payment Call

Prepare Payment Call:

Initialize the PayData class and card details then prepare the payment detail for the transaction.

```
paySDK.paymentDetails = PayData(
                            channelType: PayChannel.DIRECT,
                            envType: EnvType.SANDBOX,
                            amount : "10",
                            payGate: PayGate.PAYDOLLAR,
                            currCode: currencyCode.HKD,
                            payType: payType.NORMAL_PAYMENT,
                            orderRef: "2018102409220001",
                            payMethod: "3DS2.0",
                            lang: Language.ENGLISH,
                            merchantId: "1",
```

```
                            remark: "",
                            extraData :[:])
paySDK. paymentDetails.cardDetails = CardDetails(
                            cardHolderName: “abc abc”,
                            cardNo: "4918914107195011”,
                            expMonth: “11”,
                            expYear: “2011”,
                            securityCode: “123”)
var threeDSParams = ThreeDSParams()
threeDSParams.apiUsername = "username"
threeDSParams.apiPassword = "password"
.
.
.
.
.
.
.
.
.

paySDK.paymentDetails.threeDSParams = threeDSParams
```

Trigger the Payment:

Call the SDK to trigger the payment process. For WebView flow, it will display a In-App Webview for the payment.

```
paySDK.process()
```

Prepare Response Handler:

Initialize a payment event handler to capture the payment response and result.

```
func paymentResult(result: PayResult) {
}
```

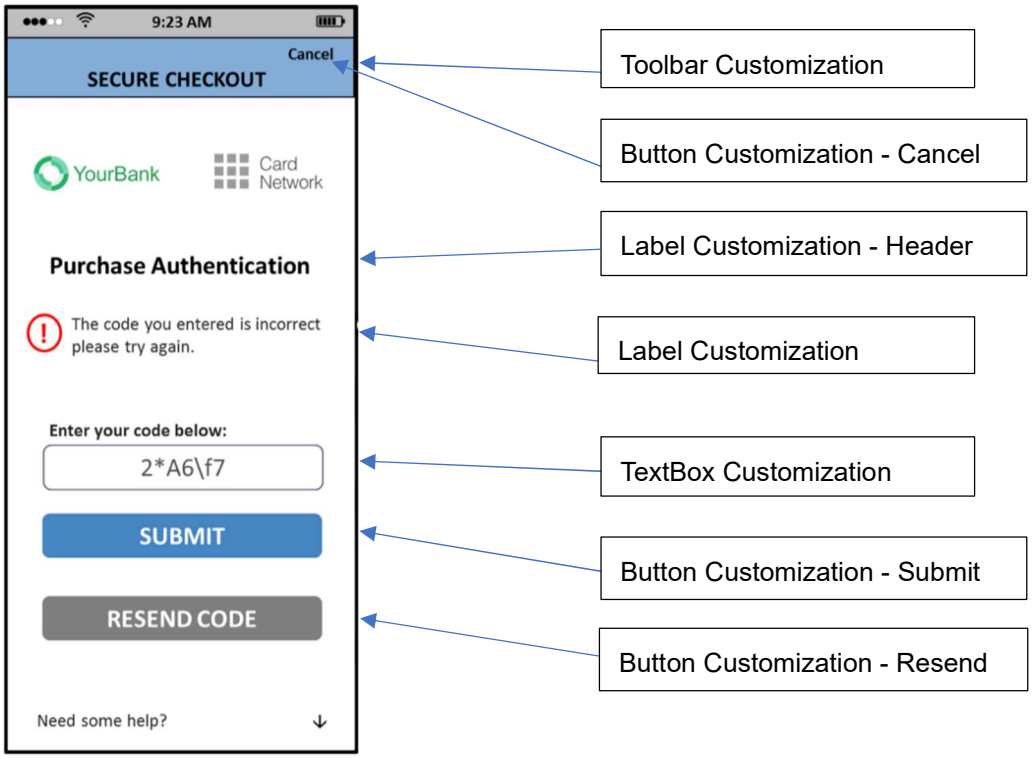
List of Parameters merchant can set	
apiUsername	threeDSAcctCreateDate
apiPassword	threeDSAcctAgeInd

threeDSCustomerEmail	threeDSAcctLastChangeDate
threeDSDeliveryEmail	threeDSAcctLastChangeInd
threeDSMobilePhoneCountryCode	threeDSAcctPwChangeDate
threeDSMobilePhoneNumber	threeDSAcctPwChangeInd
threeDSHomePhoneCountryCode	threeDSAcctPurchaseCount
threeDSHomePhoneNumber	threeDSAcctCardProvisionAttempt
threeDSWorkPhoneCountryCode	threeDSAcctNumTransDay
threeDSWorkPhoneNumber	threeDSAcctNumTransYear
threeDSBillingCountryCode	threeDSAcctPaymentAcctDate
threeDSBillingState	threeDSAcctPaymentAcctInd
threeDSBillingCity	threeDSAcctShippingAddrLastChangeDate
threeDSBillingLine1	threeDSAcctShippingAddrLastChangeInd
threeDSBillingLine2	threeDSAcctIsShippingAcctNameSame
threeDSBillingLine3	threeDSAcctIsSuspiciousAcct
threeDSBillingPostalCode	threeDSAcctAuthMethod
threeDSShippingDetails	threeDSAcctAuthTimestamp
threeDSShippingCountryCode	threeDSDeliveryTime
threeDSShippingState	threeDSPreOrderReason
threeDSShippingCity	threeDSPreOrderReadyDate
threeDSShippingLine1	threeDSGiftCardAmount
threeDSShippingLine2	threeDSGiftCardCurr
threeDSShippingLine3	threeDSGiftCardCount

Type of Challenge for Payment via 3DS 2.0

- Challenge Information TextBox
- Single and Multi-select Fields
- Out of Band
- HTML

UI - Customization for Challenge View for Merchant



UI-Customization for paySDK

Merchant can set the UiCustomization at the paySDK initialization process.

```
let customization = UiCustomization()
let submitButtonCustomization = ButtonCustomization.init("Courier", "#FF0000", 15,
"#d3d3d3", 4)
let resendButtonCustomization = ButtonCustomization.init("Courier", "#FF0000", 15,
"#d3d3d3", 4)
let cancelButtonCustomization = ButtonCustomization.init("Courier", "#FF0000", 15,
"#d3d3d3", 4)
let nextButtonCustomization = ButtonCustomization.init("Courier", "#FF0000", 15,
"#d3d3d3", 4)
let continueButtonCustomization = ButtonCustomization.init("Courier", "#FF0000",
15, "#d3d3d3", 4)
let labelCustomization = LabelCustomization.init("Courier", "FF0000", 14, "FF0000",
"Courier", 20)
let textboxCustomization = TextBoxCustomization.init("Courier", "#FF0000", 14, 5,
"#d3d3d3", 4)
let toolBarCustomization = ToolbarCustomization.init("Courier", "#FF0000", 12,
"#d3d3d3", "HEADER_LABEL")
try! customization.setButtonCustomization(submitButtonCustomization, .SUBMIT)
```

```
try! customization.setButtonCustomization(resendButtonCustomization, .RESEND)
try! customization.setButtonCustomization(cancelButtonCustomization, .CANCEL)
try! customization.setButtonCustomization(nextButtonCustomization, .NEXT)
try! customization.setButtonCustomization(continueButtonCustomization, .CONTINUE)
try! customization.setLabelCustomization(labelCustomization)
try! customization.setTextBoxCustomization(textboxCustomization)
try! customization.setToolbarCustomization(toolBarCustomization)
paySDK.customization = customization
```

Class PayData properties details:

Input Parameters	Expected Value and Description
channelType	Channel Type (PayChannel) – WebView
payGate	Name of payment gateway i.e. Paydollar or Pesopay or Siampay
envType	Production or Sandbox
merchantId	ID received after registration on Asiapay merchant portal
payType	Payment Type (Normal / Hold)
orderRef	Reference number that merchant wants to attach to transaction
payMethod	Payment Method ID - please reference to Integration Guide for detail values
lang	Language of Payment Page
currency	Transaction currency, e.g. HKD, INR etc.
amount	Transaction amount
remark	More details about transaction to tag it. This is optional.
extraData	Extra Parameter For **Installment Pay** installment_service – installment service indicator i.e. T / F installment_period – In number of months installOnly – Control the mandatory of installment payment i.e. T / F For **Schedule Pay** appId - Application Id appRef - appref is the reference for the whole schedule pay which will be used only one timecardHolder - Name available on card schType - Schedule Type e.g. Day / Month /Year schStatus - Schedule Status e.g. Active / Suspended nSch - Number of schedule sMonth - Schedule start month sDay - Schedule start day sYear - Schedule start year eMonth - Schedule end month eDay - Schedule end day eYear - Schedule end year For **Member Pay** addNewMember – false or true memberPay_ service - T or F

	memberPay_memberId - Member Id actionType - Generate Token memberId - Member Id memberPay_Token – Static Token For **Promo Pay** promotion – Promotion service indicator i.e T/F promotionCode – The promotion campaign code promotionRuleCode – The promotion rule code (Optional) promotionOriginalAmt – The promotion original amount (Optional)

Class EnvBase values details:

EnvBase Parameter	Value and Details
PayChannel	SDK payment channel WEBVIEW: WebView redirect payment DIRECT: Direct server payment
PayGate	SDK supported payment gateway PAYDOLLAR: PayDollar SIAMPAY: SiamPay PESOPAY: PesoPay
EnvType	SDK environment configuration PRODUCTION: Production environment SANDBOX: Sandbox testing environment
PayType	SDK payment type NORMAL_PAYMENT: Sales payment HOLD_PAYMENT: Authorize payment
Currency	SDK supported currency list HKD, USD, SGD, RMB, JPY, TWD, AUD, EUR, GBP, CAD, MOP, PHP, THB, MYR, IDR, KRW, BND, NZD, SAR, AED, BRL, INR, TRY, ZAR, VND, DKK, ILS, NOK, RUB, SEK, CHF, ARS, CLP, COP, CZK, EGP, HUF, KZT, LBP, MXN, NGN, PKR, PEN, PLN, QAR, RON, UAH, VEF, LKR, KWD

Class PayResult properties details:

PayResult Output Property Parameter	Description
successcode	Status of transaction.
channel	Channel Type (PayChannel) – WebView, Direct
resultCode	Transaction status code.
resultMsg	Message to describe more about transaction status. Message may come blank if the transaction status is Success.
orderID	Order ID generated by AsiaPay System
orderRef	Order Reference number set before initializing payment call in PayData object.
bankRef	Bank Order Reference number
prc	Primary response code Refer to Integration Guide Appendix A (PRC)
src	Secondary response code Refer to Integration Guide Appendix A (SRC)
txTime	Transaction Time
amount	Transaction amount
currency	Transaction currency, e.g. HKD, INR etc.
extraData	Extra Response data

9 PaySDK Android Mobile SDK

This section explains integration of PaySDK iOS SDK in merchant android mobile application.

A) SDK Integration Steps

SDK requirement

JAVA SDK Version: 1.8

Android SDK Version: 19 (Minimum), 28 (Target)

SDK Configuration

- Copy **PaySDK-1.0.0.aar** to libs folder.
- Add below line to project's gradle file:

```
repositories {
 flatDir {
   dirs 'libs'
    }
}
```

- Add below lines in the dependencies to project's gradle file:

```
implementation(name: 'PaySDK-1.0.0', ext: 'aar')
```

- Add following libraries

```
implementation fileTree(dir: 'libs', include: ['*.jar'])
implementation 'com.google.code.gson:gson:2.3.1'
implementation 'org.bouncycastle:bcprov-jdk15on:1.60'
implementation 'com.android.volley:volley:1.1.1'
implementation 'com.fasterxml.jackson.core:jackson-core:2.7.3'
implementation 'com.fasterxml.jackson.core:jackson-annotations:2.7.3'
implementation 'com.fasterxml.jackson.core:jackson-databind:2.7.3'
implementation 'org.greenrobot:eventbus:3.0.0'
```

SDK Initialization:

Create the paysdk.properties file in the assets folder with the following attributes

RSA Public Key(SDK) :- Here you need to specify RSA Public Key value by removing Header and footer of public key. The value will be provided when the merchant apply the SDK service.

e.g.

```
sdk_rsa_publickey=MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEA3hOz7Ekyf9YLDJ
RzUxJSeFfAxjIW/8Kd8v+tKK/Tc+3KyOHADlV7NYtlGDyoH5dCApUGHrxaYbzV6VzhC+kgQIaL7m
jzMXX2uK03QT8Gmcef4C20fUMpy3hrIRH5MR32THNI3IiQG7upkuKhwQSwvEz0PxTeIznCBvSY/7
wgCC6CUhHt+INZYaYpNRiozy39+yHXvgJu4N+I7VQpXJ6b4QjARk6nq6dBlUge4HokZgQzfKL86P
JfzOjIDDKABkbZcZ96AL48E1jvgXlqV0ZA2k81IEVQzJxEiuERcPpSTXzB9H3f8+sx8DWxUO/CDI
iNI+XOt54sBMelPEFQgEjkCQIDAQAB
```

NOTE: Naming of property file and its attributes must be same.

1. Declare Object of SDK

```
PaySDK paySDK;
```

2. Initialize SDK in a background task, other than main UI thread. For this create an asyn task. For 3D Secure 2.0 transaction, there can be scenario where ACS can ask for more details to authenticate the user. Hence as per the data received from ACS, SDK launch a screen to process challenge User Interface to user.
   Merchant can customise this UI by putting UI related info like text – font size, font name etc. in UiCustomisation object.

```
private class initaliseSDK extends AsyncTask<String, Void, String> {
       @Override
       protected String doInBackground(String... params) {
          Factory factory = new com.asiapay.sdk.integration.xecure3ds.Factory();
          paySDK = new PaySDK(getApplicationContext());
          return "";
       }
       @Override
       protected void onPostExecute(String result) {
       //Cancel ProgressDialog if displayed before initializing sdk
       }
       @Override
       protected void onPreExecute() {
          //show progress dialog if want
       }
       @Override
       protected void onProgressUpdate(Void... values) {
       }
}
```

3. Call the above async task:

```
new initaliseSDK().execute("");
```

B) Webview Payment Call

Prepare Payment Call :

Initialize the PayData object and prepare the payment detail for the transaction.

```
PayData payData = new PayData();
payData.setChannel(EnvBase.PayChannel.WEBVIEW);
payData.setEnvType(EnvBase.EnvType.SANDBOX);
payData.setAmount("10");
payData.setPayGate(EnvBase.PayGate.PAYDOLLAR);
payData.setCurrCode(EnvBase.Currency.HKD);
payData.setPayType(EnvBase.PayType.NORMAL_PAYMENT);
payData.setOrderRef("2018102409220001");
payData.setPayMethod("VISA");
payData.setMerchantId("1");
payData.setRemark("");
payData.setLang(EnvBase.Language.ENGLISH);
paySDK.setRequestData(payData);
```

Prepare Response Handler :

Initialize a payment event handler to capture the payment response and result.

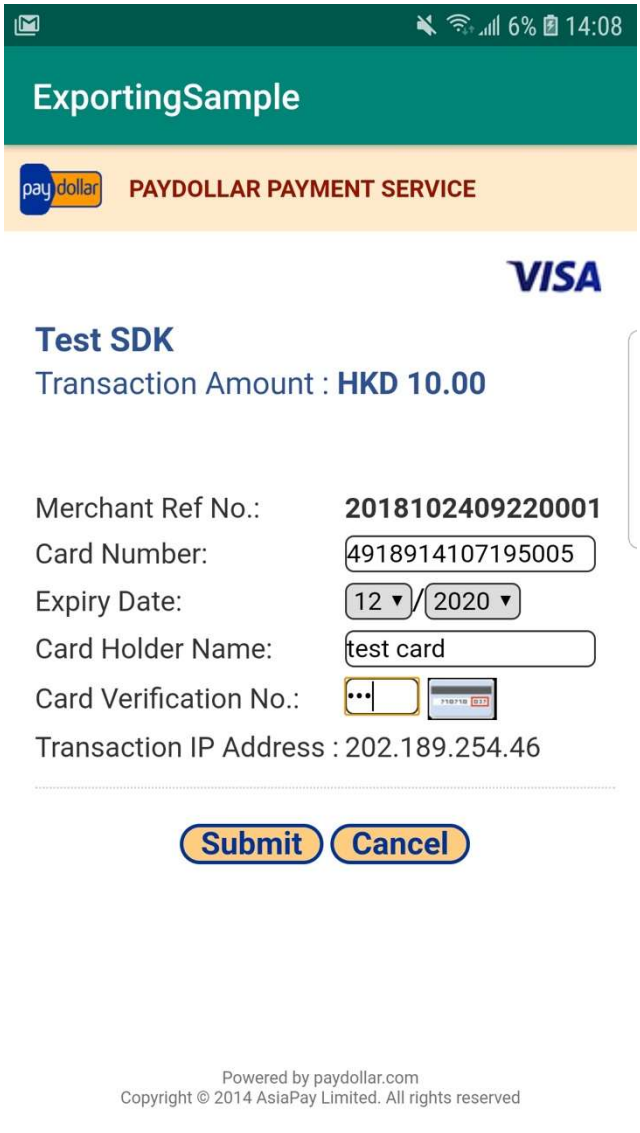
```
paySDK.responseHandler(new PaymentResponse() {
    @Override
    public void getResponse(PayResult payResult) {
    }
    @Override
    public void onError(data: Data) {
    }
});
```

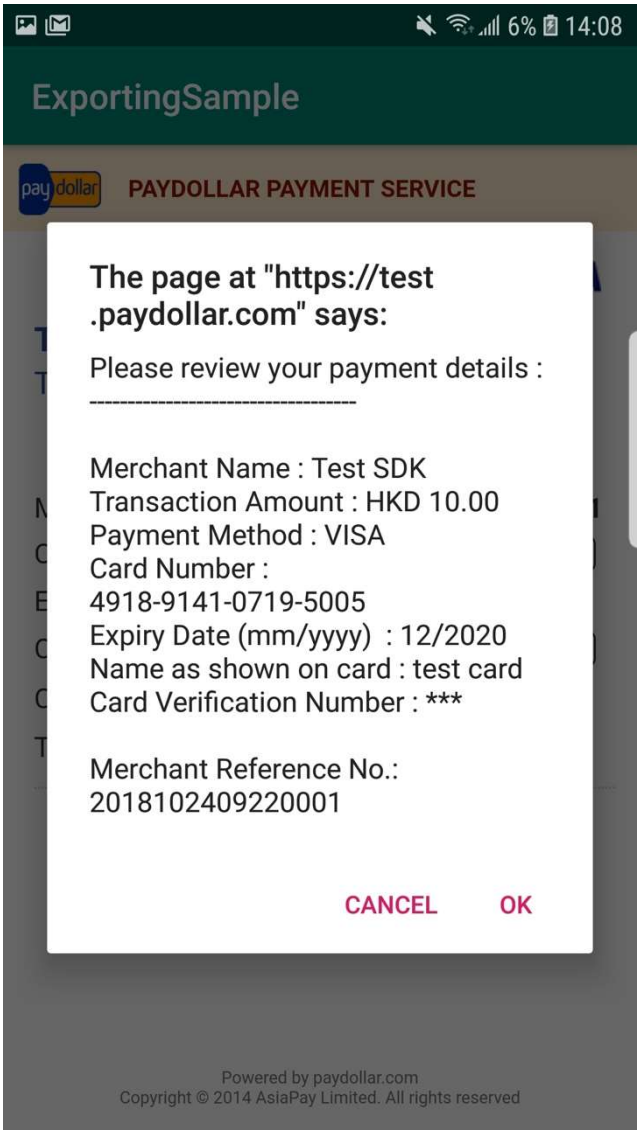
Trigger the Payment:

Call the SDK to trigger the payment process. For WebView flow, it will display a In-App Webview for the payment.

```
paySDK.process();
```

WebView Sample :





Avoid memory leak :-

Here all unused objects will be set to null and clean up the payData and payResult object.

```
paySDK.cleanMemory();
```

C) Direct Payment Call

Prepare Payment Call:

Initialize the PayData class and prepare the payment detail for the transaction.

```
payData = new PayData();
payData.setChannel(EnvBase.PayChannel.DIRECT);
payData.setEnvType(EnvBase.EnvType.SANDBOX);
payData.setAmount("15");
payData.setPayGate(EnvBase.PayGate.PAYDOLLAR);
payData.setCurrCode(EnvBase.Currency.HKD);
payData.setPayType(EnvBase.PayType.NORMAL_PAYMENT);
payData.setOrderRef("2018102409220001");
payData.setPayMethod("VISA");
payData.setLang(EnvBase.Language.ENGLISH);
payData.setMerchantId("1");
payData.setRemark(" ");

CardDetails cardDetails=new CardDetails();
cardDetails.setCardNo("4548890133258926");
cardDetails.setEpMonth("08");
cardDetails.setEpYear("2020");
cardDetails.setSecurityCode("123");
cardDetails.setCardHolder("test Card");
payData.setCardDetails(cardDetails);

paySDK.setRequestData(payData);
```

Prepare Response Handler :

Initialize a payment event handler to capture the payment response and result.

```
paySDK.responseHandler(new PaymentResponse() {
    @Override
    public void getResponse(PayResult payResult) {
    }
    @Override
    public void onError(data: Data) {
    }
```

```
});
```

Trigger the Payment:

Call the SDK to trigger the payment process.

```
paySDK.process();
```

D) Installment Payment Call

For Installment Pay with Webview & Direct Payment Call follow same process of (1.1 & 1.2) reps. & add below extra params.

```
extraData.put("installment_service","T");
extraData.put("installment_period","6");
extraData.put("installOnly","T");
```

E) Schedule Pay

For Schedule Pay with Webview & Direct Payment Call follow same process (1.1 & 1.2) reps. & add below extra params

```
extraData.put("schType","Day");  // Value could be Day/Month/Year
extraData.put("schStatus","Active"); // Value could be Active/Suspend
extraData.put("nSch","1");
extraData.put("sMonth","4");
extraData.put("sDay","26");
extraData.put("sYear","2019");
extraData.put("eMonth","");
extraData.put("eDay","");
extraData.put("eYear","");
extraData.put("name","Name");
extraData.put("email","kit@igears.com.hk");
extraData.put("appId","SP");
extraData.put("appRef","2018102409220001");   //appRef should be used only once
```

F) Member Pay

For New Member:

For Member Pay Payment, initialize the PayData class and prepare the payment detail for the transaction same as in with Webview and Direct Payment Call and add below parameters. Add parameter addNewMember. If merchant making payment for first time then value should be true

```
extraData.put("memberPay_service","T");
extraData.put("memberPay_memberId","member03");
extraData.put("addNewMember","True");
```

For Old Member:

For Member Pay Payment, initialize the PayData class and prepare the payment detail for the transaction same as in with Webview and Direct Payment Call and add below parameters. Add parameter addNewMember. If merchant making payment not for first time then value should be false and below will be extra parameters.

```
extraData.put("memberPay_service","T");
extraData.put("memberPay_memberId","member03");
extraData.put("addNewMember","False");
extraData.put("memberPay_token","7ff6c7cc1f03fd9889b000843a77e8f43e4467578926");
```

G) Promo Pay

For Promo Pay with Webview Call follow the same process of (1.1) & add the below extra params.

```
extraData.put("promotion","T");
extraData.put("promotionCode","TEST1");
extraData.put("promotionRuleCode","TESTR25");
extraData.put("promotionOriginalAmt","");
```

H) Alipay Global Payment Call

SDK Integration Steps:

1) Copy **alipaySdk-15.5.9-20181123210601.aar** to lib folder.
2) Add below lines in the dependencies to app's gradle file:

```
implementation(name: 'alipaySdk-15.5.9-20181123210601', ext: 'aar')
implementation "com.squareup.okhttp3:okhttp:3.12.0"
```

Initiating Order Payment for Alipay:

To initiate order payment, we need to instantiate PayReq class and call public method pay() by passing the payment related data to this.

Step 1: Prepare object that contains details to initialize payment.

```
PayData payData = new PayData();
payData.setChannel(EnvBase.PayChannel.DIRECT);
payData.setEnvType(EnvBase.EnvType.SANDBOX);
payData.setPayGate(EnvBase.PayGate.PAYDOLLAR);
payData.setCurrCode(EnvBase.Currency.HKD);
payData.setPayType(EnvBase.PayType.NORMAL_PAYMENT);
payData.setLang(EnvBase.Language.ENGLISH);
payData.setAmount("1");
payData.setPayMethod("ALIPAYAPP");
payData.setMerchantId("1");
payData.setOrderRef("2018102409220001");
payData.setRemark("payment remark");
payData.setActivity(MainActivity.this);
paySDK.setRequestData(payData);
```

Prepare Response Handler :

Initialize a payment event handler to capture the payment response and result.

```
paySDK.responseHandler(new PaymentResponse() {
    @Override
    public void getResponse(PayResult payResult) {
    }
    @Override
    public void onError(data: Data) {
    }
});
```

This method returns object of PayResult class.

Trigger the Payment:

Call the SDK to trigger the payment process.

```
paySDK.process();
```

I) Alipay HK Payment Call

Prepare Payment Call :

Initialize the PayData object and prepare the payment detail for the transaction.

```
PayData payData = new PayData();
payData.setChannel(EnvBase.PayChannel.DIRECT);
payData.setEnvType(EnvBase.EnvType.SANDBOX);
payData.setPayGate(EnvBase.PayGate.PAYDOLLAR);
payData.setCurrCode(EnvBase.Currency.HKD);
payData.setPayType(EnvBase.PayType.NORMAL_PAYMENT);
payData.setLang(EnvBase.Language.ENGLISH);
payData.setAmount("1");
payData.setPayMethod("ALIPAYHKAPP");
payData.setMerchantId("1");
payData.setOrderRef("2018102409220001");
payData.setRemark("payment remark");
payData.setActivity(MainActivity.this);
paySDK.setRequestData(payData);
```

Prepare Response Handler :

Initialize a payment event handler to capture the payment response and result.

```
paySDK.responseHandler(new PaymentResponse() {
    @Override
    public void getResponse(PayResult payResult) {
    }
    @Override
    public void onError(data: Data) {
    }
});
```

This method returns object of PayResult class.

Trigger the Payment:

Call the SDK to trigger the payment process.

```
paySDK.process();
```

J) Alipay China Payment Call

Prepare Payment Call :

Initialize the PayData object and prepare the payment detail for the transaction.

```
PayData payData = new PayData();
payData.setChannel(EnvBase.PayChannel.DIRECT);
payData.setEnvType(EnvBase.EnvType.SANDBOX);
payData.setPayGate(EnvBase.PayGate.PAYDOLLAR);
payData.setCurrCode(EnvBase.Currency.RMB);
payData.setPayType(EnvBase.PayType.NORMAL_PAYMENT);
payData.setLang(EnvBase.Language.ENGLISH);
payData.setAmount("1");
payData.setPayMethod("ALIPAYCNAPP");
payData.setMerchantId("1");
payData.setOrderRef("2018102409220001");
payData.setRemark("payment remark");
payData.setActivity(MainActivity.this);
paySDK.setRequestData(payData);
```

Prepare Response Handler :

Initialize a payment event handler to capture the payment response and result.

```
paySDK.responseHandler(new PaymentResponse() {
    @Override
    public void getResponse(PayResult payResult) {
    }
    @Override
    public void onError(data: Data) {
    }
});
```

This method returns object of PayResult class.

Trigger the Payment:

Call the SDK to trigger the payment process.

```
paySDK.process();
```

K) WeChat Pay Payment Call

Prepare Payment Call :

Initialize the PayData object and prepare the payment detail for the transaction.

```
PayData payData = new PayData();
payData.setChannel(EnvBase.PayChannel.DIRECT);
payData.setEnvType(EnvBase.EnvType.SANDBOX);
payData.setPayGate(EnvBase.PayGate.PAYDOLLAR);
payData.setCurrCode(EnvBase.Currency.RMB);
payData.setPayType(EnvBase.PayType.NORMAL_PAYMENT);
payData.setLang(EnvBase.Language.ENGLISH);
payData.setAmount("1");
payData.setPayMethod("WECHATAPP");
payData.setMerchantId("1");
payData.setOrderRef("2018102409220001");
payData.setRemark("payment remark");
payData.setActivity(MainActivity.this);
paySDK.setRequestData(payData);
```

Prepare Response Handler :

Initialize a payment event handler to capture the payment response and result.

```
paySDK.responseHandler(new PaymentResponse() {
    @Override
    public void getResponse(PayResult payResult) {
    }
    @Override
    public void onError(data: Data) {
    }
});
```

This method returns object of PayResult class.

Trigger the Payment:

Call the SDK to trigger the payment process.

```
paySDK.process();
```

L) 3DS 2.0 Payment Call

Prepare Payment Call:

Initialize the PayData class and prepare the payment detail for the transaction.

```
payData = new PayData();
payData.setChannel(EnvBase.PayChannel.DIRECT);
payData.setEnvType(EnvBase.EnvType.SANDBOX);
payData.setAmount("15");
payData.setPayGate(EnvBase.PayGate.PAYDOLLAR);
payData.setCurrCode(EnvBase.Currency.HKD);
payData.setPayType(EnvBase.PayType.NORMAL_PAYMENT);
payData.setOrderRef("2018102409220001");
payData.setPayMethod("3DS2.0");
payData.setLang(EnvBase.Language.ENGLISH);
payData.setMerchantId("1");
payData.setRemark(" ");

CardDetails cardDetails=new CardDetails();
cardDetails.setCardNo("4548890133258926");
cardDetails.setEpMonth("08");
cardDetails.setEpYear("2020");
cardDetails.setSecurityCode("123");
cardDetails.setCardHolder("test Card");
payData.setCardDetails(cardDetails);
ThreeDSParams threeDSParams = new ThreeDSParams();
threeDSParams.setUsername(“username”);
threeDSParams.setUsername(“password”);
.
.
.
.
.
.
.
payData.setThreeDSParams(threeDSParams);
```

```
paySDK.setRequestData(payData);
```

Prepare Response Handler :

Initialize a payment event handler to capture the payment response and result.

```
paySDK.responseHandler(new PaymentResponse() {
    @Override
    public void getResponse(PayResult payResult) {
     }

    @Override
    public void onError(data: Data) {
    }
});
```

Trigger the Payment:

Call the SDK to trigger the payment process.

```
paySDK.process();
```

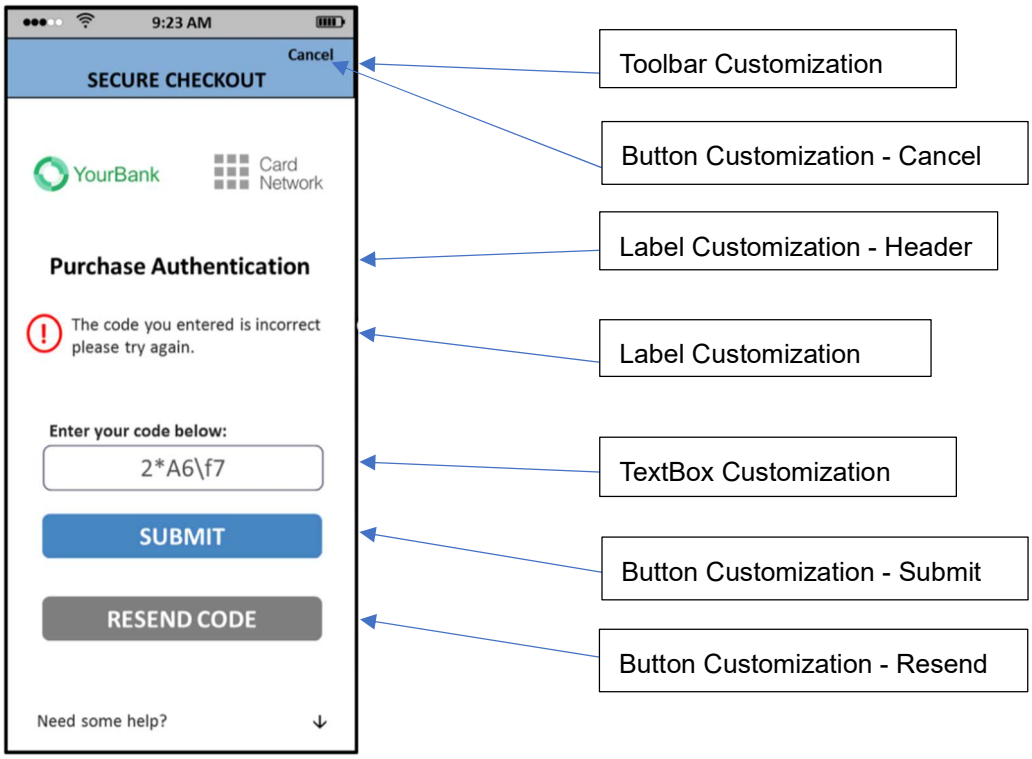
List of Parameters merchant can set	
apiUsername	threeDSAcctCreateDate
apiPassword	threeDSAcctAgeInd
threeDSCustomerEmail	threeDSAcctLastChangeDate
threeDSDeliveryEmail	threeDSAcctLastChangeInd
threeDSMobilePhoneCountryCode	threeDSAcctPwChangeDate
threeDSMobilePhoneNumber	threeDSAcctPwChangeInd
threeDSHomePhoneCountryCode	threeDSAcctPurchaseCount
threeDSHomePhoneNumber	threeDSAcctCardProvisionAttempt
threeDSWorkPhoneCountryCode	threeDSAcctNumTransDay
threeDSWorkPhoneNumber	threeDSAcctNumTransYear
threeDSBillingCountryCode	threeDSAcctPaymentAcctDate
threeDSBillingState	threeDSAcctPaymentAcctInd
threeDSBillingCity	threeDSAcctShippingAddrLastChangeDate
threeDSBillingLine1	threeDSAcctShippingAddrLastChangeInd
threeDSBillingLine2	threeDSAcctIsShippingAcctNameSame
threeDSBillingLine3	threeDSAcctIsSuspiciousAcct
threeDSBillingPostalCode	threeDSAcctAuthMethod
threeDSShippingDetails	threeDSAcctAuthTimestamp
threeDSShippingCountryCode	threeDSDeliveryTime

threeDSShippingState	threeDSPreOrderReason
threeDSShippingCity	threeDSPreOrderReadyDate
threeDSShippingLine1	threeDSGiftCardAmount
threeDSShippingLine2	threeDSGiftCardCurr
threeDSShippingLine3	threeDSGiftCardCount

Type of Challenge for Payment via 3DS 2.0

- Challenge Information TextBox
- Single and Multi-select Fields
- Out of Band
- HTML

UI - Customization for Challenge View for Merchant



UI-Customization for paySDK

Merchant can add UiCustomization at the paySDK initialization process .

```
private class initaliseSDK extends AsyncTask<String, Void, String> {
        @Override
        protected String doInBackground(String... params) {
```

```
            Factory factory = new com.asiapay.sdk.integration.xecure3ds.Factory();
            ConfigParameters configParameters = factory.newConfigParameters();
            UiCustomization uiCustomization = factory.newUiCustomization();
            paySDK = new PaySDK(AuthActivity.this, configParameters,
uiCustomization);
            return "";
        }
}
```

Class PayData properties details:

Input Parameters	Expected Value and Description
setChannel	Channel Type (EnvBase.PayChannel) – WebView, Direct
setPayGate	Name of payment gateway i.e. Paydollar or Pesopay or Siampay
setEnvType	Production or Sandbox
setMerchantID	ID received after registration on Asiapay merchant portal
setPayType	Payment Type (Normal / Hold)
setOrderRef	Reference number that merchant wants to attach to transaction
setPayMethod	Payment Method ID - please reference to Integration Guide for detail values
setLang	Language of Payment Page
setCurrency	Transaction currency, e.g. HKD, INR etc.
setAmount	Transaction amount
setAddNewMember	If merchant making payment for first time then value should be true else false
setActivity	Reference of the Activity class which is calling Alipay.
setRemark	More details about transaction to tag it. This is optional.
setExtraData	Extra Parameter For **Installment Pay** installment_service – Installment service indicator i.e T/F installment_period - In number of months installOnly - Control the mandatory of installment payment i.e. T / F For **Schedule Pay** For **Member Pay** memberPay_service – MemberPay service indicator I.e T/F memberPay_memberId – Member ID addNewMember – True / False For **Promo Pay** promotion – Promotion service indicator I.e T/F promotionCode – The promotion campaign code promotionRuleCode – The promotion rule code (Optional) promotionOriginalAmt – The promotion original amount (Optional) **For Schedule Pay** appId - Application Id appRef - appref is the reference for the whole schedule pay which will be used only one time

	schType - Schedule Type could be Day/Month/Year schStatus - Schedule Status could be Active/Suspend nSch - Number of schedule sMonth - Schedule start month sDay - Schedule start day sYear - Schedule start year eMonth - Schedule end month eDay - Schedule end day eYear - Schedule end year

Class EnvBase values details:

EnvBase Parameter	Value and Details
PayChannel	SDK payment channel WEBVIEW: WebView redirect payment DIRECT: Direct server payment
PayGate	SDK supported payment gateway PAYDOLLAR: PayDollar SIAMPAY: SiamPay PESOPAY: PesoPay
EnvType	SDK environment configuration PRODUCTION: Production environment SANDBOX: Sandbox testing environment
PayType	SDK payment type NORMAL_PAYMENT: Sales payment HOLD_PAYMENT: Authorize payment
Currency	SDK supported currency list HKD, USD, SGD, RMB, JPY, TWD, AUD, EUR, GBP, CAD, MOP, PHP, THB, MYR, IDR, KRW, BND, NZD, SAR, AED, BRL, INR, TRY, ZAR, VND, DKK, ILS, NOK, RUB, SEK, CHF, ARS, CLP, COP, CZK, EGP, HUF, KZT, LBP, MXN, NGN, PKR, PEN, PLN, QAR, RON, UAH, VEF, LKR, KWD

Class PayResult properties details:

PayResult Output Property Paramete	Description
getSuccesscode	Status of transaction.
getChannel	Channel Type (EnvBase.PayChannel) – WebView, Direct
getResultCode	Transaction status code.
getResultMsg	Message to describe more about transaction status. Message may come blank if the transaction status is Success.
getOrderID	Order ID generated by AsiaPay System
getOrderRef	Order Reference number set before initializing payment call in PayData object.
getBankRef	Bank Order Reference number
getPrc	Primary response code Refer to Integration Guide Appendix A (PRC)
getSrc	Secondary response code Refer to Integration Guide Appendix A (SRC)
getTxTime	Transaction Time
getAmount	Transaction amount
getCurrency	Transaction currency, e.g. HKD, INR etc.
getExtraData	Extra Response data
getOutTradeNo	Transaction number of Alipay for reference.

10 Frequently Asked Questions

System Setup

1. **What programming languages are supported in the Integration?**
   HTML, ASP, PHP, JSP / Servlet, and any other server side scripting languages that support HTTP protocol.

2. **Is there any consideration on firewall issues on Merchants side?**
   Merchants have to open HTTP port for data feed handling, i.e. port 80(HTTP) / port 443(HTTPS).

3. **Does PayDollar PayGate support any shopping cart software?**
   Technically yes. You are however required to know how to deploy the shopping cart software to work for your requirements. Samples include OSCommerce.

Common Problems

4. **During the integration I encounter the error message "Your payment service is not active."**
   Make sure you are using the corresponding pair of merchant ID and integration URL.
   If you are using the TESTING URL (test.paydollar.com), the TESTING merchant ID, a 6-digits number, should be used. If you are using the PRODUCTION URL (www.paydollar.com), PRODUCTION merchant ID should be used, which is 4-digits number or 8-digits number.

5. **During the integration I encounter the error message "CurrCode is incorrect."**
   One PayDollar merchant ID only allows one currency. Make sure you are using the corresponding currency for the merchant ID in the HTML form. To apply multi-currencies, please contact our salespersons to open additional merchant accounts.

6. **Can I make use of the calling of successUrl / failUrl / cancelUrl solely to determine the transaction status?**
   Customer may call the successUrl / failUrl / cancelUrl (with the merchant reference appended) in the browser and pretend the transaction is completed. Thus we recommend merchants to use data feed to determine the transaction status. Since PayDollar and the merchant are the only parties with the knowledge of the data feed URL, it is safe to determine the status by using the data feed.

7. **What is the difference between the parameters: Ref (orderRef), PayRef and Ord?**
   Ref (orderRef) is merchant's own order reference number. This comes from merchant's database or invoices.
   PayRef is PayDollar order reference number. It is unique among all orders from different merchants in PayDollar system.
   Ord is bank reference number. It is generated by acquiring banks.

To seek help on transactions, please provide PayRef (preferred) or Ref to us.

Data Feed

8. **How do I make use of the data feed?**

   You may make use of the data passed from PayDollar in the data feed to update your database records regarding the transaction. Since merchant reference can be retrieved from data feed, you can make use of this key to update the corresponding transaction records of your system(s).

9. **How to set up data feed in my merchant account?**

   You can enable or disable the data feed function in the merchant administration site > Profile > Payment Options.

10. **How do I know if the data feed is set up properly?**

    You may go to transaction details in the merchant administration site, select output columns "Data Feed Ind." and "Data Feed Return" and view the transaction records. If the data feed indicator is "T", the data feed is set up properly. However, if the indicator is "F", you may look at the "Data Feed Return" column and see what the error is. You may ask our technical team to assist when it relates to data feed setup.

    In addition, you can enable "Data Feed Failure Alert by Email" in the merchant administration site > Profile > Payment Options. When it is enabled, and if there is any failure on calling Merchant's data feed, an email will be sent to Merchant's technical contact email account.

11. **What programming languages can be used in writing data feed page?**

    Any programming languages that can handle HTTP Post request parameters can be used.

12. **How do I write the data feed page? Should the data feed page display anything on the screen?**

    The data feed page contains 3 parts, receive HTTP parameters, print 'OK' and your own backend processing (e.g. update database, send email to customer, etc.).

    The data feed page is called in the back end. It should not display anything, i.e. no HTML code. However it should respond with the word 'OK' to let our server know that you have successfully received the data feed.

13. **How do I know for sure the data feed is calling from PayDollar?**

    You may check if the data feed HTTP request is calling from these IP ranges.

    Testing: 203.105.61.183 – 203.105.61.218

    Production: 103.149.149.1 – 103.149.149.62

14. **There is data feed error return "javax.net.ssl.SSLHandshakeException: sun.security.validator.ValidatorException: No trusted certificate found"**

    Our server cannot recognize your SSL certificate provider. Please contact our I.T. team to resolve the issues. When necessary you will be asked to provide your CA root certificate.

15. **Auto retry failed datafeed**

(Applicable for merchants who have registered the datafeed link and retry function)

Sometimes, you may not be able to receive the datafeed response due to reasons like internet connection issue, incorrect datafeed URL being used, etc.

After enabling the "Auto retry failed data feed", datafeed will be resent:

(1) immediately after the original attempt is failed, and

(2) 15 minutes after if (1) is also failed

3-D Secure Authentication

16. **Can I use FRAMES in designing our website?**

Some card issuing banks require Cookies when entering their 3D authentication page. Full page (i.e. no FRAMES) is required in this situation.

Furthermore, the SSL indicator should be displayed somewhere on the browser window to let customers know that the site is secure enough to enter sensitive information such as credit card number. When FRAMES is used, the SSL indicator may not appear if the page required SSL is inside the frame window.

Therefore, it is suggested not to use FRAMES after the HTML form is submitted to PayDollar. You may consider opening another browser window to handle payment transactions.

17. **Can I ask for disabling 3-D Secure authentication for my transactions?**

This is subject to bank's sole decision. You may be asked for providing business information in order to help the bank risk management team to evaluate your company profile before special approval. Nonetheless, you should be well aware the serious consequences of fraudulent non-3D transactions.

Data Security

18. **Can I store the credit card information of my customers?**

We recommend our merchants NOT to store credit card information, especially credit card number, expiry date and CVV2 / CVC2. Even if absolute necessary, please ensure to encrypt the data compliant to Visa A.I.S. (Account Information Security) program. You may refer to the following webpage: http://www.visa-asia.com/ap/sea/merchants/riskmgmt/ais_what.shtml. Credit card information stored in PayDollar complies with Visa A.I.S. and MasterCard SDP programs.

Support

19. **Who should we contact in case we encounter problems during testing and in production?**

You can contact our I.T. team by either of the following ways.

a. Email: it@paydollar.com

b. Technical hotline: (852) – 82267981 (852) – 31731990

APPENDIX A

PayDollar Payment Response Code

PayDollar Payment Response Code is composed of the following items:

1. Primary Response Code (PRC)

The primary response code is the main response code used for identifying the authorization status of a payment transaction.

The following table provides a summary of all the response codes which may be returned:

PRC	Description
0	Success
1	Rejected by Payment Bank
3	Rejected due to Payer Authentication Failure (3D)
9	Rejected by Other Reason (Please check the Reject Reason at the transaction)
-1	Rejected due to Input Parameters Incorrect
-2	Rejected due to Server Access Error
-8	Rejected due to PayDollar Internal/Fraud Prevention Checking
-9	Rejected by Host Access Error

2. Secondary Response Code (SRC)

The secondary response code provides the detail description corresponding to the primary response code.

List of Response Code

Bank's Response Code

PRC	SRC	Description
1	01	Bank Decline
1	02	Bank Decline
1	03	Other
1	04	Other
1	05	Bank Decline
1	12	Other
1	13	Other
1	14	Input Error
1	19	Other
1	25	Other
1	30	Other
1	31	Other
1	41	Lost / Stolen Card
1	43	Lost / Stolen Card
1	51	Bank Decline
1	54	Input Error
1	55	Other
1	58	Other
1	76	Other
1	77	Other
1	78	Other
1	80	Other
1	89	Other
1	91	Other
1	94	Other
1	95	Other
1	96	Other
1	99	Other
1	2000	Other

Response Code From PayDollar

PRC	SRC	Description
-8	999	Other
-8	1000	Skipped transaction
-8	2000	Blacklist error
-8	2001	Blacklist card by system
-8	2002	Blacklist card by merchant
-8	2003	Black IP by system
-8	2004	Black IP by merchant
-8	2005	Invalid cardholder name
-8	2006	Same card used more than 6 times a day
-8	2007	Duplicate merchant reference no.
-8	2008	Empty merchant reference no.
-8	2011	Other
-8	2012	Card verification failed
-8	2013	Card already registered
-8	2014	High risk country
-8	2016	Same payer IP attempted more than pre-defined no. a day.
-8	2017	Invalid card number
-8	2018	Multi-card attempt
-8	2019	Issuing Bank not match
-8	2020	Single transaction limit exceeded
-8	2021	Daily transaction limit exceeded
-8	2022	Monthly transaction limit exceeded
-8	2023	Invalid channel type
-8	2099	Non testing card
-8	2031	System rejected(TN)
-8	2032	System rejected(TA)
-8	2033	System rejected(TR)
-8	2035	Black List Payer Card Country defined by Merchant
-8	2037	Black List Payer Country defined by Merchant
-8	2039	Special Country Restriction Requested by Merchant
-8	2041	Invalid period for installment
-8	2043	Unverified Member

-8	2045	Invalid Coupon Code
-8	2047	Cancel by user
-8	2049	Reject due to not complete EWallet process
-8	2050	Invalid card for installment
-8	2051	Invalid Expiry Date
-8	2053	Canceled transaction
-8	2054	Rejected by 3D Settings
-8	2055	Card Issuing and Payer IP country not match

Other Response Code

PRC	SRC	Description
0	0	Success
3	Any Number	Payer Authentication Fail
-1	-1	Input Parameter Error
-2	-2	Server Access Error
-9	-9	Host Access Error

List of Country Code

This list shows the country names and risk level of individual country code.

Country Code	Country Name	High risk
A2	Satellite Provider	
AD	Andorra	
AE	United Arab Emirates	
AF	Afghanistan	
AG	Antigua and Barbuda	
AI	Anguilla	
AL	Albania	
AM	Armenia	
AN	Netherlands Antilles	
AO	Angola	
AP	Asia/Pacific Region	
AQ	Antarctica	
AR	Argentina	
AS	American Samoa	
AT	Austria	
AU	Australia	
AW	Aruba	
AZ	Azerbaijan	
BA	Bosnia and Herzegovina	
BB	Barbados	
BD	Bangladesh	
BE	Belgium	
BF	Burkina Faso	

BG	Bulgaria	**
BH	Bahrain	
BI	Burundi	
BJ	Benin	
BM	Bermuda	
BN	Brunei Darussalam	
BO	Bolivia	
BR	Brazil	
BS	Bahamas	
BT	Bhutan	
BV	Bouvet Island	
BW	Botswana	
BY	Belarus	
BZ	Belize	
CA	Canada	
CD	Congo	
CF	Central African Republic	
CG	Congo	
CH	Switzerland	
CI	Cote D'Ivoire	
CK	Cook Islands	
CL	Chile	
CM	Cameroon	**
CN	China	
CO	Colombia	
CR	Costa Rica	
CU	Cuba	
CV	Cape Verde	
CY	Cyprus	
CZ	Czech Republic	
DE	Germany	
DJ	Djibouti	
DK	Denmark	
DM	Dominica	
DO	Dominican Republic	
DZ	Algeria	
EC	Ecuador	

EE	Estonia	
EG	Egypt	**
ER	Eritrea	
ES	Spain	
ET	Ethiopia	
EU	Europe	
FI	Finland	
FJ	Fiji	
FK	Falkland Islands (Malvinas)	
FM	Micronesia	
FO	Faroe Islands	
FR	France	
GA	Gabon	
GB	United Kingdom	
GD	Grenada	
GE	Georgia	
GF	French Guiana	
GH	Ghana	**
GI	Gibraltar	
GL	Greenland	
GM	Gambia	**
GN	Guinea	
GP	Guadeloupe	
GQ	Equatorial Guinea	
GR	Greece	
GT	Guatemala	
GU	Guam	
GW	Guinea-Bissau	
GY	Guyana	
HK	Hong Kong	
HM	Heard Island and McDonald Islands	
HN	Honduras	
HR	Croatia	
HT	Haiti	
HU	Hungary	
ID	Indonesia	**
IE	Ireland	

IL	Israel	**
IN	India	
IO	British Indian Ocean Territory	
IQ	Iraq	
IR	Iran	**
IS	Iceland	
IT	Italy	
JM	Jamaica	
JO	Jordan	
JP	Japan	
KE	Kenya	
KG	Kyrgyzstan	
KH	Cambodia	
KI	Kiribati	
KM	Comoros	
KN	Saint Kitts and Nevis	
KP	Korea	
KR	Korea	
KW	Kuwait	
KY	Cayman Islands	
KZ	Kazakstan	
LA	Lao People's Democratic Republic	
LB	Lebanon	
LC	Saint Lucia	
LI	Liechtenstein	
LK	Sri Lanka	
LR	Liberia	
LS	Lesotho	
LT	Lithuania	**
LU	Luxembourg	
LV	Latvia	
LY	Libyan Arab Jamahiriya	
MA	Morocco	**
MC	Monaco	
MD	Moldova	
MG	Madagascar	
MH	Marshall Islands	

MK	Macedonia	
ML	Mali	
MM	Myanmar	
MN	Mongolia	
MO	Macau	
MP	Northern Mariana Islands	
MQ	Martinique	
MR	Mauritania	
MS	Montserrat	
MT	Malta	
MU	Mauritius	
MV	Maldives	
MW	Malawi	
MX	Mexico	
MY	Malaysia	**
MZ	Mozambique	
NA	Namibia	
NC	New Caledonia	
NE	Niger	
NF	Norfolk Island	
NG	Nigeria	**
NI	Nicaragua	
NL	Netherlands	
NO	Norway	
NP	Nepal	
NR	Nauru	
NZ	New Zealand	
OM	Oman	
PA	Panama	
PE	Peru	
PF	French Polynesia	
PG	Papua New Guinea	
PH	Philippines	
PK	Pakistan	**
PL	Poland	
PR	Puerto Rico	
PS	Palestinian Territory	

PT	Portugal	
PW	Palau	
PY	Paraguay	
QA	Qatar	
RE	Reunion	
RO	Romania	**
RU	Russian Federation	**
RW	Rwanda	
SA	Saudi Arabia	
SB	Solomon Islands	
SC	Seychelles	
SD	Sudan	
SE	Sweden	
SG	Singapore	
SI	Slovenia	
SK	Slovakia	
SL	Sierra Leone	
SM	San Marino	
SN	Senegal	
SO	Somalia	
SR	Suriname	
ST	Sao Tome and Principe	
SV	El Salvador	
SY	Syrian Arab Republic	
SZ	Swaziland	
TC	Turks and Caicos Islands	
TD	Chad	
TF	French Southern Territories	
TG	Togo	
TH	Thailand	
TJ	Tajikistan	
TK	Tokelau	
TM	Turkmenistan	
TN	Tunisia	
TO	Tonga	
TR	Turkey	**
TT	Trinidad and Tobago	

TV	Tuvalu	
TW	Taiwan	
TZ	Tanzania	
UA	Ukraine	**
UG	Uganda	
UM	United States Minor Outlying Islands	
US	United States	
UY	Uruguay	
UZ	Uzbekistan	
VA	Holy See (Vatican City State)	
VC	Saint Vincent and the Grenadines	
VE	Venezuela	
VG	Virgin Islands	
VI	Virgin Islands	
VN	Vietnam	**
VU	Vanuatu	
WF	Wallis and Futuna	
WS	Samoa	
YE	Yemen	
YT	Mayotte	
YU	Yugoslavia	**
ZA	South Africa	
ZM	Zambia	
ZW	Zimbabwe	

- The End -