Web Services Integration Guide

V5.13

March 2009

COMMIDEA

1. Introduction

This document is or use when integrating to the Commidea Web Service solution – XML V4. Contained within are descriptions and examples of the record structures required, as well as a step-by-step guide to how the process works.

2. ICP XML V4 Overview

The latest version of the Web Services solution provides merchants with a more resilient design and faster service using Commidea's next generation ICP system architecture.

It also contains support for the following:

- Payer Authentication
  this module allows MasterCard and Visa payments to be verified by entering a password, should the card be enrolled in this service with the issuer.

- Token Gateway
  this functionality provides the ability to register a customer's payment details with the Token Gateway which will return a token as a reference. None of the sensitive card details therefore need to be stored by the merchant, and can be reused in future by providing the token ID.

- Ukash
  this module will allow customers to pay for items using either a Ukash Voucher or Ukash Account. Ukash account and vouchers enable people to pre-pay for items. The Voucher itself contains a 19 digit code which is entered when paying for goods online. Should there be any remaining amount from the voucher after the purchase; another code is generated for the remaining amount.

Each of the transaction types available are listed in sections throughout the manual. For each, the process will be explained, and then the message types themselves listed. This will provide an understanding of how each works, and then all the messaging information required to incorporate the functionality.

3. Integration

To enable merchants to integrate to their systems, Commidea has a fully functional test system in place for each version.

The process for new integrations is to develop to the test server and once the integrator is satisfied that the solution is fully functional, contact is made with the Implementations Department to arrange for integration testing. It is recommended that some testing is performed on the integration before booking a testing slot. To help with this there is a list of checks that will be performed included within the manual (please see Appendix A). Within this list there are tests performed on the ability to respond accordingly to declines and voice referrals – to help with this there are some default values which stimulate certain behaviour:

Value	Expected Outcome
.00	Accepted, 789DE
.02	Voice referred
.05	Declined
.08	Refund Offline

The correct address / CSC input to get a full match is: 10, ME156LH with CSC 000
Below are the different input combinations and the expected output:

CSC Value	CVCRESULT
<null>	0 – Not Provided
555	1 – Not Checked
000	2 – Matched
111	4 – Not Matched

Address Line 1 Value	AD1AVSRESULT
<null>	0 – Not Provided
55	1 – Not Checked
10	2 – Matched
11	4 – Not Matched

Post Code Value	PCAVSRESULT
<null>	0 – Not Provided
ME555LH or 555	1 – Not Checked
ME156LH or 156	2 – Matched
ME111LH or 111	4 – Not Matched

The test system is also configured to return a dummy authorisation code for every transaction; so do not be concerned by the fact that every transaction returns the same code. This will be '789DE'.

To obtain a test account, please contact the Implementations Team at implementations@commidea.com, specifying which system solution is being integrated to. They may then ask for more information before issuing a test account, dependant on which features of XML V4 are to be utilised.

3.1 Commidea Timeouts

Transaction Authorisation Database Timeout – 45 seconds
This is the period for which ICP will wait for a transaction result until returning an authorisation error as the transaction result.

Commidea Web Service Timeout – 60 seconds
This is the period after which ICP will timeout should it not be able to post the result back to the merchant.

Commidea Payer Authentication Database Timeout – 30 seconds
This is the period of time that ICP will wait until it receives a response back from the Payer Authentication application. It will return a Commidea timeout response in this instance.

3.2 Integration Testing

As aforementioned, vigorous testing is performed by Commidea before any solution can be used in a live environment. To help developers ensure the application or website is ready for this testing; below are a list of recommendations to adhere to:

- Confirmation messages should be sent in every scenario except:
  - After receiving a negative error code response
  - When processing pre authorisation transactions, as these are automatically confirmed
- A timeout period should be in place to ensure that, if no confirmation response is received after a predefined period of time, the confirmation message is resent. A new transaction should not be raised in this scenario, as this can result in duplicate orders. Should there be any issues with recurring responses not being returned please contact Implementations during testing, or the Helpdesk once the solution is being used in a live environment
- Build timeout periods into the solution to ensure that should there be any connection errors that these are captured and counteracted suitably
- Perform validation on fields locally before posting the record to the ICP server. For example, only allow numeric values to be entered into the relevant fields
- Perform card checks locally using LUHN validation (see Appendix D)
- When reporting a voice referral to the user, do not inform them that the transaction has been "declined", as this is not the case. Inform the users something similar to: "… your payment attempt was unsuccessful, please use an alternative card"
- Before having a Commidea Engineer perform Integration Testing, ensure the solution is as close to the final product which will be set live as possible. For example; all on screen messages displayed to the user will need to be checked, so the solution must be complete and in full working order before being tested

When the solution being built is a website, the following should be considered during development:

- Only include logos for card schemes that can be accepted by the site
- Disable use of the 'Back' button / ensure data from previous pages is cleared and therefore cannot be fraudulently retrieved by returning to the page
- Remove the ability for duplicate orders to be raised through the system by disabling the order button after the order has been submitted
- As mentioned in the general list of recommendations, integration of the website should be the last step of development before it is set live. In this case, it should be an exact replica of the live site, or as close a representation as possible
- Disabling the ability to copy and paste from within the web form for added security

For more information

For a table of tests to run through before releasing the solution to Commidea for testing, please see Appendix A.

3.3 Testing URLs

Please find below the test URLs required to gain access to the XML payment service:

https://testweb.commidea.com/commideagateway/commideagateway.asmx

3.4 Integration Methods

Before describing the records that are required it would make sense to discuss the available methods to invoke the Commidea Web Service. To make the solution as pliable as possible there are the following options:

a) SOAP

This is a standard for exchanging XML-based messages, and forms a foundation layer of the web services stack, which provides a basic messaging framework that more abstract layers can be built on.

To enable merchants to integrate using this method there are a set of XSDs available which can be obtained from implementations@commidea.com.

Alternatively the descriptions are available at the following URL:

https://testweb.commidea.com/commideagateway/commideagateway.asmx?op=ProcessMsg

b) Web Service Proxy

A Web Service Proxy can be created from within Microsoft Visual Studio .Net by adding a web reference to the URL of the web service, or by a tool called Web Service Description Language Tool (wsdl.exe). The proxy class that is generated from the WSDL that describes the web services has the same method signatures as the web service and hides the implementation details so that calling the web service is transparent. This can then be used to create a new instance of the web service object as though it is a local object instead of a remote one.

c) Web Service Discovery Language

To access the WSDL descriptions, please visit the following URLs:

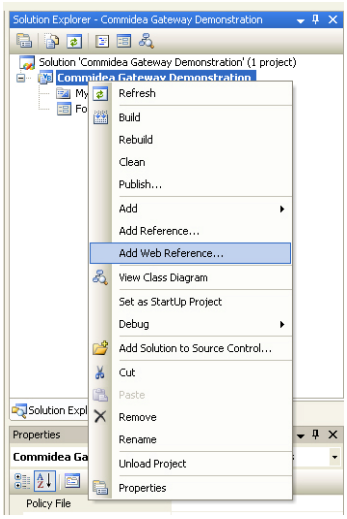
https://testweb.commidea.com/commideagateway/commideagateway.asmx?WSDL

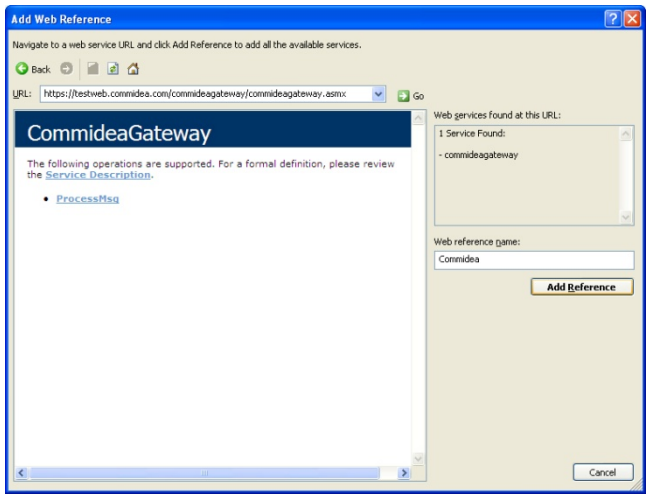
d) Web Referencing

XML V4 has been made simple to integrate into with the ability to add a web reference with Microsoft Visual Studio 2005. Following are the instructions for how to do this:

i. Right click in the project to add the reference to, and then select “Add Web Reference”:

ii. The address of the web service is then requested. Insert this into the “URL:” text box and press “Go”. The service should then be found, and “Add Reference” clicked to import it into the project.





Now that the reference has been added enabling consumption of the web service, via the use of a proxy class.

3.5 Live URLs

Once integration testing has been passed, the URLs being posted to by the solution will need to be updated. These will be supplied after integration testing has been completed and signed off.

The other change necessary would be to the merchant account specific information; the live account information to be used by the merchant will be required. This will entail updating the Merchant Header and Account ID.

4. Message Formats

All the XML data that is submitted to a Commidea Web Service must be formatted correctly; otherwise it will be rejected, and must be enclosed in the correct root element depending on the Web Service being called.

If passing data that contains any XML mark-up characters (e.g. ampersand '&' or less than / greater than symbols '< >') then it is recommended that the 'CDATAWrapping' flag within the Client Header is enabled. This informs the XML parser that it is not to be interpreted as mark-up. Here is an example, where using a reference of "Chip&Pin":

<[!CDATA[<txnrequest>…</txnrequest>]]>

Detailed below are the formats which all messages will be wrapped in.

4.1 Message

All requests and responses will be wrapped in a message type, as defined below:

Section/Fields	*Type/Format*	*Mandatory / Optional*	*Description*
<message>			
messagetype	String	M	Type of message
messagedata	String	M	Data
clientheader	ClientHeader	M	ClientHeader information
</message>			

4.2 ClientHeader

The clientheader is used to validate requests and direct them to the correct server:

Section/Fields	*Type/Format*	*Mandatory / Optional*	*Description*
<ClientHeader>			
SystemID	Decimal	M	Allocated ID
SystemGUID	String	M	Allocated GUID
Passcode	String	M	Allocated Passcode
ProcessingDB	String	M (for Confirmation Request and Rejection Request)	This indicates the database to use for processing a particular request. If left blank the default database will be used. N.B. Should only be left blank for the initial transaction request. (See 4.2.1)
SendAttempt	Integer	M	If greater than 0 this indicates that this is a resend attempt and duplicate checking should be performed. Max value of 5 before an automatic declined response is returned. Commidea will hold unconfirmed transactions up to 10 days based on acquirer authorisation expiries
CDATAWrapping	Boolean	O	If true then response messages will be CDATA wrapped. If false then they will not be wrapped. If this boolean is not passed then by default wrapping will be disabled. We highly recommend that this is enabled
</ClientHeader>			

4.2.1 Processing DB Field

To further explain the use of this field within the ClientHeader, this field does not need to be populated during the initial request, unless advised otherwise. However, when sending a Confirmation or Rejection Request this field must be populated with the same ProcessingDB as returned in the Transaction Response. This will ensure that the Confirmation or Rejection is sent to the same database which is awaiting the final decision on the transaction.

The Processing DB tag needs to be set for:

- Authentication Request (for Payer Authentication)
- Transaction Confirmation
- Transaction Rejection

Essentially, any transactions that receive a Processing DB value within the response need to include this same value within any subsequent requests.

4.3 Error Response

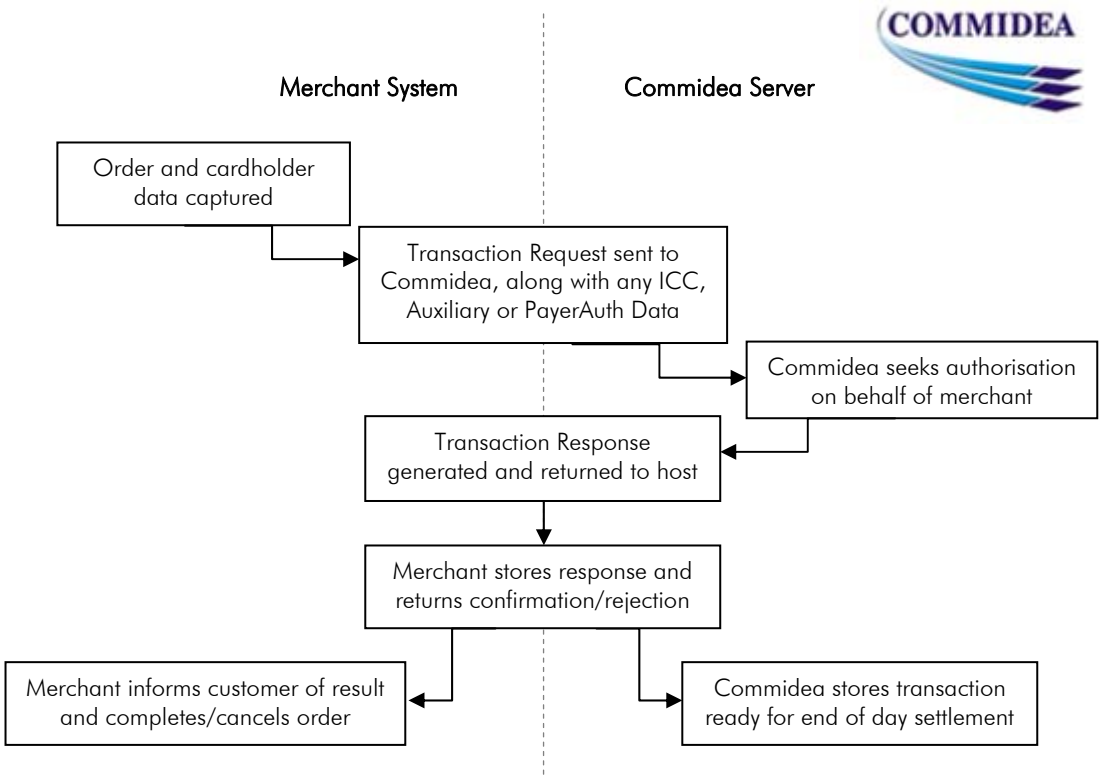
The error response will be returned in the event of a processing error:

Section/Fields	*Type/Format*	*Description*
<Error>		
Code	Integer	Code indicating error type
MsgTxt	String	Description of error
</Error>		

5. Transactions

5.1 Transaction Process

To process a transaction using XML V4 the following procedure is used:



5.2 Transaction Message Types

5.2.1 Transaction Request

The transaction request type contains all the required information to authorise the requested transaction type.

The Message Type for the transaction request is TXN and the namespace is TXN.

Section/Fields	*Type/Format*	*Mandatory / Optional*	*Description*
<transactionrequest>			
merchantreference	String	O	Merchant can add a reference to cross reference responses relating to the same transaction
accountid	Decimal	M	Account reference number, supplied by Commidea
txntype	String	M	01 – Purchase 02 – Refund 04 – Cash Advance 05 – Purchase with cash back (PWCB) 06 – Continuous Authority
transactioncurrencycode	String	M	This is the three digit currency code (numeric).
terminalcountrycode	String	M	In accordance with the numeric values defined in ISO 3166 (see Appendix C)
apacsterminalcapabilities	String	M	This is the functionality supported by the terminal in the format of that defined by the APACS standard. These are:

			3291 – Only Swiped and Contact ICC unattended 4290 – Mail Order/Telephone Order 4298 – CNP/ECommerce (if flagged for payer authorisation with acquirer; no CNP transactions are allowed with the exception of refunds) 6290 - Keyed and Swiped Customer Present 7296 – Contact (ICC) Keyed and Swiped B291 – Swiped, Contact ICC and Contactless unattended C296 – Contactless and keyed transactions (a contactless auxiliary record should be presented for all transactions passed under this terminal type) F296 – Keyed, Swiped, Contact and Contactless EMV transactions (a contactless auxiliary record should be present for all transactions passed under this terminal type) Integrators should check with Implementations to confirm that they have the correct capabilities.
capturemethod	Integer	M	This indicates how the card details were obtained. Acceptable values are: 1 – Keyed Cardholder Present 2 – Keyed Cardholder Not Present Mail Order 3 – Swiped 4 – ICC Fallback to Swipe 5 – ICC Fallback to Signature 6 – ICC PIN Only 7 – ICC PIN and Signature 8 – ICC – No CVM 9 – Contactless EMV 10 – Contactless Mag Stripe 11 – Keyed Cardholder Not Present Telephone Order 12 – Keyed Cardholder Not Present E-Commerce Order
processingidentifier	Integer	M	This indicates the type of processing that needs to be undertaken. Current available values are as follows: 1 – Auth and Charge 2 – Auth Only 3 – Charge Only All refund transactions should use the 'Charge Only' option.
tokenid	Decimal	O	Token Identifier for token transaction
pan	String	C	Card number (Conditionally required, not needed if providing a Token ID)
track2	String	C	Entire Track2 contents (including start and end sentinels and LRC) (Conditionally required, not needed if providing a Token ID)
csc	String	O	Amex Card – 3 or 4 digits (front of card) All Other Cards – 3 or 4 digits (rear security strip)
avshouse	String	O	Field checked by Address Verification System (AVS) add on module, ignored if module not enabled. AVS configuration can make this field mandatory. Numerics from house name\number
avspostcode	String	O	Field checked by Address Verification System (AVS) add on module, ignored if module not enabled. AVS configuration can make this field mandatory. Numerics from postcode

issuenumber	String	O	1 or 2 digit card issue number. Only required by some Switch, Solo and Laser cards, and only required when card is keyed
expirydate	String	C	Card expiry month and year (YYMM) (Only required when card is keyed, can be calculated from Track2) (Conditionally required, not needed if providing a Token ID)
startdate	String	O	Card start date month and year (MMYY) Only required for American Express, Diners Club International, some Switch, some Solo and some Laser cards. Not required if Track2 data supplied
Please note the format difference between the expiry and start dates are intentional			
txnvalue	Decimal	M	Total value of transaction including tax. Applies to: Purchase, Refund, Cheque Guarantee, Cash Advance, and Purchase with Cash Back. With PWCB, field should only contain the values of the goods or services provided. Decimal point recommended but optional, e.g.: 1.23 = £1.23 123 = £123 000001.23 = £1.23 Only positive values. Values will be truncated to the correct number of decimal places required for the transaction currency (set by the merchant account being used)
cashback	Decimal	O	Total Cash Back value for PWCB transactions. Values will be truncated (without rounding) to the number of decimal places required for the transaction currency. Positive values only.
gratuity	Decimal	O	Additional value to add to total (e.g. service tip)
authcode	String	O	Only supplied for Offline transactions
transactiondatetime	String	O	Date and time the transaction was started, based on GMT (dd/mm/yyyy hh:mm:ss).
iccdata	iccdata	O	Contains ICC data
vgisid	String	O	VGIS XML data (Reserved for future use)
employeeid	String	O	Field used to add information on the employee processing the transaction
payerauthauxiliarydata	String	O	Payer Authentication auxiliary data
</transactionrequest>			

5.2.2 ICC Data

When processing an ICC transaction, this message type is used to supply the extra information required.

Section/Fields	*Type/Format*	*Mandatory / Optional*	*Description*
<iccdata>			
emvterminalcapabilities	String	M	The terminal capabilities as defined in the EMV specifications.
emvterminaltype	String	M	Terminal type/Currency indicator S = Sterling E = Euro 0 = Unspecified terminal capabilities – S 1 = ICC reader only – S 2 = Magnetic stripe only – S 3 = ICC/Magnetic stripe – S 4 = No card reader – S 5 = Unspecified terminal capabilities – E

			6 = ICC reader only – E 7 = Magnetic stripe only – E 8 = ICC/Magnetic stripe – E 9 = No card reader – E
reasononlinecode	String	M	In the provisional European Standard (prENV 1750) the On-line reason codes are four digits in the form 15XX for PoS type of environment. As all PoS codes begin 15 there is no need to send this fixed value and therefore only the XX as defined in the ENV 1750 need be transmitted. Reason On-line will be used by the acquirer to determine if stand-in authorisation would be an appropriate action for this transaction. I.e. was it the ICC or the CAD which required an on-line authorisation.
arqc	String	M	Cryptogram generated by card at end of offline and online declined transactions. Can be used to validate the risk management activities for a given transaction (passed by ICC Terminal)
apppansequenceno	String	M	Identifies and differentiates cards with some PAN (ICC Card passes this information)
aip	String	M	Application Interchange Profile (passed by ICC terminal)
atc	String	M	Value of the last online transaction (passed by ICC terminal)
unpredictableno	String	M	(passed by ICC terminal)
tvr	String	M	Terminal Verification Results. Record of outcome of various application functions performed by Cardholder System (passed by ICC terminal)
cryptotxntype	String	M	Indicates transaction type used to application usage control. One of the following passed by ICC terminal: 00 – Purchase 09 – Purchase with Cash Back 20 – Refund
iad	String	M	Present if provided by ICC in GENERATE AC command response (passed by ICC terminal)
aid	String	M	Data label that identifies an application on card or terminal. E.g. AID for VSDC is 1010, Visa Electron is 2010, and Plus is 8010. Card and Terminals use AIDs to determine which applications are mutually supported; both card and terminal must support the same AID to initiate a transaction. Both cards and terminals may support multiple AIDs (passed by ICC terminal)
terminalapplicationversionnumber	String	M	A version number allocated by the payment scheme used to ensure compatibility between the IC and the terminal. (extracted from the IC Terminal Tag 9F 09)
cardapplicationversionnumber	String	M	Version number assigned by the payment system for the application on the IC card (extracted from the IC Card Tag 9F 08)
cvmr	String	M	Identifies a method of verification of the cardholder supported by the application

			e.g. Chip and Pin but in a numeric code (extracted from the IC Card)
cryptoinfodata	String	M	Please see EMVECO Application Specification Book 3 Page 16 for breakdown (passed by ICC terminal)
</iccdata>			

5.2.3 PayerAuth AuxiliaryData

After performing the PayerAuth process to check if the card has been enrolled and then authenticated; this message type is used to attach the PayerAuth results to the transaction.

Section/Fields	*Type/Format*	*Mandatory / Optional*	*Description*
<payerauthauxiliarydata>			
authenticationstatus	String	M	Indicates if the transaction authenticated or not: Y – Customer was successfully authenticated N – Customer failed authentication, and the transaction declined A – Attempts processing. APACS message will show verified enrollment but cardholder not participating U – Enrollment could not be completed, due to technical or other problem
authenticationcavv	String	M	Contains 28-byte Base-64 encoded Cardholder Authentication Verification Value (CAVV)
authenticationeci	String	M	2 digit Electronic Commerce Indicator (ECI) value
atsdata	String	M	Data to populate authorisation message
transactionid	String	M	TransactionID should be populated with the PayerAuthRequestID provided in the PayerAuth EnrollmentCheck Response
</payerauthauxiliarydata>			

5.2.4 Confirmation Request

This message type is used to confirm the transaction.

The Message Type for the confirmation request is CNF and the namespace is TXN.

Section/Fields	*Type/Format*	*Mandatory / Optional*	*Description*
<confirmationrequest>			
transactionid	Decimal	M	TransactionID from ProcessTransaction request
offlineauthcode	String	O	AuthCode if transaction was authorised offline
gratuity	Decimal	O	Additional value to add to total (e.g. service tip)
transactioncertificate	String	M for ICC	Transaction certificate from 2nd generate
arc	String	M for ICC	Auth response code
applicatonusagecontrol	String	M for ICC	Application usage control
tvr	String	M for ICC	Terminal verification results
cid	String	M for ICC	Cryptogram information data
tsi	String	M for ICC	Transaction status information
iad	String	M for ICC	Issuer Application Data
</confirmationrequest>			

5.2.5 Rejection Request

This message type is used to reject the transaction.

The Message Type for the transaction request is RJT and the namespace is TXN.

Section/Fields	*Type/Format*	*Mandatory / Optional*	*Description*
<rejectionrequest>			
transactionid	Decimal	M	TransactionID from ProcessTransaction request
tokenid	Decimal	O	Token identifier for token transaction
capturemethod	Integer	M	This indicates how the card details were obtained. Acceptable values are: Keyed Customer Present = 1 Keyed Customer Not Present Mail Order = 2 Swiped = 3 ICC Fallback to Swipe = 4 ICC Fallback to Signature = 5 ICC PIN Only = 6 ICC PIN and Signature = 7 ICC – No CVM = 8 Contactless EMV = 9 Contactless Mag Stripe = 10 Keyed Customer Not Present Telephone Order = 11 Keyed Customer Not Present E-Commerce = 12
pan	String	M	Card number (when card keyed)
track2	String	M	Entire Track2 contents (including start and end sentinels and LRC)
csc	String	O	Amex Card – 3 or 4 digits (front of card) All Other Cards – 3 or 4 digits (rear security strip)
avshouse	String	O	Field checked by Address Verification System (AVS) add on module, ignored if module not enabled. AVS configuration can make this field mandatory. Numerics from house name\number
avspostcode	String	O	Field checked by Address Verification System (AVS) add on module, ignored if module not enabled. AVS configuration can make this field mandatory. Numerics from postcode
</rejectionrequest>			

5.2.6 Transaction Response

This message type which will contain the response from the transaction.

The Message Type for the transaction response is TRM (for initial transactions result message) and FT (for the final transaction result message) and the namespace is TXN.

Section/Fields	*Type/Format*	*Description*
<transactionresponse>		
merchantreference	String	Merchant can add a reference to cross reference responses relating to the same transaction
transactionid	Decimal	Unique transaction ID
resultdatetimestring	String	Extended date and time string (YYYY-MM-DDTHH:MM:SS:ss)
processingdb	String	This indicates the database to use for processing a particular request. If left blank the default database will be used.
errormsg	String	Error message
merchantnumber	String	Unique merchant number
tid	String	Terminal ID
schemename	String	Card scheme name 1 – Amex 2 – Visa 3 – MasterCard 4 – Maestro 5 – Diners 6 – Delta 7 – JCB 8 – BT Test Host 9 – Time 10 – Solo 11 – Electron 21 – Visa CPC 23 – AllStar CPC 24 – EDC/Maestro 25 – Laser 26 – LTF 27 – CAF 28 – Creation 29 – Clydesdale 31 – BHS Gold 32 – Mothercare Card 33 – Burton Menswear 35 – BA AirPlus 36 – Amex CPC 999 – Invalid Card Range
messagenumber	String	Transaction message number (equivalent of EFTSN from previous versions of the Web Service)
authcode	String	Authorisation code return by bank. Blank if the transaction declined or if transaction value is below the floor limit
authmessage	String	Authorisation message e.g. 'RETAIN CARD'
vrtel	String	Telephone number to be called by the operator to seek manual authorisation. Only supplied for referred transactions
txnresult	String	Transaction result: ERROR REFERRAL COMMSDOWN DECLINED REJECTED CHARGED APPROVED AUTHORISED

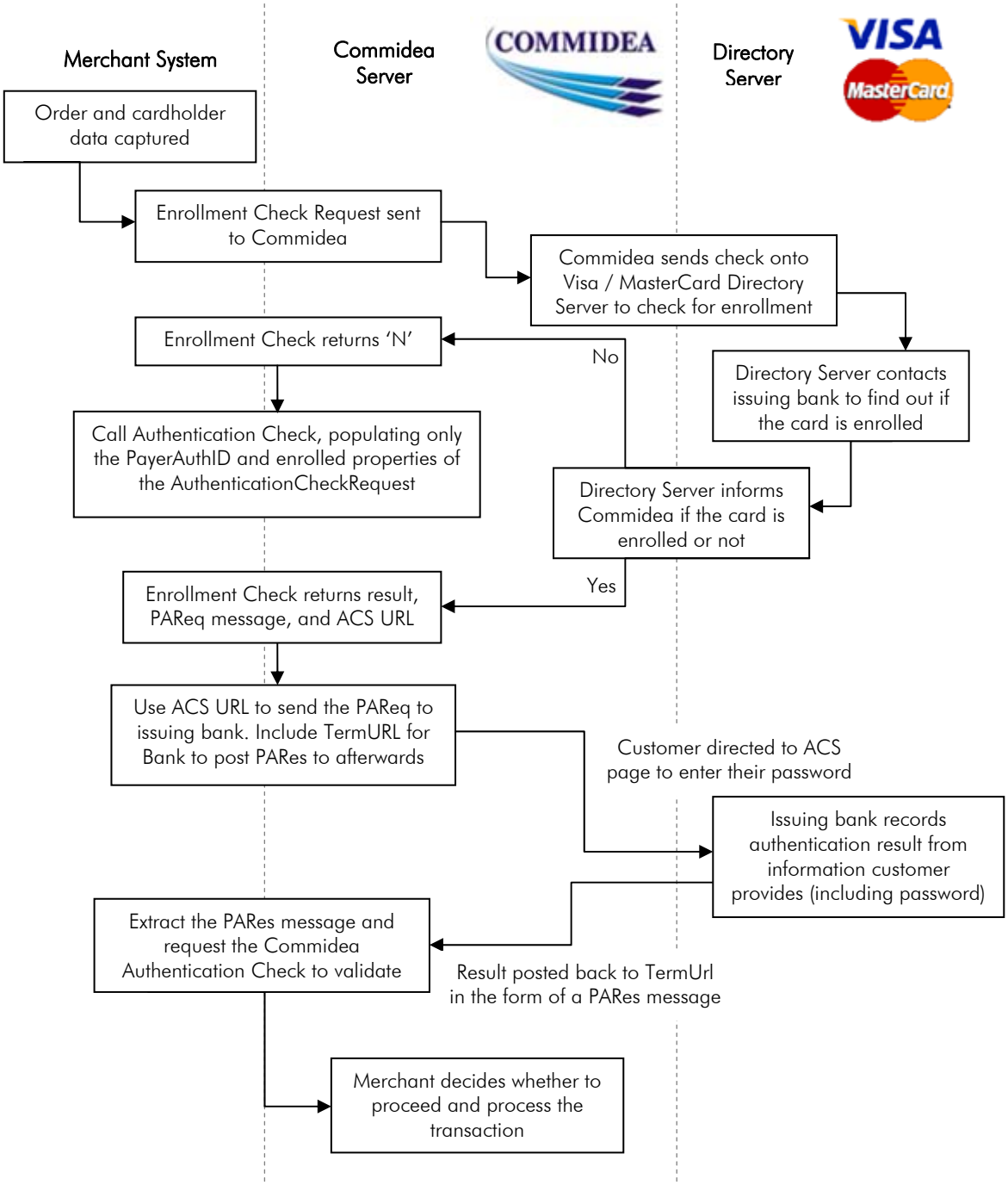
		AUTHONLY
pcavsresult	Integer	Postcode AVS result: 0 – Not provided* 1 – Not checked 2 – Matched 4 – Not matched Default result when no details are provided.
ad1avsresult	Integer	Address line 1 AVS result: 0 – Not provided* 1 – Not checked 2 – Matched 4 – Not matched Default result when no details are provided.
cvcresult	Integer	CVC result: 0 – Not provided* 1 – Not checked 2 – Matched 4 – Not matched Default result when no details are provided.
arc	String	Acquirer response code *Code Message Value - Status* 0 Authorised – Authorised 2 Referred – Refused 4 Hold Card – Refused 5 Refused – Refused 8 Approve After Identification – Refused 13 Invalid Amount – Refused 15 Invalid Card Issuer – Refused 17 Annulation By Client – Refused 28 Access Denied – Refused 29 Impossible Reference Number – Refused 33 Card Expired – Refused 34 Fraud Suspicion – Refused 38 Security Code Expired – Refused 41 Lost Card – Refused 43 Stolen Card, Pick Up – Refused 51 Limit Exceeded – Refused 55 Invalid Security Code – Refused 56 Unknown Card – Refused 57 Illegal Transaction – Refused 62 Restricted Card – Refused 63 Security Rules Violated – Refused 75 Security Code Invalid – Refused 76 Card Blocked – Refused 85 Rejected By Card Issuer – Refused 91 Creditcard Issuer Temporarily Not Reachable – Refused 97 Security Breach – Refused 3 –Invalid Acceptor – Error 12 Invalid Transaction – Error 14 Invalid Account – Error 19 Repeat Of Last Transaction – Error 20 Acquirer Error – Error 21 Reversal Not Processed, Missing Authorisation – Error 24 Update Of File Impossible – Error 25 Reference Number Cannot Be Found – Error 26 Duplicate Reference Number – Error 27 Error In Reference Number Field – Error 30 Format Error – Error 31 Unknown Acquirer Account Code – Error

		40 Requested Function Not Supported – Error 58 Transaction Not Permitted – Error 64 Amount Higher Than Previous Transaction Amount – Error 68 Transaction Timed Out – Error 80 Amount No Longer Available, Authorisation Expired – Error 92 Creditcard Type Not Processed By Acquirer – Error 94 Duplicate Request Error – Error
iadarc	String	Authorisation response cryptogram
iadoad	String	Optional additional data
isd	String	Issuer script data
authorisingentity	Integer	This indicates who actually performed the authorisation processing. Valid values are as follows: Not Provided = 0 Merchant = 1 Acquirer = 2 Card Scheme = 4 Issuer = 8
</transactionresponse>		

6. PayerAuth

6.1 PayerAuth Process

Payer Authentication checking, this adds support for Verified by Visa and MasterCard SecureCode without running additional software. These cardholder authentication services deter unauthorised card use. Additionally participating merchants receive added protection from fraudulent chargeback activity. Those who do not use these services may be liable for higher merchant fees; it is recommended to check this with the acquirer in question. Here is an overview of the entire process:



The process for this is as follows: as with standard transaction processing, the cardholder details and order information is captured, but this is then passed to the Commidea Enrollment Check. This discovers if the cardholder is enrolled by sending it onto the VISA or MasterCard Directory Server, this then contacts the issuer to check. If the cardholder is enrolled, they are redirected to the cardholder's web site by the host system (the URL is provided in the enrollment response) and they then enter their password. A string result is returned for validation, and used when calling the Commidea Authentication Check service to ensure this is valid. A response will be received; detailing the validity and the transaction can then be continued or aborted. This decision is up to the merchant and will take into account the outcome of the validity check.

To ensure that the process is clear, here are the steps required in their entirety:

i. The cardholder creates an order on the system, and clicks the 'Buy' button, which sends a post of the final buy page
ii. Create and send an Enrollment Check request, populating it with all the details from the webpage order
iii. This is sent onto the Directory Server, which contacts the issuing bank and finds out if the card is enrolled or not
iv. The Check Enrollment response is sent and if the card is enrolled contains:
    a. <Enrolled>Y</Enrolled>
    b. The PaReq message required to send to the issuing bank
    c. Access Control Server (ACS) URL

If the card is not enrolled, proceed to step x.

v. Send the PaReq message to the bank to request authentication. To do this, create a web page that only has hidden content, including a form that meets the following requirements:
    - The forms action is the ACS URL, which displays the issuing bank's dialog requesting the authentication password from the cardholder
    - The form includes the required hidden field PaReq, the value of which was returned to the merchant in the Enrollment Check response. **It is necessary to remove any White Space within this PaReq field otherwise this will cause errors when it is returned to the bank.**
    - The form includes the required field TermUrl, the value of which is the location where the merchant wants the bank to post the payment authentication response (PaRes) message.
    - The form must include the hidden field MD (merchant data); however, including a value in this field is optional. The value has no meaning to the bank, but is guaranteed to be returned without change. This allows the merchant to tag the redirect with a reference which will be returned during the redirect.
    - This page typically include JavaScript that automatically posts the form when the page loads (onload script)
vi. Open this page in the cardholder's web browser. Due to popup-blocking software, it is recommended opening this in the main browser window. The cardholder's web browser displays the issuing bank's authentication dialog, and enters their secret password for the credit card.
vii. The issuing bank records the result of the authentication dialog with the cardholder and sends it to the merchant, along with the transaction details, in a digitally signed PaRes message. The result is posted to the TermUrl on the web site, and the form posted by the issuing bank includes the PaRes.
viii. Extract the PaRes message from the form data and request the Commidea Authentication Check to validate the contents of the PaRes message.
ix. Depending upon the result of the Authentication Check; the merchant can now decide whether or not to proceed.
x. If the Enrollment Check indicated that the card was not enrolled, then call the Authentication Check populating only the PayerAuthRequestID and setting <Enrolled>N</Enrolled>within the AuthenticationCheckRequest.

If the card was enrolled and the merchant has now received the PaRes then populate PayerAuthRequestID, set the request to <Enrolled>Y</Enrolled> and include the PaRes message in the AuthenticationCheckRequest.

xi. Populate a Transaction Request with all the relevant details
xii. Invoke the Process Transaction method, passing the Transaction Request and wait for the Transaction Response to be returned

xiii. When the response is received, check the AuthResult to see if there was an error. If not then it is possible to complete the transaction with a Process Confirm; again populating the required information.

The only scenario in which a transaction should not be processed after performing Enrollment and Authentication checks would be when the following results are received:

<Enrolled>Y</Enrolled>
<AuthenticationStatus>N</AuthenticationStatus>

This represents the card being enrolled, but when the cardholder has attempted to authenticate using their password, this has not been matched correctly.

In the situation where these checks are unsuccessful, i.e. a response of <Enrolled>U</Enrolled> or <AuthenticationStatus>U</ AuthenticationStatus> is returned; it is recommended that the check is resent. Due to the fact that there has been a technical problem when checking Enrollment, charge back liability has not been shifted away from the merchant at this stage, as potentially the failure could have occurred before the information reached the Directory Server. However, the final decision on this is down to the merchant. If there is relatively low risk involved, due to a low transaction amount for example, the transaction could be continued and processed regardless.

When the card is not enrolled for Payer Authentication; the following responses will be amongst those produced:

<Enrolled>N</Enrolled>
<AuthenticationStatus>N</AuthenticationStatus>

It is important to remember that this does not mean it is not safe to proceed with the transaction; just that the cardholder has not been enrolled in the service. Due to an enrollment check being performed by the merchant, the liability is shifted to the issuer.

6.1.1 PayerAuth Expiry

One possible area which could create confusion is how long the Payer Authentication check lasts for once it has been approved, and if it can be reused.

Once the Payer Authentication check has been performed it is valid for 90days with VISA, and with MasterCard it does not expire.

One example would be that this allows use of the ID for an authorisation only transaction. If the authorisation code provided for the authorisation expires before charging the card; a full authorisation and charge transaction can be performed, using the PayerAuthRequestID that was provided initially.

6.1.2 Canadian Corporate Purchase Cards

Some Canadian Corporate Purchase Cards have been excluded from the Enrollment Check, and can result in a response of 'U' for the <Enrollment> field.

Unfortunately we are unable to confirm which bin ranges have been excluded and cannot therefore provide a specific response in this scenario.

6.1.3 Process Transaction

Once the enrollment and authentication checks have been performed, the transaction can be process by including a PayerAuth Auxiliary data record along with a Transaction Request record. Please see sections 5.2.1 and 5.2.3 for more information.

6.1.4 Payer Authentication with Token

When performing Payer Authentication in conjunction with an integration which utilises the Token Gateway, the process is to supply the TokenID with all the Payer Authentication checking records instead of supplying full card details.

Please note that each time stored card details are used to process a transaction, the Payer Authentication process must be completed.

6.1.5 Chargeback Information

Should chargeback information be required then this can be obtained from the Merchant Helpdesk.

6.2 Cardholder Authentication Implementation Guidelines

In order to provide some guidelines for how to go about implementing the cardholder authentication process, the following information has been collated from MasterCard and Visa.

1. Consumer Message on Payment Page
   In order to make the consumer aware of the merchant's participation with MasterCard SecureCode and Verified by Visa, it is recommended that a message is displayed on the payment page, similar to: "Your card may be eligible for or enrolled in MasterCard SecureCode or Verified by Visa. When you click 'Pay' below you may be prompted for further information before your order can be completed."

2. Creation of Cardholder Authentication Window
   The process with this window is that it is initially created by the merchant; however, that the actual content of the window is controlled by the cardholder's issuing financial institution. Initially it was possible to implement this using either a pop-up window or an inline window, but only the inline window implementation is now supported.

   Merchants utilising the pop-up window approach are expected to convert to an inline window implementation and inline window implementations are required for all new merchant implementations. By presenting a full-page view, it makes the SecureCode authentication process appear to be a seamless part of the merchant checkout process. Many merchants use frames to customise their deployments.

   In a frame implementation, only part of the full window is redirected to the issuer's access control server. This allows the merchant to display a branded header, as well as explanation text that can assist cardholders who are new to the cardholder authentication experience. Here are some key points for merchants implementing this approach:

   - The use of active HTML links in the branded header frame is not allowed. Below the header frame, however, it is recommended to include a link that directs the cardholder back to the checkout page in case of technical difficulties.
   - The explanation text should be clear and concise. The text should not assume that the cardholder is already enrolled and should not provide instructions that might conflict with the cardholder's issuer instructions.
   - The use of newer frame technologies such as iFrames and floating .Net frames is not recommended as some cardholders set their browsers to block such elements.
   - The merchant should make sure that the authentication window frame is fully visible and is not located too low in the page due to long text or large upper frame. A minimum space of 400x400 pixels is required for the Access Control Server (ACS) frame. It must not be necessary to scroll to see the authentication page.
   - Merchants must ensure that the 'back' button functionality works and cardholders who click on it are routed back to the checkout page.

   Inline authentication windows can also be used without frames. This will show the cardholder that they are no longer at the merchant and are now communicating with their issuing bank whilst also allowing them to check the SSL lock to ensure connection with the Issuer ACS. As a result, the 'Without frames' approach may be preferred by some cardholders.

3. TERMURL Field
   This field is provided by the merchant to the issuer during the payer authentication request process. It provides the issuer with the merchant URL where the payer authentication response message is to be sent. The use of mixed HTTP and HTTPS frames typically results in a security box being presented to the cardholder. Depending upon how the cardholder responds to this dialog, the current and all future attempts to transmit the PAReq message may fail. As a result,

merchants using inline authentication windows with frames must populate the TERMURL field with a HTTPS address.

6.3 PayerAuth Message Types

6.3.1 PayerAuth EnrollmentCheck Request

The EnrollmentCheck request is raised to check if the card is enrolled with MasterCard SecureCode or Verified By Visa.

The Message Type for the payer authentication enrollment check is PAI and the namespace is PAYERAUTH.

Section/Fields	*Type/Format*	*Mandatory/ Optional/Conditional*	*Description*
<payerauthenrollmentcheckrequest>			
merchantreference	String	O	Merchant can add a reference to cross reference responses relating to the same transaction
mkaccountid	Decimal	M	Account reference number, supplied by Commidea
mkacquirerid	Decimal	M	Acquirer reference number 1 – Barclaycard Business (BMS) [Sterling only] 2 – NatWest Streamline 3 – HMS (HSBC) 4 – Lloyds TSB Cardnet 5 – Elavon (GiroBank) 6 – Bank Of Scotland 7 – American Express 8 – Clydesdale Bank 9 – Barclaycard Business (BMS) MultiCurrency 10 – Bank of Ireland 11 – Northern Bank 12 – Yorkshire Bank 13 – GE Capital 14 – Ulster Bank 15 – Int'l Barclaycard Business (BMS) [Sterling] 16 – Int'l Lloyds TSB Cardnet 17 – Int'l HMS (HSBC) 18 – Int'l NatWest 19 – Int'l Barclaycard Business (BMS) Multi 20 – Diners 21 – Creation 23 – JCB 24 – AIB
merchantname	String Varchar(50)	M	The MerchantName must match the name shown online to the cardholder at the merchant's site and the name submitted by the merchant's acquirer in the settlement transaction
merchantcountrycode	String Varchar(50)	M	This field contains a three digit number assigned by

			the signing member or processor to identify the merchant's location country. Based on ISO Country Codes – 3166. (See Appendix C)
merchanturl	String Varchar(255)	M	This field contains the fully qualified URL of the merchant site
visamerchantbankid	String Varchar(50)	C (Only for Visa checks)	This field contains a six digit assigned Bank Identification Number issued by the merchant's member bank or processor. The acquirer Bank Identification Number (BIN) identifies the member bank that signed the merchant using the Point of Sale application
visamerchantnumber	String Varchar(50)	C (Only for Visa checks)	This field contains a unique ID number which is assigned by the signing merchant's acquirer, bank or processor. This field is used to identify the merchant within the VisaNet system
visamerchantpassword	String Varchar(50)	C (Only for Visa checks)	The alphanumeric merchant password is provided by the acquirer
mcmmerchantbankid	String Varchar(50)	C (Only for MasterCard /Maestro checks)	This field contains a six digit assigned Bank Identification Number issued by the merchant's member bank or processor. The acquirer Bank Identification Number (BIN) identifies the member bank that signed the merchant using the Point of Sale application
mcmmerchantnumber	String Varchar(50)	C (Only for MasterCard /Maestro checks)	This field contains a unique ID number which is assigned by the signing merchant's acquirer, bank or processor. This field is used to identify the merchant within the SecureCode system
mcmmerchantpassword	String Varchar(50)	C (Only for MasterCard /Maestro checks)	The alphanumeric merchant password is provided by the acquirer
tokenid	Decimal	C	Token identifier for token transaction. If none to be passed, '0' to be used.
cardnumber	String Varchar(50)	C	Card PAN
cardexpyear	String Char(4)	C	Card expiry date year YY e.g. 08 (not passed if token id supplied)

cardexpmonth	String Char(2)	C	Card expiry date month MM (not passed if token id supplied)
currencycode	String Char(3)	M	This field contains a three digit number assigned by the signing member or processor to identify the merchant's authorization currency. Based on ISO Country Code – 3166 (See Appendix C)
currencyexponent	String Char(1	M	No of decimal places in currency field ie. GBP will be 2
browseracceptheader	String Varchar(255)	O	This field contains the exact content of the HTTP accept header as sent to the merchant from the cardholder's user agent. This field is required only if the cardholder's user agent supplied a value.
browseruseragentheader	String Varchar(255)	O	This field contains the exact content of the HTTP user-agent header as sent to the merchant from the cardholder's user agent. This field is only required if the cardholder's user agent supplied a value.
transactionamount	String Varchar(50)	M	Amount to be authorised with implied decimal point ie. £10.00 is represented as 1000 and 0.10 is represented as 10.
transactiondisplayamount	String Varchar(50)	M	The transaction amount is to be presented with all currency-specific punctuation, as this will be the number displayed to the customer. E.g. 10.00
transactiondescription	String Varchar(50)	O	This field contains a description of the goods or services being purchased, determined by the merchant.
</payerauthenrollmentcheckrequest>			

6.3.2 PayerAuth EnrollmentCheck Response

The response from the check will be contained within the EnrollmentCheck response.

The Message Type for the payer authentication enrollment check response is PAER and the namespace is PAYERAUTH.

Section/Fields	*Type/Format*	*Description*
<payerauthenrollmentcheckresponse>		
merchantreference	String Varchar(50)	Merchant can add a reference to cross reference responses relating to the same transaction
processingdb	String	This indicates the database to use for processing a particular request. If left blank the default database will be used.
payerauthrequestid	Decimal	Unique Identifier
enrolled	String Char(1)	Indicates if card is enrolled in the 3D secure program.
acsurl	String Varchar(255)	Fully qualified URL of an Access Control Server.
pareq	String Varchar(1000)	This field will contain the entire XML response packet from the Directory Server.
errorcode	Integer	Error code defining the error
errordescription	String Varchar(1000)	Description of the error
</payerauthenrollmentcheckresponse>		

6.3.3 PayerAuth AuthenticationCheck Request

After the enrollment check has been performed, authentication can be sought using this message type.

The Message Type for the payer authentication check request is PAI and the namespace is PAYERAUTH.

Section/Fields	*Type/Format*	*Mandatory/ Optional/ Conditional*	*Description*
<payerauthauthenticationcheckrequest>			
merchantreference	String	O	Merchant can add a reference to cross reference responses relating to the same transaction
payerauthrequestid	Decimal	M	Unique Identifier returned during EnrollmentCheckResponse.
pares	String	C	Compressed and encoded Payer Authentication Response message, returned in response from Visa / Mastercard (Only included if received)
enrolled	String	M	Indicates if the card was enrolled – Y/N
</payerauthauthenticationcheckrequest>			

6.3.4 PayerAuth AuthenticationCheck Response

The authentication check response will contain the result of the authentication check.

The Message Type for the payer authentication response is PAAR and the namespace is PAYERAUTH.

Section/Fields	*Type/Format*	*Description*
<payerauthauthenticationcheckresponse>		
merchantreference	String	Merchant can add a reference to cross reference responses relating to the same transaction
payerauthrequestid	Decimal	PayerAuth transaction identifier
authenticationstatus	String	This property indicates whether the transaction has been authenticated or not. • Y – The customer was successfully authenticated. All data needed for clearing is included. • N – The customer failed authentication, and the transaction is denied. • A – Attempted processing. The APACS message will show verified enrolment but cardholder is not participating at this time. • U – Authentication could not be performed due to technical or other problems.
authenticationcertificate	String	The certificate that signed the Payer Authentication Response (PARes) message.
authenticationcavv	String	This property contains a 28-byte Base-64 encoded Cardholder Authentication Verification Value (CAVV).
authenticationeci	String	Two digit Electronic Commerce Indicator (ECI) value.
authenticationtime	String	The date and time in which the Payer Authentication Response (PARes) message was signed by the Access Control Server (ACS). The value is expressed in GMT and uses the format "YYYYMMDD HH:MM:SS".
atsdata	String	Additional transaction security data
errorcode	Integer	Error code defining the error
errordescription	String	Description of the error
processingdb	String	This indicates the database used for processing a particular request
</payerauthauthenticationcheckresponse>		

6.4 3D Secure Matrix

Below is a 3D Secure Matrix for both Visa VbV Transactions, and MasterCard SecureCode Transactions. This matrix provides a description of the Authentication Results and Merchant to Acquirer values presented during each scenario.

6.4.1 3D Secure Matrix – Visa VbV Transactions

Payer auth can be carried out on all Visa card ranges.

Scenario		Authentication results				Merchant to Acquirer		Liability Shift
		VERes (enrolment status)	PARes (Txn status)	PARes ECI value	PARes CAVV Present	ATSD	CAVV Present	
1.	**Confirmation** Participating Issuer, participating cardholder. Cardholder enrolment verified, authentication successful	Y	Y	05	Yes	D0C100	Yes	Yes
2.	**Attempt** Participating Issuer, participating cardholder. Cardholder enrolment verified. Visa have an Attempts ACS to act on behalf of non-participating Issuers / or non-enrolled cardholders.		A	06	Yes	D0C200	Yes	Yes
3.	**Denial** Participating Issuer, participating cardholder. Cardholder enrolment verified; Issuer declines authentication cardholder. I.e. cardholder unable to provide correct password.		N	None	No	No auth		N/A - Visa state that the transaction must not be sent for authorisation.
4.	**Authentication could not be performed** Participating Issuer, participating cardholder. Cardholder enrolment verified. Authentication could not be completed due to technical or other reasons.		U	None	No	D0C400	No	No
5.	**Non-participation** • Non-participating Issuer • Participating Issuer, card range of cardholder not registered in directory • Participating Issuer, cardholder not enrolled in ACS	N	None	None	N/A	D0C200	No	Yes

Scenario		Authentication results				Merchant to Acquirer		Liability Shift
		VERes (enrolment status)	PARes (Txn status)	PARes ECI value	PARes CAVV Present	ATSD	CAVV Present	
6.	**Unable to authenticate (2)** • Participating Issuer, ACS is not able to verify enrolment status of cardholder because card type or channel is not supported, or technical difficulties. • Non-participating Issuer, ineligible product	U	None	None	N/A	D0C400	No	Yes
7.	**Non Supporting Card Schemes**	N/A	N/A	N/A	N/A	808000	No	No

6.4.2 3D Secure Matrix – MasterCard SecureCode Transactions

Payer auth can be carried out on the following SecureCode schemes - MasterCard, Maestro International, Maestro Domestic and Solo.

Scenario		Authentication results				Merchant to Acquirer		Chargeback right
		VERes (enrolment status)	PARes (Transaction status)	PARes ECI value	PARes CAVV/ AAV Present	ATSD	CAVV/AAV Present	
1.	**Confirmation** Participating Issuer, participating cardholder. Cardholder enrolment verified, authentication successful	Y	Y	02	Yes	D09100	Yes	Yes
2.	**Attempt** Participating Issuer, participating cardholder. Cardholder enrolment verified. Visa have an Attempts ACS to act on behalf of non-participating Issuers / or non-enrolled cardholders. *MasterCard do not support attempts and the AAV must not be sent in the Auth.*		A	01	Yes	D09200	Yes	No
3.	**Denial** Participating Issuer, participating cardholder. Cardholder enrolment verified; Issuer declines authentication cardholder. I.e. cardholder unable to provide correct password.		N	None	No	No auth		Acquirer advises that the transaction is not sent for authorisation.
4.	**Authentication could not be performed** Participating Issuer, participating cardholder. Cardholder enrolment verified. Authentication could not be completed due to technical or other reasons.		U	None	No	D09400	No	Yes
5.	**Non-participation** • Non-participating Issuer • Participating Issuer, card range of cardholder not registered in directory • Participating Issuer, cardholder not enrolled in ACS.	N	None	None	N/A	D09200	No	Yes
6.	**Unable to authenticate (2)** • Participating Issuer, ACS is not able to verify enrolment status of cardholder because card type or channel is not supported, or technical difficulties. • Non-participating Issuer, ineligible product	U	None	None	N/A	D09400	No	Yes

VbV

EMV Terminal Type	30 (Visa ECI 5) =	Merchant & Cardholder are registered
	31 (Visa ECI 6) =	Merchant is registered, but Cardholder isn't.
	32 (Visa ECI 7) =	Standard E-Commerce message
APACS 70-2 Section B.4.2 (page 85)	Electronic Commerce Data Record Sub-type 01	
APACS 70-3 Section A.4 (page 93) Customer Instruction	G = Merchant & Cardholder registered H = Merchant is registered, but Cardholder isn't. J = Standard E-Commerce message	
Tests 5a & 5b	Please put a line through the scenario not being used	
Tests 6a & 6b	Please put a line through the scenario not being used	

Secure Code

EMV Terminal Type	30 (M'Card PDS 2) =	Merchant & Cardholder are registered
	31 (M'Card PDS 1) =	Merchant is registered, but Cardholder isn't.
	32 (M'Card PDS 0) =	Standard E-Commerce message
APACS 70-2 Section B.4.2 (page 85)	Electronic Commerce Data Record Sub-type 01	
APACS 70-3 Section A.4 (page 93) Customer Instruction	G = Merchant & Cardholder registered H = Merchant is registered, but Cardholder isn't. J = Standard E-Commerce message	
Tests 12a & 12b	Please put a line through the scenario not being used	
Tests 13a & 13b	Please put a line through the scenario not being used	

6.4.3 Non Supporting Card Schemes

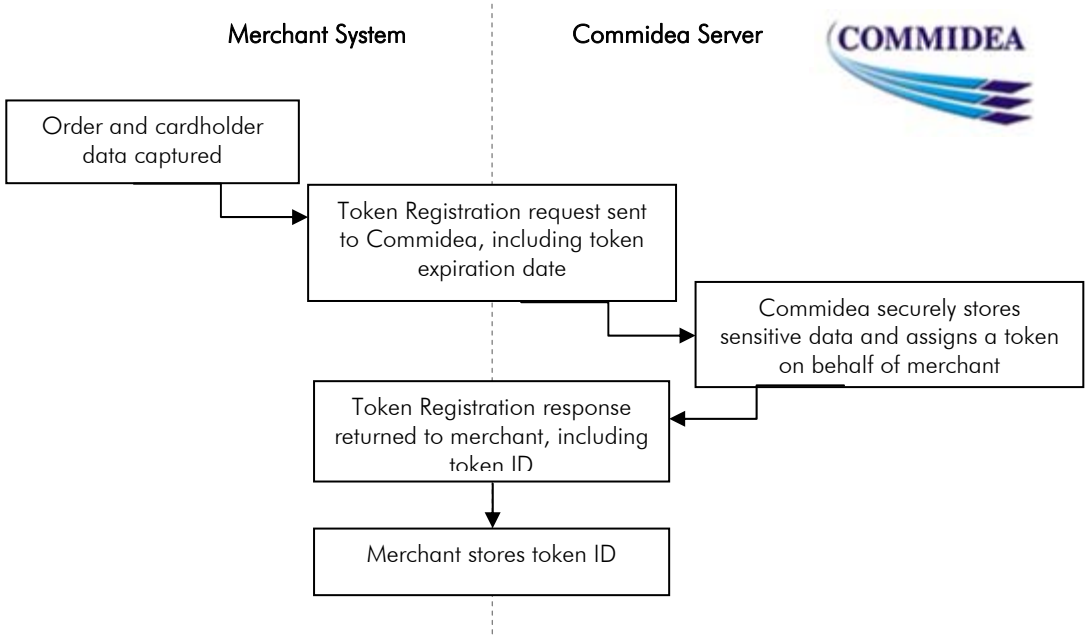
Payer Auth is not performed on non supporting schemes i.e.- Creation, AMEX, JCB, Diners. However, Additional Transaction Security Data is required to show that SSL encryption was used for the transaction.

SCHEME	VERES	PARES	ATSD	ECI
Non supporting schemes i.e.- Creation, AMEX, JCB, Diners	N/A	N/A	D08000	07

7. Token Gateway

7.1 Token Registration Process

To register a token a request should be created containing all the information required, which will include the card number, expiry date as well as boolean values to control the transaction types allowed for the token. A response will then be returned containing a TokenID. This should be stored, and can be provided in a transaction request (see section 5.2.1) to request for the stored details to be sourced and utilized to provide payment details for the transaction.



7.2 Token Message Types

7.2.1 Registration Request

The Token Registration Request type contains all the information required to register a token.

The Message Type is TKI and the namespace is TOKEN.

Section/Fields	*Type/Format*	*Mandatory/ Optional/ Conditional*	*Description*
<tokenregistrationrequest>			
merchantreference	String	O	Merchant can add a reference to cross reference responses relating to the same transaction
pan	String	M	Card number
expirydate	String	M	Card expiry month and year (YYMM) (Only required when card is keyed, can be calculated from Track2)
startdate	String	C	Card start date month and year (MMYY) Only required for American Express, Diners Club International, some Maestro, some Solo and

			some Laser cards. Not required if Track2 data supplied
Please note the format difference between the expiry and start dates are intentional			
issuenumber	String	C	1 or 2 digit card issue number as shown on the front of the card. Only required by some Switch, Solo and Laser cards. Required only when card is keyed
purchase	Boolean	M	Allow purchase txn type
refund	Boolean	M	Allow refund txn type
cashback	Boolean	M	Allow cashback txn type
tokenexpirationdate	String	M	Last date on which the token can be utilized. Format of date to be: DDMMCCYY
</tokenregistrationrequest>			

7.2.2 Token Registration Response

The Token Registration Response type contains all the result information from a token registration request.

Section/Fields	*Type/Format*	*Description*
<tokenregistrationresponse>		
merchantreference	String	Merchant can add a reference to cross reference responses relating to the same transaction
tokenid	Decimal	Unique identifier for registered PAN
errorcode	Integer	This is an error code indicating what type of error occurred, if any, while processing the transaction. See Appendix E for error codes
errormsg	String	This is a text field used to give as short text description of the error code
</tokenregistrationresponse>		

8. Ukash Message Types

The Message Type for Ukash requests is UKASH and the namespace is UKASH.

8.1 Ukash GetSettleAmount Request

Section/Fields	*Type/Format*	*Description*
<ukashrequest >		
merchantreference	String Varchar(50)	Merchant can add a reference to cross reference responses relating to the same transaction
requesttype	String Varchar(50)	The request type; ukashgetsettleamount
ukashlogin	String Char(20)	This is a login name that will be supplied to the merchant by Ukash to send with each transaction sent to the Ukash gateway
ukashpassword	String Char(20)	This is the password for the Ukash login name, which will be supplied to the merchant by Ukash
transactionid	String Char(20)	This is a unique reference to the transaction, which must be supplied by the merchant. It must be unique across the merchant's ukashLogin. E.g. for gaming clients, the format of the transactionId must be casinoId_TransNo
brandid	String Char(20)	Ukash will supply a brand id to the merchant for each of the brands he wishes to differentiate between. The appropriate brand id must then be sent through on each transaction request
vouchernumber	String Char(19)/ Char(16)	This is the number printed on the voucher or card, the number will be 19 digits for vouchers and 16 digits for cards.
vouchervalue	decimal	This is the value of the voucher presented in 2 decimal points
basecurr	String Char(3)	This is the currency in which the product/service is being sold. It is the merchant base currency for the transaction. It must be given in the character ISO standard. Refer to Appendix B to verify
vouchercurrproductcode	String Char(3)	7-9 digits of voucher number
</ukashrequest>		

8.2 Ukash PartSpendVoucher Request

Section/Fields	*Type/Format*	*Description*
<ukashrequest >		
merchantreference	String Varchar(50)	Merchant can add a reference to cross reference responses relating to the same transaction
requesttype	String Varchar(50)	The request type; ukashpartspendvoucher
ukashlogin	String Char(20)	This is a login name that will be supplied to the merchant by Ukash to send with each transaction sent to the Ukash gateway
ukashpassword	String Char(20)	This is the password for the Ukash login name, which will be supplied to the merchant by Ukash
transactionid	String Char(20)	This is a unique reference to the transaction, which must be supplied by the merchant. It must be unique across the merchant's ukashLogin. E.g. for gaming clients, the format of the transactionId must be casinoId_TransNo
brandid	String Char(20)	Ukash will supply a brand id to the merchant for each of the brands he wishes to differentiate between. The appropriate brand id must then be sent through on each transaction request
vouchernumber	String Char(19)/ Char(16)	This is the number printed on the voucher or card, the number will be 19 digits for vouchers and 16 digits for cards.
vouchervalue	decimal	This is the value of the voucher presented in 2 decimal points
ticketvalue	Decimal	This is the value, which the merchant wishes to charge from the voucher or account. It is presented in 2 decimal points in the merchant base currency.
basecurr	String Char(3)	This is the currency in which the product/service is being sold. It is the merchant base currency for the transaction. It must be given in the character ISO standard. Refer to Appendix B to verify
merchantdatetime	String Char(19)	This is the Merchant's time stamp of the transaction. Format "yyyy-mm-dd hh:mm:ss"
redemptiontype	String Char(1)/Char(2)	This indicates what the transaction is being used for. See Section 7.9 for redemption Types. The numeric identifier must be supplied
vouchercurrproductcode	String Char(3)	7-9 digits of voucher number
</ukashrequest>		

8.3 Ukash FullValueVoucher Request

Section/Fields	*Type/Format*	*Description*
<ukashrequest>		
merchantreference	String Varchar(50)	Merchant can add a reference to cross reference responses relating to the same transaction
requesttype	String Varchar(50)	The request type; ukashfullvaluevoucher
ukashlogin	String Char(20)	This is a login name that will be supplied to the merchant by Ukash to send with each transaction sent to the Ukash gateway
ukashpassword	String Char(20)	This is the password for the Ukash login name, which will be supplied to the merchant by Ukash
transactionid	String Char(20)	This is a unique reference to the transaction, which must be supplied by the merchant. It must be unique across the merchant's ukashLogin. E.g. for gaming clients, the format of the transactionId must be casinoId_TransNo
brandid	String Char(20)	Ukash will supply a brand id to the merchant for each of the brands he wishes to differentiate between. The appropriate brand id must then be sent through on each transaction request
vouchernumber	String Char(19)/ Char(16)	This is the number printed on the voucher or card, the number will be 19 digits for vouchers and 16 digits for cards.
vouchervalue	decimal	This is the value of the voucher presented in 2 decimal points
basecurr	String Char(3)	This is the currency in which the product/service is being sold. It is the merchant base currency for the transaction. It must be given in the character ISO standard. Refer to Appendix B to verify
merchantdatetime	String Char(19)	This is the Merchant's time stamp of the transaction. Format "yyyy-mm-dd hh:mm:ss"
redemptiontype		This indicates what the transaction is being used for. See Section 7.9 for redemption Types. The numeric identifier must be supplied
vouchercurrproductcode	String Char(3)	7-9 digits of voucher number
</ukashrequest>		

8.4 Ukash PartSpendAccount Request

Section/Fields	*Type/Format*	*Description*
<ukashrequest>		
merchantreference	String Varchar(50)	Merchant can add a reference to cross reference responses relating to the same transaction
requesttype	String Varchar(50)	The request type; ukashpartspendaccount
ukashlogin	String Char(20)	This is a login name that will be supplied to the merchant by Ukash to send with each transaction sent to the Ukash gateway
ukashpassword	String Char(20)	This is the password for the Ukash login name, which will be supplied to the merchant by Ukash
transactionid	String Char(20)	This is a unique reference to the transaction, which must be supplied by the merchant. It must be unique across the merchant's ukashLogin. E.g. for gaming clients, the format of the transactionId must be casinoId_TransNo
brandid	String Char(20)	Ukash will supply a brand id to the merchant for each of the brands he wishes to differentiate between. The appropriate brand id must then be sent through on each transaction request
vouchernumber	String Char(19)/ Char(16)	This is the number printed on the voucher or card, the number will be 19 digits for vouchers and 16 digits for cards.
ukashpin	String Char(4)	This is the Pin number printed on the Ukash Card. 4 digit value, field is required for all card based transactions
vouchervalue	decimal	This is the value of the voucher presented in 2 decimal points
ticketvalue	Decimal	This is the value, which the merchant wishes to charge from the voucher or account. It is presented in 2 decimal points in the merchant base currency.
basecurr	String Char(3)	This is the currency in which the product/service is being sold. It is the merchant base currency for the transaction. It must be given in the character ISO standard. Refer to Appendix B to verify
merchantdatetime	String Char(19)	This is the Merchant's time stamp of the transaction. Format "yyyy-mm-dd hh:mm:ss"
redemptiontype		This indicates what the transaction is being used for. See Section 7.9 for redemption Types. The numeric identifier must be supplied
vouchercurrproductcode	String Char(3)	7-9 digits of voucher number
</ukashrequest>		

8.5 Ukash FullSpendAccount Request

Section/Fields	*Type/Format*	*Description*
<ukashrequest>		
merchantreference	String Varchar(50)	Merchant can add a reference to cross reference responses relating to the same transaction
requesttype	String Varchar(50)	The request type; ukashfullspendaccount
ukashlogin	String Char(20)	This is a login name that will be supplied to the merchant by Ukash to send with each transaction sent to the Ukash gateway
ukashpassword	String Char(20)	This is the password for the Ukash login name, which will be supplied to the merchant by Ukash
transactionid	String Char(20)	This is a unique reference to the transaction, which must be supplied by the merchant. It must be unique across the merchant's ukashLogin. E.g. for gaming clients, the format of the transactionId must be casinoId_TransNo
brandid	String Char(20)	Ukash will supply a brand id to the merchant for each of the brands he wishes to differentiate between. The appropriate brand id must then be sent through on each transaction request
vouchernumber	String Char(19)/ Char(16)	This is the number printed on the voucher or card, the number will be 19 digits for vouchers and 16 digits for cards.
ukashpin	String Char(4)	This is the Pin number printed on the Ukash Card. 4 digit value, field is required for all card based transactions
vouchervalue	decimal	This is the value of the voucher presented in 2 decimal points
basecurr	String Char(3)	This is the currency in which the product/service is being sold. It is the merchant base currency for the transaction. It must be given in the character ISO standard. Refer to Appendix B to verify
merchantdatetime	String Char(19)	This is the Merchant's time stamp of the transaction. Format "yyyy-mm-dd hh:mm:ss"
redemptiontype		This indicates what the transaction is being used for. See Section 7.9 for redemption Types. The numeric identifier must be supplied
vouchercurrproductcode	String Char(3)	7-9 digits of voucher number
</ukashrequest>		

8.6 TransactionEnquiry Request

Used to check if there is an issue with the Ukash server, and there is a requirement to check with Ukash if the transaction was successful or not.

Section/Fields	*Type/Format*	*Description*
<ukashrequest>		
merchantreference	String Varchar(50)	Merchant can add a reference to cross reference responses relating to the same transaction
requesttype	String Varchar(50)	The request type; ukashtransactionenquiry
ukashlogin	String Char(20)	This is a login name that will be supplied to the merchant by Ukash to send with each transaction sent to the Ukash gateway
ukashpassword	String Char(20)	This is the password for the Ukash login name, which will be supplied to the merchant by Ukash
transactionid	String Char(20)	This is a unique reference to the transaction, which must be supplied by the merchant. It must be unique across the merchant's ukashLogin. E.g. for gaming clients, the format of the transactionId must be casinoId_TransNo
brandid	String Char(20)	Ukash will supply a brand id to the merchant for each of the brands he wishes to differentiate between. The appropriate brand id must then be sent through on each transaction request
vouchernumber	String Char(19)/ Char(16)	This is the number printed on the voucher or card, the number will be 19 digits for vouchers and 16 digits for cards.
ukashpin	String Char(4)	This is the Pin number printed on the Ukash Card. 4 digit value, field is required for all card based transactions
vouchervalue	decimal	This is the value of the voucher presented in 2 decimal points
basecurr	String Char(3)	This is the currency in which the product/service is being sold. It is the merchant base currency for the transaction. It must be given in the character ISO standard. Refer to Appendix B to verify
merchantdatetime	String Char(19)	This is the Merchant's time stamp of the transaction. Format "yyyy-mm-dd hh:mm:ss"
redemptiontype		This indicates what the transaction is being used for. See Section 7.9 for redemption Types. The numeric identifier must be supplied
vouchercurrproductcode	String Char(3)	7-9 digits of voucher number
</ukashrequest>		

8.7 Ukash Response

The Message Type for the Ukash response is UKASH and the namespace is TRM.

Section/Fields	*Type/Format*	*Description*
<ukashresponse>		
requesttype	String Varchar(50)	The request type
amountreference	String Char(255)	Merchants using the GetSettleAmount method need only fill in this tag. All other merchants should send a blank string in this tag.
mktransactionID	Decimal	Unique transaction ID
txcode	Integer	This is a transaction status/return code. It determines whether the voucher was successfully redeemed or not. A "0" means that the voucher was successfully redeemed. Any other code will reflect an unsuccessful redemption due to an invalid voucher or an error. See Section 7.8 for possible return codes
txdescription	String Char(255)	This is a text field used to give a short text description of the transaction status/return code
transactionid	String Char(20)	The transactionId is returned as reference to link the request and response XML
settleamount	Decimal	This is the value of the transaction in the base currency
accountbalance	Decimal	The account balance in the currency of the account. Applicable to account based transactions only.
accountcurrency	String Char(3)	This is the currency the card account. It will be given in the character ISO standard.
changeissuevouchernumber	String Char(19)	For ticket price redemption, this is the voucher number for the change. For full value redemption, this will be blank
changeissuevouchercurr	String Char(3)	This is the currency of the change issue voucher. It will be given in the character ISO standard. Refer to Appendix B
changeissueamount	Decimal	This is the value of the change presented in 2 decimal places. For full value redemption, this will be blank
changeissueexpirydate	String Char(10)	This is the expiry date for the change issue voucher in the format yyyy-mm-dd
issuedvouchernumber	String Char(19)	For issued vouchers this is the new voucher number. This tag will only be returned for IssueVoucher transactions.
issuedvouchercurr	String Char(3)	The currency of the issued voucher. It will be given in the character ISO standard. Refer to Appendix B. This tag will only be returned for IssueVoucher transactions.
issuedamount	Decimal	This is the value of the issued voucher presented in 2 decimal places. This tag will only be returned for IssueVoucher transactions.
issuedexpirydate	String Char(10)	This is the expiry date for the issued voucher in the format yyyy-mm-dd. This tag will only be returned for IssueVoucher transactions.
ukashtransactionid	String Char(50)	This is a unique reference to the transaction
currencyconversion	Boolean	This flag indicates whether currency conversion took place. For full value redemption, currency conversion may occur to determine the settleAmount in the base currency. For ticket price redemption,

		currency conversion may occur to determine the ticket price in the currency of the voucher
errcode	Integer	This is an error code indicating what type of error occurred, if any, while processing the transaction. See Section 7.10 for possible error codes
errdescription	String Char(255)	This is a text field used to give a short text description of the error code
</ukashresponse>		

8.8 Ukash Return Code List

Type of Message	Message Code	Message Description	Comments
Transaction Status	0	Accepted	Redemption successful
	1	Declined	Redemption unsuccessful
	99	Failed	An error occurred during the processing of the transaction hence the system could not successfully complete the redemption of the voucher.

8.9 Ukash Transaction Type List

Code	Description	Comments
1	Cash Withdrawal	Normal Redemption transactions. Voucher or account will be debited with the currency and amount.
2	Account Deposit	
3	Product/Service Purchase	
4	Issue Voucher	Issues Voucher based on the currency and value
8	Void Transaction	Voids a Transaction made in the last 60 seconds.
20	Account Add	Add amount to Ukash account. Used only for Top up and Top Down Cards.
21	Account Subtract	Subtracts amount from Ukash account. Used only for Top up and Top Down Cards.
22	Transaction Enquiry	Returns the state of a transaction that was executed in the last 48 hours.

8.10 Ukash Error Code List

Type of Error	Error Code	Error Description
Incoming XML Error	100	Invalid incoming XML
Data Validation Error	200	Non numeric Voucher Value
	201	Base Currency not 3 characters in length
	202	Non numeric Ticket Value
	203	Invalid BrandId
	204	Invalid MerchDateTime
	205	Invalid transactionId: greater than 20 characters
	206	Invalid Redemption Type
	207	Negative Ticket Value not allowed
	208	No decimal place given in Ticket Value
	209	No decimal place given in Voucher Value
	210	Negative Voucher Value not allowed
	211	Invalid or unsupported voucher product code
	212	AmountReference with TicketValue not allowed
	213	No ukashNumber supplied
	214	No transactionId supplied
	215	No brandId supplied
	216	Ticket Value cannot be greater than Voucher Value without Currency Conversion
	217	Base Currency and Voucher currency do not match.
	218	Brand not configured to Issue Vouchers
	219	Invalid Voucher Number
	221	Multiple Transactions found
	222	Unknown transaction status
	223	No transaction found.
Card Validation Error	250	The transaction cannot proceed with a user supplied PIN, and none was supplied,
	251	The supplied PIN had the incorrect format, e.g. was not 4 numeric characters
	252	PIN supplied with a transaction is incorrect (I.e. does not match the required pin recorded on file)
	253	PIN supplied with a transaction is incorrect and has resulted in the failure count reaching the maximum
	254	The Account has been blocked as a result of a validation/verification check failure
Login and Password	300	Invalid Login and/or Password
Login, Password and BrandID	301	Invalid Login and/or BrandID
Currency Conversion Not Supported	400	Required Currency Conversion not supported
Currency Conversion Error	500	Error In Currency Conversion
	501	Converted Settle Amount greater than Voucher Value
General Error	800	Max duration between getSettleAmount and Redemption exceeded.
	801	Invalid amountReference Submitted
Technical Error	900	Technical Error. Please contact Ukash Merchant Support
	999	Ukash Server Error. Please contact Commidea Merchant Helpdesk

8.11 Ukash Product Codes

Region	Country	State	Currency	General Issues (020-200)	Cash Back Issues (201-400)	Gambling Restricted (401-600)	Reserved (601-800)	Reserved (801-999)
United Kingdom	United Kingdom	United Kingdom	GBP	001	201	401	601	801
Europe	Europe	Europe	EUR	011	211	411	611	811
Europe	Poland	Poland	PLN	151	351	551	751	951
Europe	Austria	Austria	EUR	021	221	421	621	821
Europe	Belgium	Belgium	EUR	022	222	422	622	822
Europe	Finland	Finland	EUR	023	223	423	623	823
Europe	France	France	EUR	024	224	424	624	824
Europe	Germany	Germany	EUR	025	225	425	625	825
Europe	Greece	Greece	EUR	026	226	426	626	826
Europe	Ireland	Ireland	EUR	027	227	427	627	827
Europe	Italy	Italy	EUR	028	228	428	628	828
Europe	Luxembourg	Luxembourg	EUR	029	229	429	629	829
Europe	Netherlands	Netherlands	EUR	030	230	430	630	830
Europe	Portugal	Portugal	EUR	031	231	431	631	831
Europe	Spain	Spain	EUR	011	211	411	611	811
Europe	Switzerland	Switzerland	CHF	033	233	433	633	833
Europe	Denmark	Denmark	DKK	034	234	434	634	834
Europe	Sweden	Sweden	SEK	035	235	435	635	835
Europe	Czech Republic	Czech Republic	CZK	036	236	436	636	836
Europe	Norway	Norway	NOK	037	237	437	637	837
Europe	Romania	Romania	RON	038	238	438	638	838
Europe	Hungary	Hungary	HUF	039	239	439	639	839
Europe	Bulgaria	Bulgaria	BGL	040	240	440	640	840
Europe	Estonia	Estonia	EEK	041	241	441	641	841
North America	USA	USA	USD	99	299	499	699	899
North America	USA	Alabama	USD	100	300	500	700	900
North America	USA	Alaska	USD	101	301	501	701	901
North America	USA	Arizona	USD	102	302	502	702	902
North America	USA	Arkansas	USD	103	303	503	703	903
North America	USA	California	USD	104	304	504	704	904
North America	USA	Colorado	USD	105	305	505	705	905
North America	USA	Connecticut	USD	106	306	506	706	906
North America	USA	Delaware	USD	107	307	507	707	907
North America	USA	Florida	USD	108	308	508	708	908
North America	USA	Georgia	USD	109	309	509	709	909
North America	USA	Hawaii	USD	110	310	510	710	910
North America	USA	Idaho	USD	111	311	511	711	911
North America	USA	Illinois	USD	112	312	512	712	912
North America	USA	Indiana	USD	113	313	513	713	913
North America	USA	Iowa	USD	114	314	514	714	914

North America	USA	Kansas	USD	115	315	515	715	915
North America	USA	Kentucky	USD	116	316	516	716	916
North America	USA	Louisiana	USD	117	317	517	717	917
North America	USA	Maine	USD	118	318	518	718	918
North America	USA	Maryland	USD	119	319	519	719	919
North America	USA	Massachusetts	USD	120	320	520	720	920
North America	USA	Michigan	USD	121	321	521	721	921
North America	USA	Minnesota	USD	122	322	522	722	922
North America	USA	Mississippi	USD	123	323	523	723	923
North America	USA	Missouri	USD	124	324	524	724	924
North America	USA	Montana	USD	125	325	525	725	925
North America	USA	Nebraska	USD	126	326	526	726	926
North America	USA	Nevada	USD	127	327	527	727	927
North America	USA	New Hampshire	USD	128	328	528	728	928
North America	USA	New Jersey	USD	129	329	529	729	929
North America	USA	New Mexico	USD	130	330	530	730	930
North America	USA	New York	USD	131	331	531	731	931
North America	USA	North Carolina	USD	132	332	532	732	932
North America	USA	North Dakota	USD	133	333	533	733	933
North America	USA	Ohio	USD	134	334	534	734	934
North America	USA	Oklahoma	USD	135	335	535	735	935
North America	USA	Oregon	USD	136	336	536	736	936
North America	USA	Pennsylvania	USD	137	337	537	737	937
North America	USA	Rhode Island	USD	138	338	538	738	938
North America	USA	South Carolina	USD	139	339	539	739	939
North America	USA	South Dakota	USD	140	340	540	740	940
North America	USA	Tennessee	USD	141	341	541	741	941
North America	USA	Texas	USD	142	342	542	742	942
North America	USA	Utah	USD	143	343	543	743	943
North America	USA	Vermont	USD	144	344	544	744	944
North America	USA	Virginia	USD	145	345	545	745	945
North America	USA	Washington	USD	146	346	546	746	946
North America	USA	West Virginia	USD	147	347	547	747	947
North America	USA	Wisconsin	USD	148	348	548	748	948
North America	USA	Wyoming	USD	149	349	549	749	949
South America	Argentina	Argentina	ARS	152	352	552	752	952

North America	Canada	Canada	CAD	153	353	553	753	953
Africa	South Africa	South Africa	ZAR	150	350	550	750	950

9. Troubleshooting

This section has been included in the manual to provide integrators with information to help resolve any issues.

9.1 Deserialization Errors

When the elements which make up the XML document are populated with the necessary data, if the 'Type' of each is not adhered to then deserialization errors can occur.

Essentially, the information provided does not meet the format requirements and, when the XML document has been programmatically checked, it could not be parsed correctly.

An example of this would be:

If the 'Track2' element were to be populated with data that does not match the predefined type, (in this case, the track2 data should consist of a numeric value), then a deserialization error will be produced. The error will detail which element(s) contained the mismatch, which enables the problem to be resolved by checking what information was passed and comparing this to the required type.

1) The XML document 'TRecord' contains PAN information that doesn't satisfy the type requirements:

```
....
<PAN>CardNumber</PAN>
....
```

2) A deserialization error is returned within the 'InternalError' tag, detailing which field caused the issue (in this case 'PAN'):

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
<soap:Body>
<CardTxnResponse xmlns="https://www.commidea.webservices.com">
<CommideaTxnResponse>
<StdResponse />
<InternalError>[16277] TransactionProcessor.DeserializeXml: Bad Format in
TRecord_Pan</InternalError>
</CommideaTxnResponse>
</CardTxnResponse>
</soap:Body>
</soap:Envelope>
```

9.2 Contact Information

Should there be a need to contact Commidea for help, please use the below contact details:

In a Test Environment:

Implementations
implementations@commidea.com
08444 828 273

In a Live Environment:

Merchant Helpdesk
helpdesk@commidea.com
08444 828 222

APPENDIX A – Website Testing Script

To aid developers before Commidea integration testing is performed; please follow the table below to ensure that all recommendations and requirements have been met:

Test/Scenario	*Description/Reasoning*	*Test Result*
Perform a normal transaction	Ensure solution processes transactions correctly	PASS / FAIL
	Check the modifiers used to mark the transaction are correct, e.g. 'Purchase' (1) and 'Keyed Customer Not Present E-Commerce' (12). These may differ for different scenarios. A transaction can be marked with more than one modifier	PASS / FAIL
Perform a voice referral transaction (value ends in 2p)	Ensure voice referral transactions are rejected with a <Command> of 2 (rejected) for websites	PASS / FAIL
	Check referral message does not say the transaction has been "declined". Use "unsuccessful, as the scenario is different from that of declines	PASS / FAIL
Process a declined transaction (value ends in 5p)	Ensure solution reacts correctly to declined transactions, including supplying an appropriate a message	PASS / FAIL
Check for SSL certificate	Must be installed on the site to ensure credit card information is handled securely	PASS / FAIL
Navigating between payment forms disabled	Secure information from website pages should be cleared once the page is left, or returning using the 'Back' button should be disabled	PASS / FAIL
Process a transactions with various issue numbers	The issue number field should allowing processing of cards with issue numbers ranging from 1-99, including 01. Ensure leading 0's are not removed after submission	PASS / FAIL
Process a transaction using a card number with 13 digits	This is the least amount of digits a card number can consist of (13)	PASS / FAIL
Process a transaction with a Maestro card	This is the greatest amount of digits a card number can consist of (19)	PASS / FAIL
Process a transaction with 20 alpha characters as a card number	The card number field should validate locally and reject any attempts to enter more than 19 characters. Entry of alpha characters in this field should be disabled	PASS / FAIL
Process a transaction using invalid start date, expiry date and issue number	All fields should be validated locally, ensure that invalid start date, expiry date and issue number entry is disabled (formatting the field accordingly)	PASS / FAIL
In the event of a transaction confirmation response not being received, resend the confirmation	Confirmation should be resent rather than creating a new transaction if no confirmation response is received. This avoids duplicating orders	PASS / FAIL
Process a transaction using a CV2 value of '000' (if applicable)	This test will check leading 0's are not being removed from the record before it is sent to Commidea. On the test system, '000' is the only CV2 value accepted	PASS / FAIL
Process a transaction using an AMEX card (if applicable)	This will ensure that the CV2 field allows entry of up to 4 digits, which is a requirement for processing AMEX cards. Please note that AMEX cards do now support 3 digit CSC values	PASS / FAIL
Process a transaction using AVS information (if applicable)	Test scenarios whereby different AVS data is used. Here are three tests to perform with the relevant test data listed:	

	i)	'Matched' – 10;ME156LH	PASS / FAIL
	ii)	'Partial Match' – 11;ME156LH	PASS / FAIL
	iii)	'Not Matched' – 11;ME167LH (or any other address)	PASS / FAIL

APPENDIX B – Currency Code ISO 4217

Alpha code	Numeric code	Currency	Entity
AED	784	UAE Dirham	United Arab Emirates
AFA	4	Afghani	Afghanistan
ALL	8	Lek	Albania
ANG	532	Antillian Guilder	Netherlands Antilles
AOK	24	Kwanza	Angola
ARS	32	Argentine Peso	Argentina
ATS	40	Schilling	Austria
AUD	36	Australian Dollar	Australia
BBD	52	Barbados Dollar	Barbados
BDT	50	Taka	Bangladesh
BEF	56	Belgian Franc	Belgium
BGL	100	Lev	Bulgaria
BHD	48	Bahraini Dinar	Bahrain
BIF	108	Burundi Franc	Burundi
BMD	60	Bermudan Dollar	Bermuda
BND	96	Brunei Dollar	Brunei
BOP	68	Bolivian Peso	Bolivia
BRC	76	Cruzeiro	Brazil
BSD	44	Bahamian Dollar	Bahamas
BUK	104	Kyat	Burma
BWP	72	Pula	Botswana
BZD	84	Belize Dollar	Belize
CAD	124	Canadian Dollar	Canada
CHF	756	Swiss Franc	Switzerland
CLP	152	Chilean Peso	Chile
CNY	156	Yuan Renminbi	China
COP	170	Colombian Peso	Colombia
CRC	188	Costa Rican Colon	Costa Rica
CSK	200	Koruna	Czechoslovakia
CUP	192	Cuban Peso	Cuba
CVE	132	Cape Verde Escudo	Cape Verde
CYP	196	Cyprus Pound	Cyprus
DDM	278	Mark der DDR	German Democratic Republic
DEM	280	Deutsche Mark	Germany, Federal Republic of
DJF	262	Djibouti Franc	Djibouti
DKK	208	Danish Krone	Denmark
DOP	214	Dominican Peso	Dominican Republic
DZD	12	Algerian Dinar	Algeria
ECS	218	Sucre	Ecuador
EGP	818	Egyptian Pound	Egypt
ESP	724	Spanish Peseta	Spain
ETB	230	Ethiopian Birr	Ethiopia
EUR	978	Euro	European Union
FIM	246	Markka	Finland
FJD	242	Fiji Dollar	Fiji
FKP	238	Falkland Islands Pound	Falkland Islands (Malvinas)
FRF	250	French Franc	France
GBP	826	Pound Sterling	United Kingdom
GHC	288	Cedi	Ghana
GIP	292	Gibraltar Pound	Gibraltar
GMD	270	Dalasi	Gambia
GNS	324	Syli	Guinea
GQE	226	Ekwele	Equatorial Guinea
GRD	300	Drachma	Greece
GTQ	320	Quetzal	Guatemala
GWP	624	Guinea-Bissau Peso	Guinea-Bissau
GYD	328	Guyana Dollar	Guyana
HKD	344	Hong Kong Dollar	Hong Kong
HNL	340	Lempira	Honduras
HTG	332	Gourde	Haiti
HUF	348	Forint	Hungary
IDR	360	Rupiah	Indonesia
IEP	372	Irish Pound	Ireland
ILS	376	Shekel	Israel
INR	356	Indian Rupee	India
IQD	368	Iraqi Dinar	Iraq

IRR	364	Iranian Rial	Iran
ISK	352	Iceland Krona	Iceland
ITL	380	Lira	Italy
JMD	388	Jamaican Dollar	Jamaica
JOD	400	Jordanian dinar	Jordan
JPY	392	Yen	Japan
KES	404	Kenyan Shilling	Kenya
KHR	116	Riel	Kampuchea, Democratic
KMF	174	Comoros Franc	Comoros
KPW	408	North Korean Won	Korea, Democratic People's of
KRW	410	Won	Korea, Republic of
KWD	414	Kuwaiti Dinar	Kuwait
KYD	136	Cayman Islands Dollar	Cayman Islands
LAK	418	Kip	Lao People's Democratic Republic
LBP	422	Lebanese Pound	Lebanon
LKR	144	Sri Lanka Rupee	Sri Lanka
LRD	430	Liberian Dollar	Liberia
LSM	426	Maloti	Lesotho
LUF	442	Luxembourg Franc	Luxembourg
LYD	434	Libyan Dinar	Libyan Arab Jamahiriya
MAD	504	Moroccan Dirham	Morocco
MGF	450	Malagasy Franc	Madagascar
MLF	466	Mali Franc	Mali
MNT	496	Tugrik	Mongolia
MOP	446	Pataca	Macau
MRO	478	Ouguiya	Mauritania
MTP	470	Maltese Pound	Malta
MUR	480	Mauritius Rupee	Mauritius
MVR	462	Maldive Rupee	Maldives
MWK	454	Kwacha	Malawi
MXP	484	Mexican Peso	Mexico
MYR	458	Malaysian Ringgit	Malaysia
MZM	508	Metical	Mozambique
NGN	566	Naira	Nigeria
NIC	558	Cordoba	Nicaragua
NLG	528	Netherlands Guilder	Netherlands
NOK	578	Norwegian Krone	Norway
NPR	524	Nepalese Rupee	Nepal
NZD	554	New Zealand Dollar	New Zealand
OMR	512	Rial Omani	Oman
PAB	590	Balboa	Panama
PES	604	Sol	Peru
PGK	598	Kina	Papau New Guinea
PHP	608	Philippine Peso	Philippines
PKR	586	Pakistan Rupee	Pakistan
PLZ	616	Zloty	Poland
PTE	620	Portugese Escudo	Portugal
PYG	600	Guarani	Paraguay
QAR	634	Qatari Rial	Qatar
ROL	642	Leu	Romania
RWF	646	Rwanda Franc	Rwanda
SAR	682	Saudi Riyal	Saudi Arabia
SBD	90	Solomon Islands Dollar	Solomon Islands
SCR	690	Seychelles Rupee	Seychelles
SDP	736	Sudanese Pound	Sudan
SEK	752	Swedish Krona	Sweden
SGD	702	Singapore Dollar	Singapore
SHP	654	St. Helena Pound	St. Helena
SLL	694	Leone	Sierra Leone
SOS	706	Somali Shilling	Somalia
SRG	740	Suriname Guilder	Suriname
STD	678	Dobra	Sao Tome and Principe
SUR	810	Rouble	Union of Soviet Socialist Republics
SVC	222	El Salvador Colon	El Salvador
SYP	760	Syrian Pound	Syrian Arab Republic
SZL	748	Lilangeni	Swaziland
THB	764	Baht	Thailand
TND	788	Tunisian Dinar	Tunisia
TOP	776	Pa'anga	Tonga

TPE	626	Timor Escudo	East Timor
TRL	792	Turkish Lira	Turkey
TTD	780	Trinidad&Tobago Dollar	Trinidad and Tobago
TWD	901	New Taiwan Dollar	Taiwan, Province of China
TZS	834	Tanzanian Shilling	Tanzania, United Republic of
UGS	800	Uganda Shilling	Uganda
USD	840	US Dollar	United States
USN	997	US Dollar (Next day)	United States
USS	998	US Dollar (same day)	United States
UYP	858	Uruguayan Peso	Uruguay
VND	704	Dong	Viet Nam
VEB	862	Bolivar	Venezuela
VUV	548	Vatu	Vanuatu
WST	882	Tala	Samoa
XAF	950	CFA Franc BEAC	Cameroon, Central African Republic, Chad, Congo ,Gabon
XCD	951	East Caribbean Dollar	Antiqua
XDR	960	Special Drawing Rights	International Monetary Fund
XEU	954	European Currency Unit	Euro. Monetary Cooperation Fund (EMCF)
XOF	952	CFA Franc BCEAO	Benin, Ivory coast, Niger, Senegal, Togo, Upper Volta
XPF	953	CFP Franc	French polynesia
YDD	720	Yemeni Dinar	Yemen, Democratic
YER	886	Yemeni Rial	Yemen
YUD	890	New Yugoslavian Dinar	Yugoslavia
ZAR	710	Rand	South Africa
ZMK	894	Kwacha	Zambia
ZRZ	180	Zaire	Zaire
ZWD	716	Zimbabwe Dollar	Zimbabwe

APPENDIX C – Country Codes ISO 3166

Country Name	*Country Code*	*Country Number*
AFGHANISTAN	AF	4
ALBANIA	AL	8
ALGERIA	DZ	12
AMERICAN SAMOA	AS	16
ANDORRA	AD	20
ANGOLA	AO	24
ANGUILLA	AI	660
ANTARCTICA	AQ	10
ANTIGUA AND BARBUDA	AG	28
ARGENTINA	AR	32
ARMENIA	AM	51
ARUBA	AW	533
AUSTRALIA	AU	36
AUSTRIA	AT	40
AZERBAIJAN	AZ	31
BAHAMAS	BS	44
BAHRAIN	BH	48
BANGLADESH	BD	50
BARBADOS	BB	52
BELARUS	BY	112
BELGIUM	BE	56
BELIZE	BZ	84
BENIN	BJ	204
BERMUDA	BM	60
BHUTAN	BT	64
BOLIVIA	BO	68
BOSNIA AND HERZEGOWINA	BA	70
BOTSWANA	BW	72
BOUVET ISLAND	BV	74
BRAZIL	BR	76
BRITISH INDIAN OCEAN TERRITORY	IO	86
BRUNEI DARUSSALAM	BN	96
BULGARIA	BG	100
BURKINA FASO	BF	854
BURUNDI	BI	108
CAMBODIA	KH	116
CAMEROON	CM	120
CANADA	CA	124
CAPE VERDE	CV	132
CAYMAN ISLANDS	KY	136
CENTRAL AFRICAN REPUBLIC	CF	140
CHAD	TD	148
CHILE	CL	152
CHINA	CN	156
CHRISTMAS ISLAND	CX	162
COCOS (KEELING) ISLANDS	CC	166
COLOMBIA	CO	170
COMOROS	KM	174
CONGO	CG	178
CONGO, THE DEMOCRATIC REPUBLIC OF THE	CD	180
COOK ISLANDS	CK	184
COSTA RICA	CR	188
COTE D'IVOIRE	CI	384

CROATIA (local name: Hrvatska)	HR	191
CUBA	CU	192
CYPRUS	CY	196
CZECH REPUBLIC	CZ	203
DENMARK	DK	208
DJIBOUTI	DJ	262
DOMINICA	DM	212
DOMINICAN REPUBLIC	DO	214
EAST TIMOR	TP	626
ECUADOR	EC	218
EGYPT	EG	818
EL SALVADOR	SV	222
EQUATORIAL GUINEA	GQ	226
ERITREA	ER	232
ESTONIA	EE	233
ETHIOPIA	ET	231
FALKLAND ISLANDS (MALVINAS)	FK	238
FAROE ISLANDS	FO	234
FIJI	FJ	242
FINLAND	FI	246
FRANCE	FR	250
FRANCE, METROPOLITAN	FX	249
FRENCH GUIANA	GF	254
FRENCH POLYNESIA	PF	258
FRENCH SOUTHERN TERRITORIES	TF	260
GABON	GA	266
GAMBIA	GM	270
GEORGIA	GE	268
GERMANY	DE	276
GHANA	GH	288
GIBRALTAR	GI	292
GREECE	GR	300
GREENLAND	GL	304
GRENADA	GD	308
GUADELOUPE	GP	312
GUAM	GU	316
GUATEMALA	GT	320
GUINEA	GN	324
GUINEA-BISSAU	GW	624
GUYANA	GY	328
HAITI	HT	332
HEARD AND MC DONALD ISLANDS	HM	334
HOLY SEE (VATICAN CITY STATE)	VA	336
HONDURAS	HN	340
HONG KONG	HK	344
HUNGARY	HU	348
ICELAND	IS	352
INDIA	IN	356
INDONESIA	ID	360
IRAN (ISLAMIC REPUBLIC OF)	IR	364
IRAQ	IQ	368
IRELAND	IE	372
ISRAEL	IL	376
ITALY	IT	380
JAMAICA	JM	388
JAPAN	JP	392

JORDAN	JO	400
KAZAKHSTAN	KZ	398
KENYA	KE	404
KIRIBATI	KI	296
KOREA, DEMOCRATIC PEOPLE'S REPUBLIC OF	KP	408
KOREA, REPUBLIC OF	KR	410
KUWAIT	KW	414
KYRGYZSTAN	KG	417
LAO PEOPLE'S DEMOCRATIC REPUBLIC	LA	418
LATVIA	LV	428
LEBANON	LB	422
LESOTHO	LS	426
LIBERIA	LR	430
LIBYAN ARAB JAMAHIRIYA	LY	434
LIECHTENSTEIN	LI	438
LITHUANIA	LT	440
LUXEMBOURG	LU	442
MACAU	MO	446
MACEDONIA, THE FORMER YUGOSLAV REPUBLIC	MK	807
MADAGASCAR	MG	450
MALAWI	MW	454
MALAYSIA	MY	458
MALDIVES	MV	462
MALI	ML	466
MALTA	MT	470
MARSHALL ISLANDS	MH	584
MARTINIQUE	MQ	474
MAURITANIA	MR	478
MAURITIUS	MU	480
MAYOTTE	YT	175
MEXICO	MX	484
MICRONESIA, FEDERATED STATES OF	FM	583
MOLDOVA, REPUBLIC OF	MD	498
MONACO	MC	492
MONGOLIA	MN	496
MONTSERRAT	MS	500
MOROCCO	MA	504
MOZAMBIQUE	MZ	508
MYANMAR	MM	104
NAMIBIA	NA	516
NAURU	NR	520
NEPAL	NP	524
NETHERLANDS	NL	528
NETHERLANDS ANTILLES	AN	530
NEW CALEDONIA	NC	540
NEW ZEALAND	NZ	554
NICARAGUA	NI	558
NIGER	NE	562
NIGERIA	NG	566
NIUE	NU	570
NORFOLK ISLAND	NF	574
NORTHERN MARIANA ISLANDS	MP	580
NORWAY	NO	578
OMAN	OM	512
PAKISTAN	PK	586
PALAU	PW	585

PANAMA	PA	591
PAPUA NEW GUINEA	PG	598
PARAGUAY	PY	600
PERU	PE	604
PHILIPPINES	PH	608
PITCAIRN	PN	612
POLAND	PL	616
PORTUGAL	PT	620
PUERTO RICO	PR	630
QATAR	QA	634
REUNION	RE	638
ROMANIA	RO	642
RUSSIAN FEDERATION	RU	643
RWANDA	RW	646
SAINT KITTS AND NEVIS	KN	659
SAINT LUCIA	LC	662
SAINT VINCENT AND THE GRENADINES	VC	670
SAMOA	WS	882
SAN MARINO	SM	674
SAO TOME AND PRINCIPE	ST	678
SAUDI ARABIA	SA	682
SENEGAL	SN	686
SEYCHELLES	SC	690
SIERRA LEONE	SL	694
SINGAPORE	SG	702
SLOVAKIA (Slovak Republic)	SK	703
SLOVENIA	SI	705
SOLOMON ISLANDS	SB	90
SOMALIA	SO	706
SOUTH AFRICA	ZA	710
SOUTH GEORGIA AND THE SOUTH SANDWICH ISLANDS	GS	239
SPAIN	ES	724
SRI LANKA	LK	144
ST. HELENA	SH	654
ST. PIERRE AND MIQUELON	PM	666
SUDAN	SD	736
SURINAME	SR	740
SVALBARD AND JAN MAYEN ISLANDS	SJ	744
SWAZILAND	SZ	748
SWEDEN	SE	752
SWITZERLAND	CH	756
SYRIAN ARAB REPUBLIC	SY	760
TAIWAN, PROVINCE OF CHINA	TW	158
TAJIKISTAN	TJ	762
TANZANIA, UNITED REPUBLIC OF	TZ	834
THAILAND	TH	764
TOGO	TG	768
TOKELAU	TK	772
TONGA	TO	776
TRINIDAD AND TOBAGO	TT	780
TUNISIA	TN	788
TURKEY	TR	792
TURKMENISTAN	TM	795
TURKS AND CAICOS ISLANDS	TC	796
TUVALU	TV	798
UGANDA	UG	800

UKRAINE	UA	804
UNITED ARAB EMIRATES	AE	784
UNITED KINGDOM	GB	826
UNITED STATES	US	840
UNITED STATES MINOR OUTLYING ISLANDS	UM	581
URUGUAY	UY	858
UZBEKISTAN	UZ	860
VANUATU	VU	548
VENEZUELA	VE	862
VIET NAM	VN	704
VIRGIN ISLANDS (BRITISH)	VG	92
VIRGIN ISLANDS (U.S.)	VI	850
WALLIS AND FUTUNA ISLANDS	WF	876
WESTERN SAHARA	EH	732
YEMEN	YE	887
YUGOSLAVIA	YU	891
ZAMBIA	ZM	894
ZIMBABWE	ZW	716

APPENDIX D – Performing a LUHN Check

The following steps are involved in this calculation:

Step 1 Double the value of alternate digits beginning with the first right hand digit (low order).

Step 2 Add the individual digit comprising the products obtained in Step 1 to each of the unaffected digits in the original number.

Step 3 Subtract the total obtained in Step 2 from the next higher number ending in 0 (this is the equivalent of calculating the "ten complement" of the low order digit (unit digit) of the total). If the total obtained in Step 2 is a number ending in zero (30, 40, etc.), the check digit is 0.

Example:

Account Number without check digit 4929 123 123 12

Step 1

4	9	2	9	1	2	3	1	2	3	1	2
	X2		X2		X2		X2		X2		X2
4	18	2	18	1	4	3	2	2	6	1	4

Step 2

4+1+8+2+1+8+1+4+3+2+2+6+1+4= 47

Step 3

50 – 47 = 3

Therefore check digit is 3 and complete card number is 4929 123 123 123

APPENDIX E – Commidea Error Codes

Error Code	General Description	Additional Technical Description (if required)	Recommended Action
0001	Unspecified error		Contact Commidea
0002	Invalid transaction type	An example of this could be a Refund being passed when the site are not set up to do so. A trace of what was passed will be in the system log.	Use alternative method for transaction type.
0003	Invalid card / invalid Track2	General card error. Track2 must either be ;PAN=YYMMsss……?x or just the PAN.	Re-enter card number or re-swipe card
0004	Card scheme not recognised	The card Issuer Identification Number (IIN) has not been located in the IIN table. The IIN is typically the first 4 to 6 digits of the card number.	Prompt for alternate method of payment
0005	Card scheme not accepted	The card has been identified, but the card scheme is not accepted at the given site.	Reject Transaction
0006	Invalid card number (lcd)	The LUHN check digit is incorrect (the card has been mis-keyed or mis-swiped).	Re-enter card number or re-swipe card
0007	Invalid card number length	The length of the PAN is incorrect for the given card scheme.	Re-enter card number or re-swipe card
0008	Invalid card number (pcd)	The pen-ultimate check digit is invalid.	Re-enter card number or re-swipe card
0009	Expired card		Prompt for alternate method of payment
0010	Card not yet valid		Prompt for alternate method of payment
0011	Invalid card service code	The Track2 service code is invalid.	Prompt for alternate method of payment
0012	File or XML missing or wrong format	A required file or XML is missing or has wrong format.	Contact Commidea
0013	File permanently locked	A file required by the EFT library was still locked after EFT FIO TRIES attempts.	Contact Commidea
0014	Out of memory	The library has failed to allocate sufficient heap.	Contact Commidea
0015	Account number does not exist	The requested account number does not exist.	Check the account number configuration of the system, ensuring it matches that configured within WinTI
0016	Value exceeds ceiling limit	Purchase value exceeds card scheme ceiling limit	Prompt for alternative method of payment. Arrange to increase ceiling limits
0017	Cashback exceeds ceiling limit	Cashback value exceeds card scheme ceiling limit	Revise transaction cash-back value
0018	Transaction currency is invalid	The transaction currency code is invalid or incorrect for the given site.	
0019	Lay aways are not allowed	Attempt to lay away invalid / lay aways are not allowed	
0020	Lay away already stored	Attempt to lay away a transaction where there is already a transaction laid away on that card	Prompt for alternate method of payment
0021	EFT system not configured	The EFT system has not been configured	
0022	Internal error, buffer too small	A buffer is too small	
0023	Unknown comms device type	Invalid / unknown communications device type	Check communications configuration
0024	Configuration file is invalid	Configuration file is invalid / bad format	Check system configuration
0025	No valid accounts	There are no valid accounts specified in the TillInfo.cfg	Check system configuration
0026	Invalid channel	Invalid channel	Check> · 2 transactions aren't being passed down the same channel. · 2 tills aren't using the same channel number. · WinTI EFTChans within the registry has enough available channels set (Socket mode only).
0027	System error –module not loaded	System error (Track2 check module has not been loaded)	
0028	General transaction error		Re-enter transaction
0029	Transaction store unavailable	Transaction store unavailable	Check Live Store.

			Check hard disk space.
0030	Unspecified error	Unspecified error	Check system log for indication of error.
0031	Unspecified error:2	Transaction cancelled	Channel available for next transaction
0032	Library not open	EFT library is unavailable	
0033	Possible text for error: <fieldname> (<fieldno>) should be X to Y characters in length. <fieldname> out of range, should be X to Y. <fieldname> out of tolerance, is X, should be X +/- Z. Line discount not available for Cendant cards. Line count (X) doesn't match header -> CPC lines (Y). Separate post and packing only on Amex cards. Where <fieldname> = part number, part description, commodity code, unit of measure, quantity, net value, VAT amount, gross value, PAN, PO number, customer number, customer name, customer VAT no, destination zip, destination country code, order date, original invoice number, cost centre, invoice net amount, invoice VAT amount, post and packing VAT, invoice gross or transaction total. Invalid CPC data	The error message is made up of a combination of text (1 to 6) with the applicable field name inserted, as applicable. For example: Net value out of tolerance, is 123.45, should be 123.00 +/- 1	
0034	Modifier field invalid/missing	As the modifier is passed within the T record the host software is likely to be the cause of this	
0035	Invalid card / invalid Track 1	Track 1 is invalid	Re-swipe card
0036	Invalid card / invalid Track 3	Track 3 is invalid	Re-swipe card
0037	Invalid / missing expiry date	The expiry date is either invalid or missing. If key entered, the format should be MMYY	Re-enter expiry date or re-swipe card
0038	Invalid / missing issue number	The issue number is either invalid (value or length) or missing	Re-enter issue number or re-swipe card
0039	Invalid / missing start date	The start date is either invalid or missing. If key entered, the format should be MMYY.	Re-enter start date or re-swipe card
0040	Purchase/refund value bad or missing	The transaction value is either invalid or missing	Re-enter transaction
0041	Cash-back value bad or missing	The cash-back value is either invalid or missing	Re-enter transaction
0042	Auth code value bad or missing	The authorisation code is either invalid or missing	
0043	Cheque account number value bad or missing	The cheque account number is either invalid or missing	Re-enter cheque account number
0044	Invalid cheque sort code	The cheque sort code is either invalid or missing	Re-enter sort code
0045	Invalid / missing cheque number		Re-enter cheque number
0046	Invalid / missing cheque type		Re-enter cheque type
0047	Invalid EFT serial number	The EFT serial number is either invalid or missing in the .Cnf file	Re create *.cnf
0048	Unexpected CPC data	Purchasing card invoice data has been presented for a non-Purchasing Card (where invoice data is not valid/required)	Re-enter transaction without invoice data or prompt for a valid Purchasing Card
0049	Transaction already confirmed or rejected	Attempt to confirm or reject a transaction, which has already been confirmed or rejected	
0050	Copy protection failure	Could be a permission problem on the PC	
0051	Post confirm reversal not allowed for PWCB or Cash Advance (reserved for future use)	Attempt to perform a post confirm reversal on a PWCB or Cash Advance has been dis-allowed (as post confirm reversals are not supported when	Reverse transaction manually (as cash is involved)

		cash is involved)	
0052	Transaction data supplied in post conf rev not consistent with store (reserved for future use)	The details supplied in the post confirm reversal message is not consistent with the data stored for the transaction to be reversed	
0053	Transaction already void	Attempt to perform a post transaction reversal has failed because the transaction has already been voided/reversed	
0054	Card on hot list	The card number is on the locally stored host list (received from the acquirer and/or entered by the customer). The card must be rejected	Prompt for alternate method of payment
0055	Attempt to confirm a declined transaction	The format of the confirmation message is invalid (confirming a declining transaction). The confirmation message should contain a command value of 2 (reverse/reject) and not a value of 1 (confirm).	
0056	EFT_ERR_BAD_CV2	CV2 is invalid	Check CV2 and re-enter
0057	EFT_ERR_BAD_AVS	AVS is invalid	Check AVS and re-enter
0058	Invalid Merchant Details	Mechant Details passed in XML Gateway are Invalid.	Check both the GUID and Passcode information that being passed to the XML Gateway
0059	Invalid Mobile Number Format	The Mobile Number format passed is incorrect	Please check and re-enter the mobilenumber supplied.
0060	Invalid/missing bank account number	The bank account number within the supplied T-Record is incorrect.	Check the bank number being passed and re-enter as necessary.
0064	Unexpected / Invalid Authorisation Response	Unexpected / Invalid Authorisation Response from M-Voucher Host	Please contact Commidea Support
0067	Report Not Supported	The Report ID supplied is either invalid or does not correspond to a report that is supported	Check the Report ID that is being passed
0068	Report Failed	Integrated report failed	Contact Commidea
0069	Gratuity value exceeded	Check Gratuity Value	Check Gratuity Value
0070	Invalid Capture Not Supported	Check Ocius settings	Capture Method Not Set correctly
0071	Cashback not allowed by card	Card does not allow cashback	Use a different card or proceed without cashback
0072	Cash advance not allowed by card	Card does not allow cash advance	Use a different card
0073	Max refund value exceeded	Refund transaction value is greater than the maximum refund value set on the account	Reduce transaction value
0074	Bill Already Complete	The bill being cancelled is already completed and therefore cannot be cancelled.	N\A
0075	No ETU accounts	Attempt to process ETU transaction without ETU accounts being present on terminal	Contact Commidea
0076	Card is online only	Attempt to process an online only card whilst offline	Check network or use another card
0077	Cancel Failed - In Payment on xxx.xxx.xxx.xxx	Attempt to cancel a lodged Bill failed, usually locked on a specific terminal	Leave for configured amount of time before retrying cancel routine.
0078	Login failed	User ID or PIN is incorrect	Check login details and try again
0079	Confirmation Status Unknown	An invalid confirmation response has been received or the confirmation message to be sent was not saved	
0080	Bill Reference Already Exists	Attempt to lodge a Bill into I-Link that already exists	Clear the original Bill, or re-send this one using an alternative reference.
0081	Print Report Failed	The request report failed to generate or print	Check printer settings, network connection and try again.
0082	Network Error	Error in Network	Check network.
0083	Invalid Record	Invalid Record	The record received is invalid.
0084	PED User already logged in	A Login command has been received, but a user is already logged in	Log the terminal off first, or simply pass a transaction.
0085	PED User not logged in	The terminal needs to be logged in	Send a login command to the terminal, or manually login using the on-screen prompts, then re-send the transaction.
0086	Submission of offline transactions failed	The submission of the offline stored transactions have failed.	The transactions will still be stored on the terminal. Re-try, and if still having problems contact The Merchant

			Helpdesk.
0087	Problem in network	There has been a problem in the network.	
0088	Voice Referral Timeout	The voice referral transaction has taken too long.	Re-try or cancel.
0089	Invalid Account ID	Invalid Account ID	
0090	Service Not Allowed	Service code not supported	Use another card, or cancel the transaction
0091	Card Not Accepted	Card type not accepted	Use another card, or cancel the transaction
0092	Unknown Card	Unknown card type	Use another card, or cancel the transaction
0093	Not In IIN Range	Unknown card type	Use another card, or cancel the transaction
0094	Application Blocked	The terminal cannot accept this card type	Use another card, or cancel the transaction
0095	Card Blocked	The card has been blocked.	Use another card, or cancel the transaction.
0096	Card Error	There is a problem with the Card	Re-try or use another card.
0097	Authorisation Error	The authorisation process has been interrupted or is not responding.	Check ILink & WinTI are running – or when using ICP, contact Commidea Merchant Helpdesk.
0098	Unknown Client Unknown Transaction Source Unknown Message	When using transaction processing, if no POS Routing has been configured for the IP Address or File Name where the transaction originates from, ILink does not know where to send the transaction. It therefore rejects it with this message.	Configure POS routing for that Point Of Sale.
0099	Transaction/Bill Cancelled	When a transaction has been cancelled by the user, the system or an ICC card, this error message will be sent.	
0100	Pin Bypass Failed	ICC Card does not allow Pin Bypass.	Use another card.
0101	Invalid Terminal Country Code'	The Terminal Country Code passed is invalid	Please check the ISO Country Codes table and make sure the code being passed is correct.
0102	User has no permissions on specified account	Check account permissions in WebCom.	Please contact Commidea Support
0103	Invalid Currency Code'	The Currency Code passed is invalid.	Please check the ISO Currency Codes table and make sure the code being passed is correct.
0104	Invalid EMV Terminal Type'	The EMV Terminal Type passed is invalid	Please check the EMV Terminal Type that is being passed is valid.
0105	Unknown Message Type	The message type received by server side is not recognised	Please contact Commidea Support
0106	General Enqueue Error	General Commidea Enqueueing Error	Please contact Commidea Support
0107	Transaction Confirmation Error	The transaction confirmation has errored.	Please retry the confirmation and if continues to fail please contact Commidea Support
0108	Payer Auth Error	The Payer Auth has encountered an error.	Please check the error message response and contact Commidea support.
0109	Ukash Auth Error	The Ukash transaction has encountered an error.	Please check the error message response and Contact Commidea Support.
0110	Encryption Failure	An error has occurred in the data encryption.	Please contact Commidea Support
0111	Unable to build Auxillary Data Record	The auxillary data record failed to build correctly	Please contact Commidea Support
0112	Transaction rejection error	The attempt to reject the transaction has errored	Please retry the rejection and if continues to fail please contact Commidea support
0120	Token Server Error	The Token Server has encountered an error	Please contact Commidea Support
0121	Purchase transaction type not allowed on token	The token provided does not allow purchase transactions	Please supply another token that allows purchase transactions
0122	Refund transaction type not allowed on token	The token provided does not allow refund transactions	Please supply another token that allows refund transactions
0123	Cashback transaction type not allowed on token	The token provided does not allow cashback transactions	Please supply another token that allows cashback transactions
0124	Token expired	The token provided has passed its expiry date	Please register a new token
0125	Invalid TokenID	The token provided is invalid	Please supply another token or contact

			Commidea Support
0126	Token has no Txn Type Permissions	The Token Registration has no transaction permissions	Please resubmit the token request with transaction permissions enabled
0127	Invalid Token expiration date	The token expiration date provided is invalid	Please resubmit the token request with a valid token expiration date
0189	Invalid refund password	An invalid refund password has been supplied during the transaction, and was rejected by the database	Please contact Commidea Support