

SecurePay



Secure XML API Integration Guide

Document Control

This is a controlled document

DESCRIPTION	Secure XML API Integration Guide		
CREATION DATE	02/04/2007	CREATED BY	SecurePay
VERSION	1.11	DATE UPDATED	03/06/2025
CHANGES	V1.11 - Updated logo V1.10 - Updated supported TLS versions V1.09 Updated Appendix L Amendments have been made as part of SecurePay's PCI compliance, SecurePay must only support encryption ciphers considered secure.		

Table of Contents

1 Introduction.....	5	5.1 XML Header	25
1.1 What is Secure XML API.....	5	5.2 Common Response Elements	25
1.2 About this Guide.....	5	MessageInfo Element	25
1.3 Intended Audience	5	MerchantInfo Element	25
1.4 Authentication, Communication & Encryption	5	RequestType Element	26
1.5 Feedback	5	Status Element	26
2 Test Account and Payment URLs	6	5.3 Transaction Response Elements.....	26
2.1 Transaction URLs	6	Payment Element	26
2.2 How to use the Test Environment	6	5.4 FraudGuard Response Elements	31
Public Test Account Details	6	FraudGuard Element	31
Test Card Number	7		
Simulating Approved and Declined Transactions	7	6 Sample XML Request and Response	35
3 XML Message Format and Contents.....	8	6.1 Credit Card Payment	35
3.1 TLS Support	8	Request 35	
TLS Protocol version	8	Response35	
TLS Encryption	8	6.2 Credit Card Payment with 3DS2 details37	
TLS Cipher Suites	8	Request 37	
3.2 HTTP Message Structure	8	Response37	
Request 9		6.3 Credit Card Refund.....	39
Response10		Request 39	
3.3 Message types.....	11	Response39	
3.4 Optional Features.....	12	6.4 Account Verification	41
3.5 Transaction Types Required Elements13		Request 41	
Credit Card Transactions	13	Response41	
Direct Entry Transactions	14	6.5 Preauthorisation Payment	43
3.6 Element Types and Constraints.....	14	Request 43	
4 Request Element Definitions	16	Response43	
4.1 XML Header	16	6.6 Direct Debit.....	45
4.2 Common XML Elements.....	16	Request 45	
MessageInfo Element	16	Response45	
MerchantInfo Element	17	6.7 Card Payment with FraudGuard	47
RequestType Element	17	Request 47	
Payment Element	18	Response47	
Integer, see Appendix A: Transaction Types19		6.8 FraudGuard Only Request.....	49
4.3 FraudGuard Request Elements.....	23	Request 49	
BuyerInfo Element	23	Response49	
4.4 Echo Message Elements	24	6.9 Echo	51
Request Messages	24	Request 51	
Echo URLs	24	Response51	
5 Response Element Definitions.....	25	Appendix A: Transaction Types.....	52
		Appendix B: Card Types	53
		Appendix C: Location of CWV.....	54
		Appendix D: Timestamp String Format.....	55

Appendix E: SecurePay Status Codes	56
Appendix F: XML Request DTD.....	57
Appendix G: XML Response DTD	59
Appendix H: Currency Codes List	61
Appendix I: EBCDIC Character Set	62
Appendix K: ISO 3166 Country Codes	63
Appendix L: TLS Cipher Suites	73

1 Introduction

1.1 What is Secure XML API

The Secure XML API is a method for transmitting transaction information to SecurePay for processing. Each XML message sent to SecurePay contains one operation.

This guide covers the process of building a program within your website or application in order to integrate the XML API and can be run on any platform and in any programming language.

The below operations are available through the Secure XML API and detailed in this guide

Credit Card Operations

- Payment
- Refund
- Account Verification
- Preauthorisation (including Initial Authorisation)
- Preauthorisation Increase
- Preauthorisation Cancellation
- Preauthorisation Complete

Direct Entry Operations

- Direct Debit Payment
- Direct Credit Payment

Optional Features

- Recurring flag
- FraudGuard
- Echo

Once composed by your application, an XML Message is sent via the POST method to a HTTPS URL at SecurePay for processing. Once processing is complete, a response message is sent via the POST method back to your server. On average each message is processed in a few seconds.

This gives your application a real time response of the outcome of a credit card transaction. Direct entry payments are not processed in real time; they are stored in SecurePay's database and processed daily at 4.30pm EST.

1.2 About this Guide

This guide provides technical information about integrating and configuring SecurePay within your environment.

1.3 Intended Audience

This document is intended for developers, integrating SecurePay's Secure XML interface into their own applications or websites.

It is recommended that someone with website, XML or application programming experience reads this guide and implements the Secure XML.

1.4 Authentication, Communication & Encryption

To ensure security, each merchant is issued with a password. This password requires authentication before a request can be processed. This makes sure that unauthorised users will be unable to use the interface.

The password can be changed by the merchant via SecurePay's Merchant Management facility.

The Secure XML API uses HTTPS for communication with SecurePay's System.

Merchants using Secure XML will automatically use SecurePay's SSL certificate to encrypt requests and decrypt responses from SecurePay.

1.5 Feedback

Continuous improvement is one of SecurePay's core values. We welcome any feedback you have on our integration guides as a way to help us improve any future changes to our products.

If you wish to leave feedback, please [click here](#).

2 Test Account and Payment URLs

2.1 Transaction URLs

There are several different Payment URLs at SecurePay depending on the type of operation you are performing. Please ensure that you select the correct URL for the message type you are sending.

For credit card transactions:

Test URL: <https://test.api.securepay.com.au/xmlapi/payment>

Live URL: <https://api.securepay.com.au/xmlapi/payment>

For direct entry transactions:

Test URL: <https://test.api.securepay.com.au/xmlapi/directentry>

Live URL: <https://api.securepay.com.au/xmlapi/directentry>

For FraudGuard credit card transactions:

Test URL: <https://test.api.securepay.com.au/antifraud/payment>

Live URL: <https://api.securepay.com.au/antifraud/payment>

2.2 How to use the Test Environment

As you build your system, you can test functionality when necessary by submitting parameters to the test URL.

Public Test Account Details

You can use the below details against the SecurePay test environment.

Integration Details

This is used as part of your transaction messages.

Merchant id: ABC0001

Transaction password: abc123

You can login to the Test SecurePay Merchant Portal with the below details to see the outcome of your testing.

Test Login Details

This is used to login to the merchant portal.

URL: <https://test.login.securepay.com.au>

Merchant id: ABC
User name: test
Login Password: abc1234!!

Test Card Number

Use the following information when testing transactions:

```
Card Number: 4444333322221111  
Card Type: VISA  
Card CCV: 123  
Card Expiry: 08 / 23 (or any date greater than today)
```

Simulating Approved and Declined Transactions

You can simulate approved and declined transactions by submitting alternative payment amounts.

If the payment amount ends in 00, 08, 11, or 77 the transaction will be approved once card details are submitted. All other options will cause a declined transaction.

Payment amounts to simulate approved transactions:

```
$1.00 (100)  
$1.08 (108)  
$105.00 (10500)  
$105.08 (10508)  
(or any total ending in 00, 08)
```

Payment amounts to simulate declined transactions:

```
$1.51 (151)  
$1.05 (105)  
$105.51 (10551)  
$105.05 (10505)  
(or any totals not ending in 00, 08)
```

Note that when using the live URL for payments, the bank determines the transaction result, independent of the payment amount.

3 XML Message Format and Contents

This section describes the possible request message and relevant elements.

3.1 TLS Support

TLS Protocol version

In line with PCI DSS v3.2 requirements, connections to SecurePay shall be configured to only use the TLS v1.2 or TLSv1.3 protocol to protect data in transit. Your servers must be configured to use TLS v1.2 or TLSv1.3 only. TLS 1.1, TLS 1.0 or SSL 3.0 are not supported.

TLS Encryption

Your website must be configured to use HTTPS protocol to encrypt normal HTTP requests and responses, making it safer and secure.

TLS Cipher Suites

Cipher suites specify the cryptographic algorithms that will be used in a session. Your choice of cipher suites also impacts your ability to establish a secure connection with SecurePay. It is recommended to use those cipher suites that support perfect forward secrecy. For a list of supported cipher suites see *Appendix L*.

Cipher suites which are deemed unsecure over time will no longer be supported by SecurePay but merchants will be given ample lead time to use safer alternatives.

3.2 HTTP Message Structure

The structure of the HTTP request and response messages must conform to the HTTP 1.1 network protocol standard.

The HTTP communication between the client and SecurePay Payment Server must be done via SSL so that the sensitive information included in the request and response messages is encrypted.

Your web host cannot use Server Name Indicators (SNIs) for determining which SSL certificate to serve. This is not supported by SecurePay's systems.

The HTTP header must include a Content-Type of 'text/xml'.

The XML Encoding is plain 'UTF-8' text and must not be URL encoded.

If the request includes an 'Accept' header it must include 'text/xml' as a supported content type by the client. Use of the 'Accept' header is strongly discouraged.

Note: DOCTYPE declaration in the XML request is forbidden and will be rejected by the SecurePay server.

Below is an example standard credit card payment request and response including HTTP 1.1 headers.

Request

```
POST /xmlapi/payment HTTP/1.1
host: test.securepay.com.au
content-type: text/xml
content-length: 713
```

```
<?xml version="1.0" encoding="UTF-8"?>
<SecurePayMessage>
  <MessageInfo>
    <messageID>8af793f9af34bea0cf40f5fb750f64</messageID>
    <messageTimestamp>20162201115745000000+660</messageTimestamp>
    <timeoutValue>60</timeoutValue>
    <apiVersion>xml-4.2</apiVersion>
  </MessageInfo>
  <MerchantInfo>
    <merchantID>ABC0001</merchantID>
    <password>abc123</password>
  </MerchantInfo>
  <RequestType>Payment</RequestType>
  <Payment>
    <TxnList count="1">
      <Txn ID="1">
        <txnType>0</txnType>
        <txnSource>23</txnSource>
        <amount>200</amount>
        <currency>AUD</currency>
        <purchaseOrderNo>test</purchaseOrderNo>
        <CreditCardInfo>
          <cardNumber>4444333322221111</cardNumber>
          <expiryDate>09/10</expiryDate>
          <cvv>000</cvv>
        </CreditCardInfo>
      </Txn>
    </TxnList>
  </Payment>
</SecurePayMessage>
```

Response

The initial HTTP server response (100 continue) is to indicate that the request has been received and should be ignored. The 200 response should follow with the XML response message. If content length is 0 and no XML response is included then the request could not be understood and no response was produced.

Please Note: Example message header below, this can change.

```
HTTP/1.1 100 Continue
Server: Apache
Date: Fri, 22 Jan 2016 00:33:10 GMT
HTTP/1.1 200 OK
Content-Type: text/xml; charset=ISO-8859-1
Date: Fri, 22 Jan 2016 00:33:13 GMT
Server: Apache
Connection: close
Content-Length: 1151
```

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<SecurePayMessage>
  <MessageInfo>
    <messageID>8af793f9af34bea0cf40f5fb750f64</messageID>
    <messageTimestamp>20162201113316084000+660</messageTimestamp>
    <apiVersion>xml-4.2</apiVersion>
  </MessageInfo>
  <RequestType>Payment</RequestType>
  <MerchantInfo>
    <merchantID>ABC0001</merchantID>
  </MerchantInfo>
  <Status>
    <statusCode>000</statusCode>
    <statusDescription>Normal</statusDescription>
  </Status>
  <Payment>
    <TxnList count="1">
      <Txn ID="1">
        <txnType>0</txnType>
        <txnSource>23</txnSource>
        <amount>200</amount>
        <currency>AUD</currency>
        <purchaseOrderNo>test</purchaseOrderNo>
        <approved>Yes</approved>
        <responseCode>00</responseCode>
        <responseText>Approved</responseText>
        <settlementDate>20160122</settlementDate>
        <txnID>374389</txnID>
        <CreditCardInfo>
          <pan>444433...111</pan>
          <expiryDate>09/10</expiryDate>
          <cardType>6</cardType>
          <cardDescription>Visa</cardDescription>
        </CreditCardInfo>
      </Txn>
    </TxnList>
  </Payment>
</SecurePayMessage>
```

3.3 Message types

Type	<txnType>	Description
Payment	0	Processes a payment request against a credit card.
Refund	4	Processes a refund against a previous transaction through the SecurePay system back to the original credit card. Transactions may only be refunded up to the original amount processed. Multiple partial refunds are possible.
Preauthorisation	10	Processes a preauthorisation against a credit card. A preauthorisation will reserve funds on the card holder's account, this is generally held for 5 business days. The preauthorisation can be completed after this point, however there is no longer a guarantee that the funds are available. To settle the funds a complete needs to be processed. To process an Initial Authorisation (a preauthorisation for an initial amount), use the <initialAuth> element. Initial Authorisation is supported by Visa and Mastercard only, available on selected acquiring banks (NAB, ANZ, Westpac Qvalent and Fiserv FDMSA) and can be used by merchants in certain industry categories. Refer to the SecurePay website FAQs for more information.
Complete	11	Processes a complete against a previous preauthorisation. Only one complete may be processed against each preauthorisation.
Direct Debit	15	Processes a direct debit from a bank account. This uses the BSB and account number to charge a customer's bank account. To be eligible to use direct debit, you must have an active direct debit account with SecurePay. Direct entry payments are not processed in real time; they are stored in SecurePay's database and processed daily at 4.30pm EST.
Direct Credit	17	Processes a direct credit to a bank account. This uses the BSB and account number to send funds to a customer's bank account. To be eligible to use direct debit, you must have an active direct credit account with SecurePay. Direct entry payments are not processed in real time; they are stored in SecurePay's database and processed daily at 4.30pm EST.
Account Verification	40	Processes an account verification against a credit card. An account verification is a \$0 transaction used to verify the card details and its validity without impacting the customer's available funds. This transaction type is supported by Visa and Mastercard only and available on selected acquiring banks (NAB, ANZ, Westpac Qvalent and Fiserv FDMSA). Refer to the SecurePay website FAQs for more information.
Preauthorisation Increase	41	Processes a preauthorisation increase on a previously approved initial authorisation. A preauthorisation increase allows you to add additional funds to an existing authorisation that has not been completed. For a Visa card – a preauthorisation increase, increases the amount only. For Mastercard – a preauthorisation increase, increases the amount and extends the validity period of the authorisation. Mastercard allows a \$0 increase to extend the validity period only. This transaction is only available on an Initial Authorisation, Visa and Mastercard only, available on selected acquiring banks (NAB, ANZ, Westpac Qvalent and Fiserv FDMSA) and can be used by merchants in certain industry categories. Refer to the SecurePay website FAQs for more information.

Preauthorisation Cancellation	42	Processes a preauthorisation cancellation on a previously approved initial authorisation. A preauthorisation cancellation allows you to remove some or all of the funds from an existing authorisation that has not been completed. This transaction is only available on an Initial Authorisation, Visa and Mastercard only, available on selected acquiring banks (NAB, ANZ*, Westpac Qvalent and Fiserv FDMSA) and can be used by merchants in certain industry categories. Refer to the SecurePay website FAQs for more information. * The ANZ acquirer does not currently support partial cancellations, only full cancellations can be made.
-------------------------------	----	---

3.4 Optional Features

Type	Description
Recurring	This flags the transaction as recurring. When flagged as recurring the transaction is treated differently through the authorisation process. In most cases the expiry date and CVV are ignored by the bank. This allows for easy reoccurring billing, such as subscriptions. Transactions should only be flagged as recurring if the cardholder establishes a relationship with the merchant to receive ongoing services or goods and gives permission to the merchant for ongoing billing. This is currently only supported by Visa and MasterCard. This does not automate transaction processing, it simply flags the once off payment as recurring.
FraudGuard	FraudGuard is an optional fraud mitigation tool. FraudGuard uses a series of merchant defined rules to screen transactions, these rules need to be configured through the SecurePay merchant login. To use this feature some additional details need to be passed through in the XML message. FraudGuard may incur an additional fee and needs to be activated on your SecurePay account prior to use.
Echo	Echo requests can be sent to any of the Payment URLs to verify if the service is available. The Echo messages should not be sent more often than every 5 minutes and only if there were no real transactions processed in the last 5 minutes.

3.5 Transaction Types Required Elements

The tables below show which elements are required for each credit card transaction type. Elements are mandatory, optional, or not required.

Credit Card Transactions

Each element needs to be located within the correct parent element tag as detailed in the following section.

TXN TYPE ELEMENT	Standard Payment 0	Refund 4	Account Verification 40	Preauthorisation 10	Preauthorisation Increase 41	Preauthorisation Cancellation 42	Complete 11	FraudGuard Only Check 22	Echo NA
<messageID>	M	M	M	M	M	M	M	M	M
<messageTimestamp>	M	M	M	M	M	M	M	M	M
<timeoutValue>	M	M	M	M	M	M	M	M	M
<apiVersion>	M	M	M	M	M	M	M	M	M
<merchantID>	M	M	M	M	M	M	M	M	M
<password>	M	M	M	M	M	M	M	M	M
<RequestType>	M	M	M	M	M	M	M	M	M
<txnType>	M	M	M	M	M	M	M	M	X
<initialAuth>	X	X	X	O	X	X	X	X	X
<txnSource>	M	M	M	M	M	M	M	M	X
<amount>	M	M	M(1)	M	M	M	M	M	X
<currency>	O	X	X	O	X	X	X	O	X
<purchaseOrderNo>	M	M	M	M	M	M	M	M	X
<orderId>	O	X	X	O	X	X	X	X	X
<liabilityShiftIndicator>	O	X	X	O	X	X	X	X	X
<recurring>	O	X	X	O	X	X	X	X	X
<txnID>	X	M	X	X	X	X	X	X	X
<preauthID>	X	X	X	X	M	M	M	X	X
<cardNumber>	M	O	M	M	X(2)	X(2)	O	M	X
<cvv>	O	X	O	O	X	X	O	O	X
<expiryDate>	M	O	M	M	X(2)	X(2)	O	M	X
<cardHolderName>	O	O	O	O	O	O	O	X	X
Additional fields for FraudGuard (3)									
<ip>	X(M)	X	X(M)	X(M)	X(M)	X	X	M	X
<zipcode>	X(O)	X	X(O)	X(O)	X(O)	X	X	O	X
<town>	X(O)	X	X(O)	X(O)	X(O)	X	X	O	X
<billingCountry>	X(O)	X	X(O)	X(O)	X(O)	X	X	O	X
<deliveryCountry>	X(O)	X	X(O)	X(O)	X(O)	X	X	O	X
<emailAddress>	X(O)	X	X(O)	X(O)	X(O)	X	X	O	X
M – Mandatory O – Optional X – Not required (ignored) (#) – See notes below: (1) Amount must be 0 for Account Verification requests. (2) The card details are obtained from the original Preauthorisation transaction which is referenced in the 'preauthID' field. Any values that are provided are ignored. (3) The additional fields for FraudGuard will only be processed if sent to the FraudGuard URL. When sending to the FraudGuard URL <ip> is mandatory									

Direct Entry Transactions

Each element needs to be located within the correct parent element tag as detailed in the following section.

TXN TYPE	Direct Debit	Direct Credit
	15	17
ELEMENT		
<messageID>	M	M
<messageTimestamp>	M	M
<timeoutValue>	M	M
<apiVersion>	M	M
<merchantID>	M	M
<password>	M	M
<RequestType>	M	M
<txnType>	M	M
<txnSource>	M	M
<amount>	M	M
<currency>	X	X
<purchaseOrderNo>	M	M
<txnID>	X	X
<preauthID>	X	X
<bsbNumber>	M	M
<accountNumber>	M	M
<accountName>	M	M
M – Mandatory O – Optional X – Not required (ignored)		

Note: Any fields not listed for Direct Entry transactions are ignored.

3.6 Element Types and Constraints

The value format descriptions in sections below use keys from the following table:

Type	Constraint	Description
String	A	Alphabetic characters Value in the element is valid if it only contains characters in the specified set (alphabetic)
	N	Numeric characters Value in the element is valid if it only contains characters in the specified set (numeric)
	S	Special characters Will be followed with a list of allowed characters Value in the element is valid if it only contains characters in the specified set (special characters)
	LEN	Number of characters in the string Value in the element is valid if the length of the value is equal to the defined length
	MINLEN	Minimum number of characters in the string Value in the element is valid if the length of the value is greater than or equal to the defined minimum length
	MAXLEN	Maximum number of characters in the string Value in the element is valid if the length of the value is less than or equal to the defined maximum length

Type	Constraint	Description
Integer	DIGNO	Number of digits in the integer value Value in the element is valid if the number of digits in the value is less than or equal to the defined digits number
	MINVAL	Minimum numerical value Value in the element is valid if it is numerically greater than or equal to the defined minimum value
	MAXVAL	Maximum numerical value Value in the element is valid if it is numerically less than or equal to the defined maximum value

4 Request Element Definitions

The following section describes the elements of a transaction request message.

The following `<RequestType>` element value must be used for all Transaction messages:

`<RequestType>Payment</RequestType>`

4.1 XML Header

The XML request will begin with an XML declaration that contains the following data:

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

Markup	Usage	Explanation
<code><?></code>	required	Begins a processing instruction.
<code>xml</code>	required	Declares this to be an XML instruction.
<code>Version=""</code>	required	Identifies the version of XML specification in use.
<code>Encoding=""</code>	required	Indicates which international character set is used.
<code>?></code>	required	Terminates the processing instruction.

The XML request must contain a following top level (root) element: `<SecurePayMessage>`

4.2 Common XML Elements

Following sections describe elements common to all requests.

MessageInfo Element

Description:	Identifies the message.
Format type:	(No value)
Format constraints:	(No value)
Validated by SecurePay:	Yes
Value:	(No value)
Sub-elements:	Yes, see table below

`<MessageInfo>` sub-elements:

Element	Comments
<code><messageID></code>	Description: Unique identifier for the XML message. Format type: String Format constraints: AN, MINLEN = 0, MAXLEN = 30 Validated by SecurePay: Yes Value: Eg: "8af793f9af34bea0cf40f5fb5c630c" Sub-elements: No
<code><messageTimestamp></code>	Description: Time of the request. Format type: String, see Appendix D: Timestamp String Format Format constraints: NS ('+', '-'), LEN = 24 Validated by SecurePay: Yes Value: Eg: "20041803161306527000+660" Sub-elements: No
<code><timeoutValue></code>	Description: Timeout value used, in seconds. Format type: Integer Format constraints: DIGNO = 3, MINVAL = 1 Validated by SecurePay: Yes Value: Recommended "120" Sub-elements: No

<apiVersion>	Description: Version of the product used. Format type: String Format constraints: ANS ('-', '.'), MINLEN = 1, MAXLEN = 13 Validated by SecurePay: Yes Value: Always "xml-4.2" Sub-elements: No
--------------	---

MerchantInfo Element

Description:	Identifies the merchant.
Format type:	(No value)
Format constraints:	(No value)
Validated by SecurePay:	Yes
Value:	(No value)
Sub-elements:	Yes, see table below

<MerchantInfo> sub-elements:

Element	Comments
<merchantID>	Description: Merchant ID. 5 or 7-character merchant ID supplied by SecurePay. Format type: String Format constraints: AN, LEN = 7 Validated by SecurePay: Yes Value: 5-character merchant ID for Direct Entry transactions, eg: "ABC00" 7-character merchant ID for Credit Card transactions, eg: "ABC0001" Sub-elements: No
<password>	Description: Payment password. Password used for authentication of the merchant's request message, supplied by SecurePay. Note: The password can be changed via SecurePay's Merchant Management facility. Format type: String Format constraints: ANS (All characters are allowed), MINLEN = 6, MAXLEN = 20 Validated by SecurePay: Yes Value: Eg: "password_01" Sub-elements: No

RequestType Element

Description:	Defines the type of the request being processed.
Format type:	String
Format constraints:	A, MINLEN = 1, MAXLEN = 20
Validated by SecurePay:	Yes
Value:	One of the following: "Payment" "Echo"
Sub-elements:	No

The following <RequestType> element value must be used for all Transaction messages:

```
<RequestType>Payment</RequestType>
```

Payment Element

Description:	Contains information about financial transactions to be processed.
Format type:	(No value)
Format constraints:	(No value)
Validated by SecurePay:	Yes
Value:	(No value)
Sub-elements:	Yes, see table below

<Payment> sub-elements:

Element	Comments
<TxnList>	See TxnList Element

TxnList Element

Description:	Contains list of transactions to be processed.
Format type:	(No value)
Format constraints:	(No value)
Validated by SecurePay:	Yes
Value:	(No value)
Attributes:	Yes, see table below
Sub-elements:	Yes, see table below

<TxnList> sub-elements:

Element	Comments
<TxnList.count>	Description: Transaction count is an attribute of <TxnList> element and specifies number of <Txn> elements. Note: Currently only single transactions per request are supported. Payments submitted with more than one <Txn> element will be rejected with Status code "577". Format type: Integer Format constraints: DIGNO = 1, MINVAL = 1, MAXVAL = 1 Validated by SecurePay: Yes Value: Currently always "1" Sub-elements: No
<Txn>	See Txn Element

Txn Element

Description:	Contains information about a financial transaction.
Format type:	(No value)
Format constraints:	(No value)
Validated by SecurePay:	Yes
Value:	(No value)
Attributes:	Yes, see table below
Sub-elements:	Yes, see table below

<Txn> sub-elements:

Not all of the <Txn> sub-elements are required for different types of payments. Please refer to section 3.2 Transaction Types Required Elements for information on what elements are required for various payment types.

Element	Comments
<Txn.ID>	Description: Transaction ID is an attribute of <Txn> element and specifies transaction ID. All transactions should be numbered sequentially starting at "1". Note: Currently only single transactions per request are supported. Payments submitted with more than one <Txn> element will be rejected with Status code "577". Format type: Integer Format constraints: DIGNO = 1, MINVAL = 1, MAXVAL = 1 Validated by SecurePay: Yes Value: Currently always "1" Sub-elements: No
<txnType>	Description: Transaction type specifies the type of transaction being processed. Format type: Integer, see Appendix A: Transaction Types Format constraints: DIGNO = 2, MINVAL = 0, MAXVAL = 99 Validated by SecurePay: Yes Value: Eg: "0" Sub-elements: No
<initialAuth>	Description: Identifies a Preauthorisation transaction is for an initial amount. Default value is "no" if omitted from the Preauthorisation request. Only available to merchants in certain industry categories. Allows a Preauthorisation Increase and/or Preauthorisation Cancellation to be performed at a later date when the actual amount is known. Format type: String Format constraints: A, MINLEN = 2, MAXLEN = 3 Validated by SecurePay: Yes Value: "yes" or "no", eg <initialAuth>yes</initialAuth> Sub-elements: No
<txnSource>	Description: Transaction source specifies the source of transaction being processed. For Secure XML the source must always have a value "23". Format type: Integer Format constraints: DIGNO = 2, MINVAL = 0, MAXVAL = 99 Validated by SecurePay: Yes Value: Always "23" Sub-elements: No
<amount>	Description: Transaction amount in cents. Amount must be zero for all Account Verification transactions, and for Preauthorisation Increase transactions where the transaction is for a time extension only. Please note the time extension only applies to Mastercard transactions. Format type: Integer Format constraints: MINVAL = 0 Validated by SecurePay: Yes Value: Eg: "123" for \$1.23 Sub-elements: No

Element	Comments
<currency>	Description: Transaction currency. Note: Only applicable to Credit Card payments. Currency only needs to be set for payment and preauthorisation. Refund, Reversal and Complete transactions are processed in a currency used for the original payment or preauthorisation. If not set for payment or preauthorisation, a default currency is used. Default currency is "AUD" – Australian Dollars. Format type: String Format constraints: A, LEN = 3 Validated by SecurePay: Yes Value: Eg: "AUD" for Australian Dollars Sub-elements: No
<orderId>	Description: 3D Secure 2 (3DS2) Order Id. A unique identifier generated as a reference to the 3DS2 authentication request done for this payment or preauthorisation. Used to retrieve the 3DS2 results that will be included in the payment request submitted to the bank for authorisation. Note: Order ID only needs to be set for payment and preauthorisation. If not set for payment or preauthorisation, this transaction is treated as a transaction not authenticated with 3DS2. Format type: String Format constraints: ANS, ANS (Only hyphen "-" is allowed for special character), LEN=36 Validated by SecurePay: Yes Value: Eg: "f8633973-2b76-4517-93cf-828d789e230a" Sub-elements: No
<liabilityShiftIndicator>	Description: Optional field in the Authorisation and Preauthorisation request that is matched against the Liability Shift Indicator SecurePay has stored from 3DS2 Authentication. When Liability Shift Indicator in Authentication is N and this is not included in the request, the transaction will be declined. When Liability Shift Indicator in Authentication matches the liabilityShiftIndicator value in the authorisation request, then the transaction will proceed. Allows control over whether to proceed with unauthenticated 3DS2 transactions when merchants accept the liability has not shifted to the card issuer. Accepted values, Y or N. <i>To see more details on the field please refer to the 3D Secure 2 Integration guide in the Developer resources on the SecurePay website.</i> Format type: String Format constraints: A, LEN=1 Validated by SecurePay: Yes Value: i.e. "Y" or "N" Sub-elements: No

Element	Comments
<purchaseOrderNo>	Description: Unique merchant transaction identifier, typically an invoice number. Note: Must be the same as <purchaseOrderNo> element of the original transaction when performing a refund, reversal or advice. Format type: String Format constraints: For Credit Card payments ANS (All characters allowed except spaces and "" single quote), For Direct Entry payments EBCDIC (see Appendix I: EBCDIC Character Set), MINLEN = 1, MAXLEN = 60 For Direct Entry payments it is recommended that the purchase order number does not exceed 18 characters in length. Validated by SecurePay: Yes Value: Eg: "order_#000235" Sub-elements: No
<txnID>	Description: Bank transaction ID. Note: Must match the <txnID> element returned in the response to the original payment transaction when performing a refund or reversal. Format type: String Format constraints: AN, MINLEN = 6, MAXLEN = 16 Validated by SecurePay: Yes Value: Eg: "TX123456" Sub-elements: No
<preauthID>	Description: Authorisation code of a preauthorisation transaction. Note: Must match the <preauthID> element returned in the response to the original preauthorisation transaction when performing a Preauthorisation Increase, Preauthorisation Cancellation or Completion transaction advice. Format type: String Format constraints: N, LEN = 6 Validated by SecurePay: Yes Value: Eg: "123456" Sub-elements: No
<CreditCardInfo>	See CreditCardInfo Element
<DirectEntryInfo>	See DirectEntryInfo Element

CreditCardInfo Element

Description: Contains credit card information.

Format type: (No value)

Format constraints: (No value)

Validated by SecurePay: Yes

Value: (No value)

Sub-elements: Yes, see table below

<CreditCardInfo> sub-elements:

Element	Comments
<cardNumber>	Description: Credit card number. Format type: String Format constraints: N, MINLEN = 13, MAXLEN = 16 Validated by SecurePay: Yes Value: Eg: "4242424242424242" Sub-elements: No
<cvv>	Description: Card verification value. The CVV value assists the bank with detecting fraudulent transactions based on automatically generated card numbers, as the CVV number is printed on the physical card and cannot be generated in conjunction with a card number. If passed, the bank may check the supplied value against the value recorded against the card. See Appendix C: Location of CVV Format type: String Format constraints: N, MINLEN = 3, MAXLEN = 4 Validated by SecurePay: Yes Value: Eg: "123" Sub-elements: No
<expiryDate>	Description: Credit card expiry date. Format type: String Format constraints: NS ('/'), LEN = 5 Validated by SecurePay: Yes Value: Eg: "09/26" for May 2026 Sub-elements: No
<cardHolderName>	Description: Used to populate the card holder name in transaction details. Format type: String Format constraints: EBCDIC (see Appendix I: EBCDIC Character Set), MINLEN = 0, MAXLEN = 100 Validated by SecurePay: Yes Value: Eg: John Smith Sub-elements: No

DirectEntryInfo Element

Description:	Contains direct entry information.
Format type:	(No value)
Format constraints:	(No value)
Validated by SecurePay:	Yes
Value:	(No value)
Sub-elements:	Yes, see table below

<DirectEntryInfo> sub-elements:

Element	Comments
<bsbNumber>	Description: BSB number. Format type: String Format constraints: N, LEN = 6 Validated by SecurePay: Yes Value: Eg: "012012" Sub-elements: No
<accountNumber>	Description: Account number. Format type: String Format constraints: N, MINLEN = 1, MAXLEN = 9 Validated by SecurePay: Yes Value: Eg: "00123" Sub-elements: No

<accountName>	Description: Account name. Format type: String Format constraints: EBCDIC (see Appendix I: EBCDIC Character Set), MINLEN = 0, MAXLEN = 32 Validated by SecurePay: Yes Value: Eg: "John Smith" Sub-elements: No
---------------	---

4.3 FraudGuard Request Elements

BuyerInfo Element

Description:	Contains buyer information.
Format type:	(No value)
Format constraints:	(No value)
Validated by SecurePay:	Yes
Value:	(No value)
Sub-elements:	Yes, see table below

<BuyerInfo> sub-elements:

Element	Comments
<ip>	Description: IP address from which the transaction originated. Format type: String Format constraints: NS (Must contain three periods), MAXLEN = 15 Validated by SecurePay: Yes Value: Eg: "203.89.101.20" Sub-elements: No
<zipcode>	Description: Buyer's zip code. Format type: String Format constraints: ANS (All characters allowed), MINLEN = 0, MAXLEN = 30 Validated by SecurePay: Yes Value: Eg: "3000" Sub-elements: No
<town>	Description: The billing or delivery town of the buyer. Format type: String Format constraints: ANS (All characters allowed), MINLEN = 0, MAXLEN = 60 Validated by SecurePay: Yes Value: Eg: "Melbourne" Sub-elements: No
<billingCountry>	Description: Billing country. Can contain the 3 digit numeric ISO code or the 2 or 3 alpha character ISO code. See Appendix K: ISO 3166 Country Codes Format type: String Format constraints: N, LEN = 3 or A, MINLEN = 2, MAXLEN = 3 Validated by SecurePay: Yes Value: Eg: "AU" Sub-elements: No

Element	Comments	
<deliveryCountry>	Description: Delivery country. Can contain the 3 digit numeric ISO code or the 2 or 3 alpha character ISO code. See Appendix K: ISO 3166 Country Codes Format type: String Format constraints: N, LEN = 3 or A, MINLEN = 2, MAXLEN = 3 Validated by SecurePay: Yes Value: Eg: "AU" Sub-elements: No	
<emailAddress>	Description: Email address of the buyer. Format type: String Format constraints: ANS, MAXLEN = 100 Validated by SecurePay: Yes Value: Eg: "johnsmith@somedomain.com" Sub-elements: No	

4.4 Echo Message Elements

Request Messages

Echo requests do not have any additional elements.

*The following <RequestType> element value must be used for all Echo messages:
<RequestType>Echo</RequestType>*

The Echo messages should not be sent more often than every 5 minutes and only if there were no real transactions processed in the last 5 minutes.

Echo URLs

Echo requests can be sent to any of the Payment URLs to verify if the service is available. The Status Code returned in the Echo response will be "000" if the service is up.

5 Response Element Definitions

5.1 XML Header

The XML request will begin with an XML declaration that contains the following data:

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

5.2 Common Response Elements

Responses are the messages sent from SecurePay to the merchant in a response to a request message. Following sections describe elements common to all responses.

MessageInfo Element

Description:	Identifies the message.
Format type:	(No value)
Format constraints:	(No value)
Value:	(No value)
Sub-elements:	Yes, see table below

<MessageInfo> sub-elements:

Element	Comments
<messageID>	Description: Unique identifier for the XML message. Returned unchanged from the request. Format type: String Format constraints: AN, MINLEN = 0, MAXLEN = 30 Value: Eg: "8af793f9af34bea0cf40f5fb5c630c" Sub-elements: No
<messageTimestamp>	Description: Time of the response. Format type: String, see Appendix D: Timestamp String Format Format constraints: NS ('+', '-'), LEN = 24 Value: Eg: "20241803161306527000+660" Sub-elements: No
<apiVersion>	Description: Version of the product used. Returned unchanged from the request. Format type: String Format constraints: ANS ('-', '.'), MINLEN = 1, MAXLEN = 13 Value: Eg: "xml-4.2" Sub-elements: No

MerchantInfo Element

Description:	Identifies the merchant.
Format type:	(No value)
Format constraints:	(No value)
Value:	(No value)
Sub-elements:	Yes, see table below

<MerchantInfo> sub-elements:

Element	Comments
---------	----------

<merchantID>	Description: Merchant ID. 5 or 7-character merchant ID supplied by SecurePay. Returned unchanged from the request. Format type: String Format constraints: AN, LEN = 7 Value: 5-character merchant ID for Direct Entry transactions, eg: "ABC00" 7-character merchant ID for Credit Card transactions, eg: "ABC0001" Sub-elements: No
--------------	---

RequestType Element

Description:	Defines the type of the request being processed. Returned unchanged from the request.
Format type:	String
Format constraints:	A, MINLEN = 1, MAXLEN = 20
Value:	One of the following: "Payment" "Echo"
Sub-elements:	No

Status Element

Description:	Status of the processing of merchant's request.
Format type:	(No value)
Format constraints:	(No value)
Value:	(No value)
Sub-elements:	Yes, see table below

<Status> sub-elements:

Element	Comments
<statusCode>	Description: Status code. Format type: String, see Appendix E: SecurePay Status Codes Format constraints: N, LEN = 3 Value: Eg: "000" Sub-elements: No
<statusDescription>	Description: Status description. Format type: String, see Appendix E: SecurePay Status Codes Format constraints: ANS (All characters are allowed), MINLEN = 0, MAXLEN = 40 Value: Eg: "Normal" Sub-elements: No

5.3 Transaction Response Elements

Following sections describe elements used in Payment requests. The following elements will only be returned if Status received in the response is "000 – Normal".

Payment Element

Description:	Contains information about financial transactions processed.
Format type:	(No value)
Format constraints:	(No value)
Value:	(No value)
Sub-elements:	Yes, see table below

<Payment> sub-elements:

Element	Comments
<TxnList>	See <TxnList> sub-elements.

TxnList Element

Description:	Contains list of transactions processed.
Format type:	(No value)
Format constraints:	(No value)
Value:	(No value)
Attributes:	Yes, see table below
Sub-elements:	Yes, see table below

<TxnList> sub-elements:

Element	Comments
<TxnList.count>	Description: Transaction count is an attribute of <TxnList> element and specifies number of <Txn> elements. Returned unchanged from the request. Note: Currently only single transactions per request are supported. Payments submitted with more than one <Txn> element will be rejected with Status code "577". Format type: Integer Format constraints: DIGNO = 1, MINVAL = 1, MAXVAL = 1 Value: Currently always "1" Sub-elements: No
<Txn>	See <Txn> sub-elements.

Txn Element

Description:	Contains information about a financial transaction.
Format type:	(No value)
Format constraints:	(No value)
Value:	(No value)
Attributes:	Yes, see table below
Sub-elements:	Yes, see table below

<Txn> sub-elements:

Element	Comments
<Txn.ID>	Description: Transaction ID is an attribute of <Txn> element and specifies transaction ID. All transactions returned should be numbered sequentially starting at "1" just as they were in the request message. Returned unchanged from the request. Note: Currently only single transactions per request are supported. Payments submitted with more than one <Txn> element will be rejected with Status code "577". Format type: Integer Format constraints: DIGNO = 1, MINVAL = 1, MAXVAL = 1 Value: Currently always "1" Sub-elements: No
<txnType>	Description: Transaction type specifies the type of transaction processed. Returned unchanged from the request. Format type: Integer, see Appendix A: Transaction Types Format constraints: DIGNO = 2, MINVAL = 0, MAXVAL = 99 Value: Eg: "0" Sub-elements: No

Element	Comments	
<txnSource>	Description: Transaction source specifies the source of transaction processed. Returned unchanged from the request. Format type: Integer Format constraints: DIGNO = 2, MINVAL = 0, MAXVAL = 99 Value: Eg: "23" Sub-elements: No	
<amount>	Description: Transaction amount in cents. Returned unchanged from the request. Format type: Integer Format constraints: MINVAL = 0 Value: Eg: "123" for \$1.23 Sub-elements: No	
<currency>	Description: Transaction currency. Returned unchanged from the request. If not set in the request, a default value of "AUD" is returned. Note: Only applicable to Credit Card payments. Format type: String Format constraints: A, LEN = 3 Validated by SecurePay: Yes Value: Eg: "AUD" for Australian Dollars Sub-elements: No	
<purchaseOrderNo>	Description: Unique merchant transaction identifier, typically an invoice number. For refunds, reversals and advice transactions the purchase order number returned in response is the bank transaction ID of the original transaction. For payments and preauthorise transactions this value is returned unchanged from the request. Format type: String Format constraints: For Credit Card payments ANS (All characters allowed except spaces and "" single quote), For Direct Entry payments EBCDIC (see Appendix I: EBCDIC Character Set), MINLEN = 1, MAXLEN = 60 Value: Eg: "order_#000235" Sub-elements: No	
<approved>	Description: Indicates whether the transaction processed has been approved or not. Format type: String Format constraints: A, MINLEN = 2, MAXLEN = 3 Value: Always "Yes" or "No" Sub-elements: No	
<responseCode>	Description: Response code of the transaction. Either a 2-digit bank response or a 3-digit SecurePay/Gateway response. Element <responseText> provides more information in a textual format. Refer to the SecurePay Response Codes document via the Developer documentation link on the SecurePay website. Format type: String Format constraints: AN, MINLEN = 2, MAXLEN = 3 Value: Eg: "00" Sub-elements: No	
<responseText>	Description: Textual description of the response code received. Format type: String Format constraints: ANS (All characters allowed), MINLEN = 0, MAXLEN = 40 Value: Eg: "Approved" Sub-elements: No	

Element	Comments
<settlementDate>	Description: Bank settlement date when the funds will be settled into the merchant's account. This will be the current date mostly, however after the bank's daily cut-off time, or on non-banking days, the settlement date will be the next business day. Will not be returned if the bank did not receive the transaction. (A settlement date may be returned for declined transactions.) Format type: String Format constraints: N, LEN = 8 Value: Eg: "20040326" for 26 th March 2004 Sub-elements: No
<txnID>	Description: Bank transaction ID. Will not be returned if the transaction has not been processed or in some cases if it was not received by the bank. Format type: String Format constraints: AN, MINLEN = 6, MAXLEN = 16 Value: Eg: "TX123456" Sub-elements: No
<preauthID>	Description: Authorisation code of a preauthorisation transaction. Will not be returned if the transaction is not a Preauthorisation or has not been processed or in some cases if the preauthorisation was not received by the bank. Returned unchanged from the request. Format type: String Format constraints: N, LEN = 6 Value: Eg: "123456" Sub-elements: No
<initialAuth>	Description: Identifies a Preauthorisation transaction is for an initial amount. Format type: String Format constraints: A, MINLEN= 2, MAXLEN = 3 Value: Eg: "Yes" Sub-elements: No
<CreditCardInfo>	See <CreditCardInfo> sub-elements.
<DirectEntryInfo>	See <DirectEntryInfo> sub-elements.

CreditCardInfo Element

Description: Contains credit card information.
Format type: (No value)
Format constraints: (No value)
Value: (No value)
Sub-elements: Yes, see table below

<CreditCardInfo> sub-elements:

Element	Comments
<pan>	Description: Truncated credit card number. Contains first 6 digits of the card number, followed by "..." and then last 3 digits of the card number. Will not be returned for transactions with invalid credit card number. Format type: String Format constraints: N, LEN = 12 Value: Eg: "424242...242" Sub-elements: No

Element	Comments
<expiryDate>	Description: Credit card expiry date. Returned unchanged from the request. Format type: String Format constraints: NS ('/'), LEN = 5 Value: Eg: "05/06" for May 2006 Sub-elements: No
<cardType>	Description: Card type used. Will not be returned for transactions with invalid credit card number. Format type: Integer, see Appendix B: Card Types Format constraints: DIGNO = 1 Value: Eg: "6" for Visa cards Sub-elements: No
<cardDescription>	Description: Card description. Will not be returned for transactions with invalid credit card number. Format type: String, see Appendix B: Card Types Format constraints: A, MINLEN = 0, MAXLEN = 20 Value: Eg: "Visa" Sub-elements: No

DirectEntryInfo Element

Description:	Contains direct entry information.
Format type:	(No value)
Format constraints:	(No value)
Validated by SecurePay:	Yes
Value:	(No value)
Sub-elements:	Yes, see table below

<DirectEntryInfo> sub-elements:

Element	Comments
<bsbNumber>	Description: BSB number. May not be returned for invalid transactions. Format type: String Format constraints: N, LEN = 6 Validated by SecurePay: Yes Value: Eg: "012012" Sub-elements: No
<accountNumber>	Description: Account number. May not be returned for invalid transactions. Format type: String Format constraints: N, MINLEN = 1, MAXLEN = 9 Validated by SecurePay: Yes Value: Eg: "00123" Sub-elements: No
<accountName>	Description: Account name. May not be returned for invalid transactions. Format type: String Format constraints: EBCDIC (see Appendix I: EBCDIC Character Set), MINLEN = 0, MAXLEN = 32 Validated by SecurePay: Yes Value: Eg: "John Smith" Sub-elements: No

5.4 FraudGuard Response Elements

FraudGuard Element

Description:	Contains FraudGuard check information.
Format type:	(No value)
Format constraints:	(No value)
Value:	(No value)
Sub-elements:	Yes, see table below

<FraudGuard> sub-elements:

Element	Comments
<score>	Description: Total of checks performed by FraudGuard. A score greater than or equal to 100 will be declined. Format type: String Format constraints: N, MINLEN = 1, MAXLEN = 3 Value: Eg: "50" Sub-elements: No
<infoIpCountry>	Description: ISO 3166 three letter country code of IP address submitted in request element <IP>. Format type: String Format constraints: A, LEN = 3 Value: Eg: "AUS" Sub-elements: No
<infoCardCountry>	Description: ISO 3166 three letter country code of issuing bank location for credit card number submitted in request element <cardNumber>. Format type: String Format constraints: A, LEN = 3 Value: Eg: "AUS" Sub-elements: No
<ipCountryFail>	Description: Country returned in <infoIpCountry> is blocked in the FraudGuard settings. Element only returned if the FraudGuard rule is triggered. Format type: String Format constraints: A, LEN = 3 Value: Eg: "YES" Sub-elements: No
<minAmountFail>	Description: Amount submitted in request element <amount> is less than the minimum amount set in the FraudGuard settings. Element only returned if the FraudGuard rule is triggered. Format type: String Format constraints: A, LEN = 3 Value: Eg: "YES" Sub-elements: No
<maxAmountFail>	Description: Amount submitted in request element <amount> is more than the maximum amount set in the FraudGuard settings. Element only returned if the FraudGuard rule is triggered. Format type: String Format constraints: A, LEN = 3 Value: Eg: "YES" Sub-elements: No

Element	Comments
<openProxyFail>	Description: IP address submitted in request element <IP> is from a known open proxy. Value returned is the score set in the FraudGuard settings. Element only returned if the FraudGuard rule is triggered. Format type: String Format constraints: N, MINLEN = 1, MAXLEN = 3 Value: Eg: "15" Sub-elements: No
<IpCountryCardCountryFail>	Description: Values for response elements <infoIpCountry> and <infoCardCountry> do not match. Value returned is the score set in the FraudGuard settings. Element only returned if the FraudGuard rule is triggered. Format type: String Format constraints: N, MINLEN = 1, MAXLEN = 3 Value: Eg: "15" Sub-elements: No
<ipCardFail>	Description: Values for response elements <infoIpCountry> and <infoCardCountry> do not match. Value returned is the score set in the FraudGuard settings. Element only returned if the FraudGuard rule is triggered. Format type: String Format constraints: N, MINLEN = 1, MAXLEN = 3 Value: Eg: "20" Sub-elements: No
<ipRiskCountryFail>	Description: IP address submitted in request element <IP> is from a high risk country. Value returned is the score set in the FraudGuard settings. Element only returned if the FraudGuard rule is triggered. Format type: String Format constraints: N, MINLEN = 1, MAXLEN = 3 Value: Eg: "25" Sub-elements: No
<ipBillingFail>	Description: Response element <infoIpCountry> and billing country submitted in request element <billingCountry> do not match. Value returned is the score set in the FraudGuard settings. Element only returned if the FraudGuard rule is triggered. Format type: String Format constraints: N, MINLEN = 1, MAXLEN = 3 Value: Eg: "30" Sub-elements: No
<ipDeliveryFail>	Description: Response element <infoIpCountry> and delivery country submitted in request element <deliveryCountry> do not match. Value returned is the score set in the FraudGuard settings. Element only returned if the FraudGuard rule is triggered. Format type: String Format constraints: N, MINLEN = 1, MAXLEN = 3 Value: Eg: "35" Sub-elements: No

Element	Comments
<billingDeliveryFail>	Description: Values for request elements <billingCountry> and <deliveryCountry> do not match. Value returned is the score set in the FraudGuard settings. Element only returned if the FraudGuard rule is triggered. Format type: String Format constraints: N, MINLEN = 1, MAXLEN = 3 Value: Eg: "40" Sub-elements: No
<freeEmailFail>	Description: Email address submitted in request element <emailAddress> is from a free domain. Value returned is the score set in the FraudGuard settings. Element only returned if the FraudGuard rule is triggered. Format type: String Format constraints: N, MINLEN = 1, MAXLEN = 3 Value: Eg: "45" Sub-elements: No
<tooManySameBank>	Description: Too many transactions from same issuing bank within specified time frame. Value returned is the score set in the FraudGuard settings. Element only returned if the FraudGuard rule is triggered. Format type: String Format constraints: N, MINLEN = 1, MAXLEN = 3 Value: Eg: "10" Sub-elements: No
<tooManyDeclined>	Description: Too many declined transactions from the same IP Address within specified time frame. Value returned is the score set in the FraudGuard settings. Element only returned if the FraudGuard rule is triggered. Format type: String Format constraints: N, MINLEN = 1, MAXLEN = 3 Value: Eg: "15" Sub-elements: No
<tooManySameIp>	Description: Too many transactions from the same IP Address within specified time frame. Value returned is the score set in the FraudGuard settings. Element only returned if the FraudGuard rule is triggered. Format type: String Format constraints: N, MINLEN = 1, MAXLEN = 3 Value: Eg: "20" Sub-elements: No
<tooManySameCard>	Description: Too many transactions from the same full card number within specified time frame. Value returned is the score set in the FraudGuard settings. Element only returned if the FraudGuard rule is triggered. Format type: String Format constraints: N, MINLEN = 1, MAXLEN = 3 Value: Eg: "25" Sub-elements: No

Element	Comments
<lowHighAmount>	Description: Low Amount followed by a high amount from the same card number within specified time frame. Value returned is the score set in the FraudGuard. Element only returned if the FraudGuard rule is triggered. settings Format type: String Format constraints: N, MINLEN = 1, MAXLEN = 3 Value: Eg: "30" Sub-elements: No
<tooManySameEmail>	Description: Too many declined transactions with same customer email within specified time frame. Value returned is the score set in the FraudGuard settings. Element only returned if the FraudGuard rule is triggered. Format type: String Format constraints: N, MINLEN = 1, MAXLEN = 3 Value: Eg: "40" Sub-elements: No

6 Sample XML Request and Response

6.1 Credit Card Payment

Request

```
<?xml version="1.0" encoding="UTF-8"?>
<SecurePayMessage>
  <MessageInfo>
    <messageID>8af793f9af34bea0cf40f5fb750f64</messageID>
    <messageTimestamp>20042303111214383000+660</messageTimestamp>
    <timeoutValue>60</timeoutValue>
    <apiVersion>xml-4.2</apiVersion>
  </MessageInfo>
  <MerchantInfo>
    <merchantID>ABC0001</merchantID>
    <password>abc123</password>
  </MerchantInfo>
  <RequestType>Payment</RequestType>
  <Payment>
    <TxnList count="1">
      <Txn ID="1">
        <txnType>0</txnType>
        <txnSource>23</txnSource>
        <amount>200</amount>
        <recurring>no</recurring>
        <currency>AUD</currency>
        <purchaseOrderNo>test</purchaseOrderNo>
        <CreditCardInfo>
          <cardNumber>4444333322221111</cardNumber>
          <expiryDate>09/23</expiryDate>
          <cvv>000</cvv>
        </CreditCardInfo>
      </Txn>
    </TxnList>
  </Payment>
</SecurePayMessage>
```

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<SecurePayMessage>
  <MessageInfo>
    <messageID>8af793f9af34bea0cf40f5fb750f64</messageID>
    <messageTimestamp>20042303111226938000+660</messageTimestamp>
    <apiVersion>xml-4.2</apiVersion>
  </MessageInfo>
  <MerchantInfo>
    <merchantID>ABC0001</merchantID>
  </MerchantInfo>
  <RequestType>Payment</RequestType>
  <Status>
    <statusCode>000</statusCode>
    <statusDescription>Normal</statusDescription>
  </Status>
  <Payment>
    <TxnList count="1">
      <Txn ID="1">
```

```
<txnType>0</txnType>
<txnSource>23</txnSource>
<amount>200</amount>
<currency>AUD</currency>
<purchaseOrderNo>test</purchaseOrderNo>
<approved>Yes</approved>
<responseCode>00</responseCode>
<responseText>Approved</responseText>
<settlementDate>20040323</settlementDate>
<txnID>009887</txnID>
<CreditCardInfo>
  <pan>444433...111</pan>
  <expiryDate>09/23</expiryDate>
  <cardType>6</cardType>
  <cardDescription>Visa</cardDescription>
</CreditCardInfo>
</Txn>
</TxnList>
</Payment>
</SecurePayMessage>
```

6.2 Credit Card Payment with 3DS2 details

Request

```
<?xml version="1.0" encoding="UTF-8"?>
<SecurePayMessage>
  <MessageInfo>
    <messageID>8af793f9af34bea0cf40f5fb750f64</messageID>
    <messageTimestamp>20042303111214383000+660</messageTimestamp>
    <timeoutValue>60</timeoutValue>
    <apiVersion>xml-4.2</apiVersion>
  </MessageInfo>
  <MerchantInfo>
    <merchantID>ABC0001</merchantID>
    <password>abc123</password>
  </MerchantInfo>
  <RequestType>Payment</RequestType>
  <Payment>
    <TxnList count="1">
      <Txn ID="1">
        <txnType>0</txnType>
        <txnSource>23</txnSource>
        <amount>200</amount>
        <recurring>no</recurring>
        <currency>AUD</currency>
        <purchaseOrderNo>test</purchaseOrderNo>
        <orderId>6c03ads4-23523-4811-8ffp-12321304c</orderId>
        <liabilityShiftIndicator>Y</liabilityShiftIndicator>
        <CreditCardInfo>
          <cardNumber>4444333322221111</cardNumber>
          <expiryDate>09/23</expiryDate>
          <cvv>000</cvv>
        </CreditCardInfo>
      </Txn>
    </TxnList>
  </Payment>
</SecurePayMessage>
```

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<SecurePayMessage>
  <MessageInfo>
    <messageID>8af793f9af34bea0cf40f5fb750f64</messageID>
    <messageTimestamp>20042303111226938000+660</messageTimestamp>
    <apiVersion>xml-4.2</apiVersion>
  </MessageInfo>
  <MerchantInfo>
    <merchantID>ABC0001</merchantID>
  </MerchantInfo>
  <RequestType>Payment</RequestType>
  <Status>
    <statusCode>000</statusCode>
    <statusDescription>Normal</statusDescription>
  </Status>
  <Payment>
    <TxnList count="1">
      <Txn ID="1">
        <txnType>0</txnType>
```

```
<txnSource>23</txnSource>
<amount>200</amount>
<currency>AUD</currency>
<purchaseOrderNo>test</purchaseOrderNo>
<approved>Yes</approved>
<responseCode>00</responseCode>
<responseText>Approved</responseText>
<settlementDate>20040323</settlementDate>
<txnID>009887</txnID>
<CreditCardInfo>
  <pan>444433...111</pan>
  <expiryDate>09/23</expiryDate>
  <cardType>6</cardType>
  <cardDescription>Visa</cardDescription>
</CreditCardInfo>
</Txn>
</TxnList>
</Payment>
</SecurePayMessage>
```

6.3 Credit Card Refund

Request

```
<?xml version="1.0" encoding="UTF-8"?>
<SecurePayMessage>
  <MessageInfo>
    <messageID>8af793f9af34bea0cf40f5fb7510fd</messageID>
    <messageTimestamp>20042303111359163000+660</messageTimestamp>
    <timeoutValue>60</timeoutValue>
    <apiVersion>xml-4.2</apiVersion>
  </MessageInfo>
  <MerchantInfo>
    <merchantID>ABC0001</merchantID>
    <password>abc123</password>
  </MerchantInfo>
  <RequestType>Payment</RequestType>
  <Payment>
    <TxnList count="1">
      <Txn ID="1">
        <txnType>4</txnType>
        <txnSource>23</txnSource>
        <amount>200</amount>
        <purchaseOrderNo>test</purchaseOrderNo>
        <txnID>009887</txnID>
      </Txn>
    </TxnList>
  </Payment>
</SecurePayMessage>
```

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<SecurePayMessage>
  <MessageInfo>
    <messageID>8af793f9af34bea0cf40f5fb7510fd</messageID>
    <messageTimestamp>20042303111409395000+660</messageTimestamp>
    <apiVersion>xml-4.2</apiVersion>
  </MessageInfo>
  <MerchantInfo>
    <merchantID>ABC0001</merchantID>
  </MerchantInfo>
  <RequestType>Payment</RequestType>
  <Status>
    <statusCode>000</statusCode>
    <statusDescription>Normal</statusDescription>
  </Status>
  <Payment>
    <TxnList count="1">
      <Txn ID="1">
        <txnType>4</txnType>
        <txnSource>23</txnSource>
        <amount>200</amount>
        <currency>AUD</currency>
        <purchaseOrderNo>009887</purchaseOrderNo>
        <approved>Yes</approved>
        <responseCode>00</responseCode>
        <responseText>Approved</responseText>
        <settlementDate>20040323</settlementDate>
      </Txn>
    </TxnList>
  </Payment>
</SecurePayMessage>
```

```
<txnID>009890</txnID>
<CreditCardInfo>
  <pan>444433...111</pan>
  <expiryDate>09/23</expiryDate>
  <cardType>6</cardType>
  <cardDescription>Visa</cardDescription>
</CreditCardInfo>
</Txn>
</TxnList>
</Payment>
</SecurePayMessage>
```

6.4 Account Verification

Request

```
<?xml version="1.0" encoding="UTF-8"?>
<SecurePayMessage>
  <MessageInfo>
    <messageID>8af793f9af34bea0cf40f5fb750f64</messageID>
    <messageTimestamp>20042303111214383000+660</messageTimestamp>
    <timeoutValue>60</timeoutValue>
    <apiVersion>xml-4.2</apiVersion>
  </MessageInfo>
  <MerchantInfo>
    <merchantID>ABC0001</merchantID>
    <password>abc123</password>
  </MerchantInfo>
  <RequestType>Payment</RequestType>
  <Payment>
    <TxnList count="1">
      <Txn ID="1">
        <txnType>40</txnType>
        <txnSource>23</txnSource>
        <amount>0</amount>
        <purchaseOrderNo>test</purchaseOrderNo>
        <CreditCardInfo>
          <cardNumber>4444333322221111</cardNumber>
          <expiryDate>09/23</expiryDate>
          <cvv>123</cvv>
        </CreditCardInfo>
      </Txn>
    </TxnList>
  </Payment>
</SecurePayMessage>
```

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<SecurePayMessage>
  <MessageInfo>
    <messageID>8af793f9af34bea0cf40f5fb750f64</messageID>
    <messageTimestamp>20042303111226938000+660</messageTimestamp>
    <apiVersion>xml-4.2</apiVersion>
  </MessageInfo>
  <MerchantInfo>
    <merchantID>ABC0001</merchantID>
  </MerchantInfo>
  <RequestType>Payment</RequestType>
  <Status>
    <statusCode>000</statusCode>
    <statusDescription>Normal</statusDescription>
  </Status>
  <Payment>
    <TxnList count="1">
      <Txn ID="1">
        <txnType>40</txnType>
        <txnSource>23</txnSource>
        <amount>0</amount>
        <purchaseOrderNo>test</purchaseOrderNo>
        <approved>Yes</approved>
      </Txn>
    </TxnList>
  </Payment>
</SecurePayMessage>
```

```
<responseCode>00</responseCode>
<responseText>Approved</responseText>
<settlementDate>20200611</settlementDate>
<txnID>309213</txnID>
<CreditCardInfo>
  <pan>444433...111</pan>
  <expiryDate>09/23</expiryDate>
  <cardType>6</cardType>
  <cardDescription>Visa</cardDescription>
</CreditCardInfo>
</Txn>
</TxnList>
</Payment>
</SecurePayMessage>
```

6.5 Preauthorisation Payment

Request

```
<?xml version="1.0" encoding="UTF-8"?>
<SecurePayMessage>
  <MessageInfo>
    <messageID>8af793f9af34bea0cf40f5fb750f64</messageID>
    <messageTimestamp>20042303111214383000+660</messageTimestamp>
    <timeoutValue>60</timeoutValue>
    <apiVersion>xml-4.2</apiVersion>
  </MessageInfo>
  <MerchantInfo>
    <merchantID>ABC0001</merchantID>
    <password>abc123</password>
  </MerchantInfo>
  <RequestType>Payment</RequestType>
  <Payment>
    <TxnList count="1">
      <Txn ID="1">
        <txnType>10</txnType>
        <txnSource>23</txnSource>
        <amount>100</amount>
        <currency>AUD</currency>
        <purchaseOrderNo>test</purchaseOrderNo>
        <CreditCardInfo>
          <cardNumber>4444333322221111</cardNumber>
          <expiryDate>09/23</expiryDate>
        </CreditCardInfo>
      </Txn>
    </TxnList>
  </Payment>
</SecurePayMessage>
```

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<SecurePayMessage>
  <MessageInfo>
    <messageID>8af793f9af34bea0cf40f5fb750f64</messageID>
    <messageTimestamp>20042303111226938000+660</messageTimestamp>
    <apiVersion>xml-4.2</apiVersion>
  </MessageInfo>
  <MerchantInfo>
    <merchantID>ABC0001</merchantID>
  </MerchantInfo>
  <RequestType>Payment</RequestType>
  <Status>
    <statusCode>000</statusCode>
    <statusDescription>Normal</statusDescription>
  </Status>
  <Payment>
    <TxnList count="1">
      <Txn ID="1">
        <txnType>10</txnType>
        <txnSource>23</txnSource>
        <amount>100</amount>
        <currency>AUD</currency>
```

```
<purchaseOrderNo>test</purchaseOrderNo>
<approved>Yes</approved>
<responseCode>00</responseCode>
<responseText>Approved</responseText>
<settlementDate>20200611</settlementDate>
<txnID>218887</txnID>
<preauthID>477907</preauthID>
<CreditCardInfo>
  <pan>444433...111</pan>
  <expiryDate>09/23</expiryDate>
  <cardType>6</cardType>
  <cardDescription>Visa</cardDescription>
</CreditCardInfo>
</Txn>
</TxnList>
</Payment>
</SecurePayMessage>
```

6.6 Direct Debit

Request

```
<?xml version="1.0" encoding="UTF-8"?>
<SecurePayMessage>
  <MessageInfo>
    <messageID>8af793f9af34bea0cf40f5fb750f64</messageID>
    <messageTimestamp>20042303111214383000+660</messageTimestamp>
    <timeoutValue>60</timeoutValue>
    <apiVersion>xml-4.2</apiVersion>
  </MessageInfo>
  <MerchantInfo>
    <merchantID>ABC00</merchantID>
    <password>abc123</password>
  </MerchantInfo>
  <RequestType>Payment</RequestType>
  <Payment>
    <TxnList count="1">
      <Txn ID="1">
        <txnType>15</txnType>
        <txnSource>23</txnSource>
        <amount>200</amount>
        <purchaseOrderNo>test</purchaseOrderNo>
        <DirectEntryInfo>
          <bsbNumber>123123</bsbNumber>
          <accountNumber>0012345</accountNumber>
          <accountName>John Citizen</accountName>
        </DirectEntryInfo>
      </Txn>
    </TxnList>
  </Payment>
</SecurePayMessage>
```

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<SecurePayMessage>
  <MessageInfo>
    <messageID>8af793f9af34bea0cf40f5fb750f64</messageID>
    <messageTimestamp>20042303111226938000+660</messageTimestamp>
    <apiVersion>xml-4.2</apiVersion>
  </MessageInfo>
  <MerchantInfo>
    <merchantID>ABC00</merchantID>
  </MerchantInfo>
  <RequestType>Payment</RequestType>
  <Status>
    <statusCode>000</statusCode>
    <statusDescription>Normal</statusDescription>
  </Status>
  <Payment>
    <TxnList count="1">
      <Txn ID="1">
        <txnType>15</txnType>
        <txnSource>23</txnSource>
        <amount>200</amount>
        <purchaseOrderNo>test</purchaseOrderNo>
      </Txn>
    </TxnList>
  </Payment>
</SecurePayMessage>
```

```
<approved>Yes</approved>
<responseCode>00</responseCode>
<responseText>Transaction Accepted</responseText>
<settlementDate>20040323</settlementDate>
<txnID>009887</txnID>
<DirectEntryInfo>
  <bsbNumber>123123</bsbNumber>
  <accountNumber>0012345</accountNumber>
  <accountName>John Citizen</accountName>
</DirectEntryInfo>
</Txn>
</TxnList>
</Payment>
</SecurePayMessage>
```

6.7 Card Payment with FraudGuard

Request

```
<?xml version="1.0" encoding="UTF-8"?>
<SecurePayMessage>
  <MessageInfo>
    <messageID>8af793f9af34bea0cf40f5fb5c630c</messageID>
    <messageTimestamp>20152303111359163000+660</messageTimestamp>
    <timeoutValue>60</timeoutValue>
    <apiVersion>xml-4.2</apiVersion>
  </MessageInfo>
  <MerchantInfo>
    <merchantID>ABC0001</merchantID>
    <password>abc123</password>
  </MerchantInfo>
  <RequestType>Payment</RequestType>
  <Payment>
    <TxnList count="1">
      <Txn ID="1">
        <txnType>21</txnType>
        <txnSource>23</txnSource>
        <amount>100</amount>
        <currency>AUD</currency>
        <purchaseOrderNo>test</purchaseOrderNo>
        <CreditCardInfo>
          <cardNumber>4444333322221111</cardNumber>
          <expiryDate>09/23</expiryDate>
          <cvv>000</cvv>
        </CreditCardInfo>
        <BuyerInfo>
          <firstName>John</firstName>
          <lastName>Smith</lastName>
          <zipCode>000</zipCode>
          <town>Melbourne</town>
          <billingCountry>GBR</billingCountry>
          <deliveryCountry>NZL</deliveryCountry>
          <emailAddress>test@hotmail.com</emailAddress>
          <ip>203.89.255.137</ip>
        </BuyerInfo>
      </Txn>
    </TxnList>
  </Payment>
</SecurePayMessage>
```

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<SecurePayMessage>
  <MessageInfo>
    <messageID>8af793f9af34bea0cf40f5fb5c630c</messageID>
    <messageTimestamp>20152303111359163000+660</messageTimestamp>
    <apiVersion>xml-4.2</apiVersion>
  </MessageInfo>
  <MerchantInfo>
    <merchantID>ABC0001</merchantID>
  </MerchantInfo>
  <RequestType>Payment</RequestType>
  <Status>
```

```
<statusCode>000</statusCode>
<statusDescription>Normal</statusDescription>
</Status>
<Payment>
  <TxnList count="1">
    <Txn ID="1">
      <txnType>21</txnType>
      <txnSource>0</txnSource>
      <amount>100</amount>
      <currency>AUD</currency>
      <purchaseOrderNo>test</purchaseOrderNo>
      <approved>Yes</approved>
      <responseCode>00</responseCode>
      <responseText>Approved</responseText>
      <settlementDate>20040318</settlementDate>
      <txnID>009844</txnID>
      <CreditCardInfo>
        <pan>444433...111</pan>
        <expiryDate>09/23</expiryDate>
        <cardType>6</cardType>
        <cardDescription>Visa</cardDescription>
      </CreditCardInfo>
      <antiFraudResponseCode>000</antiFraudResponseCode>
      <antiFraudResponseText>Antifraud check passed</antiFraudResponseText>
      <FraudGuard>
        <score>85</score>
        <infoIpCountry>AUD</infoIpCountry>
        <infoCardCountry>NZL</infoCardCountry>
        <ipCountryFail>yes</ipCountryFail>
        <minAmountFail>yes</minAmountFail>
        <maxAmountFail>yes</maxAmountFail>
        <openProxyFail>5</openProxyFail>
        <IpCountryCardCountryFail>5</IpCountryCardCountryFail>
        <ipCardFail>5</ipCardFail>
        <ipRiskCountryFail>5</ipRiskCountryFail>
        <ipBillingFail>5</ipBillingFail>
        <ipDeliveryFail>5</ipDeliveryFail>
        <billingDeliveryFail>5</billingDeliveryFail>
        <freeEmailFail>5</freeEmailFail>
        <tooManySameBank>5</tooManySameBank>
        <tooManyDeclined>5</tooManyDeclined>
        <tooManySameIp>5</tooManySameIp>
        <tooManySameCard>5</tooManySameCard>
        <lowHighAmount>5</lowHighAmount>
        <tooManySameEmail>5</tooManySameEmail>
      </FraudGuard>
      <ThirdPartyResponse>
        <returnCode>0</returnCode>
        <result1>1</result1>
        <result2>1</result2>
        <additionalInfo1 />
        <additionalInfo2 />
        <PSPResult>1</PSPResult>
        <PSPScore>100</PSPScore>
        <MerchantResult>1</MerchantResult>
        <MerchantScore>100</MerchantScore>
        <ProxyIp />
        <FreeE-MailDomain />
        <IPCountry>AUS</IPCountry>
      </ThirdPartyResponse>
    </Txn ID="1">
  </TxnList count="1">
</Payment>
```

```

    <BINCountry>NZL</BINCountry>
    <Geo-RegionMatch>1</Geo-RegionMatch>
    <Geo-CountryMatch>1</Geo-CountryMatch>
  </ThirdPartyResponse>
</Txn>
</TxnList>
</Payment>
</SecurePayMessage>

```

6.8 FraudGuard Only Request

Request

```

<?xml version="1.0" encoding="UTF-8"?>
<SecurePayMessage>
  <MessageInfo>
    <messageID>8af793f9af34bea0cf40f5fb5c630c</messageID>
    <messageTimestamp>20152303111359163000+660</messageTimestamp>
    <timeoutValue>60</timeoutValue>
    <apiVersion>xml-4.2</apiVersion>
  </MessageInfo>
  <MerchantInfo>
    <merchantID>ABC0001</merchantID>
    <password>abc123</password>
  </MerchantInfo>
  <RequestType>Payment</RequestType>
  <Payment>
    <TxnList count="1">
      <Txn ID="1">
        <txnType>22</txnType>
        <txnSource>23</txnSource>
        <amount>100</amount>
        <currency>AUD</currency>
        <purchaseOrderNo>test</purchaseOrderNo>
        <CreditCardInfo>
          <cardNumber>4444333322221111</cardNumber>
          <expiryDate>09/23</expiryDate>
          <cvv>000</cvv>
        </CreditCardInfo>
        <BuyerInfo>
          <firstName>John</firstName>
          <lastName>Smith</lastName>
          <zipCode>000</zipCode>
          <town>Melbourne</town>
          <billingCountry>GBR</billingCountry>
          <deliveryCountry>NZL</deliveryCountry>
          <emailAddress>test@hotmail.com</emailAddress>
          <ip>203.89.255.137</ip>
        </BuyerInfo>
      </Txn>
    </TxnList>
  </Payment>
</SecurePayMessage>

```

Response

```

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

```

```
<messageID>8af793f9af34bea0cf40f5fb5c630c</messageID>
<messageTimestamp>20152303111359163000+660</messageTimestamp>
<apiVersion>xml-4.2</apiVersion>
</MessageInfo>
<MerchantInfo>
  <merchantID>ABC0001</merchantID>
</MerchantInfo>
<RequestType>Payment</RequestType>
<Status>
  <statusCode>000</statusCode>
  <statusDescription>Normal</statusDescription>
</Status>
<Payment>
  <TxnList count="1">
    <Txn ID="1">
      <txnType>22</txnType>
      <txnSource>0</txnSource>
      <amount>100</amount>
      <currency>AUD</currency>
      <purchaseOrderNo>test</purchaseOrderNo>
      <approved>Yes</approved>
      <responseCode />
      <responseText />
      <settlementDate />
      <txnID />
      <CreditCardInfo>
        <pan>444433...111</pan>
        <expiryDate>09/23</expiryDate>
        <cardType>6</cardType>
        <cardDescription>Visa</cardDescription>
      </CreditCardInfo>
      <antiFraudResponseCode>000</antiFraudResponseCode>
      <antiFraudResponseText>Antifraud check passed</antiFraudResponseText>
      <FraudGuard>
        <score>85</score>
        <infoIpCountry>AUD</infoIpCountry>
        <infoCardCountry>NZL</infoCardCountry>
        <ipCountryFail>yes</ipCountryFail>
        <minAmountFail>yes</minAmountFail>
        <maxAmountFail>yes</maxAmountFail>
        <openProxyFail>5</openProxyFail>
        <IpCountryCardCountryFail>5</IpCountryCardCountryFail>
        <ipCardFail>5</ipCardFail>
        <ipRiskCountryFail>5</ipRiskCountryFail>
        <ipBillingFail>5</ipBillingFail>
        <ipDeliveryFail>5</ipDeliveryFail>
        <billingDeliveryFail>5</billingDeliveryFail>
        <freeEmailFail>5</freeEmailFail>
        <tooManySameBank>5</tooManySameBank>
        <tooManyDeclined>5</tooManyDeclined>
        <tooManySameIp>5</tooManySameIp>
        <tooManySameCard>5</tooManySameCard>
        <lowHighAmount>5</lowHighAmount>
        <tooManySameEmail>5</tooManySameEmail>
      </FraudGuard>
      <ThirdPartyResponse>
        <returnCode>0</returnCode>
        <result1>1</result1>
        <result2>1</result2>
      </ThirdPartyResponse>
    </Txn ID="1">
  </TxnList count="1">

```

```

    <additionalInfo1 />
    <additionalInfo2 />
    <PSPResult>1</PSPResult>
    <PSPScore>100</PSPScore>
    <MerchantResult>1</MerchantResult>
    <MerchantScore>100</MerchantScore>
    <ProxyIp />
    <FreeE-MailDomain />
    <IPCountry>AUS</IPCountry>
    <BINCountry>NZL</BINCountry>
    <Geo-RegionMatch>1</Geo-RegionMatch>
    <Geo-CountryMatch>1</Geo-CountryMatch>
  </ThirdPartyResponse>
</Txn>
</TxnList>
</Payment>
</SecurePayMessage>

```

6.9 Echo

Request

```

<?xml version="1.0" encoding="UTF-8"?>
<SecurePayMessage>
  <MessageInfo>
    <messageID>8af793f9af34bea0cf40f5fb79f383</messageID>
    <messageTimestamp>20042403095953349000+660</messageTimestamp>
    <timeoutValue>60</timeoutValue>
    <apiVersion>xml-4.2</apiVersion>
  </MessageInfo>
  <MerchantInfo>
    <merchantID>ABC0001</merchantID>
    <password>abc123</password>
  </MerchantInfo>
  <RequestType>Echo</RequestType>
</SecurePayMessage>

```

Response

```

<?xml version="1.0" encoding="UTF-8"?>
<SecurePayMessage>
  <MessageInfo>
    <messageID>8af793f9af34bea0cf40f5fb79f383</messageID>
    <messageTimestamp>20042403095956732000+660</messageTimestamp>
    <apiVersion>xml-4.2</apiVersion>
  </MessageInfo>
  <RequestType>Echo</RequestType>
  <Status>
    <statusCode>000</statusCode>
    <statusDescription>Normal</statusDescription>
  </Status>
</SecurePayMessage>

```

Appendix A: Transaction Types

Transaction type codes define the type of financial transaction processed by SecurePay.

Code	Description
0	Standard Payment
4	Refund
10	Preauthorise
11	Preauthorise Complete
14	Recurring Payment (Deprecated)
15	Direct Debit
17	Direct Credit
21	FraudGuard Payment (Deprecated)
22	FraudGuard Only (No Transaction)
40	Account Verification
41	Preauthorisation Increase
42	Preauthorisation Cancellation

Appendix B: Card Types

SecurePay uses numeric codes to refer to credit card types in our system.

Code	Description
0	Unknown
1	JCB
2	American Express (Amex)
3	Diners Club
4	Bankcard
5	MasterCard
6	Visa

Appendix C: Location of CVV

The Card Verification Value is an anti-fraud measure used by some banks to prevent payments from generated card numbers. The CVV number is printed on the physical card, and is randomly assigned, therefore cannot be auto-generated.

The CVV number can be found in the following places:

Card Type	Location
Visa	Signature strip on back of card. Last digits of card number are re-printed in reverse italics, followed by 3-digit CVV.
MasterCard	Signature strip on back of card. Last digits of card number are re-printed in reverse italics, followed by 3-digit CVV.
Bankcard	Signature strip on back of card. Last digits of card number are re-printed in reverse italics, followed by 3-digit CVV.
Amex	4 digit CVV above card number on front of card.
Diners Club	Signature strip on back of card. Last digits of card number are re-printed in reverse italics, followed by 3-digit CVV.
JCB	Not used

Appendix D: Timestamp String Format

The format of the Timestamp or Log Time strings returned by SecurePay XMLAPI is:

YYYYDDMMHHNNSSKKK000s000

Where:

- YYYY is a 4-digit year
- DD is a 2-digit zero-padded day of month
- MM is a 2-digit zero-padded month of year (January = 01)
- HH is a 2-digit zero-padded hour of day in 24-hour clock format (midnight =0)
- NN is a 2-digit zero-padded minute of hour
- SS is a 2-digit zero-padded second of minute
- KKK is a 3-digit zero-padded millisecond of second
- 000 is a Static 0 characters, as SecurePay does not store nanoseconds
- s000 is a Time zone offset, where s is "+" or "-", and 000 = minutes, from GMT.

E.g. June 24, 2002 5:12:16.789 PM, Australian EST is:

20022406171216789000+600

Appendix E: SecurePay Status Codes

For a full list of SecurePay's response codes, please refer to the SecurePay Response Codes document via the Developer Documentation page on the SecurePay website.

Appendix F: XML Request DTD

```
<!-- ELEMENT SecurePayMessage (MessageInfo, MerchantInfo, RequestType
      Payment?)>

<!-- define elements for SecurePayMessage -->
<!-- ELEMENT MessageInfo (messageID, messageTimestamp, timeoutValue, apiVersion)>
<!-- ELEMENT MerchantInfo (merchantID, password)>
<!-- ELEMENT RequestType (#PCDATA)>
<!-- ELEMENT Payment (TxnList)>

<!-- define elements for MessageInfo -->
<!-- ELEMENT messageID (#PCDATA)>
<!-- ELEMENT messageTimestamp (#PCDATA)>
<!-- ELEMENT timeoutValue (#PCDATA)>
<!-- ELEMENT apiVersion (#PCDATA)>

<!-- define elements for MerchantInfo -->
<!-- ELEMENT merchantID (#PCDATA)>
<!-- ELEMENT password (#PCDATA)>

<!-- define elements for Payment -->
<!-- ELEMENT TxnList (Txn)>
<!-- ATTLIST TxnList
      count CDATA #REQUIRED>

<!-- define elements for TxnList -->
<!-- ELEMENT Txn (txnType, txnSource, amount, currency, purchaseOrderNo, txnID?,
      preauthID?, initialAuth?, CreditCardInfo)>
<!-- ATTLIST Txn
      ID CDATA #REQUIRED>

<!-- define elements for Txn -->
<!-- ELEMENT txnType (#PCDATA)>
<!-- ELEMENT txnSource (#PCDATA)>
<!-- ELEMENT amount (#PCDATA)>
<!-- ELEMENT currency (#PCDATA)>
<!-- ELEMENT purchaseOrderNo (#PCDATA)>
<!-- ELEMENT recurring (#PCDATA)>
<!-- ELEMENT txnID (#PCDATA)>
<!-- ELEMENT preauthID (#PCDATA)>
<!-- ELEMENT initialAuth (#PCDATA)>
<!-- ELEMENT CreditCardInfo (cardNumber, cvv?, expiryDate?)>
<!-- ELEMENT DirectEntryInfo (bsbNumber, accountNumber, accountName)>

<!-- define elements for CreditCardInfo -->
<!-- ELEMENT cardNumber (#PCDATA)>
<!-- ELEMENT cvv (#PCDATA)>
<!-- ELEMENT expiryDate (#PCDATA)>
<!-- ELEMENT cardHolderName (#PCDATA)>

<!-- define elements for DirectEntryInfo -->
<!-- ELEMENT bsbNumber (#PCDATA)>
<!-- ELEMENT accountNumber (#PCDATA)>
<!-- ELEMENT accountName (#PCDATA)>

<!-- define elements for BuyerInfo -->
<!-- ELEMENT zipCode (#PCDATA)>
```

```
<!ELEMENT town (#PCDATA)>  
<!ELEMENT billingCountry (#PCDATA)>  
<!ELEMENT deliveryCountry (#PCDATA)>  
<!ELEMENT emailAddress (#PCDATA)>  
<!ELEMENT ip (#PCDATA)>
```

Appendix G: XML Response DTD

```
<!-- define elements for SecurePayMessage -->
<ELEMENT SecurePayMessage (MessageInfo, MerchantInfo, RequestType,
    Status, Payment)>

<!-- define elements for MessageInfo -->
<ELEMENT MessageInfo (messageID, messageTimestamp, apiVersion)>
<ELEMENT MerchantInfo (merchantID)>
<ELEMENT RequestType (#PCDATA)>
<ELEMENT Status (statusCode, statusDescription)>
<ELEMENT Payment (TxnList)>

<!-- define elements for MessageInfo -->
<ELEMENT messageID (#PCDATA)>
<ELEMENT messageTimestamp (#PCDATA)>
<ELEMENT apiVersion (#PCDATA)>

<!-- define elements for MerchantInfo -->
<ELEMENT merchantID (#PCDATA)>

<!-- define elements for Status -->
<ELEMENT statusCode (#PCDATA)>
<ELEMENT statusDescription (#PCDATA)>

<!-- define elements for Payment -->
<ELEMENT TxnList (Txn*)>
<ATTLIST TxnList
    count CDATA #REQUIRED>

<!-- define elements for TxnList -->
<ELEMENT Txn (txnType, txnSource, amount, currency, purchaseOrderNo,
    approved, responseCode, responseText, settlementDate,
    txnID, preauthID?, initialAuth?, CreditCardInfo)>
<ATTLIST Txn
    ID CDATA #REQUIRED>

<!-- define elements for Txn -->
<ELEMENT txnType (#PCDATA)>
<ELEMENT txnSource (#PCDATA)>
<ELEMENT amount (#PCDATA)>
<ELEMENT currency (#PCDATA)>
<ELEMENT purchaseOrderNo (#PCDATA)>
<ELEMENT approved (#PCDATA)>
<ELEMENT responseCode (#PCDATA)>
<ELEMENT responseText (#PCDATA)>
<ELEMENT settlementDate (#PCDATA)>
<ELEMENT txnID (#PCDATA)>
<ELEMENT preauthID (#PCDATA)>
<ELEMENT initialAuth (#PCDATA)>
<ELEMENT CreditCardInfo (pan, expiryDate?, cardType?, cardDescription?)>
<ELEMENT DirectEntryInfo (bsbNumber, accountNumber, accountName)>
<ELEMENT antiFraudResponseCode (#PCDATA)>
<ELEMENT antiFraudResponseText (#PCDATA)>
<ELEMENT FraudGuard (score, infoIpCountry, infoCardCountry, ipCountryFail,
    minAmountFail, maxAmountFail, openProxyFail, IpCountryCardCountryFail,
    ipCardFail, ipRiskCountryFail, ipBillingFail, ipDeliveryFail,
    billingDeliveryFail, freeEmailFail, tooManySameBank, tooManyDeclined,
    tooManySameIp, tooManySameCard, lowHighAmount, tooManySameEmail)>
```

```
<!ELEMENT ThirdPartyResponse (returnCode, result1, result2, additionalInfo1,
additionalInfo2, PSPResult, PSPScore, MerchantResult, MerchantScore, ProxyIp,
FreeE-MailDomain, IPCountry, BINCountry, Geo-RegionMatch, Geo-CountryMatch)>
```

```
<!-- define elements for CreditCardInfo -->
```

```
<!ELEMENT pan (#PCDATA)>
<!ELEMENT expiryDate (#PCDATA)>
<!ELEMENT cardType (#PCDATA)>
<!ELEMENT cardDescription (#PCDATA)>
```

```
<!-- define elements for DirectEntryInfo -->
```

```
<!ELEMENT bsbNumber (#PCDATA)>
<!ELEMENT accountNumber (#PCDATA)>
<!ELEMENT accountName (#PCDATA)>
```

```
<!-- define elements for FraudGuard -->
```

```
<!ELEMENT score (#PCDATA)>
<!ELEMENT infoIpCountry (#PCDATA)>
<!ELEMENT infoCardCountry (#PCDATA)>
<!ELEMENT ipCountryFail (#PCDATA)>
<!ELEMENT minAmountFail (#PCDATA)>
<!ELEMENT maxAmountFail (#PCDATA)>
<!ELEMENT openProxyFail (#PCDATA)>
<!ELEMENT IpCountryCardCountryFail (#PCDATA)>
<!ELEMENT ipCardFail (#PCDATA)>
<!ELEMENT ipRiskCountryFail (#PCDATA)>
<!ELEMENT ipBillingFail (#PCDATA)>
<!ELEMENT ipDeliveryFail (#PCDATA)>
<!ELEMENT billingDeliveryFail (#PCDATA)>
<!ELEMENT freeEmailFail (#PCDATA)>
<!ELEMENT tooManySameBank (#PCDATA)>
<!ELEMENT tooManyDeclined (#PCDATA)>
<!ELEMENT tooManySameIp (#PCDATA)>
<!ELEMENT tooManySameCard (#PCDATA)>
<!ELEMENT lowHighAmount (#PCDATA)>
<!ELEMENT tooManySameEmail (#PCDATA)>
```

```
<!-- define elements for ThirdPartyResponse -->
```

```
<!ELEMENT returnCode (#PCDATA)>
<!ELEMENT result1 (#PCDATA)>
<!ELEMENT result2 (#PCDATA)>
<!ELEMENT additionalInfo1 (#PCDATA)>
<!ELEMENT additionalInfo2 (#PCDATA)>
<!ELEMENT PSPResult (#PCDATA)>
<!ELEMENT PSPScore (#PCDATA)>
<!ELEMENT MerchantResult (#PCDATA)>
<!ELEMENT MerchantScore (#PCDATA)>
<!ELEMENT ProxyIp (#PCDATA)>
<!ELEMENT FreeE-MailDomain (#PCDATA)>
<!ELEMENT IPCountry (#PCDATA)>
<!ELEMENT BINCountry (#PCDATA)>
<!ELEMENT Geo-RegionMatch (#PCDATA)>
<!ELEMENT Geo-CountryMatch (#PCDATA)>
```

Appendix H: Currency Codes List

Your SecurePay account **must be enabled for multi-currency** before using this feature. Please contact SecurePay Support or your SecurePay Account Manager regarding multi-currency support for your account.

Code	Description	Minor Units	Example*	
			Amount	Pass As
AUD	Australian Dollar	2	\$20	2000
CAD	Canadian Dollar	2	\$20	2000
CHF	Swiss Franc	2	20	2000
EUR	Euro	2	€20	2000
GBP	English Pound	2	£20	2000
HKD	Hong Kong Dollar	2	\$20	2000
JPY	Japanese Yen	0	¥20	20
NZD	New Zealand Dollar	2	\$20	2000
SGD	Singapore Dollar	2	\$20	2000
USD	US Dollar	2	\$20	2000

* To pass a multicurrency payment to SecurePay, set `<currency>` field with the value from the Code column, and set `<amount>` field with the amount to be charged, ensuring you set the correct number of Minor Units for the selected currency, as shown in the examples.

E.g. For US Dollars, \$4,125.90 is set using:

```
<amount>412590</amount>
<currency>USD</currency>
```

or for Japanese Yen, ¥67,925 is set using:

```
<amount>67925</amount>
<currency>JPY</currency>
```

Appendix I: EBCDIC Character Set

Description	Characters allowed
Numeric	0 - 9
Alphabetic	a - z, A - Z
Oblique slash	/
Hyphen	-
Ampersand	&
Period	.
Asterisk	*
Apostrophe	'
Blank space	

Appendix K: ISO 3166 Country Codes

Country	Alpha 2 code	Alpha 3 code
AFGHANISTAN	AF	AFG
ALAND ISLANDS	AX	ALA
ALBANIA	AL	ALB
ALGERIA	DZ	DZA
AMERICAN SAMOA	AS	ASM
ANDORRA	AD	AND
ANGOLA	AO	AGO
ANGUILLA	AI	AIA
ANTARCTICA	AQ	ATA
ANTIGUA AND BARBUDA	AG	ATG
ARGENTINA	AR	ARG
ARMENIA	AM	ARM
ARUBA	AW	ABW
AUSTRALIA	AU	AUS
AUSTRIA	AT	AUT
AZERBAIJAN	AZ	AZE
BAHAMAS	BS	BHS
BAHRAIN	BH	BHR
BANGLADESH	BD	BGD
BARBADOS	BB	BRB
BELARUS	BY	BLR
BELGIUM	BE	BEL
BELIZE	BZ	BLZ
BENIN	BJ	BEN
BERMUDA	BM	BMU
BHUTAN	BT	BTN

Country	Alpha 2 code	Alpha 3 code
BOLIVIA, PLURINATIONAL STATE OF	BO	BOL
BONAIRE, SAINT EUSTATIUS AND SABA	BQ	BES
BOSNIA AND HERZEGOVINA	BA	BIH
BOTSWANA	BW	BWA
BOUVET ISLAND	BV	BVT
BRAZIL	BR	BRA
BRITISH INDIAN OCEAN TERRITORY	IO	IOT
BRUNEI DARUSSALAM	BN	BRN
BULGARIA	BG	BGR
BURKINA FASO	BF	BFA
BURUNDI	BI	BDI
CAMBODIA	KH	KHM
CAMEROON	CM	CMR
CANADA	CA	CAN
CAPE VERDE	CV	CPV
CAYMAN ISLANDS	KY	CYM
CENTRAL AFRICAN REPUBLIC	CF	CAF
CHAD	TD	TCD
CHILE	CL	CHL
CHINA	CN	CHN
CHRISTMAS ISLAND	CX	CXR
COCOS (KEELING) ISLANDS	CC	CCK
COLOMBIA	CO	COL
COMOROS	KM	COM
CONGO	CG	COG
CONGO, THE DEMOCRATIC REPUBLIC OF THE	CD	COD
COOK ISLANDS	CK	COK
COSTA RICA	CR	CRI

Country	Alpha 2 code	Alpha 3 code
COTE D'IVOIRE	CI	CIV
CROATIA	HR	HRV
CUBA	CU	CUB
CURACAO	CW	CUW
CYPRUS	CY	CYP
CZECH REPUBLIC	CZ	CZE
DENMARK	DK	DNK
DJIBOUTI	DJ	DJI
DOMINICA	DM	DMA
DOMINICAN REPUBLIC	DO	DOM
ECUADOR	EC	ECU
EGYPT	EG	EGY
EL SALVADOR	SV	SLV
EQUATORIAL GUINEA	GQ	GNQ
ERITREA	ER	ERI
ESTONIA	EE	EST
ETHIOPIA	ET	ETH
FALKLAND ISLANDS (MALVINAS)	FK	FLK
FAROE ISLANDS	FO	FRO
FIJI	FJ	FJI
FINLAND	FI	FIN
FRANCE	FR	FRA
FRENCH GUIANA	GF	GUF
FRENCH POLYNESIA	PF	PYF
FRENCH SOUTHERN TERRITORIES	TF	ATF
GABON	GA	GAB
GAMBIA	GM	GMB
GEORGIA	GE	GEO

Country	Alpha 2 code	Alpha 3 code
GERMANY	DE	DEU
GHANA	GH	GHA
GIBRALTAR	GI	GIB
GREECE	GR	GRC
GREENLAND	GL	GRL
GRENADA	GD	GRD
GUADELOUPE	GP	GLP
GUAM	GU	GUM
GUATEMALA	GT	GTM
GUERNSEY	GG	GGY
GUINEA	GN	GIN
GUINEA-BISSAU	GW	GNB
GUYANA	GY	GUY
HAITI	HT	HTI
HEARD ISLAND AND MCDONALD ISLANDS	HM	HMD
HOLY SEE (VATICAN CITY STATE)	VA	VAT
HONDURAS	HN	HND
HONG KONG	HK	HKG
HUNGARY	HU	HUN
ICELAND	IS	ISL
INDIA	IN	IND
INDONESIA	ID	IDN
IRAN (ISLAMIC REPUBLIC OF)	IR	IRN
IRAQ	IQ	IRQ
IRELAND	IE	IRL
ISLE OF MAN	IM	IMM
ISRAEL	IL	ISR
ITALY	IT	ITA

Country	Alpha 2 code	Alpha 3 code
JAMAICA	JM	JAM
JAPAN	JP	JPN
JERSEY	JE	JEY
JORDAN	JO	JOR
KAZAKHSTAN	KZ	KAZ
KENYA	KE	KEN
KIRIBATI	KI	KIR
KOREA, DEMOCRATIC PEOPLE'S REPUBLIC OF	KP	PRK
KOREA, REPUBLIC OF	KR	KOR
KUWAIT	KW	KWT
KYRGYZSTAN	KG	KGZ
LAO PEOPLE'S DEMOCRATIC REPUBLIC	LA	LAO
LATVIA	LV	LVA
LEBANON	LB	LBN
LESOTHO	LS	LSO
LIBERIA	LR	LBR
LIBYAN ARAB JAMAHIRIYA	LY	LBY
LIECHTENSTEIN	LI	LIE
LITHUANIA	LT	LTU
LUXEMBOURG	LU	LUX
MACAO	MO	MAC
MACEDONIA, THE FORMER YUGOSLAV REPUBLIC OF	MK	MKD
MADAGASCAR	MG	MDG
MALAWI	MW	MWI
MALAYSIA	MY	MYS
MALDIVES	MV	MDV
MALI	ML	MLI

Country	Alpha 2 code	Alpha 3 code
MALTA	MT	MLT
MARSHALL ISLANDS	MH	MHL
MARTINIQUE	MQ	MTQ
MAURITANIA	MR	MRT
MAURITIUS	MU	MUS
MAYOTTE	YT	MYT
MEXICO	MX	MEX
MICRONESIA, FEDERATED STATES OF	FM	FSM
MOLDOVA, REPUBLIC OF	MD	MDA
MONACO	MC	MCO
MONGOLIA	MN	MNG
MONTENEGRO	ME	MNE
MONTSERRAT	MS	MSR
MOROCCO	MA	MAR
MOZAMBIQUE	MZ	MOZ
MYANMAR	MM	MMR
NAMIBIA	NA	NAM
NAURU	NR	NRU
NEPAL	NP	NPL
NETHERLANDS	NL	NLD
NEW CALEDONIA	NC	NCL
NEW ZEALAND	NZ	NZL
NICARAGUA	NI	NIC
NIGER	NE	NER
NIGERIA	NG	NGA
NIUE	NU	NIU
NORFOLK ISLAND	NF	NFK
NORTHERN MARIANA ISLANDS	MP	MNP

Country	Alpha 2 code	Alpha 3 code
NORWAY	NO	NOR
OMAN	OM	OMN
PAKISTAN	PK	PAK
PALAU	PW	PLW
PALESTINIAN TERRITORY, OCCUPIED	PS	PSE
PANAMA	PA	PAN
PAPUA NEW GUINEA	PG	PNG
PARAGUAY	PY	PRY
PERU	PE	PER
PHILIPPINES	PH	PHL
PITCAIRN	PN	PCN
POLAND	PL	POL
PORTUGAL	PT	PRT
PUERTO RICO	PR	PRI
QATAR	QA	QAT
REUNION	RE	REU
ROMANIA	RO	ROU
RUSSIAN FEDERATION	RU	RUS
RWANDA	RW	RWA
SAINT BARTHELEMY	BL	BLM
SAINT HELENA, ASCENSION AND TRISTAN DA CUNHA	SH	SHN
SAINT KITTS AND NEVIS	KN	KNA
SAINT LUCIA	LC	LCA
SAINT MARTIN (FRENCH PART)	MT	MAF
SAINT PIERRE AND MIQUELON	PM	SPM
SAINT VINCENT AND THE GRENADINES	VC	VCT
SAMOA	WS	WSM

Country	Alpha 2 code	Alpha 3 code
SAN MARINO	SM	SMR
SAO TOME AND PRINCIPE	ST	STP
SAUDI ARABIA	SA	SAU
SENEGAL	SN	SEN
SERBIA	RS	SRB
SEYCHELLES	SC	SYC
SIERRA LEONE	SL	SLE
SINGAPORE	SG	SGP
SINT MAARTEN (DUTCH PART)	SX	SXM
SLOVAKIA	SK	SVK
SLOVENIA	SI	SVN
SOLOMON ISLANDS	SB	SLB
SOMALIA	SO	SOM
SOUTH AFRICA	ZA	ZAF
SOUTH GEORGIA AND THE SOUTH SANDWICH ISLANDS	GS	SGS
SPAIN	ES	ESP
SRI LANKA	LK	LKA
SUDAN	SD	SDN
SURINAME	SR	SUR
SVALBARD AND JAN MAYEN	SJ	SJM
SWAZILAND	SZ	SWZ
SWEDEN	SE	SWE
SWITZERLAND	CH	CHE
SYRIAN ARAB REPUBLIC	SY	SYR
TAIWAN, PROVINCE OF CHINA	TW	TWN
TAJIKISTAN	TJ	TJK
TANZANIA, UNITED REPUBLIC OF	TZ	TZA

Country	Alpha 2 code	Alpha 3 code
THAILAND	TH	THA
TIMOR-LESTE	TL	TLS
TOGO	TG	TGO
TOKELAU	TK	TKL
TONGA	TO	TON
TRINIDAD AND TOBAGO	TT	TTO
TUNISIA	TN	TUN
TURKEY	TR	TUR
TURKMENISTAN	TM	TKM
TURKS AND CAICOS ISLANDS	TC	TCA
TUVALU	TV	TUV
UGANDA	UG	UGA
UKRAINE	UA	UKR
UNITED ARAB EMIRATES	AE	ARE
UNITED KINGDOM	GB	GBR
UNITED STATES	US	USA
UNITED STATES MINOR OUTLYING ISLANDS	UM	UMI
URUGUAY	UY	URY
UZBEKISTAN	UZ	UZB
VANUATU	VU	VUT
VATICAN CITY STATE (HOLY SEE)	VA	VAT
VENEZUELA, BOLIVARIAN REPUBLIC OF	VE	VEN
VIET NAM	VN	VNM
VIRGIN ISLANDS (BRITISH)	VG	VGB
VIRGIN ISLANDS (U.S.)	VI	VIR
WALLIS AND FUTUNA	WF	WLF
WESTERN SAHARA	EH	ESH
YEMEN	YE	YEM

Country	Alpha 2 code	Alpha 3 code
YUGOSLAVIA	YU	YUG
ZAMBIA	ZM	ZMB

Appendix L: TLS Cipher Suites

The following cipher suites are accepted for TLS 1.3:

TLS_AES_128_GCM_SHA256

TLS_AES_256_GCM_SHA384

TLS_CHACHA20_POLY1305_SHA256

The following cipher suites are accepted for TLS 1.2:

ECDHE-ECDSA-AES128-GCM-SHA256

ECDHE-RSA-AES128-GCM-SHA256

ECDHE-ECDSA-AES256-GCM-SHA384

ECDHE-RSA-AES256-GCM-SHA384

ECDHE-ECDSA-CHACHA20-POLY1305

ECDHE-RSA-CHACHA20-POLY1305