



Secure XML API Integration Guide

Document Control

This is a control document

| | | | |
|----------------------|----------------------------------|---------------------|------------|
| DESCRIPTION | Secure XML API Integration Guide | | |
| CREATION DATE | 02/04/2007 | CREATED BY | SecurePay |
| VERSION | 1.1 | DATE UPDATED | 07/01/2010 |

Table of Contents

| | | | |
|---|-----------|--|-----------|
| 1 Introduction..... | 4 | 8.3 Echo URLs..... | 30 |
| 1.1 About this Guide..... | 4 | 8.4 Sample XML Messages..... | 30 |
| 1.2 Intended Audience | 4 | 8.4.1 Echo | 30 |
| 2 System Overview..... | 5 | 9 XML Over HTTP..... | 32 |
| 3 Functionality | 7 | 9.1 Request..... | 32 |
| 3.1 Payment..... | 7 | 9.2 Response | 33 |
| 3.2 Echo | 7 | Appendix A: Transaction Types | 34 |
| 3.3 Authentication, Communication & Encryption | 7 | Appendix B: Transaction Sources | 35 |
| 4 XML Message Format and Contents..... | 8 | Appendix C: Card Types..... | 36 |
| 4.1 XML Header | 8 | Appendix D: Location of CVW | 37 |
| 4.2 Element Definitions..... | 8 | Appendix E: Timestamp String Format | 38 |
| 4.2.1 Element Codes | 8 | Appendix F: SecurePay Status Codes | 39 |
| 4.2.2 Element Types and Constraints | 8 | Appendix G: XML Request DTD | 40 |
| 4.3 Sample XML Request and Response... .. | 9 | Appendix H: XML Response DTD | 41 |
| 4.3.1 Request | 9 | Appendix I: Currency Codes List..... | 43 |
| 4.3.2 Response | 10 | Appendix J: EBCDIC Character Set..... | 44 |
| 5 Common XML Message Elements | 11 | | |
| 5.1 Request Messages..... | 11 | | |
| 5.1.1 MessageInfo Element | 11 | | |
| 5.1.2 MerchantInfo Element | 11 | | |
| 5.1.3 RequestType Element | 12 | | |
| 5.2 Response Messages..... | 12 | | |
| 5.2.1 MessageInfo Element | 12 | | |
| 5.2.2 MerchantInfo Element | 13 | | |
| 5.2.3 RequestType Element | 13 | | |
| 5.2.4 Status Element | 14 | | |
| 6 Payment Message Elements..... | 15 | | |
| 6.1 Request Messages..... | 15 | | |
| 6.1.1 Payment Element | 15 | | |
| 6.2 Response Messages..... | 20 | | |
| 6.2.1 Payment Element | 20 | | |
| 6.3 Sample XML Messages..... | 24 | | |
| 6.3.1 Credit Card Payment | 24 | | |
| 6.3.2 Credit Card Refund | 26 | | |
| 6.3.3 Direct Debit | 27 | | |
| 7 Payment URLs..... | 29 | | |
| 8 Echo Message Elements | 30 | | |
| 8.1 Request Messages..... | 30 | | |
| 8.2 Response Messages..... | 30 | | |

1 Introduction

1.1 About this Guide

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

SecurePay uses XML request and response messages which can be run on any platform and in any programming language. The messages are transported via HTTP over SSL.

This guide covers the process of building a program within your web site or application in order to integrate the XML API.

1.2 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 web site, XML or application programming experience reads this guide and implements the Secure XML.

2 System Overview

SecurePay's Payment Gateway provides merchants with the ability to process credit card and direct entry payments in a secure environment.

SecurePay partners with the following major banks and financial institutions in the provision of the SecurePay Payment Gateway:

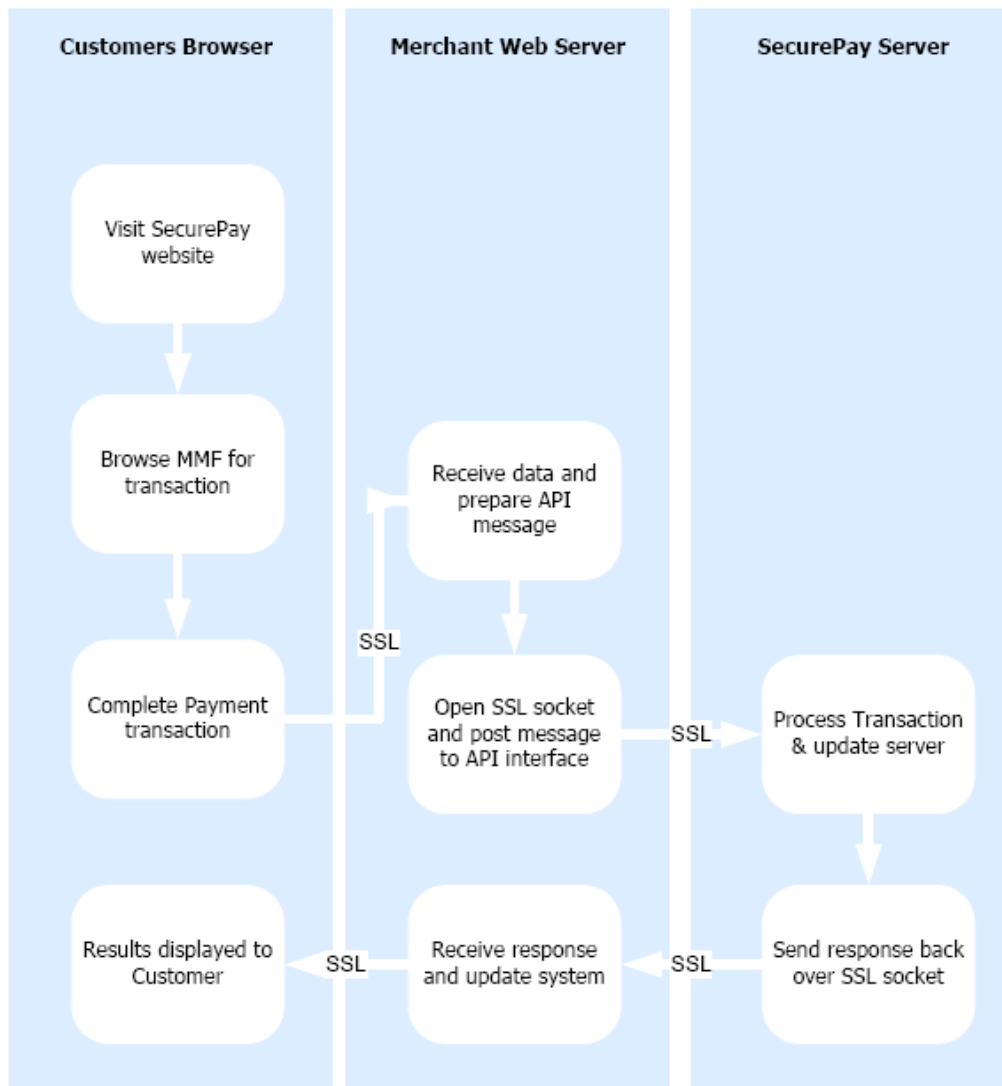
- ANZ
- American Express
- BankWest
- Commonwealth Bank
- Diners Club
- National Australia Bank
- St George (including Bank of SA)
- Westpac (including Challenge Bank and Bank of Melbourne)

Direct entry payments are not processed in real time; they are stored in SecurePay's database and processed daily at 4.30pm EST.

Secure XML supports four payment transaction types:

- Payments
- Refunds
- Preauthorise
- Preauthorise Complete

Secure XML utilises a XML request and response messages, and can be run on any platform and in any programming language. The message transport is done via HTTP protocol using SSL.



3 Functionality

Secure XML is an independent platform to process financial transactions.

3.1 Payment

The Payment requests are used to pass financial credit card transaction messages to SecurePay's Payment Server, which will authorise the transaction with the merchant's bank and customer's card issuer, and produce a response based on the banks' authorisation of the transaction.

The Payment request can be used to send following credit card transaction types:

- Credit Card Payment
- Credit Card Refund
- Credit Card Reversal (Void)
- Credit Card Preauthorise
- Credit Card Preauthorise Complete (Advice)

SecurePay's Payment Server is also capable of processing direct entry transactions. Direct entry transactions are not processed real time. SecurePay Payment Server stores direct entry transaction in a database and processes them daily at 4.30pm Melbourne time.

The Payment request can be used to send following direct entry transaction types:

- Direct Debit
- Direct Credit

3.2 Echo

The Echo message is used to verify that the SecurePay Payment Server port can be accessed. The Echo message does not validate any data, but merely sends a request message to SecurePay, and receives a response message if the SecurePay Server port is open.

3.3 Authentication, Communication & Encryption

To ensure security, each merchant is issued with 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 interface uses HTTP protocol and SSL for communication with SecurePay's Payment servers.

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

4 XML Message Format and Contents

SecurePay's Payment Server Messages can take the following formats:

- Requests are the messages sent to SecurePay's Payment Server requesting periodic or triggered transactions.
- The Echo requests are used to verify that the SecurePay's Payment Server is available. For information on Echo messages Click [Echo Message Elements](#).

4.1 XML Header

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

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

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

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

4.2 Element Definitions

4.2.1 Element Codes

The XML elements located in the Element matrices in sections below have the following definitions used to indicate whether the elements are present in the request and response messages:

| Code | Condition |
|------|--|
| P | Present, the element is required in the message and must be present. |
| O | Optional, the element is not required in the message but may be present. |
| X | Not present, the element should not be present. |

4.2.2 Element Types and Constraints

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

| Type | Constraint | Description |
|--------|------------|--|
| String | A | <ul style="list-style-type: none"> • Alphabetic characters • Value in the element is valid if it only contains characters in the specified set (alphabetic) |
| | N | <ul style="list-style-type: none"> • Numeric characters • Value in the element is valid if it only contains characters in the specified set (numeric) |
| | S | <ul style="list-style-type: none"> • 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) |

| Type | Constraint | Description |
|---------|------------|--|
| | LEN | <ul style="list-style-type: none"> 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 | <ul style="list-style-type: none"> 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 | <ul style="list-style-type: none"> 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 |
| Integer | DIGNO | <ul style="list-style-type: none"> 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 | <ul style="list-style-type: none"> Minimum numerical value Value in the element is valid if it is numerically greater than or equal to the defined minimum value |
| | MAXVAL | <ul style="list-style-type: none"> Maximum numerical value Value in the element is valid if it is numerically less than or equal to the defined maximum value |

4.3 Sample XML Request and Response

4.3.1 Request

```
<?xml version="1.0" encoding="UTF-8"?>
<SecurePayMessage>
  <MessageInfo>
    <messageID>8af793f9af34bea0cf40f5fb5c630c</messageID>
    <messageTimestamp>20041803161306527000+660</messageTimestamp>
    <timeoutValue>60</timeoutValue>
    <apiVersion>xml-4.2</apiVersion>
  </MessageInfo>
  <MerchantInfo>
    <merchantID>ABC0001</merchantID>
    <password>changeit</password>
  </MerchantInfo>
  <RequestType>Payment</RequestType>
  <Payment>
    <TxnList count="1">
      <Txn ID="1">
        <txnType>0</txnType>
        <txnSource>0</txnSource>
        <amount>1000</amount>
        <purchaseOrderNo>test</purchaseOrderNo>
        <CreditCardInfo>
          <cardNumber>4444333322221111</cardNumber>
          <expiryDate>09/15</expiryDate>
        </CreditCardInfo>
      </Txn>
    </TxnList>
  </Payment>
</SecurePayMessage>
```

4.3.2 Response

```
<?xml version="1.0" encoding="UTF-8"?>
<SecurePayMessage>
  <MessageInfo>
    <messageID>8af793f9af34bea0cf40f5fb5c630c</messageID>
    <messageTimestamp>20041803161316316000+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>0</txnSource>
        <amount>1000</amount>
        <purchaseOrderNo>test</purchaseOrderNo>
        <approved>Yes</approved>
        <responseCode>00</responseCode>
        <responseText>Approved</responseText>
        <settlementDate>20040318</settlementDate>
        <txnID>009844</txnID>
        <CreditCardInfo>
          <pan>444433...111</pan>
          <expiryDate>09/15</expiryDate>
          <cardType>6</cardType>
          <cardDescription>Visa</cardDescription>
        </CreditCardInfo>
      </Txn>
    </TxnList>
  </Payment>
</SecurePayMessage>
```

5 Common XML Message Elements

5.1 Request Messages

Requests are the messages sent to SecurePay. Following sections describe elements common to all requests.

5.1.1 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 |
|--------------------|--|
| <messageID> | 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 |
| <messageTimestamp> | Description: Time of the request. Format type: String, see Appendix E: Timestamp String Format Format constraints: NS ('+', '-'), LEN = 24 Validated by SecurePay: Yes Value: Eg: "20041803161306527000+660" Sub-elements: No |
| <timeoutValue> | Description: Timeout value used, in seconds. Format type: Integer Format constraints: DIGNO = 3, MINVAL = 1 Validated by SecurePay: Yes Value: Recommended "60" 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 |

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

5.1.3 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: <ul style="list-style-type: none"> • "Payment" • "Echo" |
| Sub-elements: | No |

5.2 Response Messages

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.

5.2.1 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 E: Timestamp String Format Format constraints: NS ('+', '-'), LEN = 24 Value: Eg: "20041803161306527000+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 |

5.2.2 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 |

5.2.3 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: <ul style="list-style-type: none"> • "Payment" • "Echo" |
| Sub-elements: | No |

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

6 Payment Message Elements

6.1 Request Messages

Following sections describe elements used in Payment requests.

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

<RequestType>Payment</RequestType>

6.1.1 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 |

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

6.1.1.1.1 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 [Transaction Type-Required Element Map](#) for information what elements are required for various payment types.

| Element | Comments |
|-------------|---|
| <Txn.ID> | <p>Description: Transaction ID is an attribute of <Txn> element and specifies transaction ID. All transactions should be numbered sequentially starting at "1".</p> <p>Note: Currently only single transactions per request are supported. Payments submitted with more than one <Txn> element will be rejected with Status code "577".</p> <p>Format type: Integer</p> <p>Format constraints: DIGNO = 1, MINVAL = 1, MAXVAL = 1</p> <p>Validated by SecurePay: Yes</p> <p>Value: Currently always "1"</p> <p>Sub-elements: No</p> |
| <txnType> | <p>Description: Transaction type specifies the type of transaction being processed.</p> <p>Format type: Integer, see Appendix A: Transaction Types</p> <p>Format constraints: DIGNO = 2, MINVAL = 0, MAXVAL = 99</p> <p>Validated by SecurePay: Yes</p> <p>Value: Eg: "0"</p> <p>Sub-elements: No</p> |
| <txnSource> | <p>Description: Transaction source specifies the source of transaction being processed. For Secure XML the source must always have a value "23".</p> <p>Format type: Integer, see Appendix B: Transaction Sources</p> <p>Format constraints: DIGNO = 2, MINVAL = 0, MAXVAL = 99</p> <p>Validated by SecurePay: Yes</p> <p>Value: Always "23"</p> <p>Sub-elements: No</p> |
| <amount> | <p>Description: Transaction amount in cents.</p> <p>Format type: Integer</p> <p>Format constraints: MINVAL = 1</p> <p>Validated by SecurePay: Yes</p> <p>Value: Eg: "123" for \$1.23</p> <p>Sub-elements: No</p> |

| Element | Comments |
|-------------------|---|
| <currency> | <p>Description: Transaction currency.</p> <p>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.</p> <p>Format type: String</p> <p>Format constraints: A, LEN = 3</p> <p>Validated by SecurePay: Yes</p> <p>Value: Eg: "AUD" for Australian Dollars</p> <p>Sub-elements: No</p> |
| <purchaseOrderNo> | <p>Description: Unique merchant transaction identifier, typically an invoice number.</p> <p>Note: Must be the same as <purchaseOrderNo> element of the original transaction when performing a refund, reversal or advice.</p> <p>Format type: String</p> <p>Format constraints: For Credit Card payments ANS (All characters allowed except spaces and "" single quote), For Direct Entry payments EBCDIC (see Appendix J: 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.</p> <p>Validated by SecurePay: Yes</p> <p>Value: Eg: "order_#000235"</p> <p>Sub-elements: No</p> |
| <txnID> | <p>Description: Bank transaction ID.</p> <p>Note: Must match the <txnID> element returned in the response to the original payment transaction when performing a refund or reversal.</p> <p>Format type: String</p> <p>Format constraints: AN, MINLEN = 6, MAXLEN = 16</p> <p>Validated by SecurePay: Yes</p> <p>Value: Eg: "TX123456"</p> <p>Sub-elements: No</p> |
| <preauthID> | <p>Description: Authorisation code of a preauthorisation transaction.</p> <p>Note: Must match the <preauthID> element returned in the response to the original preauthorisation transaction when performing an advice.</p> <p>Format type: String</p> <p>Format constraints: N, LEN = 6</p> <p>Validated by SecurePay: Yes</p> <p>Value: Eg: "123456"</p> <p>Sub-elements: No</p> |
| <CreditCardInfo> | See CreditCardInfo Element |
| <DirectEntryInfo> | See DirectEntryInfo Element |

6.1.1.1.2 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> | <p>Description: Credit card number.</p> <p>Format type: String</p> <p>Format constraints: N, MINLEN = 13, MAXLEN = 16</p> <p>Validated by SecurePay: Yes</p> <p>Value: Eg: "4242424242424242"</p> <p>Sub-elements: No</p> |
| <cvv> | <p>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 D: Location of CVV</p> <p>Format type: String</p> <p>Format constraints: N, MINLEN = 3, MAXLEN = 4</p> <p>Validated by SecurePay: Yes</p> <p>Value: Eg: "123"</p> <p>Sub-elements: No</p> |
| <expiryDate> | <p>Description: Credit card expiry date.</p> <p>Format type: String</p> <p>Format constraints: NS ('/'), LEN = 5</p> <p>Validated by SecurePay: Yes</p> <p>Value: Eg: "05/06" for May 2006</p> <p>Sub-elements: No</p> |

6.1.1.1.3 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> | <p>Description: BSB number.</p> <p>Format type: String</p> <p>Format constraints: N, LEN = 6</p> <p>Validated by SecurePay: Yes</p> <p>Value: Eg: "012012"</p> <p>Sub-elements: No</p> |

| | |
|-----------------|---|
| <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 J: EBCDIC Character Set), MINLEN = 0, MAXLEN = 32 Validated by SecurePay: Yes Value: Eg: "John Smith" Sub-elements: No |

6.1.1.1.4 Transaction Type-Required Element Map

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

| ELEMENT \ TXN TYPE | Standard Payment | Refund | Reversal | Preauthorise | Complete (Advice) |
|--|------------------|--------|----------|--------------|-------------------|
| | 0 | 4 | 6 | 10 | 11 |
| <txnType> | M | M | M | M | M |
| <txnSource> | M | M | M | M | M |
| <amount> | M | M | M | M | M |
| <currency> | O | X | X | O | X |
| <purchaseOrderNo> | M | M | M | M | M |
| <txnID> | X | M | M | X | X |
| <preauthID> | X | X | X | X | M |
| <cardNumber> | M | O | O | M | O |
| <cvv> | O | O | O | O | O |
| <expiryDate> | M | O | O | M | O |
| M – Mandatory O – Optional X – Not required (ignored) | | | | | |

The table below shows which elements are required for each direct entry transaction type. Elements are mandatory, optional or not required.

| ELEMENT \ TXN TYPE | Direct Debit | Preauthorise Complete (Advice) |
|--------------------|--------------|--------------------------------|
| | 15 | 17 |
| <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) | | |

6.2 Response Messages

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

6.2.1 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 Element |

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

6.2.1.1.1 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> | <p>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.</p> <p>Note: Currently only single transactions per request are supported. Payments submitted with more than one <Txn> element will be rejected with Status code "577".</p> <p>Format type: Integer</p> <p>Format constraints: DIGNO = 1, MINVAL = 1, MAXVAL = 1</p> <p>Value: Currently always "1"</p> <p>Sub-elements: No</p> |
| <txnType> | <p>Description: Transaction type specifies the type of transaction processed. Returned unchanged from the request.</p> <p>Format type: Integer, see Appendix A: Transaction Types</p> <p>Format constraints: DIGNO = 2, MINVAL = 0, MAXVAL = 99</p> <p>Value: Eg: "0"</p> <p>Sub-elements: No</p> |
| <txnSource> | <p>Description: Transaction source specifies the source of transaction processed. Returned unchanged from the request.</p> <p>Format type: Integer, see Appendix B: Transaction Sources</p> <p>Format constraints: DIGNO = 2, MINVAL = 0, MAXVAL = 99</p> <p>Value: Eg: "23"</p> <p>Sub-elements: No</p> |
| <amount> | <p>Description: Transaction amount in cents. Returned unchanged from the request.</p> <p>Format type: Integer</p> <p>Format constraints: MINVAL = 1</p> <p>Value: Eg: "123" for \$1.23</p> <p>Sub-elements: No</p> |
| <currency> | <p>Description: Transaction currency. Returned unchanged from the request. If not set in the request, a default value of "AUD" is returned.</p> <p>Note: Only applicable to Credit Card payments.</p> <p>Format type: String</p> <p>Format constraints: A, LEN = 3</p> <p>Validated by SecurePay: Yes</p> <p>Value: Eg: "AUD" for Australian Dollars</p> <p>Sub-elements: No</p> |

| Element | Comments |
|-------------------|---|
| <purchaseOrderNo> | <p>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 preauthorize transactions this value is returned unchanged from the request.</p> <p>Format type: String</p> <p>Format constraints: For Credit Card payments ANS (All characters allowed except spaces and “” single quote), For Direct Entry payments EBCDIC (see Appendix J: EBCDIC Character Set), MINLEN = 1, MAXLEN = 60</p> <p>Value: Eg: “order_#000235”</p> <p>Sub-elements: No</p> |
| <approved> | <p>Description: Indicates whether the transaction processed has been approved or not.</p> <p>Format type: String</p> <p>Format constraints: A, MINLEN = 2, MAXLEN = 3</p> <p>Value: Always “Yes” or “No”</p> <p>Sub-elements: No</p> |
| <responseCode> | <p>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 SecurePay Payment Response Codes documents for details of codes returned. This document may be downloaded from SecurePay’s Merchant Login website or provided via email by SecurePay’s Merchant Support team.</p> <p>Format type: String</p> <p>Format constraints: AN, MINLEN = 2, MAXLEN = 3</p> <p>Value: Eg: “00”</p> <p>Sub-elements: No</p> |
| <responseText> | <p>Description: Textual description of the response code received.</p> <p>Format type: String</p> <p>Format constraints: ANS (All characters allowed), MINLEN = 0, MAXLEN = 40</p> <p>Value: Eg: “Approved”</p> <p>Sub-elements: No</p> |
| <settlementDate> | <p>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.)</p> <p>Format type: String</p> <p>Format constraints: N, LEN = 8</p> <p>Value: Eg: “20040326” for 26th March 2004</p> <p>Sub-elements: No</p> |
| <txnID> | <p>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.</p> <p>Format type: String</p> <p>Format constraints: AN, MINLEN = 6, MAXLEN = 16</p> <p>Value: Eg: “TX123456”</p> <p>Sub-elements: No</p> |

| Element | Comments |
|-------------------|--|
| <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. Format type: String Format constraints: N, LEN = 6 Value: Eg: "123456" Sub-elements: No |
| <CreditCardInfo> | See CreditCardInfo Element |
| <DirectEntryInfo> | See DirectEntryInfo Element |

6.2.1.1.2 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 |
| <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 C: 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 C: Card Types Format constraints: A, MINLEN = 0, MAXLEN = 20 Value: Eg: "Visa" Sub-elements: No |

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

6.3 Sample XML Messages

6.3.1 Credit Card Payment

6.3.1.1 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>changeit</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>09/15</expiryDate>
  </CreditCardInfo>
</Txn>
</TxnList>
</Payment>
</SecurePayMessage>
```

6.3.1.2 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>09/15</expiryDate>
          <cardType>6</cardType>
          <cardDescription>Visa</cardDescription>
        </CreditCardInfo>
      </Txn>
    </TxnList>
  </Payment>
</SecurePayMessage>
```

6.3.2 Credit Card Refund

6.3.2.1 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>changeit</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>
        <CreditCardInfo>
          <cardNumber>4444333322221111</cardNumber>
          <expiryDate>09/15</expiryDate>
        </CreditCardInfo>
      </Txn>
    </TxnList>
  </Payment>
</SecurePayMessage>
```

6.3.2.2 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>
      </Txn>
    </TxnList>
  </Payment>
</SecurePayMessage>
```

```
<purchaseOrderNo>009887</purchaseOrderNo>
<approved>Yes</approved>
<responseCode>00</responseCode>
<responseText>Approved</responseText>
<settlementDate>20040323</settlementDate>
<txnID>009890</txnID>
<CreditCardInfo>
  <pan>444433...111</pan>
  <expiryDate>09/15</expiryDate>
  <cardType>6</cardType>
  <cardDescription>Visa</cardDescription>
</CreditCardInfo>
</Txn>
</TxnList>
</Payment>
</SecurePayMessage>
```

6.3.3 Direct Debit

6.3.3.1 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>changeit</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>
```

6.3.3.2 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>
        <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>
```

7 Payment URLs

The Payment messages must be sent to the following URLs.

For credit card transactions:

Test URL (no SSL): <http://test.securepay.com.au/xmlapi/payment>

Test URL (SSL): <https://test.securepay.com.au/xmlapi/payment>

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

For direct entry transactions:

Test URL (no SSL): <http://test.securepay.com.au/xmlapi/directentry>

Test URL (SSL): <https://test.securepay.com.au/xmlapi/directentry>

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

8 Echo Message Elements

8.1 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>*

SecurePay recommends that 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.

8.2 Response Messages

Echo responses do not return any additional elements. The <Status> element will return a response code "000" if the service is available.

8.3 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.

8.4 Sample XML Messages

8.4.1 Echo

8.4.1.1 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>changeit</password>
  </MerchantInfo>
  <RequestType>Echo</RequestType>
</SecurePayMessage>
```

8.4.1.2 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>
```

9 XML Over HTTP

The structure of the HTTP request and response messages will conform to the HTTP 1.1 network protocol. Below are examples of the expected HTTP exchange.

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

9.1 Request

```
POST /test/payment HTTP/1.1
host: www.securepay.com.au
content-type: text/xml
content-length: 677
```

```
<?xml version="1.0" encoding="UTF-
8"?><SecurePayMessage><MessageInfo><messageID>8af793f9af34bea0cf40f5fc011e0
c</messageID><messageTimestamp>20041904145505116000+600</messageTimestamp><
timeoutValue>60</timeoutValue><apiVersion>xml-
4.2</apiVersion></MessageInfo><MerchantInfo><merchantID>abc0001</merchantID
><password>abc123</password></MerchantInfo><RequestType>Payment</RequestTyp
e><Payment><TxnList count="1"><Txn
ID="1"><txnType>0</txnType><txnSource>23</txnSource><amount>200</amount><pu
rchaseOrderNo>test</purchaseOrderNo><CreditCardInfo><cardNumber>42424242424
24242</cardNumber><expiryDate>07/06</expiryDate></CreditCardInfo></Txn></Tx
nList></Payment></SecurePayMessage>
```

9.2 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: Microsoft-IIS/5.0
Date: Mon, 19 Apr 2004 06:19:48 GMT
HTTP/1.1 200 OK
Server: Microsoft-IIS/5.0
Date: Mon, 19 Apr 2004 06:20:01 GMT
Content-Type: text/xml;charset=ISO-8859-1
Content-Length: 929
```

```
<?xml version="1.0" encoding="UTF-8"?>
<SecurePayMessage><MessageInfo><messageID>8af793f9af34bea0cf40f5fc011e0c</m
essageID><messageTimestamp>20041904161959849000+600</messageTimestamp><apiV
ersion>xml-
4.2</apiVersion></MessageInfo><RequestType>Payment</RequestType><MerchantIn
fo><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><cu
rrency>AUD</currency><purchaseOrderNo>test</purchaseOrderNo><approved>Yes</
approved><responseCode>00</responseCode><responseText>Approved</responseTex
t><settlementDate>20040419</settlementDate><txnID>009729</txnID><CreditCard
Info><pan>424242...242</pan><expiryDate>07/06</expiryDate><cardType>6</card
Type><cardDescription>Visa</cardDescription></CreditCardInfo></Txn></TxnLis
t></Payment></SecurePayMessage>
```

Appendix A: Transaction Types

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

Codes with **shaded background** are permitted in Payment transactions processed using Secure XML. All other codes are provided for completeness.

| Code | Description |
|-----------|----------------------------------|
| 0 | Standard Payment |
| 1 | Mobile Payment |
| 2 | Batch Payment |
| 3 | Periodic Payment |
| 4 | Refund |
| 5 | Error Reversal (Void) |
| 6 | Client Reversal (Void) |
| 10 | Preauthorise |
| 11 | Preauth Complete (Advice) |
| 14 | Recurring Payment |
| 15 | Direct Entry Debit |
| 17 | Direct Entry Credit |
| 19 | Card-Present Payment |
| 20 | IVR Payment |

Appendix B: Transaction Sources

Transaction source codes track the origin of financial transaction processed by SecurePay.

Codes with **shaded background** are permitted in Payment transactions processed using Secure XML. All other codes are provided for completeness.

| Code | Description |
|-----------|----------------------------|
| 0 | Unknown |
| 1 | SecureLink |
| 2 | Merchant Login |
| 3 | SATM |
| 4 | SecureBill Portal |
| 5 | SecureBill Link |
| 6 | Reserved |
| 7 | SecurePOS |
| 8 | API (SecureJava) |
| 9 | Call Centre Payment Switch |
| 10 | Batch Server |
| 11 | IVR1 |
| 12 | IVR2 |
| 13 | SecureMobile |
| 14 | Reconciliation Engine |
| 15 | Reserved |
| 16 | Helpdesk Login |
| 18 | eSec Client |
| 19 | Periodic Server |
| 20 | Reserved |
| 21 | Reserved |
| 22 | Reserved |
| 23 | XML |
| 90 | Reserved |

Appendix C: 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 D: 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 E: 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 F: SecurePay Status Codes

| Status Code | Response Text | Description |
|-------------|--|--|
| 000 | Normal | Message processed correctly (check transaction response for details). |
| 504 | Invalid Merchant ID | If Merchant ID does not follow the format XXXDDDD, where X is a letter and D is a digit, or Merchant ID is not found in SecurePay's database. |
| 505 | Invalid URL | The URL passed to either Echo, Query, or Payment object is invalid. |
| 510 | Unable To Connect To Server | Produced by SecurePay Client API when unable to establish connection to SecurePay Payment Gateway |
| 511 | Server Connection Aborted During Transaction | Produced by SecurePay Client API when connection to SecurePay Payment Gateway is lost after the payment transaction has been sent |
| 512 | Transaction timed out By Client | Produced by SecurePay Client API when no response to payment transaction has been received from SecurePay Payment Gateway within predefined time period (default 80 seconds) |
| 513 | General Database Error | Unable to read information from the database. |
| 514 | Error loading properties file | Payment Gateway encountered an error while loading configuration information for this transaction |
| 515 | Fatal Unknown Error | Transaction could not be processed by the Payment Gateway due to unknown reasons |
| 516 | Request type unavailable | SecurePay system doesn't support the requested transaction type |
| 517 | Message Format Error | SecurePay Payment Gateway couldn't correctly interpret the transaction message sent |
| 524 | Response not received | The client could not receive a response from the server. |
| 545 | System maintenance in progress | The system maintenance is in progress and the system is currently unable to process transactions |
| 550 | Invalid password | The merchant has attempted to process a request with an invalid password. |
| 575 | Not implemented | This functionality has not yet been implemented |
| 577 | Too Many Records for Processing | The maximum number of allowed events in a single message has been exceeded. |
| 580 | Process method has not been called | The process() method on either Echo, Payment or Query object has not been called |
| 595 | Merchant Disabled | SecurePay has disabled the merchant and the requests from this merchant will not be processed. |

Appendix G: 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?, 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 txnID (#PCDATA)>
<!-- ELEMENT preauthID (#PCDATA)>
<!-- ELEMENT CreditCardInfo (cardNumber, cvv?, expiryDate?)>
<!-- ELEMENT DirectEntryInfo (bsbNumber, accountNumber, accountName)>

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

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

Appendix H: XML Response DTD

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

<!-- define elements for SecurePayMessage -->
<!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?, 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 CreditCardInfo (pan, expiryDate?, cardType?, cardDescription?)>
<!ELEMENT DirectEntryInfo (bsbNumber, accountNumber, accountName)>

<!-- 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)>
```

Appendix I: Currency Codes List

You **must meet** certain requirements with your bank and SecurePay before using SecurePay's multi-currency features. Please ask SecurePay if we support multi-currency payments through your bank, and if so, what currency types are available. You may also need to open multi-currency accounts with your bank for each currency you propose to transact in. Contact SecurePay Support or your SecurePay Account Manager for full details.

| 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 J: EBCDIC Character Set

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