

# ST XML Specification

For use with ST Xpay and ST Xpay4

Version 3.51

---

### Copyright

© SecureTrading 2011. All rights reserved. No part of this document may be photocopied, reproduced, stored in a retrieval system or transmitted in any form or by any means whether electronic, mechanical or otherwise without the prior written permission of SecureTrading Ltd.

### Disclaimer

This document is for informational purposes only. SecureTrading make no warranties, express or implied, through the distribution of this document. No warranty of accuracy is given concerning the contents of the information contained in this publication. To the extent permitted by law no liability (including liability to any person by reason of negligence) will be accepted by SecureTrading, its subsidiaries or employees for any direct or indirect loss or damage caused by omissions from or inaccuracies in this document.

SecureTrading reserves the right to revise the content without obligation to notify any person of such changes.

Document revised on 31-Mar-2011.

## Contents

1	Introduction.....	5
1.1	XML specification .....	5
1.1.1	XML request.....	6
1.1.2	XML response .....	7
1.1.2.1	ResponseBlock .....	7
1.1.2.2	Response .....	7
1.1.3	Future compatibility considerations.....	9
2	Transaction authorisation .....	10
2.1	Authorisation request .....	10
2.1.1	XML specification .....	10
2.1.1.1	<Operation> .....	10
2.1.1.2	<PaymentMethod> .....	11
2.1.2	Switch/Maestro Changes .....	12
2.1.2.1	<CustomerInfo>.....	13
2.1.2.1.1	<Postal> .....	13
2.1.2.1.2	<Telecom> .....	14
2.1.2.1.3	<Online> .....	14
2.1.2.2	<Order> .....	14
2.1.3	XML example.....	14
2.2	Authorisation response .....	16
2.2.1	XML specification .....	16
2.2.1.1	<OperationResponse> .....	16
2.2.1.2	<Order> .....	18
2.2.2	XML example.....	18
2.3	Authorisation reversal request.....	18
2.3.1	XML specification .....	18
2.3.2	XML example.....	19
2.4	Authorisation reversal response .....	20
2.4.1	XML specification .....	20
2.4.2	XML example.....	20
3	Transaction refunds.....	21
3.1	Refund request.....	21
3.1.1	XML specification .....	21
3.1.1.1	<Operation> .....	21
3.1.1.2	<PaymentMethod> .....	21
3.1.1.3	<CustomerInfo>.....	22
3.1.2	XML example.....	22
3.2	Refund response .....	23
3.2.1	XML specification .....	23
3.2.2	XML example.....	23
3.3	Refund reversal request .....	23
3.3.1	XML specification .....	23
3.3.2	XML example.....	24
3.4	Refund reversal response.....	25
3.4.1	XML specification .....	25
3.4.2	XML example.....	25
4	Settlement control .....	26
4.1	Settlement request .....	26
4.1.1	XML specification .....	26
4.1.1.1	<Operation> .....	26
4.1.2	XML example.....	27
4.2	Settlement response.....	28
4.2.1	XML specification .....	28
4.2.1.1	<OperationResponse> .....	28

---

4.2.2	XML example.....	29
5	Card Check (BinLookup).....	30
5.1	BinLookup request.....	30
5.1.1	XML specification.....	30
5.1.1.1	<Operation>.....	30
5.1.1.2	<Filter>.....	30
5.1.2	XML Example.....	30
5.2	BinLookup response.....	31
5.2.1	XML specification.....	31
5.2.1.1	<OperationResponse>.....	31
5.2.1.1.1	<Bin>.....	31
5.2.2	XML Example.....	31
6	Further information.....	32
6.1	Support.....	32
6.2	Further reading.....	32
7	Glossary of terms.....	33
8	APPENDICES.....	34
8.1	Xpay and Xpay4 error codes.....	34
8.2	Security check responses.....	35

## 1 Introduction

Thank you for choosing the SecureTrading ST Xpay API. This document defines the XML specification for requests using the ST Xpay and ST Xpay4 APIs. In most cases the XML is identical for each version. Where there are differences the specification will state how Xpay and Xpay4 differ.

### 1.1 XML specification

This document pays specific attention to the fields that are used to create the XML strings and which of these are mandatory. Full examples of the various requests and responses follow each request/response description.

Note: XML is *case sensitive*.

SecureTrading has applied the convention of element and tag names with capital letters at the *start* of each word, with *no spaces* separating the words. Any space in the tag precedes an attribute relating to that tag. The values of the tag attributes are *case sensitive*. In most cases the values of tag attributes *must* be in upper case, unless otherwise specified.

For example:

```
<ElementName Attribute="VALUE">  
    <TagName>    </TagName>  
</ElementName>
```

More information on XML can be found at the XML website: <http://www.xml.org>

Note: It is important to include the `Version` attribute in all ST Xpay and Xpay4 requests sent.

Data submitted within an XML tag must not have any leading or trailing white-space characters.

Future updates of the ST Xpay and Xpay4 XML will be documented in newer versions of this document. You may send over a specific version of an ST Xpay or Xpay4 XML by specifying the appropriate `Version` number in an attribute of the `<RequestBlock>`.

#### Attention

When processing ST Xpay or Xpay4 requests and responses please pay attention to the version number submitted as different version numbers may require different XML requests and could return different XML responses than detailed in this document.

### 1.1.1 XML request

All requests sent to SecureTrading originate from an XML <RequestBlock> that has one attribute. The attribute name is *Version* and must contain the ST Xpay or Xpay4 version (found on the front of this document). Within the <RequestBlock> root element there is currently support for one <Request> element and a <Certificate> element.

An ST Xpay or Xpay4 request must contain a single <RequestBlock> followed by at least one newline character.

- For ST Xpay the XML header is ignored. All character encoding is assumed to be *ascii*
- For Xpay4 the XML header specifying the version and character encoding is required. Valid encodings include 'ascii', 'utf-8' and 'iso-8859-1'. However, only characters from the latin-1 (iso-8859-1) character set are supported.

For example:

```
<?xml version="1.0" encoding="utf-8"?>
<RequestBlock Version="3.51">
  <Request Type="AUTH">
    .
    .
  </Request>
  <Certificate>
    .
    .
  </Certificate>
</RequestBlock>\n
```

The <Certificate> element is required.

- For ST Xpay it must contain a copy of your private certificate information. This should be obtained from SecureTrading support in the form of a file containing the certificate. An exact copy of the certificate should be placed in the <Certificate> element before any type of transaction can connect to a payment gateway.
- For ST Xpay4 this must contain the alias of the key/certificate you wish to use for this request. The alias is usually the same as your sitereference.

For example:

Xpay	Xpay4
<pre>&lt;Certificate&gt; -----BEGIN CERTIFICATE----- MIIGUTCCBg+gAwIBAgICAQMwCwYHKOZIZjgE nlkY8DXrEXixAhRcJrEC3JnDXLRDZc32MciH vISP9Q== -----END CERTIFICATE----- -----BEGIN RSA PRIVATE KEY----- j6D5a16AuWU0/uqL6VVe6POgBorSFeWd18Rn QLfAinVKEu8UkWQBS1Y4j5ICFGRcYaJAnR5A J4xXUK3CHVbLEBkW -----END RSA PRIVATE KEY----- &lt;/Certificate&gt;</pre>	<pre>&lt;Certificate&gt;site12345&lt;/Certificate&gt;</pre>

The <Request> element is required and consists of a number of child elements. It has one attribute, *Type*, which is required. It must contain one of the following values:

- "AUTH" An authorisation request
- "AUTHREVERSAL" An authorisation reversal request

- "REFUND" A refund request
- "REFUNDREVERSAL" A refund reversal request
- "SETTLEMENT" A settlement request

For example `<Request Type="AUTH">`

Other request types are available and are detailed in separate specification documents. For further information please refer to [Further reading](#).

The number of child elements within a `<Request>` is dependent on the type of transaction. Any additional `<RequestBlock>` or `<Request>` elements are considered invalid.

## 1.1.2 XML response

### 1.1.2.1 ResponseBlock

Within the `<ResponseBlock>` there is support for one `<Response>` element. The `<ResponseBlock>` contains two attributes:

```
<ResponseBlock Live="TRUE" Version="3.51">
```

#### Live

The *Live* attribute describes the status of the merchant account. The *Live* attribute will contain one of the following values:

"TRUE" The account is in live mode

"FALSE" The account is in test mode and will be processed through the SecureTrading test system

Initial development will normally be done using a test certificate connecting to the test bank.

- You may use the same keystore for both live and test transactions. If your keystore may have been compromised during testing you should follow the installation instructions to generate a new private key and request a new certificate.

Note this attribute will only be populated for requests where the live status is applicable. Requests which do not involve processing a new transaction may set the attribute to "".

#### Version

The version attribute contains the version number of the response. Usually this is the same as the value submitted in the request.

### 1.1.2.2 Response

A ST Xpay or Xpay4 response will contain a single `<ResponseBlock>`. Within the `<ResponseBlock>` there is currently support for one `<Response>` element.

The `<Response>` element contains one attribute: *Type*. This attribute will contain the value submitted in the request.

For example:

```
<?xml version="1.0" encoding="utf-8"?>
<ResponseBlock Live="FALSE" Version="3.51">
  <Response Type="AUTH">
    .
  </Response>
</ResponseBlock>
```

```
    .  
    </Response>  
</ResponseBlock>
```

The contents of the <Response> element will vary depending on the request type sent. It should be noted that the <Response> element will include elements of its own. One of these elements will always contain a <Result> tag.

A <Result> of 0 indicates that a problem was encountered. In this case no element other than the <Message> tag should be relied on. Other tags may exist but may not fulfil their specification. For example in the case of a 0 <Result> the <TransactionReference> may not be unique.

- For ST Xpay responses the XML header is not returned. The XML is encoded in ascii format.
- For ST Xpay4 responses the XML is encoded in utf-8 format.

---

### 1.1.3 Future compatibility considerations

In order to maintain compatibility with future updates to this specification the following criteria should be adhered to.

- A missing response tag should be considered the same as an empty tag
- A missing response element should be considered the same as an empty element
- The order of the tags within a given block may change at any time and should not be depended upon
- Any unexpected elements should be ignored <sup>1</sup>
- Any unexpected tags should be ignored <sup>1</sup>
- Any unexpected attributes should be ignored <sup>1</sup>
- White-space characters at the start and end of data within a tag should be ignored

---

<sup>1</sup> This allows for future revisions of the specification to have additional information contained in the response.

## 2 Transaction authorisation

The following section details the XML required to allow ST Xpay or Xpay4 to submit a credit or debit card authorisation, and the XML returned following the processing of the request.

### 2.1 Authorisation request

An authorisation request involves the sending of a credit or debit card authorisation to an acquiring bank.

#### 2.1.1 XML specification

To initiate an authorisation request the `<Request>` element attribute `Type` must be set to "AUTH". The element can contain up-to four child elements:

```
<Request Type="AUTH">
  <Operation>           </Operation>
  <PaymentMethod>      </PaymentMethod>
  <CustomerInfo>       </CustomerInfo>
  <Order>              </Order>
</Request>
```

The four child elements do not take any attributes. Of the four, only `<Operation>` and `<PaymentMethod>` are required.

##### 2.1.1.1 `<Operation>`

The `<Operation>` element includes four tags:

```
<Operation>
  <SiteReference>String</SiteReference>
  <Amount>INT</Amount>
  <Currency>String</Currency>
  <SettlementDay>INT</SettlementDay>
</Operation>
```

**Note:** The `<Amount>` and `<SiteReference>` tags are required.

The `<Currency>` tag is optional and defaults to GBP (Pound Sterling) if omitted. If a currency is specified in the tag it must conform to the three-character ISO Currency Code convention. For example:

```
<Currency>USD</Currency>
```

Please contact the Support team for an up to date list of currencies supported by SecureTrading.

The `<Amount>` tag should contain the monetary value of the transaction. The amount must be in base units of the currency specified in the `<Currency>` tag, i.e. there must be no decimal point or separators.

For example £10.43 would be sent as `<Amount>1043</Amount>`.

The `<SiteReference>` tag should only include a valid SecureTrading site reference, For example. `<SiteReference>testref1234</SiteReference>`

A valid SecureTrading site reference will be issued to a merchant upon completion of the application process.

Further details of the application process are available online. Please refer to the Getting Started web page: <http://www.securetrading.com/easysteps1.html>

The `<SettlementDay>` tag is optional and defaults to 1. If submitted it should include a number denoting the delay required before a transaction is sent for settlement processing.

If an alternative settlement period is required, please use the following numeric convention:

- 0: Defer Settlement (Will not settle until manual intervention)
- 1: Next available Settlement (Usually same day). **Default**
- 2: Second Available Settlement (Usually next day)

And so on.

Please note that unsettled transactions will remain valid with an acquiring bank for a limited period of time (usually 3-10 days). Clarification with the acquiring bank should be sought to determine the maximum duration a transaction will remain valid for settlement.

### 2.1.1.2 `<PaymentMethod>`

Within the `<PaymentMethod>` element there is one child element:

```
<PaymentMethod>
  <CreditCard> </CreditCard>
</PaymentMethod>
```

The `<CreditCard>` tag must be used for a new authorisation request and contains up to eight child elements:

```
<CreditCard>
  <Type>String</Type>
  <Number>INT</Number>
  <Issue>INT</Issue>
  <StartDate>String</StartDate>
  <ExpiryDate>String</ExpiryDate>
  <SecurityCode>INT</SecurityCode>
  <TransactionVerifier>String</TransactionVerifier>
  <ParentTransactionReference>String</ParentTransactionReference>
</CreditCard>
```

### Initial authorisation

On an initial authorisation, the tags `<Type>`, `<Number>` and `<ExpiryDate>` are required. Certain credit cards may also require the `<StartDate>` or `<Issue>` tags.

For a full list of cards requiring a start date and an issue number, please contact your acquiring bank.

The `<Type>` tag should include the credit card type. The following values may be used (depending upon what card types are enabled on your SecureTrading account):

Visa, MasterCard, Maestro, Delta, Amex, Electron.

## IMPORTANT NOTE

The above list of card types contains the most popular credit card types used by merchants. If your internet merchant account supports different credit card types, you must alter your list to only include your supported credit card types. A full list of card types and test card numbers is available in the SecureTrading testing document.

The `<Number>` tag should consist of a single `INT` that represents the credit card number, For example.

4111111111111111	is valid
4111 1111 1111 1111	is invalid
4111-1111-1111-1111	is invalid

The `<ExpiryDate>` tag must be in the format `mm/yy`, where `mm` is a 2-digit month and `yy` is the last 2 digits of the year, for example:

06/07	is valid as June 2007
12/08	is valid as Dec 2008
12/1	is invalid
2/05	is invalid

The `<StartDate>` tag is only required for some payment cards. For further information please contact your acquiring bank.

If `<StartDate>` is required, it will take the same format as the `<ExpiryDate>`.

The `<Issue>` tag is only required for certain payment cards. For further information please contact your acquiring bank. The format is an `INT` representing the issue number.

The whole tag should be omitted or left blank for payments that do not require it.

The `<SecurityCode>` tag can be used to send additional information on an authorisation to enhance the security checks that an acquirer will perform on a transaction. The `<SecurityCode>` tag is used in conjunction with the address details stored in the `<Postal>` element. The `<SecurityCode>` tag must contain the string of security digits provided by a cardholder.

Note: Before using this feature, please inform the SecureTrading Customer Services team of your intention to use this feature and to check the availability of this service with your acquiring bank as well as the credit card types and the currencies that this check can be performed on.

For example, only a few Laser cards support the security code check and not all acquiring banks can accept the security code check for Laser cards, therefore clarification from SecureTrading must be obtained before merchants process Laser cards.

### 2.1.2 Switch/Maestro Changes

With Maestro debit cards replacing Switch cards, Switch cards will now be processed as Maestro debit cards on the SecureTrading payment system and you are required to submit “Maestro” in place of “Switch” for the card type when processing an authorisation.

A secondary change that MasterCard is enforcing is that all Maestro debit cards must use MasterCard SecureCode. Current SecureTrading merchants will continue to be able to process Switch/Maestro cards as normal authorisations (Type: AUTH) but will be required to update their transaction processing so that all Maestro debit cards are processed using MasterCard SecureCode (Type: ST3DAUTH).

Further information regarding the processing of ST3DAUTH transactions which is required to comply with these changes can be found within the XPay and Xpay4 3D Secure document, section 0.

**IMPORTANT:** By processing Maestro debit cards after July 1<sup>st</sup> as normal authorisations, you may incur a fine from MasterCard.

For SecureTrading contact information refer to [Support](#).

### Repeat authorisation

For repeat payments only, the `<TransactionVerifier>` and `<ParentTransactionReference>` are required. These fields can be found in the response from the authorisation requiring a repeat authorisation. See [Authorisation response](#).

In some instances, it may be necessary to change one or more sections of payment information from the original transaction (if the expiry date has changed for example).

On a repeat payment where the `<TransactionVerifier>` and `<ParentTransactionReference>` have been specified, the other tags are optional and will override those implied by the `<TransactionVerifier>`.

**Note:** `<ParentTransactionReference>` is the transaction reference of the original transaction that will be repeated.

For example:

```
<PaymentMethod>
  <CreditCard>
    <TransactionVerifier>ljhLKH6H7fjhg+764JHERsdFGhKJHGjh
    see09DSs+SDF233DSFL2=XCV987SDF+L2H34LJGSDF08khdfxg09L
    H98yADSKH98yASDIUHas89yASDIU=as97ytASGYa7523PY3TNPO8Q
    </TransactionVerifier>
    <ParentTransactionReference>1-6-1632</ParentTransactionReference>
    <ExpiryDate>02/15</ExpiryDate>
  </CreditCard>
</PaymentMethod>
```

In the case where both `<Number>`, `<ExpiryDate>` and `<TransactionVerifier>` are included, the values of `<Number>` and `<ExpiryDate>` child elements will override those implied by the `<TransactionVerifier>`.

#### 2.1.2.1 `<CustomerInfo>`

The `<CustomerInfo>` element, which is optional, consists of three child elements.

```
<CustomerInfo>
  <Postal>           </Postal>
  <Telecom>         </Telecom>
  <Online>          </Online>
</CustomerInfo>
```

It is strongly recommended that all tags in each of these elements be filled in.

##### 2.1.2.1.1 `<Postal>`

The details in the `<Postal>` element include the name and postal address details of the credit card holder.

**Note:** The `<Postal>` element is required for AVS checking.

```

<Postal>
  <Name>
    <NamePrefix>String</NamePrefix>
    <FirstName>String</FirstName>
    <MiddleName>String</MiddleName>
    <LastName>String</LastName>
    <NameSuffix>String</NameSuffix>
  </Name>
  <Company>String</Company>
  <Street>String</Street>
  <City>String</City>
  <StateProv>String</StateProv>
  <PostalCode>String</PostalCode>
  <CountryCode>String</CountryCode>
</Postal>

```

The <CountryCode> tag follows the ISO country code standard.

Please contact the Support team for an up to date list of country codes supported by SecureTrading.

#### 2.1.2.1.2 <Telecom>

The tag in the <Telecom> section is for the telephone details for a particular order

```

<Telecom>
  <Phone>String</Phone>
</Telecom>

```

#### 2.1.2.1.3 <Online>

The <Online> element contains a single tag used for electronic information.

```

<Online>
  <Email>String</Email>
</Online>

```

#### 2.1.2.2 <Order>

The <Order> element, which is returned to the merchant, consists of two tags, both of which are optional. However, their inclusion is recommended.

```

<Order>
  <OrderReference>String</OrderReference>
  <OrderInformation>String</OrderInformation>
</Order>

```

Both of the tags can contain any combination of alphanumeric characters. It should be noted that the maximum number of characters a tag can store is 255.

### 2.1.3 XML example

The following is an example of the XML string sent to ST Xpay or Xpay4.

```

<?xml version="1.0" encoding="iso-8859-1"?>
<RequestBlock Version="3.51">
  <Request Type="AUTH">
    <Operation>
      <Amount>1000</Amount>
      <Currency>GBP</Currency>
      <SiteReference>testref1234</SiteReference>
      <SettlementDay>1</SettlementDay>
    </Operation>
    <CustomerInfo>
      <Postal>
        <Name>
          <NamePrefix>Mr.</NamePrefix>
          <FirstName>Joe</FirstName>
          <MiddleName>A.</MiddleName>
          <LastName>Bloggs</LastName>
          <NameSuffix>CEng.</NameSuffix>
        </Name>
        <Company>A COMPANY</Company>
        <Street>A STREET</Street>
        <City>A CITY</City>
        <StateProv>A STATE</StateProv>
        <PostalCode>TE2 3ST</PostalCode>
        <CountryCode>GBR</CountryCode>
      </Postal>
      <Telecom>
        <Phone>0000 111111</Phone>
      </Telecom>
      <Online>
        <Email>CUSTOMER@SOMEDOMAIN.COM</Email>
      </Online>
    </CustomerInfo>
    <PaymentMethod>
      <CreditCard>
        <Type>VISA</Type>
        <Number>4111111111111111</Number>
        <Issue></Issue>
        <StartDate></StartDate>
        <ExpiryDate>02/05</ExpiryDate>
        <SecurityCode>246</SecurityCode>
      </CreditCard>
    </PaymentMethod>
    <Order>
      <OrderReference>Order0001</OrderReference>
      <OrderInformation>Test Order</OrderInformation>
    </Order>
  </Request>
  <Certificate>
    -----BEGIN CERTIFICATE-----
    MIIGUTCCBg+gAwIBAgICAQMwCwYHKoZIzjgEAwUAMIG/MQswCQYDVQQGEwJVSzEP
    MA0GA1UECBMTG9uZG9uMQswCQYDVQQHEwJTVDEWMBQGA1UEChMNU2VjdXJlVHJh
    SPEeso/HNlkY8DXrEXixAhRcJrEC3JnDXLRDZc32MciHvISP9Q==
    -----END CERTIFICATE-----
    -----BEGIN RSA PRIVATE KEY-----
    TWhG9Qn9kJ+h15rJBvbQtNvFMBuLlIXEVOxZboteE+2lW7iYz5gF7HWUPgjm+RTM
    j6D5al6AuWU0/uqL6VVe6POgBorSFeWd18RnQLfAinVkEu8UkWQBS1Y4j5ICFGRc
    YaJAnR5AJ4xXUK3CHVbLEBkW
    -----END RSA PRIVATE KEY-----
  </Certificate>
  <!--NOTE for ST Xpay4 the Certificate tag should just contain your certificate alias-->
</RequestBlock>

```

## 2.2 Authorisation response

Upon processing an authorisation, an XML string is returned to the merchant's secure server.

### 2.2.1 XML specification

An authorisation will take the XML form described in [XML response](#).

Within the `<Response>` element there is currently support for two child elements:

```
<ResponseBlock Live="TRUE" Version="3.51">
  <Response Type="AUTH">
    <OperationResponse>           </OperationResponse>
    <Order>                       </Order>
  </Response>
</ResponseBlock>
```

#### 2.2.1.1 `<OperationResponse>`

The `<OperationResponse>` element contains ten tags. The tags are returned if an acquiring bank successfully processes a transaction, i.e. if the transaction was authorised or declined.

The `<Result>` tag will be returned in all cases. The other tags may be returned depending on the outcome of processing a request. If the `<Result>` tag indicates that there was a problem with the request, the `<Message>` tags will include some information explaining the problem and the contents of the other tags may not be relied upon. The information may be available by contacting SecureTrading support.

```
<OperationResponse>
  <TransactionReference>String</TransactionReference>
  <AuthCode>String</AuthCode>
  <Result>INT</Result>
  <Message>String</Message>
  <SettleStatus>Int</SettleStatus>
  <SecurityResponseSecurityCode>Int</SecurityResponseSecurityCode>
  <SecurityResponsePostCode>Int</SecurityResponsePostCode>
  <SecurityResponseAddress>Int</SecurityResponseAddress>
  <TransactionCompletedTimestamp>String</TransactionCompletedTimestamp>
  <TransactionVerifier>String</TransactionVerifier>
</OperationResponse>
```

The `<TransactionReference>` tag contains the unique SecureTrading transaction reference for that transaction. This value is generated for each individual transaction and should be quoted in any correspondence querying the transaction.

The `<AuthCode>` tag contains the authorisation code that is returned from the acquiring bank.

The `<Result>` tag contains the authorisation outcome of the transaction, which can be in one of three groups:

- 1: Successful Authorisation
- 2: Declined Authorisation
- 0: Failed/Error Authorisation attempt (Reason supplied)

The `<Message>` tag contains any message string corresponding to the authorisation, including error messages.

The `SettleStatus` tag will contain the settlement status of the transaction.

The value of this tag will determine whether the transaction is sent for settlement or not. It can have one of the following values:

- 0: Transaction will be submitted for settlement
- 2: Transaction is suspended (wont be included for settlement)

Note: A settle status of 0 in the XML response may not guarantee that the transaction will be sent for settlement. Before the transaction is submitted for settlement certain checks are performed to determine the legitimacy of the transaction. If these checks identify a problem with the transaction, its settle status will be altered and a different value to the one returned in the response will be assigned to the transaction.

The `<SecurityResponseSecurityCode>` `<SecurityResponsePostCode>` and the `<SecurityResponseAddress>` fields contain information pertaining to the additional security checks that the acquirers perform.

The fields will only be returned if the `<Result>` is a 1.

The values in these fields will be integers providing an interpretation of the response returned from an acquirer.

For a detailed breakdown of the values returned, refer to the tables in Appendix 8.2.

For new accounts SecureTrading automatically implements a security policy that will suspend an authorisation if the `<SecurityResponseSecurityCode>` is a '4'. This default policy won't suspend an authorisation based on the values of the `<SecurityResponsePostCode>` and the `<SecurityResponseAddress>` fields, but it can be configured to do so.

If you want to change this please contact support to discuss a change to your Security Policy.

### Attention

Please be aware that some banks will decline a transaction based on an invalid security code, thus nullifying any security policy you may have in place. Other banks won't decline the transaction and will leave the decision to fulfil the transaction down to the merchant.

The `<TransactionCompletedTimestamp>` tag contains a String representing the date and time that the authorisation took place on the payment gateway.

The timestamp will be of the value "YYYY-MM-DD HH:mm:ss" where:

- YYYY      Year
- MM        Month
- DD        Day
- HH        24-hour
- mm        Minute
- ss        Second

The `<TransactionVerifier>` tag contains a String that is required when making subsequent transactions with the same payment information, e.g. a refund request or repeat payment. This tag will only contain a useful value if the transaction was successfully authorised (i.e. `<Result>` is 1). Please note, transactions made using the `<TransactionVerifier>` will not be processed using the Security Code data. In most cases this is not required, but if you deem it necessary the `<SecurityCode>` tag must be included.

### 2.2.1.2 <Order>

The <Order> element, which is returned to the merchant, consists of two tags as defined in the Authorisation Request (see <Order>). If no order tags are submitted in the request then the <Order> element will not be returned.

### 2.2.2 XML example

The following is an example of the XML string returned by the SecureTrading payment gateway.

```
<ResponseBlock Live="FALSE" Version="3.51">
  <Response Type="AUTH">
    <OperationResponse>
      <TransactionReference>1-2-2432</TransactionReference>
      <AuthCode>Auth Code:6284</AuthCode>
      <Result>1</Result>
      <SettleStatus>0</SettleStatus>
      <SecurityResponseSecurityCode>1</SecurityResponseSecurityCode>
      <SecurityResponsePostCode>2</SecurityResponsePostCode>
      <SecurityResponseAddress>4</SecurityResponseAddress>
      <TransactionCompletedTimestamp>2000-10-04
      23:24:02</TransactionCompletedTimestamp>
      <TransactionVerifier>ljhLKH6H7fjhg+764J
      ERsdfGhKJHGjhdsee09DSs+</TransactionVerifier>
    </OperationResponse>
    <Order>
      <OrderReference>Order0001</OrderReference>
      <OrderInformation>Test Order</OrderInformation>
    </Order>
  </Response>
</ResponseBlock>
```

The above is an example of a successful authorisation, illustrated by the <Result> of '1'.

## 2.3 Authorisation reversal request

An authorisation reversal request involves the cancellation of an unsettled credit card transaction.

Note: After an authorisation reversal has taken place the transaction cannot be settled.

### 2.3.1 XML specification

To initiate an authorisation reversal, the Type attribute in the <Request> element must be set to "AUTHREVERSAL".

Note that a <TransactionVerifier> tag is required in place of the card tags in the <CreditCard> element.

All other fields are identical to an authorisation request (see [Authorisation request](#)).

It may take a few minutes after authorisation for a transaction to be available in the database so that an AUTHREVERSAL request can take place. It is recommended to wait 10 minutes after authorisation before performing any ST Xpay or Xpay4 AUTHREVERSAL requests.

### 2.3.2 XML example

```
<?xml version="1.0" encoding="iso-8859-1"?>
<RequestBlock Version="3.51">
  <Request Type="AUTHREVERSAL">
    <Operation>
      <SiteReference>testref1234</SiteReference>
    </Operation>
    <CustomerInfo>
      <Postal>
        <Name>
          <NamePrefix>Mr.</NamePrefix>
          <FirstName>Joe</FirstName>
          <MiddleName>A.</MiddleName>
          <LastName>Bloggs</LastName>
          <NameSuffix>CEng.</NameSuffix>
        </Name>
        <Company>A COMPANY</Company>
        <Street>A STREET</Street>
        <City>A CITY</City>
        <StateProv>A STATE</StateProv>
        <PostalCode>A POST CODE</PostalCode>
        <CountryCode>GBR</CountryCode>
      </Postal>
      <Telecom>
        <Phone>0000 111111</Phone>
      </Telecom>
      <Online>
        <Email>CUSTOMER@SOMEDOMAIN.COM</Email>
      </Online>
    </CustomerInfo>
    <PaymentMethod>
      <CreditCard>
        <TransactionVerifier>
          ZNdfKNjnsdakSJnkSAFnkA
        </TransactionVerifier>
        <ParentTransactionReference>
          1-2-2432
        </ParentTransactionReference>
      </CreditCard>
    </PaymentMethod>
    <Order>
      <OrderReference>Order0001</OrderReference>
      <OrderInformation>Additional cost</OrderInformation>
    </Order>
  </Request>
  <Certificate>
    -----BEGIN CERTIFICATE-----
    MIIGUTCCBg+gAwIBAgICAQMwCwYHKOZIZjgEAwUAMIG/MQswCQYDVQQGEwJVSzEP
    MA0GA1UECBMTG9uZG9uMQswCQYDVQQHEwJTVDEWMBQGA1UEChMNU2VjdXJlVHJH
    ZGluZzEuMCwGA1UEC9MlVGVzZdCBNzXJjaGFudCBDZXJ0aWZpY2F0aW9uIEF1dGhV
    SPEeso/HNlkY8DXrEXixAhRcJrEC3JnDXLRDZc32MciHvISP9Q==
    -----END CERTIFICATE-----
    -----BEGIN RSA PRIVATE KEY-----
    TWhG9Qn9kJ+h15rJBvbQtNvFMBuLlIXEVOxZboteE+2lW7iYz5gF7HWUPgjm+RTM
    j6D5al6AuWU0/uqL6VVe6POgBorSFeWdl8RnQLfAinVkeEu8UkWQBS1Y4j5ICFGRc
    YaJAnR5AJ4xXUK3CHVbLEBkW
    -----END RSA PRIVATE KEY-----
  </Certificate>
<!--NOTE for ST Xpay4 the Certificate tag should just contain your certificate alias -->
</RequestBlock>
```

## 2.4 Authorisation reversal response

Upon processing an authorisation reversal, the results are returned to the ST Xpay or Xpay4 client for merchant processing.

### 2.4.1 XML specification

An authorisation reversal response is identical to an authorisation response message (see [Authorisation response](#)) except for the <Response> element that has the Type attribute set to "AUTHREVERSAL".

### 2.4.2 XML example

The following is an example of the XML string returned by the SecureTrading payment gateway.

```
<ResponseBlock Live="TRUE" Version="3.51">
  <Response Type="AUTHREVERSAL">
    <OperationResponse>
      <TransactionReference>2-2-35117</TransactionReference>
      <AuthCode>Auth code:6712</AuthCode>
      <Result>1</Result>
      <Message></Message>
      <TransactionCompletedTimestamp>2000-11-04
        14:54:10</TransactionCompletedTimestamp>
    </OperationResponse>
    <Order>
      <OrderReference>Order0001</OrderReference>
      <OrderInformation>Additional Cost</OrderInformation>
    </Order>
  </Response>
</ResponseBlock>
```

## 3 Transaction refunds

This section details the XML required when submitting a refund request and the XML response.

### 3.1 Refund request

A refund request involves the crediting of a credit card after the settlement of a successful authorisation.

#### 3.1.1 XML specification

To initiate a refund request the `<Request>` element attribute `Type` must be set to "REFUND". The element can contain up-to three child elements:

```
<Request Type="REFUND">
  <Operation>           </Operation>
  <PaymentMethod>      </PaymentMethod>
  <CustomerInfo>       </CustomerInfo>
</Request>
```

The three child elements do not take any attributes. Of the three only `<Operation>` and `<PaymentMethod>` are required.

The character data within a tag must be of the correct data-type as specified. For an explanation of the data-types, please refer to the [Glossary of terms](#).

##### 3.1.1.1 `<Operation>`

The `<Operation>` element includes two tags:

```
<Operation>
  <Amount>INT</Amount>
  <SiteReference>String</SiteReference>
</Operation>
```

The `<Amount>` and `<SiteReference>` tags are required.

The `<Amount>` tag should contain the value of the refund transaction in base units of the original authorisation currency. There must be no decimal point or separators, For example. £10.43 would be sent as `<Amount>1043</Amount>`.

#### Important

A refund transaction is only permissible if it is equal to or less than the original authorisation amount.

A refund transaction can be for more than the original authorisation amount. To have this enabled, contact SecureTrading support.

The `<SiteReference>` tag should be the same site reference as in the original authorisation, for example: `<SiteReference>testref1234</SiteReference>`

##### 3.1.1.2 `<PaymentMethod>`

The `<PaymentMethod>` contains a single child element `<CreditCard>`.

The `<CreditCard>` element includes two tags:

```

<PaymentMethod>
  <CreditCard>
    <ParentTransactionReference>String</ParentTransactionReference>
    <TransactionVerifier>String</TransactionVerifier>
  </CreditCard>
</PaymentMethod>

```

Both the `<ParentTransactionReference>` and `<TransactionVerifier>` tags are required.

The `<ParentTransactionReference>` tag should contain the transaction reference of the transaction to be refunded. This was supplied in the `<TransactionReference>` tag of the original authorisation response.

The `<TransactionVerifier>` tag should contain the Transaction Verifier of the transaction to be refunded. This was supplied in the `<TransactionVerifier>` tag of the original authorisation response.

### 3.1.1.3 `<CustomerInfo>`

Please refer to the definition in the authorisation request section, see [<CustomerInfo>](#).

## 3.1.2 XML example

The following is an example of the XML string sent to the ST Xpay or Xpay4 client.

```

<?xml version="1.0" encoding="iso-8859-1"?>
<RequestBlock Version="3.51">
  <Request Type="REFUND">
    <Operation>
      <Amount>1000</Amount>
      <SiteReference>testref1234</SiteReference>
    </Operation>
    <CustomerInfo>
      <Postal>
        <Name>
          <NamePrefix>Mr.</NamePrefix>
          <FirstName>Joe</FirstName>
          <MiddleName>A.</MiddleName>
          <LastName>Bloggs</LastName>
          <NameSuffix>CEng.</NameSuffix>
        </Name>
        <Company>A COMPANY</Company>
        <Street>A STREET</Street>
        <City>A CITY</City>
        <StateProv>A STATE</StateProv>
        <PostalCode>TE2 3ST</PostalCode>
        <CountryCode>GBR</CountryCode>
      </Postal>
      <Telecom>
        <Phone>0000 111111</Phone>
      </Telecom>
      <Online>
        <Email>CUSTOMER@SOMEDOMAIN.COM</Email>
      </Online>
    </CustomerInfo>
    <PaymentMethod>
      <CreditCard>
        <ParentTransactionReference>1-2-432</ParentTransactionReference>
        <TransactionVerifier>XDJ3LVKAJ3PSDdfslkh3HERsdFGhKJHGjhdsee09DSS+</TransactionVerifier>
      </CreditCard>
    </PaymentMethod>
  </Request>
</RequestBlock>

```

```

    </PaymentMethod>
  </Request>
  <Certificate>
  -----BEGIN CERTIFICATE-----
MIIGUTCCBg+gAwIBAgICAQMwCwYHKoZIzjgEAwUAMIG/MQswCQYDVQQGEwJVSzEP
MA0GA1UECBMTG9uZG9uMQswCQYDVQQHEwJTVDEWMBQGA1UEChMNU2VjdXJlVHJh
ZGluZzEuMwCwGA1UECzMlVGZzdCBNZXJjaGFudCBDZSJ0aWZpY2F0aW9uIEF1dGhv
SPEeso/HN1kY8DXrEXixAhRcJrEC3JnDXLRDZc32MciHvISP9Q==
-----END CERTIFICATE-----
-----BEGIN RSA PRIVATE KEY-----
TWhG9Qn9kJ+h15rJBvbQtNvFMBuLlIXEVOxZboteE+2lW7iYz5gF7HWUPgjm+RTM
j6D5a16AuWU0/uqL6VVe6POgBorSFeWd18RnQLfAinVkeu8UkWQBS1Y4j5ICFGRc
YaJAnR5AJ4xXUK3CHVbLEBkW
-----END RSA PRIVATE KEY-----
  </Certificate>
<!--NOTE for ST Xpay4 the Certificate tag should just contain your certificate alias -->
</RequestBlock>

```

## 3.2 Refund response

Upon processing a refund the result is returned to ST Xpay or Xpay4.

### 3.2.1 XML specification

The XML String for a Refund Response is identical in format to the Authorisation Response (see [Authorisation response](#)) except for the <Response> attribute, "Type", which will have the value of "REFUND". Note the version number in the response will remain the same.

### 3.2.2 XML example

The following is an example of the XML string returned by the SecureTrading payment gateway.

```

<ResponseBlock Live="FALSE" Version="3.51">
  <Response Type="REFUND">
    <OperationResponse>
      <TransactionReference>1-2-32432</TransactionReference>
      <AuthCode>Auth code:6284</AuthCode>
      <Result>1</Result>
      <Message>None</Message>
      <TransactionCompletedTimestamp>2000-10-04
      23:24:02</TransactionCompletedTimestamp>
      <TransactionVerifier>SKTK2KSOD8762KSKJHksdjfk2ksdkfj2P
      SDFKH6Kkjh8732=</TransactionVerifier>
    </OperationResponse>
  </Response>
</ResponseBlock>

```

The above is an example of a successful refund, illustrated by the <Result> of '1'.

## 3.3 Refund reversal request

A refund reversal involves reversing a refund on an unsettled refund request.

### 3.3.1 XML specification

To initiate a refund reversal the <Request> element attribute Type must be set to "REFUNDREVERSAL".

Notice how the <TransactionVerifier> replaces the need for Credit Card details. All other fields are identical to an authorisation reversal request (see [Authorisation reversal request](#)).

### 3.3.2 XML example

The following is an example of the XML string sent to the SecureTrading payment gateway.

```
<?xml version="1.0" encoding="iso-8859-1"?>
<RequestBlock Version="3.51">
  <Request Type="REFUNDREVERSAL">
    <Operation>
      <SiteReference>testref1234</SiteReference>
    </Operation>
    <CustomerInfo>
      <Postal>
        <Name>
          <NamePrefix>Mr.</NamePrefix>
          <FirstName>Joe</FirstName>
          <MiddleName>A.</MiddleName>
          <LastName>Bloggs</LastName>
          <NameSuffix>CEng.</NameSuffix>
        </Name>
        <Company>A COMPANY</Company>
        <Street>A STREET</Street>
        <City>A CITY</City>
        <StateProv>A STATE</StateProv>
        <PostalCode>TE2 3ST</PostalCode>
        <CountryCode>GBR</CountryCode>
      </Postal>
      <Telecom>
        <Phone>0000 111111</Phone>
      </Telecom>
      <Online>
        <Email>CUSTOMER@SOMEDOMAIN.COM</Email>
      </Online>
    </CustomerInfo>
    <PaymentMethod>
      <CreditCard>
        <TransactionVerifier>ZNdfKNjnsdakSJnkSAFnkA
        </TransactionVerifier>
        <ParentTransactionReference>1-2-
        2432</ParentTransactionReference>
      </CreditCard>
    </PaymentMethod>
    <Order>
      <OrderReference>Order0001</OrderReference>
      <OrderInformation>Additional cost</OrderInformation>
    </Order>
  </Request>
  <Certificate>
    -----BEGIN CERTIFICATE-----
    MIIGUTCBBg+gAwIBAgICAQMwCwYHKoZIzjgEAwUAMIG/MQswCQYDVQQGEwJVSzEP
    MA0GA1UECBMTG9uZG9uMQswCQYDVQQHEwJTVDEWMBQGA1UEChMNU2VjdXJlVHJh
    ZGluZzEuMwCwGA1UECzMlVGZzdCBNZXJjaGFudCBkdXJ0aWZpY2F0aW9uIEF1dGhV
    SPEeso/HNlkY8DXrEXixAhRcJrEC3JnDXLRDZc32MciHvISP9Q==
    -----END CERTIFICATE-----
    -----BEGIN RSA PRIVATE KEY-----
    TWhG9Qn9kJ+h15rJBvbQtNvFMBuLlIXEVOxZboteE+2lW7iYz5gF7HWUPgjm+RTM
    j6D5a16AuWU0/uqL6VVe6POgBorSFeWd18RnQLfAinVkeu8UkQWBS1Y4j5ICFGRc
    YaJAnR5AJ4xXUK3CHVbLEBkW
    -----END RSA PRIVATE KEY-----
  </Certificate>
  <!--NOTE for ST Xpay4 the Certificate tag should just contain your certificate alias -->
</RequestBlock>
```

## 3.4 Refund reversal response

Upon processing a refund reversal the result is returned to ST Xpay or Xpay4.

### 3.4.1 XML specification

A refund reversal response is identical to an authorisation reversal response (see [Authorisation reversal response](#)) except for the <Response> element, which has the `Type` attribute set to "REFUNDREVERSAL"

### 3.4.2 XML example

The following is an example of the XML string returned by the SecureTrading payment gateway.

```
<ResponseBlock Live="FALSE" Version="3.51">
  <Response Type="REFUNDREVERSAL">
    <OperationResponse>
      <TransactionReference>2-2-35117</TransactionReference>
      <AuthCode>Auth code:6712</AuthCode>
      <Result>1</Result>
      <TransactionCompletedTimestamp>2000-11-04
        14:54:10</TransactionCompletedTimestamp>
    </OperationResponse>
    <Order>
      <OrderReference>Order0001</OrderReference>
      <OrderInformation>Additional Cost</OrderInformation>
    </Order>
  </Response>
</ResponseBlock>
```

## 4 Settlement control

Detailed here is the request and response XML used when processing or altering settlement requests. Please note that unless altered settlement will normally occur automatically.

### 4.1 Settlement request

Settlement involves the transfer of authorised funds to the merchant's bank account.

#### 4.1.1 XML specification

To initiate a settlement request the `Type` attribute within the `<Request>` element must be set to "SETTLEMENT". This element can contain one child element:

```
<Request Type="SETTLEMENT">
  <Operation> </Operation>
</Request>
```

For an explanation of the data-types used in a settlement request, please refer to the [Glossary of terms](#).

##### 4.1.1.1 <Operation>

The `<Operation>` element contains five tags, all of which are required:

```
<Operation>
  <SiteReference>String</SiteReference>
  <TransactionReference>String</TransactionReference>
  <SettleDate>String</SettleDate>
  <SettleStatus>INT</SettleStatus>
  <SettleAmount>INT</SettleAmount>
</Operation>
```

The `<SettleDate>` tag can take one of the following values:

"NEXT": Indicates that the transaction should be sent for settlement in the next available settlement window.  
Normally the transaction is sent for settlement that night.

Date: Indicates that the transaction will be included in the given date's settlement window

The `Date` value must be of the value "YYYY-MM-DD" where:

- YYYY      Year
- MM        Month
- DD        Day

i.e. to set transaction settlement to take place on 31 January 2004 then the `<SettleDate>` should be set to:

```
<SettleDate>2004-01-31</SettleDate>
```

The `<SettleStatus>` tag can take one of the following values:

0:        The transaction is pending settlement and will be included in the fraud checking system  
(This is the default value for all transactions)

- 1: The transaction is pending settlement and won't be included in the fraud checking system<sup>2</sup>
- 2: The transaction is suspended

The <SettleAmount> value should be the amount, of the original authorisation, in base units of the currency specified in the <Currency> tag, i.e. there must be no decimal point or separators. Normally this is the same value as that used in the original transaction.

#### IMPORTANT NOTE

A transaction can only have its settlement details changed if the status of the transaction is one of the following values:

- 0: The transaction is pending settlement and will be included in the fraud checking system. (This is the default value for all transactions)
- 1: The transaction is pending settlement and will by-pass the fraud checking system
- 2: The transaction is suspended
- 55: Transaction is suspected as a duplicate and has been suspended<sup>3</sup>.

The initial settlement status of a transaction is dependent on whether the <SettlementDay> tag for the original authorisation was set to defer the settlement, i.e. if <SettlementDay> for the original authorisation is set to 0 then the status of the transaction will be set to 2 (it will not get settled until manual intervention), otherwise the status is set to 0 (default) and will settle according to the date set in <SettleDate>.

It may take a few minutes after authorisation for a transaction to be available in the database so that a SETTLEMENT request can take place. It is recommended to wait 10 minutes after authorisation before performing any ST Xpay or Xpay4 SETTLEMENT requests.

#### 4.1.2 XML example

The following is an example of the XML string sent to the SecureTrading payment gateway.

```
<?xml version="1.0" encoding="iso-8859-1"?>
<RequestBlock Version="3.51">
  <Request Type="SETTLEMENT">
    <Operation>
      <SiteReference>testref1234</SiteReference>
      <TransactionReference>1-2-9432</TransactionReference>
      <SettleDate>NEXT</SettleDate>
      <SettleStatus>1</SettleStatus>
      <SettleAmount>499</SettleAmount>
    </Operation>
  </Request>
  <Certificate>
    -----BEGIN CERTIFICATE-----
    ZGluZzEuMCwGA1UECxMlVGZzdCBNZXJjaGFudCBDZXJ0aWZpY2F0aW9uIEF1dGhv
    SPEeso/HNlkY8DXrEXixAhRcJrEC3JnDXLRDZc32MciHvISP9Q==
    -----END CERTIFICATE-----
    -----BEGIN RSA PRIVATE KEY-----
    j6D5a16AuWU0/uqL6VVe6POgBorSFeWd18RnQLfAinVkeu8UkQBS1Y4j5ICFGRc
    YaJAnR5AJ4xXUK3CHVbLEBkW
    -----END RSA PRIVATE KEY-----
  </Certificate>
  <!--NOTE for ST Xpay4 the Certificate tag should just contain your certificate alias -->
</RequestBlock>
```

<sup>2</sup> It is useful to have this additional status so merchants can settle a transaction that was mistakenly flagged as a fraudulent. If a transaction was re-set back to status 0 the fraud check would again flag it as a potentially fraudulent, thus preventing the transaction from ever being settled.

<sup>3</sup> Every transaction is checked to determine if it is a duplicate transaction. A duplicate transaction normally occurs if a customer submits the same payment more than once, usually by repeated refreshing of the credit card submission page before the transaction has completed.

## 4.2 Settlement response

Upon processing a settlement request the result is returned to ST Xpay or Xpay4.

### 4.2.1 XML specification

As in the previous XML responses the response contains a `<ResponseBlock>` within which there is one `<Response>` element. The `<Response>` element currently has support for two child elements:

```
<ResponseBlock Live="TRUE" Version="3.51">
  <Response Type="SETTLEMENT">
    <OperationResponse>           </OperationResponse>
  </Response>
</ResponseBlock>
```

#### 4.2.1.1 `<OperationResponse>`

The `<OperationResponse>` element will contain nine tags:

```
<OperationResponse>
  <TransactionReference>String</TransactionReference>
  <SettleDate>String</SettleDate>
  <SettleAmount>INT</SettleAmount>
  <Currency>String</Currency>
  <SettleStatus>INT</SettleStatus>
  <SettledTimestamp>String</SettledTimestamp>
  <SettledAmount>INT</SettledAmount>
  <Result>INT</Result>
  <Message>String</Message>
</OperationResponse>
```

The `<TransactionReference>` tag included in the response will always contain the same data as sent in the settlement request. Because any particular response is for a corresponding request, this field will remain unchanged.

The `<SettleDate>` tag will contain one of the following values:

`"SUSPEND"`: Indicates that the transaction has been suspended and will not be settled.  
 Date: Indicates the date that the transaction will be settled.

The Date value will be in the same format as previously defined in the settlement request section. See [Settlement request](#).

The `<Currency>` tag will contain the currency of the transaction. The format of `<Currency>` is defined in the authorisation request section. See [Authorisation request](#).

The `<SettleAmount>` tag will contain the amount that is to be settled. This will be in the base units of the original transaction currency.

The `<SettleStatus>` tag will contain one of the following values:

- 0: Transaction is pending settlement and will be included in the fraud checking system (This is the default value for all transactions)
- 1: Transaction is pending settlement and will by-pass the fraud checking systems
- 2: The transaction is suspended

- 3: The transaction is cancelled<sup>4</sup>
- 10: The transaction is set for settlement and has been included in the next settlement batch
- 55: The transaction has been authorised, but is deemed a duplicate
- 100: The acquiring bank has settled the transaction. Confirmation has been received from the acquiring bank

In your XML request, if you include the `<SiteReference>` tag and a `<TransactionReference>` tag but leaving all other tags within the `<Operation>` tag blank then the `<OperationResponse>` that is returned will include the current values for the transaction.

The `<SettledTimestamp>` tag will contain the date when the transaction was settled. The timestamp will be of the value "YYYY-MM-DD HH:mm:ss" where:

- YYYY Year
- MM Month
- DD Day
- HH 24-hour
- mm Minute
- ss Second

If the transaction has not been settled then this tag will contain no value.

The `<SettledAmount>` tag will contain the amount that was settled once the transaction has a `<SettleStatus>` of 100. If no request to change the amount to be settled is given then this amount will be the same as the `<SettledAmount>` tag value.

The `<Result>` tag will contain one of the following values:

- 0: Error in processing settlement request
- 1: Settlement request approved
- 2: Settlement request declined

The `<Message>` tag will include any error messages or other responses that are returned upon processing a settlement request.

## 4.2.2 XML example

The following is an example of the XML string returned by the SecureTrading payment gateway.

```
<ResponseBlock Live="TRUE" Version="3.51">
  <Response Type="SETTLEMENT">
    <OperationResponse>
      <TransactionReference>1-2-3248</TransactionReference>
      <SettleDate>2000-11-08</SettleDate>
      <Currency>GBP</Currency>
      <SettleAmount>499</SettleAmount>
      <SettleStatus>0</SettleStatus>
      <SettledTimestamp>2000-11-08 23:30:58</SettledTimestamp>
      <SettledAmount>499</SettledAmount>
      <Result>1</Result>
    </OperationResponse>
  </Response>
</ResponseBlock>
```

The above is an example of the response to a successful settlement request, illustrated by the `<Result>` of '1'.

<sup>4</sup> Cancelled transactions cannot be altered in any way. They are only displayed for completeness.

## 5 Card Check (BinLookup)

Detailed here is the request and response XML used when processing a BinLookup request.

### 5.1 BinLookup request

A BinLookup request involves submitting a card bin number (usually the first 6 numbers of a payment card).

#### 5.1.1 XML specification

To initiate a BinLookup request the `Type` attribute within the `<Request>` element must be set to "BINLOOKUP". This element can contain one child element:

```
<Request Type="BINLOOKUP">
  <Operation> </Operation>
  <Filter> </Filter>
</Request>
```

For an explanation of the data-types used in a binlookup request, please refer to the [Glossary of terms](#).

##### 5.1.1.1 <Operation>

The `<Operation>` element contains five tags, all of which are required:

```
<Operation>
  <SiteReference>String</SiteReference>
</Operation>
```

##### 5.1.1.2 <Filter>

The `<Filter>` element contains one tag, which is required:

```
<Filter>
  <BinNumber>String</BinNumber>
</Filter>
```

#### 5.1.2 XML Example

```
<RequestBlock Version="3.51">
  <Request Type="BINLOOKUP">
    <Operation>
      <SiteReference>mysite1234</SiteReference>
    </Operation>
    <Filter>
      <BinNumber>409432</BinNumber>
    </Filter>
  </Request>
  <Certificate></Certificate>
</RequestBlock>
```

## 5.2 BinLookup response

A BinLookup response

### 5.2.1 XML specification

A BinLookup response has some elements similar to a transaction response, these include TransactionCompletedTimeStamp, TransactionReference and Result.

#### 5.2.1.1 <OperationResponse>

The <OperationResponse> element will contain nine tags:

```
<ResponseBlock Live="TRUE" Version="3.51">
  <Response Type="BINLOOKUP">
    <OperationResponse>
      <TransactionCompletedTimeStamp>    </TransactionCompletedTimeStamp>
      <TransactionReference>    </TransactionReference>
      <Result>    </Result>
      <Found>    </Found>
      <Bin>
        <Country>    </Country>
        <Number>    </Number>
        <Issuer>    </Issuer>
        <CardType>    </CardType>
      </Bin>
    </OperationResponse>
  </Response>
</ResponseBlock>
```

In addition to these the BinLookup response has two additional tags within the OperationResponse these are, the Found tag and the Bin tag.

The found tag contains the number of bins that matched the bin that was given within the request and the Bin tag contains three additional child tags that contain the details of the matched bin.

#### 5.2.1.1.1 <Bin>

The bin tag contains three child tags, the country tag which contains the country found for bin given within the request, the Issuer tag which contains the Issuer of the requested bin and the bin tag that contains the bin number that was used to perform the match.

### 5.2.2 XML Example

```
<ResponseBlock Live="TRUE" Version="3.51">
  <Response Type="BINLOOKUP">
    <OperationResponse>
      <TransactionCompletedTimeStamp>2009-05-01
11:04:51</TransactionCompletedTimeStamp>
      <TransactionReference>222-36-1</TransactionReference>
      <Result>1</Result>
      <Found>1</Found>
      <Bin>
        <Country>UNITED KINGDOM</Country>
        <Number>411111000</Number>
        <Issuer>NATWEST ISSUER</Issuer>
        <CardType>Visa</CardType>
      </Bin>
    </OperationResponse>
  </Response>
</ResponseBlock>
```

---

## 6 Further information

This section contains contact information relevant to ST Xpay and Xpay4.

### 6.1 Support

SecureTrading provides support for its software and the operation of its payment service. If you require technical support, first ensure that you have read and understood all relevant documentation.

If the problem persists, please email [support@securetrading.com](mailto:support@securetrading.com), quoting your SecureTrading sitereference and concisely stating the nature of your problem.

To help us help you, please include the original XML string sent and any error messages that are returned by the ST Xpay API verbatim.

Note: Before sending any information to SecureTrading, care should be taken to remove any sensitive information, such as the credit card number.

SecureTrading additional contact details:

Phone: 01248 672 050

Fax: 01248 672 099

### 6.2 Further reading

For further information please refer to the following documents:

In the general setup guides (<http://www.securetrading.com/support/general-setup-guides.html>) section of the SecureTrading website:

- Going live document:
- Address Verification (AVS) and security code guide:
- SecureTrading testing document:

In the ST Xpay documents (<http://www.securetrading.com/support/xpay1.html>) section of the SecureTrading website:

- SecureTrading Xpay user guide

Bundled with the ST Xpay distribution:

- ST Xpay read me: readme.txt

New features and request types will be added to ST Xpay. Information on these features will be included in newer versions of this document or separate documentation will be provided. The ST Xpay section can be found on the SecureTrading web site:

<http://www.securetrading.com/support/xpay1.html>

---

## 7 Glossary of terms

### ASCII

ASCII is a global standard of identifying characters.

### AUTHORISATION

Authorisation is the process of validating a credit or debit card transaction with an acquiring bank. Authorisation allocates the transaction amount on a customer's credit card but no money is debited from the customer's credit card account.

### AUTHORISATION REVERSAL

Authorisation reversal is in effect the opposite of an authorisation. To perform an authorisation reversal an authorisation must have already been sent to an acquirer. Authorisation reversal cancels the allocation of funds on a credit card for settlement. Note: only un-settled authorisations may be reversed. Settled transactions must be refunded instead.

### CHAR

Character type. This is a single ASCII character, e.g. 'A'

### INT

Integer type. This is a whole number, which can be positive, negative or zero.

### PAYMENT GATEWAY

A payment gateway can be thought of as a secure, reliable bridge from a merchant's web site or server to the acquiring banks.

### REFUND

A refund is a two-stage process. Firstly, a request is given to transfer monies from a merchant's account to the customer's credit card account for a previously settled transaction. The money is merely allocated; no money is credited to the account until the second stage. Secondly, the monies are transferred from the merchant to the customer's credit card account.

### REFUND REVERSAL

A refund reversal is the same as an authorisation reversal but applies to refund transactions. Note only un-settled transactions may be reversed.

### SECURITY CODE

The additional three or four digit number printed on a credit card. Card issuers use a Cardholder Verification system to check this.

### SETTLEMENT

Settlement is the process of debiting the transaction amount from a customer's card and crediting into the merchant's account, or vice versa in the case of a refund settlement. Settlement always follows an authorisation or refund request.

### STRING

A string is a sequence of characters, For example. 'Hello World.' The maximum permitted string length when passing an XML string to SecureTrading is 255 characters.

### TCP/IP

TCP/IP is a network / communication protocol commonly used by the internet. TCP/IP uses port numbers to identify the different processes when they arrive at a server.

### XML

eXtensible Markup Language provides a structured method of defining data. For further information on XML, please refer to <http://www.xml.org/>

## 8 APPENDICES

### 8.1 Xpay and Xpay4 error codes

The following table lists the possible error codes that ST Xpay ONLY can return if there are any problems in sending or receiving requests / responses.

Error Code	Message	Description
100	Timeout reading from socket	ST Xpay did not receive the full XML in the given time from your application.
101	Error reading from socket	ST Xpay did not receive a correct connection request from your application.
1000	Failed to connect to a payment gateway	The ST Xpay client was unable to find a payment gateway.
1100	Failed to receive from payment gateway	ST Xpay did not receive any response from the payment gateway.

The following table lists the possible error codes that ST Xpay and Xpay4 can return if there are any problems in sending or receiving requests / responses.

2100	Missing SiteReference or Certificate tag	The XML ST Xpay received was missing the site reference and/or your secure ST Xpay certificate
2500	[Various messages - One of the fields contains an invalid value]	The information returned contains the field name that was omitted from the request or contained incorrect information.
3000	Gateway error	A SecureTrading payment gateway failed to receive a full ST Xpay request.
3010	Transaction Not Received Successfully	A SecureTrading payment gateway obtained the request but failed to decrypt it.
<b>The following Error Code can have multiple messages</b>		
3100	Error with transaction details	The data retrieved was incorrectly defined.
3100	Error in XML: Unknown Transaction Type	The Request Type attribute was incorrect.
3100	Error Parsing XML (invalid site reference for this certificate)	The certificate included in the XML request is invalid for the SecureTrading sitereference included in the XML
3100	Gateway Connection Error	The data obtained by ST Xpay was incorrect
3100	Error parsing XML: [+reason]	The gateway obtained invalid XML
3100	Invalid Merchant Configuration	Merchant data on the gateway is inconsistent with merchant information in the XML (For example, invalid currency, payment type, etc)
3330	Transaction storage failure. Please try again later	Gateway failed to store the transaction details before authorisation.
3350	Transaction acceptance failure. Please try again later	Gateway failed to update the transaction details after performing the authorisation
5000	Transport Error	Failed to connect to acquiring bank or the acquiring bank did not respond
5100	Missing TransactionReference	The transaction cannot be found in the database.

The following table lists the possible error codes that ST Xpay4 ONLY can return if there are any problems in sending or receiving requests / responses.

10500	Various Message	Xpay4 was unable to process your request
-------	-----------------	--

## 8.2 Security check responses

The following section details the values returned by the <SecurityResponseSecurityCode>, <SecurityResponseAddress> and <SecurityCodePostCode> fields as well as their meanings.

### Security Code Security Response

Response Value	Response Text
0	NO INFORMATION AVAILABLE
1	DATA NOT CHECKED
2	DATA MATCHED
3	SHOULD HAVE SECURITY CODE
4	DATA NOT MATCHED

### Security Code Address and Security Code PostCode

Response Value	Response Text
0	NO INFORMATION AVAILABLE
1	DATA NOT CHECKED
2	DATA MATCHED
4	DATA NOT MATCHED
8	PARTIAL MATCH

### Description of Security Responses

Response Text	Description
DATA NOT CHECKED	The data submitted to the acquiring bank was not checked. Either the details were not passed or the acquiring bank couldn't perform the check based on the card details.
DATA MATCHED	The information sent to the acquiring bank was valid and has passed their security checks
DATA NOT MATCHED	The information sent to the acquiring bank was invalid and has failed their security checks
PARTIAL MATCH	Part of the information sent to the acquiring bank was valid while the remainder was either not checked or invalid.
SHOULD HAVE SECURITY CODE	The card entered has a security code but no security code was supplied.