



## XML Specification: Subscriptions

This document outlines the XML required to submit Subscription requests to the Secure Trading Subscription Engine.

**Published: 25 October 2017**

**Table of Contents**

- 1 Introduction ..... 3**
  - 1.1 Before you start ..... 3
  - 1.2 XML summary ..... 3
  - 1.3 Parent XML request types..... 4
  - 1.4 Process overview ..... 4
- 2 Subscription fields ..... 5**
  - 2.1 Unit ..... 5
  - 2.2 Frequency ..... 5
  - 2.3 Final number ..... 5
  - 2.4 Number (optional in requests)..... 6
  - 2.5 Type ..... 6
  - 2.6 Begin date (optional in requests) ..... 7
- 3 Scheduling subscriptions ..... 8**
  - 3.1 XML Request ..... 8
  - 3.2 XML Response..... 12
- 4 Querying subscriptions ..... 16**
  - 4.1 XML Request ..... 16
  - 4.2 XML Response..... 16
- 5 Updating subscriptions ..... 19**
  - 5.1 XML Request ..... 19
  - 5.2 XML Response..... 24
- 6 Subscription Notifications ..... 25**
  - 6.1 Subscription Reports ..... 25
  - 6.2 Error Reports..... 26
  - 6.3 URL Notifications ..... 26
- 7 Additional Notes ..... 27**
  - 7.1 Pending subscriptions ..... 27
  - 7.2 Declined Authorisation ..... 27
  - 7.3 Fraud and duplicate checks ..... 27
  - 7.4 End of month subscription payments ..... 27
  - 7.5 About MyST ..... 27
  - 7.6 Expiry dates ..... 28
  - 7.7 Accepted payment types..... 28
  - 7.8 3-D Secure ..... 28
  - 7.9 Direct Currency Conversion (DCC)..... 28
- 8 Further information and Support ..... 29**
  - 8.1 Secure Trading Support ..... 29
  - 8.2 Secure Trading Sales..... 29
  - 8.3 Useful Documents ..... 29
  - 8.4 Frequently Asked Questions ..... 29
- 9 Appendix ..... 30**

## 1 Introduction

Using our subscription engine, you are able to schedule payments to be processed automatically at regular intervals e.g. payments can be scheduled to be processed on the first day of each month for £10.00.

### 1.1 Before you start

- # **Your merchant ID must be capable of processing recurring or continuous authority payments.** If in doubt, we strongly recommend contacting your acquiring bank for clarification.
- # Before implementing subscriptions, we strongly recommend that you first read our [XML Specification](#) and ensure that you understand how regular authorisations are processed. This document assumes a basic understanding of how this process works.
- # We also recommend reading our [Subscription Engine document](#) for information on how subscription payments are processed on your behalf.
- # This document contains XML examples throughout, but you may prefer to download XML examples separately. These can be found in the STAPI client files within the examples folder which can be downloaded from our website:  
[http://webapp.securetrading.net/examples/STAPI\\_JAVA1.8.zip](http://webapp.securetrading.net/examples/STAPI_JAVA1.8.zip)

### 1.2 XML summary

This specification will explain the process of scheduling subscriptions in terms of **parent** and **child** XML Requests and Responses. These are described as follows:

- # **The parent XML Request** is processed first. It contains information about the authorisations that will be performed (e.g. the amount, currency, billing address, etc.). This can be a request type of **AUTH** or **ACCOUNTCHECK**. We explain the differences between these request types in the next section.
- # **The child XML Request** is processed second. It inherits information from the parent and uses this to schedule payments in our subscription engine. This has a request type of **SUBSCRIPTION**.

The two requests are submitted together in a single XML Request block. An XML Response is returned to your system, which your system will need to handle according to our specification, which is detailed within this document.

We will automatically process authorisations using the payment details provided in the parent request, at the regular intervals defined in the child request, without the need to submit further XML Requests.

### 1.3 Parent XML request types

There are two types of XML Request that can be used as the parent, both serving different purposes. Please read the information provided below and ensure you are using the most appropriate request type for your requirements.

#### 1.3.1 AUTH

AUTH XML Requests instruct us to process an authorisation immediately, the funds of which are typically settled into your account within 24 hours.

Use an AUTH request type as the parent when you would like the first payment in the subscription to be processed immediately.

#### 1.3.2 ACCOUNTCHECK (only available with select acquirers)

ACCOUNTCHECK XML Requests are similar to AUTH XML Requests except that they **do not reserve any funds**. They are instead used to perform Address Verification Service (AVS) and security code checks on the customer's card.



Account checks are only available for certain acquiring banks. Please contact our Support team for further details (see section 8.1).

Use an ACCOUNTCHECK request type as the parent when you would like to charge the customer after the first interval has passed (e.g. first month is free).

### 1.4 Process overview

#### Step 1

The customer opts to perform a subscription payment on your website.

#### Step 2

Your system submits a single XML Request, which processes an authorisation and requests that a subscription is scheduled in our subscription engine (see section 3 for specification). Subscriptions are only scheduled when the parent transaction was processed successfully.

#### Step 3

We will return an XML Response. We recommend that you store this for future reference. This information can be used to query future payments using a transaction query request (see section 4).

#### Step 4

Your system can then display confirmation of the successful payment and the subscription in the customer's browser.

## 2 Subscription fields

To schedule the subscription in the subscription engine, you will need to understand the definition of each subscription field, as these will be included in the XML Requests and Responses sent between your system and Secure Trading.

### 2.1 Unit

This field represents the unit of time between each subscription. This can be either 'DAY' or 'MONTH'.



It is imperative that this field is submitted to the gateway in CAPITALS ("DAY" or "MONTH").

### 2.2 Frequency

Combined with unit, the frequency integer defines how frequently payments are processed.

e.g. For one payment every 7 days:

`frequency = 7`  
`unit = DAY`

e.g. For one payment every 2 months:

`frequency = 2`  
`unit = MONTH`

### 2.3 Final number

This is the number of payments to be processed over the course of the subscription.

#### 2.3.1 If the parent XML Request is an AUTH

If `finalnumber = 12`, there will be 12 monthly payments in total (the initial AUTH + 11 subscription payments).

#### 2.3.2 If the parent XML Request is an ACCOUNTCHECK

If `finalnumber = 12`, there will be 11 monthly payments in total (the initial ACCOUNTCHECK + 11 subscription payments).

The initial ACCOUNTCHECK counts against the final number.



If the value is 0, the subscription engine will schedule payments indefinitely until the user manually sets the subscription to **Inactive** (as described in section 5.1.6).

## 2.4 Number (optional in requests)

Unless specified otherwise, subscriptions start with `number = 1` by default. The `number` is automatically incremented in every subsequent subscription payment until it exceeds the value of the `finalnumber` field, when no further payments will be attempted. A completed subscription is represented by a subscription `number` that is higher than the corresponding `finalnumber`.

### 2.4.1 Changing the starting number

You can change the starting number in the parent XML Request by submitting the `number` element with a numeric value. This is useful when restarting a previously cancelled subscription and you would like to maintain the number of payments processed.

e.g. Submitting `number = 5` will mean the subscription number will start from 5 instead of the default value of 1.



The parent XML Request should always start with `number=1`, unless resuming a previous subscription, as mentioned above.

#### 2.4.1.1 Important notes

- # You can never update the `number` after processing the XML Request.
- # You can only specify a new starting number in the parent XML Request.
- # **If the subscription is for a finite number of payments, increasing the starting number will reduce the number of payments processed.**

e.g. if you submit `number = 5` and `finalnumber = 12` in a parent AUTH XML Request, there will only be 8 payments in total (because you've effectively skipped payments 1, 2, 3 and 4).

See section 3.1.2.1 for the XML Specification.

### 2.4.2 Interpreting number element when returned in XML Response

When this is returned in an XML Response, it conveys the number associated with the upcoming automated subscription payment. e.g. if you are querying a subscription (see section 4) and the `number` is returned with value 6 in the response, it indicates that the upcoming payment will be labelled as payment number 6 and will be followed with 7 and then 8, etc.

## 2.5 Type

This field indicates the type of subscription, as required by the acquiring bank. The value can either be "INSTALLMENT" or "RECURRING".

**RECURRING** is used when the Customer is performing a recurring payment for a new product/service each time (for example, a magazine subscription). For most merchants, the `type` should be set to "RECURRING".

**INSTALLMENT** is only used in select cases with certain acquirers. It is designed for when a customer is purchasing a single item over several installments (for example, paying £100 a month for an item until it has been paid in full).



Installments are only accepted by certain acquirers. For further information, please contact your acquiring bank.

## 2.6 Begin date (optional in requests)

This field allows you to specify the starting date of payments processed automatically by the subscription engine. Before submitting requests with this field, please ensure you have read and understood the text below on its usage.

### 2.6.1 Subscription without begindate

When processing a subscription, it is important to remember that the parent request (AUTH or ACCOUNTCHECK) will always be processed immediately. Assuming the parent request was processed successfully, recurring payments are automatically processed at your specified interval (see section 7.1 for further information). This is summarised in the following flow:

#### Example of standard subscription flow



Today's date is 5<sup>th</sup> January.  
Your system submits a combined AUTH and SUBSCRIPTION Request, with `unit = MONTH` and `frequency = 1` (schedules a monthly subscription).  
**The first payment is processed immediately on 5<sup>th</sup> January.**

The automated payments begin on a date determined by the unit and frequency fields, based on when the first payment was processed.



### 2.6.2 Specifying the begindate

As an alternative to the flow above, you can submit the additional optional field `begindate` in the SUBSCRIPTION Request. This is used to specify a custom begin date for the automated payments, as described in the following flow.

The begin date is submitted in the format `YYYY-MM-DD` and cannot be a date in the past: e.g. "2017-01-08"

#### Example of subscription flow WITH custom begin date



Today's date is 5<sup>th</sup> January.  
Your system submits a combined AUTH and SUBSCRIPTION Request, with `unit = MONTH`, `frequency = 1` and `begindate = 2017-01-08`.  
**The first payment is processed immediately on 5<sup>th</sup> January.**



**On 8<sup>th</sup> January**, the subscription engine processes the first automated payment (using the date specified in the `begindate` field).

Subsequent automated payments will be processed on dates calculated by using the `begindate` in conjunction with the unit and frequency fields.



If the `begindate` included in the SUBSCRIPTION Request is the same as the date the parent request is processed successfully, a second payment will be processed on the following day.

## 3 Scheduling subscriptions

### 3.1 XML Request

In order to successfully process a subscription with a new AUTH or ACCOUNTCHECK, the XML your system submits to us must employ the following structure.

#### 3.1.1 XML Overview

Within the `requestblock`, there are two `<request>` tags, one for the parent authorisation and the other for the child subscription.

```
<requestblock version="3.67">
  <alias>test_site12345</alias>
  <request type="AUTH">
    ...
    REST OF XML FOR PARENT
    ...
  </request>
  <request type="SUBSCRIPTION">
    ...
    REST OF XML FOR CHILD
    ...
  </request>
</requestblock>
```

#### 3.1.2 The Parent XML Request

The parent XML Request can be either request type AUTH or ACCOUNTCHECK.

Example

```
<requestblock version="3.67">
  <alias>test_site12345</alias>
  <request type="AUTH">
    ...
    REST OF XML FOR PARENT
    ...
  </request>
  <request type="SUBSCRIPTION">
    ...
    REST OF XML FOR CHILD
    ...
  </request>
</requestblock>
```

Both AUTH and ACCOUNTCHECK XML Requests have the same XML structure as specified in the [XML Specification](#). You can add a couple of extra elements to the request. These are described on the next page.




3.1.2.1 Changing the starting subscription number

```

<requestblock version="3.67">
  <alias>site12345</alias>
  <request type="AUTH">
    ...
    REST OF XML FOR PARENT
    ...
    <billing>
      ...
      <subscription type="RECURRING">
        <number>5</number>
      </subscription>
    </billing>
  </request>
  ...
</requestblock>
  
```


The **subscription** tag is only required in the parent request if it contains the **number** element (described below). We describe what can be submitted in **subscription type** in section 2.5.

Tag	Type	Length	Required	Comment
number	n	5	N	<p>This represents the starting position of the transaction within a series of subscription payments.</p> <p>e.g. if number is '5', the subscription will start at number 5 and the next payment will be recorded as payment number 6, then 7, etc.</p> <p>If this field is left blank, the sequence starts at 1.</p>

 **Please note** that you can only set the **number** element in the parent XML Request. This field will be incremented automatically in every subsequent authorisation.

3.1.2.2 About the account type field

As with a standard AUTH or ACCOUNTCHECK request, the **accounttypedescription** value submitted in the parent request must be either "ECOM" or "MOTO" (used to indicate **e-commerce** and **Mail or Telephone Order** respectively). We recommend contacting your acquiring bank for advice on the correct value to submit in the first request in a subscription.

 **In the parent request**, your system must not submit "RECUR" in the **accounttypedescription** field.

**3.1.3 The Child XML Request**

The child XML Request type should be a SUBSCRIPTION. The subscription specific fields can be found in the **billing** tag.

**3.1.3.1 <billing>**

```

<requestblock version="3.67">
  <alias>test_site12345</alias>
  <request type="AUTH">
    ...
    REST OF XML FOR PARENT
    ...
  </request>
  <request type="SUBSCRIPTION">
    <billing>
      ...
      REST OF XML FOR CHILD
      ...
    </billing>
  </request>
</requestblock>
  
```

Tag	Type	Length	Required	Comment
amount	n	13	N	This can have a different value to the parent request, allowing the first payment to be for a different amount to the rest of the subscription.
payment			N	
active	n	1	N	The subscription status. '0' - Inactive: No subscription payments will be processed, unless updated manually. See section 5.1.7. '1' - Active: Schedules subscription payments immediately, bypassing fraud & duplicate checks (if enabled). '2' - Pending (default): See section 7.1 for further information on pending subscriptions.
subscription type=""	an	11	Y	These fields are described in section 2.
final number	n	5	Y	
begindate	an	10	N	
unit	an	5	Y	
frequency	n	11	Y	

All of these elements are always returned in the corresponding XML Response described in section 3.2.

You can include an order reference in **merchant / orderreference** in the child request. This allows you to use one order reference for the parent and another for subscription payments. If not submitted, this is inherited from the parent.

### 3.1.4 Example XML Request

Below is an example of an XML subscription with new authorisation Request:

```
<requestblock version="3.67">
  <alias>site12345</alias>
  <request type="AUTH">
    <operation>
      <sitereference>site12345</sitereference>
      <accounttypedescription>ECOM</accounttypedescription>
    </operation>
    <billing>
      <amount currencycode="GBP">100</amount>
      <town>Bangor</town>
      <country>GB</country>
      <payment type="VISA">
        <expirydate>10/2031</expirydate>
        <pan>4111111111111111</pan>
        <securitycode>123</securitycode>
      </payment>
    </billing>
  </request>
  <request type="SUBSCRIPTION">
    <merchant>
      <orderreference>Example Subscription</orderreference>
    </merchant>
    <billing>
      <amount>200</amount>
      <subscription type="RECURRING">
        <unit>MONTH</unit>
        <frequency>1</frequency>
        <finalnumber>12</finalnumber>
        <begindate>2016-04-01</begindate>
      </subscription>
    </billing>
  </request>
</requestblock>
```

## 3.2 XML Response

Following processing your XML Request, Secure Trading will return an XML Response.

### 3.2.1 XML Overview

Within the **responseblock**, there are two **<response>** tags, one for the parent section of the response (AUTH or ACCOUNTCHECK) and the other for the child (SUBSCRIPTION).

```
<responseblock version="3.67">
  <requestreference>X305548522</requestreference>
  <response type="AUTH">
    ...
    REST OF XML FOR PARENT
    ...
  </response>
  <response type="SUBSCRIPTION">
    ...
    REST OF XML FOR CHILD
    ...
  </response>
</responseblock>
```

### 3.2.2 The Parent XML Response

The structure within the parent response is the same as a normal AUTH or ACCOUNTCHECK XML Response (depending on which request type was submitted in the parent request). This is explained in the [XML Specification](#). (Full XML example can be found in section 3.2.4)

The tags included within the parent response will confirm the details of the AUTH or ACCOUNTCHECK and indicate whether or not the request was successful. Please ensure you have checked the following:

- # Check the **live** element is '0' if you are testing, or '1' when in production using your live account.
- # Ensure the **error / code** is '0' (success). Other codes indicate an error has occurred.
- # For AUTH responses, ensure the **settle / status** is not '2' or '3'.
  - o '2' indicates the payment was suspended following authorisation. You will need to manually review this payment and update the settle status to '1' for it to be settled. Subscription payments are not scheduled until the parent transaction has settled.
  - o '3' indicates the payment has been cancelled, either due to a decline by the customer's bank or an error. The first payment will not be settled, and therefore subsequent payments will not be scheduled. You will need to investigate why the problem occurred and resend the request.



It is especially important that you ensure the **settle / status** and **error / code** elements return the expected values. If there is a problem with the parent, the subscription will not be scheduled.

- # Check the **billing / amount** is for the correct amount. This is the amount associated with the first payment.

For further guidance on how to check your XML Responses, please refer to the Best Practices found in the [XML Specification](#).

### 3.2.3 The Child XML Response

The tags included within the SUBSCRIPTION response will confirm the details of the subscription and indicate whether or not the request was successful. Ensure you perform the following steps on XML Responses returned:

- # Ensure the **error / code** is '0' (success). Other codes indicate an error has occurred.



Please ensure you check the **error / code** and **error / message** in both the parent **AND** child responses, as each request is processed independently.

- # Record the **transactionreference** returned as this is required to perform future updates on your subscription.



**Please note** that a **transactionreference** will be returned for both the parent request **and** the child request. Any queries or updates relating to a subscription must pass through the **transactionreference** of the SUBSCRIPTION child (not the parent AUTH).

- # Check **billing / payment / active** has the expected value:
  - o '0' - Subscription is inactive; you can manually activate it later (section 5.1.7).
  - o '1' - Subscription is active.
  - o '2' - Subscription is pending; see section 7.1 for further information.
- # Check the **billing / amount** is for the correct amount. This can be different to the amount specified in the parent request if you have specified this in the child request.
- # Check the **billing / subscription** elements containing the subscription information are correct. You can find an explanation of all of these fields in section 2.

## 3.2.4 Example XML Response

The following is an example of an XML Response from a SUBSCRIPTION that inherits from a parent request submitted in the same request block:

```
<responseblock version="3.67">
  <requestreference>X305548522</requestreference>
  <response type="AUTH">
    <merchant>
      <merchantname>My Test Site</merchantname>
      <orderreference>Example AUTH</orderreference>
      <tid>27881234</tid>
      <merchantnumber>540436134211002</merchantnumber>
      <merchantcountryiso2a>GB</merchantcountryiso2a>
      <operatorname>site12345</operatorname>
    </merchant>
    <transactionreference>12-3-2</transactionreference>
    <security>
      <postcode>0</postcode>
      <securitycode>2</securitycode>
      <address>0</address>
    </security>
    <billing>
      <amount currencycode="GBP">100</amount>
      <payment type="VISA">
        <issuer>Test Issuer</issuer>
        <pan>411111#####1111</pan>
        <issuercountry>ZZ</issuercountry>
      </payment>
    </billing>
    <authcode>2</authcode>
    <timestamp>2016-02-27 14:05:30</timestamp>
    <settlement>
      <settleduedate>2016-02-27</settleduedate>
      <settlestatus>0</settlestatus>
    </settlement>
    <live>1</live>
    <error>
      <message>Ok</message>
      <code>0</code>
    </error>
    <acquirerresponsecode>00</acquirerresponsecode>
    <operation>
      <accounttypedescription>ECOM</accounttypedescription>
    </operation>
  </response>
  <response type="SUBSCRIPTION">
    <merchant>
      <orderreference>Example Subscription</orderreference>
    </merchant>
    <transactionreference>12-64-2</transactionreference>
    <billing>
      <amount currencycode="GBP">100</amount>
      <payment type="VISA">
        <active>2</active>
        <pan>411111#####1111</pan>
      </payment>
      <subscription type="RECURRING">
        <finalnumber>12</finalnumber>
        <begindate>2016-04-01</begindate>
      </subscription>
    </billing>
  </response>
</responseblock>
```

```
<number>2</number>
  <frequency>1</frequency>
  <unit>MONTH</unit>
</subscription>
</billing>
<timestamp>2016-03-27 00:00:00</timestamp>
<live>1</live>
<error>
  <message>Ok</message>
  <code>0</code>
</error>
<operation>
  <parenttransactionreference>12-3-2</parenttransactionreference>
  <accounttypedescription>RECUR</accounttypedescription>
</operation>
</response>
</responseblock>
```

## 4 Querying subscriptions

In order to view details of a subscription that has already been scheduled, you can submit a transaction query request, passing through the **transactionreference** of the processed subscription. An XML Response will be returned, with fields that hold information about future scheduled payments (these fields are described in section 2).

### 4.1 XML Request

#### 4.1.1 XML Overview

The structure of the XML Request used in this case is the same as a regular transaction query request (explained in the [Transaction Query document](#)).



**Please note** that you must pass through the transaction reference of the original SUBSCRIPTION, and not the transaction references of any authorisations processed automatically by the subscription engine. If you don't have this reference, see section 9.1.

#### 4.1.2 Example XML Request

Please find below an example of a transaction query request to be submitted to Secure Trading's systems.

```
<?xml version="1.0" encoding="utf-8"?>
<requestblock version="3.67">
  <alias>site12345</alias>
  <request type="TRANSACTIONQUERY">
    <filter>
      <sitereference>site12345</sitereference>
      <transactionreference>50-2-2</transactionreference>
    </filter>
  </request>
</requestblock>
```

### 4.2 XML Response

#### 4.2.1 XML Overview

Once you have successfully submitted a transaction query request, if no errors were found, you will be returned an XML Response. The structure of the XML returned in the response is similar to that of a normal transaction query response, but with additional subscription fields.

#### 4.2.2 <subscription>

Tag	Type	Length	Required	Comment
subscription type=""	an	11	Y	These fields are described in section 2.
number	n	5	Y	
finalnumber	n	5	Y	
begindate	an	10	Y	
unit	an	5	Y	
frequency	n	11	Y	



4.2.3 Example XML Response

Here is an example of a typical Transaction Query XML Response. The fields relevant to subscriptions are highlighted in **bold**. This response represents an active subscription for £10 a month, for 12 months and tells us that the upcoming subscription payment is the 4th payment in the series.



**Please note** that in cases of **number** having a higher value than the **finalnumber**, this represents a completed subscription, and no further authorisations will be processed, unless the **finalnumber** is raised in a transaction update request (see section 5.1.4.1).

```

<?xml version="1.0" encoding="utf-8"?>
<responseblock version="3.67">
  <requestreference>X78030262</requestreference>
  <response type="TRANSACTIONQUERY">
    <found>1</found>
    <timestamp>2016-03-11 10:05:52</timestamp>
    <record type="SUBSCRIPTION">
      <merchant>
        <orderreference>Test Subscription</orderreference>
        <operatorname>site12345</operatorname>
      </merchant>
      <transactionreference>50-2-2</transactionreference>
      <billing>
        <payment type="VISA">
          <issuer>Test Issuer</issuer>
          <issuercountry>ZZ</issuercountry>
          <expirydate>10/2028</expirydate>
          <active>1</active>
          <pan>4111#####1111</pan>
        </payment>
        <subscription type="RECURRING">
          <finalnumber>12</finalnumber>
          <begindate>2016-02-05</begindate>
          <number>4</number>
          <frequency>1</frequency>
          <unit>MONTH</unit>
        </subscription>
        <name>
          <last>Lastname</last>
        </name>
        <amount currencycode="GBP">1000</amount>
      </billing>
      <timestamp>2016-03-05 00:00:00</timestamp>
      <live>1</live>
      <error>
        <message>Ok</message>
        <code>0</code>
      </error>
      <operation>
        <parenttransactionreference>17-9-2</parenttransactionreference>
        <interface>CERT-XML-XML</interface>
        <sitereference>site12345</sitereference>
        <accounttypedescription>RECUR</accounttypedescription>
      </operation>
    </customer>
  </response>
</responseblock>

```

```
    <ip>1.2.3.4</ip>
  </customer>
  <settlement>
    <updatereason>subscription</updatereason>
  </settlement>
</record>
<error>
  <message>Ok</message>
  <code>0</code>
</error>
</response>
</responseblock>
```

## 5 Updating subscriptions

### 5.1 XML Request

In order to update or cancel a subscription, you can submit a transaction update request, passing through the `transactionreference` of the processed subscription.

#### 5.1.1 XML Overview

Transaction update requests consist of two tags:

- # `filter` - contains elements used to identify which subscription is being updated.
- # `updates` - contains the fields that are to be updated and their new values.

#### 5.1.2 <filter>

Tag	Type	Length	Required	Comment
filter			Y	
site reference	an	40	Y	The unique reference for the site associated with the subscription.
transaction reference	an	20	Y	The unique reference for the subscription you would like to update.



**Please note** that you must pass through the transaction reference of the original SUBSCRIPTION, and not the transaction references of any authorisations processed automatically by the subscription engine. If you don't have this reference, see section 9.1.

#### 5.1.3 <updates>

The updatable fields relevant to subscriptions are outlined in the following table:

Tag	Type	Length	Required	Comment
billing			N	
payment			N	
active	n	1	N	Update to '0' to manually disable a subscription. Update to '1' to manually enable a subscription.
expirydate	an	7	N	Use this element to update the card expiry date (in format 'MM/YYYY')
subscription			N	
finalnumber	n	5	N	See section 5.1.4.
unit	an	5	N	
frequency	n	11	N	



**Please note** that it is not possible to update the currency, payment type or certain payment details (e.g. card number) with different values in the subscription to the parent transaction.

It is, however, possible to update the expiry date of a card.

#### 5.1.4 Updating subscription fields

Within the **updates** tag, there is a **subscription** tag, which contains all of the fields that can only be updated on subscriptions.



**Please note** that the **begindate** and **number** elements can never be updated.

##### 5.1.4.1 <finalnumber>

**finalnumber** represents the position of the last payment in a series of subscription payments. Once this number has been reached, no further payments will be processed. Updating this field will change the amount of subscription payments processed in total.

*Example: Changing how many subscription payments will be processed*

Updating the **finalnumber** field from 6 to 10 will result in 4 additional payments being scheduled.

*Example: Changing a subscription so that it will continue indefinitely*

Updating the **finalnumber** field from 6 to 0 will result in the subscription continuing indefinitely until it is deactivated.

##### 5.1.4.2 <unit>

**unit** represents the unit of time used to schedule payments (“DAY” or “MONTH”). It is used in conjunction with the **frequency** field to determine the gap between payments. E.g. updating **unit** from “DAY” to “MONTH”, when **frequency** is 2, changes subscriptions from being processed every 2 days to every 2 months.

##### 5.1.4.3 <frequency>

**frequency** is the number of units that should occur before the next authorisation is processed. It is used in conjunction with the **unit** field to determine the gap between payments. E.g. updating **frequency** from 2 to 5, when **unit** is set to “DAY”, changes subscriptions from being processed every 2 days to every 5 days.

### 5.1.5 Update Subscription Request Example

Below is an example of an XML subscription update request.

This XML request would update the **amount** of the subscription to 2000 (£20), the frequency of payments to once a month, and the **finalnumber** to be 12 (which will change the total number of subscriptions processed; section 5.1.4).

```
<requestblock version="3.67">
  <alias>site12345</alias>
  <request type="TRANSACTIONUPDATE">
    <filter>
      <sitereference>site12345</sitereference>
      <transactionreference>17-9-2</transactionreference>
    </filter>
    <updates>
      <billing>
        <amount>2000</amount>
        <subscription>
          <finalnumber>12</finalnumber>
          <frequency>1</frequency>
          <unit>MONTH</unit>
        </subscription>
      </billing>
    </updates>
  </request>
</requestblock>
```

**Caution:** It is possible to increase the **finalnumber** field, to extend a subscription. However, if this is done after a subscription has completed, all payments that would have been taken if the subscription had been allowed to continue, are immediately processed.

e.g. If a £10/month subscription has completed, and five months after completion, the **finalnumber** is raised by five, five £10 payments will be processed in the next settlement run (usually within the next 24 hours).



To extend a subscription with the same billing details without processing the missing payments, you must deactivate the existing subscription (see section 5.1.6) and submit a new request (see section 3), including the **transactionreference** of the original parent (in order to use the same billing details). There are two approaches we support in this scenario:

1. Submit a new combined **AUTH SUBSCRIPTION** request on the day of the month you would like automated payments to be processed going forward.
2. Submit a new combined **ACCOUNTCHECK SUBSCRIPTION** request, including the **begindate** that you would like automated payments to be processed going forward.

### 5.1.6 Deactivate Subscription Request Example

In order to deactivate an active subscription, the **active** element in the TRANSACTIONUPDATE XML request must be set to "0". Below is an example of an XML Transaction Update Request, which would deactivate an active subscription:

```
<requestblock version="3.67">
  <alias>site12345</alias>
  <request type="TRANSACTIONUPDATE">
    <filter>
      <sitereference>site12345</sitereference>
      <transactionreference>12-64-1</transactionreference>
    </filter>
    <updates>
      <billing>
        <payment>
          <active>0</active>
        </payment>
      </billing>
    </updates>
  </request>
</requestblock>
```

### 5.1.7 Start Subscription Request Example

In order to start a pending or inactive subscription, the `active` tag in the TRANSACTIONUPDATE XML request must be set to '1'.



**Please note** that updating a pending subscription to be active will ignore the results of any fraud or duplicate checks on the parent (these are disabled by default). See section 7.1 for further information.

#### Example

```
<requestblock version="3.67">
  <alias>site12345</alias>
  <request type="TRANSACTIONUPDATE">
    <filter>
      <sitereference>site12345</sitereference>
      <transactionreference>12-64-1</transactionreference>
    </filter>
    <updates>
      <billing>
        <payment>
          <active>1</active>
        </payment>
      </billing>
    </updates>
  </request>
</requestblock>
```

**Caution:** Upon activating a pre-existing subscription, all payments that would have been taken during the period the subscription was pending or inactive are immediately processed.

e.g. If a £10/month subscription is pending or inactive for four months and then activated, four £10 payments will be processed in the next settlement run (usually within the next 24 hours).



To continue a subscription with the same billing details without immediately processing payments, you must deactivate the existing subscription (see section 5.1.6) and submit a new request (see section 3), including the **transactionreference** of the original parent (in order to use the same billing details). There are two approaches we support in this scenario:

1. Submit a new combined **AUTH SUBSCRIPTION** request on the day of the month you would like automated payments to be processed going forward.
2. Submit a new combined **ACCOUNTCHECK SUBSCRIPTION** request, including the **begindate** that you would like automated payments to be processed going forward.



**Please note** that a subscription will never take payments if the **number** is greater than the **finalnumber**. In order to reactivate a subscription in such a case, you must also specify a higher **finalnumber** in the transaction update request.

### 5.1.8 Update Monthly Subscription Date

To change the date on which a monthly subscription processes payments (e.g. from the 1st of each month to the 15th), you must deactivate the existing subscription (see section 5.1.6) and submit a new request (see section 3), passing through the updated subscription details, and including the **transactionreference** of the original parent (in order to use the same billing details). There are two approaches we support in this scenario:

1. Submit a new combined **AUTH SUBSCRIPTION** request on the day of the month you would like automated payments to be processed going forward.
2. Submit a new combined **ACCOUNTCHECK SUBSCRIPTION** request, including the **begindate** that you would like automated payments to be processed going forward.

## 5.2 XML Response

After you have successfully submitted a transaction update request, you will be returned an XML Response.

### Example

Please find below an example of a transaction update response for a subscription transaction:

```
<responseblock version="3.67">
  <requestreference>X675136983</requestreference>
  <response type="TRANSACTIONUPDATE">
    <timestamp>2010-03-11 16:38:47</timestamp>
    <error>
      <message>Ok</message>
      <code>0</code>
    </error>
  </response>
</responseblock>
```



## 6 Subscription Notifications

Once your site has been set up to process Subscriptions, it is recommended that you opt-in to receiving notifications to keep you informed of payments processed on your account by the subscription engine.

### 6.1 Subscription Reports

If you would like to receive email reports that show subscription payments processed on your site, please contact our support team (see section 8.1).

#### 6.1.1 Structure of email

If you have the email report enabled on your account, you will receive an email from *no-reply@securetrading.com*, for each day that Subscription payments are processed. You will receive a maximum of one email per day. They are presented in the format of a table (see section 9.3), with the following fields:

Field	Comment
Subscription reference	A unique Secure Trading reference for the Subscription transaction.
Transaction reference	A unique Secure Trading reference for the transaction.
Account	This will match the account type description supplied within the request. This should be RECUR.
Request	The type of request processed. <b>Note:</b> AUTH is an Authorisation Request.
Currency	The currency that the transaction was processed in.
Settle status	The current status of the transaction at the end of request.
Auth code	The authorisation code provided by the Issuing Bank. This is generated by the Issuing Bank, and will differ, depending on which bank you use.
Error code	Whether or not the transaction was successful. 0 represents a successful transaction.
Base amount	The amount of the transaction in base units, with no commas or decimal points. For example, £10 would be shown as 1000.
Sitereference	The Site reference relates to your individual account which you received on setup.
Subscription frequency	This is the time between Authorisations. It represents the gap between each payment in a Subscription.
Subscription number	This represents the position of the transaction within a series of Subscription payments. E.g. 5/10 would represent the 5 <sup>th</sup> transaction out of 10.
Subscription type	This is the type of Subscription: 'RECURRING' or 'INSTALLMENT' (see section 2.5 for further details).
(Optional) Acquirer advice code	A numeric value returned, indicating if further payments can be processed. As this field is only supported by certain acquiring banks, you will need to ask our Support team to add this to your report. See section 9.1 for a full mapping.



The email will be sent from *no-reply@securetrading.com*, with the subject "Subscription Report for [date in YYYY-MM-DD]". Please add a rule or filter to your email client to prevent the emails from being classified as junk mail / spam.

## 6.2 Error Reports

If you would like to receive an email if a subscription transaction processed on your site returns an error, please contact our support team (see section 8.1).

### 6.2.1 Structure of email

If you have the email report enabled on your account, you will receive an email from *no-reply@securetrading.com*, whenever errors are encountered while processing Subscription payments. You will receive a maximum of one email per day. More than one Subscription can appear in one email, and are separated by spaces.

Here is an example of a Subscription error email report (in this case, the customer's card has been declined):

---

Problem with processing transaction 3-64-56 – 70000 Decline subscriptionnumber:2

---



The email will be sent from [no-reply@securetrading.com](mailto:no-reply@securetrading.com), with the subject "STPP Subscription runsubscriptions problems". Please add a rule or filter to your email client to prevent the emails from being classified as junk mail / spam.

## 6.3 URL Notifications

It is possible to configure URL Notifications to be submitted to your system, following each payment processed on your site by the subscription engine. (One notification per transaction)

To configure this, sign in to [MyST](#) and navigate to the "Rule Manager".

If you are new to the Rule Manager, we recommend reading this information in conjunction with the [Rule Manager document](#).

### 1. Choose the site

Select the site through which you are processing subscriptions and the action type "URL Notification", then click "Change".

### 2. Create the condition

Create a condition for completed requests, with the following properties:

- # Account type is RECUR.
- # Request type is AUTH.
- # Select all payment types, currencies and settle statuses.
- # Error code is 0.

### 3. Create the action

If you haven't already, create a new action with the URL of your server and include the fields you would like to be included in the notification.

### 4. Create the rule

Using the Rule Manager, assign the condition to the action, and click "Save". Ensure the checkbox in the "Active" column is ticked, indicating the new rule is active. Please test the rule is working on your test site reference before applying it to your live site reference.

## 7 Additional Notes

### 7.1 Pending subscriptions

#### 7.1.1 When parent is AUTH

A subscription will not process payments until the parent AUTH has been settled into your bank account. Until then, the subscription is listed with an **active** status of 2 (pending).

#### 7.1.2 When parent is ACCOUNTCHECK

If the ACCOUNTCHECK was successful (error code of "0"), the subscription is listed with an **active** status of 2 (pending) for a short time (less than 24 hours) before being updated to active (status of "1").

#### 7.1.3 Manual override

It is possible to force the subscription to be scheduled immediately (regardless of the state of the parent) by using a transaction update request, but this would bypass the duplicate and fraud checks on the parent (if enabled; see section 7.3).

See section 5.1.7 for information on how to update a "Pending" Subscription to be "Active".

### 7.2 Declined Authorisation

It is not possible to schedule recurring payments in the subscription engine, if the parent was declined by the acquirer.

If the AUTH / ACCOUNTCHECK section of the request was declined, you will not receive a response for the SUBSCRIPTION, as no subscription will have been scheduled.

### 7.3 Fraud and duplicate checks

If enabled, fraud and duplicate checks are only performed on the parent and not on any subsequent subscription payments. These checks will defer settlement of the parent (and therefore suspend the subscription) if they fail.

Fraud and duplicate checks are not enabled by default on new Secure Trading accounts. Please contact Secure Trading Support for further information (see section 8.1).

### 7.4 End of month subscription payments

Subscription payments can be processed monthly from the date of the initial payment. If the date of the initial payment is after the 28<sup>th</sup> of the month, all following Subscription payments are scheduled for the 28<sup>th</sup> of each month. This is to prevent issues for Subscription payments on months that have fewer days than others.

### 7.5 About MyST

MyST is a password protected online management area which allows all Secure Trading merchants to monitor their transactions and perform other management on their account(s). You can use the MyST system to view and update subscriptions. This is explained in detail in the [MyST User Guide](#).

**7.6 Expiry dates**

Please ensure that the expiry dates on your active subscriptions are valid and up-to-date as some issuers may decline cards with outdated card expiry dates. These can be updated by using [MyST](#) or by performing a Transaction Update XML Request (see section 5).

**7.7 Accepted payment types**

Only VISA, MASTERCARD and AMEX branded cards can be used for processing subscription payments (excluding Maestro).

**7.8 3-D Secure**

3-D Secure cannot be utilized by the subscription engine due to the customer not being present to enter their credentials at the time the payments are processed.

For this reason, subscription payments that have an initial authorisation that has been processed with 3D Secure may not be covered by the liability shift. Please contact your acquiring bank for clarification.

**7.9 Direct Currency Conversion (DCC)**

The Secure Trading subscription engine does not currently support Dynamic Currency Conversion (DCC).

## 8 Further information and Support

This section provides useful information with regards to documentation and support for your Secure Trading solution.

### 8.1 Secure Trading Support

If you have any questions regarding integration or maintenance of the system, please contact our Support team using one of the following methods:

Method	Details
Telephone	+44 (0) 1248 672 050
Fax	+44 (0) 1248 672 099
Email	<a href="mailto:support@securetrading.com">support@securetrading.com</a>
Website	<a href="http://www.securetrading.com/support/support.html">http://www.securetrading.com/support/support.html</a>

### 8.2 Secure Trading Sales

If you do not have an account with Secure Trading, please contact our Sales team and they will inform you of the benefits of a Secure Trading account.

Method	Details
Telephone	0800 028 9151
Telephone (Int'l)	+44 (0) 1248 672 070
Fax	+44 (0) 1248 672 079
Email	<a href="mailto:sales@securetrading.com">sales@securetrading.com</a>
Website	<a href="http://www.securetrading.com">http://www.securetrading.com</a>

### 8.3 Useful Documents

The following documents should be read in conjunction with this document:

- # [STAPI Setup Guide](#) – This document outlines how to install the STAPI java client for processing XML Requests and Responses through Secure Trading.
- # [STPP Web Services User Guide](#) – This document describes how to process XML Requests and Responses through Secure Trading’s Web Services solution.
- # [STPP XML Specification](#) – This document details how to perform AUTH, REFUND and ACCOUNTCHECK XML Requests through Secure Trading.
- # [STPP Transaction Query](#) – This document details how to perform TRANSACTIONQUERY XML Requests through Secure Trading.
- # [STPP MyST User Guide](#) – This document outlines how to use the MyST interface.

Any other document regarding the STPP system can be found on Secure Trading’s website (<http://www.securetrading.com>). Alternatively, please contact our support team as outlined above.

### 8.4 Frequently Asked Questions

Please visit the FAQ section on our website (<http://www.securetrading.com/support/faq>).

## 9 Appendix

### 9.1 Querying a SUBSCRIPTION without a reference

If you do not know the specific SUBSCRIPTION reference, you can still query a Subscription if you are in possession of the initial parent (AUTH or ACCOUNTCHECK). The response will contain the missing SUBSCRIPTION reference, allowing you to perform further queries and updates as required.

You will need to submit a TRANSACTIONQUERY XML Request similar to the following:

```
<?xml version="1.0" encoding="utf-8"?>
<requestblock version="3.67">
<alias>test_site12345</alias>
<request type="TRANSACTIONQUERY">
  <filter>
    <sitereference>test_site12345</sitereference>
    <parenttransactionreference>12-3-1</parenttransactionreference>
    <requesttypedescription>SUBSCRIPTION</requesttypedescription>
  </filter>
</request>
</requestblock>
```

Please ensure your request includes the following elements:

- **Site reference** - Specify the site reference the subscription is being processed through.
- **Parent transaction reference** - Specify the transaction reference of the initial parent.
- **Request type description** - This must be "SUBSCRIPTION".

This topic is continued over the page.

If performed correctly, this will return a TRANSACTIONQUERY XML Response, containing a record of the SUBSCRIPTION, including the reference required to perform future updates. Here is an example (fields of specific interest to subscriptions have been highlighted in **bold**):

```
<?xml version='1.0' encoding='utf-8'?>
<responseblock version="3.67">
  <requestreference>X176225005</requestreference>
  <response type="TRANSACTIONQUERY">
    <record type="SUBSCRIPTION">
      <operation>
        <parenttransactionreference>12-3-1</parenttransactionreference>
        <sitereference> test_site12345</sitereference>
        <interface>CERT-XML-XML</interface>
        <accounttypedescription>RECUR</accounttypedescription>
      </operation>
      <settlement>
        <updatereason>subscription</updatereason>
      </settlement>
      <billing>
        <payment type="VISA">
          <expirydate>10/2031</expirydate>
          <active>1</active>
          <pan>411111#####1111</pan>
        </payment>
        <subscription type="RECURRING">
          <finalnumber>12</finalnumber>
          <begindate>2016-04-01</begindate>
          <number>6</number>
          <frequency>1</frequency>
          <unit>MONTH</unit>
        </subscription>
        <amount currencycode="GBP">100</amount>
      </billing>
      <live>0</live>
      <merchant>
        <orderreference>Example Subscription</orderreference>
        <operatorname>test_site12345</operatorname>
      </merchant>
      <transactionreference>12-64-2</transactionreference>
      <timestamp>2016-07-30 00:00:00</timestamp>
      <error>
        <message>Ok</message>
        <code>0</code>
      </error>
    </record>
    <found>1</found>
    <error>
      <message>Ok</message>
      <code>0</code>
    </error>
    <timestamp>2016-07-11 09:43:32</timestamp>
  </response>
  <secrand>147L</secrand>
</responseblock>
```

For further information on interpreting elements returned in the **subscription** tag, please refer to section 2.

9.2 Acquirer advice code mapping

Code	Description	Action
0	N/A	No action required
1	“New account information available” The bank has reason to believe that the customer’s card details are out-of-date (e.g. the expiry date has passed).	Please check with the customer that their card details are still correct. This field is advisory and we will allow you to re-process even if the card details are not updated, however the acquiring bank may not accept the payment. You may find it useful to contact your bank in such cases for clarification.
2	“Cannot approve at this time”	Try again later. If you are continuing to have difficulties, please contact your acquiring bank
4	“Do not try again”	Do not process further recurring transactions
8	“Payment blocked by card scheme”	

For information on how to test for different acquirer advice code values, please refer to the [Testing document](#).

9.3 Example report email for subscriptions

The following is the contents of an example email report for Subscriptions. For more information on report emails, see section 6.

[ This is an automated email ]  
 The following subscriptions were processed on 2016-03-07  
 All of these transactions were authorised:

Subscription reference	Transaction reference	Account	Request	Currency	Settle status	Auth Code	Error code	Base amount	SiteReference	Subscription frequency	Subscription number	Subscription type
3-64-56	6-9-482453	RECUR	AUTH	GBP	1 - Manual settlement	TEST	0	1200	test_site1	1 DAY	2/2	RECURRING
3-64-58	4-9-744247	RECUR	AUTH	GBP	1 - Manual settlement	TEST	0	1000	test_site1	1 MONTH	2/9	RECURRING
3-64-59	2-9-482237	RECUR	AUTH	GBP	1 - Manual settlement	TEST	0	1200	test_site1	1 DAY	2/2	RECURRING

This e-mail was sent from a notification-only address that cannot accept incoming e-mail. Please do not reply to this message