Antenor Shopmaster Protocol – M2M interface

1 About the Shopmaster

The Antenor Shopmaster is a payment device designed to perform cash transactions in self service resulting in an efficient, fast and accurate way to give change to your customers. Nowadays, the Shopmaster is used in butcher shops and bakeries in Belgium. Shop owners are attracted to the Shopmaster because of the several advantages it offers to them. For instance, food handling (clean) is completely separated from cash handling (dirty), providing an important improvement in hygiene. Furthermore, cash payments are handled a lot faster, and more accurate. Invalid notes and coins are automatically rejected, and the change is given fast and correct. At the end of the day, the revenue is easily displayed using the built-in reporting. To further extend the system, payments using a Banksys debit/credit card terminal are also made possible, using exactly the same interfaces for communication and reporting.

2 About the interface

Firstly the only way to start a transaction on the Shopmaster was by entering the required amount on a keyboard. Later on, this was at large customer request replaced by scanning a barcode containing the amount, reducing manual labour and mistakes. Now we arrive at the third generation, where the check-out register automatically instructs the Shopmaster to start payments, resulting in less paper use, faster workflows and a reduction in manual effort, providing even more profit and increased efficiency.

In order to make this interface as general as possible, Antenor relied on its years-long experience in payment protocols to define its Shopmaster machine-to-machine (M2M) protocol, as documented in this reference manual. This protocol is named ASMP, acronym for the Antenor Shopmaster Protocol. The most recently released version of the XML Schema, which you can use for validating input, can be found at http://download.antenor.eu/APS/ShopmasterXML/ASMP.xsd as indicated in the examples of root elements of message types.

Changelog

Date	Author	Description
2010/08/23	kristiaan.bonjean@antenor.be	First definition draft 1.0
2011/10/11	kristiaan.bonjean@antenor.be	Protocol flow example, integer cancel restriction instead of positiveInteger so Cancel type 0 is allowed. Definition v1.1.

3 Protocol flow

The protocol was designed using the master-slave paradigm, in which the Antenor Shopmaster is the slave and the master is for instance a check-out register. Therefore, the master sends requests to the slave, getting responses in return. Because the slave can never initiate communication to the master, the only way to get the status of the payment device is by polling. On every request the slave receives, an answer is sent as soon as possible. Every message type was designed to have both a request (master→slave) and a response (slave→master). When an incorrect parameter is specified in the request, the response will contain the error-element instead of the actual result of the query. On the other hand, when the slave receives bogus input, unknown commands or errors in the XML-formatting, the root Error-element will be sent.

Communication happens by sending an HTTP POST command to http://address-of-shopmaster/asmp.xml . This POST command sends an appropriate “request” element. The entire content should be validated using the ASMP.xsd definition.

3.1 Example

Example of debit request sent from master to slave including HTTP headers:

```
POST /asmp.xml HTTP/1.1
Host: 127.0.0.1
User-Agent: APS2.7.4Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-gb,en;q=0.5
Accept -encoding: gzip , deflate
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
<?xml version="1.0"?>

<DebitRequest                xmlns="http://download.antenor.eu/APS/ShopmasterXML"
                               xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
                               xsi:schemaLocation="   http://download.antenor.eu/APS/ShopmasterXML
                                                 http://download.antenor.eu/APS/ShopmasterXML/ASMP.xsd"
                               asmpversion="1.0">
                <Transaction>
                               <AmountRequested units="eurocent">1920</AmountRequested>
                               <TransactionIssuer>kassa 3</TransactionIssuer>
                               <TransactionReference>102</TransactionReference>
                </Transaction>
</DebitRequest>
```

Response:

```
<?xml version="1.0"?>
<DebitResponse xmlns="http://download.antenor.eu/APS/ShopmasterXML"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://download.antenor.eu/APS/ShopmasterXML
http://download.antenor.eu/APS/ShopmasterXML/ASMP.xsd"
asmpversion="1.0"><State><StateCode>3</StateCode><CurrentTransaction><TransactionReference>102
</TransactionReference><TransactionIssuer>kassa 3</TransactionIssuer><TransactionStart
format="localtime">2011-09-12T13:42:06</TransactionStart><AmountRequested units="eurocent">
1920</AmountRequested><AmountReceived units="eurocent">0</AmountReceived>
</CurrentTransaction></State></DebitResponse>
```

In the response, the transaction of € 19,20 is shown as “Busy” (in progress) and the currently received amount is 0 eurocents. On following StateRequests the AmountReceived should grow. After the transaction finishes, the StateCode will be 0 (idle), the CurrentTransaction field will be empty and the LastTransaction should show AmountRequested and AmountReceived as 1920.

4 Message types

4.1 Root elements

The root elements specified underneath are the messages used for communication using ASMP. However, because some elements contain other elements, which are used in multiple messages, re-useable elements are specified in chapter 4.2 Re-usable elements.

4.1.1 DebitRequest

```
<xs:element name="DebitRequest">
  <xs:complexType>
    <xs:complexContent>
      <xs:extension base="ASMPType">
        <xs:all>
          <xs:element name="Transaction" type="TransactionType" />
        </xs:all>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
</xs:element>
```

Listing 1: Schema for the DebitRequest-element

The DebitRequest is used to initiate payment. The Transaction-element should only contain:

- AmountRequested: the required amount to pay in eurocent
- TransactionIssuer (optionally): the issuer of the transaction, for instance the name of the person logged in at the register, or the name of the register, or a combination
- TransactionReference (optionally): a transaction reference which is stored in the transaction database, so transactions can be linked to the equivalent ticket of the register

Example:

```
<?xml version="1.0"?>

<DebitRequest   xmlns="http://download.antenor.eu/APS/ShopmasterXML"
                xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
                xsi:schemaLocation="     http://download.antenor.eu/APS/ShopmasterXML
                                         http://download.antenor.eu/APS/ShopmasterXML/ASMP.xsd"
                asmpversion="1.0">
        <Transaction>
                <AmountRequested units="eurocent">1920</AmountRequested>
                <TransactionIssuer>kassa 3</TransactionIssuer>
                <TransactionReference>102</TransactionReference>
        </Transaction>
</DebitRequest>
```

Listing 2: DebitRequest example

4.1.2 DebitResponse

```
<xs:element name="DebitResponse">
  <xs:complexType>
    <xs:complexContent>
      <xs:extension base="ASMPType">
        <xs:choice>
          <xs:element name="Error" type="ErrorType" />
          <xs:element name="State" type="StateType" />
        </xs:choice>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
</xs:element>
```

Listing 3: Schema for the DebitResponse-element

DebitResponse is sent on receipt of a DebitRequest. When the Shopmaster is available and all required DebitRequest-elements are valid, the DebitResponse will contain the State, to show the device is busy handling the transaction. Else, the DebitResponse will contain the Error-element explaining why the transaction couldn’t be initiated.

Example:

```
<?xml version="1.0"?>

<DebitResponse xmlns="http://download.antenor.eu/APS/ShopmasterXML"
               xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
               xsi:schemaLocation="     http://download.antenor.eu/APS/ShopmasterXML
                                        http://download.antenor.eu/APS/ShopmasterXML/ASMP.xsd"
               asmpversion="1.0">
       <State>
               <StateCode>3</StateCode>
               <CurrentTransaction>
                       <TransactionIssuer>kassa 3</TransactionIssuer>
                       <TransactionReference>102</TransactionReference>
                       <TransactionStart format="localtime">2010-08-15T11:28:53</TransactionStart>
                       <AmountRequested units="eurocent">1920</AmountRequested>
                       <AmountReceived units="eurocent">500</AmountReceived>
               </CurrentTransaction>
               <LastTransaction>
                       <TransactionIssuer>kassa 1</TransactionIssuer>
                       <TransactionReference>101</TransactionReference>
                       <TransactionStart format="localtime">2010-08-15T11:21:45</TransactionStart>
                       <AmountRequested units="eurocent">14320</AmountRequested>
                       <AmountReceived units="eurocent">14320</AmountReceived>
                       <PayType><PayTypeCash /></PayType>
                       <IncidentCode>3</IncidentCode>
               </LastTransaction>
       </State>
</DebitResponse>
```

Listing 4: DebitResponse example

4.1.3 StateRequest

```
<xs:element name="StateRequest">
  <xs:complexType>
    <xs:complexContent>
      <xs:extension base="ASMPType">
        <xs:sequence>
          <xs:element name="StateType">
            <xs:simpleType>
              <xs:restriction base="xs:positiveInteger">
                <xs:pattern value="101" />
              </xs:restriction>
            </xs:simpleType>
          </xs:element>
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
</xs:element>
```

Listing 5: Schema for the StateRequest-element

StateRequest is to be used when the master wants to poll the slave for its state. Currently, only ShopmasterState (StateType=101) is supported. In the future, this might be expanded to include widely used VABP status commands, for instance to poll hoppers or to return sales overviews.

Example:

```
<?xml version="1.0"?>

<StateRequest   xmlns="http://download.antenor.eu/APS/ShopmasterXML"
                xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
                xsi:schemaLocation="     http://download.antenor.eu/APS/ShopmasterXML
                                         http://download.antenor.eu/APS/ShopmasterXML/ASMP.xsd"
                asmpversion="1.0">
        <StateType>101</StateType>
</StateRequest>
```

Listing 6: StateRequest example

4.1.4 StateResponse

```
<xs:element name="StateResponse">
  <xs:complexType>
    <xs:complexContent>
      <xs:extension base="ASMPType">
        <xs:choice>
          <xs:element name="Error" type="ErrorType" />
          <xs:element name="State" type="StateType" />
        </xs:choice>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
</xs:element>
```

Listing 7: Schema for the StateResponse-element

StateResponse is replied by the slave after receiving a StateRequest. Normally it will contain the State element, unless the StateType of the StateRequest was either not specified, either invalid, in which case the Error-element will be replied in the StateResponse.

Example:

```
<?xml version="1.0"?>

<StateResponse xmlns="http://download.antenor.eu/APS/ShopmasterXML"
                xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
                xsi:schemaLocation="     http://download.antenor.eu/APS/ShopmasterXML
                                         http://download.antenor.eu/APS/ShopmasterXML/ASMP.xsd"
                asmpversion="1.0">
        <State>
                <StateCode>2</StateCode>
                <LastTransaction>
                        <TransactionIssuer>kassa 2</TransactionIssuer>
                        <TransactionReference>121</TransactionReference>
                        <TransactionStart format="localtime">2010-08-16T17:21:45</TransactionStart>
                        <AmountRequested units="eurocent">1532</AmountRequested>
                        <AmountReceived units="eurocent">2000</AmountReceived>
                        <PayType><PayTypeCash /></PayType>
                        <IncidentCode>1</IncidentCode>
                </LastTransaction>
        </State>
</StateResponse>
```

Listing 8: StateResponse example where € 20 was received for a transaction of € 15,32 when the Shopmaster was unable to deliver change money, resulting in IncidentCode 1 (error) for the transaction and StateCode 2 (out of order) for the Shopmaster.

4.1.5 CommandRequest

```
<xs:element name="CommandRequest">
  <xs:complexType>
    <xs:complexContent>
      <xs:extension base="ASMPType">
        <xs:sequence>
          <xs:element name="Cancel">
            <xs:simpleType>
              <xs:restriction base="xs:integer">
                <xs:pattern value="0|1|3|4|101|102|103|104|105|106" />
              </xs:restriction>
            </xs:simpleType>
          </xs:element>
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
</xs:element>
```

Listing 9: Schema for the Amount-element

CommandRequest can be used to instruct the Shopmaster to do several actions, depending on the Cancel-type. Possible values:

- 0: cancel the currently active transaction without return of money
- 1: cancel with return of already inserted money (refund)
- 3: software reset to re-initialize payment components
- 4: hardware reset to reboot computer
- 101: hardware shutdown
- 102: put machine out of order (example: maintenance)
- 103: puts machine back in order after 102
- 104: reset totals of reporting
- 105: Start/stop filling Bill to Bill and/or CCTalk
- 106: Show/Hide Last Transaction(s) on machine

Example:

```
<?xml version="1.0"?>

<CommandRequest xmlns="http://download.antenor.eu/APS/ShopmasterXML"
                xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
                xsi:schemaLocation="     http://download.antenor.eu/APS/ShopmasterXML
                                         http://download.antenor.eu/APS/ShopmasterXML/ASMP.xsd"
                asmpversion="1.0">
       <Cancel>1</Cancel>
</CommandRequest>
```

Listing 10: CommandRequest example to cancel with a refund

4.1.6 CommandResponse

```
<xs:element name="CommandResponse">
  <xs:complexType>
    <xs:complexContent>
      <xs:extension base="ASMPType">
        <xs:choice>
          <xs:element name="Error" type="ErrorType" />
          <xs:element name="State" type="StateType" />
        </xs:choice>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
</xs:element>
```

Listing 11: Schema for the CommandResponse-element

CommandResponse is the reply sent after a CommandRequest, containing either an Error-element when the CommandRequest was invalid, or the State to show what the Shopmaster is doing.

Example:

```
<?xml version="1.0"?>

<CommandResponse xmlns="http://download.antenor.eu/APS/ShopmasterXML"
                xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
                xsi:schemaLocation="    http://download.antenor.eu/APS/ShopmasterXML
                                        http://download.antenor.eu/APS/ShopmasterXML/ASMP.xsd"
                asmpversion="1.0">
        <State>
                <StateCode>0</StateCode>
                <LastTransaction>
                        <TransactionIssuer>kassa 3</TransactionIssuer>
                        <TransactionReference>103</TransactionReference>
                        <TransactionStart format="localtime">2010-08-16T13:41:32</TransactionStart>
                        <AmountRequested units="eurocent">1500</AmountRequested>
                        <AmountReceived units="eurocent">0</AmountReceived>
                        <PayType><PayTypeCash /></PayType>
                        <IncidentCode>3</IncidentCode>
                </LastTransaction>
        </State>
</CommandResponse>
```

Listing 12: CommandResponse example after the above cancel with refund. Note that AmountReceived displays the amount the Shopmaster "gained" during the transaction. Because all inserted money was refunded, this is 0.

4.1.7 Error

```
<xs:element name="Error" type="ErrorType" />
```

Listing 13: Schema for the Error Root-element

Error will only be replied by the slave to the master when invalid input was received. This can be about invalid XML-syntax, unknown commands, etc

Example:

```
<?xml version="1.0"?>

<Error xmlns="http://download.antenor.eu/APS/ShopmasterXML"
                xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
                xsi:schemaLocation="    http://download.antenor.eu/APS/ShopmasterXML
                                        http://download.antenor.eu/APS/ShopmasterXML/ASMP.xsd">
        <ErrorDetails>
                <Code>15</Code>
                <Description>Malformed XML</Description>
        </ErrorDetails>
        <Received><![CDATA[
<?xml version="1.0"?>

<ebitRequest    xmlns="http://download.antenor.eu/APS/ShopmasterXML"
                xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
                xsi:schemaLocation="    http://download.antenor.eu/APS/ShopmasterXML
                                        http://download.antenor.eu/APS/ShopmasterXML/ASMP.xsd"
                asmpversion="1.0">
        <Transaction>
                <AmountRequested units="eurocent">1290</AmountRequested>
                <TransactionIssuer>kassa 5</TransactionIssuer>
                <TransactionReference>142</TransactionReference>
        </Transaction>
</DebitRequest>
]]></Received>
</Error>
```

Listing 14: Error example: typo caused ebitRequest instead of Debitrequest

Possible error codes for XML Syntax errors are defined by the MS XML 6.0 definition.
The Received-element contains the full received faulty data, encapsulated in a CDATA-section.

4.2 Re-usable elements

4.2.1 Transaction

```
<xs:complexType name="TransactionType">
  <xs:all>
    <xs:element name="AmountRequested" type="AmountType" minOccurs="1" />
    <xs:element name="AmountReceived" type="AmountType"  minOccurs="0"/>
    <xs:element name="TransactionIssuer" minOccurs="0">
      <xs:simpleType>
        <xs:restriction base="xs:string">
          <xs:maxLength value="50"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:element>
    <xs:element name="TransactionReference" type="xs:positiveInteger" minOccurs="0" />
    <xs:element name="TransactionStart" minOccurs="0">
      <xs:complexType>
        <xs:simpleContent>
          <xs:extension base="xs:dateTime">
            <xs:attribute name="format" use="required">
              <xs:simpleType>
                <xs:restriction base="xs:string">
                  <xs:pattern value="localtime" />
                </xs:restriction>
              </xs:simpleType>
            </xs:attribute>
          </xs:extension>
        </xs:simpleContent>
      </xs:complexType>
    </xs:element>
    <xs:element name="PayType" minOccurs="0">
      <xs:complexType>
        <xs:choice>
          <xs:element name="PayTypeCash" type="EmptyType" />
          <xs:element name="PayTypeCzam">
            <xs:complexType>
              <xs:sequence>
                <xs:element name="CzamErrorCode"  type="xs:string" />
                <xs:element name="CzamErrorDescription"  type="xs:string" />
              </xs:sequence>
            </xs:complexType>
          </xs:element>
        </xs:choice>
      </xs:complexType>
    </xs:element>
    <xs:element name="CardidCzam" type="xs:string" minOccurs="0"/>
    <xs:element name="IncidentCode" type="xs:positiveInteger" minOccurs="0"/>
  </xs:all>
</xs:complexType>
```

Listing 15: Schema for the Transaction-element

The transaction element is used multiple times. First it's used in the DebitRequest to initiate payment, and in every State-element it's used to specify either the currently active, the previously active transaction (or both if applicable).
IncidentCode can be:

- 1: error
- 2: transaction in progress
- 3: transaction finished in normal way
- 4: transaction cancelled by time-out
- 5: transaction cancelled (time-out or by user) but with error
- 6: transaction cancelled by user

- 11: transaction type Fill Bill To Bill and/or CCTalk
- 12: transaction type Drop bills in Bill To Bill
- 18: transaction type Drop X coins from CCTalk
- 20: transaction type ClearCounters
- 21: transaction type Drop all Coins
- Others: TBD

4.2.2 State

```
<xs:complexType name="StateType">
  <xs:sequence>
    <xs:element name="StateCode">
      <xs:simpleType>
        <xs:restriction base="xs:integer">
          <xs:pattern value="0|1|2|3|4" />
        </xs:restriction>
      </xs:simpleType>
    </xs:element>
    <xs:element name="CurrentTransaction" type="TransactionType" minOccurs="0" />
    <xs:element name="LastTransaction" type="TransactionType" minOccurs="0" />
  </xs:sequence>
</xs:complexType>
```

Listing 16: Schema for the State-element

State contains a StateCode when StateType 101 (Shopmasterstate) was asked:

- 0: idle, ready for new transaction
- 1: low, hopper low
- 2: error, out of order/error/hopper empty
- 3: busy, transaction in progress
- 4: init, initializing components

Furthermore, when a transaction is in progress, CurrentTransaction is returned, containing a transaction type element. By monitoring AmountReceived, the progress of payment can be monitored. When the Shopmaster has finished at least one transaction since start-up, the LastTransaction element is also returned.

4.2.3 Error

```
<xs:complexType name="ErrorType">
  <xs:sequence>
    <xs:element name="ErrorDetails">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="Code" type="xs:positiveInteger" />
          <xs:element name="Description" type="xs:string" />
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <xs:element name="Received" type="xs:string" />
  </xs:sequence>
</xs:complexType>
```

Listing 17: Schema for the re-usable Error-element

Errors will be commonplace when designing the master to interact with the Shopmaster. Hopefully the descriptions will be useful to you. Most likely there is a command requested at an illegal machine state.

The Received-element contains the full received faulty data, encapsulated in a CDATA-section.

Error codes defined by the Antenor machine are:

- 1: general error, see description for details

4.2.4 Amount

```
<xs:complexType name="AmountType">
  <xs:simpleContent>
    <xs:extension base="AmountValueType">
      <xs:attribute name="units" use="required">
        <xs:simpleType>
          <xs:restriction base="xs:string">
            <xs:pattern value="eurocent" />
          </xs:restriction>
        </xs:simpleType>
      </xs:attribute>
    </xs:extension>
  </xs:simpleContent>
</xs:complexType>

<xs:simpleType name="AmountValueType">
  <xs:restriction base="xs:unsignedInt">
    <xs:minInclusive value="0"/>
    <xs:maxInclusive value="9999999"/>
  </xs:restriction>
</xs:simpleType>
```

Listing 18: Schema for the Amount-element

The Amount element specifies 2 prerequisites:

- Every Amount-type element has to contain the AmountValueType argument, which has to be “eurocent”.
- The specified amount should be between 0 and 9999999 eurocent (or € 99 999,99).