



ECOMMERCE CONNECT GATEWAY. XML PAYMENT PROTOCOL.
ADMINISTRATOR GUIDE.

Version 1.0

KYIV 2020

Table of contains

Purpose	3
Possibilities	3
Brief description of commands.....	3
XMLMPI authorization	4
Transportation.....	4
Request format.....	4
Response format	5
Request on Authorization:	7
Response on Authorization:	8
Request on Refund:	9
Response on Refund:	10
Request on TransactionStateReq:.....	11
Response on TransactionStateReq:	12
Request on Pre-Authorization:	13
Response on Pre-Authorization:.....	14
Request on Post-Authorization:.....	15
Response on Post-Authorization:	16
XMLMPI authorization	17
Response codes of MPI	22
Signature Generation	26
Example of authorization request with signature	27

Purpose

The protocol is designed as an access point to the authorization system with validation and filtering functions of incoming requests. The protocol is made for banks, stores and organizations that provide 3D Secure Acquiring or that participate in the 3D Secure schema. To gain access to the protocol the organization must be PCI DSS compliant, or compliant to any other security standard that meets PCI DSS requirements as usage of the protocol involves collection and processing of confidential and strictly confidential data.

Possibilities.

With this protocol it is possible to perform the following operations

- Authorization (c);
- Reversal/return (Refund);
- Preauthorization (Preauthorization);
- Postauthorization (Postauthorization);
- Request of the transaction status (TransactionStateReq)
- Card to card transaction (C2C)
- XMLMPI authorization

Brief description of commands

Authorization.

Used to perform eCommerce transactions of Secure Channel (ECI 07) type. Also possible to perform authorization for Full 3D Secure transactions, i.e. authorization to use CAVV.

Return/Reversal.

Cancel of transaction, refund to the cardholder. Having equal command Refund, reversal and return have some differences.

Reversal – cancel of transaction before closure of operation day. Funds are returned immediately after the reversal.

Return – cancel of transaction after closure of operation day. Unlike reversal during the return there is no request to the authorization system and the refund is performed after 3 bank days because the refund is formed during clearing.

Preauthorization/Postauthorization.

Preauthorization is used to block funds without clearing.

Postauthorization is used for clearing. It is a completion of transaction initiated by preauthorization.

Request of the transaction status.

Returns code of command completion. Protocol does not guarantee 100% delivery of response with command performance results and in case of absence of answer on the expiration of timeout it is recommended to use this request.

Card to Card (C2C):

a service that allows cardholders to transfer funds between card accounts issued by the same bank.

XMLMPI authorization

Used to perform authorization for Full 3D Secure (Secure Electronic Commerce transaction) (ECI 05), (ECI 06) and allow merchant to check card for participating in program SecureCode/3dSecure .

Transportation.

To perform a transaction should be used XML method of exchanging documents, formed in accordance with the schema at <https://secure.upc.ua/go/pub/schema/xmlpay-1.13.xsd> (address schema, as well as test or production environment parameters can be found by sending a letter to ec@upc.ua)
Request delivery is used as transportation (XML of the document) using HTTP POST.

Request format.

Top level element (ROOT) is called <ECommerceConnect> and consists of <Message> or <Error>.

Element <Message> contains <XMLPayRequest> or <XMLPayResponse>. <Message> also contains attributes id (unique identifier of the message to match requests and responses) and version (current version 1.0)

<XMLPayRequest> contains elements <RequestData> and <Signature> (is not used in the current version).

Element <RequestData> contains merchant's parameters, such as <MerchantID> and <TerminalID> and element <Transactions>, that can include one or more elements <Transaction>. Element <Transaction> contains attribute id, that must be definitely used having more that one <Transaction> in <Transactions>.

Element <Transaction> can contain the following elements:

- <Authorization> - blocking with following clearing;
- <Refund> - performance of return/reversal;
- <Preauthorization> - blocking without clearing;
- <Postauthorization> - clearing of blocking set by <Preauthorization>;
- <TransactionStateReq> - request of the transaction status.

<Authorization>

Each element must include element <PayData>, that contains a mandatory element <Invoice> and <Card>, as well as optional <PAREs> (Full 3D Secure authorization using CAVV). Detailed description of Invoice и Card see in Table 1.

<Refund>

Element <Refund> includes only <RefundData>. element <RefundData> includes elements of Invoice, <AuthorizationRef> and optional element <RefundAmount>. Detailed description of elements <AuthorizationRef>

and <Invoice> see in Table 1. If there is the <RefundAmount> than the sum of return/reversal is determined by the value of element. Otherwise, the sum of return/reversal is equal to the original transaction.

<Preauthorization>

Fully repeats format of <Authorization>.

<Postauthorization>

Can be successfully completed during 30 days after <Preauthorization>.

Element <Postauthorization> contains a mandatory element

<PostauthorizationData>, that contains <Invoice>,

<PreauthorizationRef> and optional element

<PostauthorizationAmount>.

<PostauthorizationAmount> indicates the amount of completion

<Preauthorization> in minimum currency units. If

<PostauthorizationAmount> is absent than Postauthorization should be performed on the amount of original Preauthorization. Completion amount should be within Cymma +/- 20% from Preauthrorization amount.

<TransactionStateReq>

<TransactionStateReq> contains a mandatory element

<TransactionStateReqData>, that in its turn contains mandatory element

<Invoice>.

Response format

The answer is provided in elements <XMLPayResponse>, <ResponseData>, <TransactionResults> → <TransactionResult>

Structure of <TransactionResult> see in Table1

If the request had several transactions and each <Transaction> had an attribute «id», than this attribute shall be set appropriately to each value of to <TransactionResult> to match requests and responses. If in the request specified one transaction than «id» can de missed.

Table 1. Format and description of elements

Parent element	Child element	Format	Description
RequestData	MerchantID	n7	Unique merchant identification
	TerminalID	an8	Unique terminal identification
Invoice	OrderID	ans1..20	Unique order_id of our system
	Date	N12	Transaction date yyMMddHHmmss
	TotalAmount	N1..12	Transaction Amount
	Currency	N3	Transaction Currency
	Description	ans1...512	Description
Card	CardNum	N10...19	Card number
	CardType		Reserved ,No using
	ExpYear	N4	Exp year
	ExpMonth	N2	Exp Month
	CVNum	N3	CVC2/CVV2
	NameOnCard		Reserved ,No using
	ExtData		Reserved ,No using
Card	EncryptedData		Reserved ,No using
RefundData	RefundAmount	n1..12	Refund amount in Transaction Currency
AuthorizationRef PreauthorizationRef	ApprovalCode	an1..6	Approval Code received from issuer
	Rrn	n12	Reference retrieval number
TransactionResult	TranCode	n3	
	ApprovalCode	an1..6	Approval Code received from issuer
	Comment	asn64	Comment
	CVResult	a1	Value of checkCVNum , may be 'M', 'P', 'U', 'N'
	HostCode	n3	Completion code ,can be different than TranCode
	ExtData	n3	Additional data

Request on Authorization:

```
<?xml version="1.0" encoding="UTF-8"?>
<ECommerceConnect xmlns:xenc="http://www.w3.org/2001/04/xmlenc#"
  xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="https://secure.upc.ua/go/pub/schema/xmlpay-
1.4.xsd">
  <Message id="748112032436">
    <XMLPayRequest>
      <RequestData>
        <MerchantID>1700000</MerchantID>
        <TerminalID>E7000000</TerminalID>
        <Transactions>
          <Transaction Id="3">
            <Authorization>
              <PayData>
                <Invoice>
                  <OrderID>10vtlpu55</OrderID>
                  <Date>1234567890</Date>
                  <TotalAmount>5000</TotalAmount>
                  <Currency>980</Currency>
                  <Description>Rec#17</Description>
                </Invoice>
                <Card>
                  <CardNum>4999999999999011</CardNum>
                  <ExpYear>2005</ExpYear>
                  <ExpMonth>12</ExpMonth>
                </Card>
              </PayData>
            </Authorization>
          </Transaction>
        </Transactions>
      </RequestData>
    </XMLPayRequest>
  </Message>
</ECommerceConnect>
```

Response on Authorization:

```
<?xml version="1.0" encoding="UTF-8"?>
<ECommerceConnect xmlns:xenc="http://www.w3.org/2001/04/xmlenc#"
  xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="https://secure.upc.ua/go/pub/schema/xmlpay-
1.0.xsd">
  <Message id="748112032436" version="1.0">
    <XMLPayResponse>
      <ResponseData>
        <MerchantID>1700000</MerchantID>
        <TerminalID>E7000000</TerminalID>
        <TransactionResults>
          <TransactionResult>
            <TranCode>000</TranCode>
            <ApprovalCode>094224</ApprovalCode>
            <Rrn>120516000145</Rrn>
            <Comment>Approved</Comment>
            <HostCode>000</HostCode>
          </TransactionResult>
        </TransactionResults>
      </ResponseData>
    </XMLPayResponse>
  </Message>
</ECommerceConnect>
```


Request on Refund:

```
<?xml version="1.0" encoding="UTF-8"?>
<ECommerceConnect xmlns:xenc="http://www.w3.org/2001/04/xmlenc#"
xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="https://secure.upc.ua/go/pub/schema/xmlpay-
1.2.xsd">
<Message id="1653689" version="1.0">
  <XMLPayRequest>
    <RequestData>
      <MerchantID>1700000</MerchantID>
      <TerminalID>E7000000</TerminalID>
      <Transactions>
        <Transaction>
          <Refund>
            <RefundData>
              <Invoice>
                <OrderID>xmlPay504</OrderID>
                <Date>091112174516</Date>
                <TotalAmount>8000</TotalAmount>
                <Currency>980</Currency>
                <Description>Test xml</Description>
              </Invoice>
              <AuthorizationRef>
                <ApprovalCode>547042</ApprovalCode>
                <Rrn>111309003550</Rrn>
              </AuthorizationRef>
              <RefundAmount>7000</RefundAmount>
            </RefundData>
          </Refund>
        </Transaction>
      </Transactions>
    </RequestData>
  </XMLPayRequest>
</Message>
</ECommerceConnect>
```

Response on Refund:

```
<?xml version="1.0" encoding="UTF-8"?>
<ECommerceConnect xmlns:xenc="http://www.w3.org/2001/04/xmlenc#"
xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="https://secure.upc.ua/go/pub/schema/xmlpay-
1.2.xsd">
  <Message id="1653689" version="1.0">
    <XMLPayResponse>
      <ResponseData>
        <MerchantID>1700000</MerchantID>
        <TerminalID>E7000000</TerminalID>
        <TransactionResults>
          <TransactionResult>
            <TranCode>000</TranCode>
          </TransactionResult>
        </TransactionResults>
      </ResponseData>
    </XMLPayResponse>
  </Message>
</ECommerceConnect>
```

Request on TransactionStateReq:

```
<?xml version="1.0" encoding="UTF-8" ?>
<ECommerceConnect xmlns:xenc="http://www.w3.org/2001/04/xmlenc#"
xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="http://localhost:8080/ecgtest/pub/schema/xmlpay-1.4.xsd">
  <Message id="748112032436" version="1.0">
    <XMLPayRequest>
      <RequestData>
        <MerchantID>1752550</MerchantID>
        <TerminalID>E7879900</TerminalID>
      <Transactions>
        <Transaction id="000302">
          <TransactionStateReq>
            <TransactionStateReqData>
              <Invoice>
                <OrderID>xmlPay053</OrderID>
                <Date>090717105300</Date>
                <TotalAmount>21</TotalAmount>
                <Currency>980</Currency>
              </Invoice>
            </TransactionStateReqData>
          </TransactionStateReq>
        </Transaction>
      </Transactions>
    </RequestData>
  </XMLPayRequest>
</Message>
</ECommerceConnect>
```

Response on TransactionStateReq:

```
<?xml version="1.0" encoding="UTF-8"?>
<ECommerceConnect xmlns:xenc="http://www.w3.org/2001/04/xmlenc#"
xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="https://secure.upc.ua/go/pub/schema/xmlpay-
1.4.xsd">
  <Message id="748112032437" version="1.0">
    <XMLPayResponse>
      <ResponseData>
        <MerchantID>1700000</MerchantID>
        <TerminalID>E7000000</TerminalID>
        <TransactionResults>
          <TransactionResult>
            <TranCode>000</TranCode>
            <ApprovalCode>913688</ApprovalCode>
            <Rrn>072116013654</Rrn>
            <Comment>Approved</Comment>
          </TransactionResult>
        </TransactionResults>
      </ResponseData>
    </XMLPayResponse>
  </Message>
</ECommerceConnect>
```

Request on Pre-Authorization:

```
<ECommerceConnect xmlns:xenc="http://www.w3.org/2001/04/xmlenc#"
xmlns:ds="http://www.w3.org/2000/09/xmldsig#" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xsi:noNamespaceSchemaLocation="https://secure.upc.ua/go/pub/schema/xmlpay-1.4.xsd">
<Message id="748112036" version="1.0">
  <XMLPayRequest>
    <RequestData>
      <MerchantID>1752739</MerchantID>
      <TerminalID>E7880539</TerminalID>
    </RequestData>
    <Transactions>
      <Transaction id="999999">
        <Preauthorization>
          <PayData>
            <Invoice>
              <OrderID>test1</OrderID>
              <Date>1107071171700</Date>
              <TotalAmount>100</TotalAmount>
              <Currency>980</Currency>
              <Description>bann</Description>
            </Invoice>
            <Card>
              <CardNum>4999*****99990011</CardNum>
              <ExpYear>2010</ExpYear>
              <ExpMonth>07</ExpMonth>
              <CVNum>123</CVNum>
            </Card>
          </PayData>
        </Preauthorization>
      </Transaction>
    </Transactions>
  </XMLPayRequest>
</Message>
</ECommerceConnect>
```

Response on Pre-Authorization:

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

```
<ECommerceConnect xmlns:xenc="http://www.w3.org/2001/04/xmlenc#"
xmlns:ds="http://www.w3.org/2000/09/xmldsig#" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xsi:noNamespaceSchemaLocation="https://secure.upc.ua/go/pub/schema/xmlpay-1.4.xsd">
```

```
  <Message id="748112036" Version="1.0">
```

```
    <XMLPayResponse>
```

```
      <ResponseData>
```

```
        <MerchantID>1752739</MerchantID>
```

```
        <TerminalID>E7880539</TerminalID>
```

```
        <TransactionResults>
```

```
          <TransactionResult>
```

```
            <TranCode>000</TranCode>
```

```
            <ApprovalCode>395006</ApprovalCode>
```

```
            <Rrn>070711413549</Rrn>
```

```
          </TransactionResult>
```

```
        </TransactionResults>
```

```
      </ResponseData>
```

```
    </XMLPayResponse>
```

```
  </Message>
```

```
</ECommerceConnect>
```

Request on Post-Authorization:

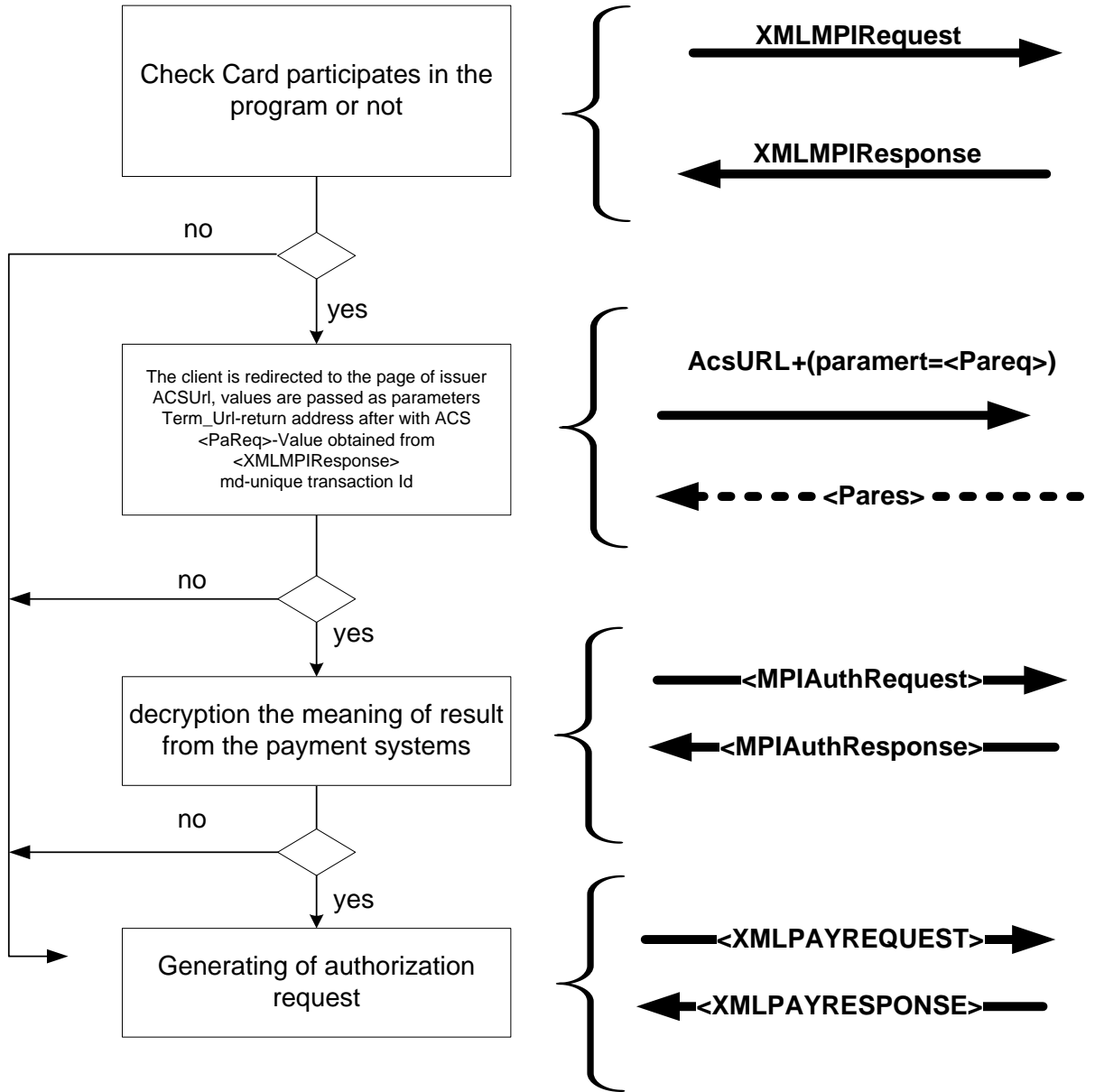
```
<ECommerceConnect xmlns:xenc="http://www.w3.org/2001/04/xmlenc#"
xmlns:ds="http://www.w3.org/2000/09/xmldsig#" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xsi:noNamespaceSchemaLocation="https://secure.upc.ua/go/pub/schema/xmlpay-1.4.xsd">
<Message id="748112036" version="1.0">
  <XMLPayRequest>
    <RequestData>
      <MerchantID>1752739</MerchantID>
      <TerminalID>E7880539</TerminalID>
    </RequestData>
    <Transactions>
      <Transaction id="999999">
        <Postauthorization>
          <PostauthorizationData>
            <Invoice>
              <OrderID>test1</OrderID>
              <Date>1107071171700</Date>
              <TotalAmount>100</TotalAmount>
              <Currency>980</Currency>
              <Description>BANAN</Description>
            </Invoice>
            <PreauthorizationRef>
              <ApprovalCode>395006</ApprovalCode>
              <Rrn>070711413549</Rrn>
            </PreauthorizationRef>
            <PostauthorizationAmount>120</PostauthorizationAmount>
          </PostauthorizationData>
        </Postauthorization>
      </Transaction>
    </Transactions>
  </XMLPayRequest>
</Message>
</ECommerceConnect>
```

Response on Post-Authorization:

```
<?xml version="1.0" encoding="UTF-8" ?>
<ECommerceConnect xmlns:xenc="http://www.w3.org/2001/04/xmlenc#"
xmlns:ds="http://www.w3.org/2000/09/xmldsig#" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xsi:noNamespaceSchemaLocation="https://secure.upc.ua/go/pub/schema/xmlpay-1.4.xsd">
  <Message id="748112036" version="1.0">
    <XMLPayResponse>
      <ResponseData>
        <MerchantID>1752739</MerchantID>
        <TerminalID>E7880539</TerminalID>
        <TransactionResults>
          <TransactionResult>
            <TranCode>000</TranCode>
          </TransactionResult>
        </TransactionResults>
      </ResponseData>
    </XMLPayResponse>
  </Message>
</ECommerceConnect>
```


XMLMPI authorization

There are two possible scenarios of XMLMPI authorization



Example:

a) When a card participates in programs SecureCode/3dSecure

1) To check the card needs to send a request by post to <MPIEnrolRequest> by method Post to url <https://secure.upc.ua/go/service/03>

Example :

```
<ECommerceConnect xmlns:xenc="http://www.w3.org/2001/04/xmlenc#"
xmlns:ds="http://www.w3.org/2000/09/xmldsig#" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xsi:noNamespaceSchemaLocation="https://secure.upc.ua/ecgtest/pub/schema/xmlpay-
1.5.xsd">
<Message id="274318361" version="1.0">
  <XMLMPIRequest>
    <MerchantID>your mid</MerchantID>
    <TerminalID>your tid</TerminalID>
    <MPIRequest ID="11122">
      <MPIEnrolRequest>
        <CardNum>49999999999990011</CardNum>
        <ExpYear>2013</ExpYear>
        <ExpMonth>11</ExpMonth>
        <TotalAmount>250000</TotalAmount>
        <Currency>980</Currency>
        <Description>TestMpiSpirin1</Description>
        <DeviceCategory>0</DeviceCategory>
      </MPIEnrolRequest>
    </MPIRequest>
  </XMLMPIRequest>
</Message>
</ECommerceConnect>
```

If the card participates in SecureCode/3dSecure the response from the gateway will contain tag <enrolled> Y, Code=0 in <MPIEnrolResponse><?xml version="1.0" encoding="UTF-8"?>

```
<ECommerceConnect xmlns: xenc="http://www.w3.org/2001/04/xmlenc#"
xmlns:ds="http://www.w3.org/2000/09/xmldsig#" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xsi:noNamespaceSchemaLocation="https://secure.upc.ua/go/pub/schema/xmlpay-
1.5.xsd"><Message id="74318361" version="1.0">
<XMLMPIResponse><MPIResponse>
<Code>0</Code>
<MPIEnrolResponse>
<Enrolled>Y</Enrolled>
<AcsURL>https://secure.upc.ua/ecgtest/acstest?cardNumber=49999999999990011</AcsURL><PaReq
>eJxVUttO20AQ/RXL7/Hu+p5osigLqRrRhAicSjxa9oq4YDv4Qpl3BH3vWx/7DQhRtUoD/YX1H3U2hNDKs
nbOmTOX3Rk4WKaX2rUoyiTpujozqK6JLMrjJDv6tPgfcvXDzgEs0KI/qml6kJwGImyDM+FlsRdfRKeiCtm
WR6lrpccm3l4Or5D3bbN2p7r21TnMOMhjsOuEsdChgnkFWLklpgFWcUhhK7eDcfcZnbboUB2EFJRDPvct
ahnKvoFQhamgg9aR5/OguMj4/B4ZEx7QLYsRHmdVcWK+9QG8ggqLi75rKrmHUIWi4UhWhfXqyg/MKI8
NeoQiPIDEWtnUiurxHzLJOajfm8xCoaLcTCgoyCqxp/PVuNgyl4Ggy4QpYA4rAQ3KTOpw2yN2R3L7pjY8Z
aHMFwNcM+xDdfH271AmKsqvb1Puf6IAN+9wLGseNvHvHsEYjnPM4EKfMu9DbEoly6/yz/yt7xvbuW9Jp
+bL/jfyA1SP+STfGq+avJRPst1c4ckUs1t800+aPKnJh/QepRrFY2nkV+SG5Q3NyheY9DGwPuolkDe3ufwg
xpeVKKxeY5vWqxNTYc5LmVqjFuHajjBMeCOWNuOFQCiQsluQ8hUU9D6b+v+AiX0/cg=</PaReq><XID>
12061112-
860797</XID></MPIEnrolResponse></MPIResponse></XMLMPIResponse></Message></ECommerceC
onnect>
```

If the card participates in SecureCode/3dSecure

It's necessary to make redirect of client to the page <AcUrl> with parameters <Pareq>, MD-a unique value for a particular transaction, TermUrl- return address, where client must be returned

Response from the payment system will include MD, Pares and will be delivered to TermUrl

Response from the payment system is necessary to convert to MPI format, by sending parameter <Pares> with Post method to <https://secure.upc.ua/go/service/03>

```
<ECommerceConnect xmlns:xenc="http://www.w3.org/2001/04/xmlenc#"
xmlns:ds="http://www.w3.org/2000/09/xmldsig#" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xsi:noNamespaceSchemaLocation="https://secure.upc.ua/ecgtest/pub/schema/xmlpay-
1.5.xsd">
<Message id="7418361" version="1.0">
  <XMLMPIRequest>
    <MerchantID>your mid</MerchantID>
    <TerminalID> your tid </TerminalID>
  <MPIRequest ID="1111">
    <MPIAuthRequest>
      <PaRes>
        eNrNWGmzokrS/isdPR+Ze9IVbnhORBW7ArJlfmOTXRRQkF8/qN2nz3R0xNt3PrwzGgZVaeVWflkUms7a5OEs5L
o2ibvazXpuiBNvutx21c9MjMLTpJLDFs8vgsKX85PekVjC4bCmeVirWFF39c6MJPuxfEY/S7bLWm7vDm9439ifxJr9Pt0NqG
NsuDUv6+D6AJ7Z3CKYbG1ui36bpOWpl7X5DYkniQX9M1+oNPvz5G3ezOmMfvKgcG1ZYHzeYx1Y56rfDvmu3gis2/rdHH
inUc9Mk7geEERuPUF5z6i6T+ImbRT/r6/BAH6uY6y17S1GK1Rj+T1vPOtckpur8zq5nnY7ZOxnNzSuYVs3cf4zX6w7pzcHrHP
n1WJEBNsmfq2t6/r/u8/mwV/bCKmjU86eud/pr9+6v0W+jdRTcbu8AAMg6mpkYj5GaaYJAg9dn9va5ZJ1E+TtGz0bNzycXqN
Kmfzsfpj674Q1+jAFfZ7s+9rK09OsrE2+jHV16t6+Zn1//gtFh2H4cyD/bNoUJWZHUixB5wVxl6f/+PriSmL5dGz+FhsbnJpTHg
VVPgX9HBpq0mdN/OXDtl+Jsc2HJBw1efaPwDQfEU6d/nhQMBKnZ5nor4V+8ux3tPxsbnNsFf3RZGD8U/CTofW0mx+QREck
Xx5Tfvv7jbyUJl6dJ1/8nNn2357OE7/LcoLom71F78EoxiTFxaYdMNJ3Q7iYxEmSw5u0732vLgV1w4puHr+P8tG2vhcbtjgub8Ni
31Ny4l6XZDsoHLUTRex0xNK2ijQZu00dVFRH0nvXqaWMSQIbIaVuuVGAxzm1VZ5nEhfLl85X8vtqe5c5CztqZTVrxzdXse
FERSHzpa96x00hZGeXOIzBtk9LJ6RW+hUD5nGbtuKZJQR0DMJFRED1giUddzRWJ9c6yZsL22Nvn47qm5fb5P7yak9jDBf0
wWtkXcMimXEjmNOO1d7cpM2PeRLDu5t3wSgz/Pjd5jF5xrHbP7/snDfTAn+w/s4G8+xNewbakMfJf3jNqzq/pV+sjsqT/v7PL
+ybs12jP6t56mWTtp8+8VDwoAVO55lrEMjPujGYCuzgYzglQ2gWJebkiXazpp71AZkw3r2irGwBn+Zts50wWacDgBWiAYT/xigp
KEeAODzOVdV115GygvFRzIWhsKBw2DsaPM1heX7TO3uChc0TwwqeXR2GG/ufp78xwSdBay0J7nROBplcWLU0QwReAJ
WOAx12jidRVgTz1gVMWDhXvBXstUSO05e5Zvy6PGgbvGyYpNg/a9G+0qhk+ydbnW7ccCD/bqu10VksVrfwFJ8PopMaJ/c
cirNtdZQ6T1tcLLF5S4XgKYMD1Y1TaVBxtIONm1C10mGT+pxrGBw/Mt7sy/lQV9WB39BuXU2+N8uX4MPfIX7MbW0uIJ
BQZfH48I2LlqllhyRn8BAYDgCUqAGOhmxbhanB5tgKL26bHN3jiAuWpvaMj16lFCqODSldmMfOfWKEremtEcK7tiPZ02U
HSoqeM3jKZFd8pMtC0q0lBHZJ5mY0075itZTSqSEKJ9Q38l7TK97ihdSbg+R/eqasl6dThkdxv5hybHnfWdhiDfBLA+5+dUUR
YHUA+Wu11UIUuZ+sFY6V1qDzAEDwIaSofOILQcbuGHEfXPTgSGhEDhz6UwA4MDxsfeSpfliB7wU2pXCb1rhnpEGLugT
vcUtCStpxlbdAqjPtea8+8VDwoAVO55lrEMjPujGYCuzgYzglQ2gWJebkiXazpp71AZkw3r2irGwBn+Zts50wWacDgBWiAYT/xigp
ANKGeVFG20m7OUXlii4SFLsO61egwUtXoCMbynSKHwUpkC06WcUkq5Enli3qzrsfaG50NhNxECHiHaB6RyQt1Y5MqHa
Gz8gsCvmmAqoQXpBYAFhOggNcAQ757hp1ekkmKQOF9E0UeUdtx+W0cCnz7X2Cy7TYYYxiD5i0gb0CzIBrxbz+wdLidB
LIBh4qLLOIA++vB18CA1HAinvCUSGxRJYKHdmBr3o+iw7L9i5RWJ/nmMesi+7jnAeg4KF6ASiHrNstfGH/xP2njEgcIb3Kf7
czS0kjRcMcvd0dWFUNpeoWDKfMDiDL6nU1TUWh5c93gcmF1E93J5w52jw4Mw0D58hT6ucmnrnFLP34f0j9IE393fxjxWM9l
v7+6Z8Qnma9P8u0m2nHITP0U9htDjmu0PPvT9g8yX3/wVudy+4XT3LFwQqCwx/klzn3u8g9Hh53R3G3e23Usw5DqqXrSsfl
fuEq+P159yAz9yYz8tgBc/cdeQFSzMI5vMMOzHlxRWh0V8cN5ZyCQxkSXE1ae0YR4LsbTpy5ckoyKTEw6aZaccs3gvuAikT
h3AtZLFpMssm+SUDikRYnjcmJ/YnTEocqmPzyYgnekEXPRp2SxmFrqmlyMnZ1ts9JcT5drenEGROWIGprzzv+4mE4HCLykbt
La28TVbLWDufj+V+QsH5HpB3Q6wca3FrKpk6FXRohkv0Yq36PkTorK51C+7tfmi82+gMdxYnzqbfNcKwqg9FmqGK6LZnc
VmfuE1Ext5+WeYHRJkbfc7AA1K4caOTkjHlJG0sLu+8MRWBhOKIw/vkXgsBdqa8oVzSHiKP3MprIW+0Vxk2mO/Vy645
pHazeV7uTB4ysdtE0PhTYGI5wj4AUR+ATSYlpezEVmwB6wJgAOTqrRDazxhHFx7ldcZ+H1FXSvzjJTebesbMPmbyqLfe826/
8gNEeJA8ELsmfA/wbZn1NLFp7pOkhZpKlzp6wVMjE/xx2nEt6dVjxp2Az9o1bwd6/437M9Tfn81+VDhtwAH+9vQfNM15vL0
J5ULzKiZseKPSoc00k6xQr2K0/6nsNCDNXruMiI1Ubec5duZ+uSRFyUTYGY12tGVu4MEQUx1V1Ydpw+dKa54z0EN7ar0Az
c7ZJcEK2+TSU8RfnkyOoykVESqbsTNUM3XloFjVsNsXu4eLhL1UqO/ngSDN65D56Aq/LBx2ureYrdNL5mDGGU80q5rVi
Yg9A0zRRHikFh7IRjmvPOnqWrJUVhO1rXoBW9+b2Q6f58PYozCuk2qnlzJJaUkiz7UQxqewVWBylYp1tWm11wU13lta
Bsc1q57Ksle3cODl+fhAbEo7actj2tDhAFFXubBVgTh1ojTC2zFQ4n2y8DcET6NUD9K5KwdiYRTzSYHVs4vmh0fJMgQVP
Dv2gUs/vwEY3FyeRBU+4yPmUsObMc3dyaqu3wc04tRkckIEUkNkaUeun36RQ2A3Qygp9qAj/sxYpWypTTPMtHzab917G
FQs/req7kTzo01uUEvLuhRXPSP3fN5kTmxVodhZcFFJTFjmauld/fy8UxcTxf0ChdZW+Cr5MFLk8bBS9L+njN0WMobrM+Cn
VSgUcJz0z9KtivrE2+lhws/ziOGKkt1Mxoqclsh5mkpv220PGxQbmdNelEjJ0uhi5c3UmWQJeNvvTHJDiNmhsZcg1ehWO+
JBX3cHc6DMjgr0fH52gIEJs5vjkwFMqtGunRgCBcelioAIRkrWgXc7+AYo7hlxOHMaThiugpQvBvQrGjA1ZvIEc1YMrnm8q
nK4Li4Kz3JrKaTihZtsNjh8rK2cfEmnJV0r/sSN7z9qmNgf9xfoB93Gj9u0553pc9r3sft3ufr338BTOAwMw==
```

```

    </PaRes>
      </MPIAuthRequest>
    </MPIRequest>
  </XMLMPIRequest>
</Message>
</ECommerceConnect>

```

Response from UPC contains data for generation authorization request/

```

<?xml version="1.0" encoding="UTF-8" ?>
<ECommerceConnect xmlns:xenc="http://www.w3.org/2001/04/xmlenc#" xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="https://secure.upc.ua/go/pub/schema/xmlpay-1.5.xsd">
  <Message id="7418361" version="1.0">
    <XMLMPIResponse>
      <MPIResponse>
        <Code>0</Code>
        <MPIAuthResponse>
          <Status>Y</Status>
          <CAVV>AAABCUNReQAAABMhNFF5AAAAAAA=</CAVV>
          <ECI>05</ECI>
          <CavvAlgorithm>2</CavvAlgorithm>
        </MPIAuthResponse>
      </MPIResponse>
    </XMLMPIResponse>
  </Message>
</ECommerceConnect>

```

Parameters, which are using for the final authorization, must be transferred to url

<https://secure.upc.ua/go/service/02>

```

<ECommerceConnect xmlns:xenc="http://www.w3.org/2001/04/xmlenc#" xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="https://secure.upc.ua/ecgtest/pub/schema/xmlpay-1.5.xsd">
  <Message id="748112036" version="1.0">
    <XMLPayRequest>
      <RequestData>
        <MerchantID>your tid</MerchantID>
        <TerminalID>your mid</TerminalID>
      </RequestData>
    </XMLPayRequest>
  </Message>
</ECommerceConnect>

```

```

        </Authorization>
</Transaction>

</Transactions>
  </RequestData>
  </XMLPayRequest>
</Message>
</ECommerceConnect>

```

b) When a card doesn't participate in program SecureCode/3dSecure
If the card doesn't participate in SecureCode/3dSecure the response from the gateway will contain tag <enrolled> <> "N" , Code<>0 в
<MPIEnrolResponse><?xml version="1.0" encoding="UTF-8"?>

Example:

```

<ECommerceConnect xmlns:xenc="http://www.w3.org/2001/04/xmlenc#" xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="https://secure.upc.ua/go/pub/schema/xmlpay-1.5.xsd"><Message id="274318361"
version="1.0">
  <XMLMPIResponse>
    <MPIResponse>
      <Code>1007</Code>
      <MPIEnrolResponse>
        <Enrolled>N</Enrolled>
        <XID>12070218-869140</XID>
      </MPIEnrolResponse>
    </MPIResponse>
  </XMLMPIResponse>
</Message>
</ECommerceConnect>

```

It's necessary to generate authorisation request

```

<ECommerceConnect xmlns:xenc="http://www.w3.org/2001/04/xmlenc#" xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="https://secure.upc.ua/ecgtest/pub/schema/xmlpay-1.5.xsd">
<Message id="748112036" version="1.0">
  <XMLPayRequest>
    <RequestData>
      <MerchantID>your tid</MerchantID>
      <TerminalID>your mid</TerminalID>
    </RequestData>
  </XMLPayRequest>
</Message>
</ECommerceConnect>

<Transactions>
  <Transaction id="12">
    <Authorization>
      <PayData>
        <Invoice>
          <OrderID>98</OrderID>
          <Date>100510105300</Date>
          <TotalAmount>2500000</TotalAmount>
          <Currency>980</Currency>
          <Description>TestMpiSpirin1</Description>
        </Invoice>
        <Card>
          <CardNum>4999999999990011</CardNum>
          <ExpYear>2013</ExpYear>
          <ExpMonth>11</ExpMonth>
          <CVNum>***</CVNum>
        </Card>
        <XID>12070218-869140</XID>
      </PayData>
    </Authorization>
  </Transaction>
</Transactions>
  </RequestData>
  </XMLPayRequest>
</Message>
</ECommerceConnect>

```

Response codes of MPI

MPI_ERROR_CODE_ID	DESCRIPTION
-1	Other/Not listed error
0	Approved
1	Root element invalid.
2	Message element not a defined message.
3	Required element missing.
4	Critical element not recognized.
5	Format of one or more elements is invalid according to the specification.
6	Protocol version too old.
1001	Cannot process payer authentication request
1003	Invalid Merchant ID/Acquirer BIN
1004	A required parameter is missing
1005	Invalid PAN
1006	Visa directory cache error
1007	Non participating PAN
1008	Cannot process payer authentication request
1009	Invalid currency code
1010	Invalid transaction ID
1014	Invalid merchant transaction stamp
1015	Invalid Expiry Date
1017	Invalid Password.
1018	Invalid Total Amount
1019	Invalid Purchase Description
1020	Error in processing VEReq at 3-D Secure layer
1021	Error in processing VEReq at HTTP layer
1022	Verify enrolment processing error
1024	VEReq processing error
1025	VEReq processing error
1040	Message conversion error
1041	Message conversion error
1042	Message conversion error
1060	Error in connection to Visa directory
1061	Error in connection to Visa directory
1062	Error in connection to Visa directory
1063	Time out error in connection to Visa directory
1065	VERes processing error
1066	VERes processing error
1067	Connection to Visa directory cannot be closed
1069	None of the directory servers are available
1080	VERes processing error
1081	VERes processing error
1084	VERes format error
1085	VERes format error

1090	DOM processing error
1091	DOM processing error
1092	SAX processing error
1093	DOM processing error
1094	DOM processing error
1100	VERes processing error
1101	VERes processing error
1104	VERes processing error
1105	PARes processing error
1106	Cannot process payer authentication response
1107	Cannot process payer authentication response
1108	Cannot process payer authentication response
1109	PAN is not enrolled
1111	PAReq processing error
1120	PARes processing error
1121	PARes processing error
1122	PARes processing error
1123	PARes processing error
1124	PARes processing error
1125	PARes processing error
1126	PARes signature verification failed
1131	Cannot process payer authentication request
1132	Cannot process payer authentication request
1133	Cannot process payer authentication request
1134	Invalid PARes
1136	Cannot process payer authentication request
1137	Cannot process payer authentication request
1138	PARes processing error
1139	HTTP request processing error
1140	Cannot process payer authentication request
1141	Invalid PARes
1142	Invalid PARes
1143	Cardholder could not be authenticated
1144	PARes verification error
1145	Cannot process payer authentication request
1146	HTTP PUT error
1150	AAV verification error
1151	AAV verification error
1153	AAV verification error
1154	AAV verification error
1155	Cannot process SPA request
1156	Cannot process SPA request
1157	AAV verification error
1158	A required parameter is missing in the HTTP request
1160	Cannot process payer authentication request
1161	Cannot process payer authentication request

1163	Cannot process payer authentication request
1180	DOM processing error
1181	DOM processing error
1182	DOM processing error
1183	SAX processing error
1184	DOM processing error
1190	PARes timeout
1200	A required element was not found
1201	Invalid request
1210	3-D Secure protocol error
1220	Invalid license key
1230	Unsupported request
1240	The interface is disabled for this merchant
1250	Cannot log transaction
1260	Cannot be mapped to a valid transaction
1270	Merchant ID does not uniquely identify this account. Please specify Acquirer BIN in order to proceed.
1280	Invalid Merchant Profile ID (PID)
2000	Error loading startup info
2001	Configuration file (config.xml) is invalid.
2002	File not found
2003	Error reading file
2004	Error parsing config.xml
2005	Unable to read the key from KeyStore
2006	Unable to read the keystore
2007	Error reading the Certificate
2008	Class not found
2009	No such algorithm
2010	Incorrect password
2011	The content of one or more elements is invalid
3000	Invalid request length
3001	Invalid http request
3002	permission denied
3003	Unhandled exception
3004	Invalid parameter
3005	Error in Inflating PARes ver 1.0.x
3006	Error in Deflating PAREq ver 1.0.x
3007	Invalid merchant
3008	Error creating Success-Page
3009	Session has expired
3010	Merchant has no valid license
3011	CRRes processing failed
3012	Invalid XML message
3013	timed out reading socket
3014	Invalid session
3015	System currently unavailable, please try again in a few

	minutes
3016	Decryption error
3017	Error in connecting to database
3018	Error reading keystore from database
3019	Error saving transaction
3020	Database exception
3021	Invalid Acquirer BIN for merchant.
109	Internal error in MPI interactor

Signature Generation

This guide was developed for online merchant administrators and web programmers to simplify payment gateway connection procedure and solve a number of issues arising while working with digital signature of messages.

Besides this guide, this archive contains batch-files; batch-files processing procedure is described below.

OpenSSL installation

Prior to operation, Win32 OpenSSL should be downloaded from <https://www.openssl.org/source/>.

[Win32 OpenSSL v0.9.8e](#) has been used for this guide.

After installation, path to the bin folder should be registered in the “Path” variable (e.g., path=C:\OpenSSL\bin)

Key Generation

Key generation and exchange is carried out after a registration request has been submitted and Merchant attributes have been designated, in particular a MerchantID.

Before key generation the config.dat file should be edited according to data of the merchant being connected. Config.dat data do not need to strictly comply with the data on connection request, the data are not used for signature generation or verification and serve only for file certificate identification.

The run.bat command with MerchantID parameter (e.g., run.bat 1770000) generates three files:

1770000.pem – private key

1770000.pub – public key

1770000.crt – certificate

Certificate file (*.crt) should be sent to UPC for Merchant message authenticity check by the gateway.

Example of authorization request with signature:

```
<?xml version="1.0" encoding="UTF-8"?>
<ECommerceConnect xsi:noNamespaceSchemaLocation="https://secure.upc.ua/go/pub/schema/xmlpay-
1.2.xsd" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xenc="http://www.w3.org/2001/04/xmlenc#"
xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
<Message version="1.0" id="11187304">
  <XMLPayRequest>
    <RequestData>
      <MerchantID>6369938</MerchantID>
      <TerminalID>P0142935</TerminalID>
      <Transactions>
        <Transaction>
          <Authorization>
            <PayData>
              <Invoice>
                <OrderID>10215887</OrderID>
                <Date>100802120043</Date>
                <TotalAmount>3000</TotalAmount>
                <Currency>980</Currency>
                <Description/>
              </Invoice>
              <Card>
                <CardNum>3924</CardNum>
                <ExpYear>2011</ExpYear>
                <ExpMonth>01</ExpMonth>
                <CVNum>NOCVV</CVNum>
              </Card>
            </PayData>
          </Authorization>
        </Transaction>
      </Transactions>
    </RequestData>
  </XMLPayRequest>
</Message>
<Signature xmlns="http://www.w3.org/2000/09/xmldsig#">
  <SignedInfo>
    <CanonicalizationMethod Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#" />
    <SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1" />
    <Reference URI="">
      <Transforms>
        <Transform Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature" />
      </Transforms>
      <DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1" />
      <DigestValue>2WLLLjQi3irizzqve+2orm92ANU=</DigestValue>
    </Reference>
  </SignedInfo>
  <SignatureValue>Bo/60T8vYvOXY+ggZuGZNVWgxFtVmAxAW56Euyj+OJzskZRN3ffVLyWyEK9
DpHy++ecctDGcGRd+
Ok5tArpvV8+dTS0YI2inr/qj1obX3AoyRP0xr5AXqS+AYcZIZ7IHEAodoajr6DtR9PwtV01fO/TK
zcgzAyJTWsObDdUaZQ7Kz+Z957uK1e9RXW60g7cbNfv4LIWi1GmnSgK11vTQBqmNmjZhzaTV
OHgp qorxU7P82157id2QeWrFhoJzzXbpt4U+tsoFdb+kEhL5Lgsp6eEGyTIZWi9wQPuW3FEIelp9Zd1A
Tx5HjdiVTiXNaRpmegksaTD7bHkQuT9vA3Qhw==</SignatureValue>
</Signature>
</ECommerceConnect>
```

Source code:

```
String xml = "<?xml version=\"1.0\" encoding=\"UTF-8\"?>\n" +
" <ECommerceConnect xmlns:xenc=\"http://www.w3.org/2001/04/xmlenc#\"\n" +
"   xmlns:ds=\"http://www.w3.org/2000/09/xmldsig#\"\n" +
"   xmlns:xsi=\"http://www.w3.org/2001/XMLSchema-instance\"\n" +
"
xsi:noNamespaceSchemaLocation=\"https://secure.upc.ua/go/pub/schema/xmlpay-1.7.xsd\">\n"
+
" <Message id=\"11187304\" version=\"1.0\">\n" +
" <XMLPayRequest>\n" +
" <RequestData>\n" +
" <MerchantID></MerchantID>\n" +
" <TerminalID>5</TerminalID>\n" +
" <Transactions>\n" +
" <Transaction>\n" +
" <Authorization>\n" +
" <PayData>\n" +
" <Invoice>\n" +
" <OrderID>10215887</OrderID>\n" +
" <Date>100802120043</Date>\n" +
" <TotalAmount>3000</TotalAmount>\n" +
" <Currency>980</Currency>\n" +
" <Description></Description>\n" +
" </Invoice>\n" +
" <Card>\n" +
" <CardNum>3924</CardNum>\n" +
" <ExpYear>2011</ExpYear>\n" +
" <ExpMonth>01</ExpMonth>\n" +
" <CVNum>NOCVV</CVNum>\n" +
" </Card>\n" +
" </PayData>\n" +
" </Authorization>\n" +
" </Transaction>\n" +
" </Transactions>\n" +
" </RequestData>\n" +
" </XMLPayRequest>\n" +
" </Message>\n" +
" </ECommerceConnect>";
```

```
KeyStore ks = KeyStore.getInstance("JKS");
// get user password and file input stream
char[] password = "123456".toCharArray();
java.io.FileInputStream fis =
    new java.io.FileInputStream("D:/test/new_test.jks");
ks.load(fis, password);
fis.close();

KeyStore.PrivateKeyEntry pkEntry =
(KeyStore.PrivateKeyEntry)ks.getEntry("merchant_pairr", new
KeyStore.PasswordProtection("123456".toCharArray()));
java.security.PrivateKey privateKey = pkEntry.getPrivateKey();
```

```

    // sign
    String providerName = System.getProperty("jsr105Provider",
"org.jcp.xml.dsig.internal.dom.XMLDSigRI");
    XMLSignatureFactory fac = XMLSignatureFactory.getInstance("DOM", (Provider)
Class.forName(providerName).newInstance());
    // Create a Reference to the enveloped document (isigning the whole document, so a URI
of "" signifies
    DigestMethod dm = fac.newDigestMethod(DigestMethod.SHA1, null);
    Transform transform = fac.newTransform(Transform.ENVELOPED,
(TransformParameterSpec) null);
    Reference ref = fac.newReference("", dm, Collections.singletonList(transform), null,
null);
    // Create the SignedInfo.
    SignedInfo si = fac.newSignedInfo(
fac.newCanonicalizationMethod(CanonicalizationMethod.EXCLUSIVE, (C14NMethodParamete
rSpec) null),
    fac.newSignatureMethod(SignatureMethod.RSA_SHA1, null),
    Collections.singletonList(ref)
);

    CertificateFactory cf = CertificateFactory.getInstance("X.509");
    X509Certificate cert = (X509Certificate)ks.getCertificate("merchant_pairr");
    // Create the KeyInfo containing the X509Data.
    KeyInfoFactory kif = fac.getKeyInfoFactory();
    List x509Content = new ArrayList();
    x509Content.add(cert.getSubjectX500Principal().getName());
    x509Content.add(cert);
    X509Data xd = kif.newX509Data(x509Content);
    KeyInfo ki = kif.newKeyInfo(Collections.singletonList(xd));
    // Instantiate the document to be signed.
    DocumentBuilderFactory dbf = DocumentBuilderFactory.newInstance();
    ByteArrayInputStream bisXml = new ByteArrayInputStream(xml.getBytes());
    Document doc = dbf.newDocumentBuilder().parse(bisXml);
    // Create a DOMSignContext and specify the RSA PrivateKey and
// location of the resulting XMLSignature's parent element.
    DOMSignContext dsc = new DOMSignContext(privateKey,
doc.getDocumentElement());
    XMLSignature signature = fac.newXMLSignature(si, ki);
    // Marshal, generate, and sign the enveloped signature.
    //replace provider
    signature.sign(dsc);
    // Output the resulting document.
    ByteArrayOutputStream os = new ByteArrayOutputStream();
    TransformerFactory tf = TransformerFactory.newInstance();
    Transformer trans = tf.newTransformer();
    // remove KeyInfo node
    Node node = doc.getElementsByTagName("KeyInfo").item(0);
    doc.getElementsByTagName("KeyInfo").item(0).getParentNode().removeChild(node);
    trans.transform(new DOMSource(doc), new StreamResult(os));
    os.flush();

```

```
os.close();  
os.toByteArray();  
String sign
```