ČSOB BUSINESSBANKING 24 USER MANUAL

OFX FORMAT DESCRIPTION

Contents

1 OFX

OFX protocol is designed for communication of the client with the bank. It doesn't solve the issue of purchases in virtual department stores, i.e. it doesn't solve the communication within the triplet customer-shop-payment gate (bank).

The client communicates with the gate and sends it his requests, the payment gate responds with answers. The data flowing between the client and the payment gate in the OFX format. The OFX format message can be additionally secured with SSL. We are going to use S/MIME security. The MS Money client uses for example the SSL security. From this point on I will only deal with the OFX format, i.e. the S/MIME format is not described here. The S/MIME format is described for example in http://info.pvt.net/prirucka/smime/smime0.htm .

The MS-Money client will also manage to import OFX format messages. Besides the OFX format the client shall import messages in ABO format. MS-Money imports also files from the competing software Quicken (www.quicken.com).

The communication in the Open Financila Exchange (OFX) protocol resembles the client's communication with the WEB-server. This means that the request or the response start with main headers consisting of headers similar to those used by HTTP protocol (or SMTP or NNTP as the case may be).

The body of the request is written in the OFX language resembling HTML language (OFX is derived from the SGML in the same way as HTML). The body starts with <OFX> tag and ends with </OFX> tag. The header is separated from the body with one empty line, i.e. two CRLF codes. The ends of lines always consist of CRLF codes (x'0d0a').

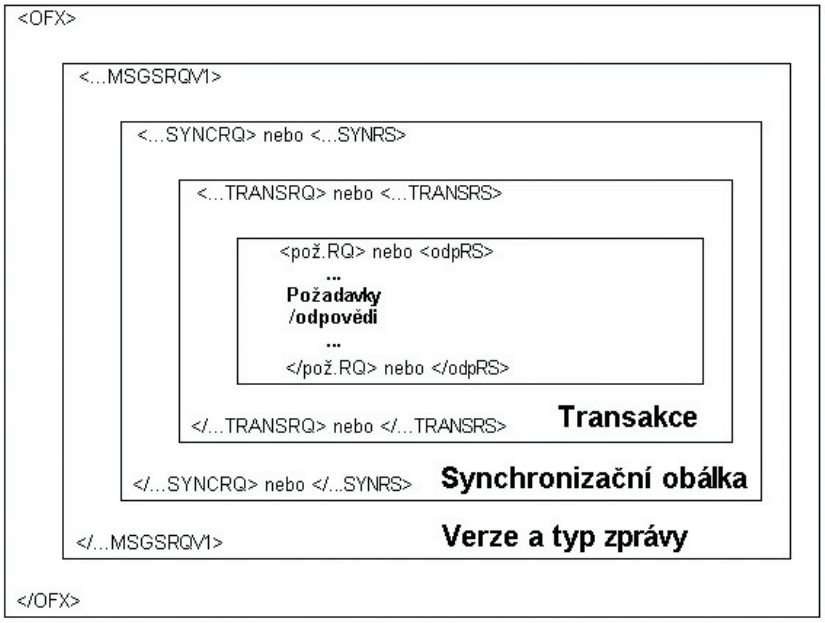
Below please find an example of the OFX version 1.5 protocol standard description (the OFX language doesn't allow any comments; the comments in the angel brackets are purely for the purpose of clarification and studying the code easier. They MUST NOT appear in the data.) :

```
OFXHEADER:100                          <!—Headers -->
DATA:OFXSGML
VERSION:150
SECURITY:TYPE1
ENCODING:USASCII

<OFX>                                  <!-- Begin request data -->
    <SIGNONMSGSRQV1>
        <SONRQ>                        <!-- Begin signon -->
            <DTCLIENT>19961029101000   <!-- Oct. 29, 1996, 10:10:00 am -->
            <USERID>123-45-6789        <!-- User ID (that is, SSN) -->
            <USERPASS>MyPassword       <!-- Password (SSL encrypts whole) -->
            <LANGUAGE>ENG              <!-- Language used for text -->
            <FI>                       <!-- ID of receiving institution -->
                <ORG>NCH               <!-- Name of ID owner -->
                <FID>1001              <!-- Actual ID -->
            </FI>
            <APPID>MyApp
            <APPVER>0500
        </SONRQ>                       <!-- End of signon -->
    </SIGNONMSGSRQV1>
    <BANKMSGSRQV1>
        <STMTTRNRQ>                    <!-- First request in file -->
            <TRNUID>1001
            <STMTRQ>                   <!-- Begin statement request -->
                <BANKACCTFROM>         <!-- Identify the account -->
                    <BANKID>121099999  <!-- Routing transit or other FI ID -->
                    <ACCTID>999988     <!-- Account number -->
                    <ACCTTYPE>CHECKING <!-- Account type -->
                    </BANKACCTFROM>    <!-- End of account ID -->
                <INCTRAN>              <!-- Begin include transaction -->
                    <INCLUDE>Y         <!-- Include transactions -->
                </INCTRAN>             <!-- End of include transaction -->
            </STMTRQ>                  <!-- End of statement request -->
        </STMTTRNRQ>                   <!-- End of first request -->
    </BANKMSGSRQV1>
</OFX>                                 <!-- End of request data -->
```

We are not going to deal with the main header below this point as it is always the same. The only header worth mentioning is SECURITY. This header can hold values NONE (no security on the application layer) or TYPE1, i.e. with SSL, but Microsoft expects SSL only with the server authentication. As we are going to use S/MIME security, we will use the value TYPEx.

General OFX message structure:



The OFX contains one or several messages. Under the term message is understood requirements (or responses) of the same type. For example the client identification, payment orders, interbank transfers etc.

The tag identifying the message contains the MSGS string. The message can contain an embedded synchronisation envelope used for synchronisation of requests sent by the client. The tag for the synchronisation envelope contains the SYNCRQ string in case of the client's request and SYNRS string in case of the payment gate's response.

The synchronisation envelope contains individual transactions. A transaction starts and ends with a tag containing the TRANSRQ string. The payment gate's response starts and ends with a tag containing the TRANSRS string.

inside there are individual instructions or responses. It says that the request/response is enclosed in the transaction envelope.

First we are going to describe the messages without the synchronisation and transaction envelopes.

1.1 CONVENTION

Mandatory items are printed in bold font.

Individual items will have their maximal length specified after the type designation (Numeric – Alphanumeric). The separator used between the type and the length is a hyphen. Example: A-22 specifies an alphanumeric item 22 characters long.

Square brackets [ ] mark an optional part.

1.1.1 Date

The date is always used in the following format:

RRRRMMDDHHMMSS.[SSS gmt offset:tz name]

name = name of the time zone, such as CET.

1.1.2 Structured tag and elementary tag

The tags enclosed in angle brackets are either structures (aggregate) and start with a tag ("left bracket") and end with a bracket with a slash in front of the name of the tag("right bracket"):

<STRUKTURA>

...

</STRUKTURA>

or the tag specifies only one item (element), in which case it doesn't end with a tag with the slash sign (they have no "right bracket"), i.e.

<ITEM>value.

1.1.3 Boolean type tag

There are also tags which mark neither structure nor an item – the only importance is their occurrence. It has values Y (true) or A (false).

1.1.4 Tag versions

A curiosity in case a structure is so called tag "version". Some tags have at the moment two versions, i.e. the end either with the string V1 or V2 (second version). The second versions of the tag may contain newly introduced structure items. In case one wants to use such newly introduced item one has to use the second version tag. I the is important that this version has no direct relation to the OFX protocol version – such as the current OFX version 1.5 contains a tag <BANKMSGSRQVn> in both version 1 and 2.

1.1.5 Status

The response may contain as a part of the transaction envelope a status, i.e. result code. Status is a structure with the following format:

Tag	Version	Description
<STATUS>		Beginning the a structure ("left bracket")
<CODE>		Error code (result code), N-6
<SEVERITY>		Severity of the error: INFO = information WARN = warning ERROR = serious error (the request hasn't been processed)
<MESSAGE>	Only in version 1	Error text description. A-255
<MESSAGE2>	Only in version 2	Error text description (Version 2 allows longer description). A-2000
</STATUS>		End of the structure ("right bracket")

1.1.6 Frequently used elements of the structures

1.1.6.1 <MEMO> and <MEMO2>

These tags describe dom additional information. The <MEMO> format is A-255 and the <MEMO2> format is A-390.

1.1.6.2 <FITID>

The Financial Institution Transaction ID <FITID> is an identifier used in the response generated by the payment gate. With this identifier the responses generated by the payment gate get a unique identifier allowing to distinguish a duplicate response.

<FITID> is not globally unique. The required uniqueness can be reached by chanining of the following items:

Financial Institution Identifier + account number + <FITID>

The <FITID> format is in theory A-255 but in practice A-32 should be enough.

1.1.6.3 Server-Assigned ID <SRVRTID>, <SRVRTID2>

This is an identifier of an object stored in the payment gate. If it is possible, the same value as <FITID> is used.

The server-assigned ID is unique within the financial institution, which means that required uniqueness can be reached by chanining of the following items:

Financial Institution Identifier + <SRVRTID>

1.1.6.4 Client-Assigned Transaction UID <TRNUID>

This means the identifier of the transaction, generated by the client. The client generates this identifier. The payment gate then copies this identifier to the respective response. This identifier thus serves paring the responses with the respective instruction.

The <TRNUID> tag is a core of the message transaction envelope. Each transaction within the concrete client has its unique identifier <TRNUID>.

<TRNUID> has to be unique within the framework of the concrete client. The <TRNUID> format is A-36. The algorithm used for generation of <TRNUID> is the algorithm specified in OSF DCE, i.e. 36 characters long identifier is a result of 128-bit number.

1.1.6.5 Transaction Amount <TRNAMT>

The amount specifies the sum of the money transferred. Its format is A-32, it can contain the decimal point or comma, must not contain any separators of thousands, millions etc. In case the value doesn't contain the decimal point, it is automatically added after the last digit.

The transaction amount may contain a sigh before the first digit from the lefthand side. In case the value doesn't contain the sign, the "plus" sign is added implicitly.

1.1.6.6 Token <TOKEN>, <TOKEN2>

A token serves for synchronisation of the client's requests. The token is used to secure the history of the requests. The token format is A-10 and the <TOKEN2> format is A-36.

The token is unique within the type of the synchronisation request. In case the synchronisation is performed within framework of the account number, the token is unique within the account. In case the synchronisation is performed within the framework of the entire payment gate, the token has to be unique within the framework of the entire payment gate.

1.2 SYNCHRONISATION

Synchronisation is used for securing of the continuity of the work of the client. It may happen that for example in case of a drive failure the client restores the drive image from an older date and is not sure, what transactions were sent and which were not. A similar situation can occure when the client has two computers (such as at home and in the office). He will also work in his office and after coming home he needs to carry on where he started.

The synchronisation is usually performed within the framework of a specific account. When the client wants for example to work with another account, the synchronisation is again required. A synchronisation is possible also within the framework of individual types of commands or on the other hand sfk of the entire payment server.

Messages (commands) coming from the client to the payment gate have their order identified by means of the <TOKEN> tag. Generallyu it doesn't have to be an ascending series of numbers, but such picture perhaps describes the function best.

For example the client sends from his office a message on identification in a format <TOKEN>number followed by a message <TOKEN>číslo+1 etc. When the client comes home from the office, he has to synchronise, i.e. to find out what was the last number. Let say the last number identified is for instance 100 and he sends a message with the number 92, the payment gate has to identify an error – lost synchronisation.

There are 4 synchronisation cases recognised:

1. The client knows that the synchronisation was lost. Therefore he doesn't send any payment order but only a request for synchronisation with the last <TOKEN>number known. The payment gate would return <TOKEN>číslo.
2. The client doesn't know that loss of synchronisation might have occurred, but he is not sure, therefore he wraps his messages in the synchronisation envelope and assignes them numbers <TOKEN>number. The payment gate checks the connection and it increases the series of numbers automatically in its database. When the payment gate discovers an error:
   - Depending on the falue of the <REJECTIFMISSING> tag the payment gate performs the payment order or rejects it.
   - Rturns responses from the number <TOKEN>number sent to the client up to the last number saved in the payment gate.
3. The client sends the messages without the synchronisation envelope. No check is performed. The payment gate increases the <TOKEN>number automatically. We are not going to support this option.
4. The client sends to the payment gate a request with the number = zero, i.e. <TOKEN>0. In case this is the first request of the client, the payment gate confirms it, otherwise it would include the entire history of its responses it keeps in its records.

1.2.1 Tag <REJECTIFMISSING>

The cautions client in the option 2 ("doesn't know that loss of synchronisation might have occurred") is cautions and is afraid not to enter a duplicite order. He can set the Boolean tag <REJECTIFMISSING> in his message. In such cas the payment gate would not process an order with a smaller token in any way.

1.2.2 Tag <TOKENONLY>

In case the client lost the synchronisation and wants to get the last used <TOKEN> without being flooded by the payment gate with the history of its responses, he should use the <TOKENONLY> tag. This tag type is Boolean.

1.2.3 Tag <REFRESH>

The <REFRESH> tag explicitly expresses that the server should return copies of its responses. This tag type is Boolean.

1.2.4 Tag <LOSTSYNC>

This tag type is Boolean. This is used in case the client sends so old <TOKEN>number to the payment gate, that the number is lower than the last stored in the payment gate. In such case the payment gate isn't able to repeat all the responses and marks this situation with the <LOSTSYNC> tag.

1.2.5 Tag <SYNCERROR>

The tag format is N-6; it only occurs in the response and spefifies the error number explaining why the payment gate wasn't able to return the <TOKEN>. In case the payment gate cannot return the token, it returns <TOKEN>-1.

1.2.6 Example of synchronisation from the OFX standard

A client sends a request for synchronisation:

```
<PMTSYNCRQ>
    <TOKEN>123
    <REJECTIFMISSING>N
    <BANKACCTFROM>
        <BANKID>121000248
        <ACCTID>123456789
        <ACCTTYPE>CHECKING
    </BANKACCTFROM>
</PMTSYNCRQ>
```

The server responds:

```
<PMTSYNCRS>
    <TOKEN>125
    <LOSTSYNC>N
    <BANKACCTFROM>
        <BANKID>121000248
        <ACCTID>123456789
        <ACCTTYPE>CHECKING
    </BANKACCTFROM>
    <PMTTRNRS>
        <TRNUID>123
        <STATUS>
            ... status details
        </STATUS>
        <PMTRS>
            ... details on a payment response
        </PMTRS>
    </PMTTRNRS>
    <PMTTRNRS>
        <TRNUID>546
        <STATUS>
            ... status details
        </STATUS>
        <PMTRS>
            ... details on another payment response
        </PMTRS>
    </PMTTRNRS>
</PMTSYNCRS>
```

In this example it is pointed out, that the numbers <TOKEN> don't have to succeed each other – they may form generally any sequence.

2 TRANSACTIONS AND SYNCHRONISATION

When I want to send a payment order, I have to perform a transaction. The transaction has to have its unique identifier, assigned to the transaction vy the client; usually the <TRNUID> is used. In case the payment gate needs to assigne the unique identifier, it would use the <FITID>.

But most often the payment gate neest to identify the objects it keeps in its database. For identification of these objects <SRVRTID> or <SRVRTID2> are used.

The response to a request for a transaction to be performed contains the result of the transaction processin within the structure <STATUS>...</STATUS>.

In other wrods the transaction has a unique identifier, so that when the client for example complains about the given transaction it can be found using the identifier.

Synchronisation is used to prevent duplication of the transaction, which means it prevernts the client to pay twice by mistake. Therefore the order of transactions is monitored (transaction history). In case of a lost response (such as after system BackUp) the payment gate can repeat the lost responses. This is a very important feature for example when vieweing responses on a web server as the web server cannot keep the history.

3 SIGNON MESSAGE TYPE

Using this message the client proves its identity. The message is mandatory – each batch starts with this message. This message uses neither the synchronisation nonor the transaction envelope.

The client's request for authentication SONRQ will be always used when we do not take into account authentication usnig a password. The SONRQ message has the following structure:

```
<SIGNONMSGSRQV1>
        <SONRQ>                          <!-- Signon beginning -->
            <DTCLIENT>19961029101000     <!-- Client's time: Oct. 29, 1996, 10:10:00 -->
            <USERID>123-45-6789          <!-- User ID based on the PS database, A-32 -->
            <USERPASS>                   <!-- Empty password A-171 -->
            <LANGUAGE>CZE                <!-- Language, English clinet: ENG-->
            <APPID>IPB01                 <!-- Our application identifier A-5 -->
            <APPVER>0100                 <!-- Our software version x 100 A-3 -->
        </SONRQ>                         <!-- End of signon -->
</SIGNONMSGSRQV1>
```

The response of the payment gate shall look like this:

```
<SIGNONMSGSRQV1>
        <SONRS>                          <!-- Signon beginning -->
            <STATUS>
                <CODE>kód                <!-- Error code N-6 -->
                <SEVERITY>závažnost      <!-- INFO, WARN, ERROR -->
                <MESSAGE>text chyby      <!-- Text describing the error A-255 -->
            </STATUS>
            <DTSERVER>19961029101000     <!-- Server time: Oct. 29, 1996, 10:10:00 -->
            <LANGUAGE>CZE                <!-- Language -->
        </SONRS>                         <!-- End of signon -->
</SIGNONMSGSRQV1>
```

where the length of the items is given by the type (A or N) followed by a hyphen and the length as the number of characters. A-5 for example is an alphanumeric item with maximum of 5 characters.

Error code in the structure <STATUS>...</STATUS> is as follows:

Value	Severity	Czech error description		English error description, <LANGUAGE>ENG
0	INFO	Úspěch	Success	Success
2000	ERROR	Všeobecná chyba	General error	General error
15000	INFO	Musíš změnit heslo	You have to change your pasword.	Must change USERPASS
15500	ERROR	Požadavek SIGNO byl chybný	SIGNON request inalid.	Signon invalid
15501	ERROR	Zákazníkův přístup je právě používán – tato chybová hláška má smysl pouze v případě, že platební server by neakceptoval atomické operace (což není náš případ)	The account in use – this error description makes sense only when the payment server doesn't accept atomic operations (not our case).	Customer account already in use
15502	ERROR	Zákazníkův přístup je zamknut	User access is locked	USERPASS Lockout
15505	ERROR	Zákazník chce použít nepodporovaný jazyk	The cmr wants to use an unsupported language	Country signon transaction not supported
15506	ERROR	V požadavku byla prázdná (nebyla) zpráva SIGNON	The request contained an empty or no signon message	Empty signon transaction not supported

4 BANK TRANSFERS (BANKING)

First I am going to describe several structures which will appear in individual requests / responses.

The core of the message is an information on the payer's bank account number (BANACCOUNTFROM) and the payee's bank account number (BANKACCOUNTTO). Both structures have very similar formats:

Tag	Version	Description
<BANKACCTFROM>		Beginning of the BANACCOUNTFROM structure
<BANKID>		Bank number, A-9
<BRANCHID>		Abbreviation of the bank's name,A-22
<ACCTID>		Account number, A-22
<ACCTTYPE>		Account type: • CHECKING • SAVINGS • MONEYMRKT (Money Market) • CREDITLINE (Line of Credit) • CMA (Cash management Account)
<ACCTKEY>		Check sum, A-22
</BANKACCTFROM>		End of the structure

As the BANKACCOUNTTO structure has a very similar format, I am not going to write down the description of individual items:

Tag	Version	Description
<BANKACCTTO>		
<BANKID>		
<BRANCHID>		
<ACCTID>		
<ACCTTYPE>		
<ACCTKEY>		
<EXTBANKACCTTO>	V2 only	Further information on the beneficiary's bank
</EXTBANKACCTTO>	V2 only	
</BANKACCTTO>		

The description of the concrete transfer is done using the XFERINFO structure:

Tag	Version	Description
<XFERINFO>		Beginning of the structure
<BANKACCTFROM>		BANKACCOUNTFROM structure (see the previous table)
...		
</BANKACCTFROM>		
<BANKACCTTO>		BANKACCOUNTTO structure (see the previous table))
....		
</BANKACCTTO>		
<TRNAMT>		Transferred amount (this has to be a positive number)
<DTDUE>	V2 only	Date when the order is to be sent to processing
<DTAVAIL>	V2 only	Date when the beneficiary is to receive the amount
<MEMO2>		Attached comment
</XFERINFO>		

The response to the order request (payment) contains particularly the XFERPRCSTC structure, which specifies when and whether at all the individual transfer is to be executed:

Value	Czech error description		English
<XFERPRCSTS>	Beginning of the structure	Beginning of the structure	
<XFERPRCCODE>	Výsledek zpracování:	Processing result:	
	WILLPROCESSON – Bude zpracováno v <DTXFERPRC>	WILLPROCESSON – Will be processed on <DTXFERPRC>	Will be processed on <DTXFERPRC> Posted on <DTXFERPRC>
	POSTEDON – Odesláno do zpracování v <DTXFERPRC>	POSTEDON – Posted to processing on <DTXFERPRC>	Funds not available to make transfer on <DTXFERPRC>
	NOFUNDSON – Platba nebyla dne <DTXFERPRC> kryta	NOFUNDSON – Payment not covered by available funds on <DTXFERPRC>	User canceled payment on <DTXFERPRC>
	CANCELEDON – Požadavek byl zrušen uživatelem dne <DTXFERPRC>	CANCELEDON – The request was cancelled by the user on <DTXFERPRC>	Unable to make transfer
	FAILEDON – Převod (platba) není možný z jiného důvodu.	FAILEDON – Transfer (payment) cannot be performed for unspecified reason.	for unspecified reasons on <DTXFERPRC>
<DTXFERPRC>	Datum převodu – závisí na <XFERPRCCODE>	Transfer date – depends on	
</XFERPRCSTS>	End of the structure	End of the structure	

And now to individual commands.

4.1 GETTING INFORMATION ON THE ACTIVITY ON THE ACCOUNT (STATEMENT) AND THE ACCOUNT BALANCE (BALANCES)

The following tags are used in the OFX protocol for this purpose: <STMTRQ> for the command and <STMTRS> for the response. When you work with a credit card the tags <CCSTMTRQ> and <CCSTMTRS> will be used.

In the batch processing these details will be entered directly in text format and driven through HTTPS of S/MIME protocol to the client without formatting.

This chapter will be extended for the OnLine processing.

4.2 CLOSING

We are not going to implement this feature for the time being.

4.3 STOP CHECK

We are not going to implement this feature for the time being.

4.4 TRANSFER WITHIN THE BANK (INTRABANK FUNDS TRANSFER)

The command is performed using the <INTRARS> structure. The command transaction has to be wrapped in the transaction envelope <INTRATRNRS>...</INTRATRNRS>. But now to the description of the <INTRARS> structure:

Tag	Description
<INTRARQ> <XFERINFO> ... </XFERINFO> </INTRARQ>	Beginning of the structure XFERINFO structure (see the previous text)

The request has to be wrapped in a transaction envelope <INTRATRNRS>...<INTRATRNRS>. The response INTRARS has the following structure:

Tag	Description
<INTRARS>	Beginning of the structure
<CURDEF>	Currency
<SRVRTID>	Server record identifier
<XFERINFO>	XFERINFO structure (see the previous text)
...	
</XFERINFO>	
Only one of the following tags can be used	
<DTXFERPRJ>	Expected transaction date
-or-	
<DTPOSTED>	Current transaction date
<XFERPRCSTS>	Result code structure
...	
</XFERPRCSTS>	
</INTRARS>	

The <STATUS>...</STATUS> structure expressing whether the transaction was accepted or rejected in whole is also a part of the transaction envelope. It is necessary to acknowledge that this is different information then the one contained in the < XFERPRCSTC>...< XFERPRCSTC> structure, which specifies when a single transfer is to be made.

Transaction result code:

Code	Czech description		English description
0	Úspěch	Success	Success (INFO)
2000	Chyba	Error	General error (ERROR)
2002	Chybný účet	Incorrect account	General account error (ERROR)
2006	Účet plátce neexistuje	Source account doesn't exist	Source account not found (ERROR)
2007	Účet plátce uzavřen	Source account closed	Source account closed (ERROR)
2008	Operace na použitém účtu plátce není povolena	The operation on the source account not authorized	Source account not authorized (ERROR)
2009	Příjemcův účet neexistuje	Destination account doesn't exist	Destination account not found (ERROR)
2010	Příjemcův účet uzavřen	Destination account closed	Destination account closed (ERROR)
2011	Operace na použitém účtu příjemce není povolena	The operation on the destination account not authorized	Destination account not authorized (ERROR)
2012	Chyba v sumě	Invalid amount	Invalid amount (ERROR)
2014	Prošle datum	Expired date	Date too soon (ERROR)
2015	Datum je příliš vzdálené	Date too far in future	Date too far in future (ERROR)
2019	Duplikující se požadavek	Duplicate request	Duplicate request (ERROR)
10504	Nepoužitelná měna	Given currency cannot be used	Insufficient funds (ERROR)

An example in OFX standard:

Request:

```
<OFX>                                    <!-- Begin request data -->
    <SIGNONMSGSRQV1>
        <SONRQ>                          <!-- Begin signon -->
            <DTCLIENT>19960828101000     <!-- Aug 28, 1996, 10:10:00 am -->
            <USERID>123-45-6789          <!-- User ID (e.g. SSN) -->
            <USERPASS>MyPassword         <!-- Password (SSL encrypts whole) -->
            <LANGUAGE>ENG                <!-- Language used for text -->
            <FI>                                 <!-- ID of receiving institution -->
                <ORG>NCH                 <!-- Name of ID owner -->
                <FID>1001                <!-- Actual ID -->
            </FI>
            <APPID>MyApp
            <APPVER>0500
        </SONRQ>                         <!-- End of signon -->
    </SIGNONMSGSRQV1>
```

```
    <BANKMSGSRQV1>
        <INTRATRNRQ>                        <!-- First request in file -->
            <TRNUID>1001                    <!-- Client's ID for this request -->
            <INTRARQ>                       <!-- Begin transfer request -->
                <XFERINFO>                  <!-- Begin transfer aggregate -->
                    <BANKACCTFROM>          <!-- Identify the account -->
                        <BANKID>121099999   <!-- Routing transit or other FI ID -->
                        <ACCTID>999988      <!-- Account number -->
                        <ACCTTYPE>CHECKING  <!-- Account type -->
                    </BANKACCTFROM>         <!-- End of account ID -->
                    <BANKACCTTO>            <!-- Identify the account -->
                        <BANKID>121099999   <!-- Routing transit or other FI ID -->
                        <ACCTID>999977      <!-- Account number -->
                        <ACCTTYPE>SAVINGS   <!-- Account type -->
                    </BANKACCTTO>           <!-- End of account ID -->
                    <TRNAMT>200.00          <!-- Amount of transfer -->
                </XFERINFO>                 <!-- End of transfer aggregate -->
            </INTRARQ>                      <!-- End of transfer request -->
        </INTRATRNRQ>                       <!-- End of first request -->
    </BANKMSGSRQV1>
</OFX>                                      <!-- End of request data -->
```

Odpověď:

```
<OFX>                                       <!-- Begin response data -->
    <SIGNONMSGSRSV1>
        <SONRS>                             <!-- Begin signon -->
            <STATUS>                        <!-- Start of status aggregate -->
                <CODE>0                     <!-- OK -->
                <SEVERITY>INFO
            </STATUS>
            <DTSERVER>19960828101003        <!-- Aug 28, 1996, 10:10:03 am -->
            <LANGUAGE>ENG                   <!-- Language used in response -->
            <DTPROFUP>19961029101003        <!-- Last update to profile -->
            <DTACCTUP>19961029101003        <!-- Last account update -->
        </SONRS>                            <!-- End of signon -->
    </SIGNONMSGSRSV1>

 <BANKMSGSRSV1>
        <INTRATRNRS>                        <!-- First response in file -->
            <TRNUID>1001                    <!-- Client ID sent in request -->
            <STATUS>                        <!-- Start status aggregate -->
                <CODE>0                     <!-- OK -->
                <SEVERITY>INFO
            </STATUS>
            <INTRARS>                       <!-- Begin transfer response -->
                <CURDEF>USD
                <SRVRTID>1001               <!-- Server assigned ID -->
                <XFERINFO>                  <!-- Begin transfer aggregate -->
                    <BANKACCTFROM>          <!-- Identify the account -->
                        <BANKID>121099999   <!-- Routing transit or other FI ID -->
                        <ACCTID>999988      <!-- Account number -->
                        <ACCTTYPE>CHECKING  <!-- Account type -->
                    </BANKACCTFROM>         <!-- End of account ID -->
                    <BANKACCTTO>            <!-- Identify the account -->
```

```
                    <BANKID>121099999          <!-- Routing transit or other FI ID -->
                    <ACCTID>999977             <!-- Account number -->
                    <ACCTTYPE>SAVINGS          <!-- Account type -->
                  </BANKACCTTO>                <!-- End of account ID -->
                  <TRNAMT>200.00               <!-- Amount of transfer -->
                </XFERINFO>                    <!-- End of transfer aggregate -->
                <DTXFERPRJ>19960829100000      <!-- Projected posting date -->
              </INTRARS>                       <!-- End of transfer response -->
            </INTRATRNRS>                      <!-- End of first response -->
          </BANKMSGSRSV1>
</OFX>                                         <!-- End of response data -->
```

Unfortunately the XFERPRCSTS is missing to make it perfect. Although it is not in the standard explicitly mentioned, it is a status, so It should (in my opinion) should have syntax as the <STATUS> ... </STATUS> structure.

4.5 CHANGE OF THE TRANSFER REQUEST WITHIN THE BANK (INTRABANK TRANSFER MODIFIKATION)

We are not going to implement this feature for the time being.

4.6 CANCELLING THE REQUEST FOR A TRANSFER WITHIN THE BANK (INTRABANK FUNDS TRANSFER)

Cancelling of such request in the OFX protocol is performed with <INTRACANRQ> tag for the request and with <INTRACANRS> tag for the response.

We are not going to implement this feature for the time being.

4.7 INTERBANK FUNDS TRANSFER

The request and the response are almost identical with the transfer within the bank.

The request has to be wrapped in the transaction envelope <INTERTRNRQ>...<INTERTRNRQ>. The structure of the request is as follows:

Tag	Description
<INTERRQ> <XFERINFO> ... </XFERINFO> </INTRTRQ>	Beginning of the structure XFERINFO structure (see the previous text)

The response has to be wrapped in the transaction envelope <INTERTRNRS>...<INTERTRNRS>. The structure of the INTERRS response is as follows:

Tag	Description
<INTERRS> <CURDEF> <SRVRTID> <XFERINFO> ... </XFERINFO> Only one of the following tags can be used <DTXFERPRJ> -or- <REFNUM> <RECSRVRTID> <DTPOSTED> <XFERPRCSTS> ... </XFERPRCSTS> </INTERRS>	Beginning of the structure Currency Server identifier XFERINFO structure (see the previous text) Expected transaction date String generated by the server, A-32 String generated by the server (Description is particularly for the standing orders) Transaction current date Result code structure

Result codes structure <STATUS>:

Code	Czech description		English description
0	Úspěch	Success	Success (INFO)
2000	Chyba	Error	General error (ERROR)
2002	Chybný účet	Incorrect account	General account error (ERROR)
2006	Účet plátce neexistuje	Source account doesn't exist	Source account not found (ERROR)
2007	Účet plátce uzavřen	Source account closed	Source account closed (ERROR)

Code	Czech description		English description
2008	Operace na použitém účtu plátce není povolena	The operation on the source account not authorized	Source account not authorized (ERROR)
2009	Příjemcův účet neexistuje	Destination account doesn't exist	Destination account not found (ERROR)
2010	Příjemcův účet uzavřen	Destination account closed	Destination account closed (ERROR)
2011	Operace na použitém účtu příjemce není povolena	The operation on the destination account not authorized	Destination account not authorized (ERROR)
2012	Chyba v sumě	Invalid amount	Invalid amount (ERROR)
2014	Prošle datum	Expired date	Date too soon (ERROR)
2015	Datum je příliš vzdálené	Date too far in future	Date too far in future (ERROR)
2019	Duplikující se požadavek	Duplicate request	Duplicate request (ERROR)
10504	Nepoužitelná měna	Given currency cannot be used	Insufficient funds (ERROR)

4.8 CHANGE AND CANCELLING AN INTERBANK TRANSFER

We are not going to implement this feature for the time being.

4.9 INSTANT TRANSFERS (WIRE FUND TRANSFER)

We are not going to implement this feature for the time being.

4.10 RECURRING PAYMENTS (RECURRING FUNDS TRANSFER)

We are not going to implement this feature for the time being.

4.11 E-MAIL NOTIFICATION

We are not going to implement e-mail using the OFX protocol for the time being. We will use the S/MIME protocol for the e-mail messages.

4.12 SYNCHRONISATION

4.12.1 Intrabank Transfer Request

4.12.1.1 <INTRASYNCRQ> command

Tag	Version
<INTRASYNCRQ>	
One of the following tags: <TOKEN>, <TOKEN2>,<TOKENONLY>, <REFRESH>:	
<TOKEN> <TOKEN2> <TOKENONLY> <REFRESH> <REJECTIFMISSING> <BANKACCTFROM> … </BANKACCTFROM> <INTRATRNRQ> … </INTRATRNRQ> </INTRASYNCRQ>	V1 V2

4.12.1.2 <INTRASYNCRS> response

Tag	Version
<INTRASYNCRS> <TOKEN> <TOKEN2> <LOSTSYNC> <SYNCERROR> <BANKACCTFROM> … </BANKACCTFROM> <INTRATRNRS> … </INTRATRNRS> </INTRASYNCRS>	 V1 V2 V2

4.12.2 Interbank Transfers Request

4.12.2.1 <INTERSYNCRQ>

Tag	Version
<INTERSYNCRQ>	
One of the tags <TOKEN>, <TOKEN2>, <TOKENONLY>, <REFRESH>	
<TOKEN>	V1
<TOKEN2>	V2
<TOKENONLY>	
<REFRESH>	
<REJECTIFMISSING>	
<BANKACCTFROM>	
...	
</BANKACCTFROM>	
<INTERTRNRQ>	
</INTERTRNRQ>	
<MULTIINTERTRNRQ>	V2
</MULTIINTERTRNRQ>	
</INTERSYNCRQ>	

4.12.2.2 Response <INTERSYNCRS>

Tag	Version
<INTERSYNCRS>	
<TOKEN>	V1
<TOKEN2>	V2
<LOSTSYNC>	
<SYNCERROR>	V2
<BANKACCTFROM>	
</BANKACCTFROM>	
<INTERTRNRS>	
</INTERTRNRS>	
<MULTIINTERTRNRS>	V2
</MULTIINTERTRNRS>	
</INTERSYNCRS>	

4.13 OVERALL FRAMEWORK OF THE MESSAGE AND RESPONSE

The overall framework of the message for a bank funds transfer contains not only individual transfer commands, but also their synchronisation and transaction envelopes. The <BANKMSGSET>...</BANKMSGSET> structure contains messages Version 1 (BANKMSGSETV1) or Version 2 (BANKMSGSETV2). A message then consists of the Version 1 or 2 requests (BANKMSGSRQV1 or BANKMSGSRQV2) or responses (BANKMSGSRSV1 or 2).

As this text doesn't cover the description of the entire OFX standard the messages not described here are in italics.

4.13.1 Bank Message Set Request Messages

Message Set	Message
<BANKMSGSET>	
<BANKMSGSETV1>	
<BANKMSGSRQV1>	STMTTRNRQ_
	STMTRQ_
	STMTENDTRNRQ_
	STMTENDRQ_
	STPCHKTRNRQ_
	STPCHKRQ_
	INTRATRNRQ_
	INTRARQ_
	INTRAMODRQ_
	INTRACANRQ_
	RECINTRATRNRQ_
	RECINTRARQ_
	RECINTRAMODRQ_
	RECINTRACANRQ_
	BANKMAILTRNRQ_
	BANKMAILRQ_
	STPCHKSYNCRQ_
	INTRASYNCRQ_
	RECINTRASYNCRQ_
	BANKMAILSYNCRQ_
</BANKMSGSRQV1>	
</BANKMSGSETV1>	
<BANKMSGSETV2>	
<BANKMSGSRQV2>	STMTTRNRQ_
	STMTRQ_
	STMTENDTRNRQ_
	STMTENDRQ_
	STPCHKTRNRQ_
	STPCHKRQ_
	INTRATRNRQ_
	INTRARQ_
	INTRAMODRQ_
	INTRACANRQ_
	RECINTRATRNRQ_
	RECINTRARQ_
	RECINTRAMODRQ_
	RECINTRACANRQ_
	BANKMAILTRNRQ_
	BANKMAILRQ_
	STPCHKSYNCRQ_
	INTRASYNCRQ_
	RECINTRASYNCRQ_
	BANKMAILSYNCRQ_
</BANKMSGSRQV2>	
</BANKMSGSETV2>	
</BANKMSGSET>	

4.13.2 Bank Message Set Response Messages

Message Set	Message
<BANKMSGSET>	
<BANKMSGSETV1>	
<BANKMSGSRSV1>	STMTTRNRS
	STMTRS
	STMTENDTRNRS
	STMTENDRS
	STPCHKTRNRS
	STPCHKRS
	INTRATRNRS
	INTRARS
	INTRAMODRS
	INTRACANRS
	RECINTRATRNRS
	RECINTRARS
	RECINTRAMODRS
	RECINTRACANRS
	BANKMAILTRNRS
	BANKMAILRS
	CHKMAILRS
	DEPMAILRS
	STPCHKSYNCRS
	INTRASYNCRS
	RECINTRASYNCRS
	BANKMAILSYNCRS
</BANKMSGSRSV1>	
</BANKMSGSETV1>	
<BANKMSGSETV2>	
<BANKMSGSRSV2>	STMTTRNRS
	STMTRS
	STMTENDTRNRS
	STMTENDRS
	STPCHKTRNRS
	STPCHKRS
	INTRATRNRS
	INTRARS
	INTRAMODRS
	INTRACANRS
	RECINTRATRNRS
	RECINTRARS
	RECINTRAMODRS
	RECINTRACANRS
	BANKMAILTRNRS
	BANKMAILRS
	CHKMAILRS
	DEPMAILRS
	STPCHKSYNCRS
	INTRASYNCRS
	RECINTRASYNCRS
	BANKMAILSYNCRS
</BANKMSGSRSV2>	
</BANKMSGSETV2>	
</BANKMSGSET>	

5 PAYMENTS

The main problem of payments is the question how to address the beneficiary. When a payment is addressed to a mail address, there is no substantial problem too. The problem arises when the payment is addressed to a bank account. Constant, variable and specific symbols are used in the Czech Republic as additional information for such payment. These symbols have no alternative both abroad and in the OFX protocol, therefore it is necessary to find OFX protocol items, where these symbols can be entered.

The payments are addressed:

- To the payee to his mail address <PAYEE>. It can be used only for the first payment to the address. During the first payment the payment gate remembers the address and returns such address identifier <PAYEEID>.
- The next payment to the mail address is addressed to the <PAYEEID> identifier.
- A payment to a group address – so called payment list. The group address points to the list of addresses <PAYEELSTID>. The list of addresses contains a list of individual payees.
- A payment to a bank account. The beneficiary's bank account is identified with the <BANKACCTTO>...</BANKACCTTO> structure described in the chapter on bank transfers. (The only acceptable payee for us).

5.1 THE <PMTRQ> PAYMENT ORDER AND THE <PMTRS> RESPONSE

The <PMTRQ> payment order has to be enclosed in a transaction envelope <PMTTRNRQ>...</PMTTRNRQ>. The payment order consists either of the <PMTINFO> structure or <PMTINFO2> structure:

Tag	Version
<PMTRQ>	
<PMTINFO>	V1
...	
</PMTINFO>	
<PMTINFO2>	V2
...	
</PMTINFO2>	
</PMTRQ>	

The <PMTRS> response also has to be enclosed in the transaction envelope, i.e. in the <PMTTRNRS>...<PMTTRNRS> structure:

Tag	Version	Description
<PMTRS>		
<SRVRTID>	V1	ID assigned to the payment order by the payment gate
<SRVRTID2>	V2	ID assigned to the payment order by the payment gate
<PAYEELSTID>		The payment list identifier assigned by the payment gate.
<CURDEF>		Default currency for recurring payments
<PMTINFO>	V1	
...		
</PMTINFO>		
<PMTINFO2>	V2	
...		
</PMTINFO2>		
<EXTDPAYEE>		
...		
</EXTDPAYEE>		
<CHECKNUM>		Check sum (optional), A-12
<PMTPRCSTS>		Result status
...		
</PMTPRCSTS>		
<RECSRVRTID>	V1	Reference, that the payment was generated as a recurring payment
<RECSRVRTID2>	V2	Reference, that the payment was generated as a recurring payment
</PMTRS>		

The result status described in the previous structure consists of the <PMTPRCSTS>... </PMTPRCSTS> structure described below. The result status says when and whether at all the payment is going to be processed.

However the response is wrapped in the transaction envelope. The <STATUS>...</STATUS> structure containing the status code informing whether the transaction was not rejected in whole is a part of the transaction envelope.

Status codes:

Value	Meaning	Description
0	Success (INFO)	Success
2000	General error (ERROR)	General error

Value	Meaning	Description
2002	General account error (ERROR)	Account error
2006	Source account not found (ERROR)	Source account doesn't exist
2007	Source account closed (ERROR)	Source account closed
2008	Source account not authorized (ERROR)	Unauthorised access to the source account
2009	Destination account not found (ERROR)	Destination account doesn't exist
2010	Destination account closed (ERROR)	Destination account closed
2011	Destination account not authorized (ERROR)	Destination access to the source account
2012	Invalid amount (ERROR)	Invalid amount
2014	Date too soon (ERROR)	Expired date
2015	Date too far in future (ERROR)	Datum je příliš vzdálené
2019	Duplicate request (ERROR)	Date too far in future
10501	Invalid payee (ERROR)	Invalid payee
10502	Invalid payee address (ERROR)	Invalid payee address
10503	Invalid payee account number (ERROR)	Invalid payee account number
10510	Invalid payee ID (ERROR)	Invalid payee ID
10511	Invalid payee city (ERROR)	Invalid city in the payee's address
10512	Invalid payee state (ERROR)	Invalid country in the payee's address
10513	Invalid payee postal code (ERROR)	Invalid postal code
10517	Invalid payee name (ERROR)	Invalid payee name
10519	Invalid payee list ID (ERROR)	Invalid payee list ID

5.2 PAYMENT INFORMATION STRUCTURE <PMTINFO>, <PMTINFO2>

The information on the payment order itself is stored in the <PMTINFO> structure or in richer <PMTINFO2> structure.

Tag	Version	Description
<PMTINFO>	V1	
<BANKACCTFROM>		
...		
</BANKACCTFROM>		
<TRNAMT>		The amount, it has to be positive
It is possible to use either <PAYEEID> or <PAYEE>.		In Version 2, PAYEE can be left out
<PAYEEID>		See below. The identification of the pay with his address. It is a SRVRTID type item, i.e. its value is assigned by the payment gate
<PAYEE>		See below. Either <PAYEEID> or <PAYEE> can be used, but not both of them.
</PAYEE>		
<PAYEELSTID>		Payee list identifier, see below, SRVRTID
<BANKACCTTO>		Payee account identifier (<PAYEE> is also required as additional information). This tag is described in detail in the chapter on bank transfers.
</BANKACCTTO>		
<EXTDPMT>		A structure allowing to add additional information such as invoice. See below.
</EXTDPMT>		
<PAYACCT>		Payer account number, A-32
<DTDUE>		Due date, date format.
<MEMO>		Additional information sent by the payer to the payee, memo
<BILLREFINFO>		The payer's additional information, it can be structured. It is probably possible to use it for variable and specific symbol.
		A-80
</PMTINFO>		

Due to our national specifics perhaps we may have to use the <PMTINFO2> structure:

Tag	Version	Description
<PMTINFO2>	V2	
<PMTTYPE>		Payment type, we will probably use it for constant symbol
<BANKACCTFROM>		Payer account identifier, see the chapter on bank accounts.
...		
</BANKACCTFROM>		
<TRNAMT>		The amount, it has to be positive
It is possible to use either		In Version 2, PAYEE can be left out
<PAYEEID2> or <PAYEE2>.		
<PAYEEID2>		See below
<PAYEE2>	V2	See below
...		
</PAYEE2>		
<PAYEELSTID2>		Payee list identifier, SRVRTID2
<BANKACCTTO>		Payee account identifier (<PAYEE> is also required as additional information)
...		
</BANKACCTTO>		
<EXTDPMT>		See below
...		
</EXTDPMT>		
<PAYACCT>	V2	Payee account number, A-32 Required only in the version 1. In the version 2 it is used only in case this information is not included in the SIGNON message.
<DTDUE>	V2	Due date, date when the payment should be credited to the payee
<DTAVAIL>		Due date, but I don't understand exactly what for should this optional field be. The difference in the standard is in the verbs DUE and AVAILABLE.
<PMTFOR>		Name of the person on whose order the payment was released, in case this tag is not used the name is taken from <BANKACCTFROM>, A-32 This tag is currently used only for <COUNTRY> = ITA.
<BOOKINGTEXT>		A text describing the payment for the payer's bank, A-40 This tag is at the moment used only for <COUNTRY> = CHE.
<MEMO2>		Additional information sent by the payer to the payee, memo2
<BILLREFINFO>		The payer's additional information, it can be structured. It is probably possible to use it for variable and specific symbol. A-80
</PMTINFO2>		

5.2.1 Payee <PAYEE>, <PAYEE2>

The <PAYEE> tag specifies the payment beneficiary (payee) address.

Tag	Version	Description
<PAYEE>	V1	
<NAME>		Name. A-32
<ADDR1>		1st line of the address, A-32
<ADDR2>		2nd line of the address, A-32
<ADDR3>		3rd line of the address, A-32
<CITY>		City, A-32
<STATE>		State, country (higher administrative territorial part), A-5
<POSTALCODE>		Postal code, A-11
<COUNTRY>		3-character country code pursuant to ISO/DIS-3166, A-3
<PHONE>		Phone number, A-32
</PAYEE>		

Tag	Version	Description
<PAYEE2>	V2	
<NAME>		A-32
<ADDR1>		A-32
<ADDR2>		A-32
<ADDR3>		A-32
<CITY>		A-32
<STATE>		A-5
<POSTALCODE>		A-11
<COUNTRY>		A-3
<PHONE>		A-32
</PAYEE2>		

5.2.2 Extended Payment <EXTDPMT>

The <EXTDPMT>...</EXTDPMT> structure allows particularly to specify particularly invoices paid by the payment. It also allows to specify individual items of each invoice.

Tag	Version	Description
<EXTDPMT>		
<EXTDPMTFOR>		Payment type: INDIVIDUAL or BUSINESS
<EXTDPMTCHK>	V1	Number of the check used for this payment, N-10
<EXTDPMTCHK2>	V2	Number of the check used for this payment, A-12
The payment description contains either <EXTDPMTDSC> tag or <EXTDPMTINV> structure.		
<EXTDPMTDSC>		Free text, A-255
<EXTDPMTINV>		Invoice
<INVOICE>		Beginning of a description of one invoice. One command can contain several invoices i.e. several <INVOICE>..</INVOICE> structures
<INVNO>		Invoice number (variable symbol), A-32
<INVTOTALAMT>		Total amount, amount format It should be a positive number.
<INVPAIDAMT>		The amount to be paid, amount It should be a positive number.
<INVDATE>		Invoice issue date, datetime format
<INVDESC>		Invoice description, A-80
<DISCOUNT>	V1	A structure describing a discount, one invoice can have only one discount.
<DSCRATE>		Percent discount It is possible to use either the <DSCRATE> or <DSCAMT> tag, not both of them.
<DSCAMT>		Discount amount
<DSCDATE>		Date of granting the discount
<DSCDESC>		Discount description, A-80
</DISCOUNT>		
<DISCOUNT2>	V2	A version 2 structure describing a discount, one invoice can have only one discount.
<DSCRATE>	V2	Percent discount It is possible to use either the <DSCRATE> or <DSCAMT> tag, not both of them.
<DSCAMT>	V2	Discount amount
<DSCDATE>		Date of granting the discount
<DSCDESC>		Discount description, A-80
</DISCOUNT2>		
<ADJUSTMENT>		Invoice approval, only one per invoice
<ADJNO>		Approval number, A-32
<ADJDESC>		Approval description, A-80
<ADJAMT>		Approved amount In case this amount has a sign, it specifies how much more or less is approved.
<ADJDATE>		Approval date
</ADJUSTMENT>		
<LINEITEM>		Structure describing individual invoice items
<LITMCODE>	V2	Posting Code, The description of this optional tag isn't clear to me, A-32
<LITMAMT>		Amount
<LITMDESC>		Item description, A-80
</LINEITEM>		
</INVOICE>		
</EXTDPMTINV>		
</EXTDPMT>		

5.2.3 Extended Payee <EXTDPAYEE>

The payment gate returns this structure to the client in the case the client addresses the payments to a postal address. The postal address is stored in the payment gate and the client then uses in the next payments only this returned identifier. This address identifier is of the SRVRTID or SRVRTID2 type.

The validity of this identifier can be global from the point of the payment gate or only within the concrete user of the payment gate. The validity can be limited to certain time.

Tag	Version	Description
<EXTDPAYEE>	V1	
<PAYEEID>	V2	The address identifier assigned by the payment gate, SRVRTID
<PAYEEID2>		The address identifier assigned by the payment gate, SRVRTID2
<IDSCOPE>		The identifier is GLOBAL = applicable within the entire payment gate USER = applicable only for the concrete user of the payment gate
<NAME>		Payee's name, A-32
<DAYSTOPAY>	V2	Minimal period of validity specified in business days, N-3
</EXTDPAYEE>		

5.2.4 Payment Processing Status <PMTPRCSTS>

Error status code in the response:

Tag	Description
<PMTPRCSTS>	
<PMTPRCCODE>	Error code
<DTPMTPRC>	Date the description of which depends on the error code.
</PMTPRCSTS>	

Error code:

Value	Czech error description		English
WILLPROCESSON	Bude zpracováno v <DTPMTPRC>	Will be processed on <DTPMTPRC>	Will be processed on <DTPMTPRC>
PROCESSEDON	Bylo zpracováno v <DTPMTPRC>	Was processed for payment on <DTPMTPRC>	Was processed for payment on <DTPMTPRC>
NOFUNDSON	V <DTPMTPRC> nebude platba kryta	Funds will not be available to make payment on <DTPMTPRC>	Funds not available to make payment on <DTPMTPRC>
FAILEDON	Platbu v <DTPMTPRC> není možno provést z nespecifikovaných důvodů	Unable to make payment on <DTPMTPRC> for unspecified reasons	Unable to make payment for unspecified reasons on <DTPMTPRC>
CANCELEDON	Klient zrušil platbu v <DTPMTPRC>	User cancelled payment on <DTPMTPRC>	User cancelled payment on <DTPMTPRC>

5.3 RECURRENT PAYMENTS

We are not going to implement this feature for the time being.

5.4 PAYEE LISTS

We are not going to implement this feature for the time being.

5.5 SYNCHRONISATION

The request for synchronisation of payment orders is done using the <PMTSYNCRQ>...</PMTSYNCRQ> structure. The principle is completely similar as in the case of the bank transfer synchronisation (it also contains more detailed description). Again it is possible to choose only one of the options <TOKEN>, <TOKEN2>, <TOKENONLY> or <REFRESH>:

Tag	Version
<PMTSYNCRQ>	
<TOKEN>	V1
<TOKEN2>	V2
<TOKENONLY>	
<REFRESH>	
<REJECTIFMISSING>	
<BANKACCTFROM>	
</BANKACCTFROM>	
<PMTTRNRQ>	
</PMTTRNRQ>	
</PMTSYNCRQ>	

The <PMTSYNCRS> response follows:

Tag	Version	Description
<PMTSYNCRS>		
<TOKEN>	V1	
<TOKEN2>	V2	
<LOSTSYNC>		
<SYNCERROR>	V2	Result code, N-6
<BANKACCTFROM>		
</BANKACCTFROM>		
<PMTTRNRS>		
</PMTTRNRS>		
</PMTSYNCRS>		

Result codes:

Code	Meaning
0	Success (INFO)
2000	General error (ERROR)
2002	General account error (ERROR)
2003	Account not found (ERROR)
2004	Account closed (ERROR)
2005	Account not authorized (ERROR)

5.6 OVERALL MESSAGE FRAMEWORK

Request framework:

Message Set	Message
<BILLPAYMSGSET> <BILLPAYMSGSETV1> <BILLPAYMSGSRQV1>	PMTTRNRQ_ PMTRQ_ PMTMODRQ_ PMTCANCRQ_ RECPMTTRNRQ_ RECPMTRQ_ RECPMTMODRQ_ RECPMTCANCRQ_ PAYEETRNRQ_ PAYEERQ_ PAYEEMODRQ_ PAYEEDELRQ_ PMTINQTRNRQ_ PMTINQRQ_ PMTMAILTRNRQ_ PMTMAILRQ_ PMTSYNCRQ_ RECPMTSYNCRQ_ PAYEESYNCRQ_ PMTMAILSYNCRQ_
</BILLPAYMSGSRQV1> </BILLPAYMSGSETV1> <BILLPAYMSGSETV2>	
<BILLPAYMSGSRQV2>	PMTTRNRQ_ PMTRQ_ PMTMODRQ_ PMTCANCRQ_ RECPMTTRNRQ_ RECPMTRQ_ RECPMTMODRQ_ RECPMTCANCRQ_ PAYEETRNRQ_ PAYEERQ_ PAYEEMODRQ_ PAYEEDELRQ_ PMTINQTRNRQ_ PMTINQRQ_ PMTMAILTRNRQ_ PMTMAILRQ_ PMTSYNCRQ_ RECPMTSYNCRQ_ PAYEESYNCRQ_ PMTMAILSYNCRQ_
</BILLPAYMSGSRQV2> </BILLPAYMSGSETV2> </BILLPAYMSGSET>	

Response framework:

Message Set	Message
<BILLPAYMSGSET>	
<BILLPAYMSGSETV1>	
<BILLPAYMSGSRSV1>	PMTTRNRS
	PMTRS
	PMTMODRS
	PMTCANCRS
	RECPMTTRNRS
	RECPMTRS
	RECPMTMODRS
	RECPMTCANCRS
	PAYEETRNRS
	PAYEERS
	PAYEEMODRS
	PAYEEDELRS
	PMTINQTRNRS
	PMTINQRS
	PMTMAILTRNRS
	PMTMAILRS
	PMTSYNCRS
	RECPMTSYNCRS
	PAYEESYNCRS
	PMTMAILSYNCRS
</BILLPAYMSGSRSV1>	
</BILLPAYMSGSETV1>	
<BILLPAYMSGSETV2>	
<BILLPAYMSGSRSV2>	PMTTRNRS
	PMTRS
	PMTMODRS
	PMTCANCRS
	RECPMTTRNRS
	RECPMTRS
	RECPMTMODRS
	RECPMTCANCRS
	PAYEETRNRS
	PAYEERS
	PAYEEMODRS
	PAYEEDELRS
	PMTINQTRNRS
	PMTINQRS
	PMTMAILTRNRS
	PMTMAILRS
	PMTSYNCRS
	RECPMTSYNCRS
	PAYEESYNCRS
	PMTMAILSYNCRS
</BILLPAYMSGSRSV2>	
</BILLPAYMSGSETV2>	
</BILLPAYMSGSET>	

5.7 AN EXAMPLE OF A PLANNED PAYMENT FROM THE OFX STANDARD

Create a payment to "J.C. Counts" for $123.45 to be paid on September 1,1997 using funds in a checking account:

```
<!-- payment example 1 -->
<OFX>
    <SIGNONMSGSRQV1>
        <SONRQ>
            <DTCLIENT>19961029101000
            <USERID>123-45-6789
            <USERPASS>MyPassword
            <LANGUAGE>ENG
            <FI>
                <ORG>NCH
                <FID>12321
```

```
            </FI>
            <APPID>MyApp
            <APPVER>0700
        </SONRQ>
    </SIGNONMSGSRQV1>
    <BILLPAYMSGSRQV1>
        <PMTTRNRQ>
            <TRNUID>1001
            <PMTRQ>
                <PMTINFO>
                    <BANKACCTFROM>
                        <BANKID>123432123
                        <ACCTID>516273
                        <ACCTTYPE>CHECKING
                    </BANKACCTFROM>
                    <TRNAMT>123.45
                    <PAYEE>
                        <NAME>J. C. Counts
                        <ADDR1>100 Main St.
                        <CITY>Turlock
                        <STATE>CA
                        <POSTALCODE>90101
                        <PHONE>415.987.6543
                    </PAYEE>
                    <PAYACCT>10101
                    <DTDUE>19971001
                    <MEMO>payment #3
                </PMTINFO>
            </PMTRQ>
        </PMTTRNRQ>
    </BILLPAYMSGSRQV1>
</OFX>
```

The server responds indicating that it will make the payment on the date requested and that the payee is a standard payee:

```
<OFX>
    <SIGNONMSGSRSV1>
        <SONRS>
            <STATUS>
                <CODE>0
                <SEVERITY>INFO
            </STATUS>
            <DTSERVER>19961029101003
            <LANGUAGE>ENG
            <DTPROFUP>19961029101003
            <DTACCTUP>19961029101003
        </SONRS>
    </SIGNONMSGSRSV1>
    <BILLPAYMSGSRSV1>
        <PMTTRNRS>
            <TRNUID>1001
            <STATUS>
                <CODE>0
                <SEVERITY>INFO
            </STATUS>
```

```
            <PMTRS>
                <SRVRTID>1030155
                <PAYEELSTID>123214
                <CURDEF>USD
                <PMTINFO>
                    <BANKACCTFROM>
                        <BANKID>123432123
                        <ACCTID>516273
                        <ACCTTYPE>CHECKING
                    </BANKACCTFROM>
                    <TRNAMT>123.45
                    <PAYEE>
                        <NAME>J. C. Counts
                        <ADDR1>100 Main St.
                        <CITY>Turlock
                        <STATE>CA
                        <POSTALCODE>90101
                        <PHONE>415.987.6543
                    </PAYEE>
                    <PAYACCT>10101
                    <DTDUE>19971001
                    <MEMO>payment #3
                </PMTINFO>
                <EXTDPAYEE>
                    <PAYEEID>9076
                    <IDSCOPE>USER
                    <NAME>J. C. Counts
                    <DAYSTOPAY>3
                </EXTDPAYEE>
                <CHECKNUM>20111
        <PMTPRCSTS>
         <PMTPRCCODE>WILLPROCESSON
         <DTPMTPRC>19971001
        </PMTPRCSTS>
            </PMTRS>
         </PMTTRNRS>
    </BILLPAYMSGSRSV1>
</OFX>
```

Create a second payment to the payee, using the payee ID returned in the previous example:

```
<!-- payment example 2 -->
<OFX>
    <SIGNONMSGSRQV1>
        <SONRQ>
            <DTCLIENT>19961029101000
            <USERID>123-45-6789
            <USERPASS>MyPassword
            <LANGUAGE>ENG
            <FI>
                <ORG>NCH
                <FID>12321
            </FI>
            <APPID>MyApp
            <APPVER>0700
        </SONRQ>
```

```
    </SIGNONMSGSRQV1>
    <BILLPAYMSGSRQV1>
        <PMTTRNRQ>
            <TRNUID>1001
            <PMTRQ>
                <PMTINFO>
                    <BANKACCTFROM>
                        <BANKID>123432123
                        <ACCTID>516273
                        <ACCTTYPE>CHECKING
                    </BANKACCTFROM>
                    <TRNAMT>123.45
                    <PAYEEID>9076
                    <PAYACCT>10101
                    <DTDUE>19971101
                    <MEMO>Payment #4
                </PMTINFO>
            </PMTRQ>
        </PMTTRNRQ>
    </BILLPAYMSGSRQV1>
</OFX>
```

The server responds indicating that it will make the payment on the date requested:

```
<OFX>
    <SIGNONMSGSRSV1>
        <SONRS>
            <STATUS>
                <CODE>0
                <SEVERITY>INFO
            </STATUS>
            <DTSERVER>19961029101003
            <LANGUAGE>ENG
            <DTPROFUP>19961029101003
            <DTACCTUP>19961029101003
        </SONRS>
    </SIGNONMSGSRSV1>
    <BILLPAYMSGSRSV1>
        <PMTTRNRS>
            <TRNUID>1001
            <STATUS>
                <CODE>0
                <SEVERITY>INFO
            </STATUS>
            <PMTRS>
                <SRVRTID>1068405
                <PAYEELSTID>123432
                <CURDEF>USD
                <PMTINFO>
                    <BANKACCTFROM>
                        <BANKID>123432123
                        <ACCTID>516273
                        <ACCTTYPE>CHECKING
                    </BANKACCTFROM>
                    <TRNAMT>123.45
```

```
                    <PAYEEID>9076
                    <PAYACCT>10101
                    <DTDUE>19971101
                    <MEMO>payment #4
                </PMTINFO>
                <EXTDPAYEE>
                    <PAYEEID>9076
                    <IDSCOPE>USER
                    <NAME>J. C. Counts
                    <DAYSTOPAY>3
                </EXTDPAYEE>
        <PMTPRCSTS>
          <PMTPRCCODE>WILLPROCESSON
          <DTPMTPRC>19971101
        </PMTPRCSTS>
            </PMTRS>
         </PMTTRNRS>
    </BILLPAYMSGSRSV1>
</OFX>
```

6 PORTFOLIO (INVESTMENTS)

We are not going to implement this feature for the time being.

7 BILL PRESENTMENT

We are not going to implement this feature for the time being.