Open Financial Exchange Specification 1.0.2

May 30, 1997

Chapter 11

CONTENT

11Banking

Open Financial Exchange enables financial institution (FI) customers to keep their finances up-to-date and to manage their bank accounts conveniently in several ways. Customers can download transactions and update account balances on a daily basis. They can retrieve a closing statement that contains the same information that they are accustomed to seeing on a paper statement. They can transfer funds between accounts at a financial institution, either immediately upon going online or on a regular schedule. Customers can schedule transfers between accounts on a recurring basis and can transfer funds between accounts at different financial institutions. If necessary, customers can request a wire funds transfer. Open Financial Exchange also enables requests to stop payment on pending checks.

Using customer notification, an FI can notify customers of important events regarding their accounts, such as returned checks or deposits.

11.1Consumer and Business Banking

Open Financial Exchange supports banking for both consumers and businesses. Some customers might use some areas more heavily within Open Financial Exchange Banking (such as credit card download); other areas might be more appropriate for businesses (such as wire transfers). Yet all of the functionality defined for Banking is appropriate to some extent for both consumer and business applications.

11.2Credit Card Data

Credit card data is available to Open Financial Exchange clients through the statement download facility. Statement download provides a way to download credit card transaction data and balances on an as-needed basis. Statement closing information can be made available to clients as well.

11.3Common Banking Aggregates

This section describes several aggregates used throughout the Banking portion of Open Financial Exchange.

11.3.1Banking Account <BANKACCTFROM>

Open Financial Exchange uses the Banking Account aggregate to identify an account at an FI. The aggregate contains enough information to uniquely identify an account for the purposes of statement

download, bill payment, and funds transfer. <CCACCTFROM> identifies credit card accounts; see section 11.3.2.

Tag	Description
<BANKACCTFROM>	Bank-account-from aggregate
<BANKID>	Routing and transit number, *A-9*
<BRANCHID>	Bank identifier for international banks, *A-22*
<ACCTID>	Account number, *A-22*
<ACCTTYPE>	Type of account, see section 11.3.1.1
<ACCTKEY>	Checksum for international banks, *A-22*
</BANKACCTFROM>	

The <BANKACCTTO> aggregate contains the same elements.

11.3.1.1Account Types for <ACCTTYPE> Element

Type	Description
CHECKING	Checking
SAVINGS	Savings
MONEYMRKT	Money Market
CREDITLINE	Line of credit

11.3.2Credit Card Account <CCACCTFROM>

Open Financial Exchange uses the Credit Card Account aggregate to identify a credit card account at an FI. The aggregate contains enough information to uniquely identify an account for the purposes of statement downloads and funds transfer.

Tag	Description
<CCACCTFROM>	Credit-card-account-from aggregate
<ACCTID>	Account number, *A-22*
<ACCTKEY>	Checksum for international banks, *A-22*
</CCACCTFROM>	

The <CCACCTTO> aggregate contains the same elements.

11.3.3 Bank Account Information <BANKACCTINFO>

Open Financial Exchange uses the bank account information aggregate to download account information from an FI. It includes account number specification in <BANKACCTFROM> as well as the status of the service.

Tag	Description
<BANKACCTINFO>	Bank-account-information aggregate
<BANKACCTFROM>	Bank-account-from aggregate
</BANKACCTFROM>	
<SUPTXDL>	Y if account supports transaction detail downloads, N if it is balance-only, *Boolean*
<XFERSRC>	Y if account is enabled as a source for an intrabank or interbank transfer, *Boolean*
<XFERDEST>	Y if account is enabled as a destination for an intrabank or interbank transfer, *Boolean*
<SVCSTATUS>	Status of the account
	AVAIL = Available, but not yet requested
	PEND = Requested, but not yet available
	ACTIVE = In use
</BANKACCTINFO>	

11.3.4 Credit Card Account Information <CCACCTINFO>

Open Financial Exchange uses the credit card account information aggregate to download account information from an FI. It includes credit card number specification in <CCACCTFROM> as well as the status of the service.

Tag	Description
<CCACCTINFO>	Credit-card-account-information aggregate
<CCACCTFROM>	Credit-card-account-from aggregate
</CCACCTFROM>	
<SUPTXDL>	Y if account supports transaction detail downloads, N if it is balance-only, *Boolean*
<XFERSRC>	Y if account is enabled as a source for an intrabank or interbank transfer, *Boolean*
<XFERDEST>	Y if account is enabled as a destination for an intrabank or interbank transfer, *Boolean*
<SVCSTATUS>	Status of the account
	AVAIL = Available, but not yet requested
	PEND = Requested, but not yet available
	ACTIVE = In use
</CCACCTINFO>	

11.3.5 Transfer Information <XFERINFO>

Many of the transfer requests and responses use an <XFERINFO> aggregate. This aggregate identifies accounts that are part of the transfer, amount of money to be transferred, and the date of the transfer.

Tag	Description
<XFERINFO>	Transfer-information aggregate
Account-from options. Choose either <BANKACCTFROM> or <CCACCTFROM>.	
<BANKACCTFROM>	Account-from aggregate, see section 11.3.1
</BANKACCTFROM>	
-or-	
<CCACCTFROM>	Credit-card-account-from aggregate, see section 11.3.2
</CCACCTFROM>	
Account-to options. Choose either <BANKACCTTO> or <CCACCTTO>.	
<BANKACCTTO>	Account-to aggregate, see section 11.3.1
</BANKACCTTO>	
-or-	
<CCACCTTO>	Credit-card-account-to aggregate, see section 11.3.2
</CCACCTTO>	
<TRNAMT>	Amount of the transfer, *amount*
<DTDUE>	Date that the transfer is to be sent. If the client does not specify <DTDUE>, the transfer occurs as soon as possible. <DTDUE> is required for scheduled or repeating transfers, *datetime*
</XFERINFO>	

11.3.6 Transfer Processing Status <XFERPRCSTS>

The Transfer Processing Status aggregate contains the current processing status for a transfer. The interpretation of the date value depends on the value of <XFERPRCCODE>.

Tag	Description
<XFERPRCSTS>	Modification-response aggregate
<XFERPRCCODE>	See table 11.3.6.1
<DTXFERPRC>	Transfer processing date; value depends on <XFERPRCCODE>
</XFERPRCSTS>	

11.3.6.1Transfer Processing Status Values <XFERPRCCODE>

Value	Description
WILLPROCESSON	Will be processed on <DTXFERPRC>
POSTEDON	Posted on <DTXFERPRC>
NOFUNDSON	Funds not available to make transfer on <DTXFERPRC>
CANCELEDON	User canceled payment on <DTPMTPRC>
FAILEDON	Unable to make transfer for unspecified reasons on <DTXFERPRC>

11.4Downloading Transactions and Balances

Statement download allows a customer to receive transactions and balances that are typically part of a regular paper statement. Clients can retrieve transactions and balances on a daily basis if they wish. Coupled with the information returned by statement closing information request (see section 11.5), a client can construct an "electronic statement" that contains all of the information that appears on a regular paper statement.

Clients typically allow customers to view these transactions and guide customers through a process of updating their account registers based on the downloaded transactions. By using transaction IDs supplied by financial institutions, Open Financial Exchange makes it possible for clients to ensure that a server downloads each transaction only once. The request also contains starting and ending dates to limit the amount of downloaded data. Clients can remember the last date they received data and use it as the starting date in the next request.

The messages in this chapter are appropriate for checking, savings, money market, credit card, and line of credit accounts. Investment statement download is a superset of bank statement download. Chapter 13 describes the messages specific to investment statement download.

Statement download requires the client to designate an account for the download, and to indicate if the server should download transactions and/or balances. If the client wishes to download transactions, it can specify a date range that the transactions fall within.

The server returns transactions that match the date range (if the client specifies one), and balance information for the account.

Client Sends	Server Responds
Account information	
Include transactions?	
Date range	
	Transactions
	Cycle-ending information

11.4.1Bank Statement Download

A client can request a download of balances separately from transaction detail. The server downloads transactions only if the <INCTRAN> aggregate is present and the <INCLUDE> flag is set to Y. The current ledger balance (and balance date) are always downloaded.

You can use the <STMTRQ> … <STMTRS> request and response pair to download transactions and balances for checking, savings, money market, and line of credit accounts. Section 11.4.2 describes download for credit card accounts.

<DTSTART> and <DTEND> should be interpreted by clients and servers as described in Chapter 3.

11.4.1.1Request <STMTRQ>

The <STMTRQ> request must appear within a <STMTTRNRQ> transaction wrapper.

Tag	Description
<STMTRQ>	Statement-request aggregate
<BANKACCTFROM>	Bank-account-from aggregate
<BANKID>	Routing and transit number, *A-9*
<BRANCHID>	Bank identifier for international banks, *A-22*
<ACCTID>	Account number, *A-22*
<ACCTTYPE>	Type of account, see 11.3.1.1
<ACCTKEY>	Checksum for international banks, *A-22*
</BANKACCTFROM>	
<INCTRAN>	Include-transactions aggregate
<DTSTART>	Start date of statement requested, *datetime*
<DTEND>	End date of statement requested, *datetime*
<INCLUDE>	Include transactions flag, *Boolean*
</INCTRAN>	
</STMTRQ>	

11.4.1.2Response <STMTRS>

A statement response comprises tags supplying various balances, plus zero or more <STMTTRN> aggregates, each describing one statement transaction.

The <STMTRS> response must appear within a <STMTTRNRS> transaction wrapper.

See Chapter 3, "Common Aggregates, Elements, and Data Types," for size and type information for common elements (such as dollar values).

Tag	Description
<STMTRS>	Statement-response aggregate
<CURDEF>	Default currency for the statement, *currsymbol*
<BANKACCTFROM>	Account-from aggregate, see section 11.3.1
</BANKACCTFROM>	
<BANKTRANLIST>	Statement-transaction-data aggregate
<DTSTART>	Start date for transaction data, *date*
<DTEND>	Value that client should send in next <DTSTART> request to ensure that it does not miss any transactions, *date*
<STMTTRN>	Opening tag for each statement transaction (0 or more), see section 11.4.2.3.1
</STMTTRN>	End tag for each statement transaction
</BANKTRANLIST>	
<LEDGERBAL>	Ledger balance aggregate
<BALAMT>	Ledger balance amount, *amount*
<DTASOF>	Balance date, *datetime*
</LEDGERBAL>	
<AVAILBAL>	Available balance aggregate
<BALAMT>	Available balance amount, *amount*
<DTASOF>	Balance date, *datetime*
</AVAILBAL>	
<MKTGINFO>	Marketing information (at most 1), *A-360*
</STMTRS>	

11.4.1.3 Status Codes

Code	Meaning
0	Success
2000	General error (ERROR)
2002	General account error (ERROR)
2003	Account not found (ERROR)
2004	Account closed (ERROR)
2005	Account not authorized (ERROR)
2019	Duplicate request (ERROR)

11.4.2 Credit Card Statement Download

The credit card download request is semantically identical to the bank statement download request. However, the <CCSTMTRQ> aggregate contains the credit card request, not the <STMTRQ> aggregate.

11.4.2.1 Request <CCSTMTRQ>

The <CCSTMTRQ> request must appear within a <CCSTMTTRNRQ> transaction wrapper.

Tag	Description
<CCSTMTRQ>	Credit-card-download-request aggregate
<CCACCTFROM>	Credit-card-account-from aggregate
<ACCTID>	Account number, *A-22*
<ACCTKEY>	Checksum for international banks, *A-22*
</CCACCTFROM>	
<INCTRAN>	Include transactions
<DTSTART>	Start date of statement requested, *datetime*
<DTEND>	Ending date of statement requested, *datetime*
<INCLUDE>	Include transactions flag, *Boolean*
</INCTRAN>	
</CCSTMTRQ>	

11.4.2.2Response <CCSTMTRS>

The credit card download response is semantically identical to the bank statement download response. However, the <CCSTMTRS> aggregate contains the credit card response, not the <STMTRS> aggregate.

The <CCSTMTRS> response must appear within a <CCSTMTTRNRS> transaction wrapper.

Tag	Description
<CCSTMTRS>	Credit-card-download-response aggregate
<CURDEF>	Default currency for the statement, *currsymbol*
<CCACCTFROM>	Account from aggregate, see section 11.3.2
</CCACCTFROM>	
<BANKTRANLIST>	Opening tag for statement transaction data
<DTSTART>	Start date for transaction data, *date*
<DTEND>	Value client should send in next <DTSTART> request to ensure that it does not miss any transactions, *date*
<STMTTRN>	Opening tag for each statement transaction (0 or more), see section 11.4.2.3.1
</STMTTRN>	End tag for each statement transaction
</BANKTRANLIST>	
<LEDGERBAL>	Ledger-balance aggregate
<BALAMT>	Ledger balance amount, *amount*
<DTASOF>	Balance date, *datetime*
</LEDGERBAL>	
<AVAILBAL>	Available balance aggregate
<BALAMT>	Available balance amount, *amount*
<DTASOF>	Balance date, *datetime*
</AVAILBAL>	
<MKTGINFO>	Marketing information (at most 1), *A-360*
</CCSTMTRS>	

11.4.2.3Status Codes

Code	Meaning
0	Success
2001	Invalid account (ERROR)
2002	General account error (ERROR)
2003	Account not found (ERROR)
2004	Account closed (ERROR)
2005	Account not authorized (ERROR)
2019	Duplicate request (ERROR)

11.4.2.3.1 Statement Transaction <STMTTRN>

A <STMTTRN> aggregate describes a single transaction. It identifies the type of the transaction and the date it was posted. The aggregate can also provide additional information to help the customer recognize the transaction: check number, payee name, and memo. The transaction can have a Standard Industrial Code that a client can use to categorize the transaction.

Each <STMTTRN> contains an <FITID> that the client uses to detect whether the server has previously downloaded the transaction.

Transaction amounts are signed from the perspective of the customer. For example, a credit card payment is positive while a credit card purchase is negative.

Tag	Description
<STMTTRN>	Statement-transaction aggregate
<TRNTYPE>	Transaction type, see section 11.4.2.3.1.1 for possible values
<DTPOSTED>	Date transaction was posted to account, *datetime*
<DTUSER>	Date user initiated transaction, if known, *datetime*
<DTAVAIL>	Date funds are available, *datetime*
<TRNAMT>	Amount of transaction, *amount*
<FITID>	Transaction ID issued by financial institution. Used to detect duplicate downloads, *FITID*
<CORRECTFITID>	If present, the FITID of a previously sent transaction that is corrected by this record. This transaction replaces or deletes the transaction that it corrects, based on the value of <CORRECTACTION> below, *FITID*

Tag	Description
<CORRECTACTION>	Actions can be REPLACE or DELETE. REPLACE replaces the transaction referenced by CORRECTFITID; DELETE deletes it.
<SRVRTID>	Server assigned transaction ID; used for transactions initiated by client, such as payment or funds transfer. *SRVRTID*
<CHECKNUM>	Check (or other reference) number, *A-12*
<REFNUM>	Reference number that uniquely identifies the transaction. Can be used in addition to or instead of a <CHECKNUM>, *A-32*
<SIC>	Standard Industrial Code, *N-6*
<PAYEEID>	Payee identifier if available, *A-12*
Payee options. Choose either <NAME> or <PAYEE>. _ _ _ _	
<NAME>	Name of payee or description of transaction, *A-32*
	NOTE: Provide NAME or PAYEE, not both
or-	
<PAYEE>	Payee aggregate, see section 12.5.2.1
</PAYEE>	
Account-to options. Choose either <BANKACCTTO> or <CCACCTTO>. _ _ _ _ _ _	
<BANKACCTTO>	If this was a transfer to an account and the account information is available, see section 11.3.1
</BANKACCTTO>	
or-	
<CCACCTTO>	
</CCACCTTO>	
<MEMO>	Extra information (not in <NAME>), *A-255*
Currency options. Choose either <CURRENCY> or <ORIGCURRENCY>. _ _ _ _ _	
<CURRENCY>	Currency, if different from CURDEF
</CURRENCY>	
-or-	
<ORIGCURRENCY>	
</ORIGCURRENCY>	
</STMTTRN>	

11.4.2.3.1.1 Transaction types used in <TRNTYPE>

Type	Description
CREDIT	Generic credit
DEBIT	Generic debit
INT	Interest earned or paid
	NOTE: *depends on signage of amount*
DIV	Dividend
FEE	FI fee
SRVCHG	Service charge
DEP	Deposit
ATM	ATM debit or credit
	NOTE: *depends on signage of amount*
POS	Point of sale debit or credit
	NOTE: *depends on signage of amount*
XFER	Transfer
CHECK	Check
PAYMENT	Electronic payment
CASH	Cash withdrawal
DIRECTDEP	Direct deposit
DIRECTDEBIT	Merchant initiated debit
REPEATPMT	Repeating payment/standing order
OTHER	Other

11.5 Statement Closing Information

Open Financial Exchange provides a way for customers to receive closing statement information that typically appears as part of a paper statement. This information includes opening and closing dates and balances for a statement period, as well as a detailed breakdown of debits, credits, fees, and interest that are usually part of a paper statement. In addition to this information, clients receive a date range for transactions that correspond to the closing statement. Clients might wish to use this date range to retrieve transactions through statement download in order to present the user with an "electronic" statement.

To request statement information, the client is **REQUIRED** to designate an account for the download. The client can also specify a date range to limit the number of closing information aggregates that the

server returns. If the client does not specify a date range, the server returns as many closing information aggregates as it can.

Client Sends	Server Responds
Account Information	
Date range	
	Cycle-ending information (0 or more)

11.5.1 Statement Closing Download

You can use the <STMTENDRQ> …<STMTENDRS> request and response pair to download statement closing information for checking, savings, money market, and line of credit accounts. Section 11.5.3 describes download for credit card accounts.

11.5.1.1 Request <STMTENDRQ>

The <STMTENDRQ> request must appear within a <STMTENDTRNRQ> transaction wrapper.

Tag	Description
<STMTENDRQ>	Closing-statement-request aggregate
<BANKACCTFROM>	Bank-account-from aggregate
</BANKACCTFROM>	
<DTSTART>	Start date for statement closing information, *datetime*
<DTEND>	End date of statement closing information, *datetime*
</STMTENDRQ>	

11.5.1.2 Response <STMTENDRS>

The <STMTENDRS> response must appear within a <STMTENDTRNRS> transaction wrapper.

Tag	Description
<STMTENDRS>	Closing-statement-response aggregate
<CURDEF>	Default currency used for closing information, *currsymbol*
<BANKACCTFROM>	Account from aggregate, see section 11.3.1
</BANKACCTFROM>	
<CLOSING>	Statement information (0 or more), see section 11.5.2
</CLOSING>	
</STMTENDRS>	

11.5.1.3 Status Codes

Code	Meaning
0	Success
2000	General error (ERROR)
2002	General account error (ERROR)
2003	Account not found (ERROR)
2004	Account closed (ERROR)
2005	Account not authorized (ERROR)
2019	Duplicate request (ERROR)

11.5.2 Non-Credit Card Statement <CLOSING>

A checking, savings, or money market account uses the <CLOSING> aggregate to describe statement closing information.

The <FITID> provides a way for the client to distinguish one closing statement from another.

For each <CLOSING> aggregate returned, clients can retrieve corresponding transactions by using <DTPOSTSTART> and <DTPOSTEND> as <DTSTART> and <DTEND> in a <STMTRQ> request.

Tag	Description
<CLOSING>	Non-credit-card-account-types aggregate
<FITID>	Unique identifier for this statement, *FITID*
<DTOPEN>	Opening statement date, *date*
<DTCLOSE>	Closing statement date, *date*
<DTNEXT>	Closing date of next statement, *date*
<BALOPEN>	Opening statement balance, *amount*
<BALCLOSE>	Closing statement balance, *amount*
<BALMIN>	Minimum balance in statement period, *amount*
<DEPANDCREDIT>	Total of deposits and credits, including interest, *amount*
<CHKANDDEB>	Total of checks and debits, including fees, *amount*
<TOTALFEES>	Total of all fees, *amount*
<TOTALINT>	Total of all interest, *amount*
<DTPOSTSTART>	Start date of transaction data for this statement, *date*
	A client should be able to use this date in a <STMTRQ> to request transactions that match this statement.
<DTPOSTEND>	End date of transaction data for this statement, *date*
	A client should be able to use this date in a <STMTRQ> to request transactions that match this statement.
<MKTGINFO>	Marketing information (at most 1), *A-360*
Currency options. Choose either <CURRENCY> or <ORIGCURRENCY>	
<CURRENCY> </CURRENCY>	Currency, if different from CURDEF
-or-	
<ORIGCURRENCY> </ORIGCURRENCY>	
</CLOSING>	

11.5.3 Credit Card Statement Closing Request <CCSTMTENDRQ>

The credit card statement closing request is semantically identical to the bank statement closing request. However, the <CCSTMTENDRQ> aggregate contains the credit card request, not the <STMTENDRQ> aggregate.

The <CCSTMTENDRQ> request must appear within a <CCSTMTENDTRNRQ> transaction wrapper.

Tag	Description
<CCSTMTENDRQ>	Credit-card-closing-statement-request aggregate
<CCACCTFROM>	Credit-card-account-from aggregate
</CCACCTFROM>	
<DTSTART>	Start date for statement closing information, *datetime*
<DTEND>	End date of statement closing information, *datetime*
</CCSTMTENDRQ>	

11.5.4 Credit Card Statement Closing Response <CCSTMTENDRS>

The credit card statement closing response is semantically identical to the bank statement closing response. However, the <CCSTMTENDRS> aggregate contains the credit card response, not the <STMTENDRS> aggregate.

The <CCSTMTENDRS> response must appear within a <CCSTMTENDTRNRS> transaction wrapper.

Tag	Description
<CCSTMTENDRS>	Credit-card-closing-statement-response aggregate
<CURDEF>	Default currency for closing information, *currsymbol*
<CCACCTFROM>	Account from aggregate, see section 11.3.2
</CCACCTFROM>	
<CCCLOSING>	Statement information (0 or more). See section 11.5.2
</CCCLOSING>	
</CCSTMTENDRS>	

11.5.4.1 Status Codes

Code	Meaning
0	Success
2000	General error (ERROR)
2002	General account error (ERROR)
2003	Account not found (ERROR)
2004	Account closed (ERROR)
2005	Account not authorized (ERROR)
2019	Duplicate request (ERROR)

11.5.4.2Credit Card Statement <CCCLOSING>

A credit card account uses the <CCCLOSING> aggregate to describe statement closing information.

The <FITID> provides a way for the client to distinguish one closing statement from another.

For each <CCCLOSING> returned, clients should be able to retrieve corresponding transactions by using <DTPOSTSTART> and <DTPOSTEND> as <DTSTART> and <DTEND> in a <CCSTMTRQ> request.

Tag	Description
<CCCLOSING>	Credit-card-statement-information aggregate
<FITID>	Unique identifier for this statement, *FITID*
<DTOPEN>	Opening statement date, *date*
<DTCLOSE>	Closing statement date, *date*
<DTNEXT>	Closing date of next statement, *date*
<BALOPEN>	Opening statement balance, *amount*
<BALCLOSE>	Closing statement balance, *amount*
<DTPMTDUE>	Payment due date, *date*
<MINPMTDUE>	Minimum amount due, *amount*
<FINCHG>	Finance charges, *amount*
<PAYANDCREDIT>	Total of payments and credits, *amount*
<PURANDADV>	Total of purchases and cash advances, *amount*
<DEBADJ>	Debit adjustments, *amount*
<CREDITLIMIT>	Current credit limit, *amount*
<DTPOSTSTART>	Start date of transaction data for this statement, *date*
	A client should be able to use this date in a <CCSTMTRQ> to request transactions that match this statement.
<DTPOSTEND>	End date of transaction data for this statement, *date*
	A client should be able to use this date in a <CCSTMTRQ> to request transactions that match this statement.
<MKTGINFO>	Marketing information (at most 1), *A-360*
Currency options. Choose either <CURRENCY> or <ORIGCURRENCY>.	
<CURRENCY> </CURRENCY>	Currency, if different from CURDEF
-or-	
<ORIGCURRENCY> </ORIGCURRENCY>	
</CCCLOSING>	

11.6Stop Check

Open Financial Exchange supports a request to issue a stop payment for one or more outstanding checks. The stop request can be for a single check or for a range of checks. There must be one request for each check or range of checks the user wants to stop.

When stopping a single check, the client can provide a payee name and/or amount instead of a check number to describe the check to stop. Not all servers can support this behavior.

Examples:

Stop check 22	– one request
Stop check to "Acme Lighting"	– one request
Stop checks 200-224	– one request
Stop checks 275-280, 283	– two requests (first stops 275-280, the next stops 283)

Client Sends	Server Responds
Account information	
Check number(s) to stop	
-or-	
Check description	
	Status for each check

11.6.1Stop Check Add

Stop Check Add is subject to synchronization.

11.6.1.1Request <STPCHKRQ>

The <STPCHKRQ> request must appear within a <STPCHKTRNRQ> transaction wrapper.

Tag	Description
<STPCHKRQ>	Stop-check-request aggregate
<BANKACCTFROM>	Account-from aggregate, see section 11.3.1
</BANKACCTFROM>	
Check options. Choose either<CHKRANGE> or <CHKDESC>.	
<CHKRANGE>	Check range aggregate
<CHKNUMSTART>	Start check number to cancel, *A-12*
<CHKNUMEND>	Ending check number to cancel; omit if only one check is to be stopped, *A-12*
</CHKRANGE>	
-or-	
<CHKDESC>	Check description aggregate, see section 11.6.1.1.2
<NAME>	Payee name or description, *A-32*
<CHECKNUM>	Check number to cancel, *A-12*
<DTUSER>	Date on check, *datetime*
<TRNAMT>	Amount, *amount*
</CHKDESC>	
</STPCHKRQ>	

11.6.1.1.1Check Range <CHKRANGE>

Tag	Description
<CHKRANGE>	Check-range aggregate
<CHKNUMSTART>	Start check number to cancel, *A-8*
<CHKNUMEND>	Ending check number to cancel; omit if only one check is to be stopped, *A-8*
</CHKRANGE>	

11.6.1.1.2Check Description <CHKDESC>

A check description must include a payee name or description. It can also include a check number, the date the user wrote the check, and a transaction amount.

Tag	Description
<CHKDESC>	Check description aggregate
<NAME>	Payee name or description, *A-32*
<CHECKNUM>	Check number, *A-12*
<DTUSER>	Date on check, *datetime*
<TRNAMT>	Amount, *amount*
</CHKDESC>	

11.6.1.2Response <STPCHKRS>

Consistent with all responses, the stop check response contains a global status that describes whether the response could be delivered. If the server provides a response, it returns a <STPCHKNUM> aggregate for each check for which the client requested a stop payment. Status code 10000 should be returned if the stop check request is in process; a subsequent synchronization should obtain an updated response with a final status.

The <STPCHKRS> response must appear within a <STPCHKTRNRS> transaction wrapper.

Tag	Description
<STPCHKRS>	Stop-check-response aggregate
<CURDEF>	Default currency for stop check response, *currsymbol*
<BANKACCTFROM>	Account-from aggregate, see section 11.3.1
</BANKACCTFROM>	
<STPCHKNUM>	Stopped check aggregate (1 or more), see section 11.6.1.2.1
</STPCHKNUM>	
<FEE>	Fee for stop check, *amount*
<FEEMSG>	Description of fee, *A-80*
</STPCHKRS>	

11.6.1.2.1 Stopped Check <STPCHKNUM>

This aggregate contains a status code that indicates whether or not a specific check was canceled.

Tag	Description
<STPCHKNUM>	Stopped-check-item aggregate
<CHECKNUM>	Check number, *A-12*
<NAME>	Payee name or description, *A-32*
<DTUSER>	Date on check, *datetime*
<TRNAMT>	Amount, *amount*
<CHKSTATUS>	Status code for individual stop check request 0 = OK 1 = rejected 100 = check not found 101 = check already posted
<CHKERROR>	Further textual explanation, *A-255*
Currency options. Choose either <CURRENCY> or <ORIGCURRENCY>.	
<CURRENCY></CURRENCY> *-or-* *<ORIGCURRENCY></ORIGCURRENCY>*	Currency, if different from CURDEF

11.6.2 Status Codes

Code	Meaning
0	Success
2000	General error (ERROR)
2002	General account error (ERROR)
2003	Account not found (ERROR)
2004	Account closed (ERROR)
2005	Account not authorized (ERROR)
2019	Duplicate request (ERROR)
10000	Stop check in process (INFO)
10500	Too many checks to process (ERROR)

11.7 Intrabank Funds Transfer

Open Financial Exchange supports transferring funds between two accounts at the same financial institution. Funds transfers in Open Financial Exchange can be immediate or scheduled. Scheduled transfers can repeat at specified intervals.

Financial institutions can choose to support:

- Immediate transfers

- Immediate and scheduled transfers

- Immediate, scheduled, and recurring transfers

Recurring transfers require support for scheduled transfers.

All Intrabank Funds Transfer requests are subject to synchronization.

11.7.1 Intrabank Funds Transfer Addition

The Intrabank Funds Transfer Add request provides a way for a client to set up a single transfer. The request designates source and destination accounts and the amount of the transfer. The client must provide a date if it has scheduled the transfer.

Client Sends	Server Responds
Source account	
Destination account	
Amount	
Date of transfer (optional)	
	Server ID for the transfer
	Source account
	Destination account
	Amount
	Expected/actual posting date

Intrabank Funds Transfer Add is subject to synchronization.

11.7.1.1Request <INTRARQ>

The <INTRARQ> request must appear within an <INTRATRNRQ> transaction wrapper.

Tag	Description
<INTRARQ>	Intrabank-transfer-request aggregate
<XFERINFO>	Transfer information aggregate, see section 11.3.5
</XFERINFO>	
</INTRARQ>	

11.7.1.2Response <INTRARS>

A server cannot, in all cases, provide complete confirmation for the transfer. The server can confirm only that it received the transfer instruction; and possibly whether it validated the accounts, amount, and date specified in the transfer. For any transfer where the client does not know the status at the time of the response, a server should confirm that it accepted the instruction and indicate the expected posting date of the transfer. A client can pick up the confirmation at a later date through a synchronization request.

If the request is for an immediate transfer and the server can perform the transfer in real time, the server should indicate whether the transfer succeeded and should return the date of the transfer in <DTPOSTED>. In this case, synchronization is not required.

The <INTRARS> response must appear within an <INTRATRNRS> transaction wrapper.

Tag	Description
<INTRARS>	Intrabank-transfer-response aggregate
<CURDEF>	Default currency for the intrabank transfer response, *currsymbol*
<SRVRTID>	Server ID for this transfer, *SRVRTID*
<XFERINFO>	Transfer information aggregate, see section 11.3.5
</XFERINFO>	
Transfer-date options. Choose either <DTXFERPRJ> or <DTPOSTED> — — — — — —	
<DTXFERPRJ>	Projected date of the transfer; response can contain either a <DTXFERPRJ> or a <DTPOSTED> but not both; *datetime*
-or-	
<DTPOSTED>	Actual date of the transfer, *datetime*
<RECSRVRTID>	If the response is generated by a recurring transfer model, this ID references it, see section 11.10
<XFERPRCSTS>	Transfer-processing status, see section 11.3.6
</XFERPRCSTS>	
</INTRARS>	

NOTE: The server can deliver this response to a client immediately after the request is made (for an immediate or one-time scheduled transfer). The server should also return this response for any transfers that were generated by a model.

11.7.1.3 Status Codes

Code	Meaning
0	Success (INFO)
2000	General error (ERROR)
2002	General account error (ERROR)
2006	Source account not found (ERROR)
2007	Source account closed (ERROR)
2008	Source account not authorized (ERROR)
2009	Destination account not found (ERROR)
2010	Destination account closed (ERROR)
2011	Destination account not authorized (ERROR)
2012	Invalid amount (ERROR)
2014	Date too soon (ERROR)
2015	Date too far in future (ERROR)
2019	Duplicate request (ERROR)
10504	Insufficient funds (ERROR)

11.7.2 Intrabank Funds Transfer Modification

The client sends a Transfer Modification request to modify a scheduled transfer. When modifying a transfer, the client must specify all of the tags within the <XFERINFO> aggregate that were specified when the transfer was created, not just the tags that the client wants to modify. <SRVRTID> specifies the transfer the user wants to modify. Not all servers can support the ability to modify some tag values.

Intrabank Funds Transfer Modification is subject to synchronization.

11.7.2.1 Request <INTRAMODRQ>

The <INTRAMODRQ> request must appear within an <INTRATRNRQ> transaction wrapper.

Tag	Description
<INTRAMODRQ>	Modification-request aggregate
<SRVRTID>	ID assigned by the server to the transfer being modified, *SRVRTID*
<XFERINFO>	Transfer information aggregate, see section 11.3.5
</XFERINFO>	
</INTRAMODRQ>	

11.7.2.2 Response <INTRAMODRS>

This response normally just echoes the values passed by the client. However, if the status of a scheduled transfer changes in any way, clients should expect to receive modification responses when they synchronize with the server. For example, when a server completes a transfer, the status of the transfer goes from *pending* to *posted*. Clients should expect servers to notify them of this status change.

The <INTRAMODRS> response must appear within an <INTRATRNRS> transaction wrapper.

Tag	Description
<INTRAMODRS>	Modification-response aggregate
<SRVRTID>	ID assigned by the server to the transfer being modified, *SRVRTID*
<XFERINFO>	Transfer information aggregate, see section 11.3.5
</XFERINFO>	
<XFERPRCSTS>	Transfer processing status, see section 11.3.6
</XFERPRCSTS>	
</INTRAMODRS>	

11.7.2.3 Status Codes

Code	Meaning
0	Success (INFO)
2000	General error (ERROR)
2002	General account error (ERROR)
2006	Source account not found (ERROR)
2007	Source account closed (ERROR)
2008	Source account not authorized (ERROR)
2009	Destination account not found (ERROR)
2010	Destination account closed (ERROR)
2011	Destination account not authorized (ERROR)
2012	Invalid amount (ERROR)
2014	Date too soon (ERROR)
2015	Date too far in future (ERROR)
2016	Already committed (ERROR)
2017	Already canceled (ERROR)
2018	Unknown server ID (ERROR)
2019	Duplicate request (ERROR)
10500	Too many checks to process (ERROR)

11.7.3 Intrabank Funds Transfer Cancellation

The client sends a Transfer Cancellation request to cancel a scheduled transfer, where <SRVRTID> identifies the transfer.

Intrabank Funds Transfer Cancellation is subject to synchronization.

11.7.3.1Request <INTRACANRQ>

The <INTRACANRQ> request must appear within an <INTRATRNRQ> transaction wrapper.

Tag	Description
<INTRACANRQ>	Transfer-cancellation-request aggregate
<SRVRTID>	ID of the transfer the user wants to cancel. The server must have previously assigned this ID to a transfer. *SRVRTID*
</INTRACANRQ>	

11.7.3.2Response <INTRACANRS>

The <INTRACANRS> response must appear within an <INTRATRNRS> transaction wrapper.

Tag	Description
<INTRACANRS>	Transfer-cancellation-response aggregate
<SRVRTID>	ID of the transfer the user wants to cancel. The server must have previously assigned this ID to a transfer. *SRVRTID*
</INTRACANRS>	

11.7.3.3Status Codes

Code	Meaning
0	Success
2000	General error (ERROR)
2016	Already committed (ERROR)
2017	Already canceled (ERROR)
2018	Unknown server ID (ERROR)
2019	Duplicate request (ERROR)

11.8Interbank Funds Transfer

The Interbank Funds Transfer Add request provides a way for a client to set up a single transfer between accounts at different financial institutions. Like intrabank funds transfers, the request designates source and destination accounts and the amount of the transfer. Also, as in intrabank funds transfers, the FI must

be able to authenticate the source account. However, interbank funds transfers differ from intrabank funds transfers in the following respects:

- The routing and transit number of the destination account differs from the source account.

- At the discretion of an FI, the destination account can be subject to pre-notification.

- Source and destination accounts must be enabled for the Automated Clearing House (ACH).

Use the ACH system to implement the Interbank Funds Transfer, which is subject to the rules and regulations governing the ACH network.

In all other respects, interbank funds transfers function like intrabank funds transfers. The user can schedule, modify, and cancel them. They can recur at regular intervals.

11.8.1 Interbank Funds Transfer – US

In the United States, interbank funds transfers usually use only the <XFERINFO> portion of the request and response.

Client Sends	Server Responds
Source account	
Destination account	
Amount	
Date of transfer (optional)	
	Server ID for the transfer
	Source account
	Destination account
	Amount
	Expected/actual posting date

Interbank Funds Transfer Add is subject to synchronization.

11.8.2 Interbank Funds Transfer – International Usage

In countries where the funds transfer is the basis of the payments system, the Open Financial Exchange payments messages allow specifying payees by destination account (see Chapter 12).

Interbank Funds Transfer Add is subject to synchronization.

11.8.2.1Interbank Funds Transfer Request <INTERRQ>

The <INTERRQ> request must appear within an <INTERTRNRQ> transaction wrapper.

Tag	Description
<INTERRQ>	Interbank-transfer-request aggregate
<XFERINFO>	Transfer information aggregate, see section 11.3.5
</XFERINFO>	
</INTERRQ>	

11.8.2.2Interbank Funds Transfer Response <INTERRS>

The server cannot provide complete confirmation for interbank transfer. It can confirm only that the FI received the transfer instruction and possibly validated the source account, amount, and date specified in the transfer. Since the client does not know the status of the transfer at the time of the response, the server should confirm that it accepted the instruction and indicate the expected posting date of the transfer. The client can pick up the confirmation at a later date through a synchronization request.

The <INTERRS> response must appear within an <INTERTRNRS> transaction wrapper.

Tag	Description
<INTERRS>	Interbank-transfer-response aggregate
<CURDEF>	Currency used in transfer, *currsymbol*
<SRVRTID>	Server ID for this transfer, *SRVRTID*
<XFERINFO>	Transfer information aggregate, see section 11.3.5
</XFERINFO>	
Transfer-date options. Choose either <DTXFERPRJ> or <DTPOSTED>	
<DTXFERPRJ>	Projected date of the transfer; response can contain either a <DTXFERPRJ> or a <DTPOSTED> but not both; *datetime*
-or-	
<DTPOSTED>	Actual date of the transfer, *datetime*
<REFNUM>	Server can generate a reference or check for the transfer, *A-32*
<RECSRVRTID>	If server generates the response by a recurring transfer model, this ID references it. *SRVRTID*
<XFERPRCSTS>	Transfer-processing status, see section 11.3.6
</XFERPRCSTS>	
</INTERRS>	

> *NOTE: A server can deliver this response to a client immediately after the client makes the request (for an immediate or one-time scheduled transfer). In response to a synchronization request by a*

client, the server should provide a second response containing complete status regarding the transfer. It should also return any transfers that it generates by a model.

11.8.2.3 Status Codes

Code	Meaning
0	Success
2000	General error (ERROR)
2002	General account error (ERROR)
2006	Source account not found (ERROR)
2007	Source account closed (ERROR)
2008	Source account not authorized (ERROR)
2009	Destination account not found (ERROR)
2010	Destination account closed (ERROR)
2011	Destination account not authorized (ERROR)
2012	Invalid amount (ERROR)
2014	Date too soon (ERROR)
2015	Date too far in future (ERROR)
2019	Duplicate request (ERROR)
10504	Insufficient funds (ERROR)

11.8.3 Interbank Funds Transfer Modification

The client sends a Transfer Modification request to modify a scheduled transfer. When modifying a transfer, the client must specify all of the tags within the <XFERINFO> aggregate that were specified when the transfer was created, not just the tags that the client wants to modify. <SRVRTID> specifies which transfer to modify. Not all servers will support the ability to modify some tag values.

Interbank Funds Transfer Modification is subject to synchronization.

11.8.3.1 Request <INTERMODRQ>

The <INTERMODRQ> request must appear within an <INTERTRNRQ> transaction wrapper.

Tag	Description
<INTERMODRQ>	Modification-request aggregate
<SRVRTID>	ID assigned by the server to the transfer being modified, *SRVRTID*
<XFERINFO>	Transfer information aggregate, see section 11.3.5
</XFERINFO>	
</INTERMODRQ>	

11.8.3.2 Response <INTERMODRS>

The <INTERMODRS> response must appear within an <INTERTRNRS> transaction wrapper.

Tag	Description
<INTERMODRS>	Modification-response aggregate
<SRVRTID>	ID assigned by the server to the transfer being modified, *SRVRTID*
<XFERINFO>	Transfer information aggregate; server returns if client provided an <XFERINFO> in the request, see section 11.3.5
</XFERINFO>	
<XFERPRCSTS>	Processing status for transfer, see section 11.3.6
</XFERPRCSTS>	
</INTERMODRS>	

11.8.3.3 Status Codes

Code	Meaning
0	Success
2000	General error (ERROR)
2002	General account error (ERROR)
2006	Source account not found (ERROR)
2007	Source account closed (ERROR)
2008	Source account not authorized (ERROR)
2009	Destination account not found (ERROR)
2010	Destination account closed (ERROR)
2011	Destination account not authorized (ERROR)
2012	Invalid amount (ERROR)
2014	Date too soon (ERROR)
2015	Date too far in future (ERROR)
2016	Already committed (ERROR)
2017	Already canceled (ERROR)
2018	Unknown server ID (ERROR)
2019	Duplicate request (ERROR)
10504	Insufficient funds (ERROR)
10505	Cannot modify element (ERROR)

11.8.4 Interbank Funds Transfer Cancellation

The client sends a Transfer Cancellation request to cancel a scheduled, where <SRVRTID> identifies the transfer.

Interbank Funds Transfer Cancellation is subject to synchronization.

11.8.4.1 Request <INTERCANRQ>

The <INTERCANRQ> request must appear within an <INTERTRNRQ> transaction wrapper.

Tag	Description
<INTERCANRQ>	Transfer-cancellation-request aggregate
<SRVRTID>	ID of the transfer to cancel. The server must have previously assigned this ID to a transfer. *SRVRTID*
</INTERCANRQ>	

11.8.4.2 Response <INTERCANRS>

The <INTERCANRS> response must appear within an <INTERTRNRS> transaction wrapper.

Tag	Description
<INTERCANRS>	Transfer-cancellation-response aggregate
<SRVRTID>	ID of the transfer to cancel. The server must have previously assigned this ID to a transfer. *SRVRTID*
</INTERCANRS>	

11.8.4.3 Status Codes

Code	Meaning
0	Success
2000	General error (ERROR)
2016	Already committed (ERROR)
2017	Already canceled (ERROR)
2018	Unknown server ID (ERROR)
2019	Duplicate request (ERROR)

11.9 Wire Funds Transfer

Open Financial Exchange enables clients to set up wire funds transfers. Wire funds transfers are similar to other types of funds transfers. Clients designate a source account that the FI can authenticate and a destination account at the same or a different institution. Clients also designate an amount and an optional date.

The FI must know the originator of the transfer. The beneficiary of the transfer might be an established customer at the same institution.

Open Financial Exchange implements wire funds transfers using the FedWire system, and is subject to its rules and regulations.

In almost all respects, wire funds transfers work like interbank funds transfers. A user can schedule and cancel them. Unlike interbank funds transfers, a user cannot modify Wire funds transfers once they have been set up. A user cannot set up wire funds transfers to recur at regular intervals.

Client Sends	Server Responds
Source account	
Originator	
Receiver	
Amount	
Date of transfer (optional)	
	Server ID for the transfer
	Originator
	Receiver
	Amount
	Expected/actual posting date

11.9.1 Wire Funds Transfer Addition

Wire Funds Transfer Add is subject to synchronization.

11.9.1.1 Request <WIRERQ>

The client prepares a <BANKACCTFROM> aggregate to describe the source account. The <WIREBENEFICIARY> aggregate specifies the destination account. The <WIREDESTBANK> aggregate describes the beneficiary's bank.

The <WIRERQ> request must appear within a <WIRETRNRQ> transaction wrapper.

Tag	Description
<WIRERQ>	Wire-transfer-request aggregate
<BANKACCTFROM>	Source of funds
</BANKACCTFROM>	
<WIREBENEFICIARY>	Wire transfer beneficiary, see section 11.9.1.1.1
</WIREBENEFICIARY>	
<WIREDESTBANK>	Beneficiary's bank
<EXTBANKDESC>	Extended bank description, see section 11.9.1.1.2
</EXTBANKDESC>	
</WIREDESTBANK>	
<TRNAMT>	Transfer amount, *amount*
<DTDUE>	Date to occur, *datetime*
<PAYINSTRUCT>	Payment instructions, *A-255*
</WIRERQ>	

11.9.1.1.1 Wire Beneficiary Aggregate

The wire beneficiary aggregate describes the receiver of a wire transfer.

Tag	Description
<WIREBENEFICIARY>	Wire-beneficiary aggregate
<NAME>	Name of beneficiary, *A-32*
<BANKACCTTO>	Bank details for beneficiary
</BANKACCTTO>	
<MEMO>	Information for the beneficiary, *memo*
</WIREBENEFICIARY>	

11.9.1.1.2Extended Bank Description aggregate

Tag	Description
<EXTBANKDESC>	Extended-bank-description aggregate
<NAME>	Abbreviated name of bank, *A-32*
<BANKID>	Routing: ABA number or S.W.I.F.T. number, *A-9*
<ADDR1>	Bank's address line 1, *A-32*
<ADDR2>	Bank's address line 2, *A-32*
<ADDR3>	Bank's address line 3, *A-32*
<CITY>	Bank's city, *A-32*
<STATE>	Bank's state or province, *A-5*
<POSTALCODE>	Bank's zip code, *A-11*
<COUNTRY>	Bank's country; 3-letter country code from ISO/DIS-3166, *A-3*
<PHONE>	Bank's phone number, *A-32*
</EXTBANKDESC>	

11.9.1.2Response <WIRERS>

The server cannot provide complete confirmation for the transfer. It can confirm only that the server received the transfer instruction and possibly that it validated the source account, amount, and date specified in the transfer. For any transfer where the client does not know the status at the time of the response, the server should confirm that it accepted the instruction and indicate the expected posting date of the transfer. The client can pick up the confirmation at a later date through a synchronization request.

The server can indicate the fee assessed for the transfer by using the <FEE> element in the response. The server can also include a confirmation message in the response.

The <WIRERS> response must appear within a <WIRETRNRS> transaction wrapper.

Tag	Description
<WIRERS>	Wire-transfer-response aggregate
<CURDEF>	Currency used in transfer, *currsymbol*
<SRVRTID>	Server ID for this transfer, *SRVRTID*
<BANKACCTFROM>	Source of funds
</BANKACCTFROM>	
<WIREBENEFICIARY>	Wire transfer beneficiary, see section 11.9.1.1.1
</WIREBENEFICIARY>	
<WIREDESTBANK>	Beneficiary's bank
<EXTBANKDESC>	Extended bank description, see section 11.9.1.1.2
</EXTBANKDESC>	
</WIREDESTBANK>	
<TRNAMT>	Transfer amount, *amount*
<DTDUE>	Date to occur, echoed if provided in request, *datetime*
<PAYINSTRUCT>	Payment instructions, echoed if provided in request, *A-255*
Transfer-date options. Choose either <DTXFERPRJ> or <DTPOSTED>	
<DTXFERPRJ>	Projected date of the transfer; response can contain either a <DTXFERPRJ> or a <DTPOSTED> but not both; *datetime*
-or-	
<DTPOSTED>	Actual date of the transfer, *datetime*
<FEE>	Fee assessed for the transfer, *amount*
<CONFMSG>	Confirmation message, *A-255*
</WIRERS>	

11.9.1.3 Status Codes

Code	Meaning
0	Success
2000	General error (ERROR)
2002	General account error (ERROR)
2006	Source account not found (ERROR)
2007	Source account closed (ERROR)
2008	Source account not authorized (ERROR)
2012	Invalid amount (ERROR)
2014	Date too soon (ERROR)
2015	Date too far in future (ERROR)
2019	Duplicate request (ERROR)
10504	Insufficient funds (ERROR)
10516	Wire beneficiary invalid (ERROR)

11.9.2 Wire Funds Transfer Cancellation

The client sends a Wire Funds Transfer Cancellation Request to cancel a scheduled transfer, where <SRVRTID> identifies the transfer.

Wire Funds Transfer Cancellation is subject to synchronization.

11.9.2.1 Request <WIRECANRQ>

The <WIRECANRQ> request must appear within a <WIRETRNRQ> transaction wrapper.

Tag	Description
<WIRECANRQ>	Wire-transfer-cancellation-request aggregate
<SRVRTID>	ID of the transfer to cancel; server must have previously assigned this ID to a transfer, *SRVRTID*
</WIRECANRQ>	

11.9.2.2Response <WIRECANRS>

The <WIRECANRS> response must appear within a <WIRETRNRS> transaction wrapper.

Tag	Description
<WIRECANRS>	Transfer-cancellation-response aggregate
<SRVRTID>	ID of the transfer to cancel; server must have previously assigned this ID to a transfer, *SRVRTID*
</WIRECANRS>	

11.9.2.3Status Codes

Code	Meaning
0	Success
2000	General error (ERROR)
2016	Already committed (ERROR)
2017	Already canceled (ERROR)
2019	Duplicate request (ERROR)

11.10Recurring Funds Transfer

Open Financial Exchange uses a Recurring Funds Transfer Add request to set up a recurring transfer model. The transfer model generates transfers according to its schedule. Transfers created by a model and retrieved by a customer can be modified or canceled without impacting the model.

A user can create recurring funds transfer models to generate two types of scheduled transfers: interbank and intrabank. You cannot set up recurring wire funds transfers.

For more information on recurring transactions, see Chapter 10.

11.10.1Recurring Intrabank Funds Transfer Addition

A Recurring Intrabank Funds Transfer Add request sets up an intrabank funds transfer that repeats at a specified interval for a specified period of time.

Model-created transfers are retrieved by means of a synchronization request.

Client Sends	Server Responds
Source account	
Destination account	
Amount	
Date of first transfer	
Frequency	
Duration	
	Server ID for the model
	Source account
	Destination account
	Amount
	Date of first transfer
	Frequency
	Duration

Recurring Intrabank Funds Transfer Add is subject to synchronization.

11.10.1.1Request <RECINTRARQ>

The <RECINTRARQ> request must appear within a <RECINTRATRNRQ> transaction wrapper.

Tag	Description
<RECINTRARQ>	Recurring-transfer-request aggregate
<RECURRINST>	Recurring-instructions aggregate, see section 10.2
</RECURRINST>	
<INTRARQ>	Intrabank-transfer-request aggregate, see section 11.7.1.1
</INTRARQ>	
</RECINTRARQ>	

11.10.1.2Response <RECINTRARS>

The <RECINTRARS> response must appear within a <RECINTRATRNRS> transaction wrapper.

Tag	Description
<RECINTRARS>	Recurring-transfer-response aggregate
<RECSRVRTID>	Server-assigned ID for this model, *SRVRTID*
<RECURRINST>	Recurring-instructions aggregate
</RECURRINST>	
<INTRARS>	Intrabank-transfer-response aggregate, see section 11.7.1.2
</INTRARS>	
</RECINTRARS>	

11.10.1.3Status Codes

Code	Meaning
0	Success
2000	General error (ERROR)
2002	General account error (ERROR)
2006	Source account not found (ERROR)
2007	Source account closed (ERROR)
2008	Source account not authorized (ERROR)
2009	Destination account not found (ERROR)
2010	Destination account closed (ERROR)
2011	Destination account not authorized (ERROR)
2014	Date too soon (ERROR)
2015	Date too far in future (ERROR)
2019	Duplicate request (ERROR)
10508	Invalid frequency (ERROR)

11.10.2Recurring Intrabank Funds Transfer Modification

The client sends a Recurring Intrabank Funds Transfer Modification request to modify a recurring intrabank transfer model.

Recurring Intrabank Funds Transfer Modification is subject to synchronization.

11.10.2.1Request <RECINTRAMODRQ>

<RECSRVRTID> identifies the model. The client can indicate whether the changes should apply to pending transfers.

The <RECINTRAMODRQ> request must appear within a <RECINTRATRNRQ> transaction wrapper.

Tag	Description
<RECINTRAMODRQ>	Recurring-modification-request aggregate
<RECSRVRTID>	ID assigned by the server to the model being modified, *SRVRTID*
<RECURRINST>	Recurring-instructions aggregate
</RECURRINST>	
<INTRARQ>	Intrabank-transfer-request aggregate, see section 11.7.1.1
</INTRARQ>	
<MODPENDING>	Modify pending flag, *Boolean*
	If the client sets this flag, the server must modify pending and future transfers.
</RECINTRAMODRQ>	

11.10.2.2Response <RECINTRAMODRS>

The <RECINTRAMODRS> response must appear within a <RECINTRATRNRS> transaction wrapper.

Tag	Description
<RECINTRAMODRS>	Recurring-transfer-modification-request aggregate
<RECSRVRTID>	ID assigned by the server to the model being modified, *SRVRTID*
<RECURRINST>	Recurring-instructions aggregate
</RECURRINST>	
<INTRARS>	Intrabank transfer response aggregate, see section 11.7.1.2
</INTRARS>	
<MODPENDING>	Y if client requested that the server modify pending and future transfers. N if the client did not request that the server modify pending and future transfers. *Boolean*
</RECINTRAMODRS>	

11.10.2.3Status Codes

Code	Meaning
0	Success (INFO)
2000	General error (ERROR)
2002	General account error (ERROR)
2006	Source account not found (ERROR)
2007	Source account closed (ERROR)
2008	Source account not authorized (ERROR)
2009	Destination account not found (ERROR)
2010	Destination account closed (ERROR)
2011	Destination account not authorized (ERROR)
2012	Invalid amount (ERROR)
2014	Date too soon (ERROR)
2015	Date too far in future (ERROR)
2016	Already committed (ERROR)
2017	Already canceled (ERROR)
2019	Duplicate request (ERROR)
10500	Too many checks to process (ERROR)
10505	Cannot modify element (ERROR)
10508	Invalid frequency (ERROR)
10518	Unknown model ID (ERROR)

11.10.3Recurring Intrabank Funds Transfer Cancellation

The client sends a Recurring Intrabank Funds Transfer Cancellation request to cancel a recurring intrabank transfer model.

Recurring Intrabank Funds Transfer Cancellation is subject to synchronization.

11.10.3.1Request <RECINTRACANRQ>

<RECSRVRTID> identifies the model the user wants to cancel. The client can indicate whether the cancel should apply to pending transfers.

The <RECINTRACANRQ> request must appear within a <RECINTRATRNRQ> transaction wrapper.

Tag	Description
<RECINTRACANRQ>	Recurring-transfer-cancellation-request aggregate
<RECSRVRTID>	ID assigned by the server to the model being canceled, *SRVRTID*
<CANPENDING>	Cancel pending flag, *Boolean*
	If the client sets this flag, the server must cancel pending and future transfers.
</RECINTRACANRQ>	

11.10.3.2 Response <RECINTRACANRS>

The <RECINTRACANRS> response must appear within a <RECINTRATRNRS> transaction wrapper.

Tag	Description
<RECINTRACANRS>	Recurring-transfer-cancellation-response aggregate
<RECSRVRTID>	ID assigned by the server to the model being canceled, *SRVRTID*
<CANPENDING>	Cancel pending flag, *Boolean*
	Y if the client requested that the server cancel pending and future transfers. N if the client did not request that the server cancel pending and future transfers.
</RECINTRACANRS>	Ending tag for recurring transfer cancellation response

11.10.3.3 Status Codes

Code	Meaning
0	Success
2000	General error (ERROR)
2019	Duplicate request (ERROR)
10509	Model already canceled (ERROR)
10518	Unknown model ID (ERROR)

11.10.4 Recurring Interbank Funds Transfer Addition

A Recurring Interbank Funds Transfer Add request sets up an interbank funds transfer that repeats at a specified interval for a specified period of time.

The client retrieves model-created transfers by means of a synchronization request.

Client Sends	Server Responds
Source account	
Destination account	
Amount	
Date of first transfer	
Frequency	
Duration	
	Server ID for the model
	Source account
	Destination account
	Amount
	Date of first transfer
	Frequency
	Duration

Recurring Interbank Funds Transfer Add is subject to synchronization

11.10.4.1Request <RECINTERRQ>

The <RECINTERRQ> request must appear within a <RECINTERTRNRQ> transaction wrapper.

Tag	Description
<RECINTERRQ>	Recurring-transfer-request aggregate
<RECURRINST>	Recurring-instructions aggregate
</RECURRINST>	
<INTERRQ>	Interbank-transfer-request aggregate, see section 11.8.2.1
</INTERRQ>	
</RECINTERRQ>	

11.10.4.2Response <RECINTERRS>

The <RECINTERRS> response must appear within a <RECINTERTRNRS> transaction wrapper.

Tag	Description
<RECINTERRS>	Recurring-transfer-response aggregate
<RECSRVRTID>	Server-assigned ID for this model, *SRVRTID*
<RECURRINST>	Recurring-instructions aggregate, see section 10.2
</RECURRINST>	
<INTERRS>	Interbank funds transfer response, see section 11.8.2.2
</INTERRS>	
</RECINTERRS>	

11.10.4.3Status Codes

Code	Meaning
0	Success
2000	General error (ERROR)
2002	General account error (ERROR)
2006	Source account not found (ERROR)
2007	Source account closed (ERROR)
2008	Source account not authorized (ERROR)
2009	Destination account not found (ERROR)
2010	Destination account closed (ERROR)
2011	Destination account not authorized (ERROR)
2012	Invalid amount (ERROR)
2014	Date too soon (ERROR)
2015	Date too far in future (ERROR)
2019	Duplicate request (ERROR)
10504	Insufficient funds (ERROR)
10508	Invalid frequency (ERROR)

11.10.5Recurring Interbank Funds Transfer Modification

The client sends a Recurring Interbank Funds Transfer Modification request to modify a recurring interbank transfer model.

Recurring Interbank Funds Transfer Modification is subject to synchronization.

11.10.5.1 Request <RECINTERMODRQ>

<RECSRVRTID> identifies the model. The client can indicate whether the changes should apply to pending transfers.

The <RECINTERMODRQ> request must appear within a <RECINTERTRNRQ> transaction wrapper.

Tag	Description
<RECINTERMODRQ>	Recurring-modification-request aggregate
<RECSRVRTID>	ID assigned by the server to the model being modified, *SRVRTID*
<RECURRINST>	Recurring-instructions aggregate
</RECURRINST>	
<INTERRQ>	Interbank-funds-transfer-request aggregate, see section 11.8.2.1
</INTERRQ>	
<MODPENDING>	Modify pending flag
	If the client sets this flag, the server must modify pending and future transfers. *Boolean*
</RECINTERMODRQ>	

11.10.5.2 Request <RECINTERMODRS>

The <RECINTERMODRS> response must appear within a <RECINTERTRNRS> transaction wrapper.

Tag	Description
<RECINTERMODRS>	Recurring-transfer-modification-response aggregate
<RECSRVRTID>	ID assigned by the server to the model being modified, *SRVRTID*
<RECURRINST>	Recurring-instructions aggregate
</RECURRINST>	
<INTERRS>	Interbank-funds-transfer-response, see section 11.8.2.2
</INTERRS>	
<MODPENDING>	Modify pending flag, *Boolean*
	Y if the client requested that the server modify pending and future transfers. N if the client did not request that the server modify pending and future transfers.
</RECINTERMODRS>	

11.10.5.3Status Codes

Code	Meaning
0	Success
2000	General error (ERROR)
2002	General account error (ERROR)
2006	Source account not found (ERROR)
2007	Source account closed (ERROR)
2008	Source account not authorized (ERROR)
2009	Destination account not found (ERROR)
2010	Destination account closed (ERROR)
2011	Destination account not authorized (ERROR)
2012	Invalid amount (ERROR)
2014	Date too soon (ERROR)
2015	Date too far in future (ERROR)
2016	Already committed (ERROR)
2017	Already canceled (ERROR)
2019	Duplicate request (ERROR)
10504	Insufficient funds (ERROR)
10505	Cannot modify element (ERROR)
10508	Invalid frequency (ERROR)
10510	Invalid payee ID (ERROR)
10518	Unknown model ID (ERROR)

11.10.6Recurring Interbank Funds Transfer Cancellation

The client sends a Recurring Transfer Cancellation request to cancel a recurring transfer model.

Recurring Transfer Cancellation is subject to synchronization.

11.10.6.1Request <RECINTERCANRQ>

<RECSRVRTID> identifies the model the client wants to cancel. The client can indicate whether the cancel should apply to pending transfers.

The <RECINTERCANRQ> request must appear within a <RECINTERTRNRQ> transaction wrapper.

Tag	Description
<RECINTERCANRQ>	Recurring-transfer-cancellation-request aggregate
<RECSRVRTID>	ID assigned by the server to the model being canceled, *SRVRTID*
<CANPENDING>	Cancel pending flag, *Boolean*
	If the client sets this flag, the server must cancel pending and future transfers.
</RECINTERCANRQ>	

11.10.6.2Response <RECINTERCANRS>

The <RECINTERCANRS> response must appear within a <RECINTERTRNRS> transaction wrapper.

Tag	Description
<RECINTERCANRS>	Recurring-transfer-cancellation-response aggregate
<RECSRVRTID>	ID assigned by the server to the model being canceled, *SRVRTID*
<CANPENDING>	Cancel pending flag, *Boolean*
	Y if the client requested that the server cancel pending and future transfers. N if the client did not request that the server cancel pending and future transfers
</RECINTERCANRS>	

11.10.6.3Status Codes

Code	Meaning
0	Success
2000	General error (ERROR)
2019	Duplicate request (ERROR)
10509	Model already canceled (ERROR)
10518	Unknown model ID (ERROR)

11.11E-Mail and Customer Notification

Open Financial Exchange enables customers to contact their FIs when they have questions regarding their accounts. FIs can also notify their customers of significant events that have occurred regarding their accounts. For example, notification can occur if a customer writes a check that does not clear due to insufficient funds. The server prepares the notification and the client picks it up the next time it synchronizes with the server.

11.11.1Banking E-Mail

Open Financial Exchange currently defines one banking e-mail message that clients can send to an FI. With this message, the user can prepare a message to the FI regarding one of his accounts. The server acknowledges receipt of the message. The FI prepares the response that the client picks up when it synchronizes with the server.

Client Sends	Server Responds
Addressed message	
Bank account information	
	Acknowledgment
.	
.	
.	
Synchronization request	
	Response to customer

11.11.1.1Request <BANKMAILRQ>

The client must identify to which bank account the customer query is related.

The <BANKMAILRQ> request must appear within a <BANKMAILTRNRQ> transaction wrapper.

Tag	Description
<BANKMAILRQ>	Bank-e-mail-request aggregate
Account-from options. Choose either <BANKACCTFROM> or <CCACCTFROM>.	
<BANKACCTFROM>	Account-from aggregate, see section 11.3.1
</BANKACCTFROM>	
-or-	
<CCACCTFROM>	Credit-card-account-from aggregate, see section 11.3.2
</CCACCTFROM>	
<MAIL>	To, from, message information, see Chapter 9, "Customer to FI Communication"
</MAIL>	
</BANKMAILRQ>	

11.11.1.2 Response <BANKMAILRS>

The <BANKMAILRS> response must appear within a <BANKMAILTRNRS> transaction wrapper.

Tag	Description
<BANKMAILRS>	Bank-e-mail-response aggregate
Account-from options. Choose either <BANKACCTFROM> or <CCACCTFROM>.	
<BANKACCTFROM>	Account-from aggregate, see section 11.3.1
</BANKACCTFROM>	
-or-	
<CCACCTFROM>	Credit-card-account-from aggregate, see section 11.3.2
</CCACCTFROM>	
<MAIL>	To, from, message information, see Chapter 9, "Customer to FI Communication"
</MAIL>	
</BANKMAILRS>	

11.11.1.3 Status 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)
2019	Duplicate request (ERROR)
16500	HTML not allowed (ERROR)
16501	Unknown mail To: (ERROR)

11.11.2 Notifications

Open Financial Exchange currently defines two banking notifications that an FI can support:

- Returned check
- Returned deposit

You can implement banking notifications through e-mail and synchronization. The client provides a

<TOKEN> representing its current state with regard to banking notification. (See section 3.2.4.) The server can respond by returning a new token and one or more notification e-mail responses.

Client Sends	Server Responds
Synchronization request with current token	
	New token
	Bank e-mail
	Mail for returned check
	Mail for returned deposit

11.11.3 Returned Check and Deposit Notification

11.11.3.1 Response <CHKMAILRS>

The server returns this response (when a check has been returned), if it receives a banking e-mail synchronization message.

The <CHKMAILRS> response must appear within a <BANKMAILTRNRS> transaction wrapper.

Tag	Description
<CHKMAILRS>	Notification-message-response aggregate
<BANKACCTFROM>	Account-from aggregate, see section 11.3.1
</BANKACCTFROM>	
<MAIL>	To, from, message information, see Chapter 9, "Customer to FI Communication"
</MAIL>	
<CHECKNUM>	Check number, *A-12*
<TRNAMT>	Amount of check, *amount*
<DTUSER>	Customer date on check, *date*
<FEE>	Fee assessed for NSF, *amount*
</CHKMAILRS>	

11.11.3.2Response <DEPMAILRS>

The server returns this response (when a deposit has been returned), if it receives a banking e-mail synchronization message.

The <DEPMAILRS> response must appear within a <BANKMAILTRNRS> transaction wrapper.

Tag	Description
<DEPMAILRS>	Notification-message-response aggregate
<BANKACCTFROM>	Account-from aggregate, see section 11.3.1
</BANKACCTFROM>	
<MAIL>	To, from, message information, see Chapter 9, "Customer to FI Communication"
</MAIL>	
<TRNAMT>	Amount of deposit, *amount*
<DTUSER>	Customer date of deposit, *date*
<FEE>	Fee assessed for NSF, *amount*
</DEPMAILRS>	

11.12Data Synchronization for Banking

Banking customers must be able to obtain the current status of transactions previously sent to the server for processing. For example, once a client schedules a transfer and the transfer date has passed, the customer might wish to verify that the server made the transfer as directed. Also, Open Financial Exchange allows for interactions with the server through multiple clients. This means, for example, that the customer can perform some transactions from a home PC and others from an office computer, with each session seamlessly incorporating the activities performed on the other.

To accomplish these actions, the client uses a synchronization scheme to ensure that it has an accurate copy of the server data that is relevant to the client application.

Banking requires synchronization in the following areas: Stop Check, IntraBank Transfers, InterBank Transfers, Wire Transfers, and Banking Notifications.

11.12.1Data Synchronization for Stop Check

11.12.1.1Request <STPCHKSYNCRQ>

Tag	Description
<STPCHKSYNCRQ>	Synchronization-request aggregate
Client synchronization option; <TOKEN>, <TOKENONLY>, or <REFRESH>	
<TOKEN>	Previous value of <TOKEN> received for this type of synchronization request from server; 0 for first-time requests; *token*
<TOKENONLY>	Request for just the current <TOKEN> without the history, *Boolean*
<REFRESH>	Request for refresh of current state, *Boolean*
<REJECTIFMISSING>	If Y, do not process requests if client <TOKEN> is out of date, *Boolean*
<BANKACCTFROM>	Bank account of interest; token must be interpreted in terms of this account
</BANKACCTFROM>	
<STPCHKTRNRQ>	Stop-check transactions (0 or more)
</STPCHKTRNRQ>	
</STPCHKSYNCRQ>	

11.12.1.2Response <STPCHKSYNCRS>

Tag	Description
<STPCHKSYNCRS>	Synchronization-response aggregate
<TOKEN>	New synchronization token, *token*
<LOSTSYNC>	Y if the token in the synchronization request is older than the earliest entry in the server's history table. In this case, some responses have been lost. N if the token in the synchronization request is newer than or matches a token in the server's history table. *Boolean*
<BANKACCTFROM>	Bank account of interest; token must be interpreted in terms of this account
</BANKACCTFROM>	
<STPCHKTRNRS>	Stop-check transactions (0 or more)
</STPCHKTRNRS>	
</STPCHKSYNCRS>	

11.12.2 Data Synchronization for Intrabank Funds Transfers

11.12.2.1 Request <INTRASYNCRQ>

Tag	Description
<INTRASYNCRQ> Client synchronization option; <TOKEN>, <TOKENONLY>, or <REFRESH>	Synchronization-request aggregate
<TOKEN>	Previous value of <TOKEN> received for this type of synchronization request from server; 0 for first-time requests; *token*
<TOKENONLY>	Request for just the current <TOKEN> without the history, *Boolean*
<REFRESH>	Request for refresh of current state, *Boolean*
<REJECTIFMISSING>	If Y, do not process requests if client <TOKEN> is out of date, *Boolean*
<BANKACCTFROM>	Bank account of interest; token must be interpreted in terms of this account
</BANKACCTFROM>	
<INTRATRNRQ>	Intrabank-funds-transfer transactions (0 or more)
</INTRATRNRQ>	
</INTRASYNCRQ>	

11.12.2.2 Response <INTRASYNCRS>

Tag	Description
<INTRASYNCRS>	Synchronization-response aggregate
<TOKEN>	New synchronization token, *token*
<LOSTSYNC>	Y if the token in the synchronization request is older than the earliest entry in the server's history table. In this case, some responses have been lost. N if the token in the synchronization request is newer than or matches a token in the server's history table. *Boolean*
<BANKACCTFROM>	Bank account of interest; token must be interpreted in terms of this account
</BANKACCTFROM>	
<INTRATRNRS>	Intrabank-funds-transfer transactions (0 or more)
</INTRATRNRS>	
</INTRASYNCRS>	

11.12.3 Data Synchronization for Interbank Funds Transfers

11.12.3.1 Request <INTERSYNCRQ>

Tag	Description
<INTERSYNCRQ>	Synchronization-request aggregate
Client synchronization option; <TOKEN>, <TOKENONLY>, or <REFRESH>	
<TOKEN>	Previous value of <TOKEN> received for this type of synchronization request from server; 0 for first-time requests; *token*
<TOKENONLY>	Request for just the current <TOKEN> without the history, *Boolean*
<REFRESH>	Request for refresh of current state, *Boolean*
<REJECTIFMISSING>	If Y, do not process requests if client <TOKEN> is out of date, *Boolean*
<BANKACCTFROM>	Bank account of interest; token must be interpreted in terms of this account
</BANKACCTFROM>	
<INTERTRNRQ>	Interbank-funds-transfer transactions (0 or more)
</INTERTRNRQ>	
</INTERSYNCRQ>	

11.12.3.2 Response <INTERSYNCRS>

Tag	Description
<INTERSYNCRS>	Synchronization-response aggregate
<TOKEN>	New synchronization token, *token*
<LOSTSYNC>	Y if the token in the synchronization request is older than the earliest entry in the server's history table. In this case, some responses have been lost. N if the token in the synchronization request is newer than or matches a token in the server's history table. *Boolean*
<BANKACCTFROM>	Bank account of interest; token must be interpreted in terms of this account
</BANKACCTFROM>	
<INTERTRNRS>	Interbank-funds-transfer transactions (0 or more)
</INTERTRNRS>	
</INTERSYNCRS>	

11.12.4 Data Synchronization for Wire Funds Transfers

11.12.4.1 Request <WIRESYNCRQ>

Tag	Description
<WIRESYNCRQ>	Synchronization-request aggregate
Client synchronization option; <TOKEN>, <TOKENONLY>, or <REFRESH>	
<TOKEN>	Previous value of <TOKEN> received for this type of synchronization request from server; 0 for first-time requests; *token*
<TOKENONLY>	Request for just the current <TOKEN> without the history, *Boolean*
<REFRESH>	Request for refresh of current state, *Boolean*
<REJECTIFMISSING>	If Y, do not process requests if client <TOKEN> is out of date, *Boolean*
<BANKACCTFROM>	Bank account of interest; token must be interpreted in terms of this account.
</BANKACCTFROM>	
<WIRETRNRQ>	Wire-transfer transactions (0 or more)
</WIRETRNRQ>	
</WIRESYNCRQ>	

11.12.4.2 Response <WIRESYNCRS>

Tag	Description
<WIRESYNCRS>	Synchronization-response aggregate
<TOKEN>	New synchronization token, *token*
<LOSTSYNC>	Y if the token in the synchronization request is older than the earliest entry in the server's history table. In this case, some responses have been lost. N if the token in the synchronization request is newer than or matches a token in the server's history table. *Boolean*
<BANKACCTFROM>	Bank account of interest; token must be interpreted in terms of this account
</BANKACCTFROM>	
<WIRETRNRS>	Wire-transfer transactions (0 or more)
</WIRETRNRS>	
</WIRESYNCRS>	

11.12.5 Data Synchronization for Recurring Intrabank Funds Transfers

11.12.5.1 Request <RECINTRASYNCRQ>

This request will synchronize the client with the server in relation to recurring intrabank transfer models. To synchronize individual transfers that were created by the model (and perhaps canceled by another client), the client must also issue an <INTRASYNCRQ>.

Tag	Description
<RECINTRASYNCRQ> Client synchronization option; <TOKEN>, <TOKENONLY>, or <REFRESH>	Synchronization request
<TOKEN>	Previous value of <TOKEN> received for this type of synchronization request from server; 0 for first-time requests; *token*
<TOKENONLY>	Request for just the current <TOKEN> without the history, *Boolean*
<REFRESH>	Request for refresh of current state, *Boolean*
<REJECTIFMISSING>	If Y, do not process requests if client <TOKEN> is out of date, *Boolean*
<BANKACCTFROM>	Bank account of interest; token must be interpreted in terms of this account.
</BANKACCTFROM>	
<RECINTRATRNRQ>	Recurring-intrabank-funds-transfer transactions (0 or more)
</RECINTRATRNRQ>	
</RECINTRASYNCRQ>	

11.12.5.2 Response <RECINTRASYNCRS>

Tag	Description
<RECINTRASYNCRS>	Synchronization-response aggregate
<TOKEN>	New synchronization token, *token*
<LOSTSYNC>	Y if the token in the synchronization request is older than the earliest entry in the server's history table. In this case, some responses have been lost. N if the token in the synchronization request is newer than or matches a token in the server's history table. *Boolean*
<BANKACCTFROM>	Bank account of interest; token must be interpreted in terms of this account
</BANKACCTFROM>	
<RECINTRATRNRS>	Recurring-intrabank-funds-transfer transactions (0 or more)
</RECINTRATRNRS>	
</RECINTRASYNCRS>	

11.12.6Data Synchronization for Recurring Interbank Funds Transfers

11.12.6.1Request <RECINTERSYNCRQ>

This request will synchronize the client with the server in relation to recurring interbank transfer models. To synchronize individual funds transfers that were created by the model (and perhaps canceled by another client), the client must also issue an <INTERSYNCRQ>.

Tag	Description
<RECINTERSYNCRQ>	Synchronization-request aggregate
Client synchronization option; <TOKEN>, <TOKENONLY>, or <REFRESH>	
<TOKEN>	Previous value of <TOKEN> received for this type of synchronization request from server; 0 for first-time requests; *token*
<TOKENONLY>	Request for just the current <TOKEN> without the history, *Boolean*
<REFRESH>	Request for refresh of current state, *Boolean*
<REJECTIFMISSING>	If Y, do not process requests if client <TOKEN> is out of date, *Boolean*
<BANKACCTFROM>	Bank account of interest; token must be interpreted in terms of this account
</BANKACCTFROM>	
<RECINTERTRNRQ>	Recurring-transfer transactions (0 or more)
</RECINTERTRNRQ>	
</RECINTERSYNCRQ>	

11.12.6.2Response <RECINTERSYNCRS>

Tag	Description
<RECINTERSYNCRS>	Synchronization-response aggregate
<TOKEN>	New synchronization token, *token*
<LOSTSYNC>	Y if the token in the synchronization request is older than the earliest entry in the server's history table. In this case, some responses have been lost. N if the token in the synchronization request is newer than or matches a token in the server's history table. *Boolean*
<BANKACCTFROM>	Bank account of interest; token must be interpreted in terms of this account
</BANKACCTFROM>	
<RECINTERTRNRS>	Recurring-interbank-funds-transfer transactions (0 or more)
</RECINTERTRNRS>	
</RECINTERSYNCRS>	

11.12.7 Data Synchronization for Bank Mail

11.12.7.1 Request <BANKMAILSYNCRQ>

Tag	Description
<BANKMAILSYNCRQ>	Synchronization-request aggregate
Client synchronization option; <TOKEN>, <TOKENONLY>, or <REFRESH>	
<TOKEN>	Previous value of <TOKEN> received for this type of synchronization request from server; 0 for first-time requests; *token*
<TOKENONLY>	Request for just the current <TOKEN> without the history, *Boolean*
<REFRESH>	Request for refresh of current state, *Boolean*
<REJECTIFMISSING>	If Y, do not process requests if client <TOKEN> is out of date, *Boolean*
<INCIMAGES>	Y if the client accepts mail with images in the message body. N if the client does not accept mail with images in the message body. *Boolean*
<USEHTML>	Y if client wants an HTML response, N if client wants plain text, *Boolean*
Account-from options. Choose either <BANKACCTFROM> or <CCACCTFROM>.	
<BANKACCTFROM>	Account-from aggregate, see section 11.3.1
</BANKACCTFROM>	
-or-	
<CCACCTFROM>	Credit-card-account-from aggregate, see section 11.3.2
</CCACCTFROM>	
<BANKMAILTRNRQ>	Bank-mail transactions (0 or more)
</BANKMAILTRNRQ>	
</BANKMAILSYNCRQ>	

11.12.7.2Response <BANKMAILSYNCRS>

Tag	Description
<BANKMAILSYNCRS>	Synchronization-response aggregate
<TOKEN>	New synchronization token, *token*
<LOSTSYNC>	Y if the token in the synchronization request is older than the earliest entry in the server's history table. In this case, some responses have been lost. N if the token in the synchronization request is newer than or matches a token in the server's history table. *Boolean*
Account-from options. Choose either <BANKACCTFROM> or <CCACCTFROM>.	
<BANKACCTFROM>	Account-from aggregate, see section 11.3.1
</BANKACCTFROM>	
-or-	
<CCACCTFROM>	Credit-card-account-from aggregate, see section 11.3.2
</CCACCTFROM>	
<BANKMAILTRNRS>	Bank-mail transactions (0 or more)
</BANKMAILTRNRS>	
</BANKMAILSYNCRS>	

11.12.8Status Codes

All synchronization responses can return the following status codes based on the <BANKACCTFROM> in the synchronization request:

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)

11.13Message Sets and Profile

Open Financial Exchange separates messages that the client and server send into groups called message sets. Each FI defines the message sets that the institution supports. The messages described in this section fall into the following types:

- Banking – includes statement download, closing statement download, bank e-mail, notification, and intrabank funds transfer

- Credit Card – credit card statement download and closing statement download

- Interbank Funds Transfers

- Wire Funds Transfers

Each message set contains options and attributes that allow an FI to customize its use of Open Financial Exchange. For example, an institution can support the Interbank Funds Transfer Message Set (INTERXFERMSGSETV1), but it can choose not to support the recurring form of these transfers.

The profile defines the options and attributes as part of each message-set definition. Each set of options and attributes appears within an aggregate that is specific to a message set. For example, <WIREXFERMSGSETV1> contains all of the options and attributes that pertain to wire transfers.

11.13.1Message Sets and Messages

11.13.1.1Bank Message Set and Messages

11.13.1.1.1 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
</BANKMSGSRQV1>	
</BANKMSGSETV1>	
</BANKMSGSET>	

11.13.1.1.2 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
</BANKMSGSRSV1>	
</BANKMSGSETV1>	
</BANKMSGSET>	

11.13.1.2 Credit Card Message Set and Messages

11.13.1.2.1 Credit Card Message Set Request Messages

Message Set	Message

<CREDITCARDMSGSET>
 	
  <CREDITCARDMSGSETV1>
 	
    <CREDITCARDMSGSRQV1>
 	
    </CREDITCARDMSGSRQV1>
 	
  </CREDITCARDMSGSETV1>
 	
</CREDITCARDMSGSET>
	
	CCSTMTTRNRQ
	
	  CCSTMTRQ
	
	CCSTMTENDTRNRQ
	
	  CCSTMTENDRQ

11.13.1.2.2 Credit Card Message Set Response Messages

Message Set	Message

<CREDITCARDMSGSET>
 	
  <CREDITCARDMSGSETV1>
 	
    <CREDITCARDMSGSRSV1>
 	
    </CREDITCARDMSGSRSV1>
 	
  </CREDITCARDMSGSETV1>
 	
</CREDITCARDMSGSET>
	
	CCSTMTTRNRS
	
	  CCSTMTRS
	
	CCSTMTENDTRNRS
	
	  CCSTMTENDRS

11.13.1.3 Interbank Transfer Message Set and Messages

11.13.1.3.1 Interbank Transfer Message Set Request Messages

Message Set	Message
<INTERXFERMSGSET>	
<INTERXFERMSGSETV1>	
<INTERXFERMSGSRQV1>	INTERTRNRQ
	INTERRQ
	INTERMODRQ
	INTERCANRQ
	RECINTERTRNRQ
	RECINTERRQ
	RECINTERMODRQ
	RECINTERCANRQ
	INTERSYNCRQ
	RECINTERSYNCRQ
</INTERXFERMSGSRQV1>	
</INTERXFERMSGSETV1>	
</INTERXFERMSGSET>	

11.13.1.3.2 Interbank Transfer Message Set Response Messages

Message Set	Message
<INTERXFERMSGSET>	
<INTERXFERMSGSETV1>	
<INTERXFERMSGSETRSV1>	INTERTRNRS
	INTERRS
	INTERMODRS
	INTERCANRS
	RECINTERTRNRS
	RECINTERRS
	RECINTERMODRS
	RECINTERCANRS
	INTERSYNCRS
	RECINTERSYNCRS
</INTERXFERMSGSRSV1>	
</INTERXFERMSGSETV1>	
</INTERXFERMSGSET>	

11.13.1.4Wire Transfer Message Set and Messages

11.13.1.4.1Wire Transfer Message Set Request Messages

Message Set	Message

\<WIREXFERMSGSET\>
  \<WIREXFERMSGSETV1\>
    \<WIREXFERMSGSRQV1\>
 	
    \</WIREXFERMSGSRQV1\>
  \</WIREXFERMSGSETV1\>
\</WIREXFERMSGSET\>
	WIRETRNRQ
	  WIRERQ
	  WIRECANRQ
	WIRESYNCRQ

11.13.1.4.2Wire Transfer Message Set Response Messages

Message Set	Message

\<WIREXFERMSGSET\>
  \<WIREXFERMSGSETV1\>
    \<WIREXFERMSGSRSV1\>
 	
    \</WIREXFERMSGSRSV1\>
  \</WIREXFERMSGSETV1\>
\</WIREXFERMSGSET\>
	WIRETRNRS
	  WIRERS
	  WIRECANRS
	WIRESYNCRS

11.13.2Bank Message Set Profile

Tag	Description
\<BANKMSGSET\>	Message set for banking
\<BANKMSGSETV1\>	Version 1 of message set
\<MSGSETCORE\>	Common message-set core
\</MSGSETCORE\>	
\<INVALIDACCTTYPE\>	Account type not supported in \<BANKACCTFROM\>; 0 or more of account types, see section 11.3.1.1 for values
\<CLOSINGAVAIL\>	Closing statement information available, *Boolean*
\<XFERPROF\>	Intrabank transfer profile (if supported)
\<PROCDAYSOFF\>	Days of week that no processing occurs: MONDAY, TUESDAY, WEDNESDAY, THURSDAY, FRIDAY, SATURDAY, or SUNDAY. 0 or more \<PROCDAYSOFF\> can be sent.
\<PROCENDTM\>	Time of day that day's processing ends, *time*
\<CANSCHED\>	Supports scheduled transfers, *Boolean*
\<CANRECUR\>	Supports recurring transfers, *Boolean*. Requires \<CANSCHED\>
\<CANMODXFERS\>	Permit modifications to transfers, i.e. \<INTRAMODRQ\>, *Boolean*
\<CANMODMDLS\>	Permit modifications to models, i.e. \<RECINTRAMODRQ\>, *Boolean*
\<MODELWND\>	Model window; the number of days before a recurring transaction is scheduled to be processed that it is instantiated on the system, *N-3*
\<DAYSWITH\>	Number of days before processing date that funds are withdrawn, *N-3*
\<DFLTDAYSTOPAY\>	Default number of days to pay, *N-3*
\</XFERPROF\>	
\<STPCHKPROF\>	Stop check profile (if supported)
\<PROCDAYSOFF\>	Days of week that no processing occurs: MONDAY, TUESDAY, WEDNESDAY, THURSDAY, FRIDAY, SATURDAY, or SUNDAY. 0 or more \<PROCDAYSOFF\> can be sent.
\<PROCENDTM\>	Time of day that day's processing ends, *time*
\<CANUSERANGE\>	Can stop a range of checks, *Boolean*.
\<CANUSEDESC\>	Can stop by description, *Boolean*.
\<STPCHKFEE\>	Default stop check free *Amount*
\</STPCHKPROF\>	
\<EMAILPROF\>	E-mail profile
\<CANEMAIL\>	Supports generalized banking e-mail, *Boolean*
\<CANNOTIFY\>	Supports notification (of any kind), *Boolean*
\</EMAILPROF\>	
\</BANKMSGSETV1\>	End of bank message set version 1
\</BANKMSGSET\>	End of bank message set

11.13.3 Credit Card Message Set Profile

Tag	Description
<CREDITCARDMSGSET>	Beginning tag for credit card message set
<CREDITCARDMSGSETV1>	Version 1 of message set
<MSGSETCORE>	Common message-set core
</MSGSETCORE>	
<CLOSINGAVAIL>	Closing statement information available, *Boolean*
</CREDITCARDMSGSETV1>	Ending tag of credit card message set version 1
</CREDITCARDMSGSET>	Ending tag of credit card message set

11.13.4 Interbank Funds Transfer Message Set Profile

Tag	Description
<INTERXFERMSGSET>	Beginning tag for interbank transfers message set
<INTERXFERMSGSETV1>	Version 1 of message set
<MSGSETCORE>	Common message-set core
</MSGSETCORE>	
<XFERPROF>	Interbank transfer profile
<PROCDAYSOFF>	Days of week that no processing occurs: MONDAY, TUESDAY, WEDNESDAY, THURSDAY, FRIDAY, SATURDAY, or SUNDAY. 0 or more <PROCDAYSOFF> can be sent.
<PROCENDTM>	Time of day that day's processing ends, *time*
<CANSCHED>	Supports scheduled transfers, *Boolean*
<CANRECUR>	Supports recurring transfers, *Boolean* (Requires <CANSCHED>)
<CANMODXFERS>	Permit modifications to transfers, i.e. <INTERMODRQ>, *Boolean*
<CANMODMDLS>	Permit modifications to models, i.e. <RECINTERMODRQ>, *Boolean*
<MODELWND>	Model window; the number of days before a recurring transaction is scheduled to be processed that it is instantiated on the system, *N-3*
<DAYSWITH>	Number of days before processing date that funds are withdrawn, *N-3*
<DFLTDAYSTOPAY>	Default number of days to pay, *N-3*
</XFERPROF>	
<CANBILLPAY>	Server is capable of handling bill payment as a form of transfers, *Boolean*
<CANCELWND>	Number of days after an interbank transfer occurs that it can be canceled, *N-3*
<DOMXFERFEE>	Standard fee for a domestic interbank transfer, *N-8*
<INTLXFERFEE>	Standard fee for an international interbank transfer, *N-8*
</INTERXFERMSGSETV1>	End of interbank transfer message set version 1
</INTERXFERMSGSET>	End of interbank transfer message set

11.13.5Wire Transfer Message Set Profile

Tag	Description
\<WIREXFERMSGSET\>	Core message set for wire transfers
\<WIREXFERMSGSETV1\>	Version 1 of message set
\<MSGSETCORE\>	Common message-set core
\</MSGSETCORE\>	
\<PROCDAYSOFF\>	Days of week that no processing occurs: MONDAY, TUESDAY, WEDNESDAY, THURSDAY, FRIDAY, SATURDAY, or SUNDAY. 0 or more \<PROCDAYSOFF\> can be sent.
\<PROCENDTM\>	Time of day that day's processing ends, *time*
\<CANSCHED\>	Supports scheduled transfers, *Boolean*
\<DOMXFERFEE\>	Standard fee for a domestic wire transfer, *N-8*
\<INTLXFERFEE\>	Standard fee for an international wire transfer, *N-8*
\</WIREXFERMSGSETV1\>	End of wire transfer message set version 1
\</WIREXFERMSGSET\>	Ending tag of wire transfer message set

11.14Examples

11.14.1Statement Download

This example represents a customer who requests a statement download for a checking account. The request omits \<DTSTART\> and \<DTEND\> because the client is interested in getting all available data. The response contains an updated balance for the account and two transactions.

The request file:

```
<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 -->
```

The response file:

```
<OFX>                                   <!-- Begin response data -->
  <SIGNONMSGSRSV1>
     <SONRS>                            <!-- Begin signon -->
        <STATUS>                        <!-- Begin status aggregate -->
           <CODE>0                      <!-- OK -->
           <SEVERITY>INFO
        </STATUS>
        <DTSERVER>19961029101003        <!-- Oct. 29, 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>
     <STMTTRNRS>                        <!-- First response -->
        <TRNUID>1001                    <!-- Client ID sent in request -->
        <STATUS>                        <!-- Start status aggregate -->
           <CODE>0                      <!-- OK -->
           <SEVERITY>INFO
        </STATUS>
        <STMTRS>                        <!-- Begin statement response -->
           <CURDEF>USD
           <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 -->
           <BANKTRANLIST>               <!-- Begin list of statement trans. -->
              <DTSTART>19961001         <!-- Start date: Oct. 1, 1996 -->
              <DTEND>19961028           <!-- End date: Oct. 28, 1996 -->
              <STMTTRN>                 <!-- First statement transaction -->
                 <TRNTYPE>CHECK         <!-- Check -->
                 <DTPOSTED>19961004     <!-- Posted on Oct. 4, 1996 -->
                 <TRNAMT>200.00         <!-- $200.00 -->
                 <FITID>00002           <!-- Unique ID -->
                 <CHECKNUM>1000         <!-- Check number -->
              </STMTTRN>                <!-- End statement transaction -->
              <STMTTRN>                 <!-- Second transaction -->
                 <TRNTYPE>ATM           <!-- ATM transaction -->
                 <DTPOSTED>19961020     <!-- Posted on Oct. 20, 1996 -->
                 <DTUSER>19961020       <!-- User date of Oct. 20, 1996 -->
                 <TRNAMT>300.00         <!-- $300.00 -->
                 <FITID>00003           <!-- Unique ID -->
              </STMTTRN>                <!-- End statement transaction -->
           </BANKTRANLIST>             <!-- End list of statement trans. -->
           <LEDGERBAL>                  <!-- Ledger balance aggregate -->
              <BALAMT>200.29            <!-- Bal amount: $200.29 -->
              <DTASOF>199610291120      <!-- Bal date: 10/29/96, 11:20 am -->
           </LEDGERBAL>                 <!-- End ledger balance -->
           <AVAILBAL>                   <!-- Available balance aggregate -->
              <BALAMT>200.29            <!-- Bal amount: $200.29 -->
```

```
            <DTASOF>199610291120          <!-- Bal date: 10/29/96, 11:20 am -->
        </AVAILBAL>                       <!-- End available balance -->
      </STMTRS>                           <!-- End statement response -->
    </STMTTRNRS>                          <!-- End of transaction -->
  </BANKMSGSRSV1>
</OFX>                                    <!-- End of response data -->
```

11.14.2

11.14.3Intrabank Funds Transfer

This example is for a customer who requests an immediate funds transfer of $200.00 from a checking account to a savings account.

The request file:

```
<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 -->
```

The response file:

```
<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 -->
```

11.14.4

11.14.5Stop Check

This example represents a customer who requests a stop for checks 200 through 202. The response indicates that the first check (200) has already posted; the server has stopped the rest of the checks in the range.

The request file:
```
<OFX>                                    <!-- Begin request data -->
   <SIGNONMSGSRQV1>
      <SONRQ>                            <!-- Begin signon -->
         <DTCLIENT>19961029101000        <!-- Oct. 29, 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>
      <STPCHKTRNRQ>                      <!-- First request in file -->
         <TRNUID>1001                    <!-- Client's ID for this request -->
         <STPCHKRQ>                      <!-- Begin stop check 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 -->
            <CHKRANGE>                      <!-- Cancel a range of checks -->
                <CHKNUMSTART>200            <!-- Starting check number -->
                <CHKNUMEND>202              <!-- Ending check number -->
            </CHKRANGE>                     <!-- End range -->
        </STPCHKRQ>                         <!-- End of stop check request -->
    </STPCHKTRNRQ>                          <!-- End of first request -->
  </BANKMSGSRQV1>
</OFX>                                      <!-- End of request data -->
```

A response file:

```
<OFX>                                       <!-- Begin response data -->
  <SIGNONMSGSRSV1>
    <SONRS>                                 <!-- Begin signon -->
        <STATUS>                            <!-- Start status aggregate -->
            <CODE>0                         <!-- OK -->
            <SEVERITY>INFO
        </STATUS>
        <DTSERVER>19961029101003            <!-- Oct. 29, 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>
    <STPCHKTRNRS>                           <!-- First response in file -->
        <TRNUID>1001                        <!-- Client ID sent in request -->
        <STATUS>                            <!-- Begin status aggregate -->
            <CODE>0                         <!-- OK -->
            <SEVERITY>INFO
        </STATUS>                           <!-- End of status aggregate -->
        <STPCHKRS>                          <!-- Begin stop check response -->
            <CURDEF>USD
            <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 -->
            <STPCHKNUM>                     <!-- First stopped check -->
                <CHECKNUM>200               <!-- Check 200 -->
                <CHKSTATUS>101              <!-- Too late - already posted -->
            </STPCHKNUM>                    <!-- End of first stopped check -->
            <STPCHKNUM>                     <!-- Second stopped check -->
                <CHECKNUM>201               <!-- Check 201 -->
                <CHKSTATUS>0                <!-- OK -->
            </STPCHKNUM>                    <!-- End of second stopped check -->
            <STPCHKNUM>                     <!-- Third stopped check -->
                <CHECKNUM>202               <!-- Check 202 -->
                <CHKSTATUS>0                <!-- OK -->
            </STPCHKNUM>                    <!-- End of third stopped check -->
            <FEE>10.00
            <FEEMSG>Fee for stop payment.
        </STPCHKRS>                         <!-- End stop check response -->
    </STPCHKTRNRS>                          <!-- End of transaction -->
  </BANKMSGSRSV1>
</OFX>                                       <!-- End of response data -->
```

11.14.6

11.14.7 Recurring Transfers

This example represents a customer who creates a transfer model and then cancels it. To follow the life of the model (and the transfers it creates), the example includes sessions that occur over a two month period.

The model is added on November 1 and scheduled to start on November 15. The model creates transfers of $1000 from a checking to a savings account. The schedule is open-ended.

The client sends two requests: one to create the model and another to collect any transfers created by the model. The second request is a simple synchronize request.

The request file that creates the model

The client sends the file on November 1:

```
<OFX>                                   <!-- Begin request data -->
   <SIGNONMSGSRQV1>
      <SONRQ>                           <!-- Begin signon -->
         <DTCLIENT>19961101101000       <!-- Nov 1, 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>
      <RECINTRATRNRQ>                   <!-- First request in file -->
         <TRNUID>1001                   <!-- Client's ID for this request -->
         <RECINTRARQ>                   <!-- Begin request -->
            <RECURRINST>                <!-- Begin recurring aggregate -->
               <FREQ>MONTHLY            <!-- Monthly schedule -->
            </RECURRINST>               <!-- End recur aggregate -->
            <INTRARQ>
            <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 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>1000.00          <!-- Amount of transfer    -->
               <DTDUE>19961115          <!-- First transfer - Nov.15 -->
            </XFERINFO>                 <!-- End transfer aggregate -->
               </INTRARQ>
         </RECINTRARQ>                  <!-- End transfer request -->
```

```
        </RECINTRATRNRQ>                <!-- End first request -->
        <INTRASYNCRQ>                   <!-- Sync intrabank transfers -->
            <TOKEN>0                    <!-- Token held by the client -->
            <REJECTIFMISSING>N
                <BANKACCTFROM>          <!-- Identify the account -->
                    <BANKID>121099999
                    <ACCTID>999988
                    <ACCTTYPE>CHECKING
                </BANKACCTFROM>
        </INTRASYNCRQ>                  <!-- End of sync request -->
    </BANKMSGSRQV1>
</OFX>                                  <!-- End of request data -->




<OFX>                                   <!-- Begin request data -->
    <SIGNONMSGSRQV1>
        <SONRQ>                         <!-- Begin signon -->
            <DTCLIENT>19961101101000    <!-- Nov 1, 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>
        <RECINTRATRNRQ>                 <!-- First request in file -->
            <TRNUID>1001                <!-- Client's ID for this request -->
            <RECINTRARQ>                <!-- Begin request -->
                <RECURRINST>            <!-- Begin recurring aggregate -->
                    <FREQ>MONTHLY       <!-- Monthly schedule -->
                </RECURRINST>           <!-- End recur aggregate -->
                <INTRARQ>
                <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 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>1000.00       <!-- Amount of transfer    -->
                  <DTDUE>19961115       <!-- First transfer - Nov.15 -->
                </XFERINFO>            <!-- End transfer aggregate -->
                </INTRARQ>
            </RECINTRARQ>               <!-- End transfer request -->
        </RECINTRATRNRQ>                <!-- End first request -->
        <INTRASYNCRQ>                   <!-- Sync intrabank transfers -->
            <TOKEN>0                    <!-- Token held by the client -->
            <REJECTIFMISSING>N
                <BANKACCTFROM>          <!-- Identify the account -->
                    <BANKID>121099999
                    <ACCTID>999988
```

```
            <ACCTTYPE>CHECKING
         </BANKACCTFROM>
      </INTRASYNCRQ>                       <!-- End of sync request -->
   </BANKMSGSRQV1>
</OFX>                                     <!-- End of request data -->
```

The server response provides status for the add recurring transfer request. Assuming that the user creates transfers 30 days prior to posting, and since the client included a funds transfer synchronize request, the server returns status for the first transfer that the model created. This response comes back since the first transfer is scheduled to occur on November 15 and this date falls within 30 days of our session. Had the starting date been more than 30 days from our signon date, the response would have contained only status for the add recurring transfer request.

The response file from the server:

```
<OFX>                                      <!-- Begin response data -->
   <SIGNONMSGSRSV1>
      <SONRS>                              <!-- Begin signon -->
         <STATUS>                          <!-- Start of status aggregate-->
            <CODE>0                        <!-- OK -->
            <SEVERITY>INFO
         </STATUS>
         <DTSERVER>19961101101003          <!-- Nov. 1, 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>
      <RECINTRATRNRS>                      <!-- First response in file -->
         <TRNUID>1001                      <!-- Client ID sent in request -->
         <STATUS>                          <!-- Start of status aggregate -->
            <CODE>0                        <!-- OK -->
            <SEVERITY>INFO
         </STATUS>
         <RECINTRARS>                      <!-- Begin response -->
            <RECSRVRTID>20000              <!-- Server assigned ID -->
            <RECURRINST>                   <!-- Begin recurring aggregate -->
               <FREQ>MONTHLY               <!-- Monthly schedule -->
            </RECURRINST>                  <!-- End of recurring aggregate -->
               <INTRARS>
               <CURDEF>USD
               <SRVRTID>1000
            <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>1000.00             <!-- Amount of transfer -->
               <DTDUE>19961115             <!-- First transfer - Nov. 15 -->
            </XFERINFO>                    <!-- End of transfer aggregate -->
               </INTRARS>
```

```
        </RECINTRARS>                   <!-- End of response -->
      </RECINTRATRNRS>                  <!-- End of first response -->
      <INTRATRNRS>                      <!-- Second response in file -->
        <TRNUID>1001                    <!-- Client ID sent in request -->
        <STATUS>                        <!-- Success -->
          <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>1000.00             <!-- Amount of transfer -->
          </XFERINFO>                   <!-- End transfer aggregate -->
          <DTXFERPRJ>19961115           <!-- Date (interpret with code) -->
          <RECSRVRTID>20000             <!-- Model that created this xfer -->
        </INTRARS>                      <!-- End of transfer response -->
      </INTRATRNRS>                     <!-- End of second response -->
    </BANKMSGSRSV1>
</OFX>                                  <!-- End of response data -->
```

Suppose the customer does not attempt to connect between November 1 and January 1. When the customer does attempt to connect, it is to cancel the recurring transfer model. Since the client sets the <CANPENDING> flag, the cancel is immediate. Any pending transfers are canceled along with the model.

In this example, the client tries to synchronize with the server by requesting any uncollected transfer responses and recurring transfer responses. It does this by sending two synchronization requests. The first is to collect recurring transfer responses. To accomplish this, the recurring transfer cancel request is "wrapped" in a synchronization request. The second synchronization request is to collect individual transfer responses.

The tokens provided in the sync requests tell the server how up to date the client is with respect to recurring transfer models and individual transfers.

The request file:

```
<OFX>                                  <!-- Begin request data -->
  <SIGNONMSGSRQV1>
    <SONRQ>                            <!-- Begin signon -->
      <DTCLIENT>19970101101000         <!-- Jan. 1, 1997, 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>
      <RECINTRASYNCRQ>                  <!-- Sync recurring transfers -->
         <TOKEN>324789987              <!-- Token held by the client -->
         <REJECTIFMISSING>N
         <BANKACCTFROM>
             <BANKID>121099999
             <ACCTID>99998
             <ACCTTYPE>CHECKING
         </BANKACCTFROM>
         <RECINTRATRNRQ>                <!-- First request in file -->
             <TRNUID>1005              <!-- Client's ID for this request -->
             <RECINTRACANRQ>           <!-- Begin recur transfer cancel -->
                 <RECSRVRTID>20000     <!-- ID of the model -->
                 <CANPENDING>Y         <!-- Cancel pending transfers -->
             </RECINTRACANRQ>          <!-- End recur transfer cancel -->
         </RECINTRATRNRQ>              <!-- End of first request -->
      </RECINTRASYNCRQ>                <!-- End of sync request -->
      <INTRASYNCRQ>                     <!-- Sync intrabank transfers -->
         <TOKEN>222289987              <!-- Token held by the client -->
         <REJECTIFMISSING>N
         <BANKACCTFROM>
             <BANKID>121099999
             <ACCTID>99998
             <ACCTTYPE>CHECKING
         </BANKACCTFROM>
      </INTRASYNCRQ>                    <!-- End of sync request -->
  </BANKMSGSRQV1>
</OFX>                                  <!-- End of request data -->
```

The server responds to this message by canceling the model and by canceling any pending transfers that the model has created. Since the customer last connected, two transfers have posted, November 15 and December 15, and a third transfer has been scheduled for January 15. Since the customer has not connected after November 1, the December and January the client has not retrieved the transfer responses. The server will return a total of four transfer responses; two for each of the transfers that the client has not received. This includes two responses for adding the December 15 and January 15 transfers, and two responses for the posting of the December 15 transfer and the cancel of the January 15 transfers.

The response file:
```
<OFX>                                   <!-- Begin response data -->
  <SIGNONMSGSRSV1>
      <SONRS>                           <!-- Begin signon -->
         <STATUS>                       <!-- Start of status aggregate -->
             <CODE>0                   <!-- OK -->
             <SEVERITY>INFO
         </STATUS>
         <DTSERVER>19970101101003      <!-- Nov. 1, 1997, 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>
```

```
       <RECINTRASYNCRS>                      <!-- Sync response -->
          <TOKEN> 4567888898                 <!-- New token -->
<BANKACCTFROM>
          <BANKID>121099999
          <ACCTID>99998
          <ACCTTYPE>CHECKING
       </BANKACCTFROM>
       <RECINTRATRNRS>                       <!-- First response in file -->
          <TRNUID>1005                       <!-- Client ID sent in request -->
          <STATUS>                           <!-- Start of status aggregate -->
             <CODE>0                         <!-- OK -->
             <SEVERITY>INFO
          </STATUS>
          <RECINTRACANRS>                    <!-- Begin cancel model -->
             <RECSRVRTID>20000               <!-- Model that was canceled -->
                   <CANPENDING>Y
          </RECINTRACANRS>                   <!-- End of cancel model -->
       </RECINTRATRNRS>                      <!-- End of first response -->
     </RECINTRASYNCRS>                       <!-- End of first sync response -->
     <INTRASYNCRS>                           <!-- Sync response for xfers -->
          <TOKEN> 2356789011                 <!-- New token -->
<BANKACCTFROM>
          <BANKID>121099999
          <ACCTID>99998
          <ACCTTYPE>CHECKING
       </BANKACCTFROM>
       <INTRATRNRS>                          <!-- First response -->
          <TRNUID>1005                       <!-- Client ID sent in request -->
          <STATUS>                           <!-- Success -->
             <CODE>0                         <!-- OK -->
             <SEVERITY>INFO
          </STATUS>
          <INTRARS>                          <!-- This is the Dec. 15 add -->
            <CURDEF>USD
            <SRVRTID>1003                    <!-- 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>19961215              <!-- Date (interpret with code) -->
            <RECSRVRTID>20000                <!-- Model that was canceled -->
          </INTRARS>                         <!-- End of Dec. 15 transfer -->
       </INTRATRNRS>                         <!-- End of first response -->
       <INTRATRNRS>                          <!-- Second response -->
          <TRNUID>1005                       <!-- Client ID sent in request -->
          <STATUS>
             <CODE>0                         <!-- OK -->
             <SEVERITY>INFO
          </STATUS>
          <INTRAMODRS>                       <!-- This is the Dec. 15 post -->
             <SRVRTID>1003                   <!-- 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 -->
      <XFERPRCSTS>                  <!-- Status of transfer -->
         <XFERPRCCODE>POSTEDON      <!-- Status code -->
         <DTXFERPRC>19961215        <!-- Date (interpret with code) -->
      </XFERPRCSTS>                 <!-- End of transfer status -->
   </INTRAMODRS>                    <!-- End of Dec. 15 transfer -->
</INTRATRNRS>                       <!-- End of second response -->
<INTRATRNRS>                        <!-- Third response -->
   <TRNUID>1006                     <!-- Client ID sent in request -->
   <STATUS>
      <CODE>0                       <!-- OK -->
      <SEVERITY>INFO
   </STATUS>
   <INTRARS>                        <!-- This is the Jan. 15 add -->
      <CURDEF>USD
      <SRVRTID>1004                 <!-- 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>19970115           <!-- Date of expected posting -->
      <RECSRVRTID>20000             <!-- Model that was canceled -->
   </INTRARS>                       <!-- End of Jan. 15 transfer -->
</INTRATRNRS>                       <!-- End of third response -->
<INTRATRNRS>                        <!-- Fourth response -->
   <TRNUID>1007                     <!-- Client ID sent in this request -->
   <STATUS>
      <CODE>0                       <!-- OK -->
      <SEVERITY>INFO
   </STATUS>
   <INTRACANRS>                     <!-- This is the Jan. 15 cancel -->
      <SRVRTID>1004                 <!-- Server ID for Jan. 15 xfer -->
   </INTRACANRS>                    <!-- End of Jan. 15 cancel -->
</INTRATRNRS>                       <!-- End of third xfer response -->
</INTRASYNCRS>                      <!-- End of second sync response -->
  </BANKMSGSRSV1>
</OFX>                              <!-- End of response -->
```