Open Financial Exchange Specification 1.0.2

May 30, 1997

© 1997 CheckFree Corp., Intuit Inc., Microsoft Corp. All rights reserved

Chapters 12 & 13

Contents

12. PAYMENTS.....

12.1 CONSUMER AND BUSINESS PAYMENTS.....
12.2 THE PAYEE MODEL.....
12.2.1 Payee Identifiers.....
12.2.2 Payee Lists.....
12.2.3 Standard Payee Lists.....
12.2.4 Summary – Identifying Payees.....
12.3 IDENTIFIERS USED IN PAYMENT TRANSACTIONS.....
12.4 THE PAYMENT LIFE CYCLE.....
12.4.1 Payment Creation.....
12.4.2 Payment Modification.....
12.4.3 Payment Status Inquiry.....
12.4.4 Payment Cancellation.....
12.4.5 Delayed Payee Matching.....
12.5 COMMON PAYMENTS AGGREGATES.....
12.5.1 Payments Account Information <BPACCTINFO>.....
12.5.2 Payment Information <PMTINFO>.....
12.6 PAYMENTS FUNCTIONS.....
12.6.1 Payment Creation.....
12.6.2 Payment Modification.....
12.6.3 Payment Cancellation.....
12.6.4 Payment Status Inquiry.....
12.7 RECURRING PAYMENTS.....
12.7.1 Creating a Recurring Payment.....
12.7.2 Recurring Payment Modification.....
12.7.3 Recurring Payment Cancellation.....
12.8 PAYMENT MAIL.....
12.8.1 Payment Mail Request and Response.....
12.8.2 Payment Mail Synchronization.....
12.9 PAYEE LISTS.....
12.9.1 Adding a Payee to the Payee List.....
12.9.2 Payee Modification.....
12.9.3 Payee Deletion.....
12.9.4 Payee List Synchronization.....
12.10 DATA SYNCHRONIZATION FOR PAYMENTS.....
12.10.1 Payment Synchronization.....
12.10.2 Recurring Payment Synchronization.....
12.10.3 Discussion.....
12.11 MESSAGE SETS AND PROFILE.....
12.11.1 Bill Pay Message Sets and Messages.....
12.11.2 Bill Pay Message Set Profile <BILLPAYMSGSET>.....
12.12 EXAMPLES.....
12.12.1 Scheduling a Payment.....
12.12.2 Modifying a Payment.....
12.12.3 Canceling a Payment.....
12.12.4 Updating Payment Status.....
12.12.5 Scheduling a Recurring Payment.....
12.12.6 Modifying a Recurring Payment.....
12.12.7 Canceling a Recurring Payment.....
12.12.8 Adding a Payee to the Payee List.....
12.12.9 Synchronizing Scheduled Payments.....

13. INVESTMENTS

13.1 Types of Response Information
13.2 Sub-Accounts
13.3 Units, Precision, and Signs
13.3.1 Units
13.3.2 Precision
13.3.3 Signs
13.4 Bank and Investment Transactions
13.5 Money Market Funds
13.5.1 Separate Account at the Financial Institution
13.5.2 Sweep Account Within an Investment Account
13.5.3 Position Within an Investment Account
13.6 Investment Accounts
13.6.1 Specifying the Investment Account <INVACCTFROM>
13.6.2 Investment Account Information <INVACCTINFO>
13.6.3 Brokerage, Mutual Fund, and 401K Accounts
13.7 Investment Message Sets and Profile
13.7.1 Investment Statement Download
13.7.2 Security Information
13.8 Investment Securities
13.8.1 Security Identification <SECID>
13.8.2 Security List Request
13.8.3 Security List Response
13.8.4 Security List <SECLIST>
13.8.5 Securities Information
13.9 Investment Statement Download
13.9.1 Investment Statement Request
13.9.2 Investment Statement Response
13.10 Investment E-Mail
13.10.1 Investment E-Mail Request and Response
13.10.2 Investment E-Mail Synchronization
13.11 Complete Example

12Payments

This section describes the Payments portion of Open Financial Exchange. Open Financial Exchange Payments consists of a set of functions for scheduling and maintaining payment transactions, and for synchronizing with the server to obtain an accurate status of all recent and scheduled transactions.

Clients use payment requests to schedule payments and to modify or delete payments if necessary. Open Financial Exchange also supports business payments, as described in section 12.1.

The recurring payments function allows the client to schedule automatic generation of a series of recurring payments by means of a single request. As with individual payments, the client can modify or delete these requests.

The payments function incorporates the synchronization features of Open Financial Exchange, allowing multiple client applications to synchronize with the server to obtain the current status of all payment transactions known to the server.

In many international environments, payments are performed using intrabank funds transfers. Open Financial Exchange Payments supports this by allowing a payee to be designated as a destination bank account. Servers can implement these messages as transfers where appropriate.

12.1Consumer and Business Payments

Open Financial Exchange Payments is designed to support both consumer and business payments. Businesses have additional requirements for payments. In particular, there is a need to include itemized instructions that specify how a payment should be disbursed across multiple invoices and/or line items. Open Financial Exchange supports this requirement through the inclusion of the <EXTDPMT> aggregate within payment requests. The Payment Modification Request <PMTMODRQ> also supports changes to <EXTDPMT> data.

12.2The Payee Model

The payee model in Open Financial Exchange is designed to provide support for both “pay-some” and “pay-any” payment systems. “Pay-some” systems are those that restrict users to only make payments to payees that appear on an approved list. Such payees are often referred to as “standard payees,” or “standard merchants.” These are generally larger corporations that receive high volumes of payments, such as telephone companies or power utilities. In contrast, **“**pay-any” systems allow payments to any payee for which the user provides accurate billing information. These systems often also include a list of standard payees.

12.2.1Payee Identifiers

Open Financial Exchange is designed to be flexible in the requirements for payee identifiers. It supports systems where all, some, or no payees are assigned a payee ID. In addition, it enables the server to assign an ID to a payee that was previously being paid by billing address.

You must implement the scope of payee such that the ID is at least global across the user's set of payments-enabled accounts with the payments provider. For example, if the user has both a checking and a money market account enabled for payments with the payments provider, then a payee ID obtained for a payment made from one of these accounts should identify the same payee if used for a payment drawn on the other account. This simplifies client support for allowing a user to choose from which account to make a payment.

Open Financial Exchange requires payee identifiers to have a one-to-one relationship with the corresponding PAYEE information. In other words, different payee IDs must also differ in their corresponding payee billing description or payee name (NAME). Similarly, a payee ID must be independent of a user's account number with the payee. However, the payment system is free to use the user's account number in combination with the payee ID to determine the routing of a payment. These rules are intended to simplify the payee model for the user, insuring that different payee IDs will have discernibly different descriptions associated with them. They also insure that the user will not be required to maintain multiple payee entries for a payee with which the user holds multiple accounts.

Open Financial Exchange includes a tag for indicating the scope of a payee ID returned from the server. This allows clients to adapt by expanding or restricting their functionality depending on the scope of payee IDs it encounters.

A payee list for each user, maintained on the server, allows the server to manage the identifiers assigned to a user's payees. This functionality is described in the next section.

12.2.2Payee Lists

Open Financial Exchange specifies that a server-hosted payee list is maintained for each payments user. This list contains the payees that a user has paid through the payment system, or has set up to pay. Updates to this list are available through the synchronization mechanism. This insures that multiple clients have access to the full list of payees the user has configured. It is only necessary to enter each payee once.

Some payment systems require a first time setup before using a payee. This can occur externally to the client and server software, for example by filling out a paper form or telephoning the bank. In this case, payee list synchronization provides a way for the payee to become accessible to the client software when the FI completes the setup.

The list can contain payees with or without payee IDs. An important function of the payee list is to communicate payee changes from the server to the client. This includes changes in processing date parameters and conversion of a payee to a standard payee.

Once added to the list, the payment system makes payments by the payee list ID. This makes it clear to a client when the user is adding to a payee list, and when he or she modifies an existing payee on the list.

Although the messages make it clear whether a client is trying to add a new payee, a careful server will check for exact matches on payee adds and not create new payee list entries unnecessarily.

"Pay-any" systems can perform background processing that matches billing addresses with standard payees. When this occurs the server can update the relevant payee lists, and update the clients when they synchronize with the modified list data.

Each payee entry in the list can also include a list of the user's accounts with that payee. This further reduces the data entry required by a user to make a payment, and facilitates the implementation of lightweight clients.

12.2.3Standard Payee Lists

Many payment systems maintain a list of payees that receive payments from a large number of users. Payments to these payees are usually consolidated into a few electronic funds transfers or are mailed in large batches to the payee. Payees that receive this special processing are generally referred to as standard payees. In a "pay-some" system, all the approved payees can be considered to be standard payees. When a user pays a standard payee, there might be different processing lead-times used to calculate the payment and/or processing date of a payment.

When a payment system includes a standard payee list, it might be desirable to present the list to the user, who can then select payees he or she wants to pay. Unfortunately, it is cumbersome to provide this functionality in the client software due to the potential size of this list, which makes it problematic to keep updated and to present to the user. While the list can contain thousands of payee entries, a user will typically need less than ten or twenty entries from the list. It can also be difficult for a user to choose the correct payee entry when the list contains a number of similarly named payees.

Therefore, Open Financial Exchange does not provide a mechanism for delivering these lists to the client. However, there are several ways that an external presentation of such a list can be integrated into the client or server. For example, a payment provider's Web site could present a search engine that assists the user to locate the correct payee. Once identified, the payees can either be imported into the client, or inserted into the user's payee list on the server. In the latter case, synchronizing the payee list will make the newly added payees visible to the client.

12.2.4Summary – Identifying Payees

Clients can identify a payee in one of four ways:

- Name & address, by means of <PAYEE> – should be done only once for each payee. Thereafter, clients should use the assigned <PAYEELSTID> or <PAYEEID>. If the clients sends both <PAYEE> and <PAYEELSTID>, it is an implicit payee modification request. If a client sends just <PAYEE>, it is an implicit payee add request. Duplicate payee list entries can occur if clients are not careful to send the payee list ID in subsequent requests.
- Destination bank account <BANKACCTTO> – should be done only once for each payee. Thereafter, clients should use the assigned <PAYEELSTID> or <PAYEEID>, as with name and address payees. The <PAYEE> aggregate is required to provide name and address information as a backup to account transfers.
- Payee list ID <PAYEELSTID> – after a payee has been added to the list, for payees that have not been assigned a standard payee ID <PAYEEID>.
- Standard payee ID <PAYEEID> – for any payee that has been assigned a standard payee ID. This could happen before a closed system makes any payments, or anytime after the server has notified the client that a payee has a standard payee ID.

12.3Identifiers Used in Payment Transactions

Payment transactions use four types of identifiers. It is important to understand the purpose, scope, and life span of these identifiers.

The client-to-request messages assign the Transaction Universal Identifier (TRNUID). Its purpose is to allow the client to easily match responses from the server to their corresponding requests. A given transaction ID is used only for a client request and the corresponding server response.

The Server Identifier (SRVRTID or RECSRVRTID) is assigned by the server to a payment "object," which can either be a payment or a recurring payment model (in which case it is named RECSRVRTID). Both the client and server use the ID thereafter to refer to the payment or model in any transactions that operate on them. For example, the SRVRTID is used to identify a payment in a request to modify or cancel it. The SRVRTID is valid for the life span of the payment within the payment system. Similarly the RECSRVRTID is valid as long as the associated model exists, that is until the model generates all payments, or the model is canceled. Once a server processes a payment or a model generates all its required payments, the associated SRVRTID (or RECSRVRTID) is no longer known to the server. Note that the payment system might continue to maintain knowledge of a payment SRVRTID or model RECSRVRTID for some specified period after it completes processing. This allows clients to access the "completed" status of these operations.

A payment system can assign the Payee Identifier (PAYEEID) to a payee. There is no requirement that all or any payees are assigned a PAYEEID. The usage of this identifier will vary by payment system. For example, in "pay-some" systems usually every payee has a payee ID with a scope that is known globally, while in "pay-any" systems there might only be PAYEEIDs assigned to standard payees. When a payee has an assigned PAYEEID, the life span of the ID will depend on its scope. If the scope is global, such as for payees in some "pay-some" systems or those with standard payees, then the PAYEEID is expected to be valid as long as that payee is identifiable by ID. If the payee ID is user-specific in scope, then the PAYEEID is valid as long as the payee appears in the user's server-hosted payee list.

The Payee List Identifier (PAYEELSTID) is assigned by the server to each entry in a user's server-hosted payee list. The need for this identifier is to support the variety of payee models employed in various payment systems. As discussed above, some payment systems assign a payee identifier (PAYEEID) to every payee (this is particularly the case with pay-some systems); others assign PAYEEIDs only to standard payees. There are also systems that cannot map a payee billing address to a PAYEEID in real time. Also, there are systems that can convert a payee from a standard payee with an assigned PAYEEID to one that is identified only by billing address. Therefore, systems employ the PAYEELSTID to insure that, in systems where payees will not always have a PAYEEID, there is another identifier that can be used to reference every payee. This insures that a client can correctly link payments to their payees. The PAYEELSTID must be valid as long as the user's payee list includes the payee it identifies, even if the server subsequently assigns a PAYEEID to the payee.

12.4The Payment Life Cycle

12.4.1Payment Creation

The client formulates a PMTRQ that includes the payee, the date, and the amount of the payment. If supported by the user's payment system, the billing address can specify the payee.

The server will look up the payee in the user's payee list. If it is not already in the table, the server will add it and issue a payee list identifier (PAYEELSTID). This form of payment request performs an implicit Payee Request (PAYEERQ), which is equivalent to explicitly adding the payee (by means of a PAYEERQ), prior to issuing the PMTRQ. It has the advantage of being atomic. If the payment request fails, the payee is not added to the user's payee list. Conversely the payment request will fail if the payee information is invalid.

The server responds to the PMTRQ with a Payment Response (PMTRS). Some servers will not be able to immediately return a payee ID at this point, or might not issue payee IDs for all payees. Therefore the PAYEELSTID contained in the response functions as the linkage between the payee and the payment. Payment systems use the SRVRTID returned in the PMTRS to identify the payment for the length of its instantiation on the payment system.

12.4.2Payment Modification

Between the time the client schedules a payment and the time the server processes the payment, the client can request changes to the parameters of that payment. For example, the amount or date of the payment can be modified. The system uses the Payment Modification Request (PMTMODRQ) for this purpose, where the SRVRTID from the PMTRS identifies the targeted payment. The user request must specify the full contents of the payment request, including both modified and unmodified data.

Full-featured servers will use PMTMODRS messages, conveyed to the client during synchronization (PMTSYNCRS), to inform the client about changes in the state of the client that occur due to server processing. This would include reporting the date the server actually processed a payment, or it failed due to insufficient funds. Servers that are unable to generate PMTMODRS responses for this purpose must support the PMTINQRQ message described below.

12.4.3Payment Status Inquiry

As a scheduled payment progresses through its "life-cycle" on the server, the processing status changes accordingly from "scheduled to be processed" to "was processed" or "failed processing." A processing date is associated with these states. The preferred method for providing updated processing status to a client is by use of server-generated Payment Modification messages (PMTMODRS), as discussed above. However it is possible that less full-featured servers might have difficulty in implementing this form of notification. In this case, Open Financial Exchange requires such servers to implement the Payment Status Inquiry message (PMTINQRQ), which provides an interface for the client to explicitly request the processing status of individual payments.

12.4.4Payment Cancellation

In the interval between successful processing of a PMTRQ and the actual processing of the payment, the client can cancel the payment by issuing a Payment Cancellation Request (PMTCANCRQ). The SRVRTID value returned in PMTRS identifies the payment.

When a payment system cancels a payment, servers can generate a PMTCANCRS. This might occur if the user requests payment cancellation by way of a telephone call to customer support or through an e-mail message. The client will receive this response when performing a synchronization (PMTSYNCRQ/PMTSYNCRS).

12.4.5Delayed Payee Matching

Payment systems that allow payment by payee billing address often perform a matching operation to determine if the payee is a standard payee. If this matching occurs in the processing of a PMTRQ, and the server recognizes the payee as a standard one, then the server returns the payee ID and payment parameters in the EXTDPAYEE aggregate of the PMTRS. However some payment systems will not be able to perform "payee matching" at this point in processing. In this case, the server sends updated payee information to the client by using PAYEESYNRQ to synchronize the payee list. The client can link payee information in the PAYEETRNRS messages to payments with matching PAYEELSTID identifiers.

12.5Common Payments Aggregates

This section documents several aggregates used throughout the Payments portion of the Open Financial Exchange specification.

12.5.1Payments Account Information <BPACCTINFO>

Open Financial Exchange uses the payments 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. In Open Financial Exchange, Banking and Payments share the BANKACCTFROM aggregate to identify a specific account. For more information, see section 11.3.1.

Tag	*Description*
<BPACCTINFO>	Payments-account-information aggregate
<BANKACCTFROM>	Bank-account-from aggregate
</BANKACCTFROM>	
<SVCSTATUS>	Status of the account AVAIL = Available, but not yet requested PEND = Requested, but not yet available ACTIVE = In use
</BPACCTINFO>	

12.5.2Payment Information <PMTINFO>

The Payment Information aggregate is used to specify detailed payment information. It is used for both single payments and recurring payments. Clients must send the <PAYEELSTID> and <PAYEEID> if known. Clients send a <PAYEE> aggregate if this is an implicit payee add or modify. See section 12.2.4 on identifying payees, above. The <EXTDPMT> aggregate (see section 12.5.2.2) allows the inclusion of disbursement instructions to be printed with the payment. This aggregate is optional.

Tag	*Description*
<PMTINFO>	
<BANKACCTFROM>	Account-from aggregate, see section 11.3.1
</BANKACCTFROM>	
<TRNAMT>	Payment amount, *amount*
<PAYEEID>	Server payee identifier (required if assigned). Either <PAYEEID> or <PAYEE> can be sent, but not both. *A-12*
Specify payee; either <PAYEEID> or <PAYEE>.	
<PAYEEID>	Payee identifier, *A-12*
<PAYEE>	Complete payee billing information, see section 12.5.2.1. Either <PAYEEID> or <PAYEE> can be sent, but not both.
</PAYEE>	
<PAYEELSTID>	Payee list ID (required if assigned), *A-12*
<BANKACCTTO>	Destination account information, for systems that pay by transfers (<PAYEE> also required)
</BANKACCTTO>	
<EXTDPMT>	Extended Payment aggregate, see section 12.5.2.2, optional
</EXTDPMT>	
<PAYACCT>	User account number with the payee, *A-32*
<DTDUE>	Payment due date or the date by which payment must be received by payee, *datetime*
<MEMO>	Memo from user to payee, *memo*
<BILLREFINFO>	Biller-supplied reference information when paying a bill, if available, *A-80*
</PMTINFO>	

12.5.2.1Payee <PAYEE>

<PAYEE> specifies a complete billing address for a payee.

Tag	*Description*
<PAYEE>	
<NAME>	Name of payee. *A-32*
<ADDR1>	Payee's address line 1, *A-32*
<ADDR2>	Payee's address line 2, *A-32*
<ADDR3>	Payee's address line 3, *A-32*
<CITY>	Payee's city, *A-32*
<STATE>	Payee's state, *A-5*
<POSTALCODE>	Payee's zip code, *A-11*
<COUNTRY>	Payee's country; 3-letter country code from ISO/DIS-3166, *A-3*
<PHONE>	Payee's telephone number, *A-32*
</PAYEE>	

12.5.2.2Extended Payment <EXTDPMT>

The Extended Payment aggregate provides the payee with information for applying a payment across multiple invoices. It is structured to allow for electronic processing of the invoice date, and allows multiple invoices, as well as multiple line items per invoice, to be specified.

In this case, EXTDPMT can specify a block of free text to be transmitted with the payment, by using the EXTDPMTDSC instead of the EXTDPMTINV element.

All amount fields (INVAMT, DSCAMT, ADJAMT, LITMAMT) can be negative.

Tag	*Description*
<EXTDPMT>	Extended Payment aggregate
<EXTDPMTFOR>	INDIVIDUAL or BUSINESS. Indicates whether the payment is for an individual or business account. This allows the payment processor to remit payments to the appropriate address for consumers or businesses.
<EXTDPMTCHK>	Check number to use for this payment. Overrides "next check in range." *N-10*
Payment description. Either or both of the following aggregates: <EXTDPMTDSC> or <EXTDPMTINV>.	
<EXTDPMTDSC>	Free text to communicate with the payment, *A-255*
<EXTDPMTINV>	Invoices aggregate
<INVOICE>	Start tag for the invoice aggregate. There can be one or more invoices per payment request.
<INVNO>	Invoice number associated with the payment. *A-32*
<INVTOTALAMT>	This value represents the total invoice amount, *amount*
<INVPAIDAMT>	This value represents the amount of the invoice being paid, *amount*
<INVDATE>	Date to apply the invoice, *datetime*
<INVDESC>	Invoice description, *A-80*
<DISCOUNT>	Discount aggregate; only one discount aggregate per invoice
<DSCRATE>	Discount rate, optional if <DSCAMT> tag is used, *rate*
<DSCAMT>	Discount amount, optional if <DSCRATE> tag is used, *amount*
<DSCDATE>	Date to apply the discount, *datetime*
<DSCDESC>	Discount description, *A-80*
</DISCOUNT>	
<ADJUSTMENT>	Adjustment aggregate; only one adjustment aggregate per invoice
<ADJNO>	Adjustment number associated with the payment, *A-32*
<ADJDESC>	Adjustment description, *A-80*
<ADJAMT>	Amount of the adjustment, *amount*
<ADJDATE>	Date of adjustment, *datetime*
</ADJUSTMENT>	
<LINEITEM>	Line item aggregate; there can be multiple line items per invoice
<LITMAMT>	Amount of the line item, *amount*
<LITMDESC>	Line item description, *A-80*
</LINEITEM>	
</INVOICE>	
</EXTDPMTINV>	
</EXTDPMT>	

12.5.2.3Extended Payee <EXTDPAYEE>

The Extended Payee aggregate communicates a payee identifier to the client. It also contains the processing day parameters for a payee. It can be sent to the client for any payee whose processing day parameters are different from the processor's default values, even for payees with no IDs.

Tag	*Description*
<EXTDPAYEE>	Extended-payee aggregate
<PAYEEID>	Server-assigned payee ID, *A-12*
<IDSCOPE>	Scope of the payee ID; one of (GLOBAL, USER), where GLOBAL = payee ID valid across the entire payment system USER = payee ID valid with all FI accounts set up for the user's payments account
<NAME>	Standard payee name, *A-32*
<DAYSTOPAY>	Minimum number of business days needed to process, *N-3*
</EXTDPAYEE>	

12.5.2.4Payment Processing Status <PMTPRCSTS>

The Payment Processing Status aggregate contains the current processing status for a payment. The interpretation of the date value depends on the value of <PMTPRCCODE>.

Tag	*Description*
<PMTPRCSTS>	
<PMTPRCCODE>	See table 12.6.2.1
<DTPMTPRC>	Payment processing date; interpretation depends on <PMTPRCCODE>, *datetime*
</PMTPRCSTS>	Ending tag for payment processing status

12.6Payments Functions

Payments functions allow a client to create a Payment Request to pay a bill on a specified date. The Payment Request identifies the payee and the amount to pay.

Client Sends	*Server Responds*
Account information	
Payment date	
Amount	
Payee address, list ID, transfer acct, or standard ID	
	Payment status
	Check number
	Server-assigned ID

From the time the client issues a Payment Request until it is paid, the client can modify the transaction through the Payment Modification Request, <PMTMODRQ>; see section 12.4.2. This request allows payment parameters such as the payment date and payment amount to be changed.

Client Sends	*Server Responds*
Account information	
Server-assigned ID	
Information to change:	
Payment date,	
Amount,...	
	Acknowledgment or Error

The client can cancel a Payment Request with a Payment Cancellation Request, <PMTCANCRQ>; see section 12.4.4.

Client Sends	*Server Responds*
Account information	
Server-assigned ID	
	Acknowledgment or Error

12.6.1Payment Creation

A Payment Request is used to schedule an electronic payment. The server responds with a Payment Response. Separate transactions are provided for modifying and canceling a Payment Request.

12.6.1.1Payment Request <PMTRQ>

The <PMTRQ> request must appear within a <PMTTRNRQ> transaction wrapper.

Tag	*Description*
<PMTRQ>	Payment-request aggregate
<PMTINFO>	Payment Information aggregate, see section 12.5.2
</PMTINFO>	
</PMTRQ>	

12.6.1.2Payment Response <PMTRS>

The server sends a Payment Response in response to a Payment Request. The processing status code for a new payment is normally WILLPROCESSON, but in the case of synchronization it can return other status codes.

> ***NOTE:*** If the <PMTRQ> created a new payee, the server must create and store a payee response that would be available for a payee synchronization request. The server need only send the <PMTRS> as a response to the <PMTRQ>, but the ability to send the payee synchronization response is required in case another client logs on and requests only a payee synchronization.

The <PMTRS> response must appear within a <PMTTRNRS> transaction wrapper.

Tag	*Description*
<PMTRS>	Payment-response aggregate
<SRVRTID>	ID assigned by the server to the payment being created, *SRVRTID*
<PAYEELSTID>	Server-assigned payee list record ID for this payee, *A-12*
<CURDEF>	Default currency for the Recurring Payment Response, *currsymbol*
<PMTINFO>	Payment Information aggregate, see section 12.5.2
</PMTINFO>	
<EXTDPAYEE>	Standard payee information if payee is a standard payee, or payee has non-default processing day parameters; see section 12.5.2.3
</EXTDPAYEE>	
<CHECKNUM>	Check number, *A-12*
<PMTPRCSTS>	Payment processing status
</PMTPRCSTS>	
<RECSRVRTID>	References the payment if it was generated by a recurring payment, *SRVRTID*
</PMTRS>	

12.6.1.3Status 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)
2019	Duplicate request (ERROR)
10501	Invalid payee (ERROR)
10502	Invalid payee address (ERROR)
10503	Invalid payee account number (ERROR)
10510	Invalid payee ID (ERROR)
10511	Invalid payee city (ERROR)
10512	Invalid payee state (ERROR)
10513	Invalid payee postal code (ERROR)
10517	Invalid payee name (ERROR)
10519	Invalid payee list ID (ERROR)

12.6.1.4Discussion

Once the server has assigned a payee identifier to the payee, it includes the <EXTDPAYEE> in any <PMTRS> for that transaction. If the <EXTDPAYEE> aggregate is present in the Payment Response (<PMTRS>), the client records the standard payee information for use in future payments to the same payee.

When a payment is made using the <PAYEE> aggregate, and no <PAYEELSTID> is present, the payee is implicitly added to the payee list. This is therefore equivalent to first transmitting a <PAYEERQ> for the payee. For payment systems that can immediately return payee IDs, it is preferable to use the single <PMTRQ> message to both add the payee and create the payment. If either operation fails, the server will not complete the other. If <PAYEELSTID> and <PAYEE> are both included, it is equivalent to sending a <PAYEEMODRQ> first.

The <PMTRS> response will include the <EXTDPAYEE> aggregate if the processor has assigned a payee ID to the payee specified in the payment. It will also appear in the response when the payee has no assigned ID, but has processing day parameters that different from the processor's defaults for these values. This might occur, for instance, if the processor notes that the zip code of the payee indicates a certain proximity to the payer, and therefore wishes to offer a shorter <DAYSTOPAY> value.

12.6.2Payment Modification

The Payment Modification Request allows a client to modify a previously scheduled payment. When modifying a payment, the client must specify all of the tags within the <PMTINFO> aggregate that were specified during the payment creation, not just the tags that it wants to modify. The ability to modify some tag values might not be supported by all servers.

12.6.2.1Payment Processing Status Values <PMTPRCCODE>

Value	*Description*
WILLPROCESSON	Will be processed on <DTPMTPRC>
PROCESSEDON	Was processed for payment on <DTPMTPRC>
NOFUNDSON	Funds not available to make payment on <DTPMTPRC>
FAILEDON	Unable to make payment for unspecified reasons on <DTPMTPRC>
CANCELEDON	User canceled payment on <DTPMTPRC>

12.6.2.2Payment Modification Request <PMTMODRQ>

The client sends a Payment Modification Request to request modification of a payment. The client must provide the full <PMTINFO> including both changed and unchanged values.

The client is free to modify any data in PMTINFO, excluding the payee name (the NAME element of PAYEE) and payee ID (element PAYEEID).

The <PMTMODRQ> request must appear within a <PMTTRNRQ> transaction wrapper.

Tag	*Description*
<PMTMODRQ>	Modification-request this references
<SRVRTID>	ID assigned by the server to the payment being modified, *SRVRTID*
<PMTINFO>	Payment Information aggregate, see section 12.5.2
</PMTINFO>	
</PMTMODRQ>	

12.6.2.3Payment Modification Response <PMTMODRS>

The server sends a Payment Modification Response in response to a Payment Modification Request.

> ***NOTE:*** If the <PMTMODRQ> created a new payee, the server must create and store a payee response that would be available for a payee synchronization request. The server need only send the <PMTMODRS> as a response to the <PMTMODRQ>, but the ability to send the payee synchronization response is required in case another client logs on and requests only a payee synchronization.

The <PMTMODRS> response must appear within a <PMTTRNRS> transaction wrapper.

Tag	*Description*
<PMTMODRS>	Payment-modification-response this references
<SRVRTID>	ID assigned by the server to the payment being modified, *SRVRTID*
<PMTINFO>	Payment Information aggregate, see section 12.5.2
</PMTINFO>	
<PMTPRCSTS>	Payment processing status, see section 12.5.2.4
</PMTPRCSTS>	
</PMTMODRS>	

12.6.2.4Status 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)
10501	Invalid payee (ERROR)
10502	Invalid payee address (ERROR)
10503	Invalid payee account number (ERROR)
10505	Cannot modify element (ERROR)
10510	Invalid payee ID (ERROR)
10511	Invalid payee city (ERROR)
10512	Invalid payee state (ERROR)
10513	Invalid payee postal code (ERROR)
10517	Invalid payee name (ERROR)
10519	Invalid payee list ID (ERROR)

12.6.2.5Discussion

Servers can initiate <PMTMODRS> messages to communicate changes in the processing status of a payment as it moves through the payment system. This mechanism allows a client to capture the updated status of payments every time it synchronizes.

12.6.3Payment Cancellation

The Payment Cancellation Request allows a client to cancel a previously scheduled payment created with a Payment Request (<PMTRQ> in section 12.6.1.1).

12.6.3.1Request <PMTCANCRQ>

The client sends a Payment Cancellation to cancel a scheduled payment request.

The <PMTCANRQ> request must appear within a <PMTTRNRQ> transaction wrapper.

Tag	*Description*
<PMTCANCRQ>	Cancellation-request this references
<SRVRTID>	ID assigned by the server to the payment being canceled, *SRVRTID*
</PMTCANCRQ>	

12.6.3.2Response <PMTCANCRS>

The server sends a Payment Cancellation Response in response to a Payment Cancellation Request.

The <PMTCANRS> response must appear within a <PMTTRNRS> transaction wrapper.

Tag	*Description*
<PMTCANCRS>	Cancellation-response this references
<SRVRTID>	ID assigned by the server to the payment being canceled, *SRVRTID*
</PMTCANCRS>	

12.6.3.3Status 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)

12.6.4Payment Status Inquiry

The Payment Status Inquiry Request allows a client to obtain the current processing status of a payment from the server.

12.6.4.1Request <PMTINQRQ>

The client sends a Payment Status Inquiry Request to obtain the current processing status of a payment.

The <PMTINQRQ> request must appear within a <PMTINQTRNRQ> transaction wrapper.

Tag	*Description*
<PMTINQRQ>	Payment-status-inquiry-request aggregate
<SRVRTID>	ID assigned by the server to the payment being queried, *SRVRTID*
</PMTINQRQ>	

12.6.4.2Response <PMTINQRS>

The server sends a Payment Status Inquiry Response in response to a Payment Status Inquiry Request.

The <PMTINQRS> response must appear within a <PMTINQTRNRS> transaction wrapper.

Tag	*Description*
<PMTINQRS>	Payment-status-inquiry-response aggregate
<SRVRTID>	ID assigned by the server to the payment being queried, *SRVRTID*
<PMTPRCSTS>	Payment processing status
</PMTPRCSTS>	
<CHECKNUM>	Check number assigned by the server to this payment, *A-12*
</PMTINQRS>	

12.6.4.3Status Codes

Code	*Meaning*
0	Success (INFO)
2000	General error (ERROR)
2018	Unknown server ID (ERROR)
2019	Duplicate request (ERROR)

12.7Recurring Payments

Recurring payments are used when a payment is to be made repeatedly at some known interval. Setting up a recurring payment is similar to creating an individual payment, but with additional information about the frequency and number of payments. After a recurring payment is created, the server will generate payments transactions when there are a specified number of days remaining until the next projected payment is due (usually 30 days). The client will be made aware of any generated payment transactions through the synchronization process. Chapters 10 and 11 provide additional details on models and recurring transactions, and defines the recurring transaction aggregates.

The table below lists the functional elements for creating a recurring payment:

Client Sends	*Server Responds*
Account information	
Payment frequency	
Number of payments	
Payment date	
Amount	
Payee address, list ID or payee ID	
	Standard payee information
	Server-assigned ID

The table below lists the functional elements for modifying a recurring payment:

Client Sends	*Server Responds*
Account information	
Server-assigned ID	
Information to change:	
Payment frequency,	
Number of payments,	
Payment date,	
Amount,...	
	Acknowledgment or Error

The table below lists the functional elements for canceling a recurring payment:

Client Sends	*Server Responds*
Account information	
Server-assigned ID	
	Acknowledgment or Error

12.7.1 Creating a Recurring Payment

Use a Recurring Payment Request to set up a recurring electronic payment. The user can specify the frequency and duration of the payments using the Recurring Instructions aggregate <RECURRINST>. The <PMTINFO> aggregate (see section 12.5.2) specifies the payment information.

12.7.1.1 Request <RECPMTRQ>

The <RECPMTRQ> request must appear within a <RECPMTTRNRQ> transaction wrapper.

Tag	*Description*
<RECPMTRQ>	Recurring-payment-request aggregate
<RECURRINST>	Recurring Instructions aggregate, see section 10.2.
</RECURRINST>	
<PMTINFO>	Payment-Information aggregate, see section 12.5.2
</PMTINFO>	
<INITIALAMT>	Amount of the initial payment, if different than the following payments, *amount*
<FINALAMT>	Amount of the final payment, if different than the preceding payments, *amount*
</RECPMTRQ>	

12.7.1.2Response <RECPMTRS>

The server sends a Recurring Payment Response upon receipt of a Recurring Payment Request.

The <RECPMTRS> response must appear within a <RECPMTTRNRS> transaction wrapper.

Tag	*Description*
<RECPMTRS>	Recurring-payment-response aggregate
<RECSRVRTID>	Server-assigned ID for this transaction, *SRVRTID*
<PAYEELSTID>	Server-assigned record ID for this payee record, *A-12*
<CURDEF>	Default currency for the Recurring Payment Response, *currsymbol*
<RECURRINST>	Recurring-instructions aggregate, see section 10.2.
</RECURRINST>	
<PMTINFO>	Payment-information aggregate, see section 12.5.2
</PMTINFO>	
<INITIALAMT>	Amount of the initial payment, if different than the following payments, *amount*
<FINALAMT>	Amount of the final payment, if different than the preceding payments, *amount*
<EXTDPAYEE>	Extended payee information, see section 12.5.2.3.
</EXTDPAYEE>	
</RECPMTRS>	

12.7.1.3Status 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)
2019	Duplicate request (ERROR)
10501	Invalid payee (ERROR)
10502	Invalid payee address (ERROR)
10503	Invalid payee account number (ERROR)
10508	Invalid frequency (ERROR)
10510	Invalid payee ID (ERROR)
10511	Invalid payee city (ERROR)
10512	Invalid payee state (ERROR)
10513	Invalid payee postal code (ERROR)
10517	Invalid payee name (ERROR)
10519	Invalid payee list ID (ERROR)

12.7.1.4Discussion

The <DTDUE> element of <PMTINFO> specifies when the first payment will be processed.

12.7.2Recurring Payment Modification

The client sends a Recurring Payment Modification Request to request modifications to a recurring payment previously created with a Recurring Payment Request. The payment frequency (<RECURRINST>), the payment parameters (<PMTINFO>), or both, can be changed.

12.7.2.1Request <RECPMTMODRQ>

The client sends a Recurring Payment Modification Request to request changes to a recurring payment model.

As with individual payments, the client is free to modify any data in PMTINFO, excluding the payee name (the NAME element of PAYEE) and payee ID (element PAYEEID).

The <RECPMTMODRQ> request must appear within a <RECPMTTRNRQ> transaction wrapper.

Tag	*Description*
<RECPMTMODRQ>	Modification-request aggregate
<RECSRVRTID>	ID assigned by the server to the payment being modified, *SRVRTID*
<RECURRINST>	Recurring Instructions aggregate, see section 10.2
</RECURRINST>	
<PMTINFO>	Payment Information aggregate, see section 12.5.2
</PMTINFO>	
<INITIALAMT>	Amount of the initial payment, if different than the following payments, *amount*
<FINALAMT>	Amount of the final payment, if different than the preceding payments, *amount*
<MODPENDING>	Modify pending flag If the client sets this flag, the server must modify pending and future payments. *Boolean*
</RECPMTMODRQ>	

12.7.2.2Response <RECPMTMODRS>

The server sends a Recurring Payment Modification Response in response to a Recurring Payment Modification Request.

The <RECPMTMODRS> response must appear within a <RECPMTTRNRS> transaction wrapper.

Tag	*Description*
<RECPMTMODRS>	Modification-response aggregate
<RECSRVRTID>	ID assigned by the server to the payment being modified, *SRVRTID*
<RECURRINST>	Recurring-Instructions aggregate, see section 10.2
</RECURRINST>	
<PMTINFO>	Payment-Information aggregate, see section 12.5.2
</PMTINFO>	
<INITIALAMT>	Amount of the initial payment, if different than the following payments, *amount*
<FINALAMT>	Amount of the final payment, if different than the preceding payments, *amount*
<MODPENDING>	Y if the client requested that the server modify pending and future payments. N if the client did not request that the server modify pending and future payments., *Boolean*
</RECPMTMODRS>	

12.7.2.3Status 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)
2019	Duplicate request (ERROR)
10501	Invalid payee (ERROR)
10502	Invalid payee address (ERROR)
10503	Invalid payee account number (ERROR)
10508	Invalid frequency (ERROR)
10511	Invalid payee city (ERROR)
10512	Invalid payee state (ERROR)
10513	Invalid payee postal code (ERROR)
10517	Invalid payee name (ERROR)
10518	Unknown model ID (ERROR)
10519	Invalid payee list ID (ERROR)

12.7.3Recurring Payment Cancellation

The client sends a Recurring Payment Cancellation Request to cancel a recurring payment previously created with a Recurring Payment Request.

12.7.3.1Request <RECPMTCANCRQ>

The <RECPMTCANCRQ> request must appear within a <RECPMTTRNRQ> transaction wrapper.

Tag	*Description*
<RECPMTCANCRQ>	Cancellation-request aggregate
<RECSRVRTID>	ID assigned by the server to the payment being canceled, *SRVRTID*
<CANPENDING>	Cancel pending flag If the client sets this flag, the server must cancel pending and future payments. *Boolean*
</RECPMTCANCRQ>	

12.7.3.2Response <RECPMTCANCRS>

The server sends a Recurring Payment Cancellation Response in response to a Recurring Payment Cancellation Request.

The <RECPMTCANCRS> response must appear within a <RECPMTTRNRS> transaction wrapper.

Tag	*Description*
<RECPMTCANCRS>	Modification-request aggregate
<RECSRVRTID>	ID assigned by the server to the payment being modified, *SRVRTID*
<CANPENDING>	Y if the client requested that the server cancel pending and future payments. N if the client did not request that the server cancel pending and future payments. *Boolean*
</RECPMTCANCRS>	

12.7.3.3Status Codes

Code	*Meaning*
0	Success
2000	General error (ERROR)
2019	Duplicate request (ERROR)
10518	Unknown model ID (ERROR)

12.8Payment Mail

Users can correspond by way of e-mail to resolve problems or ask questions about their payments accounts. This function makes use of the general Open Financial Exchange e-mail facility, which is described in Chapter 9.

12.8.1Payment Mail Request and Response

12.8.1.1Request <PMTMAILRQ>

The <PMTMAILRQ> allows a client to issue an e-mail to the payments processor. If the message refers to a specific payment, then both <SRVRTID> and <PMTINFO> are required to identify the payment to the processor.

The <PMTMAILRQ> request must appear within a <PMTMAILTRNRQ> transaction wrapper.

Tag	*Description*
<PMTMAILRQ>	Payment e-mail-request aggregate
<MAIL>	General e-mail aggregate
</MAIL>	
<SRVRTID>	Transaction ID of the payment that is the subject of the correspondence, *SRVRTID*
<PMTINFO>	Payment Information aggregate, see section 12.5.2
</PMTINFO>	
</PMTMAILRQ>	

12.8.1.2Response <PMTMAILRS>

The server sends <PMTMAILRS> in response to a Payment E-mail request.

The <PMTMAILRS> response must appear within a <PMTMAILTRNRS> transaction wrapper.

Tag	*Description*
<PMTMAILRS>	Payment e-mail-response aggregate
<MAIL>	General e-mail aggregate, see Chapter 9
</MAIL>	
<SRVRTID>	Transaction ID of the payment that is the subject of the correspondence, *SRVRTID*
<PMTINFO>	Payment Information aggregate, see section 12.5.2
</PMTINFO>	
</PMTMAILRS>	

12.8.1.3Status 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)
2018	Unknown server ID (ERROR)
2019	Duplicate request (ERROR)
16500	HTML not allowed (ERROR)
16501	Unknown mail To: (ERROR)

12.8.2Payment Mail Synchronization

Payment mail is subject to synchronization. The scope of the synchronization request is all of the accounts for which the user might have sent mail, not a specific account.

12.8.2.1Request <PMTMAILSYNCRQ>

Tag	*Description*
<PMTMAILSYNCRQ> 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*
<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*
<PMTMAILTRNRQ> </PMTMAILTRNRQ> **</PMTMAILSYNCRQ>**	Payment-mail transactions (0 or more)

12.8.2.2Response <PMTMAILSYNCRS>

Tag	*Description*
<PMTMAILSYNCRS>	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*
<PMTMAILTRNRS>	Payment-mail transactions (0 or more)
</PMTMAILTRNRS>	
</PMTMAILSYNCRS>	

12.9Payee Lists

Payments-system servers store lists of payees set up for payment by each user. Some systems require this before the user can issue a payment to a payee. In other payment systems, this feature enables the sharing of payee entry among multiple clients, and simplifies server payee maintenance.

A server-assigned payee list-entry ID identifies entries in the payee list. The following set of messages allows clients to obtain this list of payees. Users can add, modify, and delete individual entries in the list. The request for the list is synchronized, so that multiple clients can use the list.

Creating a payee:

Client Sends	*Server Responds*
Server-assigned payee identifier, or payee billing address	
User's account number with the payee	
	Payee address
	Standard payee information

Modifying a payee:

Client Sends	*Server Responds*
Server-assigned payee identifier	
User's account number with the payee	
Information to change:	
payee name	
address	
city	
state	
postal code	
phone number	
new payee account #	
	Extended payee information if payee is a standard payee or has non-default processing lead times
	Acknowledgment or Error

Deleting a payee:

Client Sends	*Server Responds*
Server-assigned payee identifier	
User's account number with the payee	
	Acknowledgment or Error

12.9.1Adding a Payee to the Payee List

The user can use the Payee Request to add a payee to the server payee list. The server responds with a Payee Response, which can contain a complete billing address for the payee, or if the payee is a standard payee, the lead time and payee name values.

12.9.1.1Payee Request <PAYEERQ>

The <PAYEERQ> requests the addition of a payee entry to the server's payee list. Note that the user can use a <PMTRQ> to simultaneously set up a payee. Open Financial Exchange does not require the client to send a <PAYEERQ> before making an initial <PMTRQ> to a payee.

The <PAYEERQ> request must appear within a <PAYEETRNRQ> transaction wrapper.

Tag	*Description*
<PAYEERQ>	Payee-request aggregate
<PAYEEID>	Server-assigned payee identifier. Either <PAYEEID> or <PAYEE> but not both may be sent. *A-12*
<PAYEE>	Complete payee billing information, see section 12.5.2.1. Either <PAYEEID> or <PAYEE> but not both may be sent
</PAYEE>	
<BANKACCTTO>	The destination bank account, specified in countries that pay using transfers. The <PAYEE> (above) must also be specified.
</BANKACCTTO>	
<PAYACCT>	User's account number(s) with the payee (0 or more), *A-32*
</PAYEERQ>	

12.9.1.2Payee Response <PAYEERS>

The server sends the Payee Response in response to a Payee Request. It contains the full billing information for the payee if it is not a standard payee. Otherwise, it contains the standard payee information, including lead time and payee name. If the server identifies the payee as having an assigned payee ID, then the server will include the EXTDPAYEE aggregate in the response.

If the response indicates that the payee does not have an assigned payee ID, the client should specify the full billing address (PAYEE) information in subsequent payment requests (PMTRQ) to the payee.

If the response indicates that the payee does have a payee ID, then the client should use the payee ID for making payments to that payee.

The <PAYEERS> response must appear within a <PAYEETRNRS> transaction wrapper.

Tag	*Description*
<PAYEERS>	Payee-response aggregate
<PAYEELSTID>	Server-assigned record ID for this payee record, *A-12*
<PAYEE>	Complete payee billing information, see section 12.5.2.1
</PAYEE>	
<BANKACCTTO>	The destination bank account, specified in countries that pay using transfers. The <PAYEE> (above) must also be specified.
</BANKACCTTO >	
<EXTDPAYEE>	Extended payee information, see section 12.5.2.3
</EXTDPAYEE>	
<PAYACCT>	User's account number(s) with the payee, *A-32*
</PAYEERS>	

12.9.1.3Status Codes

Code	*Meaning*
0	Success (INFO)
2000	General error (ERROR)
2001	Invalid account (ERROR)
2002	General account error (ERROR)
2009	Destination account not found (ERROR)
2010	Destination account closed (ERROR)
2011	Destination account not authorized (ERROR)
2019	Duplicate request (ERROR)
10501	Invalid payee (ERROR)
10502	Invalid payee address (ERROR)
10503	Invalid payee account number (ERROR)
10511	Invalid payee city (ERROR)
10512	Invalid payee state (ERROR)
10513	Invalid payee postal code (ERROR)

12.9.2Payee Modification

The Payee Modification Request allows the client to make changes to payee entries in the server's payee list.

12.9.2.1Request <PAYEEMODRQ>

The client sends the Payee Modification Request to request changes to an existing payee list entry. The <PAYEE> aggregate must specify the changed and unchanged payee information.

The <PAYEEMODRQ> request must appear within a <PAYEETRNRQ> transaction wrapper.

Tag	*Description*
<PAYEEMODRQ>	Modification-request aggregate
<PAYEELSTID>	Server-assigned record ID for this payee record, *A-12*
<PAYEE>	Payee information to modify
</PAYEE>	
<BANKACCTTO>	Destination account for countries that pay using transfers (<PAYEE> required)
</BANKACCTTO >	
<PAYACCT>	Payer account number(s) with the payee (0 or more), *A-32*
</PAYEEMODRQ>	

12.9.2.2Response <PAYEEMODRS>

The server returns a Payee Modification Response in reply to a Payee Modification Request.

When a server-initiated change occurs to the extended payee information for a payee (for example a change in the payee's lead time), the server can include this information in the <EXTDPAYEE> of the response.

If a server-initiated response indicates either that a payee now has a payee ID, or no longer has one, then the client should use the appropriate form of designating the payee in any future payment requests (PMTRQ) to that payee.

The <PAYEEMODRS> response must appear within a <PAYEETRNRS> transaction wrapper.

Tag	*Description*
<PAYEEMODRS>	Modification-response aggregate
<PAYEELSTID>	Server-assigned record ID for this payee record, *A-12*
<PAYEE>	Payee information that was modified
</PAYEE>	
<BANKACCTTO>	Destination account for countries that pay bills using transfers (<PAYEE> required as well)
</BANKACCTTO >	
<PAYACCT>	Payer's account number(s) with the payee (0 or more), *A-32*
<EXTDPAYEE>	Extended payee information, see section 12.5.2.3
</EXTDPAYEE>	
</PAYEEMODRS>	

12.9.2.3Status Codes

Code	*Meaning*
0	Success (INFO)
2000	General error (ERROR)
2001	Invalid account (ERROR)
2002	General account error (ERROR)
2009	Destination account not found (ERROR)
2010	Destination account closed (ERROR)
2011	Destination account not authorized (ERROR)
2019	Duplicate request (ERROR)
10501	Invalid payee (ERROR)
10502	Invalid payee address (ERROR)
10503	Invalid payee account number (ERROR)
10510	Invalid payee ID (ERROR)
10511	Invalid payee city (ERROR)
10512	Invalid payee state (ERROR)
10513	Invalid payee postal code (ERROR)
10515	Payee not modifiable by client (ERROR)

12.9.3Payee Deletion

The Payee Deletion Request allows a client to delete a payee entry from the server's list of the user's payees. The combination of <PAYEEID> and <PAYACCT> specifies entries.

12.9.3.1Request <PAYEEDELRQ>

The <PAYEEDELRQ> requests the deletion of a payee entry.

The <PAYEEDELRQ> request must appear within a <PAYEETRNRQ> transaction wrapper.

Tag	*Description*
<PAYEEDELRQ>	Deletion-request aggregate
<PAYEELSTID>	Server-assigned record ID for this payee record, *A-12*
</PAYEEDELRQ>	

12.9.3.2Response <PAYEEDELRS>

The server sends the Payee Deletion Response <PAYEEDELRS> in response to a Payee Deletion Request <PAYEEDELRQ>.

The <PAYEEDELRS> response must appear within a <PAYEETRNRS> transaction wrapper.

Tag	*Description*
<PAYEEDELRS>	Deletion-response aggregate
<PAYEELSTID>	Server-assigned record ID for this payee record, *A-12*
</PAYEEDELRS>	

12.9.3.3Status Codes

Code	*Meaning*
0	Success (INFO)
2000	General error (ERROR)
2019	Duplicate request (ERROR)
10519	Invalid payee list ID (ERROR)

12.9.4Payee List Synchronization

This set of messages allows clients to obtain a list of payees stored on the server that it has configured for use in payments. In a "pay some" system, users are sometimes required to explicitly configure the payees to whom the system will make payments. This can be done by means of a telephone call to the payments provider or through some other interface. The client can then use this message set to obtain the user's list of configured payees. In other systems, the payments provider can elect to store a list of all payees that the user has paid. This is a convenience to the client. It allows payees set up on one client to be accessible from a user's other clients. The support of this message set is optional, but can be required by "pay some" providers that rely on this functionality.

12.9.4.1Request <PAYEESYNCRQ>

Tag	*Description*
<PAYEESYNCRQ> Client synchronization option; <TOKEN>, <TOKENONLY>, or <REFRESH>	Payee-list-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*
<PAYEETRNRQ>	Payee transactions (0 or more)
</PAYEETRNRQ>	
</PAYEESYNCRQ>	

12.9.4.2Response <PAYEESYNCRS>

Tag	*Description*
<PAYEESYNCRS>	Payee-list-request 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*
<PAYEETRNRS>	Payee transactions (0 or more)
</PAYEETRNRS>	
</PAYEESYNCRS>	

12.9.4.3Status Codes

Code	*Meaning*
0	Success (INFO)
2000	General error (ERROR)

12.10Data Synchronization for Payments

Users of Open Financial Exchange Payments need to be able to obtain the current status of transactions previously sent to the server for processing. For example, once the user schedules a payment and the payment date has passed, the user might want to verify that the server made the payment as directed. Additionally, Open Financial Exchange allows for interactions with the server through multiple clients. This means, for example, that the user can perform some transactions from a home PC and others from an office computer with each session incorporating the activities performed on the other.

In order to accomplish these tasks, the client uses a synchronization scheme to insure that it has an accurate copy of the server data that is relevant to the client application. The intent of this scheme is to address three scenarios in which the client might lose synchronization with the server:

- A transaction has changed its state based on processing actions on the server. For example, a scheduled payment has passed its due date and has been paid or rejected.
- Transactions relevant to the client's application state have been added, deleted, or modified by a second client. For example, a user might enter or change transactions from more than one PC or application.
- A communications session between the client and server was interrupted or completed abnormally. As a result the client does not have responses from the server indicating that all the transactions were received and processed.

12.10.1Payment Synchronization

12.10.1.1Request <PMTSYNCRQ>

Tag	*Description*
<PMTSYNCRQ> 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>	Opening tag for account from aggregate, see section 11.3.1
</BANKACCTFROM>	
<PMTTRNRQ>	Payment transactions (0 or more)
</PMTTRNRQ>	
</PMTSYNCRQ>	

12.10.1.2Response <PMTSYNCRS>

Tag	*Description*
<PMTSYNCRS>	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>	Opening tag for account from aggregate, see section 11.3.1
</BANKACCTFROM>	
<PMTTRNRS>	Payment transactions (0 or more)
</PMTTRNRS>	
</PMTSYNCRS>	

12.10.2Recurring Payment Synchronization

12.10.2.1Request <RECPMTSYNCRQ>

Tag	*Description*
<RECPMTSYNCRQ> 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>	Opening tag for account from aggregate, see section 11.3.1
</BANKACCTFROM>	
<RECPMTTRNRQ>	Recurring-payment transactions (0 or more)
</RECPMTTRNRQ>	
</RECPMTSYNCRQ>	

12.10.2.2Response <RECPMTSYNCRS>

Tag	*Description*
<RECPMTSYNCRS>	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>	Opening tag for account from aggregate, see section 11.3.1
</BANKACCTFROM>	
<RECPMTTRNRS>	Recurring-payment transactions (0 or more)
</RECPMTTRNRS>	
</RECPMTSYNCRS>	

12.10.2.3Status 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)

12.10.3Discussion

This section describes specific synchronization processing for the Open Financial Exchange Payments functions. Chapter 6 provides a more extensive discussion of the Open Financial Exchange synchronization mechanism.

The client follows the steps below to perform a synchronization:

1. The client sends a <PMTSYNCRQ> and/or <RECPMTSYNCRQ> containing the token it has stored from its last successful synchronization (or the special initial token value).

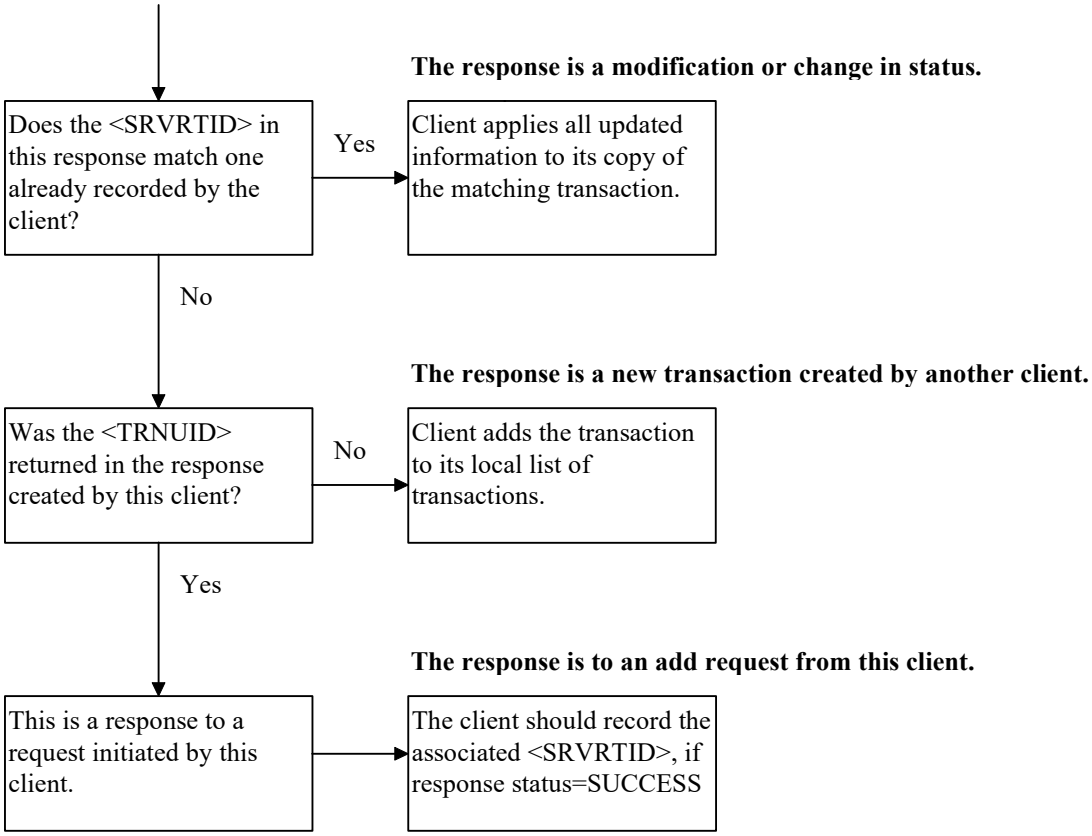
2. The client processes the <PMTSYNCRS> and/or <RECPMTSYNCRS> response from the server.

When the client has requested the server to add a transaction, a response that contains a <TRNUID> matching a transaction originally sent by the client—for which the client has not recorded an associated <SRVRTID>—is the normal scenario. This scenario could also occur if the server response did not reach the client in the previous session. In either case, the client should add these server IDs to their associated transactions at this point.

If the client previously recorded the <SRVRTID>, this response is a change in status or in the contents of the transaction. The request might have originated from this client, another client, or might be the result of server processing.

If the <TRNUID> does not match any transaction known to the client, a second client initiated this transaction. In rarer cases the response might be a transaction initially requested by this client, for which the client has lost its record; for example, the client has reverted to a backup.

The diagram below describes the processing and interpretation of <SRVRTID> and <TRNUID> identifiers by the client:



After the server has processed all the synchronization responses, the client scans its database of transactions to verify that they have all been assigned a <SRVRTID>. Any transactions missing this identifier were never received by the server and should be resent (using the originally assigned <TRNUID> to avoid duplicate requests). Additionally, the client should record the <TOKEN> received in the response.

12.11Message Sets and Profile

Open Financial Exchange separates messages that the client and server send into groups called message sets. Each financial institution defines the message sets that a particular institution will support. Currently, all the payment messages described in this chapter fall into a single message set.

The message set contains options and attributes that allow a financial institution to customize its use of Open Financial Exchange. The options and attributes are defined in the profile as part of the message set definition. Each set of options and attributes appears within an aggregate that is specific to a message set. Specifically, all of the options and attributes that pertain to payments are contained within <BILLPAYMSGSETV1>.

12.11.1 Bill Pay Message Sets and Messages

12.11.1.1 Bill Pay Message Set Request Messages

Message Set	*Messages*
<BILLPAYMSGSET>	
<BILLPAYMSGSETV1>	
<BILLPAYMSGSRQV1>	PMTTRNRQ
	PMTRQ
	PMTMODRQ
	PMTCANCRQ
	RECPMTTRNRQ
	RECPMTRQ
	RECPMTMODRQ
	RECPMTCANCRQ
	PAYEETRNRQ
	PAYEERQ
	PAYEEMODRQ
	PAYEEDELRQ
	PMTINQTRNRQ
	PMTINQRQ
	PMTMAILTRNRQ
	PMTMAILRQ
	PMTSYNCRQ
	RECPMTSYNCRQ
	PAYEESYNCRQ
	PMTMAILSYNCRQ
</BILLPAYMSGSRQV1>	
</BILLPAYMSGSETV1>	
</BILLPAYMSGSET>	

12.11.1.2Bill Pay Message Set Response Messages

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

12.11.2Bill Pay Message Set Profile <BILLPAYMSGSET>

Tag	*Description*
<BILLPAYMSGSET>	
<BILLPAYMSGSETV1>	
<MSGSETCORE>	
</MSGSETCORE>	
<DAYSWITH>	Offset to withdrawal date, such that (DTDUE - DAYSTOPAY) + (DAYSWITH) determines the date on which the funds are withdrawn from the user's account. NOTE: If the value of DAYSWITH is "-1," then the withdrawal date is the same as the payment date (DTDUE).
<DFLTDAYSTOPAY>	Default number of days to pay by check (except by transfer), *N-3*
<XFERDAYSWITH>	Number of days before processing date that funds are withdrawn for payment by transfer, *N-3*
<XFERDFLTDAYSTOPAY>	Default number of days to pay by transfer, *N-3*
<PROCDAYSOFF>	Days of week that no processing occurs; 0 or more of (MONDAY, TUESDAY, WEDNESDAY, THURSDAY, FRIDAY, SATURDAY, SUNDAY)
<PROCENDTM>	Time of day that day's processing ends, *time*
<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*
<POSTPROCWND>	Number of days after a transaction is processed that it is accessible for status inquiries, *N-3*
<STSVIAMODS>	If Y, server supports communication of server-initiated payment status changes by means of the PMTMODRS message
<PMTBYADDR>	The payment provider supports payments to payees identified by billing address, that is, the PAYEE aggregate, *Boolean*
<PMTBYXFER>	The payment provider supports payments to payees identified by destination account, *Boolean*
<PMTBYPAYEEID>	The payment provider supports payments to payees identified by a user-supplied payee ID, *Boolean*
<CANADDPAYEE>	User can add payees. if no, the user is restricted to payees added to the user's payee list by the payment system, *Boolean*
<HASEXTDPMT>	Supports the EXTDPMT business payment aggregate, *Boolean*
<CANMODPMTS>	Permits modifications to payments, that is PMTMODRQ, *Boolean*
<CANMODMDLS>	Permits modifications to models, that is REQPMTMODRQ, *Boolean*
<DIFFFIRSTPMT>	Support for specifying a different amount for the first payment generated by a model, *Boolean*
<DIFFLASTPMT>	Support for specifying a different amount for the last payment generated by a model, *Boolean*
</BILLPAYMSGSETV1>	
</BILLPAYMSGSET>	

12.12Examples

12.12.1Scheduling a Payment

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

```
<!-- payment example 1 -->

<OFX>
   <SIGNONMSGSRQV1>
      <SONRQ>
         <DTCLIENT>19961029101000
         <USERID>123-45-6789
         <USERPASS>MyPassword
         <LANGUAGE>ENG
         <FI>
            <ORG>NCH
            <FID>12321
         </FI>
         <APPID>MyApp
         <APPVER>0700
      </SONRQ>
   </SIGNONMSGSRQV1>
   <BILLPAYMSGSRQV1>
      <PMTTRNRQ>
         <TRNUID>1001
         <PMTRQ>
            <PMTINFO>
               <BANKACCTFROM>
                  <BANKID>123432123
                  <ACCTID>516273
                  <ACCTTYPE>CHECKING
               </BANKACCTFROM>
               <TRNAMT>123.45
               <PAYEE>
                  <NAME>J. C. Counts
                  <ADDR1>100 Main St.
                  <CITY>Turlock
                  <STATE>CA
                  <POSTALCODE>90101
                  <PHONE>415.987.6543
               </PAYEE>
               <PAYACCT>10101
               <DTDUE>19971001
               <MEMO>payment #3
            </PMTINFO>
         </PMTRQ>
      </PMTTRNRQ>
   </BILLPAYMSGSRQV1>
</OFX>
```

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

```
<OFX>
   <SIGNONMSGSRSV1>
      <SONRS>
         <STATUS>
            <CODE>0
            <SEVERITY>INFO
         </STATUS>
         <DTSERVER>19961029101003
         <LANGUAGE>ENG
         <DTPROFUP>19961029101003
         <DTACCTUP>19961029101003
      </SONRS>
   </SIGNONMSGSRSV1>
   <BILLPAYMSGSRSV1>
      <PMTTRNRS>
         <TRNUID>1001
         <STATUS>
            <CODE>0
            <SEVERITY>INFO
         </STATUS>
         <PMTRS>
            <SRVRTID>1030155
            <PAYEELSTID>123214
            <CURDEF>USD
            <PMTINFO>
               <BANKACCTFROM>
                  <BANKID>123432123
                  <ACCTID>516273
                  <ACCTTYPE>CHECKING
               </BANKACCTFROM>
               <TRNAMT>123.45
               <PAYEE>
                  <NAME>J. C. Counts
                  <ADDR1>100 Main St.
                  <CITY>Turlock
                  <STATE>CA
                  <POSTALCODE>90101
                  <PHONE>415.987.6543
               </PAYEE>
               <PAYACCT>10101
               <DTDUE>19971001
               <MEMO>payment #3
            </PMTINFO>
            <EXTDPAYEE>
               <PAYEEID>9076
               <IDSCOPE>USER
               <NAME>J. C. Counts
               <DAYSTOPAY>3
            </EXTDPAYEE>
            <CHECKNUM>20111
            <PMTPRCSTS>
               <PMTPRCCODE>WILLPROCESSON
               <DTPMTPRC>19971001
            </PMTPRCSTS>
         </PMTRS>
      </PMTTRNRS>
   </BILLPAYMSGSRSV1>
</OFX>
```

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

```
<!-- payment example 2 -->

<OFX>
   <SIGNONMSGSRQV1>
      <SONRQ>
         <DTCLIENT>19961029101000
         <USERID>123-45-6789
         <USERPASS>MyPassword
         <LANGUAGE>ENG
         <FI>
            <ORG>NCH
            <FID>12321
         </FI>
         <APPID>MyApp
         <APPVER>0700
      </SONRQ>
   </SIGNONMSGSRQV1>
   <BILLPAYMSGSRQV1>
      <PMTTRNRQ>
         <TRNUID>1001
         <PMTRQ>
            <PMTINFO>
               <BANKACCTFROM>
                  <BANKID>123432123
                  <ACCTID>516273
                  <ACCTTYPE>CHECKING
               </BANKACCTFROM>
               <TRNAMT>123.45
               <PAYEEID>9076
               <PAYACCT>10101
               <DTDUE>19971101
               <MEMO>Payment #4
            </PMTINFO>
         </PMTRQ>
      </PMTTRNRQ>
   </BILLPAYMSGSRQV1>
</OFX>
```

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

```
<OFX>
   <SIGNONMSGSRSV1>
      <SONRS>
         <STATUS>
            <CODE>0
            <SEVERITY>INFO
         </STATUS>
         <DTSERVER>19961029101003
         <LANGUAGE>ENG
         <DTPROFUP>19961029101003
         <DTACCTUP>19961029101003
      </SONRS>
   </SIGNONMSGSRSV1>
   <BILLPAYMSGSRSV1>
      <PMTTRNRS>
         <TRNUID>1001
         <STATUS>
            <CODE>0
            <SEVERITY>INFO
         </STATUS>
         <PMTRS>
            <SRVRTID>1068405
            <PAYEELSTID>123432
            <CURDEF>USD
            <PMTINFO>
               <BANKACCTFROM>
                  <BANKID>123432123
                  <ACCTID>516273
                  <ACCTTYPE>CHECKING
               </BANKACCTFROM>
               <TRNAMT>123.45
               <PAYEEID>9076
               <PAYACCT>10101
               <DTDUE>19971101
               <MEMO>payment #4
            </PMTINFO>
            <EXTDPAYEE>
               <PAYEEID>9076
               <IDSCOPE>USER
               <NAME>J. C. Counts
               <DAYSTOPAY>3
            </EXTDPAYEE>
            <PMTPRCSTS>
               <PMTPRCCODE>WILLPROCESSON
               <DTPMTPRC>19971101
            </PMTPRCSTS>
         </PMTRS>
      </PMTTRNRS>
   </BILLPAYMSGSRSV1>
</OFX>
```

12.12.2Modifying a Payment

Change the amount of a payment to $125.99

```
<OFX>
   <SIGNONMSGSRQV1>
      <SONRQ>
         <DTCLIENT>19961029101000
         <USERID>123-45-6789
         <USERPASS>MyPassword
         <LANGUAGE>ENG
         <FI>
            <ORG>NCH
            <FID>12321
         </FI>
         <APPID>MyApp
         <APPVER>0700
      </SONRQ>
   </SIGNONMSGSRQV1>
   <BILLPAYMSGSRQV1>
      <PMTTRNRQ>
         <TRNUID>1021
         <PMTMODRQ>
            <SRVRTID>1030776
            <PMTINFO>
               <BANKACCTFROM>
                  <BANKID>123432123
                  <ACCTID>516273
                  <ACCTTYPE>CHECKING
               </BANKACCTFROM>
               <TRNAMT>125.99                    <!-- changed amount -->
               <PAYEE>
                  <NAME>J. C. Counts
                  <ADDR1>100 Main St.
                  <CITY>Turlock
                  <STATE>CA
                  <POSTALCODE>90101
                  <PHONE>415.987.6543
               </PAYEE>
               <PAYACCT>10101
               <DTDUE>19971001
               <MEMO>payment #3
            </PMTINFO>
         </PMTMODRQ>
      </PMTTRNRQ>
   </BILLPAYMSGSRQV1>
</OFX>
```

Server response echoes the request:

```
<OFX>
   <SIGNONMSGSRSV1>
      <SONRS>
         <STATUS>
            <CODE>0
            <SEVERITY>INFO
         </STATUS>
         <DTSERVER>19961029101003
         <LANGUAGE>ENG
         <DTPROFUP>19961029101003
         <DTACCTUP>19961029101003
      </SONRS>
   </SIGNONMSGSRSV1>
   <BILLPAYMSGSRSV1>
      <PMTTRNRS>
         <TRNUID>1021
         <STATUS>
            <CODE>0
            <SEVERITY>INFO
         </STATUS>
         <PMTMODRS>
            <SRVRTID>1030776
            <PMTINFO>
               <BANKACCTFROM>
                  <BANKID>123432123
                  <ACCTID>516273
                  <ACCTTYPE>CHECKING
               </BANKACCTFROM>
               <TRNAMT>125.99                    <!-- changed amount -->
               <PAYEE>
                  <NAME>J. C. Counts
                  <ADDR1>100 Main St.
                  <CITY>Turlock
                  <STATE>CA
                  <POSTALCODE>90101
                  <PHONE>415.987.6543
               </PAYEE>
               <PAYACCT>10101
               <DTDUE>19971001
               <MEMO>payment #3
            </PMTINFO>
         </PMTMODRS>
      </PMTTRNRS>
   </BILLPAYMSGSRSV1>
</OFX>
```

Change the date of a payment to December 12, 1997

```
<OFX>
   <SIGNONMSGSRQV1>
      <SONRQ>
         <DTCLIENT>19961029101000
         <USERID>123-45-6789
         <USERPASS>MyPassword
         <LANGUAGE>ENG
         <FI>
            <ORG>NCH
            <FID>12321
         </FI>
         <APPID>MyApp
         <APPVER>0700
      </SONRQ>
   </SIGNONMSGSRQV1>
   <BILLPAYMSGSRQV1>
      <PMTTRNRQ>
         <TRNUID>32456
         <PMTMODRQ>
            <SRVRTID>1030776
            <PMTINFO>
               <BANKACCTFROM>
                  <BANKID>123432123
                  <ACCTID>516273
                  <ACCTTYPE>CHECKING
               </BANKACCTFROM>
               <TRNAMT>125.99
               <PAYEE>
                  <NAME>J. C. Counts
                  <ADDR1>100 Main St.
                  <CITY>Turlock
                  <STATE>CA
                  <POSTALCODE>90101
                  <PHONE>415.987.6543
               </PAYEE>
               <PAYACCT>10101
               <DTDUE>19971212                  <!-- changed date -->
               <MEMO>payment #3
            </PMTINFO>
         </PMTMODRQ>
      </PMTTRNRQ>
   </BILLPAYMSGSRQV1>
</OFX>
```

Server response echoes the request:

```
<OFX>
   <SIGNONMSGSRSV1>
      <SONRS>
         <STATUS>
            <CODE>0
            <SEVERITY>INFO
         </STATUS>
         <DTSERVER>19961029101003
         <LANGUAGE>ENG
         <DTPROFUP>19961029101003
         <DTACCTUP>19961029101003
      </SONRS>
   </SIGNONMSGSRSV1>
   <BILLPAYMSGSRSV1>
      <PMTTRNRS>
         <TRNUID>32456
         <STATUS>
            <CODE>0
            <SEVERITY>INFO
         </STATUS>
         <PMTMODRS>
            <SRVRTID>1030776
            <PMTINFO>
               <BANKACCTFROM>
                  <BANKID>123432123
                  <ACCTID>516273
                  <ACCTTYPE>CHECKING
               </BANKACCTFROM>
               <TRNAMT>125.99
               <PAYEE>
                  <NAME>J. C. Counts
                  <ADDR1>100 Main St.
                  <CITY>Turlock
                  <STATE>CA
                  <POSTALCODE>90101
                  <PHONE>415.987.6543
               </PAYEE>
               <PAYACCT>10101
               <DTDUE>19971212                  <!-- changed date -->
               <MEMO>payment #3
            </PMTINFO>
         </PMTMODRS>
      </PMTTRNRS>
   </BILLPAYMSGSRSV1>
</OFX>
```

12.12.3Canceling a Payment

Cancel a payment:

```
<OFX>
   <SIGNONMSGSRQV1>
      <SONRQ>
         <DTCLIENT>19971029101000
         <USERID>123-45-6789
         <USERPASS>MyPassword
         <LANGUAGE>ENG
         <FI>
            <ORG>NCH
            <FID>12321
         </FI>
         <APPID>MyApp
         <APPVER>0700
      </SONRQ>
   </SIGNONMSGSRQV1>
   <BILLPAYMSGSRQV1>
      <PMTTRNRQ>
         <TRNUID>54601
         <PMTCANCRQ>
            <SRVRTID>1030776
         </PMTCANCRQ>
      </PMTTRNRQ>
   </BILLPAYMSGSRQV1>
</OFX>
```

Server responds:

```
<OFX>
   <SIGNONMSGSRSV1>
      <SONRS>
         <STATUS>
            <CODE>0
            <SEVERITY>INFO
         </STATUS>
         <DTSERVER>19971029101003
         <LANGUAGE>ENG
         <DTPROFUP>19961029101003
         <DTACCTUP>19961029101003
      </SONRS>
   </SIGNONMSGSRSV1>
   <BILLPAYMSGSRSV1>
      <PMTTRNRS>
         <TRNUID>54601
         <STATUS>
            <CODE>0
            <SEVERITY>INFO
         </STATUS>
         <PMTCANCRS>
            <SRVRTID>1030776
         </PMTCANCRS>
      </PMTTRNRS>
   </BILLPAYMSGSRSV1>
</OFX>
```

12.12.4Updating Payment Status

Update payment status:

```
<OFX>
   <SIGNONMSGSRQV1>
      <SONRQ>
         <DTCLIENT>19970129101000
         <USERID>123-45-6789
         <USERPASS>MyPassword
         <LANGUAGE>ENG
         <FI>
            <ORG>NCH
            <FID>12321
         </FI>
         <APPID>MyApp
         <APPVER>0700
      </SONRQ>
   </SIGNONMSGSRQV1>
   <BILLPAYMSGSRQV1>
      <PMTINQTRNRQ>
         <TRNUID>7865
         <PMTINQRQ>
            <SRVRTID>565321
         </PMTINQRQ>
      </PMTINQTRNRQ>
   </BILLPAYMSGSRQV1>
</OFX>
```

Server responds with updated status and check number:

```
<OFX>
   <SIGNONMSGSRSV1>
      <SONRS>
         <STATUS>
            <CODE>0
            <SEVERITY>INFO
         </STATUS>
         <DTSERVER>19961029101003
         <LANGUAGE>ENG
         <DTPROFUP>19961029101003
         <DTACCTUP>19961029101003
      </SONRS>
   </SIGNONMSGSRSV1>
   <BILLPAYMSGSRSV1>
      <PMTINQTRNRS>
         <TRNUID>7865
         <STATUS>
            <CODE>0
            <SEVERITY>INFO
         </STATUS>
         <PMTINQRS>
            <SRVRTID>565321
            <PMTPRCSTS>
               <PMTPRCCODE>PROCESSEDON
               <DTPMTPRC>19970201
            </PMTPRCSTS>
            <CHECKNUM>6017
```

```
        </PMTINQRS>
      </PMTINQTRNRS>
   </BILLPAYMSGSRSV1>
</OFX>
```

12.12.5Scheduling a Recurring Payment

Create a recurring payment of 36 monthly payments of $395 to a standard payee. The first payment will be on November 15, 1997:

```
<OFX>
   <SIGNONMSGSRQV1>
      <SONRQ>
         <DTCLIENT>19961029101000
         <USERID>123-45-6789
         <USERPASS>MyPassword
         <LANGUAGE>ENG
         <FI>
            <ORG>NCH
            <FID>12321
         </FI>
         <APPID>MyApp
         <APPVER>0700
      </SONRQ>
   </SIGNONMSGSRQV1>
   <BILLPAYMSGSRQV1>
      <RECPMTTRNRQ>
         <TRNUID>1001
         <RECPMTRQ>
            <RECURRINST>
               <NINSTS>36
               <FREQ>MONTHLY
            </RECURRINST>
            <PMTINFO>
               <BANKACCTFROM>
                  <BANKID>555432180
                  <ACCTID>763984
                  <ACCTTYPE>CHECKING
               </BANKACCTFROM>
               <TRNAMT>395.00
               <PAYEEID>77810
               <PAYACCT>444-78-97572
               <DTDUE>19971115
               <MEMO>Auto loan payment
            </PMTINFO>
         </RECPMTRQ>
      </RECPMTTRNRQ>
   </BILLPAYMSGSRQV1>
</OFX>
```

Server response includes the assigned server transaction ID:

```
<OFX>
   <SIGNONMSGSRSV1>
      <SONRS>
         <STATUS>
            <CODE>0
            <SEVERITY>INFO
         </STATUS>
         <DTSERVER>19971029101003
         <LANGUAGE>ENG
         <DTPROFUP>19961029101003
         <DTACCTUP>19961029101003
      </SONRS>
   </SIGNONMSGSRSV1>
   <BILLPAYMSGSRSV1>
      <RECPMTTRNRS>
         <TRNUID>1001
         <STATUS>
            <CODE>0
            <SEVERITY>INFO
         </STATUS>
         <RECPMTRS>
            <RECSRVRTID>387687138
            <PAYEELSTID>27983
            <CURDEF>USD
            <RECURRINST>
               <NINSTS>36
               <FREQ>MONTHLY
            </RECURRINST>
            <PMTINFO>
               <BANKACCTFROM>
                  <BANKID>555432180
                  <ACCTID>763984
                  <ACCTTYPE>CHECKING
               </BANKACCTFROM>
               <TRNAMT>395.00
               <PAYEEID>77810
               <PAYACCT>444-78-97572
               <DTDUE>19971115
               <MEMO>Auto loan payment
            </PMTINFO>
         </RECPMTRS>
      </RECPMTTRNRS>
   </BILLPAYMSGSRSV1>
</OFX>
```

12.12.6Modifying a Recurring Payment

Change the amount of a recurring payment:

```
<OFX>
   <SIGNONMSGSRQV1>
      <SONRQ>
         <DTCLIENT>19961029101000
         <USERID>123-45-6789
         <USERPASS>MyPassword
         <LANGUAGE>ENG
         <FI>
            <ORG>NCH
            <FID>12321
         </FI>
         <APPID>MyApp
         <APPVER>0700
      </SONRQ>
   </SIGNONMSGSRQV1>
   <BILLPAYMSGSRQV1>
      <RECPMTTRNRQ>
         <TRNUID>2001
         <RECPMTMODRQ>
            <RECSRVRTID>387687138
            <RECURRINST>
               <NINSTS>36
               <FREQ>MONTHLY
            </RECURRINST>
            <PMTINFO>
               <BANKACCTFROM>
                  <BANKID>555432180
                  <ACCTID>763984
                  <ACCTTYPE>CHECKING
               </BANKACCTFROM>
               <TRNAMT>399.95                    <!-- changing amount -->
               <PAYEEID>77810
               <PAYACCT>444-78-97572
               <DTDUE>19971115
               <MEMO>Auto loan payment
            </PMTINFO>
            <MODPENDING>N
         </RECPMTMODRQ>
      </RECPMTTRNRQ>
   </BILLPAYMSGSRQV1>
</OFX>
```

Server responds:

```
<OFX>
   <SIGNONMSGSRSV1>
      <SONRS>
         <STATUS>
            <CODE>0
            <SEVERITY>INFO
         </STATUS>
         <DTSERVER>19971029101003
         <LANGUAGE>ENG
         <DTPROFUP>19961029101003
         <DTACCTUP>19961029101003
      </SONRS>
   </SIGNONMSGSRSV1>
   <BILLPAYMSGSRSV1>
      <RECPMTTRNRS>
         <TRNUID>2001
         <STATUS>
            <CODE>0
            <SEVERITY>INFO
         </STATUS>
         <RECPMTMODRS>
            <RECSRVRTID>387687138
            <RECURRINST>
               <NINSTS>36
               <FREQ>MONTHLY
            </RECURRINST>
            <PMTINFO>
               <BANKACCTFROM>
                  <BANKID>555432180
                  <ACCTID>763984
                  <ACCTTYPE>CHECKING
               </BANKACCTFROM>
               <TRNAMT>399.95                    <!-- changing amount -->
               <PAYEEID>77810
               <PAYACCT>444-78-97572
               <DTDUE>19971115
               <MEMO>Auto loan payment
            </PMTINFO>
            <MODPENDING>N
         </RECPMTMODRS>
      </RECPMTTRNRS>
   </BILLPAYMSGSRSV1>
</OFX>
```

12.12.7Canceling a Recurring Payment

Cancel a recurring payment:

```
<OFX>
   <SIGNONMSGSRQV1>
      <SONRQ>
         <DTCLIENT>19971123101000
         <USERID>123-45-6789
         <USERPASS>MyPassword
         <LANGUAGE>ENG
         <FI>
            <ORG>NCH
            <FID>12321
         </FI>
         <APPID>MyApp
         <APPVER>0700
      </SONRQ>
   </SIGNONMSGSRQV1>
   <BILLPAYMSGSRQV1>
      <RECPMTTRNRQ>
         <TRNUID>10103
         <RECPMTCANCRQ>
            <RECSRVRTID>387687138
            <CANPENDING>Y
         </RECPMTCANCRQ>
      </RECPMTTRNRQ>
   </BILLPAYMSGSRQV1>
</OFX>
```

Server responds with:

```
<OFX>
   <SIGNONMSGSRSV1>
      <SONRS>
         <STATUS>
            <CODE>0
            <SEVERITY>INFO
         </STATUS>
         <DTSERVER>19971123101003
         <LANGUAGE>ENG
         <DTPROFUP>19961029101003
         <DTACCTUP>19961029101003
      </SONRS>
   </SIGNONMSGSRSV1>
   <BILLPAYMSGSRSV1>
      <RECPMTTRNRS>
         <TRNUID>10103
         <STATUS>
            <CODE>0
            <SEVERITY>INFO
         </STATUS>
         <RECPMTCANCRS>
            <RECSRVRTID>387687138
            <CANPENDING>Y
         </RECPMTCANCRS>
      </RECPMTTRNRS>
   </BILLPAYMSGSRSV1>
</OFX>
```

12.12.8

12.12.9Adding a Payee to the Payee List

The user sends a request to add a payee to the user's payee list:

```
<OFX>
   <SIGNONMSGSRQV1>
      <SONRQ>
         <DTCLIENT>19970129101000
         <USERID>123-45-6789
         <USERPASS>MyPassword
         <LANGUAGE>ENG
         <FI>
            <ORG>NCH
            <FID>12321
         </FI>
         <APPID>MyApp
         <APPVER>0700
      </SONRQ>
   </SIGNONMSGSRQV1>
   <BILLPAYMSGSRQV1>
      <PAYEETRNRQ>
         <TRNUID>127688
         <PAYEERQ>
            <PAYEE>
               <NAME>ACME Rocket Works
               <ADDR1>101 Spring St.
               <ADDR2>Suite 503
               <CITY>Watkins Glen
               <STATE>NY
               <POSTALCODE>12345-6789
               <PHONE>888.555.1212
            </PAYEE>
            <PAYACCT>1001-99-8876
         </PAYEERQ>
      </PAYEETRNRQ>
   </BILLPAYMSGSRQV1>
</OFX>
```

The server responds:

```
<OFX>
   <SIGNONMSGSRSV1>
      <SONRS>
         <STATUS>
            <CODE>0
            <SEVERITY>INFO
         </STATUS>
         <DTSERVER>19961029101003
         <LANGUAGE>ENG
         <DTPROFUP>19961029101003
         <DTACCTUP>19961029101003
      </SONRS>
   </SIGNONMSGSRSV1>
   <BILLPAYMSGSRSV1>
      <PAYEETRNRS>
         <TRNUID>127688
         <STATUS>
            <CODE>0
            <SEVERITY>INFO
```

```
        </STATUS>
        <PAYEERS>
           <PAYEELSTID>78096786
           <PAYEE>
              <NAME>ACME Rocket Works
              <ADDR1>101 Spring St.
              <ADDR2>Suite 503
              <CITY>Watkins Glen
              <STATE>NY
              <POSTALCODE>12345-6789
              <PHONE>888.555.1212
           </PAYEE>
           <EXTDPAYEE>
              <PAYEEID>88878
              <IDSCOPE>GLOBAL
              <NAME>ACME Rocket Works, Inc.
              <DAYSTOPAY>2
           </EXTDPAYEE>
           <PAYACCT>1001-99-8876
        </PAYEERS>
     </PAYEETRNRS>
   </BILLPAYMSGSRSV1>
</OFX>
```

12.12.10Synchronizing Scheduled Payments

A client wishes to obtain all Payment and Recurring Payment transactions active on the server:

```
<OFX>
   <SIGNONMSGSRQV1>
      <SONRQ>
         <DTCLIENT>19971123101000
         <USERID>123-45-6789
         <USERPASS>MyPassword
         <LANGUAGE>ENG
         <FI>
            <ORG>NCH
            <FID>12321
         </FI>
         <APPID>MyApp
         <APPVER>0700
      </SONRQ>
   </SIGNONMSGSRQV1>
   <BILLPAYMSGSRQV1>
      <PMTSYNCRQ>
         <TOKEN>7647909384
         <REJECTIFMISSING>N
         <BANKACCTFROM>
            <BANKID>555432180
            <ACCTID>763984
            <ACCTTYPE>CHECKING
         </BANKACCTFROM>
      </PMTSYNCRQ>
      <RECPMTSYNCRQ>
         <TOKEN>9647409384
         <REJECTIFMISSING>N
         <BANKACCTFROM>
            <BANKID>555432180
            <ACCTID>763984
            <ACCTTYPE>CHECKING
         </BANKACCTFROM>
```

```
      </RECPMTSYNCRQ>
   </BILLPAYMSGSRQV1>
</OFX>
```

The server responds with one payment and one Recurring Payment, as well as current token values.

```
<OFX>
   <SIGNONMSGSRSV1>
      <SONRS>
         <STATUS>
            <CODE>0
            <SEVERITY>INFO
         </STATUS>
         <DTSERVER>19971123101003
         <LANGUAGE>ENG
         <DTPROFUP>19961029101003
         <DTACCTUP>19961029101003
      </SONRS>
   </SIGNONMSGSRSV1>
   <BILLPAYMSGSRSV1>
      <PMTSYNCRS>
         <TOKEN>3247989384
         <BANKACCTFROM>
            <BANKID>555432180
            <ACCTID>763984
            <ACCTTYPE>CHECKING
         </BANKACCTFROM>
         <PMTTRNRS>
            <TRNUID>10103
            <STATUS>
               <CODE>0
               <SEVERITY>INFO
            </STATUS>
            <PMTRS>
               <SRVRTID>1030155
               <PAYEELSTID>3486578
               <CURDEF>USD
               <PMTINFO>
                  <BANKACCTFROM>
                     <BANKID>123432123
                     <ACCTID>516273
                     <ACCTTYPE>CHECKING
                  </BANKACCTFROM>
                  <TRNAMT>123.45
                  <PAYEE>
                     <NAME>J. C. Counts
                     <ADDR1>100 Main St.
                     <CITY>Turlock
                     <STATE>CA
                     <POSTALCODE>90101
                     <PHONE>415.987.6543
                  </PAYEE>
                  <PAYACCT>10101
                  <DTDUE>19971001
                  <MEMO>payment #3
               </PMTINFO>
               <EXTDPAYEE>
                  <PAYEEID>9076
                  <IDSCOPE>USER
                  <NAME>J. C. Counts
                  <DAYSTOPAY>3
               </EXTDPAYEE>
                <PMTPRCSTS>
                  <PMTPRCCODE>WILLPROCESSON
                  <DTPMTPRC>19971001
                </PMTPRCSTS>
            </PMTRS>
```

```
      </PMTTRNRS>
    </PMTSYNCRS>
    <RECPMTSYNCRS>
      <TOKEN>94879835
      <BANKACCTFROM>
        <BANKID>555432180
        <ACCTID>763984
        <ACCTTYPE>CHECKING
      </BANKACCTFROM>
      <RECPMTTRNRS>
        <TRNUID>10202
        <STATUS>
          <CODE>0
          <SEVERITY>INFO
        </STATUS>
        <RECPMTRS>
          <RECSRVRTID>387687138
          <PAYEELSTID>938469
          <CURDEF>USD
          <RECURRINST>
            <NINSTS>36
            <FREQ>MONTHLY
          </RECURRINST>
          <PMTINFO>
            <BANKACCTFROM>
              <BANKID>555432180
              <ACCTID>763984
              <ACCTTYPE>CHECKING
            </BANKACCTFROM>
            <TRNAMT>395.00
            <PAYEEID>77810
            <PAYACCT>444-78-97572
            <DTDUE>19971115
            <MEMO>Auto loan payment
          </PMTINFO>
        </RECPMTRS>
      </RECPMTTRNRS>
    </RECPMTSYNCRS>
  </BILLPAYMSGSRSV1>
</OFX>
```

13Investments

Open Financial Exchange supports download of security information and detailed investment account statements including transactions, open orders, balances, and positions.

Client Sends	*Server Responds*
Account identifier	
Whether to download open orders	
Whether to download transactions	
Date range if transactions should be downloaded	
Whether to download positions	
Whether to download balances	
Additional securities to send information about	
	Date and time for statement
	Default currency for statement
	Account identifier
	Investment transactions
	Banking transactions
	Open orders
	Positions
	Account balances
	Available Cash Balance
	Short Balance
	Margin Balance
	Buying power
	Marketing message
	List of securities

NOTE: *This release of Open Financial Exchange does not support trading or tax lots.*

13.1Types of Response Information

The response consists of five types of information:

- Transactions – a combination of bank transaction detail records and investment transaction detail records. Transactions only within the specified start and stop dates are sent.
- Positions – positions a user has at a brokerage. Each statement response must contain a complete set of position records, even if no transactions occurred in the requested statement period for a particular holding.
- Balances – current balances typically reported on an FI statement, such as cash balance or buying power. They can also convey other numbers of interest, such as current interest rates.
- Open Orders – current open trading orders that a user has at a brokerage.
- Securities – any security referenced in either transactions, positions, open orders or explicitly requested.

13.2Sub-Accounts

Many FIs distinguish between activity and positions in cash, margin, and short accounts, with some FIs having many other types of "sub-accounts." Open Financial Exchange defines four standard types of sub-accounts: Cash, Margin, Short, Other. Position, Transaction, and Open Order records identify the sub-account.

13.3Units, Precision, and Signs

This section provides information about numerical values for investment transactions. For more information about common data types used within Open Financial Exchange, refer to Chapter 3.

13.3.1Units

The units for security units and unit price are those commonly used on brokerage statements, and differ for each type of security.

- Stocks and Other – use number of shares for units and dollar value for unit price.
- Mutual Funds – in most cases shares are used, but in some cases the dollar value is used. The unit type is specified in cases in which it can be either.
- Bonds – use face value for units and percentage of par for unit price. For example, a $25,000 bond trading at $88 would use 25000 as the units and 88 as the unit price.
- Options – use number of contracts (not shares) for units, and price per share (not contract) for unit price.

13.3.2Precision

Open Financial Exchange does not specify the precision of fields since the precision is client-dependent. However, it is recommended that clients and servers follow these rules:

- Clients and servers should send as much precision as they have
- Clients and servers should use a precision equal to or better than 1/256 of a share

13.3.3Signs

Chapter 3 describes how to use positive and negative numbers. Briefly, quantities and total values should be signed from the perspective of the user. In a stock buy, the total value is negative, the unit price is always positive, and the number of units is positive.

13.4Bank and Investment Transactions

Many FIs provide investment accounts that allow users to write checks and perform other traditional banking transactions, as well as investment transactions. Open Financial Exchange requires FIs to indicate in the download whether check writing privileges exist for a given account.

FIs need to use the correct transaction record, bank or investment, for each real-world transaction. Use the following guidelines:

- Checks, electronic funds transfers, and ATM transactions associated with CMA or money market sweep accounts are always represented with a bank transaction record.
- Investment actions that involve securities (buy, sell, stock split, reinvest, etc.) are always represented with an investment record. Actions that are cash-only but are directly associated with a security are also investment actions (for example, dividends).
- Other cash-only actions require careful analysis by the FI. Those that affect investment performance analysis should be sent using the appropriate investment action (investment income - miscellaneous, investment expense). Those that are completely unrelated to investment should be sent as a bank record.

13.5Money Market Funds

Money market funds can be handled in one of three different ways depending on how the fund is modeled at the financial institution

- Separate account at the financial institution
- Sweep account within an investment account
- Position within an investment account

13.5.1 Separate Account at the Financial Institution

In this case, the money market fund is in its own account with its own account number, distinct from the investment account. In Open Financial Exchange, you should model the money market fund as a separate money market bank account; see Chapter 11. The banking <STMTRQ> request aggregate and <STMTRS> response aggregate will be used to download transactions.

13.5.2 Sweep Account Within an Investment Account

Open Financial Exchange uses the money market as a "sweep" account, where cash is "swept" as needed when buying and selling securities. The money market fund does not have its own account number. The customer sees the money market fund as an investment-account cash balance. In Open Financial Exchange, checks, ATMs, electronic fund transfer, deposit, and withdrawal transactions should be downloaded using banking transactions within the investment account. However, the sweep transactions in and out of the money market fund should not be downloaded to the client.

13.5.3 Position Within an Investment Account

The customer purchases the money market fund and is held in the account as a position. The money market fund does not have its own account number. In Open Financial Exchange, the money market fund should be returned as a <POSOTHER> position in the <INVPOSLIST>, with a <UNITPRICE> of 1.00 and <UNITS> as the current value of the position. Purchases and redemptions should be modeled as <BUYOTHER> and <SELLOTHER> transactions with a <UNITPRICE> of 1.00 and <UNITS> as the transaction amount.

13.6 Investment Accounts

Investment account information is downloaded using the account information response aggregate <ACCTINFORS>. For more information, refer to Chapter 8. <INVACCTFROM> specifies the account. The <INVACCTINFO> aggregate specifies the investment-specific information.

13.6.1 Specifying the Investment Account <INVACCTFROM>

Tag	*Description*
<INVACCTFROM>	Account-from aggregate
<BROKERID>	Unique identifier for the FI, *A-22*
<ACCTID>	Account number at FI, *A-22*
</INVACCTFROM>	

The <INVACCTTO> aggregate contains the same elements

13.6.2Investment Account Information <INVACCTINFO>

The <INVACCTINFO> aggregate should appear in the <ACCTINFO> aggregate for accounts that support investment statement download. For more information about the <ACCTINFO> aggregate, refer to Chapter 8.

Tag	*Description*
<INVACCTINFO>	Investment-account-information-record aggregate
<INVACCTFROM>	Account at FI
</INVACCTFROM>	
<USPRODUCTTYPE>	Classification of account. See section 13.6.2.1 for values
<CHECKING>	Whether the account has check writing privileges, *Boolean*
<SVCSTATUS>	Activation status for investment statement download for the account. ACTIVE (signed up), PEND (in the process of signing up), AVAIL (have not signed up).
<INVACCTTYPE>	Type of account. INDIVIDUAL, JOINT, TRUST, CORPORATE
<OPTIONLEVEL>	Text description of option trading privileges, *A-40*
</INVACCTINFO>	

If an investment account has payments functionality, the analogous PMTINFO aggregate (see 12.5.1) should also be sent in the ACCTINFO for the account. Payment information will be sent using the message sets described in the Chapter 12.

13.6.2.1Values for <USPRODUCTTYPE>

<USPRODUCTTYPE> classifies accounts according to their account type. Valid values are:

Product Type	*Description*
401K	A 401(K) account
403B	A 403(B) account
IRA	An IRA account
KEOGH	Keogh (Money Purchase/Profit Sharing)
OTHER	Other account type
SARSEP	Salary Reduction Simplified Employer Pension plan
SIMPLE	Savings Incentive Match Plan for employees
NORMAL	Regular account
TDA	Tax Deferred Annuity
TRUST	Trust (including UTMA)
UGMA	Custodial account

13.6.2.2International Note

The <USPRODUCTTYPE> tag is intended for use by FIs in the United States. Open Financial Exchange will be expanded to provide equivalent tags to support the needs for other countries.

13.6.3 Brokerage, Mutual Fund, and 401K Accounts

Investment accounts include brokerage accounts, mutual fund accounts, 401K accounts, and other retirement accounts. Open Financial Exchange supports transactions, positions, balances, and open orders for all of these account types.

13.7 Investment Message Sets and Profile

Open Financial Exchange separates messages that the client and server send into groups called message sets. Each financial institution defines the message sets that the institution supports. The messages described in this chapter fall into two message sets:

- Investment Statement Download
- Security Information

Each message set contains options that allow a financial institution to customize its use of Open Financial Exchange. For example, an institution can support the Investment Statement Download Set (INVSTMTMSGSETV1), but it can choose not to support the download of open orders.

The options and attributes are defined in the profile 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, <INVSTMTMSGSETV1> contains all the options and attributes that pertain to investment statement download.

13.7.1 Investment Statement Download

13.7.1.1 Investment Statement Message Set Profile <INVSTMTMSGSET>

The investment statement message set profile aggregate <INVSTMTMSGSET> is used in the response to a financial institution profile request (see Chapter 7) to specify which activities it supports.

Tag	*Description*
<INVSTMTMSGSET>	Investment-statement-message-set-profile aggregate
<INVSTMTMSGSETV1>	Version 1 message set
<MSGSETCORE>	Common message set information, see Chapter 7
</MSGSETCORE>	
<TRANDNLD>	Whether the FI server downloads investment statement transactions, *Boolean*
<OODNLD>	Whether the FI server downloads investment open orders, *Boolean*
<POSDNLD>	Whether the FI server downloads investment statement positions, *Boolean*
<BALDNLD>	Whether the FI server downloads investment balances, *Boolean*
<CANEMAIL>	Whether the FI supports investment e-mail. Use generic e-mail profile to specify whether generic e-mail is supported; see Chapter 9. *Boolean*
</INVSTMTMSGSETV1>	
</INVSTMTMSGSET>	

13.7.1.2 Investment Statement Message Set and Messages

13.7.1.2.1 Investment Statement Message Set Request Messages

Tag	*Description*
<INVSTMTMSGSET>	
<INVSTMTMSGSETV1>	
<INVSTMTMSGSRQV1>	INVSTMTTRNRQ
	INVSTMTTRQ
	INVMAILTRNRQ
	INVMAILRQ
	INVMAILSYNCRQ
</INVSTMTMSGSRQV1>	
</INVSTMTMSGSETV1>	
</INVSTMTMSGSET>	

13.7.1.2.2 Investment Statement Message Set Response Messages

Tag	*Description*
<INVSTMTMSGSET>	
<INVSTMTMSGSETV1>	
<INVSTMTMSGSRSV1>	INVSTMTTRNRS
	INVSTMTRS
	INVMAILTRNRS
	INVMAILRS
	INVMAILSYNCRS
</INVSTMTMSGSRSV1>	
</INVSTMTMSGSETV1>	
</INVSTMTMSGSET>	

13.7.2Security Information

13.7.2.1Security List Message Set Profile <SECLISTMSGSET>

The security list message set profile aggregate <SECLISTMSGSET> is used in the response to an FI profile request (see Chapter 7) to specify which activities it supports.

Tag	*Description*
<SECLISTMSGSET>	Security-information-message-set-profile aggregate
<SECLISTMSGSETV1>	Version 1 message set
<MSGSETCORE>	Common message set information, see Chapter 7
</MSGSETCORE>	
<SECLISTRQDNLD>	Whether the FI server responds to security list requests, *Boolean*
</SECLISTMSGSETV1>	
</SECLISTMSGSET>	

13.7.2.2Security List Message Set and Messages

13.7.2.2.1Security List Message Set Request Messages

Tag	*Description*
<SECLISTMSGSET>	
<SECLISTMSGSETV1>	
<SECLISTMSGSRQV1>	SECLISTTRNRQ SECLISTRQ
</SECLISTMSGSRQV1>	
</SECLISTMSGSETV1>	
</SECLISTMSGSET >	

13.7.2.2.2Security List Message Set Response Messages

Tag	*Description*
<SECLISTMSGSET>	
<SECLISTMSGSETV1>	
<SECLISTMSGSRSV1>	SECLISTTRNRS SECLISTRS
</SECLISTMSGSRSV1>	
</SECLISTMSGSETV1>	
</SECLISTMSGSET >	

13.8Investment Securities

13.8.1Security Identification <SECID>

Securities must be consistently identified to allow client applications to prepare accurate investment reports across all user investment accounts, even at multiple FIs. At this time, neither a security name nor its symbol is standardized. Therefore, Open Financial Exchange uses CUSIP numbers (a unique 9-digit alphanumeric identifier) to identify securities. CUSIP numbers are available for the vast majority of securities traded today, including those without symbols such as bonds. For a security that does not have a CUSIP, a financial institution must follow the standard procedure of assigning a CUSIP by using itself as the issuer to avoid conflict with any other CUSIP.

Tag	*Description*
<SECID>	Security-identifier aggregate
<UNIQUEID>	Unique identifier for the security. CUSIP for US FIs. *A-32*
<UNIQUEIDTYPE>	Name of standard used to identify the security i.e., "CUSIP" for FIs in the United States, *A-10*
</SECID>	

13.8.1.1International Note

Non-US financial institutions that do not have access to CUSIP numbers must supply a unique identifier for each security in the UNIQUEID field of this aggregate. Open Financial Exchange will be expanded to include other security identifying standards.

13.8.2Security List Request

The user can use the SECLISTTRNRQ and SECLISTRQ aggregates to request information about specific securities. The SECLISTTRNRQ is the transaction-level aggregate that contains the SECLISTRQ. The SECLISTRQ aggregate specifies for which securities information is being requested.

13.8.2.1Security List Transaction Request <SECLISTTRNRQ>

Tag	*Description*
<SECLISTTRNRQ>	Transaction-request aggregate
<TRNUID>	Client-assigned globally unique ID for this transaction *trnuid*
<CLTCOOKIE>	Data to be echoed in the transaction response, *A-32*
<TAN>	Transaction authorization number; used in some countries with some types of transactions. Country-specific documentation will define messages that require a <TAN>, *A-80*
<SECLISTRQ>	Aggregate for the security list request (see section 13.8.2.2)
</SECLISTRQ>	
</SECLISTTRNRQ>	

13.8.2.2Security List Request <SECLISTRQ>

For the security list request, securities must be specified with either a SECID aggregate, a ticker symbol, or an FI assigned identifier.

Tag	*Description*
<SECLISTRQ>	Security-list-request aggregate
<SECRQ>	Security request (one or more)
Security identification. Specify either <SECID>, <TICKER>, or <FIID>.	
<SECID>	Security identifier aggregate
</SECID>	
-or-	
<TICKER>	Ticker symbol, *A-32*
-or-	
<FIID>	FI specific ID for the security, *A-32*
</SECRQ>	
</SECLISTRQ>	

13.8.3Security List Response

If the client sends a security list request to an FI, then the server must send back a security list response to the client application. The security list response is used primarily to report the status of the security list request. The actual security information should be sent in the security list SECLIST aggregate described in section 13.8.4.

13.8.3.1Security List Transaction Response <SECLISTTRNRS>

Tag	*Description*
<SECLISTTRNRS>	Transaction-response aggregate
<TRNUID>	Client-assigned globally unique ID for this transaction *trnuid*
<STATUS>	Status aggregate
</STATUS>	
<CLTCOOKIE>	Client-provided data, **REQUIRED** if provided in request, *A-32*
<SECLISTRS>	Aggregate for the security list response, see 13.8.3.3
</SECLISTRS>	
</SECLISTTRNRS>	

13.8.3.2Status Codes

Code	*Meaning*
0	Success
2000	General error (ERROR)
2019	Duplicate request (ERROR)
12500	One or more securities not found (ERROR)

13.8.3.3Security List Response <SECLISTRS>

The security list response aggregate, the only empty aggregate in OFX, is used to respond to the <SECLISTRQ>. It is used to signify that the security list is generated as a result of a security list request. The actual security information should be included in the <SECLIST> aggregate.

Tag	*Description*
<SECLISTRS> </SECLISTRS>	Security-list-response (the only empty aggregate in OFX).

13.8.4Security List <SECLIST>

The SECLIST should be sent in the following two cases.

- In response to a SECLISTTRNRQ sent by a client application where the SECLIST should contain information for each security specified in the SECLISTTRNRQ.
- When the response file contains an investment statement download that has positions, transactions, or open orders. The SECLIST should contain information about security referenced in the investment statement download. Clients are completely dependent on the security list to provide descriptive information for the securities referenced in positions, transactions, and open orders.

Tag	*Description*
<SECLIST>	Security-list-request aggregate
<*XXX*INFO> </*XXX*INFO>	Security information aggregates (one or more): <DEBTINFO>, <MFINFO>, <OPTINFO>, <OTHERINFO>, or <STOCKINFO>.
</SECLIST>	

13.8.5Securities Information

The <MFINFO>, <STOCKINFO>, <OPTINFO>, <DEBTINFO>, and <OTHERINFO> aggregates provide security information. They define the type of security, and one or more sets of descriptive information. These aggregates relate the <SECID> used in positions, transactions, and open orders to descriptive information about those securities. In this way, the system describes a given security only once, no matter how many times it is referenced.

13.8.5.1 General Securities Information <SECINFO>

The <SECINFO> aggregate contains fields that are common to all security types. This aggregate is used in the security type specific aggregates in the following sections.

Tag	*Description*
<SECINFO>	Security-information aggregate
<SECID>	Security-identifier aggregate
</SECID>	
<SECNAME>	Full name of security, *A-120*
<TICKER>	Ticker symbol (at most one), *A-32*
<FIID>	FI ID number for this security (at most one), *A-32*
<RATING>	Rating, *A-10*
<UNITPRICE>	Current price of security, *unitprice*
<DTASOF>	Date as of for the unit price, *datetime*
<CURRENCY>	Overriding currency aggregate for unit price, see section 5.2
</CURRENCY>	
<MEMO>	*Memo*
</SECINFO>	

13.8.5.2Debt Information <DEBTINFO>

Tag	*Description*
<DEBTINFO>	Opening tag for debt information aggregate
<SECINFO>	Security information aggregate
</SECINFO>	
<PARVALUE>	Par value, *amount*
<DEBTTYPE>	Debt type (at most one) COUPON = coupon ZERO = zero coupon
<DEBTCLASS>	Classification of debt. TREASURY, MUNICIPAL, CORPORATE, OTHER.
<COUPONRT>	Bond coupon rate for next closest call date (at most one), *rate*
<DTCOUPON>	Maturity date for next coupon, *date*
<COUPONFREQ>	When coupons mature. One of the following values: MONTHLY, QUARTERLY, SEMIANNUAL, ANNUAL, or OTHER.
<CALLPRICE>	Bond call price (at most one), *unitprice*
<YIELDTOCALL>	Yield to next call, *rate*
<DTCALL>	Next call date (at most one), *date*
<CALLTYPE>	Type of next call. CALL, PUT, PREFUND, MATURITY
<YIELDTOMAT>	Yield to maturity, *rate*
<DTMAT>	Debt maturity date (at most one), *date*
<ASSETCLASS>	Asset Class (at most one), DOMESTICBOND, INTLBOND, LARGESTOCK SMALLSTOCK, INTLSTOCK, MONEYMRKT, OTHER
<FIASSETCLASS>	Text string containing an FI defined asset class, *A-32*
</DEBTINFO>	

13.8.5.3Mutual Fund Information <MFINFO>

Tag	*Description*
<MFINFO>	Mutual-fund-information aggregate
<SECINFO>	Security-information aggregate
</SECINFO>	
<MFTYPE>	Mutual fund type. OPENEND, CLOSEEND, OTHER
<YIELD>	Current yield reported as portion of the fund's assets (at most one), *rate*
<DTYIELDASOF>	As-of date for yield value, *datetime*
<MFASSETCLASS>	Asset class breakdown for the mutual fund
<PORTION>	Portion of the mutual fund with a specific asset classification (one or more)
<ASSETCLASS>	Asset Class, DOMESTICBOND, INTLBOND, LARGESTOCK SMALLSTOCK, INTLSTOCK, MONEYMRKT, OTHER
<PERCENT>	Percentage of the fund that falls under this asset class, *rate*
</PORTION>	
</MFASSETCLASS>	
<FIMFASSETCLASS>	FI defined asset class breakdown for the mutual fund
<FIPORTION>	Portion of the mutual fund with a specific asset classification (one or more)
<FIASSETCLASS>	Text string containing an FI defined asset class, *A-32*
<PERCENT>	Percentage of the fund that falls under this asset class, *rate*
</FIPORTION>	
</FIMFASSETCLASS>	
</MFINFO>	

13.8.5.4Option Information <OPTINFO>

Tag	*Description*
<OPTINFO>	Option-information aggregate
<SECINFO>	Security-information aggregate
</SECINFO>	
<OPTTYPE>	Option type: PUT = put CALL = call
<STRIKEPRICE>	Strike price *unitprice*
<DTEXPIRE>	Expiration date, *date*
<SHPERCTRCT>	Shares per contract, *N-5*
<SECID>	Security ID of the underlying security
</SECID>	
<ASSETCLASS>	Asset Class (at most one), DOMESTICBOND, INTLBOND, LARGESTOCK SMALLSTOCK, INTLSTOCK, MONEYMRKT, OTHER
<FIASSETCLASS>	Text string containing an FI defined asset class, *A-32*
</OPTINFO>	

13.8.5.5Other Security Type Information <OTHERINFO>

Use this aggregate for security types other than debts, mutual funds, options, and stocks.

Tag	*Description*
<OTHERINFO>	Other aggregate.
<SECINFO>	Security information aggregate
</SECINFO>	
<TYPEDESC>	Description of security type, *A-32*
<ASSETCLASS>	Asset Class (at most one), DOMESTICBOND, INTLBOND, LARGESTOCK SMALLSTOCK, INTLSTOCK, MONEYMRKT, OTHER
<FIASSETCLASS>	Text string containing an FI defined asset class, *A-32*
</OTHERINFO>	

13.8.5.6Stock Information <STOCKINFO>

Tag	*Description*
<STOCKINFO>	Stock-information aggregate
<SECINFO>	Security-information aggregate
</SECINFO>	
<STOCKTYPE>	Stock type: COMMON, PREFERRED, CONVERTIBLE, OTHER
<YIELD>	Current yield reported as the dividend expressed as a portion of the current stock price (at most one), *rate*
<DTYIELDASOF>	As-of date for yield value, *datetime*
<ASSETCLASS>	Asset Class (at most one): DOMESTICBOND, INTLBOND, LARGESTOCK SMALLSTOCK, INTLSTOCK, MONEYMRKT, OTHER
<FIASSETCLASS>	Text string containing an FI defined asset class, *A-32*
</STOCKINFO>	

13.8.5.7Asset Class Descriptions

Asset Class	*Description*
DOMESTICBOND	The Domestic Bonds asset class consists of government or corporate bonds issued in the United States.
INTLBOND	The International Bonds asset class consists of government or corporate bonds issued in foreign countries or the United States.
LARGESTOCK	The Large Cap Stocks asset class consists of stocks for U.S. companies with market capitalizations of $2 billion or more.
SMALLSTOCK	The Small Cap Stocks asset class consists of stocks for U.S. companies with market capitalizations of approximately $100 million to $2 billion.
INTLSTOCK	The International Stocks asset class consists of publicly-traded stocks for companies based in foreign countries.
MONEYMRKT	The Money Market asset class consists of stable, short-term investments which provide income that rises and falls with short-term interest rates.
OTHER	The Other asset class consists of investments which do not fit in any of the other asset classes.

13.9 Investment Statement Download

Investment statement download allows a customer to receive transactions, positions, open orders, and balances that are typically part of a regular paper statement. Clients can retrieve statement data on a daily basis if they wish.

Clients usually allow customers to view investment transactions and guide customers through a process of updating their account registers based on the downloaded transactions. By using transaction IDs supplied by FIs, Open Financial Exchange makes it possible for clients to insure that each transaction is downloaded only once. The request also contains starting and ending dates to limit the amount of downloaded data. Clients can remember the last date they receive a download, and use that date as the starting date in the next request.

Investment statement download requires the client to designate an account for the download, and to indicate what type of data should be downloaded. 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 one is specified. If a date range is not specified, the server returns all available transactions for the account.

13.9.1 Investment Statement Request

Investment statement download can be requested using the INVSTMTTRNRQ and INVSTMTRQ aggregates. The INVSTMTTRNRQ is the transaction level aggregate that contains the INVSTMTRQ. The INVSTMTRQ aggregate specifies what types of information to include in the statement download and from which account to download the information.

13.9.1.1 Investment Statement Transaction Request <INVSTMTTRNRQ>

Tag	*Description*
<INVSTMTTRNRQ>	Transaction-request aggregate
<TRNUID>	Client-assigned globally unique ID for this transaction, *trnuid*
<CLTCOOKIE>	Data to be echoed in the transaction response, *A-32*
<TAN>	Transaction authorization number; used in some countries with some types of transactions. Country-specific documentation will define messages that require a <TAN>, *A-80*
<INVSTMTRQ>	Aggregate for the investment statement download request (see section 13.9.1.2)
</INVSTMTRQ>	
</INVSTMTTRNRQ>	

13.9.1.2Investment Statement Request <INVSTMTRQ>

The following table shows the Investment Statement Request record. It is similar to a bank statement request, except that there are extra tags to indicate which pieces the user desires. Note that because transaction and position requests require date information, they use aggregates, whereas the other requests are elemental of type Boolean.

Clients and servers should interpret <DTSTART> and <DTEND> as described in Chapter 3.

If <DTASOF> is not included with the <INCPOS> aggregate, the server should return the most current position information available.

Tag	*Description and Type*
<INVSTMTRQ>	Investment-request aggregate
<INVACCTFROM>	Account-from aggregate
</INVACCTFROM>	
<INCTRAN>	Include-transactions aggregate (at most one)
<DTSTART>	Start date of request, *datetime*
<DTEND>	Ending date of request (at most one),.*datetime*
<INCLUDE>	Whether to include transactions in the statement download, *Boolean*
</INCTRAN>	
<INCOO>	Include investment open orders in response, *Boolean*
<INCPOS>	Include investment positions in response
<DTASOF>	Date that positions should be sent down for, *datetime*
<INCLUDE>	Whether to include positions in the statement download, *Boolean*
</INCPOS>	
<INCBAL>	Include investment balance in response, *Boolean*
</INVSTMTRQ>	

13.9.2 Investment Statement Response

13.9.2.1 Investment Statement Transaction Response <INVSTMTTRNRS>

Tag	*Description*
<INVSTMTTRNRS>	Transaction-response aggregate
<TRNUID>	Client-assigned globally unique ID for this transaction, *trnuid*
<STATUS>	Status aggregate
</STATUS>	
<CLTCOOKIE>	Client-provided data, **REQUIRED** if provided in request, *A-32*
<INVSTMTRS>	Aggregate for the investment statement download response (see section 13.9.2.2)
</INVSTMTRS>	
</INVSTMTTRNRS>	

13.9.2.2Investment Statement Response <INVSTMTRS>

The response can contain transaction, position, open order, and/or balance detail records; each in its own aggregate. The transaction list aggregate can contain a mixture of bank statement records and investment transactions, as specified below.

Tag	*Description*
<INVSTMTRS>	Investment-response aggregate
<DTASOF>	As of date & time for the statement download, *datetime*
<CURDEF>	Default currency for the statement, *currsymbol*
<INVACCTFROM>	Which account at FI
</INVACCTFROM>	
<INVTRANLIST>	Begin transaction list (at most one)
<DTSTART>	Start date for transaction data, *datetime*
<DTEND>	This is the value that should be sent in the next <DTSTART> request to insure that no transactions are missed, *datetime*
(investment transaction aggregates)	Investment statement transaction aggregates (zero or more) ; see section 13.9.2.4.4.
<INVBANKTRAN>	Banking-related transactions for the investment account (zero or more)
</INVBANKTRAN>	(See section 13.9.2.3)
</INVTRANLIST>	End of investment transaction list
<INVPOSLIST>	Beginning of investment position list (at most one)
<POSxxxxx>	Security type specific position aggregates (zero or more): POSMF, POSSTOCK, POSDEBT, POSOPT, POSOTHER
</POSxxxxx>	
</INVPOSLIST>	End of investment position list
<INVBAL>	Balances aggregate, see section 13.9.2.7
</INVBAL>	
<INVOOLIST>	Beginning of investment open order list (at most one)
<OOxxxxx>	Action and security type specific open order aggregates (zero or more): see section 13.9.2.5.2
</OOxxxxx>	
</INVOOLIST>	End of investment open order list
<MKTGINFO>	Marketing information (at most one), *A-360*.
</INVSTMTRS>	

The various sections of the investment statement download are returned only if requested.

13.9.2.2.1Note on Margin Calls

For investment statement download, margin call information should be included in the balances section. Margin call information should be contained in a <BAL> aggregate and included in the balance list <BALLIST>.

13.9.2.3Bank Transactions <INVBANKTRAN>

Use the INVBANKTRAN aggregate to download bank transactions in an investment statement download.

Tag	*Description*
<INVBANKTRAN>	Banking related transactions for the investment account
<STMTTRN>	Bank (cash) transaction aggregates
</STMTTRN>	(See chapter 11)
<SUBACCTFUND>	The sub-account associated with the funds for the transaction; see section 13.9.2.4.2
</INVBANKTRAN>	

13.9.2.4Investment Transactions

Note that the following types of investment actions found on statements should NOT be sent in Open Financial Exchange:

- Transaction-specific miscellaneous fees – fees that affect the basis of the transaction should be incorporated into the <COMMISSION>, <FEES>, <LOAD>, or <TAXES> amounts.
- Settlement actions.
- Sweeps, unless handled as any other investment position.

For transactions that involve securities, the client can create transactions based on the formula

total = (units * (unitprice +/- markup/markdown)) +/- (commission + fees + load + taxes)
(after adjusting quantity and unitprice to standard units based on the type of security.)

Thus, it is important the FIs incorporate all other transactional fees into the commission field. Clients can account for bond accrued interest and withholdings using separate client transactions.

13.9.2.4.1General Transaction Aggregate <INVTRAN>

The INVTRAN aggregate contains fields common to many of the investment transactions. It is referenced within the transaction aggregates in the following sections.

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

Tag	*Description*
<INVTRAN>	Investment-transaction-response aggregate
<FITID>	Unique FI-assigned transaction ID. This ID is used to detect duplicate downloads. *FITID*
<SRVRTID>	Server assigned transaction ID, *SRVRTID*
<DTTRADE>	Trade date; for stock splits, day of record, *datetime*
<DTSETTLE>	Settlement date; for stock splits, execution date, *datetime*
<MEMO>	Other information about transaction (at most one), *memo*
</INVTRAN>	

13.9.2.4.2Transaction Aggregate Elements

The following tags are referenced within of the following investment transaction aggregates.

Tag	*Description*
<ACCRDINT>	For debt purchases, accrued interest, *amount*
<AVGCOSTBASIS>	Average cost basis, *amount*
<BUYTYPE>	Type of purchase: BUY, BUYTOCOVER
<COMMISSION>	Transaction commission. *Amount*
<DENOMINATOR>	For stock splits, split ratio denominator, *quantity*
<DTPURCHASE>	The security's original purchase date, *date*
<GAIN>	For sales, total gain, *amount*
<FEES>	Fees applied to trade, *amount*
<FRACCASH>	Cash for fractional units., (used for stock splits), *amount*
<INCOMETYPE>	Type of investment income: CGLONG (capital gains-long term), CGSHORT (capital gains-short term), DIV (dividend), INTEREST, MISC
<LOAD>	Load on the transaction, *amount*
<MARKDOWN>	Portion of the unit price that is attributed to the dealer markdown, *unitprice*
<MARKUP>	Portion of the unit price that is attributed to the dealer markup, *unitprice*
<NEWUNITS>	For stock splits, number of shares after the split, *quantity*

<NUMERATOR>	For stock splits, split ratio numerator, *quantity*
<OLDUNITS>	For stock splits, number of shares before the split, *quantity*
<OPTACTION>	For options, action type: EXERCISE, ASSIGN, EXPIRE
<OPTBUYTYPE>	For options, type of purchase: BUYTOOPEN, BUYTOCLOSE
<OPTSELLTYPE>	For options, type of sell: SELLTOCLOSE, SELLTOOPEN
<POSTYPE>	Position type. LONG, SHORT
<RELFITID>	ID of related trade, *FITID*
<RELTYPE>	Related option transaction type: SPREAD, STRADDLE, NONE, OTHER
<SECURED>	How an option is secured: NAKED, COVERED
<SELLREASON>	Reason the sell of a debt security was generated: CALL (the debt was called), SELL (the debt was sold), MATURITY(the debt reached maturity)
<SELLTYPE>	Type of sell. SELL, SELLSHORT
<SHPERCTRCT>	For options, number of shares per contract, *N-5*
<SUBACCTFROM>	Sub-account that security or cash is being transferred from: CASH, MARGIN, SHORT, OTHER
<SUBACCTFUND>	Where did the money for the transaction come from or go to? CASH, MARGIN, SHORT, OTHER
<SUBACCTSEC>	Sub-account type for the security: CASH, MARGIN, SHORT, OTHER
<SUBACCTTO>	Sub-account that security or cash is being transferred to: CASH, MARGIN, SHORT, OTHER
<TOTAL>	Transaction total. Buys, sells, etc.:((quan. * (price +/- markup/markdown)) +/- (commission + fees + load + taxes)). Distributions, interest, margin interest, misc. expense, etc.: amount. Return of cap: cost basis; *amount*
<TAXES>	Taxes on the trade, *amount*
<TAXEXEMPT>	Tax-exempt transaction, *Boolean*
<TFERACTION>	Action for transfers: IN, OUT
<UNITPRICE>	Price per commonly-quoted unit. Does not include markup/markdown, *unitprice*. Share price for stocks, mutual funds, and others Percentage of par for bonds Per share (not contract) for options
<UNITS>	For security-based actions other than stock splits, *quantity* Shares for stocks, mutual funds, and others. Face value for bonds. Contracts for options.
<UNITTYPE>	Type of the units value: SHARES, CURRENCY
<WITHHOLDING>	Tax withholdings, *amount*

13.9.2.4.3 Investment Buy/Sell Aggregates <INVBUY>/<INVSELL>

These aggregates are referenced within investment transaction aggregates.

Aggregate Name	*Elements*
INVBUY	**<INVTRAN>** aggregate **<SECID>** aggregate **<UNITS>** **<UNITPRICE>** <MARKUP> <COMMISSION> <TAXES> <FEES> <LOAD> **<TOTAL>** <CURRENCY> aggregate <ORIGCURRENCY> aggregate **<SUBACCTSEC>** **<SUBACCTFUND>**
INVSELL	**<INVTRAN>** aggregate **<SECID>** aggregate **<UNITS>** **<UNITPRICE>** <MARKDOWN> <COMMISSION> <TAXES> <FEES> <LOAD> <WITHHOLDING> <TAXEXEMPT> **<TOTAL>** <GAIN> <CURRENCY> aggregate <ORIGCURRENCY> aggregate **<SUBACCTSEC>** **<SUBACCTFUND>**

13.9.2.4.4 Investment Transaction Aggregates

Aggregate Name	*Elements*	*Description*
<BUYDEBT>	**<INVBUY>** aggregate <ACCRDINT>	Buy debt security
<BUYMF>	**<INVBUY>** aggregate **<BUYTYPE>** <RELFITID>	Buy mutual fund The BUYTOCOVER buy type used to close short sales. RELFITID used to relate transactions associated with mutual fund exchanges.
<BUYOPT>	**<INVBUY>** aggregate **<OPTBUYTYPE>** **<SHPERCTRCT>**	Buy option The BUYTOOPEN buy type is like "ordinary" buying of option and works like stocks.
<BUYOTHER>	**<INVBUY>** aggregate	Buy other security type
<BUYSTOCK>	**<INVBUY>** aggregate **<BUYTYPE>**	Buy stock The BUYTOCOVER buy type used to close short sales.
<CLOSUREOPT>	**<INVTRAN>** aggregate **<SECID>** aggregate **<OPTACTION>** **<UNITS>** **<SHPERCTRCT>** **<SUBACCTSEC>** <RELFITID> <GAIN>	Close a position for an option. The EXERCISE action is used to close out an option that is exercised. The ASSIGN action is used when an option writer is assigned. The EXPIRE action is used when the option's expired date is reached. When the action is EXERCISE or ASSIGN another transaction must be generated by the server to represent the buy or sell of the underlying security. RELFITID refers to the transaction ID of the underlying buy or sell.
<INCOME>	**<INVTRAN>** aggregate **<SECID>** aggregate **<INCOMETYPE>** **<TOTAL>** **<SUBACCTSEC>** **<SUBACCTFUND>** <TAXEXEMPT> <WITHHOLDING> <CURRENCY> aggregate <ORIGCURRENCY> aggregate	Investment income is realized as cash into the investment account. A negative TOTAL is used to denote adjustments to income.
<INVEXPENSE>	**<INVTRAN>** aggregate <**SECID**> aggregate <**TOTAL**> <**SUBACCTSEC**> <**SUBACCTFUND**> <CURRENCY> aggregate <ORIGCURRENCY> aggregate	Misc. investment expense that is associated with a specific security. If the expense is associated with the account then an INVBANKTRAN - DEBIT should be used.
<JRNLFUND>	**<INVTRAN>** aggregate **<SUBACCTTO>** **<SUBACCTFROM>** **<TOTAL>**	Journaling cash holdings between sub-accounts within the same investment account.
<JRNLSEC>	**<INVTRAN>** aggregate **<SECID>** aggregate **<SUBACCTTO>** **<SUBACCTFROM>** **<UNITS>**	Journaling security holdings between sub-accounts within the same investment account.

<MARGININTEREST>	**<INVTRAN>** aggregate **<TOTAL>** **<SUBACCTFUND>** <CURRENCY> aggregate <ORIGCURRENCY> aggregate	Margin interest expense
<REINVEST>	**<INVTRAN>** aggregate **<SECID>** aggregate **<INCOMETYPE>** **<TOTAL>** **<SUBACCTSEC>** **<UNITS>** **<UNITPRICE>** <COMMISSION> <TAXES> <FEES> <LOAD> <TAXEXEMPT> <CURRENCY> aggregate <ORIGCURRENCY> aggregate	Reinvestment of income
<RETOFCAP>	**<INVTRAN>** **<SECID>** **<TOTAL>** **<SUBACCTSEC>** **<SUBACCTFUND>** <CURRENCY> aggregate <ORIGCURRENCY> aggregate	Return of capital
<SELLDEBT>	**<INVSELL>** aggregate **<SELLREASON>** <ACCRDINT>	Sell debt security. Used when debt is sold, called, or reached maturity.
<SELLMF>	**<INVSELL>** aggregate **<SELLTYPE>** <AVGCOSTBASIS> <RELFITID>	Sell mutual fund RELFITID used to relate transactions associated with mutual fund exchanges.
<SELLOPT>	**<INVSELL>** aggregate **<OPTSELLTYPE>** **<SHPERCTRCT>** <RELFITID> <RELTYPE> <SECURED>	Sell option The SELLTOCLOSE action is selling a previously bought option. The SELLTOOPEN action is writing an option
<SELLOTHER>	**<INVSELL>** aggregate	Sell other type of security
<SELLSTOCK>	**<INVSELL>** aggregate **<SELLTYPE>**	Sell stock
<SPLIT>	**<INVTRAN>** aggregate **<SECID>** aggregate **<SUBACCTSEC>** **<OLDUNITS>** **<NEWUNITS>** **<NUMERATOR>** **<DENOMINATOR>** <CURRENCY> aggregate <ORIGCURRENCY> aggregate <FRACCASH> <SUBACCTFUND>	Stock or Mutual Fund Split **Note**: the trade date is interpreted as the “day of record” for the split.

<TRANSFER>	**<INVTRAN>** aggregate **<SECID>** aggregate **<SUBACCTSEC>** **<UNITS>** **<TFERACTION>** **<POSTYPE>** <INVACCTFROM> aggregate <AVGCOSTBASIS> <UNITPRICE> <DTPURCHASE>	Transfer holdings in and out of the investment account.

13.9.2.4.5Valid Transactions by Security Type

	Debt	Mutual Fund	Option	Other	Stock
BUYDEBT	⌋				
BUYMF		⌋			
BUYOPT			⌋		
BUYOTHER				⌋	
BUYSTOCK					⌋
CLOSUREOPT			⌋		
INCOME	⌋	⌋		⌋	⌋
INVEXPENSE	⌋	⌋	⌋	⌋	⌋
JRNLFUND					
JRNLSEC	⌋	⌋	⌋	⌋	⌋
MARGININTEREST					
REINVEST	⌋	⌋		⌋	⌋
RETOFCAP	⌋	⌋	⌋	⌋	⌋
SELLDEBT	⌋				
SELLMF		⌋			
SELLOPT			⌋		
SELLOTHER				⌋	
SELLSTOCK					⌋
SPLIT		⌋			⌋
TRANSFER	⌋	⌋	⌋	⌋	⌋

Since JRNLFUND and MARGININTEREST do not refer to securities, there are no checks in any of the security columns for these transactions.

13.9.2.4.6Notes on Mutual Fund Exchanges

In investment statement download, two transactions are needed to reflect mutual fund exchanges. A SELLMF should be generated for the mutual fund being switched from and a BUYMF should be generated for the mutual fund being switched to. You can use the RELFITID tag to link these two transactions to each other. You should use the MEMO tag of the individual transactions to explain that a mutual fund exchange occurred.

13.9.2.4.7Notes on Corporate Actions

Since corporate actions can often be very complicated, it is difficult to define a single action aggregate that encompasses all possible scenarios. Instead, you should describe corporate actions using one or more of the provided basic action types. You should use the memo field of the individual transactions to link transactions to an encompassing corporate action.

13.9.2.4.8Notes on Option Splits

When the underlying security for an option splits, a new security is generated for the option since the strike price changes. In investment statement download, you need two transactions to reflect this activity. There should be a TRANSFER transaction to show that the old option security is removed from the account and another TRANSFER transaction to show that the new option security is moved into the account.

13.9.2.4.9Notes on Option actions

For options, the overall sequence of actions is as follows:

- For an option writer:

  Position is opened with Sell to Open.

  Position is closed with one of the following:
  - Buy to Close
  - Expire
  - Assigned

- For an option buyer:

  Position is opened with Buy to Open.

  Position is closed with one of the following:
  - Sell to Close
  - Expire
  - Exercise

13.9.2.5Open Orders

13.9.2.5.1General Open Order Aggregate <OO>

The <OO> aggregate contains fields common to all open orders. Use this aggregate to define the open order aggregates as show in the following section.

Tag	*Description*
<OO>	General-open-order aggregate
<FITID>	Unique FI-assigned transaction ID, *FITID*
<SRVRTID>	Unique server-assigned transaction ID, *SRVRTID*
<SECID>	Security identified aggregate
</SECID>	
<DTPLACED>	Date-time the order was placed, *datetime*
<UNITS>	Quantity of the security the open order is for, *unitprice*
<SUBACCT>	Sub-account type. CASH, MARGIN, SHORT, OTHER
<**DURATION**>	How long the order is good for: DAY, GOODTILCANCEL, IMMEDIATE
<**RESTRICTION**>	Special restriction on the order: ALLORNONE, MINUNITS, NONE
<MINUNITS>	Minimum number of units that must be filled for the order, *quantity*
<LIMITPRICE>	Limit price, *unitprice*
<STOPPRICE>	Stop price, *unitprice*
<MEMO>	Other information about order (at most one), *memo*
<CURRENCY>	Overriding currency aggregate
</CURRENCY>	
</OO>	

NOTE: *An open order is assumed to be a market order if no limit price or stop price is specified.*

13.9.2.5.2 Investment Open Order Aggregates

Open Order Aggregates	*Elements*	*Description of Elements*
<OOBUYDEBT>	**<OO>** aggregate	
	<AUCTION>	Whether the debt should be purchased at the auction, *Boolean*
	<DTAUCTION>	Date of the auction, *date*
<OOBUYMF>	**<OO>** aggregate	
	<BUYTYPE>	Type of purchase: BUY, BUYTOCOVER.
	<UNITTYPE>	What the units represent: SHARES, CURRENCY
<OOBUYOPT>	**<OO>** aggregate	
	<OPTBUYTYPE>	Type of purchase: BUYTOOPEN, BUYTOCLOSE
<OOBUYOTHER>	**<OO>** aggregate	
	<UNITTYPE>	What the units represent: SHARES, CURRENCY
<OOBUYSTOCK>	**<OO>** aggregate	
	<BUYTYPE>	Type of purchase: BUY, BUYTOCOVER
<OOSELLDEBT>	**<OO>** aggregate	
<OOSELLMF>	**<OO>** aggregate	
	<SELLTYPE>	Type of sale: SELL, SELLSHORT
	<UNITTYPE>	What the units represent: SHARES, CURRENCY.
	<SELLALL>	Sell entire holding, *Boolean*
<OOSELLOPT>	**<OO>** aggregate	
	<OPTSELLTYPE>	Type of sale: SELLTOOPEN, SELLTOCLOSE
<OOSELLOTHER>	**<OO>** aggregate	
	<UNITTYPE>	What the units represent: SHARES, CURRENCY
<OOSELLSTOCK>	**<OO>** aggregate	
	<SELLTYPE>	Type of sale: SELL, SELLSHORT
<SWITCHMF>	**<OO>** aggregate	
	<SECID> aggregate	Security ID of the mutual fund to switch to or purchase
	<UNITTYPE>	What the units represent: SHARES, CURRENCY
	<SWITCHALL>	Switch entire holding, *Boolean*

13.9.2.6Investment Positions

Position records represent a user's current positions, regardless of the transactional history. Prices and values should be the most recent available, even if different from a transaction price on the same day.

In position records, securities are identified as being either short or long. Because each FI has different rules regarding which sub-accounts can be used for short compared to long activity, FIs must explicitly indicate the type of position in addition to specifying the sub-account where the position takes place.

For options, position type SHORT is equivalent to WRITING an option, and position type LONG is equivalent to HOLDING an option. For security types where there is only one type (for example, bonds), use LONG.

13.9.2.6.1General Position Information <INVPOS>

The INVPOS aggregate contains fields relevant to all investment position types. It is included in the position aggregates as shown in the following sections.

Tag	*Description*
<INVPOS>	General-position aggregate
<SECID>	Security identifier
</SECID>	
<HELDINACCT>	Sub-account type CASH, MARGIN, SHORT, OTHER
<POSTYPE>	SHORT = Writer for options, Short for all others. LONG = Holder for options, Long for all others.
<UNITS>	For stocks, MFs, other, number of shares held. Bonds = face value. Options = number of contracts *quantity*
<UNITPRICE>	For stocks, MFs, other, price per share. Bonds = percentage of par Option = premium per share of underlying security *unitprice*
<MKTVAL>	Market value of this position, *amount*
<DTPRICEASOF>	Date and time of unit price and market value. Can be 0 if unit price and market value are unknown, *datetime*
<CURRENCY>	Currency information if different from default currency.
</CURRENCY>	
<MEMO>	Comment, *memo*
</INVPOS>	

13.9.2.6.2Investment Positions

Investment Position Aggregates	*Elements*	*Description of Elements*
<POSDEBT>	**<INVPOS>** aggregate	
<POSMF>	**<INVPOS>** aggregate	
	<UNITSSTREET>	Units in the FI's street name, *quantity*
	<UNITSUSER>	Units in the user's name directly, *quantity*
	<REINVDIV>	Reinvest dividends, *Boolean*
	<REINVCG>	Reinvest capital gains, *Boolean*
<POSOPT>	**<INVPOS>** aggregate	
	<SECURED>	How the option is secured. NAKED, COVERED.
<POSOTHER>	**<INVPOS>** aggregate	
<POSSTOCK>	**<INVPOS>** aggregate	
	<UNITSSTREET>	Units in the FI's street name, *quantity*
	<UNITSUSER>	Units in the user's name directly, *quantity*
	<REINVDIV>	Reinvest dividends, *Boolean*

13.9.2.7Investment Balances <INVBAL>

The <INVBAL> aggregate contains five specified balances. It can also contain a <BALLIST> aggregate that contains one or more <BAL> aggregates. The <BAL> aggregate (see Chapter 3) allows an FI to send any number of balances to the user, complete with description and Help text. The intent is to capture the same type of balance information present on the first page of many FI brokerage statements. You can also use the <BAL> aggregate to send margin call information.

Tag	*Description*
<INVBAL>	Balances aggregate
<AVAILCASH>	Cash balance across all sub-accounts. Should include sweep funds. *amount*
<MARGINBALANCE>	Margin balance. A positive balance indicates a positive cash balance, while a negative balance indicates the customer has borrowed funds. *amount*
<SHORTBALANCE>	Market value of all short positions, *amount*
<BUYPOWER>	Buying power, *amount*
<BALLIST>	Beginning of Investment balance list (at most one)
<BAL>	Balance aggregates (one or more)
</BAL>	See Chapter 3
</BALLIST>	
</INVBAL>	

13.9.2.8Status 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)
12250	Investment transaction download not supported (WARN)
12251	Investment position download not supported (WARN)
12252	Investment positions for specified date not available (WARN)
12253	Investment open order download not supported (WARN)
12254	Investment balances download not supported (WARN)

13.10Investment E-Mail

Open Financial Exchange currently defines one investment 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 their accounts. The server acknowledges receipt of the message. The FI prepares the response that the client picks up when it synchronizes with the server. E-mail is subject to synchronization, using <INVMAILSYNCRQ> / <INVMAILSYNCRS>.

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

13.10.1Investment E-Mail Request and Response

13.10.1.1Request <INVMAILRQ>

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

Tag	*Description*
<INVMAILRQ>	Investment-e-mail-request aggregate
<INVACCTFROM>	Account-from aggregate
</INVACCTFROM>	
<MAIL>	To, from, message information, see section 9.2.2
</MAIL>	
</**INVMAILRQ**>	

13.10.1.2Response <INVMAILRS>

Tag	*Description*
<INVMAILRS>	Investment-e-mail-response aggregate
<INVACCTFROM>	Account-from aggregate
</INVACCTFROM>	
<MAIL>	To, from, message information, see section 9.2.2
</MAIL>	
</**INVMAILRS**>	

13.10.1.3Status 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)

13.10.2Investment E-Mail Synchronization

13.10.2.1Request <INVMAILSYNCRQ>

Tag	*Description*
<INVMAILSYNCRQ > 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*
<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*
<INVACCTFROM>	Investment account of interest; token must be interpreted in terms of this account
</INVACCTFROM>	
<INVMAILTRNRQ> </INVMAILTRNRQ>	Investment-mail transactions (0 or more)
</INVMAILSYNCRQ>	

13.10.2.2Response <INVMAILSYNCRS>

Tag	*Description*
<INVMAILSYNCRS>	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*
<INVACCTFROM>	Investment account of interest; token must be interpreted in terms of this account
</INVACCTFROM>	
<INVMAILTRNRS> </INVMAILTRNRS>	Investment-mail transactions (0 or more)
</INVMAILSYNCRS>	

13.11Complete Example

This example is for a user who requests an investment statement download for a single account.

The request file:

```
<OFX>                                   <!--Beginning of request data-->
 <SIGNONMSGSRQV1>
   <SONRQ>                              <!--Beginning of signon-->
      <DTCLIENT>19960828101000          <!--Datetime, 8/28/96, 10:10:00 am-->
      <USERID>123-45-6789               <!--User ID (e.g. SSN) -->
      <USERPASS>MYPASSWORD              <!--Password(SSL encrypts whole) -->
      <LANGUAGE>ENG                     <!--Use English for the response -->
      <APPID>MyApp                      <!--Client application ID-->
      <APPVER>0500                      <!--Version 5.0-->
   </SONRQ>                             <!--End of signon-->
 </SIGNONMSGSRQV1>
 <INVSTMTMSGSRQV1>
   <INVSTMTTRNRQ>                       <!--First request in file-->
      <TRNUID>1001                      <!--Unique ID for this request-->
      <INVSTMTRQ>                       <!--Beginning of statement download-->
         <INVACCTFROM>                  <!--Identify the account: -->
            <BROKERID>121099999         <!--FI ID-->
            <ACCTID>999988              <!--Account number-->
         </INVACCTFROM>                 <!--End of account ID-->
         <INCTRAN>                      <!--Request transactions-->
            <DTSTART>19960824130105     <!--Send transactions posted after-->
                                        <!--Aug 24, 1996 1:01:05pm-->

            <INCLUDE>Y                  <!--Include transactions -->
         </INCTRAN>
         <INCOO>Y                       <!--Include open orders in response-->
         <INCPOS>                       <!--Request positions -->
           <INCLUDE>Y                   <!--Include current positions -->
         </INCPOS>
         <INCBAL>Y                      <!--Include balances in request-->
      </INVSTMTRQ>
   </INVSTMTTRNRQ>                      <!--End of first request-->
 </INVSTMTMSGSRQV1>
</OFX>                                  <!--End of Open Financial Exchange
                                                request data-->
```

A typical server response:

This user has one investment transaction, one bank transaction, one open order, two position entries, and one balance entry. The user deposits some money and buys shares in Acme. The user has an open limit order to buy 100 shares of Hackson Unlimited at $50/share. The holdings show the user already had 100 shares of Acme and now has 200 shares. The user also has one option contract to sell Lucky Airlines shares, bought before this download.

```
<OFX>                                   <!--Beginning of request data-->
 <SIGNONMSGSRSV1>
   <SONRS>                              <!--Beginning of signon-->
     <STATUS>
         <CODE>0                        <!--0 = OK-->
         <SEVERITY>INFO
      </STATUS>
      <DTSERVER>19960828101003          <!--Timestamp, 8/28/96, 10:10:03 am-->
         <LANGUAGE>ENG                        <!--English is used in this
response-->
```

```
      <DTPROFUP>19960101120000          <!--Profile updated, 1/1/96, 12 am-->
         <DTACCTUP>19960101120000       <!--Account info updated,1/1/96,12am-->
      </SONRS>                                <!--End of signon-->
</SIGNONMSGSRSV1>
<INVSTMTMSGSRSV1>
   <INVSTMTTRNRS>                       <!--First request in file-->
      <TRNUID>1001                      <!--Client ID for this request-->
      <STATUS>
         <CODE>0                        <!--0 = accepted, good data follows-->
         <SEVERITY>INFO
      </STATUS>
      <INVSTMTRS>                       <!--Beginning of statement download-->
         <DTASOF>19960827010000         <!--Statement as of Aug 27, 1996 1am-->
         <CURDEF>USD                    <!--Default currency is US Dollar-->
         <INVACCTFROM>                  <!--Beginning of account information-->
            <BROKERID>121099999         <!--FI ID-->
            <ACCTID>999988              <!--Account number-->
         </INVACCTFROM>                 <!--End of account information-->
         <INVTRANLIST>                  <!--Beginning of transactions-->
            <DTSTART>19960824130105     <!--Send transactions posted after-->
                                        <!--Aug 24, 1996 1:01:05pm-->
            <DTEND>19960828101000       <!--End timestamp (now) -->
            <BUYSTOCK>                  <!--Buy stock transaction-->
               <INVBUY>
                  <INVTRAN>
                     <FITID>23321       <!--FI transaction ID-->
                     <DTTRADE>19960825  <!--Trade date Aug 25, 1996-->
                     <DTSETTLE>19960828 <!--Settlement date Aug 28, 1996-->
                  </INVTRAN>
                  <SECID>               <!--Security ID-->
                     <UNIQUEID>123456789 <!--CUSIP for ACME -->
                     <UNIQUEIDTYPE>CUSIP
                  </SECID>
                  <UNITS>100            <!--100 shares-->
                  <UNITPRICE>50.00      <!--$50/share-->
                  <COMMISSION>25.00     <!--$25 commission -->
                  <TOTAL>5025.00        <!--Total amount $5025.00-->
                  <SUBACCTSEC>CASH      <!--Holding resides in cash account-->
                  <SUBACCTFUND>CASH     <!--Bought in cash account-->
               </INVBUY>
               <BUYTYPE>BUY             <!--Normal buy-->
            </BUYSTOCK>                 <!--End of buy stock transaction-->
            <INVBANKTRAN>               <!--Investment acct bank transaction-->
               <STMTTRN>                <!--Beginning of a bank transaction-->
                  <TRNTYPE>CREDIT       <!--Generic credit-->
                  <DTPOSTED>19960825    <!--Aug 25, 1996-->
                  <DTUSER>19960825      <!--Aug 25, 1996-->
                  <TRNAMT>1000.00       <!--$1,000.00-->
                  <FITID>12345          <!--FI transaction ID 12345-->
                  <NAME>Customer deposit <!--Description of transaction-->
                  <MEMO>Your check #1034 <!--Optional memo from FI-->
               </STMTTRN>               <!--End of bank transaction-->
               <SUBACCTFUND>CASH        <!--Credited to the cash account -->
            </INVBANKTRAN>
         </INVTRANLIST>                 <!--End of transactions-->
         <INVPOSLIST>                   <!--Beginning of positions list-->
            <POSSTOCK>                  <!--Beginning of position -->
               <INVPOS>
                  <SECID>               <!--Security ID-->
                     <UNIQUEID>123456789 <!--CUSIP for Acme Development, Inc.-->
                     <UNIQUEIDTYPE>CUSIP
                  </SECID>
                  <HELDINACCT>CASH      <!--Cash account-->
```

```
            <POSTYPE>LONG           <!--Long position-->
            <UNITS>200              <!--200 shares-->
            <UNITPRICE>49.50        <!--Latest price-->
            <MKTVAL>9900.00         <!--Current market value $9900.00-->
            <DTPRICEASOF>19960827010000 <!--Prices as of Aug27,1996 1am-->
            <MEMO>Next dividend payable Sept 1
          </INVPOS>
        </POSSTOCK>                 <!--End of position-->
        <POSOPT>                    <!--Beginning of position-->
          <INVPOS>
            <SECID>                 <!--Security ID-->
              <UNIQUEID>000342222  <!--CUSIP for the option -->
              <UNIQUEIDTYPE>CUSIP
            </SECID>
            <HELDINACCT>CASH        <!--Cash account-->
            <POSTYPE>LONG           <!--Long position-->
            <UNITS>1                <!--100 shares-->
            <UNITPRICE>5            <!--Latest price-->
            <MKTVAL>500             <!--Current market value $500.00-->
            <DTPRICEASOF>19960827010000 <!--Prices as of Aug27,1996 1am-->
            <MEMO> Option is in the money
          </INVPOS>
        </POSOPT>                   <!--End of option position -->
      </INVPOSLIST>                 <!--End of position -->
      <INVBAL>
        <AVAILCASH>200.00           <!--$200.00 cash balance-->
        <MARGINBALANCE>-50.00       <!--$50.00 owed on margin balance-->
        <SHORTBALANCE>0             <!--$0 short balance-->

        <BALLIST>                   <!--Beginning of FI-defined balances-->
          <BAL>                     <!--Beginning of a balance-->
            <NAME>Margin Interest Rate <!--Name of balance entry-->
            <DESC>Current interest rate on margin balances
                                    <!--Help text for this balance-->
            <BALTYPE>PERCENT        <!--Format as percent-->
            <VALUE>7.85             <!--Will be formatted 7.85%-->
            <DTASOF>19960827010000  <!--Rate as of Aug 27, 1996 1am-->
          </BAL>                    <!--End of balance entry-->
        </BALLIST>                  <!--End of balances-->
      </INVBAL>
      <INVOOLIST>
        <OOBUYSTOCK>
          <OO>
            <FITID>23321            <!--FI transaction ID-->
            <SECID>                 <!--Security ID-->
              <UNIQUEID>666678578  <!--CUSIP for Hackson Unlimited-->
              <UNIQUEIDTYPE>CUSIP
            </SECID>
            <DTPLACED>1996062431505 <!--Order placed 6/24/96 3:15:05pm-->
            <UNITS>100              <!--100 shares-->
            <SUBACCT>CASH           <!--Purchase with cash-->
            <DURATION>GOODTILCANCEL <!--GOODTILCANCEL-->
            <RESTRICTION>NONE       <!--No special restrictions-->
          <LIMITPRICE>50.00         <!--Limit price $50/share-->
          </OO>
          <BUYTYPE>BUY              <!--Normal buy-->
        </OOBUYSTOCK>
      </INVOOLIST>
    </INVSTMTRS>
  </INVSTMTTRNRS>                   <!--End of first response-->
</INVSTMTMSGSRSV1>
<SECLISTMSGSRSV1>
  <SECLIST>                         <!--Beginning of securities list-->
```

```
      <STOCKINFO>                        <!--Beginning of 1st security ID-->
        <SECINFO>
          <SECID>                        <!--Security ID-->
            <UNIQUEID>123456789          <!--CUSIP for the stock -->
            <UNIQUEIDTYPE>CUSIP
          </SECID>
          <SECNAME>Acme Development, Inc.
          <TICKER>ACME                   <!--Ticker symbol-->
          <FIID>1024                     <!--FI internal security identifier-->
        </SECINFO>
        <YIELD>10                        <!--$10/share/year-->
        <ASSETCLASS>SMALLSTOCK           <!--Small Capital Stock asset class-->
      </STOCKINFO>                       <!--End of security ID-->
      <STOCKINFO>
        <SECINFO>
          <SECID>                        <!--Security ID-->
            <UNIQUEID>666678578          <!--CUSIP for the stock -->
            <UNIQUEIDTYPE>CUSIP
          </SECID>
          <SECNAME> Hackson Unlimited, Inc.
          <TICKER> HACK                  <!--Ticker symbol-->
          <FIID>1027                     <!--FI internal security identifier-->
        </SECINFO>
        <YIELD>17                        <!--$10/share/year-->
        <ASSETCLASS>SMALLSTOCK           <!--Small Capital Stock asset class-->
      </STOCKINFO>
      <OPTINFO>                          <!--End of security ID-->
        <SECINFO>
          <SECID>                        <!--Security ID-->
            <UNIQUEID>000342222          <!--CUSIP for the option -->
            <UNIQUEIDTYPE>CUSIP
          </SECID>
          <SECNAME>Lucky Airlines Jan 97 Put
          <TICKER>LUAXX                  <!--Ticker symbol-->
          <FIID>0013                     <!--FI internal security identifier-->
        </SECINFO>
        <OPTTYPE>PUT
        <STRIKEPRICE>35.00               <!--Strike price $35/share-->
        <DTEXPIRE>19970121               <!--Option expires Jan 21, 1997-->
        <SHPERCTRCT>100                  <!--100 shares per contract-->
        <SECID>                          <!--Security ID-->
          <UNIQUEID>000342200            <!--CUSIP for the underlying stock -->
          <UNIQUEIDTYPE>CUSIP
        </SECID>
        <ASSETCLASS>LARGESTOCK           <!--Large Capital Stock asset class-->
      </OPTINFO>                         <!--End of option information-->
    </SECLIST>                           <!--End of securities list-->
  </SECLISTMSGSRSV1>
</OFX>                                   <!--End of Open Financial Exchange
                                                request data-->
```