Interactive Financial Exchange

Version 1.4.0

- Public Review -

Business Message Specification
18 February 2003

Disclaimer

The IFX Forum makes no warranties whatsoever with respect to the contents of this specification. Without limitation, the IFX Forum makes no warranty (i) that the information contained in the specification is accurate, error-free or describes a practically realizable product or service, or (ii) that the product or service described in the specification can be produced or provided without infringing third-party rights or violating applicable laws or regulations.

RESERVATION OF RIGHTS: The contents of this specification are protected by copyright and other intellectual property laws. The IFX Forum expressly reserves all rights in such content.

Table of Contents

1 Overview

1.1 Introduction

The Interactive Financial Exchange (IFX) Specification provides a robust and scalable framework for the exchange of financial data and instructions independent of a particular network technology or computing platform. The information-sharing potential of IFX has been designed to support communication not only between a Financial Institution and its customers, but also between a Financial Institution and its Service Providers, between Financial Institutions, and eventually directly between customers (e.g., "electronic wallet"). This specification supports existing and emerging financial services and is extensible and customizable for future growth.

The IFX Specification has been developed as a cooperative industry effort among major Financial Institutions, Service Providers, and information technology vendors to these institutions and their customers in the small business and consumer markets. This specification builds on previous industry experience, including the OFX and Gold specifications that are currently implemented by major Financial Institutions and Service Providers to enable the electronic exchange of financial data between them and their customers. The IFX Specification provides a comprehensive specification for new financial industry services and software while providing a common strategic direction for the evolution of existing products and services.

The IFX Specification provides Financial Institutions, their Service Provider vendors, and financial software developers for the small business and consumer markets with a generalized model for financial industry data communications. This generalized model consists of two parts:

1. A business-level Message Specification and its associated Data Dictionary, which are contained in Volumes 1 and 2, and

2. Implementation specifications, which are available separately at the IFX Forum website.

1.1.1 The IFX Business Message Specification

This document provides the financial services industry with a common set of *Business Messages* that may be used to provide message-processing services across multiple organizations and networks. This document defines the information that must be sent in a message request and in a message response from a business perspective and provides message semantics for each request and response pair. This document provides the basis for business-level information flow between Financial Institutions, their customers, and third-party Service Providers.

For example, an individual account holder may bank from home using a secure connection over the public Internet to connect to their Financial Institution. The same request message sent from the personal computer in the home may enter the Financial Institution's private data network for processing or be routed to a third-party Service Provider that processes that message on behalf of the Financial Institution. Regardless of the organization that actually processes the message or what computing and network architecture that organization has installed, the customer receives a response message with standard semantics.

1.1.2 IFX Implementation Specifications

Complementary documents called *Interactive Financial Exchange Implementation Specifications* are expected to provide additional detail on how the business messages defined in this document may be physically represented. Each IFX Implementation Specification is an agreement between vendors and the financial services industry on how to implement the business messages defined in this document. IFX Implementation Specifications also provide the basis for interworking among products and services from various vendors and custom software developed by Financial Institutions and Service Providers.

1.2 Design Principles

The IFX Specification has been designed to meet the following principles:

Support a Broad Range of Financial Activities—The IFX Specification is envisioned to incorporate the broad feature sets of the existing OFX and Gold specifications. Version 1.0 provides the following functions:

- Bank account balances, account information, statement download for deposit and loan accounts;
- Credit card statement download;
- Funds transfers, including recurring transfers;
- Consumer payments, including recurring payments;
- Business payments, including recurring payments;
- Bill presentment; and
- Customer service.

Support Financial Communications Among a Broad Range of Parties—The IFX Specification supports financial communications among:

- Banks
- Brokerage houses (future)
- Service Providers
- Financial advisors (future)
- Small businesses
- Consumers

Support a Broad Range of Client Devices—The IFX Specification allows Financial Institutions to support customers using a broad range of client devices, including, but not limited to:

- World Wide Web access using any standard Web Browser software,
- Personal Computers with Personal Financial Manager (PFM) software,
- Voice Response Units (VRUs) that provide Bank by Phone services,
- Automated Teller Machines (ATMs), and
- Consumer Handheld Devices such as Personal Digital Assistants (PDAs) or Mobile Telephones with data capabilities.

Support Customers Using Multiple Client Devices—The IFX Specification allows a customer to use multiple client devices to interact with a Financial Institution. All devices get the same data for that customer and provide the customer with a consistent experience. The IFX Specification contains requests, which allow an intelligent client to discover what messages the customer has performed using other client devices.

Flexible—The IFX Specification is designed to provide Financial Institutions and Service Providers with the flexibility they need to rapidly develop, test, and deploy new services. The specification is intended to specify the minimum necessary functionality to provide reliable interactions between systems owned and maintained by customers, their Financial Institutions, and their Service Providers.

Customizable—The IFX Specification allows Financial Institutions and Service Providers to add custom elements, aggregates, or entire messages to rapidly deploy new services or add functionality to existing services. Custom elements, aggregates, and messages should be registered with the IFX Specification governance organization to protect against name collisions. However, registration of custom objects does not obligate any organization to use them in order to be in compliance with the specification.

Extensible—The IFX Specification has been designed to allow for constant evolution through the addition of standardized services that may be easily implemented by Financial Institutions and their customers.

Open—The IFX Specification is publicly available. Solutions for financial communications based on the specification may be built by anyone, independent of any specific technology, vendor, or Financial Institution. IFX is currently maintained through cooperative industry effort and will be evolved into a formal standard over time.

Robust—The IFX Specification is used for the execution of important financial messages and the communication of sensitive financial data. The specification provides customers with confirmation that messages have occurred as planned and notification when scheduled messages fail.

Secure—The IFX Specification provides a secure framework for the development of secure online financial services. IFX relies on industry standard mechanisms to provide secure channels between client and server and provides application-level authentication of customers and Financial Institution and Service Provider staff. Note that application-level authentication between different Service Providers for Electronic Bill Presentment and Payment may be provided through alternate means.

Support Batch and Interactive Sessions—The IFX Specification may be implemented using either batch or interactive session management. The business-level messages are not biased towards either batch or interactive sessions.

International Support—The IFX Specification incorporates significant features for international support, including multiple currencies, and languages. Additional requirements for international support will be addressed as the specification evolves and international requirements are identified.

Platform Independent—The IFX Specification makes no assumptions about the hardware or software available as a client or server. IFX may be implemented on any computing platform.

Transport Independent—The IFX Specification makes no assumptions about the network used for the transport of business-level messages. IFX Implementation Specifications provide details on transport for a specific type of network.

1.3 Benefits to Financial Institutions and Service Providers

Customer Acquisition and Retention—Standardization of message sets for financial message processing will remove technical barriers that currently limit an FI's ability to acquire customers for online banking services. These standards will level the competitive playing field for electronic delivery of financial services and allow Financial Institutions to serve new customers for whom appropriate channels have not previously existed. Broader vendor support for these message sets and the associated implementation specifications will stimulate end user adoption of online banking through the availability of a diverse set of client applications that interoperate with any FI that adopts this specification.

Flexibility—Because the semantics and syntax for the request and response message pair are defined end-to-end in IFX and the same message is passed regardless of the network technology, Financial Institutions and Service Providers are not locked into a single vendor or technology. They may find other ways to meet their business objectives if their service is not well supported by a vendor or network provider.

Manageability—Standardization of message sets for financial message processing will remove technical barriers that currently limit an FI's ability to acquire customers for online banking services. This common framework will provide Financial Institutions and their outsourced Service Providers with significant benefits in terms of their ability to manage the development and operation of a portfolio of services for their customers.

Cost Savings—The increased flexibility and manageability of solutions based on the IFX Specification will directly affect the profitability of an online financial service. Financial Institutions and Service Providers that invest in solutions based on IFX will see their investments returned quickly through operational cost savings.

1.4 Benefits to Consumers and Businesses

Reliability and Performance—Individual consumers and small businesses will receive benefits in reliability and performance through the increased integration of their software with a Financial Institution's systems and networks. The IFX Specification is the product of an unprecedented level of cooperation between developers of consumer and small business financial applications and financial industry high-volume message processing experts.

Consistency of Experience with a Financial Institution—Consumers of online financial services will notice significant improvements in the consistency of their interactions with the Financial Institution through multiple

channels as the organizations take advantage of the improved service manageability and flexibility enabled by use of the IFX.

Common Standard Across all Financial Institutions—Businesses and Consumers will also benefit from increased financial industry use of the IFX by their increased ability to manage their relationships with Financial Institutions using off-the-shelf software.

2 Structure

This section describes a number of important foundations of the IFX Specification, including terminology, data types, elements, aggregates, messages, services, authentication realms, naming conventions, versioning and governance, usage rules, and documentation conventions.

2.1 Definitions

The following terms are used extensively in IFX Specification documentation and may have very specific meanings within this context.

2.1.1 Biller

A *Biller* is a company or organization that sends a Bill or Statement to a customer, usually a request for payment for a product or service.

2.1.2 Biller Payment Provider (BPP)

A *Biller Payment Provider* (or *BPP*) is an agent (usually a financial institution) of the Biller that originates and accepts payments on behalf of the Biller.

2.1.3 Biller Service Provider (BSP)

A *Biller Service Provider* (or *BSP*) is an agent of the Biller that provides an electronic bill presentment and payment service for the Biller

2.1.4 Client

Client refers to the sender of an IFX Request Message. The client may be a computer system that a Customer is logged into, or it may be some kind of proxy device that is making IFX Requests on behalf of the Customer. This scenario is typical in the case of a Customer using a Web browser to perform financial messages. The Web server may be communicating using IFX Messages to some back end system. In this case, the Web server is considered a proxy client and the back end system is the server. It should be noted that clients might also be Service Providers for certain messages, such as the Pay/No-Pay message and Reverse Pay Decision message.

2.1.5 Customer

A *Customer* is an individual or small business that is a consumer of financial services provided by a Financial Institution.

2.1.6 Customer Payment Provider (CPP)

A *Customer Payment Provider* (or *CPP*) is an agent (usually a financial institution) of the Customer that originates payments on behalf of the Customer.

2.1.7 Customer Service Provider (CSP)

A *Customer Service Provider* (or *CSP*) is an agent of the Customer that provides an interface directly to customers, businesses, or others for bill presentment. A CSP enrolls customers, enables presentment, and provides customer care, among other functions.

2.1.8 Financial Institution (FI)

A *Financial Institution* (or *FI*) is an organization that provides branded financial services to customers. Financial Institutions develop and market financial services to individual and small business customers. Financial Institutions may serve as the processor for their own services or may choose to outsource processing to a Service Provider. In an effort to group the services different providers provide, the Financial Institution is also referred to as a Financial Institution in this document.

2.1.9   Provider

When the term *provider* is used generically, typically qualified by a service name (e.g., Pay provider), it refers to an organization that processes messages that support financial services provided to individual and small business customers. A provider may be either of the following:

- A Financial Institution that has chosen to perform its own message processing in support of its services *or*
- A Service Provider that performs message processing on behalf of a Financial Institution that has decided to outsource this function.

2.1.10   Server

Server refers to a system that receives IFX Request Messages and responds with IFX Response Messages. For clarity in the documentation, the server is assumed to be an endpoint that composes the entire response message and sends it back to the client. In the real world, systems may be implemented with multiple stages of message processing that are transparent to the client. As long as the complete response message is delivered to the client, the protocol will work as designed.

2.1.11   Service

Service specifically refers to a collection of related messages. For example, the Bank service encompasses banking messages such as requesting bank statement, initiating stop checks, etc. This definition correlates to an application-level concept of service that refers to a single function or a collection of similar functions that are branded and marketed to individuals and small businesses by a Financial Institution or Service Provider. Within this specification, each SP maintains a list of the services it supports. Services that are currently defined in this specification are Banking (Bank), Payment (Pay), Presentment (Pres), Valuable Media (Media), and Root (Root). Customer Service and a few other functions including Service Profile are grouped together in a chapter called Base Service (Base). Additional services may be defined and implemented by Financial Institutions or Service Providers as extensions to this specification.

2.1.12   Service Provider (SP)

A *Service Provider* (or *SP*) is an organization that provides services to an individual or to other organizations. An example of a Service Provider is a message processor for a Financial Institution that has chosen to outsource its message processing for a particular service. Service Providers typically provide services for multiple individuals or organizations. A Biller may be considered a Service Provider; however, in this document the term "Service Provider" is used to refer generically to an FI, CSP, CPP, BSP and/or BPP. Organizations that provide services to end user customers are referred to specifically as Financial Institutions or CSPs where possible.

> *Note:*   *The labels CSP, CPP, BSP and BPP are used to define a collection of functions and responsibilities. They do not necessarily refer to specific physical or business entities. Any entity wishing to perform a particular role needs to address the issues and responsibilities defined for that role. It is also understood that one entity may perform more than one role, or that an entity may wish to outsource one or more functions of a role to other entities. For example, some Billers may serve as their own BSPs, some Financial Institutions may perform the role of both CSP and CPP, and some CSPs may outsource functions such as customer care to other service providers. For more information about the responsibilities of the various roles, see the Electronic Bill Presentment and Payment Business Practices at http://www.nacha.org/billpay/businesspractices.htm.*

2.1.13   May, Should, and Must

The terms *may*, *should*, and *must* are used frequently within this specification.

May indicates that the described behavior is not required for IFX compliance, nor is there any preferred behavior. Generally, the behavior is described to inform of possible behaviors of which client or server designers and developers should be aware.

Should indicates that, while the described behavior is not required for IFX compliance, it *is* preferred. Generally, the behavior is described to provide a better experience for the communicating parties.

Must indicates that the described behavior is required for IFX compliance.

2.2   Documentation Conventions

Documentation conventions in the specification include:

- All documentation is written and maintained in US English.

- Required elements and aggregates have "Required" in the Usage column.
  - In a request message, "Required" means that the client must include the element.
  - In a response message, "Required" means that the server must return the element if the message is successful, with the exception of <CustId> and <RqUID>, which must always be returned. There are exceptions to the <CustId> requirement; those are explicitly called out in the message definitions within this specification.

- Optional elements and aggregates have "Optional" in the Usage column. If there are specific cases where an optional element or aggregate may be required, it is noted by "*but see Description*" in the Usage column, to call attention to this condition.

- For successful synchronous messages (i.e., responses with a <Status> <Severity> of Info or Warn) in a session where the client did not request suppression of echo (via <SuppressEcho> = True, see section 4.4.2.1) or the server does not support echo suppression, a server must echo all fields provided in the request that are indicated in the response by "Echoed" in the Usage column. For unsuccessful messages (i.e., responses with a <Status> <Severity> of Error), successful messages in a session where the client requested to suppress echo (via <SuppressEcho> = True, see section 4.4.2.1), or asynchronous responses, a server must echo <RqUID>, and, if present in the request, <AsyncRqUID> and <CustId>. For more information about synchronous responses and echoes, see section 3.2.12.1.

- Some elements and aggregates are required only if the Service Provider indicates this through the Service Profile for the service. These elements and aggregates have "Profiled requirement" in the Usage column.

- Some elements have valid values defined by the Service Profile. These elements have "Profiled values" in the Usage column.

- Some elements and aggregates are only valid if support for them is indicated in the Service Profile. These elements have "Profiled support" in the Usage column.

- Repeating elements and aggregates may appear more than once, and are indicated by "Repeating" in the Usage column.

- An "inclusive or" condition requires at least one of the members of the group to be present, and is indicated with "OR" in the Usage column. A thin dashed line separates the members of the OR, while a thicker dotted line indicates the boundaries of the entire OR group.

- An "exclusive or" condition requires one, and only one, of the members of the group to be present, and is indicated with "XOR" in the Usage column. A thin dashed line separates the members of the XOR, while a thicker dotted line indicates the boundaries of the entire XOR group.

- An "and" condition requires all or none of the members of the group to be present, and is indicated with a "AND" in the Usage column. A thin dashed line separates the members of the AND, while a thicker dotted line indicates the boundaries of the entire AND group.

- Indentation indicates that the indented element or aggregate is contained in the most recent aggregate that has one less indentation.

Tag	*Type*	*Usage*	*Description*
<Element1>	C-12	Required	Required Element.
			Occurs exactly once in each Request, or exactly once in each successful Response.
<Element1>	C-36	Optional	Optional Element.
			Occurs zero or one times.

Tag	Type	Usage	Description
\<Element1\>	C-12	Optional Repeating	Optional Repeating Element. Occurs zero or more times.
\<Aggregate1\>	Aggregate	Optional Profiled support	Profiled Aggregate. Occurs zero or one times. Usage is controlled by an entry in the Service Profile.
\<Element1\>	Open Enum	Optional Repeating Profiled values	Profiled Open Enum. Occurs zero or more times. Supported values are controlled by an entry in the Service Profile.
\<Aggregate1\>	Aggregate	Optional Echoed	Echo Aggregate. Occurs zero or one times. Must be included in the response message if it was received in the request.
\<Element1\>	Long	Optional AND	Optional and.
\<Element2\>	Decimal	Optional AND	Two valid cases (1) both \<Element1\> and \<Element2\>, (2) neither.
\<Element1\>	Long	Required OR	Inclusive or.
\<Element2\>	Decimal	Required OR	Three valid cases (1) \<Element1\>, (2) \<Element2\>, (3) both \<Element1\> and \<Element2\>.
\<Element1\>	Long	Optional XOR	Optional exclusive or.
\<Element2\>	Decimal	Optional XOR	Three valid cases: (1) neither \<Element1\> nor \<Element2\>. (2) \<Element1\>, but not \<Element2\>. (3) \<Element2\>, but not \<Element1\>.
\<Element1\>	Long	Required XOR	Required Exclusive or.
\<Element2\>	Decimal	Required XOR	Two valid cases: (1) \<Element1\>, but not \<Element2\>. (2) \<Element2\>, but not \<Element1\>.
\<Element1\>	Long	Required XOR	Required Exclusive or.
\<Element2\>	Decimal	Required XOR	Two valid cases: (1) \<Element1\>, but neither \<Element2\> nor \<Element3\>. (2) \<Element2\> *and* \<Element3\>, but not \<Element1\>.
\<Element3\>	C-10	Required	
\<Element1\>	Long	Required XOR	Required Exclusive or.
\<Element2\>	Decimal	Required XOR	Three valid cases: (1) \<Element1\>, but neither \<Element2\> nor \<Element3\>. (2) \<Element2\> *and* \<Element3\>, but not \<Element1\>. (3) \<Element2\>, but neither \<Element1\> nor \<Element3\>.
\<Element3\>	C-10	Optional	

2.3 Data Types

The IFX Specification is designed around a small number of data types that are used to represent all data passed between clients and servers using the messages defined in this specification. All information elements are based on these data types. Supported data types are:

Character	Closed Enum
Narrow Character	Open Enum
Binary	Long
Boolean	Identifier
YrMon, Date, Time, DateTime, and Timestamp	Phone Number
Decimal	Universally Unique Identifier (UUID)
Currency Amount	URL

IFX defines the semantics and logical attributes of each data type. The physical representations are defined within the implementation specifications.

2.3.1   Character

Character indicates an element that allows character data up to a maximum number of characters, regardless of the number of bytes required to represent each character. The number after the hyphen specifies the maximum number of characters. For example, C-12 specifies an element of characters with maximum length 12 characters. C-∞ indicates an element with no maximum length. It is expected that *character* type elements may contain multibyte representations of characters in some implementations, depending on the allowable character sets.

2.3.2   Narrow Character

Elements of type *Narrow Character* are elements of *character* data type with the additional restriction that the only allowable characters are those contained within the ISO Latin-1 character set.

2.3.3   Binary

The *Binary* data type is a compound type consisting of three logical elements:

Tag	Type	Usage	Description
<ContentType>	Open Enum	Optional	Specified in IETF RFC 2046.
<BinLength>	Long	Required	Identifies the size of the binary data in number of bytes.
<BinData>	Raw Binary Data	Required	Binary data.

2.3.4   Boolean

Boolean indicates a logical *True* or *False* condition. The physical representation of Boolean data is specified by each implementation specification corresponding to this message specification.

2.3.4.1   Boolean Conventions and Selection Criteria

The general usage of Booleans within the IFX specification is:

<TagName> (optional)

- If *True*, then…
- If *False* or omitted, then…

This usage of Booleans in IFX allows optional Booleans to be added in future revisions while maintaining upward compatibility.

There are instances where it is considered too error prone to allow a default of *False*. In these cases the Boolean is required.

The convention for the use of selection criteria for read messages is:

For Booleans:

<TagName> (optional)

- If *True*, then the selected set must include at least those items where the condition is *True*.
- If *False*, then the selected set must include at least those items where the condition is *False*.
- If omitted, then this item is ignored as a selection criterion.

Usage notes specify when a Boolean is used as a selection criterion.

For others:

<TagName> (optional)

- If present, the selected set must include at least those items where the value in the request matches a value for the corresponding tag in the searched region.
- If absent, the selected set must include items with any value, including NULL, for the corresponding tag in the searched region.

2.3.5 YrMon, Date, Time, DateTime, and Timestamp

The IFX Specification includes five time-related compound data types: YrMon, Date, Time, DateTime, and Timestamp. In all types that describe Date information, IFX 1.0.1 uses the Gregorian calendar. Other calendars may be considered for future versions as requirements are identified. Data types including time information refer to a 24-hour clock.

All date and time types include (with the largest units given first): year, month, day, hour, minute, second, and fractions of a second. Any particular type may include a subset of these possible values. Types including time information (hour, minute, etc.) may also include an offset from Coordinated Universal Time (UTC).

> *Note: In a DateTime element, specifying a date without a time or time zone will result in a time of midnight, UTC. This will result in the previous date appearing for all time zones in the western hemisphere. For example, October 5, 2002, without a specified time zone offset, will appear to be October 5, 2002, for the eastern hemisphere, but October 4, 2002, for the western hemisphere. Therefore, for DateTime elements where a single date is desired worldwide, the time must be included, and it must result in noon, UTC (e.g., "12:00:00" or "09:00:00-03:00").*

> *Note: IFX does not require servers or clients to use the full precision specified. However, they are required to accept any of these forms without complaint.*

As a general rule for *date* and *time* compound data types, values may be entered that omit the smallest logical elements. In every case, the value is taken to mean the same thing as if the minimum values (such as zeroes) were included. (The default is always the start of an otherwise ambiguous range for types other than *YrMon*.) For example, a *DateTime* value omitting the time portion means the start of the day (12:00 midnight). Note that time zone qualifiers (in *time* and *DateTime* values) are an exception to this rule, as they may be included even if times are not specified to the millisecond.

The logical elements appearing in each of these compound data types are summarized below. "Required" means that the element must occur in all instances of the data type. "Recommended" means that the element should be included in all instances of the data type. "Optional" elements may be omitted from an instance of the data type. Optional elements must be included if smaller elements are to be included. For example, month must not be omitted from a *date* value if day is included.

	Contains	YrMon	Date	Time	DateTime	Timestamp
Year	YYYY 0000-9999	Required	Required	N/A	Required	Required
Month	MM 1-12	Required	Optional	N/A	Required	Required
Day	DD 1-31	N/A	Optional	N/A	Required	Required

	Contains	YrMon	Date	Time	DateTime	Timestamp
Hours	HH 0-23	N/A	N/A	Required	Optional	Required
Minutes	MM 0-59	N/A	N/A	Optional	Optional	Required
Seconds	SS 0-60	N/A	N/A	Optional	Optional	Required
Fractional Seconds	XXX (minimum) Precision is determined by the implementation	N/A	N/A	Optional	Optional	Optional
UTC offset Hours	HH 0-23	N/A	N/A	Recommended	Recommended	Recommended
UTC offset Minutes (time zone indication)	MM 0-59	N/A	N/A	Recommended	Recommended	Recommended

2.3.5.1　YrMon

Elements of data type *YrMon* contain an indication of a particular month. This data type describes a unique period of time (not a repeating portion of every year). This may (for example) be used to describe an expiration date for a credit card. In that case, the period represented is actually a shorthand for the last day (or millisecond or whatever minimum interval) of the month described.

Logically, the *YrMon* data type must contain a month and 4-digit year. Unlike the other date and type types, values of this type must **not** be shortened by any omission of its portions: Both the year and month are required.

YrMon Data type	Type	Usage	Description
<Year>	Long	Required	4-digit year value.
<Month>	Long	Required	Number of the represented month. Value must be within the range 1 through 12.

2.3.5.2　Date

Elements of data type *Date* contain an indication of a particular day. This data type describes a unique period of time, normally 24 hours (not a repeating portion of every year).

Logically, this data type must contain a 4-digit year, and may contain a month number, and day number.

Date Data type	Type	Usage	Description
<Year>	Long	Required	4-digit year value.
<Month>	Long	Optional but see Description	Number of the represented month. Value must be within the range 1 through 12. Must be included if <Day> is included. If absent, the value is assumed to be 1 (January).
<Day>	Long	Optional	Number of the represented day. Value must be within the range 1 through 31. If absent, the value is assumed to be 1.

2.3.5.3   Time

Elements of data type *Time* contain an indication of a particular time during a date. This data type describes a repeating portion of a day. That is, each time described (ignoring leap seconds) occurs once per calendar date. In the IFX specification, it is required that a *time* data type be able to represent a specific period with indefinite precision. Milliseconds are the minimum required precision of the *time* data type.

A time represented using this data type must not be ambiguous with respect to morning and afternoon. That is, the time must occur once and only once each 24-hour period.

In addition, the *Time* data type must not be ambiguous with respect to location at which the time occurs. If unspecified, the time zone defaults to Coordinated Universal Time (UTC). Generally, use of a specific time zone in the representation is preferred. The time zone should always be specified to avoid ambiguous communication between clients and servers.

Time Data type	Type	Usage	Description
\<Hour>	Long	Required	Number of the represented hour. Value must be within the range 0 through 23.
\<Minute>	Long	Optional but see Description	Number of the represented minute. Value must be within the range 0 through 59. Must be included if \<Second> is included. If absent, the value defaults to 0.
\<Second>	Long	Optional but see Description	Number of the represented second. Value must be within the range 0 through 60. The value "60" is used only to represent leap seconds. Must be included if \<Fraction> is included. If absent, the value defaults to 0.
\<Fraction>	Long	Optional	Number of represented microseconds. Value must be within the range 0 through 999,999. Particular implementations may choose to allow representations of smaller fractions. If absent, the value defaults to 0.
\<UTCOffsetHour>	Long	Optional	Offset from UTC in hours. Value must be within the range of 0 thorugh 23. Must be included if \<UTCOffsetMinute> is included. If absent, the value defaults to 0.
\<UTCOffsetMinute>	Long	Optional	Offset from UTC in minutes. Value must be within the range of 0 through 59.

2.3.5.4   DateTime

Elements of data type *DateTime* contain all of the information expressed by the *date* and *time* data types. All ambiguities mentioned with *Date* and *Time* (see Sections 2.3.5.2 and 2.3.5.3) should be resolved in a similar fashion here.

DateTime Data type	Type	Usage	Description
\<Year>	Long	Required	4-digit year value.
\<Month>	Long	Required	Number of the represented month. Value must be within the range 1 through 12.
\<Day>	Long	Required	Number of the represented day. Value must be within the range 1 through 31.

DateTime Data type	Type	Usage	Description
<Hour>	Long	Optional but see Description	Number of the represented hour. Value must be within the range 0 through 23.
			Must be included if <Minute> is included. If absent, the value defaults to 0.
<Minute>	Long	Optional but see Description	Number of the represented minute. Value must be within the range 0 through 59.
			Must be included if <Second> is included. If absent, the value defaults to 0.
<Second>	Long	Optional but see Description	Number of the represented second. Value must be within the range 0 through 60. The value "60" is used only to represent leap seconds.
			Must be included if <Fraction> is included. If absent, the value defaults to 0.
<Fraction>	Long	Optional	Number of represented microseconds. Value must be within the range 0 through 999,999. Particular implementations may choose to allow representations of smaller fractions.
			If absent, the value defaults to 0.
<UTCOffsetHour>	Long	Optional	Offset from UTC in hours. Value must be within the range of 0 thorugh 23.
			Must be included if <UTCOffsetMinute> is included. If absent, the value defaults to 0.
<UTCOffsetMinute>	Long	Optional	Offset from UTC in minutes. Value must be within the range of 0 through 59.

2.3.5.5   Timestamp

Elements of data type *Timestamp* contain the same information as *DateTime* values. Unlike that data type, *Timestamp* information is not intended to have meaning at the other end of the communication. In addition, microseconds are the minimum required precision of the time portion of this data type.

The intent here is to describe a type identical to *DateTime* but without semantic meaning between two machines. The general *DateTime* data type has meaning on both ends of the protocol (even though time synchronization is not required by this specification). *Timestamp* indicates an exact point in time with respect to the generating application.

For example, a *Timestamp* value may be generated at a server when creating an audit response. The client application may return that value to the server in later requests, but the client software should not interpret the information.

Timestamp Data type	Type	Usage	Description
<Year>	Long	Required	4-digit year value.
<Month>	Long	Required	Number of the represented month. Value must be within the range 1 through 12.
<Day>	Long	Required	Number of the represented day. Value must be within the range 1 through 31.
<Hour>	Long	Required	Number of the represented hour. Value must be within the range 0 through 23.
<Minute>	Long	Required	Number of the represented minute. Value must be within the range 0 through 59.

Timestamp Data type	Type	Usage	Description
<Second>	Long	Required	Number of the represented second. Value must be within the range 0 through 60. The value "60" is used only to represent leap seconds.
<Fraction>	Long	Required	Number of represented microseconds. Value must be within the range 0 through 999,999. Particular implementations may choose to allow representations of smaller fractions.
<UTCOffsetHour>	Long	Optional	Offset from UTC in hours. Value must be within the range of 0 thorugh 23. Must be included if <UTCOffsetMinute> is included. If absent, the value defaults to 0.
<UTCOffsetMinute>	Long	Optional	Offset from UTC in minutes. Value must be within the range of 0 through 59.

2.3.6   Phone Number

Phone Number indicates a string of up to 32 narrow characters in length (NC-32). It must begin with a plus sign "+" followed by country code, a hyphen, city/area code, another hyphen, then the local phone number. If a PBX extension is to be included, it must appear at the end of the field, separated from the rest of the telephone number by a plus sign.

For example, "+1-800-5551212+739" indicates PBX extension 739 at phone number 5551212 within area code 800 of North America (country code 1).

2.3.7   Decimal

Decimal indicates a numeric value that meets the following rules:

- The value is up to fifteen (15) digits in length, excluding any punctuation (e.g., sign, decimal, currency symbol, etc.).
- The value is not restricted to integer values and has a decimal point that may be placed anywhere from the left of the leftmost digit to the right of the rightmost digit (e.g., +.12345678901234 is acceptable while 12345678901234567 is not).
- The sign is always optional. If it is absent, the value is assumed to be positive.

2.3.8   Long

Long is an integer value, which may be positive, negative, or zero, with values ranging from –2147483648 to 2147483647.

2.3.9   Currency Amount

A *Currency Amount* is a compound data type consisting of four logical elements:

Tag	Type	Usage	Description
<Amt>	Decimal	Required	Amount.
<CurCode>	NC-3	Optional but see Description	Currency code. Required if <CurRate> is specified.
<CurRate>	Decimal	Optional AND	Exchange rate. The ratio of the foreign currency and the currency of the account. The <CurConvertRule> specifies which amount is the numerator and which is the denominator.
<CurConvertRule>	Closed Enum	Optional AND	Currency Conversion Indicator. Valid values are Direct and Indirect. See below for more information.

All monetary amounts in the IFX Specification are handled with the *Currency Amount* data type. When included, this data type contains a decimal value for the amount, an optional three-letter currency code defined

---

in ISO-4217, an optional currency rate, and an optional exchange rate indicator. When Currency Code is not included, Amount is assumed to represent the currency of the account (in particular the "from" account in the case of a transfer or payment).

The valid values for the exchange rate indicator are Direct and Indirect. The indirect rate is the amount of foreign currency per unit of base currency. The direct rate is the amount of base currency per unit of foreign currency. If the exchange rate is present, the currency indicator must be specified. The currency indicator identifies the way in which the rate is calculated.

Derive domestic equivalent from a specified foreign amount:

- If the rate is Indirect, divide the foreign amount by the rate,
- If the rate is Direct, multiply the foreign amount by the rate.

Derive foreign amount from a specified domestic amount:

- If the rate is Indirect, multiply the domestic amount by the rate,
- If the rate is Direct, divide the domestic amount by the rate.

Examples (USD is the base):

USD↔GBP	Indirect = .6037	Direct = 1.6590
USD↔DEM	Indirect = 1.6866	Direct = .5929

The following table is intended to illustrate when the elements within the *Currency Amount* data type are used.

Use Cases	<Amt>	<CurCode>	<CurRate> and <CurConvertRule>
Initiating a transaction…			
In the currency of the FROM account	Required		
In a currency different than the FROM account	Required	Required	Only used when a pre-committed exchange rate is used, see the Foreign Exchange Rate message
Reporting a transaction e.g. within a statement…			
In the currency of the account.	Required		
Originating in a currency different from the currency of the account.	Required	Required	Required

2.3.10 Enum

Enum is a Narrow Character type that has a limited number of specified valid values, each of which is represented by a tag of up to 80 characters each. The *Enum* data type is either a *Closed Enum* or an *Open Enum*. Adding a value to a Closed Enum requires a spec update, while adding a value to an Open Enum only requires out-of-band agreement by the end points.

2.3.10.1 Closed Enum

A *Closed Enum* is an element where a number of valid values are defined within this specification. All other values should be rejected as invalid.

2.3.10.2 Open Enum

An *Open Enum* is an element where a number of valid values are defined within this specification, but other values should not be rejected as invalid by any system other than the final message destination. Open Enums provide a mechanism for a client and final destination server to communicate with values that may be known to both endpoints but not to all intermediate servers that route the message. If the client sends a value that was not listed in the Service Profile as being a supported value in a Profiled Values element, the server must respond

with the most specific response code possible (for example, "Frequency not supported"). Otherwise, if the client or server receives a value that it does not recognize, it must be treated as the type "other."

Open Enums are typically used for elements related to system message processing and have been defined as open to support extensibility and customization of the specification.

2.3.11   Identifier

Object identifiers in the IFX Specification are of the data type "Identifier." This is a Narrow Character data type with a maximum length of 36.

2.3.12   Universally Unique Identifier (UUID)

UUID elements are Narrow Character with a maximum length of 36.

A UUID is an identifier that is unique across both space and time, with respect to the space of all UUIDs. To be precise, the UUID consists of a finite bit space. Thus, the time value used for constructing a UUID is limited and will roll over in the future (approximately at A.D. 3400, based on the specified algorithm). A UUID may be used for multiple purposes, from tagging objects with an extremely short lifetime to reliably identifying very persistent objects across a network. The following information on UUID is based on Internet-Draft <leach-uuids-uuids-01.txt>:

> The generation of UUIDs does not require that a registration authority be contacted for each identifier. Instead, it requires a unique value over space for each UUID generator. This spatially unique value is specified as an IEEE 802 address, which is usually already available to network-connected systems. This 48-bit address may be assigned based on an address block obtained through the IEEE registration authority. This section of the UUID specification assumes the availability of an IEEE 802 address to a system desiring to generate a UUID, but if one is not available, section 4 specifies a way to generate a probabilistically unique one that cannot conflict with any properly assigned IEEE 802 address.3.1 Format. In its most general form, all that may be said of the UUID format is that a UUID is 16 octets, and that some bits of octet 8 of the UUID called the variant field (specified in the next section) determine finer structure.

> For use in human-readable text, a UUID string representation is specified as a sequence of fields, some of which are separated by single dashes. Each field is treated as an integer and has its value printed as a zero-filled hexadecimal digit string with the most significant digit first. The hexadecimal values a to f inclusive are output as lower case characters, and are case insensitive on input. The sequence is the same as the UUID constructed type. The formal definition of the UUID string representation is provided by the following extended BNF:

UUID	<time_low> "-" <time_mid> "-" <time_high_and_version> "-" <clock_seq_and_reserved> <clock_seq_low> "-" <node>
time_low	4*<hexOctet>
time_mid	2*<hexOctet>
time_high_and_version	2*<hexOctet>
clock_seq_and_reserved	<hexOctet>
clock_seq_low	<hexOctet>
node	6*<hexOctet
hexOctet	<hexDigit> <hexDigit>
hexDigit	zero	"1"	"2"	"3"	"4"	"5"	"6"	"7"	"8"	"9"| "a"	"b"	"c"	"d"	"e"	"f"	"A"	"B"	"C"	"D"	"E"	"F"

The following is an example of the string representation of a UUID:

f81d4fae-7dec-11d0-a765-00a0c91e6bf6

2.3.13 URL

An element of the Uniform Resource Locator *URL* data type specifies the URL where a customer may access information. A URL is of the Narrow Character data type with a length of 1024 Characters (NC-1024). The format of a URL begins with a string that identifies which protocol is to be used to access the information, such as "http://".

2.3.14 HexBin

Each "HexBin" character will define the 4-bit "nibble" of the 8-bit byte transformed into a hex character, with the maximum length (n) of the transformation specified.

The HexBin represents arbitrary hex-encoded binary data. The value space of HexBin is the set of finite-length sequences of binary octets. Each binary octet is encoded as a character tuple, consisting of two hexadecimal digits ([0-9, A-F]) representing the octet code. For example, "0FB7" is the *hex* encoding for the 16-bit integer 4023 (whose binary representation is 111110110111).

2.4 Building Blocks

The IFX Specification is constructed using the following building blocks:

Element	The most basic unit of data in the IFX Specification to define a single piece of information (of a specific data type) that is passed between the client and the server.
Aggregate	A group of related elements to provide a mechanism for coding logic rules and a convenient method to refer to related information using a single name.
Message	A collection of elements and/or aggregates to be passed from the client to the server (Request Message) or from the server to the client (Response Message).
Service	A single function or a collection of similar functions that are branded and marketed to individuals and small businesses by a financial institution.
Document	A collection of services and messages sent as a single unit between client and server.

2.4.1 Element

An *Element* is the most basic unit of data in the IFX Specification. An element is defined based on one of the supported data types to define a single piece of information that is passed between the client and server. An element is named according to specific rules and has a definition associated with it to provide additional information on what it contains. An element may also have some usage rules associated with it, which describe how the client and server interact with the element.

For example, the <ChkNum> element is based on the Character data type with a maximum of 12 characters and specifies a Check Number. Depending on where it is used in the specification, it may be provided by the client or the server.

2.4.2 Aggregate

A number of related elements may be grouped together into an *Aggregate*. An aggregate provides a mechanism for coding logic rules ("element 1 or element 2 must be provided") and also provides a convenient way for programmers to specify all of the related information by using a single name.

Note that multiple aggregates may use the same structure. This may occur to allow use of a more descriptive name, e.g., <BillingAddr> in the <PresAcctId> aggregate has the same structure as the <PostAddr> aggregate. This may also occur when a unique name must be used due to the same aggregate being used more than once within a request/response message, e.g. TO/FROM account within a transfer message.

2.4.3 Message

A *Message* is a collection of elements and/or aggregates, which is passed from the client to the server (Request Message) or from the server to the client (Response Message).

A response message is typically a superset of the request that echoes back the information included in the request and adds new information as appropriate to the message being conducted. All requests are sent from client to server and all responses are sent from server to client.

IFX uses several common message types to perform specific functions. Within IFX, the following naming conventions are used, where the messages associated with objects of type *xxx* include:

- Add request <*xxx*AddRq> and response <*xxx*AddRs>
- Modify request <*xxx*ModRq> and response <*xxx*ModRs>
- Delete request <*xxx*DelRq> and response <*xxx*DelRs>
- Cancel request <*xxx*CanRq> and response <*xxx*CanRs>
- Inquiry request <*xxx*InqRq> and response <*xxx*InqRs>
- Audit request <*xxx*AudRq> and response <*xxx*AudRs>
- Synchronize request <xxxSyncRq> and response <xxxSyncRs>
- Advise request <xxxAdviseRq> and response <xxxAdviseRs>
- Reversal request <xxxRevRq> and response <xxxRevRs>

2.4.3.1   Add Message

The Add IFX message has a name structure of <*xxx*AddRq>/<*xxx*AddRs>. It is used to create a new instance of object *xxx* (such as creating a new payment using <PmtAddRq>).

2.4.3.2   Modify Message

The modify IFX message has a name structure of <*xxx*ModRq>/<*xxx*ModRs>. It is used to modify an existing instance of object *xxx* (such as modifying an existing payment using <PmtModRq>.

A client modifies a record on the server using the appropriate Modify request message and replaces the *entire* existing object (all properties) with the newly submitted object. Therefore, within a modify request message, all properties of the object that are intended to be stored must be included, even if the values are not modified.

2.4.3.3   Delete and Cancel Messages

The delete and cancel IFX messages have a name structure of <*xxx*DelRq>/<*xxx*DelRs> and <*xxx*CanRq>/<*xxx*CanRs>, respectively. They are used to delete an existing instance of object *xxx* (such as deleting a payee from the customer payee list using <CustPayeeDelRq>), or to cancel an existing scheduled object (such as canceling a pending payment using <PmtCanRq>.

2.4.3.3.1   Cascading Deletes

The following objects are considered related (the leftmost object cascades into the object on the right of the colon):

- Presentment service: Bill Presentment Accounts
- Payment service: Payment Service Accounts, Pending payments (that are not yet in process or processed), Recurring payment models, Customer payees
- Payment Service Accounts: Pending Payments (that are not yet in process or processed), Recurring Payment models. While Customer payees are not truly related objects to accounts, a customer payee object may contain default payment information <DfltPmtInfo> with a <DepAcctIdFrom> or <CardAcctIdFrom>. If the Account being deactivated is listed as a default payment funding account for any customer payee, the customer payee object should be updated to remove the reference to the deactivated account. Since default payment information accounts are optional, this will not present any data integrity problems.
- Banking Service: Banking Service Accounts, Pending Transfers (that are not yet in process or processed), Recurring transfer models
- Banking Service Accounts: Pending Transfers (that are not yet in process or processed), Recurring transfer models

- Bill Presentment Accounts: Bills (Conditional)
    - New, Delivered or Viewed bills must be deleted.
    - Retired, Withdrawn or Undeliverable bill summaries should remain at the CSP/BSP. The CSP/BSP is responsible for allowing continued customer access to those bill summaries.
- Customer Payees: Pending payments (that are not yet in process or processed), Recurring payment models
- Pending Payments: none
- Recurring Payment Models: Pending payments (that are not yet in process or processed)
- Pending Transfers: none
- Recurring Transfer Models: Pending transfers (that are not yet in process or processed)
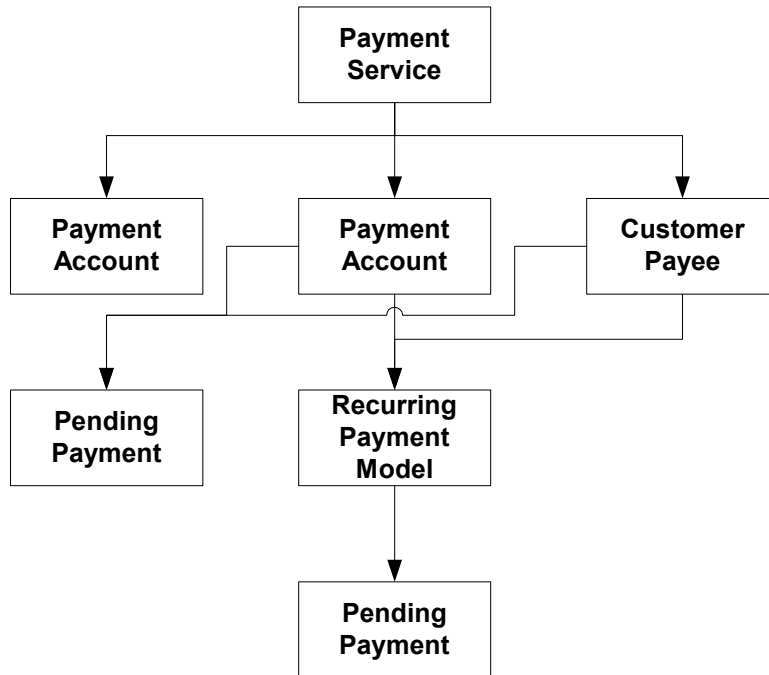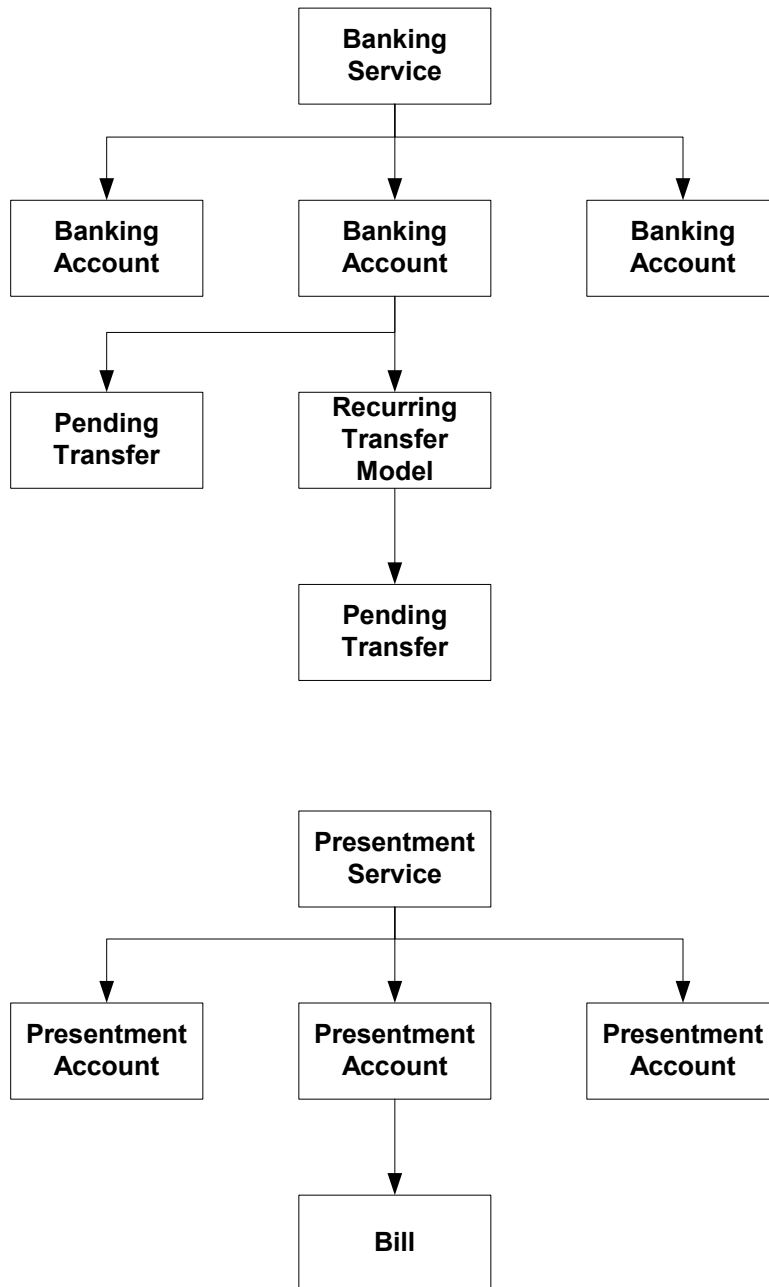
Each object defined above is represented in the hierarchy. Whenever an object is disabled/deactivated/deleted, all the objects below that are also deleted if <CascadeDel> = *True*.

If related objects are deleted as part of a cascade delete, <*xxx*DelRs> and <*xxx*CanRs> messages must be placed in appropriate Audits and Syncs to properly communicate what was removed as part of the cascade delete. The sole exception to this is when an entire service is removed. If the service removal is successful, only the service Rs is required in the Audit or Sync.

Cascade Delete should *not* apply to completed/processed payments or retired/withdrawn/undelivered bills. It is expected that the service will denormalize the payment-payee-account data for a processed payment such that the record of the payment (and all its related data) is self-contained. It is also expected that a service will denormalize the biller-billing account-bill data for a retired/withdraw/undelivered bill such that a record of the bill is self-contained.

A <BillStatusModRq> should be issued when a payment related to a bill is deleted as a result of a cascade from a higher-level object delete. The <BillStatusModRq> must include a <BillPmtStatus> aggregate with the <BillPmtStatusCode> supplied with a value of Cancelled. The specification provides for supplying a bill identifier when adding/modifying a payment to establish this relationship.

If a server is unable to remove the entire dependent object tree during a cascade delete, then all of them must remain and the request must respond with a code of 1300.

2.4.3.4   Inquiry Message

The inquiry IFX message has a name structure of <*xxx*InqRq>/<*xxx*InqRs>. It is used to search for and/or gain information about the current state of existing objects *xxx* (such as finding one or more existing payments using <PmtInqRq>). Inquiry messages limit the response set to records matching the selection criteria used in the request. Selection criterion elements in the request are sometimes repeating elements; where more than one value is given for a particular element (i.e., a repeating element), the query ORs those values. Where selection criteria across multiple different elements are provided, the query ANDs those values. Where an element is absent, the query is not filtering on that element.

Several <Status> codes are available to help the client understand the results of an Inquiry Message:

- 1120 indicates that all of the search criteria that the server was aware of were applied, and no records that matched the search criteria were found.
- 1140 indicates that some, but not all, of the search criteria that the server was aware of were applied, and records were found that matched the search criteria used.
- 1160 indicates that none of the search criteria that the server was aware of were applied, and an unfiltered set of records were found for the implicit search criteria, e.g. <CustId>.

The phrase "that the server was aware of" is used to recognize the situation that a client is implemented using a more recent level of the IFX spec than the server has implemented. If the later version added optional search criteria, the server won't be aware of them. The client may test the <Version> value within the <SvcCore> aggregate to determine if the server is implemented to an older version of IFX. The client may use this knowledge to better interpret the <Status> code values.

> *Note: A server is not obligated to support filtering on all selection criterion elements. If a server chooses not to support a particular element as a selection criterion, it must treat that element as if it were not present in the request. That is, the server must return the appropriate record set for the elements on which it does support filtering. As a result, clients should be prepared to receive records outside the scope of the selection criteria submitted in the request.*

2.4.3.5   Audit and Synchronization Messages

This section introduces the concepts of message auditing and data synchronization as used in the IFX Specification. As its name implies, message auditing is a server function that records all message activity (i.e., creates and stores records for audit trail purposes) affecting the state of IFX defined objects. Examples of such objects are transfers, payees, and payments. The specific audit messages provide a user (typically a customer service representative [CSR]) the capability to retrieve a time-sequenced listing of message responses that added, modified or deleted a user's objects. Informational messages like inquiries do not affect the state of the business objects, and therefore are not recorded in the audit logs.

Data Synchronization is a server function providing clients that store their object data locally (e.g., personal finance managers [PFMs]) a means to update the state of their local data by requesting and comparing data from the server. This allows the client to learn about data resulting from actions that occurred at the server but are unknown to the client. For example, a user may have multiple PFM clients or employ other channel devices, such as telephones or web browsers, to submit requests to the server that changes the state of their business object data.

The IFX audit and synchronization messages are similar in that their responses contain a list of add, modify or delete records. As described above, the difference between audit and synchronize is in their intended use, which is manifested in their message requests as differences in the selection criteria.

The sync messages are specifically tailored for PFM-like clients (i.e., clients that keep local "state"). The only selection criterion for the synchronization requests is the <Token>, which, by standard convention, is the value that determines the starting point for the synchronization. The server assigns the <Token> value, except when the client sets the token to zero to request the first synchronization or to request a refresh. In the synchronization response, the server must return an updated <NewToken> that the client should save for use in its next synchronization request. A more detailed description of the token is provided in Section 3.1.7.

Audit messages are used (typically by CSR's) to trace the sequence of messages that users may have issued during a certain timeframe that affected the state of their objects. The audit requests contain optional selection criteria specifically defined for flexible and effective tracing of user message activity. For example, a CSR may want to resolve a payment dispute, by verifying that a change to a scheduled payment had been made prior to the payment being processed. Via the payment audit message, the CSR can request from the server all modifications made to a scheduled payment that occurred before it was processed.

As in the case of user or CSR initiated actions (i.e., adds, mods or deletes), server-initiated actions for spawning instances from recurring models should also create and log audit records (i.e., "adds") for these instances. In addition, whenever the server spawns a skipped instance, it should create and log an "add" audit record for the skipped instance, and must create and log a "mod'' audit record for the model to account for the skip count adjustment made to the model. The "mod" audit record for the model is required because the user randomly may initiate the skip function any time after the model has been created. However, normal model status changes (e.g., number of remaining instances) do not require any model "mod" audit records, since these changes are predictable from the definition of the model.

2.4.3.5.1   Audit Message

The IFX audit message has a name structure of <*xxx*AudRq>/<*xxx*AudRs>. Audit in the IFX Specification refers to those messages that allow clients to receive specific message responses that change the state of an object, or class of objects, stored on the server since some past point in time. This "past point in time" is either identified by customer-specified audit selection criteria, including a DateTime range.

Clients initiate audits to find out what messages were executed that caused the server to get to the current state. Clients initiate inquiry messages to view the current state of an object, or class of objects.

"Thin" (stateless) clients also use audit messages to allow a customer or CSR to receive a change history of an object, such as a payment, typically to assist in problem resolution.

The server should return as many state changes as are practical in response to an audit request. The specific type of object determines state changes. Most objects have a clear life cycle, from creation to deletion. Each status change must be reported using the appropriate add, modify, delete/cancel message response.

The life cycle of some objects, e.g. a payment, is more complex. Information about a payment may change after it moves from a pending state to a processed state. The state change from pending to processed, and any subsequent changes must be included in the Audit response as modification response messages.

Audit must contain all messages where the <Severity> within <Status> is Info or Warn. Audit may contain messages where the <Severity> is Error, at server discretion.

2.4.3.5.2   Synchronization Message

The IFX synchronization message has a name structure of <*xxx*SyncRq>/<*xxx*SyncRs>. Synchronization in the IFX Specification refers to those messages that allow clients to receive specific message responses that change the state of an object, or class of objects, stored on the server since some past point in time. This "past point in time" is identified by a server-assigned token. The server-assigned token is only meaningful to the server that assigned it.

"Thick" (stateful) clients send synchronization requests to find out what messages were executed that caused the server to get to the current state, so as to synchronize their local data with the server (system of record). Clients send inquiry requests to view the current state of an object, or class of objects.

The server should return state changes using the same rules as Audit. However, at server discretion, multiple messages in a Sync may be "collapsed" into a single message. The resultant collapsed message must provide

information sufficient to bring the client up to date. For example, if an object has been modified several times, then deleted, it is sufficient for the server to return only the <*xxx*DelRs> response in a Sync.

The Sync response includes a playback of messages for a single customer. In it, the messages that affect a specific object must be returned in the order that that server processed then, i.e. ascending by <EffDt>. Examples of objects that the Sync refers to are a payment, in the case of <PmtSyncRq>. A client that maintains local data, sometimes referred to as "thick" or "stateful" client, should apply the changes in the order received to synchronize its local data with that maintained by the server (system of record).

The server may, at server discretion, not maintain sufficient history to reliably update the client to be consistent with the current data at the server. The server must detect this condition by checking the <Token> supplied by the client in the Sync request versus the oldest <Token> maintained by the server. When the <Token> supplied by the client is older than the history maintained by the server, the server must return <Status> with <Severity>Error. The server must not return any message records within the response. The typical client error recovery should be to issue an inquiry message to receive the current status of the object class, e.g. payments, intrabank transfers, etc.; and use it to synchronize its local database.

2.4.3.6   Advise Message

The IFX advise message set has a name structure of <*xxx*AdviseRq>/<*xxx*AdviseRs>. Advise in the IFX specification refers to those messages that allow an entity to advise another entity of an object or a property of an object. The Advise message set does not imply any action to take such as add, or update. It is up to the receiving entity to decided what action to take based on the contents of the Advise message. The <*xxx*AdviseRq> message contains the object record information or object id and property. The <*xxx*AdviseRs> messages allow a status to be returned acknowledging receipt of the request.

A likely scenario for this message type is an entity, who has the ability to receive <xxxAdviseRq> messages, wishes to be advised on the status of an object, e.g. a payment previously added. This entity is interested in being updated as the payment goes through its life cycle from pending to processed, and any other events related to the payment. In this scenario, as the payment status gets updated, a <PmtStatusAdviseRq> would be sent to advise the entity of the payment status change.

2.4.3.7   Reversal Message

The Reversal IFX message has a name structure of <xxxRevRq>/<xxxRevRs>. It is used to reverse an existing instance of an object xxx (such as reversing an existing debit using <DebitRevRq>) or a message (such as reversing a balance inquiry using <BalRevRq>).

The Reversal message can be useful in situations when an IFX client experiences unusual delays in receiving IFX responses, or for situations where the Customer who initiated a transaction cancels or walks-away from an

ATM/POS terminal before their transaction completed, or for situations where an ATM/POS terminal was not able to complete fulfillment successfully.

In general, the Reversal message provides the capability to "undo" a previous request. For example, performing a reversal on a modify request would place that object back to the state it was in prior to the modify. Unlike the use of a Delete or Cancel message, if the IFX client uses a Reversal message not only would the prior request be reversed but also any implications related to back-end processes or fees would also be reversed.

For a client to be able to send a reversal message request, it must be capable of saving or recreating the original request message that is being reversed. Also, a reversal message can not be reversed.

A request to reverse a message is not guaranteed to be successful. The reversal of the last operation on an object (O-1) is likely to be successful. The reversal of an operation prior to the most recent one on an object (O-2, O-3, etc.) is not as likely to be successful. (Example 1 – If an object add was followed by an object modify, it is likely an object modify reversal will cause the object to be back to the original object add state. Example 2 – If an object add was followed by an object modify #1 then an object modify #2, an object modify #1 reversal would not likely be successful however an object modify #2 reversal would have a good chance of being successful.)

2.4.4   Service

A *Service* is a collection of related messages.

Service wrappers are aggregates containing one or more IFX messages of the same service. Except for Signon and Signoff, all IFX messages have a service associated with them, and must be contained in a service wrapper. <*xxx*SvcRq> is used in requests, where *xxx* is the name of the service (e.g., Pay, Bank, Base). <*xxx*SvcRs> is used in corresponding responses.

Within a service wrapper is an optional <SPName> element. The client must provide this value in cases where the destination service provider is ambiguous. One example of such a scenario is where there are multiple service providers for a given service, supporting the same messages.

The service response wrapper also contains an optional <Status>. If the associated service provider returns an error to the CSP, or the CSP is unable to forward the messages to the associated service provider, the <Status> in the service response wrapper is used to communicate this failure to the client.

A service wrapper for a particular service may repeat. One possible use for this is a case where one SP supports different messages than another. In such a situation, the client may use one SP for some messages, and another SP for others. This also makes it possible for a single SP to route internally using the service wrapper, simply by giving each destination a different <SPName> in the Service Profile.

2.4.4.1   Request <xxx**SvcRq**>

Tag	Type	Usage	Description
<RqUID>	UUID	Required	Request Identifier. Sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header aggregate.
<AsyncRqUID>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously. For more information, see section 3.2.12.1.
<SPName>	Identifier	Optional	Service provider name. May be provided by the client to clearly identify which service provider is the intended destination for this collection of messages. It is required in instances where the destination would otherwise be ambiguous.
<*xxx*Rq>	Message	Optional Repeating	A collection of various requests associated with the service named in the service wrapper.

2.4.4.2   Response <xxx**SvcRs**>

Tag	Type	Usage	Description
<Status>	Aggregate *see section 3.2.12.1*	Optional	Response status.
<RqUID>	UUID	Required Echoed	Request Identifier.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header aggregate.
<MsgRsHdr>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header aggregate.

Tag	Type	Usage	Description
<AsyncRqUID>	UUID	Optional Echoed	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously. For more information, see section 3.2.12.1.
<SPName>	Identifier	Optional Echoed	Service provider name. May be provided by the client to clearly identify which service provider is the intended destination for this collection of messages. It is required in instances where the destination would otherwise be ambiguous.
<*xxx*Rs>	Message	Optional Repeating	A collection of various responses associated with the service named in the service wrapper.

2.4.5   Document

An IFX *Document* is a collection of services and messages sent as a single unit between client and server.

> *Note: both the request document and the response document share the <IFX> tag name.*

2.4.5.1   Request <IFX>

An IFX request document must contain at least a Signon request.

Tag	Type	Usage	Description
<SignonRq>	Message	Required	Signon Request Message.
<*xxx*SvcRq>	Service	Optional Repeating	Service Request Wrapper.
<SignoffRq>	Message	Optional	Signoff Request Message.

2.4.5.2   Response <IFX>

An IFX response document varies, depending on the validity of the associated IFX request document.

Tag	Type	Usage	Description
<Status>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, <StatusCode> defaults to 0 (zero). If this aggregate is absent, <StatusCode> defaults to 0 (zero).
<SignonRs>	Message	Optional *but see Description*	Signon Response Message. Must be provided in cases where the IFX request document was valid.
<*xxx*SvcRs>	Service	Optional *but see Description* Repeating	Service Response Wrapper. Must be provided for each <*xxx*SvcRq> provided in the request document.
<SignoffRs>	Message	Optional *but see Description*	Signoff Response Message. Must be provided if <SignoffRq> was provided in the request document.

2.4.6   Order

IFX has the following rules related to order:

- Within an IFX document, the signon message must come before any service wrapper or signoff message.

- Within an IFX document, the signon message must be processed before any service wrappers or the signoff message.
- Within an IFX document, the order of service wrappers is not defined.
- Within an IFX document, the processing order of service wrappers is not defined.
- Within an IFX document, the signoff message must come after the signon message and any service wrappers.
- Within an IFX document and if present, the signoff message must be processed after the signon message and after any service wrappers.
- Within a service wrapper, the order of messages is not defined.
- Within a service wrapper, the processing order of the messages is sequential based on the message order submitted by the requestor.
- Within a message, the business message specifiction defines the element and aggregate order for that message.

2.5 Naming Convention

Naming in the IFX Specification follows a convention based on a small set of rules and a set of standard abbreviations. These rules and abbreviations apply to names of elements, aggregates, messages, messages, and services.

2.5.1 Rules for Naming in the IFX Specification

- Names should be designed for clarity of semantic meaning.
- Names are case-sensitive.
- Within an aggregate, the sequence of fields matters.
- Names must follow standard abbreviations when possible.
- The letters "Rq" and "Rs" must only appear as the final two letters of a name if it is a Request message, Response message, Request Service Wrapper, or Response Service Wrapper.
- Message names must follow Class, Object, Property, Method, Direction order; e.g., <PmtAddRq>.
- All custom values for Open Enum elements must be based on the established naming conventions and must use a name to designate the organization that developed the extension.
- All custom tags must have a prefix and a suffix. The delimiter between the prefix and the suffix may be specific to the implementation. For example all extensions developed by ABC Corporation must use the syntax <com.abccorp:Something> when ":" is specified as the delimiter. The governance process will support the registration of prefixes.

2.6 International Support

2.6.1 Country Codes

All representations of countries in the IFX Specification are the three-character codes for countries that are defined in ISO-3166. These country codes are used in this specification for the following purposes:

- In postal addresses,
- With language codes to specify a dialect of a language used in multiple countries, and
- To identify country-specific tags that may be used to pass and store data relevant only in a single country.

2.6.2 Character Sets

The IFX Specification defines two data types to represent character data.

The most general data type used for the representation of most character data is called *Character*. The IFX Specification places no restrictions on the character sets used to represent elements based on the *Character* data type. The *Character* data type is intended to allow data in either single or multi-byte character sets to be passed between client and server and stored on the server to facilitate implementation in the broadest possible number of countries. It is expected, but not required, that multi-byte implementations be based on the UTF representation of Unicode.

The other data type used for representation of character data is called *Narrow Character*. This data type is used in the IFX Specification to restrict some elements to single-byte characters so that implementation may be simplified. These elements tend to be related to system functions and should not interfere with the ability of an implementation to support multiple-byte character sets for most character data.

2.6.3   Reporting Foreign Exchange Transactions

Messages that allow a customer to execute or schedule a financial transaction allow the customer to optionally specify the currency for the message. The SP may refuse to honor the message in the requested currency.

The client may use the Deposit Account Transaction Inquiry or similar messages for other services to retrieve financial transaction detail. The detail available for a message that was executed with an amount in a foreign currency must include the original financial transaction amount in the requested currency with the actual exchange rate applied against the customer's account for the financial transaction and the rate indicator. The converted amount in the account currency is also explicitly provided with the financial transaction detail.

The <EU.Cur> element in the Signon message supports the new laws governing the use of the euro in Europe. This allows the client to specify whether they want to see financial amounts in euros or the currency of the local country.

2.6.4   Languages and Dialects

Whenever languages are specified in the IFX Specification, they use the RFC-1766 standard, which uses the ISO-639 international standard for *two*-letter language codes. Languages also include an optional ISO-3166 *two*-letter country code to account for differences in dialect when the same language is used in multiple countries.

> *Note: this is different from the use of country codes elsewhere in the specification, where* three-letter codes are used. Two-letter codes are used here because RFC-1766 did not have a provision for a standard use of three-letter codes for language definitions.

The customer may, in the <SignonRq>, specify a preferred language and dialect from the list provided by the server in <BaseSvcProfInfo>. If the client has not yet obtained a list of languages from the server, the customer may choose any language, but the server may default to a particular language if the preferred language is not among those supported.

2.6.5   Times and Time Zones

Whenever times are specified in the IFX Specification, they are specified with an offset from Coordinated Universal Time (UTC). This specificity allows clients and servers in different time zones to communicate without ambiguity.

2.6.6   Country-Specific Tags

To accommodate differences in financial regulations and practices in different countries, the IFX Specification allows extensions to be created that allow additional information to be passed between clients and servers that implement messages under an individual country's regulations. Elements, Aggregates, or Messages that are added to the IFX Specification to support the needs of a single country should use custom tag names with a prefix that is the ISO-3166 three-letter country code. All three-letter tag name prefixes in the IFX Specification are reserved for country-specific tag names.

In addition, the two-character prefix "EU" has been reserved for use by the European Union.

2.7 Versioning and Specification Evolution

The IFX Specification has been designed to support two separate but related mechanisms for evolution. The specification evolves formally through an open governance process that provides a mechanism for features that are widely useful to be incorporated into the core specification, which is defined in this document, and the associated IFX implementation specifications.

In addition, individual Financial Institutions or Service Providers may design their own custom extensions to the current version of the specification to quickly add needed functionality. Through this process, new custom elements or custom aggregates may be added to existing messages, custom messages and custom profile options may be added to existing services, and entirely new custom services may be developed.

2.7.1 Extension and Customization

Since all tag names in the IFX Specification follow the same naming conventions, it is relatively easy for a Financial Institution or Service Provider to design customized extensions.

All custom tag names and custom values for Open Enum elements must be based on the established naming conventions and must use a name to designate the organization that developed the extension. All custom names and values have a prefix and a suffix. The delimiter between the prefix and the suffix may be specific to the implementation. For example, all extensions developed by ABC Corporation must use the syntax com.abccorp:Something, where ":" is specified as the delimiter.

To avoid clashes, all organizations using custom tags must either register their prefix with IFX Forum or follow the convention described below to name their prefixes:

- Use the organization's fully qualified Internet domain name, reverse the order, with the top level domain first, e.g. org.ifxforum for IFX Forum (with a domain name of ifxforum.org)
- If the organization wants to have multiple prefixes for different services or different versions of customization, it may attach a sub-domain name, e.g. org.ifxforum.banking or org.ifxforum.v2
- The prefix must consist only of lower-case alphanumeric characters or dash ('-')

  Note: the maximum length of a custom value is 80 characters.

The governance process will support the registration of prefixes. All ISO country-specific values are reserved and assigned country specific prefixes. Therefore, it is required to reserve three-letter prefixes for such use. Unreserved values for private tags (i.e. ones that may be registered with the IFX Forum) are four or more positions in length.

2.7.2 Mandatory and Optional Elements

The IFX Specification documentation conveys a number of rules for usage of elements and aggregates. This section provides a general taxonomy of rules for mandatory and optional elements and aggregates. For documentation conventions, see Section 2.2. These rules may be categorized as follows:

- *Required messages within a service*—Each chapter that defines messages contains a table (normally section *x*.2, where *x* is the chapter number) that includes a column labeled "Req." "Yes" in this column indicates that a server supporting the service that the chapter is part of must support this message if it supports the service. The server indicates support for optional messages (i.e., those with blank in the "Req." column) using <MsgSupt> in the section of the Service Profile that corresponds to that service. (The Service Profile section for each service is typically documented at the end of each chapter.)
- *Message Level Required Element or Aggregate*—Within each message, the elements and aggregates that appear at the highest level (as opposed to within nested aggregates) and that are required are marked. The meaning of required is slightly different for request and response messages.
  - In a request message, an element or aggregate that is marked as required must be included.

o In a response message, an element or aggregate that is marked as required must be included in every successful response, but need not be included in failed responses. In addition, whether the message succeeds or fails, every response must contain a Request Unique ID <RqUID>, a Response Status aggregate <Status>, and a Customer Identification aggregate <CustId> if it was provided in the request.

- ***Elements and Aggregates Within a Higher Level Aggregate***—Within an aggregate, elements and aggregates are marked in the same way that they are at the message level. When an element or aggregate is marked required within the context of a higher-level aggregate that is optional, there is an implied "required if the aggregate is included."

- ***Or and Exclusive Or Conditions***—There are places in the specification where one or more elements or aggregates from a list must be included. Where exactly one item must be included it is marked as an "Exclusive Or" or XOR condition. Where one or more items must be included it is marked as an OR condition.

- ***Required by Service Profile***—Some but not all Financial Institutions or Service Providers may require many elements and aggregates. Each FI or SP must indicate to the client through the Service Profile for the service whether or not these elements and aggregates are required.

- ***Required by Biller Directory***—Some but not all Billers may require some elements and aggregates. Each Biller must indicate to the client through their entry in the Biller Directory whether or not these elements and aggregates are required. Note that the Biller Service Provider (BSP) may actually create/maintain the directory entry for each Biller that it supports.

- ***Complex Usage Rules***—Some elements and aggregates have usage rules that are more complex than the conditions described above. Where this is the case, these conditions are described in text in the Description column.

- ***Definition of "optional"***—A client or server may include optional fields within a message. There is no requirement that the receiver process optional fields. In order to maintain upward compatibility, the receiver of a message must ignore tags that it does not support. Note that it is not unusual for an optional field to be required based on the context. This condition is documented by the words "*but see Description*" in the Usage column.

3 Common Elements and Aggregates

3.1 Common Elements

A number of elements are used throughout the IFX Specification.

3.1.1 Edit Masks

Edit masks are elements that provide a way for a user interface to determine and check what type or actual value must be in each character position of a particular related element. For example, edit mask is used in <AcctMask> in the <BillerAcctIdInfo> Aggregate to provide a way to determine whether the number entered by the user for <BillingAcct> in <PresAcctId> is a valid billing account number for that biller. Edit mask is used by <SecretMask> in <SecretPrompt> to check the characters input by the user for <Secret>.

Multiple edit masks may exist for the customer accounts at each biller, as identified by a unique <BillerNum>. In this case, the client may apply one rule at a time until one succeeds. If all edit masks fail for the account number given, the client should consider the account number as invalid. Clients are not required to support this function; instead, they may rely on the presentment service provider to do it for them. Clients may wish to support this checking for performance reasons.

The table below identifies the characters that may be in the mask and what they mean.

Mask Position Contains	Position in Account Number Must Contain	Explanation
A	Alpha characters A through Z	A lowercase "a" means an alpha character A through Z must appear in that position.
B	Space (blank)	A space must appear in that position.
C	Alphanumeric characters, no space	An alpha character, A through Z or a through z, or a number, 0 through 9, must appear in that position.
I	Ignore	Ignore this position. Do not check for any character's presence or absence.
N	Digits 0 through 9	A numeric character from 0 through 9 must appear in that position
X	Alphanumeric characters, space allowed	An alpha character, A through Z or a through z, a number, 0 through 9, or a space must appear in that position.
A through Z	That exact character	An upper case letter means that that alpha character must appear in that position.
0 through 9	That exact number	A number means that that specific number must appear in that position.
Special Characters—/ * $ # , @	The specified character	A special character means that that exact special character must appear in that position in the account number.

Note: <AcctMask> is only 32 characters long, while the <SecretMask> is 80 characters long. The length of the edit mask element matches the length of the element being masked.

3.1.1.1 Account Mask <AcctMask>

Tag	Type	Usage	Description
<AcctMask>	NC-32		Account Mask. String describing the valid values to be checked against values input by the user.

3.1.1.2 Secret Mask <SecretMask>

Tag	Type	Usage	Description
<SecretMask>	NC-80		Secret Mask. String describing the valid values to be checked against values input by the user.

3.1.2 Customer Identifiers

The IFX Specification uses two different identifiers for customers. The first, Customer Login ID <CustLoginId>, is a user-friendly name or number (such as tax id) that is used with a password for customer authentication. The second identifier, Customer Permanent ID <CustPermId>, is a unique permanent id used by the SP as a database key and is not typically known by the customer. The <CustPermId> is used by the CSP to identify the Customer in messages to other service providers. When the <CustPermId> is used outside of the entity that assigned it, further qualification may be required. See Section 3.2.1.1 for more information.

Clients do not typically need to use <CustPermId>. When <CustPermId> is used, it is returned as part of the Signon Response and added to each subsequent message by an intermediate server prior to the message being delivered to the SP. Both IDs may appear together in the <CustId> aggregate (See Section 3.2.1.1).

3.1.2.1 Customer Login ID <CustLoginId>

Tag	Type	Usage	Description
<CustLoginId>	NC-32		Customer Login ID. Used as a user-friendly name for the customer for authentication purposes. Some implementations may allow a user to change his or her Login ID.

3.1.2.2 Customer Permanent ID <CustPermId>

Tag	Type	Usage	Description
<CustPermId>	NC-32		Customer Permanent ID. Used as a database key to uniquely identify an FI or CSP customer. Cannot be changed by the customer.

3.1.3 Request Identifier <RqUID>

A client uses <RqUID> to uniquely identify a request message. It is a universally unique ID (UUID) that is generated according to an algorithm specified by the Open Software Foundation Distributed Computing Environment (OSF/DCE) standards to produce a 36-character hexadecimal encoding of a 128-bit number.

A Request Identifier <RqUID> is included in request messages. A server must echo the received <RqUID> in the response message it generates.

The server must store the <RqUID> for as long as the server stores the audit/sync log. The client may then correlate the asynchronous response with the request when the response is not immediate.

Tag	Type	Usage	Description
<RqUID>	UUID		Request Identifier. Sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.

3.1.4 Service Provider Reference Identifiers

Customer Service Provider Reference Identifier <CSPRefId> and Service Provider Reference Identifier <SPRefId> are typically optional elements that allow both the CSP and SP to return message reference numbers for a message. These identifiers provide permanent reference for the message and may be used by the customer in reporting problems to Customer Service.

The reference identifiers may be different for different methods applied to a given object. For example, the reference identifier in the <PmtAddRs> may be different from the reference identifier in a subsequent <PmtModRs> applied to the same payment. Therefore, it is not safe to assume that the reference identifiers may be used to uniquely identify a particular instance of an object (such as a payment).

3.1.4.1 Customer Service Provider Reference Identifier <CSPRefId>

Tag	Type	Usage	Description
<CSPRefId>	Identifier		Customer Service Provider Reference Identifier.

3.1.4.2 Service Provider Reference Identifier <SPRefId>

Tag	Type	Usage	Description
<SPRefId>	Identifier		Service Provider Reference Identifier.

3.1.5 Service Provider Name <SPName>

The Service Provider Name element is a globally unique identifier for a service provider, e.g., the domain name of the SP. <SPName> is used for 2 distinct purposes.

- To specify the destination of a message, when multiple service providers provide the same service, e.g. a CSP interacts with multiple BSPs to implement Bill Presentment.
- To qualify the id of an object that it owns, e.g. <BillerId>, <PmtId>, <XferId>. There are cases where different <SPName>s may be used in the same message, e.g. the BSPs <SPName> is used to qualify the <BillerId> in a message that is sent to the CPP.

To ensure uniqueness, <SPName> must be an Internet domain name registered to the SP. For example, the Banker's Roundtable would use "org.bankersround" as the value for <SPName>.

Tag	Type	Usage	Description
<SPName>	Identifier		Service Provider Name. To ensure uniqueness, a <SPName> should be an Internet domain name.

3.1.6 Organization <Org>

The Organization <Org> element is used where information must be qualified by the organization that assigned it. For example, a tax authority assigns the tax type in the <BankAcctTaxInqRq>, so <Org> is used to identify the state or country tax authority. In general, <Org> is used to make identifiers globally unique in cases where there is no existing real-world globally unique identifier. An organization may register a globally unique value of <Org> through the IFX Governance Process to avoid conflicts with other organizations with similar names. The value of <Org>, like <SPName>, may also be an Internet domain name registered to the Organization defining that name space.

Tag	Type	Usage	Description
<Org>	Identifier		Organization. Organization defining this name space.

3.1.7   Token <Token>

The IFX Specification allows clients to perform synchronization of data with servers using a server-assigned <Token>. This synchronization is performed using synchronization messages, where the client inserts a <Token> it has previously received from the server as an indication of the point in history from which the synchronization should occur. The server must return all messages that are relevant to the type of object being synchronized and have occurred since the <Token> was originally sent to the client. A client may also include <Token>0 in a Sync request, which results in the server returning all known messages of the relevant type.

The "assignor" determines the scope of uniqueness. The scope may either be globally across all customers known to the assignor or specific to a customer.

The server must echo the <Token> in the response and also generate and return a new token <NewToken>, which may be used by the client in the future to specify the current point in history. See Section 2.4.3.5.2 for information regarding synchronization message definition.

Tag	Type	Usage	Description
<Token>	Identifier		Token. Server assigned. Client should make no assumptions about the value of this token relative to others that it may have received. The token is only significant to the server that originally assigned it.
			This is a token that has been previously sent by the server, or zero for first-time requests.

3.1.8   Reversal Reason Code <RevReasonCode>

Tag	Type	Usage	Description
<RevReasonCode>	Open Enum		Reversal reason code. The reason why the client is requesting a reversal.
			Defined values: CustomerCancellation, Unspecified-NoActionTaken, SuspectedMalfunction, FormatError-NoActionTaken, CompletedPartially, OriginalAmountIncorrect, ResponseReceivedTooLate, CardAcceptorDeviceUnableToCompleteTransaction, DepositOutOfBalance, NoCheckInEnvelope, PaymentOutOfBalance, DepositOutOfBalance-AppliedContents, PaymentOutOfBalance-AppliedContents, UnableToDeliverMessageToPointOfService, SuspectedMalfunction-CardRetained, SuspectedMalfunction-CardReturned, SuspectedMalfunction-Track3NotUpdated, SuspectedMalfunction-NoCashDispensed, TimedOutAtTakingMoney-NoCashDispensed, TimedOutAtTakingCard-CardRetainedAndNoCashDispensed, InvalidResponse-NoActionTaken, TimeoutWaitingForResponse
			(reference ISO 8583:1993 Annex A.11)

3.2   Common Aggregates

3.2.1   Message Headers

3.2.1.1   Message Request Header <MsgRqHdr>

The message header request aggregate contains common information for all request messages.

Tag	Type	Usage	Description

Tag	Type	Usage	Description
<EMVRqData>	Aggregate *see section 3.2.2.1*	Optional	EMV Request Data Aggregate.
<NetworkTrnInfo>	Aggregate *see section 3.2.18*	Optional	Network Transaction Information. Identification and location of the terminal from which the message originates.
<MsgAuthCode>	Aggregate *see section 3.2.1.3*	Optional	Message Authentication Code (MAC) aggregate, used by the receiver of this message to verify the sender's identity and authenticate the message contents.

3.2.1.2 Message Response Header <MsgRsHdr>

The message header response aggregate contains common information for all response messages.

Tag	Type	Usage	Description
<EMVRsData>	Aggregate *see section 3.2.2.2*	Optional	EMV Response Data Aggregate.
<ServerTerminalSeqId>	Identifier	Optional	Server Terminal Sequence Identifier. A terminal sequence identifier generated by the server (CSP) in an ATM or POS environment.
<MsgAuthCode>	Aggregate *see section 3.2.1.3*	Optional	Message Authentication Code (MAC) aggregate, used by the receiver of this message to verify the sender's identity and authenticate the message contents.

3.2.1.3 Message Authentication Code Aggregate <MsgAuthCode>

Tag	Type	Usage	Description
<MacValue>	HexBin-∞	Required	The Message Authentication Code value. Calculated using an encryption key shared by both endpoints according to a previously agreed algorithm such as ANSI X9.9/FIPS113 or ANSI X9.71/FIPS198.
<SecObjId>	Identified	Optional	The Security Object ID of the encryption key used to calculate the MAC Value. Used in lieu of check digits to help identify key synchronization errors.*

3.2.2 Europay, MasterCard, Visa (EMV) Data

Europay, MasterCard, and Visa (EMV) have published jointly-developed specifications on how smart cards can be used for improved fraud reduction for transactions that the service providers choose to be more secure. This joint effort is handled by mutually agreed upon smart card standards organization, called EMV Co. EMV specifications allow the clients and servers to utilize the Integrated Circuit Card (ICC) technology, which is also commonly called "smart card", for better fraud risk management. The EMV specifications define a set of requirements that ensure interoperability for credit and debit payment applications between smart cards and terminals on a global basis, regardless of where the card is used.

For more information about the EMV, refer to their website at http://www.emvco.com.

3.2.2.1 EMV Request Data Aggregate <EMVRqData>

This aggregate contains the EMV request data that is sent by the device, which supports EMV functionality, to the acquirer host in an EMV transaction authorization request message. The current EMV specification contains many more elements (for EMV supported transaction requests) than those specified here. The EMV data elements that are specified here are the common 'sub-set' of the current EMV implementation by the device vendors who have chosen to support the EMV functionality.

The usage of all EMV data elements that are specified here is intentionally specified as 'Optional', although some data elements are required in order to verify the EMV support for a transaction request (and the usage column for these elements have the *'but see Description'* comments specified). The reasons for this is:

- The relationship between different device vendors and acquiring hosts might dictate different EMV data element requirements, and
- If a 'Required' EMV data element is not present in a transaction request, then the acquiring host might want to 'fall-back' on the alternate method of authorizing the transaction request. Therefore it is better that the decision to either proceed with the transaction request processing, using the alternate method of verification/authorization, or to decline the transaction request is left with the acquiring host.

Tag	Type	Usage	Description
<AppId>	HexBin-16	Optional	Application Identifier (AID). It is also referred to as the Dedicated File (DF) in some implementations or applications of EMV. The data value contained by this element is in hexadecimal characters, e.g., Hex "4F" value is specified in two characters as "4" & "F". The AID consists of two parts: (i) A Registered Application Provider Identifier (RID) of 5 bytes, unique to an application provider and assigned according to ISO/IEC 7816-5. (ii) An optional field assigned by the application provider of up to 11 bytes. This field is known as a Proprietary Application Identifier Extension (PIX) and may contain any 0-11 byte value specified by the provider. The meaning of this field is defined only for the specific RID and need not be unique across different RIDs.
<Trk2EquivData>	HexBin-20	Optional	Track 2 Equivalent Data. Indicates the Track 2 Data as read from the Integrated Circuit Card (ICC). The format of this data is according to ISO/IEC 7813 standards, excluding start sentinel, end sentinel, and LRC. It includes the following data elements: Primary account Number (PAN), a variable length numeric character string of up to 19 characters; Field Separator (hex "D" value); Expiration Date in YYMM format, a 4-character numeric string; Service Code, a 3-character string; Discretionary Data (define by individual payment systems), a variable length character string of up to 12 characters; and the last character is "padded" with hex. 'F', if needed to ensure whole bytes. The data associated with this element is not needed if the Track 2 data already contains this information elsewhere in the transaction message.
<AppPan>	HexBin-10	Optional	Application Primary Account Number (PAN). Indicates the valid cardholder account number, padded to the right with hex 'F' values. The data value contained by this element is in hexadecimal characters. The data associated with this element is not needed if the <Trk2EquivData> already contains this information.
<AppPanSeq>	HexBin-1	Optional	Application PAN Sequence Number. Sometimes, also referred to as the Member Number. Identifies and differentiates cards with the same PAN (e.g., cards with the same PAN for different family members). The data value contained by this element is in hexadecimal characters.

Tag	Type	Usage	Description
<CryptoInfoData>	HexBin-1	Optional *but see Description*	Cryptogram Information Data. It indicates the type of cryptogram returned (approved, online, or denied) by the card and the actions to be performed by the terminal (advice or not, reason for advice). The data value contained by this element is in hexadecimal characters.
			For exact values based on the bit positions (the data is in hex), refer to the EMV '96 Integrated Circuit Card Specification for Payment Systems (Table II-17).
			Required if the online EMV support is needed for this transaction, and the user does not intend to 'fall back' on the alternative method of authorizing the transaction request.
<AuthRqCrypto>	HexBin-8	Optional *but see Description*	Authorization Request Cryptogram (ARQC). Cryptogram returned by the ICC in response to the GENERATE AC command. The data value contained by this element is in hexadecimal characters.
			Required if the online EMV support is needed for this transaction, and the user does not intend to 'fall back' on the alternative method of authorizing the transaction request.
<IssAppData>	HexBin-32	Optional *but see Description*	Issuer Application Data (IAD). Indicates the proprietary application data for transmission to the issuer in an online transaction. It also provides the data elements that the Issuer has elected to have in the authorization message. For example, derivation key index and algorithm version numbers.
			Required if the online EMV support is needed for this transaction, and the user does not intend to 'fall back' on the alternative method of authorizing the transaction request.
<AppICProf>	HexBin-2	Optional *but see Description*	Application Interchange Profile (AIP). It contains the capabilities of the card to support specific functions in the application, as defined in the AIP table. The data value contained by this element is in hexadecimal characters.
			For exact values based on bit positions (the data is in hex), refer to the EMV '96 Integrated Circuit Card Application Specification for Payment Systems (Table A-1).
			Required if the online EMV support is needed for this transaction, and the user does not intend to 'fall back' on the alternative method of authorizing the transaction request.
<AppTrnCounter>	HexBin-4	Optional *but see Description*	Application Transaction Counter (ATC). A counter maintained by the application in the Integrated Circuit Card (ICC). It indicates sequential order of the transaction performed by the application. Incrementing of the ATC is managed by the ICC. The data value contained by this element is in hexadecimal characters.
			Required if the online EMV support is needed for this transaction, and the user does not intend to 'fall back' on the alternative method of authorizing the transaction request.

Tag	Type	Usage	Description
<AuthAmt>	Currency Amount	Optional *but see Description*	Authorization Amount. It contains the transaction amount, excluding any adjustments, that needs to be authorized.
			Required if the online EMV support is needed for this transaction, and the user does not intend to 'fall back' on the alternative method of authorizing the transaction request.
<SecAmt>	Currency Amount	Optional *but see Description*	Secondary Amount. It contains the secondary (or other) amount of the transaction. It contains any secondary amount that transaction might have. E.g., Cashback amount.
			Required if the online EMV support is needed for this transaction, and the user does not intend to 'fall back' on the alternative method of authorizing the transaction request.
<TerminalCountryCode>	NC-3	Optional *but see Description*	Terminal Country Code. Indicates the country code of the merchant or terminal associated with the transaction. This is the ISO country code based on ISO-3166.
			Required if the online EMV support is needed for this transaction, and the user does not intend to 'fall back' on the alternative method of authorizing the transaction request.
<TerminalVrfyRslt>	HexBin-5	Optional *but see Description*	Terminal Verification Result (TVR). It defines the status of different functions as seen from the terminal. The data value contained by this element is in hexadecimal characters.
			For exact values based on the bit positions (the data is in hex), refer to the EMV '96 Integrated Circuit Card Application Specification for Payment Systems (Table A5).
			Required if the online EMV support is needed for this transaction, and the user does not intend to 'fall back' on the alternative method of authorizing the transaction request.
<CurCode>	NC-3	Optional *but see Description*	Currency Code. Indicates the currency code associated with the transaction amount. This is the ISO currency code based on ISO-4217.
			Required if the online EMV support is needed for this transaction, and the user does not intend to 'fall back' on the alternative method of authorizing the transaction request.
<TrnDt>	Date	Optional *but see Description*	Transaction Date. Indicates the local date the transaction was authorized.
			Required if the online EMV support is needed for this transaction, and the user does not intend to 'fall back' on the alternative method of authorizing the transaction request.
<EMVTrnType>	NC-2	Optional *but see Description*	Transaction Type. Indicates the type (code) of the transaction, represented by the first two digits the ISO 8583 (1987) Processing Code.
			Required if the online EMV support is needed for this transaction, and the user does not intend to 'fall back' on the alternative method of authorizing the transaction request.

Tag	Type	Usage	Description
\<UnpredictNum\>	HexBin-4	Optional *but see Description*	Unpredictable Number. An unpredictable number, generated by the terminal, which may be used for input to the application cryptogram algorithm to ensure the unpredictability of data input to this calculation or for random transaction selection for terminal risk management. It contains the value to provide variability and uniqueness to the generation of the application cryptogram in order to reduce fraud risk. The data value contained by this element is in hexadecimal characters. An unpredictable number shall be generated in accordance with an individual payment system's specifications. One example of a method for generating the Unpredictable Number is performing an exclusive-OR operation on all the previous Authorization Request Cryptograms (ARQCs), Transaction Certificates (TCs), Application Authentication Cryptograms (AACs), and Application Authorization Referrals (AARs). Required if the online EMV support is needed for this transaction, and the user does not intend to 'fall back' on the alternative method of authorizing the transaction request.
\<TerminalCaps\>	HexBin-3	Optional	Terminal Capabilities. Indicates the card data input, cardholder verification method (CVM), and security capabilities of the terminal. The data value contained by this element is in hexadecimal characters.
\<CVMRslts\>	HexBin-3	Optional	Cardholder Verification Method (CVM) Results. Indicates the results of the last cardholder verification method (CVM) performed. The data value contained by this element is in hexadecimal characters.
\<TrnStatInfo\>	HexBin-2	Optional	Transaction Status Information. Indicates the functions performed in a transaction. The data value contained by this element is in hexadecimal characters.
\<AppVersionNum\>	HexBin-2	Optional	Application Version Number. The version number of the application assigned by the payment system.

3.2.2.2 EMV Response Data Aggregate \<EMVRsData\>

This aggregate contains the EMV response data that is returned by the acquirer host to the device, which supports EMV functionality, in an EMV transaction authorization response message. The EMV data elements that are specified here are the common 'super set' of the current EMV implementation by the device vendors who have chosen to support the EMV functionality.

Tag	Type	Usage	Description

Tag	Type	Usage	Description
\<AuthRsCode\>	C-2	Required	Authorization Response Code. A code that defines the disposition of an EMV response message.
			As a minimum, the codes reflecting the following responses should be defined in the response messages from the Issuer to the terminal: Online Approval, Online Decline, Referral (initiated by Issuer), and Capture Card.
			In addition, the terminal shall be able to generate (and transmit to the ICC) the codes reflecting the following new responses when transactions are not authorized online: Unable to go Online, Offline Approved; Unable to go Online, Offline Declined; Offline Approved; Offline Declined; Approval (after card-initiated referral); and Decline (after card-initiated referral).
			The codes are to be set by the individual payment systems.
			The terminal shall never modify the Authorization Response Code returned in the response message. The card's final decision is reflected in the Cryptogram Information Data and not in the authorization Response Code.
\<IssAuthData\>	HexBin-32	Optional *but see Description*	Issuer Authentication Data. Contains the data sent to the ICC for online issuer authentication. If present, the value contained by this element is at least 16 characters long, and it contains the Application Response Cryptogram. The data value is in hexadecimal characters.
			Required if the online EMV support is needed for this transaction, and the original request contained the EMV support data (the user did not 'fall back' on the alternative method of authorizing the transaction request).
\<IssScriptData\>	HexBin-256	Optional	Issuer Script Data Aggregate.
			If omitted, the issuer does not intend to send any Script Data (commands) to the terminal for further passing on to the ICC (chip card).

3.2.2.3   Issuer Script Data Aggregate \<IssScriptData\>

 The \<IssScriptData\> aggregate specifies the format for Issuer Script Commands for transmission to the card for processing by the card application. These commands are issued by the terminal, in the order specified in this aggregate, to the ICC (chip card) for processing. The issuer script commands include both types of commands that are issues by the terminal before (\<IssPreScriptData\>) and after (\<IssPostScriptData\>) issuing the final EMV GENERATE AC command.

Tag	Type	Usage	Description
\<IssPreScriptData\>	Aggregate	Optional Repeating	Issuer 'Pre' Script Data (commands issued by the terminal before the final EMV GENERATE AC command).
\<IssScriptId\>	HexBin-4	Optional *but see Description*	Issuer Script Identification. Uniquely identifies the 'pre' script command.
			Required if the \<IssPreScriptData\> occurs more than once (multiple 'pre' script commands are present).

Tag	Type	Usage	Description
<IssScriptCmd>	HexBin-124	Required Repeating	Issuer Script Command. Contains an Issuer 'Pre' Script Command to be processed by the ICC card at the terminal. If multiple Issuer 'Pre' Script Commands are present here, then the terminal must process them in the sequence they are listed here.
</IssPreScriptData>			
<IssPostScriptData>	Aggregate	Optional Repeating	Issuer 'Post' Script Data (commands issued by the terminal after the final EMV GENERATE AC command).
<IssScriptId>	HexBin-4	Optional *but see Description*	Issuer Script Identification. Uniquely identifies the 'post' script command. Required if the <IssPostScriptData> occurs more than once (multiple 'post' script commands are present).
<IssScriptCmd>	HexBin-124	Required Repeating	Issuer Script Command. Contains an Issuer 'Post' Script Command to be processed by the ICC card at the terminal. If multiple Issuer 'Post' Script Commands are present here, then the terminal must process them in the sequence they are listed here.
</IssPostScriptData>			

3.2.3   Customer Data

3.2.3.1   Customer Identification Aggregate <CustId>

The <CustId> aggregate is used to uniquely identify the customer who submits a request. Its use is optional when the customer ID may be inferred from the session (i.e., when it matches the <CustId> in the <SignonRq>).

When the <CustId> is used in messages between a Customer and CSP, the <CustPermId> element is an optional element and the <CustLoginId> is a required element. This is because the CSP server assigns the <CustPermId>, and the client may not know this ID the first time it logs in. When <CustId> is used in messages between CSP and BSP or other service provider, <CustPermId> may be required with <CustLoginId> optional. The <SPName> indicates the CSP that issued the customer permanent id.

Because the <CustPermId> is used as a key field with many SPs, the <CustPermId> must not be changed once assigned. Future versions of IFX may permit this identifier to be changed by providing messages to communicate these changes to other SPs.

Tag	Type	Usage	Description
<SPName>	Identifier	Required	Service Provider Name. The SP that assigned the <CustPermId>.
<CustPermId>	Identifier	Required OR	Customer Permanent ID. Used as a database key to uniquely identify an FI or CSP customer. Cannot be changed by the customer. When <CustId> is used in messages between CSP and BSP or other service provider, <CustPermId> may be required with <CustLoginId> optional.
<CustLoginId>	Identifier	Required OR	Customer Login ID. Used as a user-friendly name for the customer for authentication purposes. Maps directly to Customer Permanent ID. Some implementations may allow a user to change his or her Login ID. When the <CustId> is used in messages between a Customer and CSP, the <CustPermId> element is an optional element and the <CustLoginId> is a required element.

Tag	Type	Usage	Description
<CardMagData>	Aggregate *see section 3.2.7.1.3.1*	Required OR	Card Magnetic Stripe Data

3.2.3.2   Customer Name Aggregate <CustName>

The <CustName> aggregate is used to specify a customer's name and, optionally, nickname. This aggregate will be deprecated in IFX 2.0, replaced by <PersonName>.

Tag	Type	Usage	Description
<LastName>	C-40	Required Repeating	Customer Last Name.
<FirstName>	C-40	Required Repeating	Customer First Name.
<MiddleName>	C-40	Optional Repeating	Customer Middle Name.
<TitlePrefix>	C-8	Optional	Customer Title Prefix. For example, "Ms.", or "Dr."
<NameSuffix>	C-40	Optional	Customer Name Suffix. For example, "MD" or "Jr."
<Nickname>	C-40	Optional	Customer Nickname. Assigned by customer if desired.
<LegalName>	C-96	Optional	Used by entities like trusts, or businesses in the case where an individual is "doing business as" a company. Organizations and companies will be supported more fully in future versions of IFX.

3.2.3.3   Customer Contact Aggregate <CustContact>

The <CustContact> aggregate appears wherever the customer's contact information is needed. The customer provides this information to the SP regarding how and when to contact him or her. This aggregate is most commonly used to allow the customer to override the default customer contact information stored as part of the Customer Profile. This aggregate will be deprecated in IFX 2.0, replaced by <ContactInfo>.

Tag	Type	Usage	Description
<CustContactPref>	Open Enum	Optional	Customer Contact Preference. This is a customer-provided preference for contact by FI and SP staff.
			Defined values: DayPhone, EvePhone, DayFax, EveFax, Email, Post.
<PrefTimeStart>	Time	Optional	Preferred Customer Contact Start Time. This is a customer-provided start time preference for contact by SP staff.
<PrefTimeEnd>	Time	Optional	Preferred Customer Contact End Time. This is a customer-provided end time preference for contact by SP staff.
<DayPhone>	Phone Number	Optional *but see Description*	Customer Daytime Telephone Number. Required if <CustContactPref> = DayPhone.
<EvePhone>	Phone Number	Optional *but see Description*	Customer Evening Telephone Number. Required if <CustContactPref> = EvePhone.
<DayFax>	Phone Number	Optional *but see Description*	Customer Daytime Fax Number. Required if <CustContactPref> = DayFax.
<EveFax>	Phone Number	Optional *but see Description*	Customer Evening Fax Number. Required if <CustContactPref> = EveFax.

Tag	Type	Usage	Description
<EmailAddr>	NC-128	Optional *but see Description*	Customer Email Address. Required if <CustContactPref> = Email.

3.2.3.4   Customer Name/Address Aggregate <CustNameAddr>

This aggregate will be deprecated in IFX 2.0, to be replaced by <PersonInfo>.

Tag	Type	Usage	Description
<NameAddrType>	Open Enum	Required	Name/Address type. Defined values: Billing, Delivery, Customer
<FullName>	C-96	Required Repeating XOR	Concatenated customer name associated with the address.
<CustName>	Aggregate *see section 3.2.3.2*	Required XOR	Customer name. The customer name associated with the address.
<PersonName>	Aggregate *see section 3.2.4.1.1*	Required XOR	Person name.
<PostAddr>	Aggregate *see section 3.2.3.7.1*	Required	Postal address.
<CustContact>	Aggregate *see section 3.2.3.3*	Optional XOR	Customer contact aggregate.
<ContactInfo>	Aggregate *see section 3.2.3.7*	Optional XOR	Contact Information Aggregate.

3.2.3.5   Customer Preference Aggregate <CustPref>

The <CustPref> aggregate is used to specify a customer's preferences or for the service provider to store values that are specific to a customer. Applications can use the values stored in this aggregate to tailor the customer's experience during an interaction with a service.

Tag	Type	Usage	Description
<Language>	NC-17	Optional	Language. This element contains the customer's preferred language of communication. See section 2.6.4 for more information on the format of this element.
<MktgInfo>	C-255	Optional	Marketing Information.
<CustBankSvcPref>	Aggregate	Optional	Customer Banking Service Preferences
<CustWithdrawalPref>	Aggregate	Optional	Customer Withdrawal Preferences. Used to store the customer's preferred withdrawal parameters for "fast cash" ATM withdrawals.
<CurAmt>	Currency Amount	Required	Currency Amount. Identifies the customer's preferred amount to be withdrawn.
<DepAcctId>	Aggregate *see section 3.2.7.1.2*	Optional	Deposit Account Identifier. This is the customer's preferred deposit account identifier. If absent, the value defaults to the primary account for the customer (usually a checking account).

Tag	Type	Usage	Description
</CustWithdrawalPref>			
</CustBankSvcPref>			

3.2.3.6 Composite Contact Information Aggregate <CompositeContactInfo>

Tag	Type	Usage	Description
<ContactInfoType>	Open Enum	Required	Contact Information Type. The specific type of the contact referenced in the <ContactInfo> aggregate.
			Defined values: CustServ, LostCard, GeneralInfo
<ContactInfo>	Aggregate *see section 3.2.3.7*	Required	Contact Information.

3.2.3.7 Contact Information Aggregate <ContactInfo>

The <ContactInfo> aggregate appears wherever a person's or organization's contact information is needed. It is used in IFX 1.1 and later versions as a replacement for the <CustContact> and <OrgContact> aggregates that will be deprecated in IFX 2.0.

Tag	Type	Usage	Description
<ContactPref>	Open Enum	Optional	Contact Preference. This is the preferred method to reach the entity to whom the <ContactInfo> refers.
			Defined values: DayPhone, EvePhone, DayFax, EveFax, Email, Post.
<PrefTimeStart>	Time	Optional	Preferred Contact Start Time. This is the beginning of the window of time within which contact is preferred.
<PrefTimeEnd>	Time	Optional	Preferred Contact End Time. This is the end of the window of time within which contact is preferred.
<PhoneNum>	Aggregate	Optional *but see Description* Repeating	Phone number aggregate. Required if <ContactPref> = DayPhone, EvePhone, DayFax, or EveFax.
<PhoneType>	Open Enum	Required	Phone number type.
			Defined values: DayPhone, EvePhone, DayFax, EveFax
<Phone>	Phone Number	Required	Phone Number.
</PhoneNum>			
<EmailAddr>	NC-128	Optional *but see Description*	Email Address.
			Required if <ContactPref> = Email.
<URL>	URL	Optional	Web Site Address.
<PostAddr>	Aggregate *see section 3.2.3.7.1*	Optional *but see Description* Repeating	Postal Address. Required if <ContactPref> = Post

3.2.3.7.1 Postal Address Aggregates <PostAddr>, <BillingAddr>, <BillRetAddr>, <RemitAddr>

The <PostAddr> aggregate is used wherever a postal address is needed. There are some cases where a synonym is used to distinguish the aggregate from another of the same structure in the same aggregate (e.g., the Billing Address <BillingAddr>, where more than one postal address is present).

Tag	Type	Usage	Description
<Addr1>	C-64	Required	Address Line 1.
<Addr2>	C-64	Optional *but see Description*	Address Line 2. Required if <Addr3> is present.
<Addr3>	C-64	Optional *but see Description*	Address Line 3. Required if <Addr4> is present.
<Addr4>	C-64	Optional	Address Line 4.
<City>	C-32	Optional	City.
<StateProv>	C-32	Optional	State or Province.
<PostalCode>	C-11	Optional	Postal Code.
<Country>	NC-3	Required	Country. Values are defined by ISO-3166 3-letter codes.
<AddrType>	Open Enum	Optional	Address Type. Defined values: Seasonal, Primay, Secondary, Business, Home
<StartDt>	DateTime	Optional	Start Date. Effective start date for this address.
<EndDt>	DateTime	Optional	End Date. Effective end date for this address. Address is current if <EndDt> is not present.

3.2.4   Person Data

3.2.4.1   Person Information Aggregate <PersonInfo>

The <PersonInfo> aggregate is used to specify the details about a person. It is used in IFX 1.1 and later versions as a replacement for the <CustNameAddr> aggregate that will be deprecated in IFX 2.0.

Tag	Type	Usage	Description
<NameAddrType>	Open Enum	Required	Name/Address type. Defined values: Billing, Delivery, Customer
<FullName>	C-96	Required XOR	Concatenated person name.
<PersonName>	Aggregate *see section 3.2.4.1.1*	Required XOR	Person name.
<ContactInfo>	Aggregate *see section 3.2.3.7*	Optional	Contact information aggregate.
<TINInfo>	Aggregate *see section 3.2.4.1.3*	Optional	Customer Tax Identifier Number aggregate.
<BirthDt>	Date	Optional	Date of Birth.
<DriversLicense>	Aggregate *see section 3.2.4.1.2*	Optional	Driver's License aggregate.
<MotherMaidenName>	C-40	Optional	Mother's Maiden Name.

Tag	Type	Usage	Description
<SpouseName>	C-40	Optional	Spouse's Name.
<EmploymentHistory>	Aggregate *see section 3.2.4.1.4*	Optional Repeating	Employment History Aggregate.
<Gender>	Open Enum	Optional	Gender.
			Defined values:
			Male, Female, Unknown
<MaritalStatus>	Open Enum	Optional	Marital Status.
			Defined values:
			Married, Single, Divorced, Widow
<USA.MilitaryRank>	C-32	Optional	Military Rank.
<Dependents>	Long	Optional	Number of Dependents.
<PassportNumber>	C-32	Optional	Passport Number.
<PassportCountry>	NC-3	Optional	Passport Country.
<OEDCode>	Open Enum	Optional	Officer, Employee, Director Code.
			Defined values:
			Officer, Director, Employee
<OEDInstitution>	Open Enum	Optional	OED Institution.
			Defined values:
			Unknown, Affiliate Financial Institution, Holding Company, Other Financial Institution, This Financial Institution

3.2.4.1.1   Person Name Aggregate <PersonName>

The <PersonName> aggregate is used to specify a person's name and, optionally, nickname. It is used in IFX 1.1 and later versions as a replacement for the <CustName> aggregate that will be deprecated in IFX 2.0.

Tag	Type	Usage	Description
<LastName>	C-40	Required Repeating	Customer Last Name.
<FirstName>	C-40	Required	Customer First Name.
<MiddleName>	C-40	Optional Repeating	Customer Middle Name.
<TitlePrefix>	C-8	Optional	Customer Title Prefix. For example, "Ms.", or "Dr."
<NameSuffix>	C-40	Optional	Customer Name Suffix. For example, "MD" or "Jr."
<Nickname>	C-40	Optional	Customer Nickname. Assigned by customer if desired.
<LegalName>	C-96	Optional	Used by entities like trusts, or businesses in the case where an individual is "doing business as" a company. Organizations and companies will be supported more fully in future versions of IFX.

3.2.4.1.2   Drivers License Aggregate <DriversLicense>

Driver's license.

Tag	Type	Usage	Description

Tag	Type	Usage	Description
<LicenseNum>	C-32	Required	Driver's License Number.
<StateProv>	C-32	Optional	Driver's License State/Province.
<Country>	NC-3	Optional	Driver's License Country.

3.2.4.1.3   Tax Identification Number Aggregate <TINInfo>

Tax identification number aggregate.

Tag	Type	Usage	Description
<TINType>	Open Enum	Optional	TIN Code Type.
			Defined values:
			EIN (Business TIN), SSN (Personal TIN), NRA Business, NRA Personal.
<TaxId>	NC-12	Required	Tax Identifier.
<CertCode>	Open Enum	Optional	Certification Code for the Tax ID.
			Defined values:
			Certification Not on File, Certification on File, Account Opened After 1987, Account Opened 1984-1987, Account Open Before 1984, Certified, Uncertified, Notice 1, Notice 2, Cert Not Required

3.2.4.1.4   Employment History Aggregate <EmploymentHistory>

Employment history aggregate.

Tag	Type	Usage	Description
<EmploymentStatus>	Open Enum	Optional	Employment Status.
			Defined values:
			Current, Previous
<OrgInfo>	Aggregate *see section 3.2.5.1.2*	Optional	Employer Organization Information.
<Occupation>	C-50	Required	Occupation
<Income>	Currency Amount	Optional	Income Amount
<JobTitle>	C-32	Optional	Job Title.
<StartDt>	DateTime	Optional	Employment Start Date.
<EndDt>	DateTime	Optional	Employment End Date.

3.2.5   Organization Data

3.2.5.1   Organization Record <OrgRec>

Tag	Type	Usage	Description
<OrgId>	Aggregate *see section 3.2.5.1.1*	Optional Repeating	Organization Identifier

Tag	Type	Usage	Description
<OrgInfo>	Aggregate *see section 3.2.5.1.2*	Required	Organization Information.

3.2.5.1.1 Organization Information <OrgId>

Tag	Type	Usage	Description
<OrgIdType>	Open Enum	Required	Organization Identifier Type.
			Defined Values: DunAndBradstreet, USA.TaxId, OrgSpecific
<OrgIdNum>	Identifier	Required	Organization Identifier Number

3.2.5.1.2 Organization Information <OrgInfo>

Tag	Type	Usage	Description
<IndustId>	Aggregate *see section 3.2.13*	Optional	Industry Identifier. The identifier of the industry in which the organization does business.
<Name>	C-40	Optional	Name. The name of the organization.
<LegalName>	C-96	Optional	Legal Name. The legal name of the organization.
<CompositeContactInfo>	Aggregate *see section 1.1.1.1*	Optional Repeating	Composite Contact Information Aggregate.
<TINInfo>	Aggregate *see section 3.2.4.1.3*	Optional	Tax Identifier Number Aggregate.
<EstablishDt>	Date	Optional	Date Established.
<NumEmployees>	Long	Optional	Number of Employees.

3.2.5.2 Organization Contact Aggregate <OrgContact>

The <OrgContact> aggregate appears wherever contact information for an organization is needed. Contact type indicates the function of the contact. This aggregate will be deprecated in IFX 2.0, replaced by <ContactInfo>.

Tag	Type	Usage	Description
<ContactType>	Open Enum	Required	Contact Type.
			Defined values: CustSvc, Enroll, Tech, Personal
<Desc>	C-80	Optional	Description. Text describing the purpose of the group providing contact information is provided.
<Phone>	Phone Number	Optional	Telephone Number.
<Fax>	Phone Number	Optional	Fax Number.
<EmailAddr>	NC-128	Optional	Email Address.
<URL>	URL	Optional	Web Site Address.

3.2.6 Biller Data

3.2.6.1 Biller Contact Aggregate <BillerContact>

The <BillerContact> aggregate is used to provide complete information about a Biller.

Tag	Type	Usage	Description
<LegalName>	C-96	Optional	Biller Legal Name. Assigned by the service provider. Cannot be changed by the client.
<Name>	C-40	Optional	Biller Business Name. Assigned by the service provider. Cannot be changed by the client.
<PostAddr>	Aggregate *see section 3.2.3.7.1*	Optional	Biller Address Aggregate. Assigned by the service provider. Cannot be changed by the customer.
<BillRetAddr>	Aggregate *see section 3.2.3.7.1*	Optional	Bill Return Address Aggregate. Uses the same structure as <PostAddr>. Assigned by the service provider. Cannot be changed by the customer.
<RemitName>	C-40	Optional	Remittance Name. Uses the same structure as <PostAddr>. Assigned by the service provider. Cannot be changed by the customer.
<RemitAddr>	Aggregate	Optional	Remittance Address Aggregate. Uses the same structure as <PostAddr>. Assigned by the service provider. Cannot be changed by the customer.
<OrgContact>	Aggregate *see section 3.2.5.2*	Optional XOR Repeating	Support Contact Detail Aggregate.
<ContactInfo>	Aggregate *see section 3.2.3.7*	Optional XOR Repeating	Contact Information Aggregate.

3.2.6.2 Biller Pay Information Aggregate <BillerPayInfo>

The <BillerPayInfo> identifies payment information associated with a Biller.

Tag	Type	Usage	Description
<PmtInst>	Aggregate *see section 3.2.15*	Optional Repeating Profiled values	Payment Instrument Aggregate. Types of payment that the biller accepts via the Pay provider. This is used to restrict the methods by which the customer may pay for the bill. The Service Provider indicates which Payment Instruments are supported via profile. The Biller may choose to override this list by including <PmtInst> in the Bill Summary <BillRec>.
<DaysToEPost>	Long	Optional	Days to post an electronic payment. The maximum days between receipt of an electronic payment by the biller or BPP and posting to the customer's account.
<PrenoteReqd>	Boolean	Optional	Pre-Note Required. If set to *True*, the Biller requires pre-noting by the Service Provider. **Note:** *Pre-noting is the transmission of a zero dollar message to verify consumer information, usually the payee account information.*

3.2.7 Account Data

3.2.7.1 Bank Account Record Aggregate <BankAcctRec>

The <BankAcctRec> aggregate provides information about a customer banking account.

Tag	Type	Usage	Description
<DepAcctId>	Aggregate *see section 3.2.7.1.2*	Required XOR	Deposit Account Identification Aggregate.
<CardAcctId>	Aggregate *see section 3.2.7.1.3*	Required XOR	Card Account Identification Aggregate.
<LoanAcctId>	Aggregate *see section 3.2.7.1.4*	Required XOR	Loan Account Identification Aggregate.
<BankAcctInfo>	Aggregate *see section 3.2.7.1.5*	Required	Bank Accont Information Aggregate
<BankAcctStatus>	Aggregate *see section 3.2.7.1.6*	Optional	Bank Account Status Aggregate
<AcctTaxType>	Open Enum	Optional	Account Tax Type. Defined values: TaxDeferred, Standard.
<XferFromSupt>	Aggregate	Optional	Transfer From Supported Aggregate. If present, indicates that this account is a valid source account for a transfer. This aggregate will be deprecated in IFX 2.0, to be replaced by <BankAcctFeatSupt>.
<MinCurAmt>	Currency Amount	Optional	Minimum amount that may be transferred from the account, if applicable.
<MaxCurAmt>	Currency Amount	Optional	Maximum amount that may be transferred from the account, if applicable.
</XferFromSupt>			
<XferToSupt>	Aggregate	Optional	Transfer To Supported Aggregate. If present, indicates that this account is a valid target account for a transfer. This aggregate will be deprecated in IFX 2.0, to be replaced by <BankAcctFeatSupt>.
<MinCurAmt>	Currency Amount	Optional	Minimum amount that may be transferred to the account, if applicable.
<MaxCurAmt>	Currency Amount	Optional	Maximum amount that may be transferred to the account, if applicable.
</XferToSupt>			
<PaySupt>	Aggregate	Optional	Payment Supported Aggregate. If present, indicates that this account is a valid source account for payment. This aggregate will be deprecated in IFX 2.0, to be replaced by <BankAcctFeatSupt>.
<MinCurAmt>	Currency Amount	Optional	Minimum amount that may be paid from the account, if applicable.
<MaxCurAmt>	Currency Amount	Optional	Maximum amount that may be paid from the account, if applicable.
</PaySupt>			

Tag	Type	Usage	Description
\<BankAcctFeatSupt\>	Aggregate *see section 3.2.7.1.7*	Optional repeating	Supported Bank Account Features Aggregate. If present, indicates the various features supported for this Bank Account
\<AcctBal\>	Aggregate *see section 3.2.8.1*	Optional Repeating	Account Balance Aggregate. Repeated for each balance to be included for this account.

3.2.7.1.1   Bank Information Aggregate \<BankInfo\>

The Bank Information aggregate contains additional information used to identify a bank. Sufficient information *must* be provided within this aggregate to uniquely identify the bank at which the account is held. Failure to do so will result in an error being returned from the server.

Tag	Type	Usage	Description
\<BankIdType\>	Open Enum	Optional	Bank Identifier Type. Type of bank identifier. Defined Values: SWIFT, ABA, FedWire, CHIP, CHAP, SortCode. Default value is ABA.
\<BankId\>	NC-34	Optional	Bank Identifier. Qualifies account number if known by the customer/client. Usage is expected to be routing and transit number in the US or the equivalent in an international implementation. This is required in USA.
\<RefInfo\>	Aggregate *see section 3.2.17*	Optional Repeating	Additional Reference information to uniquely identify the bank. Defined values for contained \<RefType\>: CountrySpecific, BankSpecific
\<Name\>	C-40	Optional *but see Description*	Bank name, required in CHE, DEU, and ITA.
\<BranchId\>	NC-22	Optional	Branch Identifier. Qualifies account number if known by the customer/client. Used to indicate which branch the account is with. Usage is expected to be primarily in countries where regulations require an account to be assigned to a branch office, but a US-based FI may also require this field for operational reasons.
\<BranchName\>	C-40	Optional *but see Description*	Bank branch name, required in ITA.
\<PostAddr\>	Aggregate *see section 3.2.3.7.1*	Optional XOR	Bank Postal Address. Either \<PostAddr\> or broken-out \<City\>, \<StateProv\>, \<PostalCode\>, and/or \<Country\> may be used, but not both.
\<City\>	C-32	Optional *but see Description* XOR	Bank Branch City. Required in CHE. Either \<PostAddr\> or broken-out \<City\>, \<StateProv\>, \<PostalCode\>, and/or \<Country\> may be used, but not both.
\<StateProv\>	C-32	Optional *but see Description*	Bank Branch State or Province. Either \<PostAddr\> or broken-out \<City\>, \<StateProv\>, \<PostalCode\>, and/or \<Country\> may be used, but not both.
\<PostalCode\>	C-11	Optional *but see Description*	Bank Branch Postal Code. Required in CHE. Either \<PostAddr\> or broken-out \<City\>, \<StateProv\>,\<PostalCode\>, and/or \<Country\> may be used, but not both.

Tag	Type	Usage	Description
<Country>	NC-3	Optional but see Description	Country. Bank Branch Country Code. Either <PostAddr> or broken-out <City>, <StateProv>, <PostalCode>, and/or <Country> may be used, but not both.

3.2.7.1.2 Deposit Account Identifier Aggregates <DepAcctId> <DepAcctIdFrom> <DepAcctIdTo>

The <DepAcctId> aggregate is used to uniquely identify a deposit-type account. When a single account is specified in a message or aggregate, it is referred to as <DepAcctId>. When multiple accounts appear in a message or aggregate, the <DepAcctIdFrom> and <DepAcctIdTo> aggregates are used for clarity. The <DepAcctIdFrom> and <DepAcctIdTo> aggregates have the same structure as the <DepAcctId> aggregate.

Tag	Type	Usage	Description
<AcctId>	C-32	Required	Account Identifier. Sometimes known as account number, but not restricted to numeric characters.
			Note: *An International Bank Account Number (IBAN) can be used here to replace the domestic account number.*
<AcctType>	Open Enum	Required	Account Type. See Data Dictionary for details.
			Defined values:
			DDA, SDA, CDA, MMA, CMA.
			Note: *An Open Enum data type permits the addition of account types specific to an FI, country, etc.*
<AcctKey>	NC-22	Optional	Account Key. Checksum for international banks.
<AcctCur>	NC-22	Optional	Account Currency. Currency of the account. This may be necessary to uniquely identify the account, as many countries allow for a single account to contain multiple subaccounts, each in a different currency.
<BankInfo>	Aggregate *see section 3.2.7.1.1*	Required	Bank Information aggregate.

3.2.7.1.2.1 Intermediary Deposit Account Aggregate <IntermediaryDepAcct>

The Intermediary Bank Account aggregate is used to identify all intermediary bank accounts in a bank chain. This aggregate is used within the payment instruction to identify all banks that route the payment from the originating bank or <DepAcctIdFrom> to the receiving bank or <DepAcctIdTo>.

Tag	Type	Usage	Description
<SeqNum>	Long	Required	Sequence number of the Intermediary bank in a bank chain.
<DepAcctId>	Aggregate *see section 3.2.7.1.2*	Required	Deposit Account Identification Aggregate for the intermediary bank.

3.2.7.1.3 Card Account Identifier Aggregates <CardAcctId> <CardAcctIdFrom> <CardAcctIdTo>

The <CardAcctId> aggregate is used to uniquely identify a card-type account, such as credit card or debit card. When a single account is specified in a message or aggregate, it is referred to as <CardAcctId>. When multiple accounts appear in a message or aggregate, the <CardAcctIdFrom> and <CardAcctIdTo> aggregates are used for clarity. The <CardAcctIdFrom> and <CardAcctIdTo> aggregates have the same structure as the <CardAcctId> aggregate.

Tag	Type	Usage	Description
<AcctId>	NC-32	Required XOR	Account Identifier. Sometimes known as account number, but not restricted to numeric characters.
<CardMagData>	Aggregate *see section 3.2.7.1.3.1*	Required XOR	Card Magnetic Stripe Data
<AcctType>	Open Enum	Required	Account Type. See Data Dictionary for details. Defined values: CCA, DDA, SDA, Default, Unknown
<CCMotoAcct>	Aggregate	Optional	Credit Mail Order/Telephone Order Account Aggregate. Used for defining additional Credit Card data when required for "card not present" transactions, such as when activating a Credit Card as a funding account for payment.
<ExpDt>	DateTime	Required	Expiration date for card. If the card has only month and year expiration, the *last* day of the month must be specified here.
<Name>	C-40	Optional	Customer Name. Name on the front of this card (exactly as name appears on the card).
<PostAddr>	Aggregate *see section 3.2.3.7.1*	Optional XOR	Customer Address Aggregate. Customer address associated with this card.
<PostalCode>	C-11	Optional XOR	Customer Postal Code. The postal code associated with this card.
<Phone>	Phone Number	Optional	Customer phone number associated with this card.
<Brand>	Open Enum	Optional	Type of credit card (VISA, AMEX, etc.). This is just a check against the information provided by the account number. Defined values: AmEx, CarteBlanche, DinersClub, Discover, Europay, JCB, MasterCard.
</CCMotoAcct>			

3.2.7.1.3.1 Card Magnetic Stripe Data <CardMagData>

Tag	Type	Usage	Description
<MagData1>	NC-79	Required OR	Track 1 data
<MagData2>	NC-40	Required OR	Track 2 data
<MagData3>	NC-107	Required OR	Track 3 data

3.2.7.1.4 Loan Account Identifier Aggregates <LoanAcctId> <LoanAcctIdFrom> <LoanAcctIdTo>

The <LoanAcctId> aggregate is used to uniquely identify a loan-type account. When a single account is specified in a message or aggregate, it is referred to as <LoanAcctId>. When multiple accounts appear in a message or aggregate, the <LoanAcctIdFrom> and <LoanAcctIdTo> aggregates are used for clarity. The <LoanAcctIdFrom> and <LoanAcctIdTo> aggregates have the same structure as the <LoanAcctId> aggregate.

Tag	Type	Usage	Description
<AcctId>	NC-32	Required	Account Identifier. Sometimes known as account number, but not restricted to numeric characters.

Tag	Type	Usage	Description
<AcctType>	Open Enum	Required	Account Type. See Data Dictionary for details.
			Defined values: MLA, ILA, LOC, EQU, CLA.
			Note: *An Open Enum data type permits the addition of account types specific to an FI, country, etc.*
<BankInfo>	Aggregate *see section 3.2.7.1.1*	Required	Bank Information aggregate.

3.2.7.1.5   Bank Account Information Aggregate <BankAcctInfo>

Tag	Type	Usage	Description
<CurCode>	NC-3	Required	Currency Code. As defined by ISO-4217.
<Desc>	C-80	Optional	Account Description. Assigned by the Financial Institution. Typically product name. Does not include account number.
<CustName>	Aggregate *see section 3.2.3.2*	Optional XOR	Customer Name in which this account is held.
<PersonName>	Aggregate *see section 3.2.4.1.1*	Optional XOR	Person Name.
<OrgContact>	Aggregate *see section 3.2.5.2*	Optional XOR	FI Customer Service Contact Information.
<ContactInfo>	Aggregate *see section 3.2.3.7*	Optional XOR	Contact Information Aggregate.
<CustAcctUse>	Open Enum	Optional	Customer Account Use.
			Defined values: Retail, Business.
<PrimaryAcct>	Boolean	Optional	Primary Account Indicator. If *True*, used as default account for a class of accounts (i.e., checking). Typically used in ATM networks. May be used for other applications by some FIs.
<Term>	Aggregate *see section 3.2.9.1*	Optional	Term Aggregate.
<IntRateInfo>	Aggregate *see section 6.3.30.4*	Optional	Interest Rate Information Aggregate.
<BackupWithholding>	Boolean	Optional	Backup Withholding Indicator.
<TaxCountry>	NC-3	Optional	Tax Country.
<TaxProvince>	C-32	Optional	Tax Province.
<Ownership>	C-42	Optional	Ownership Identifier.
<ProductId>	C-42	Optional	Product Identifier.
<InitialDeposit>	Currency Amount	Optional	Initial Deposit Amount.

3.2.7.1.6   Bank Account Status <BankAcctStatus>

Tag	Type	Usage	Description

Tag	Type	Usage	Description
\<BankAcctStatusCode\>	Closed Enum	Required	Account Status.
			Valid values: Open, Closed, Inactive, NotAvail
\<StatusDesc\>	C-255	Optional	Status Description. Explanatory text associated with this bank account status.
\<EffDt\>	DateTime	Optional	Effective Date/Time. The date/time the \<BankAcctStatusCode\> took effect.
\<StatusModBy\>	Open Enum	Optional	Status Modified By. If present, indicates who modified the Bank Status Code.
			Defined values: Customer, FI, CSP, CSPSR, BSP, BSPSR, CPP, CPPSR, BPP, and BPPSR.

3.2.7.1.7   Bank Account Feature Support \<BankAcctFeatSupt\>

The \<BankAcctFeatSupt\> aggregate provides information about various features supported by a customer banking account. This aggregate contains two \<TrnSrc\> elements. The first \<TrnSrc\> element (contained within \<BankAcctFeatLimit\>) indicates features supported by the transaction source. The second \<TrnSrc\> indicates the transaction source that has a limit.

Tag	Type	Usage	Description
\<BankAcctFeatType\>	Open Enum	Required	Bank Account Feature Type. Indicates the type of feature supported.
			Defined Values:
			XferFrom, XferTo, Pay, Debit, Credit
\<TrnSrc\>	Open Enum	Optional Repeating	Transaction Source. Indicates the transaction source for which this feature is supported (absence indicates all sources). This is used to indicate features supported by transaction source regardless of specific limits.
			Defined values: ATM, Teller, POS, VRU, HomeBank, ACH
\<BankAcctFeatLimit\>	Aggregate	Optional Repeating	Bank Account Feature Limit. Indicates any limits relating to the feature described by \<BankAcctFeatType\>.
\<BankAcctFeatLimitType\>	Open Enum	Required	Bank Account Feature Limit Type. Indicates the type of limit.
			Defined Values:
			Min, Max, DailyMin, DailyMax, WeeklyMin, WeeklyMax, MonthlyMin, MonthlyMax
\<TrnSrc\>	Open Enum	Optional	Transaction Source. Indicates the transaction source for which this limit applies (absence indicates that the limit applies to all sources). This is used if the limits are different for different transaction sources.
			Defined values: ATM, Teller, POS, VRU, HomeBank, ACH
\<CurAmt\>	Currency Amount	Required	Currency Amount. The amount of the limit.
\</BankAcctFeatLimit\>			

3.2.7.2 Presentment Account Record Aggregate <PresAcctRec>

Tag	Type	Usage	Description
<PresAcctId>	Aggregate *see section 3.2.7.2.1*	Required	Presentment Account Identification Aggregate
<PresAcctInfo>	Aggregate *see section 3.2.7.2.2*	Required	Presentment Account Information Aggregate
<PresAcctStatus>	Aggregate *see section 3.2.7.2.3*	Optional	Presentment Account Status Aggregate

3.2.7.2.1 Presentment Account Identification <PresAcctId>

The <PresAcctId> aggregate uniquely identifies a customer's account with a biller by the combination of BSP <Org>, biller identifier <BillerNum>, and account number <BillingAcct>. <BillerNum>s must be unique within a BSP.

Upon account activation, the <CustNameAddr> (billing account information) has to be correlated to the <CustPermId> from the CSP's records by either or both the CSP and the BSP to ensure that the Customer's identification as known to the CSP is properly matched to the Customer's identification as known to the Biller. For account activation, this aggregate also provides the CSP with the ability to send the BSP the <StdPayeeId> or the <CustPayeeId> that is associated with this biller at the appropriate CPP. If the BSP stores these data, they must be returned in any inquiry response that includes <PresAcctId> (e.g. <BillInqRs>).

Tag	Type	Usage	Description
<BillingAcct>	C-32	Required	Customer account with the Biller.
<BillerId>	Aggregate *see section 8.3.1.1*	Required	Biller Identifier Aggregate. This is an identifier created by the Biller or BSP.

3.2.7.2.2 Presentment Account Information Aggregate <PresAcctInfo>

Tag	Type	Usage	Description
<StdPayeeId>	Aggregate *see section 7.3.1.1*	Optional XOR	Standard Payee Identifier. The identification for this biller at the CPP. When sent in account activation, it is intended for storage on the BSP database, such that it may be returned in subsequent inquiries utilizing this aggregate. This may be used by the CSP to help match the BSP's Biller ID to the Standard Payee ID as known to the CPP.
<CustPayeeId>	Identifier	Optional XOR	Customer Payee Identifier. The identification for this biller at the CPP. When sent in account activation, it is intended for storage on the BSP database, such that it may be returned in subsequent inquiries utilizing this pair of elements. This may be used by the CSP to help match the BSP's Biller ID to the Standard Payee ID as known to the CPP.
<SPName>	Identifier	Required	Service Provider Name. Used to scope <CustPayeeId>.

3.2.7.2.3 Presentment Account Status <PresAcctStatus>

The <PresAcctStatus> aggregate describes the status of the customer's account with a biller.

Tag	Type	Usage	Description

Tag	Type	Usage	Description
<PresAcctStatusCode>	Closed Enum	Required	Account Status.
			Valid values: Open, Closed, Inactive, NotAvail
<StatusDesc>	C-255	Optional	Status Description. Explanatory text associated with this presentment account status.
<EffDt>	DateTime	Optional	Effective Date/Time. The date/time the <BankAcctStatusCode> took effect.
<StatusModBy>	Open Enum	Optional	Status Modified By. If present, indicates who modified the status.
			Defined values: Customer, FI, CSP, CSPSR, BSP, BSPSR, CPP, CPPSR, BPP, and BPPSR.

3.2.8   Account Balances

The following tables identify the account balances applicable to a specific account type. The first table identifies the association between the account type and the specific account classification <DepAcctId>, <CardAcctId> and <LoanAcctId>. The second table identifies the relationship between account classification and balance types. All account types use the balances associated with their account classification.

The <EffDt> associated with each balance indicates the time and date as of which the balance is current. The frequency of changing this date depends on how often the FI/SP refreshes the balances for a specific account type. This time and date might be updated by every update message in an on-line bank or be refreshed once a day by a strip file implementation. Even within a single FI/SP, the Demand Deposit ("Checking") Account (DDA) might be online, while Savings (SDA) or Certificates of Deposit (CDA) might be updated overnight.

3.2.8.1   Account Balance Aggregate <AcctBal>

The <AcctBal> aggregate is used to express an account balance. The balance being specified is identified in <BalType>, which is an Open Enum. When <AcctBal> is used, it may repeat, to allow multiple balances to be specified for the given account.

Tag	Type	Usage	Description
<BalType>	Open Enum	Required	Balance Type.
			Defined Values: Ledger, OpeningLedger, ClosingLedger, MinLedger, AvgLedger, Avail, Current, Outstanding, OpeningOutstanding, ClosingOutstanding, AvailCredit, CreditLimit, PayoffAmt, Principal, Escrow, BAI:*xxx* or TMA:*xxx* (Reference BAI Code List at http://www.bai.org/operations/bai_codes.html, or TMA Code List at http://www.AFPonline.org/Information_Center/Publications/Bookstore/afpbooks/afpbooks.html as maintained by Association of Financial Professionals (AFP)) Example: BAI:010 (Beginning Ledger Balance), BAI:140 (ACH Credits), etc.
<CurAmt>	Currency Amount	Required	Currency Amount. This currency amount references the Balance Amount.
<EffDt>	DateTime	Optional	Effective DateTime.
<ExpDt>	DateTime	Optional	Expiration DateTime. Expiration Date of the Balance. For example, a payoff amount expiration date.
<Desc>	C-80	Optional	Description.

3.2.8.1.1   Balance Type and Classification Association

	Account Classification		

Balance Type	Deposit (Asset)	Loan (Liability)	Card (Asset or Liability)
Ledger	×		×
OpeningLedger	×		×
ClosingLedger	×		×
MinLedger	×		×
AvgLedger	×		×
Avail	×		×
Current	×		×
Outstanding		×	×
OpeningOutstanding		×	×
ClosingOutstanding		×	×
AvailCredit		×	×
CreditLimit		×	×
PayoffAmt		×	×
Principal		×	
Escrow		×	

3.2.8.2 Extended Account Balance Aggregate <ExtAcctBal>

The <ExtAcctBal> aggregate is used to express an extended account balance. The balance being specified is identified in <ExtBalType>, which is an Open Enum. When <ExtAcctBal> is used, it may repeat, to allow multiple balances to be specified for the given account.

Tag	Type	Usage	Description
<ExtBalType>	Open Enum	Required	Balance Type. Defined Values: PrepayPenalty, Orig, UnpaidAccruedInt, Redemption, YTDWithhold, LastYrWithhold, Overdraft, Overlimit, PastDue, CashLine, OutstandingCashAdv, CashAvail, PendAuthAmt, LastStmtBal, YTDInterest, LastYrInt, YTDFees, PeriodFees, LastYrFees, CashDepositAvail.
<CurAmt>	Currency Amount	Required	Balance Amount.
<EffDt>	DateTime	Required	Balance DateTime.
<ExpDt>	DateTime	Optional	Expiration Date of the Balance. For example, a payoff amount expiration date.
<Desc>	C-80	Optional	Description.

3.2.8.2.1 Extended Balance Type and Classification Association

Extended Balance Type	Account Classification		
	Deposit (Asset)	Loan (Liability)	Card (Asset or Liability)
Orig	×	×	×
UnpaidAccruedInt		×	
Redemption			
YTDWithhold			

LastYrWithhold			
Overdraft	×		
Overlimit			×
PastDue		×	×
CashLine		×	×
OutstandingCashAdv		×	×
CashAvail	×		×
PendAuthAmt		×	×
LastStmtBal	×	×	×
YTDInterest	×	×	×
LastYrInt	×	×	×
YTDFees	×	×	×
PeriodFees	×	×	×
LastYrFees	×	×	×

3.2.8.3 Account Type and Classification Association

<AcctType>	Account Classification		
	Deposit (Asset)	Loan (Liability)	Card (Asset or Liability)
CCA—Credit Card Account			×
CDA—Certificate of Deposit	×		
CLA—Commercial Loan Account		×	
CMA—Cash Management Account	×		
DDA—Demand Deposit Account	×		×
EQU—Home Equity Loan		×	
ILA—Installment Loan Account		×	
LOC—Consumer Line of Credit		×	
MLA—Mortgage Loan Account		×	
MMA—Money Market Account	×		
SDA—Savings Account	×		×

The sign of account balances are stated as positive values when greater than zero (outstanding balance due on a customer's liability account and positive balance in a customer's asset account) and negative numbers when less than zero (credit balance in a customer's liability account and overdrawn balance in a customer's asset account). For a Card Account, the <AcctType> is used to indicate whether the account is an asset or liability account.

3.2.9 Term Data

3.2.9.1 Deposit Term Aggregate <Term>

The <Term> aggregate contains details of the term of a deposit or a loan.

Tag	Type	Usage	Description

Tag	Type	Usage	Description
<Count>	Long	Required	Count. In this context, <Count> is the number of <TermUnits> for which a term deposit is effective. At the end of this period, the Term Deposit ceases to exist. ***Note:*** *If <TermUnits>=Days, this is the number of calendar days, not business days.*
<TermUnits>	Closed Enum	Required	Term Units. Units in which the Term Deposit is measured. Defined values: Days, Weeks, Months, Years, Indefinite
<Desc>	C-80	Optional	Short Description.
<DaysCall>	Long	Optional *but see Description*	Days Call. The number of business days' notice required for withdrawal. Required if <TermUnits>=Indefinite.

3.2.10 Selection Criteria

3.2.10.1 Selection Range Date Aggregates <SelRangeDt>, <SelRangeDueDt>, <SelRangePrcDt>

The <SelRangeDt> aggregate identifies the start date and/or the end date as a selection criterion. The Selection Range Due Date <SelRangeDueDt> and Selection Range Processing Date <SelRangePrcDt> have the same structure as the <SelRangeDt>.

The <SelRangeDt> aggregate contains two optional fields: <StartDt> and <EndDt>.

The following rules are established for the <StartDt> and <EndDt> fields:

- If <StartDt> but not <EndDt> is supplied, then the search must match dates that are later than or equal to <StartDt>.
- If <EndDt> is supplied but not <StartDt>, then the search must match dates that are earlier than or equal to the <EndDt>.
- If both dates are supplied, then the search must match dates that are between or equal to the two dates supplied.
- If a client wishes to search for a specific date, then both <StartDt> and <EndDt> should be supplied and set to the same date, with <EndDt> set for 23:59:59. If <StartDt> and <EndDt> both have no times included, the semantic intent of the request is to search for items that occurred precisely at midnight on the specified date.

   Note: *<StartDt> and <EndDt> are date fields typically entered by a consumer and are not timestamp fields. <UpDt>, on the other hand, is a timestamp (date and time) field found in many <xxxInqRq> messages. A client uses this timestamp to ask for any "xxx" updates created on the server after the value of <UpDt>. Along with any updates, the server returns <NewUpDt>, also a timestamp, that is the time that the last update was made on the server. The client may use <NewUpDt> as the value of <UpDt> in a subsequent <xxxInqRq>.*

Tag	Type	Usage	Description
<StartDt>	DateTime	Required OR	Selection Start Date.
<EndDt>	DateTime	Required OR	Selection End Date.

3.2.10.2 Selection Range Currency Amount Aggregate <SelRangeCurAmt>

The <SelRangeCurAmt> aggregate contains two optional fields: <LowCurAmt> and <HighCurAmt>. The following rules are established for these fields:

- If <LowCurAmt> but not <HighCurAmt> is supplied, then the search must match amounts that are greater than or equal to <LowCurAmt>.
- If <HighCurAmt> is supplied but not <LowCurAmt>, then the search must match amounts that are smaller than or equal to the <HighCurAmt>.
- If both amounts are supplied, then the search must match amounts that are between or equal to the two amounts supplied.
- If a client wishes to search for a specific amount, then both <LowCurAmt> and <HighCurAmt> should be supplied and set to the same value.
- Both amounts must be of the same currency.

Tag	Type	Usage	Description
<LowCurAmt>	Currency Amount	Required OR	Selection Low Amount.
<HighCurAmt>	Currency Amount	Required OR	Selection End Amount.

3.2.11   Recurring Model Data

3.2.11.1 Recurring Model Information Aggregate <RecModelInfo>

Tag	Type	Usage	Description
<Freq>	Open Enum	Required	Recurring Model Frequency. Defined values: Daily, Weekly, Biweekly, TwiceMonthly, Monthly, EndOfMonth, FourWeeks, BiMonthly, Quarterly, SemiAnnually, Annually, Manually.
<NumInsts>	Long	Optional XOR Profiled support	Recurring Model Total Instances. If none of <NumInsts>, <FinalPrcDt>, or <FinalDueDt> is provided, the model is open-ended.
<FinalPrcDt>	Date	Optional XOR Profiled support	Final Processing Date. Subject to server support for Processing Date Model in profile. Only payments allow the option of processing date. All transfers must use due date. If none of <NumInsts>, <FinalPrcDt>, or <FinalDueDt> is provided, the model is open-ended.
<FinalDueDt>	Date	Optional XOR Profiled support	Final Due Date. Subject to server support for Due Date Model in profile. If none of <NumInsts>, <FinalPrcDt>, or <FinalDueDt> is provided, the model is open-ended.
<InitialCurAmt>	Currency Amount	Optional Profiled support	Recurring Model Initial Amount. May be different from nominal instance amount. If omitted, the default instance amount is used for the initial transfer or payment.
<FinalCurAmt>	Currency Amount	Optional Profiled support	Recurring Model Final Amount. May be different from nominal instance amount. If omitted, the default instance amount is used for the final transfer or payment.

Tag	Type	Usage	Description
<Nickname>	C-40	Optional Profiled support	Nickname. Optionally assigned by the customer. Pay provider may indicate support for longer-term storage of nicknames in the Service Profile. May be modified by the client.
<SkipNextN>	Long	Optional Profiled support	Skip Next *N* Instances. A non-zero value indicates that the next *N* transaction instances are to be skipped. Subject to server support in Service Profile.
			Note: *The server must decrement the Skip Next N count and generate the instance with a processing status code of Skipped, based on the timing of the recurring model. The server must handle the skipped instance like any other instance, except that the server does not execute the instance for fulfillment purposes, including creating audit and sync records for the instance addition, and for the recurring model status changes. Therefore, each decrement must be reflected as a separate <RecXferModRs> within audits and syncs, and each status change to the specific instances of the transfers must be reflected as a separate <XferModRs>.*
			The same behavior applies to <RecChkOrdModRs> and <ChkOrdModRs>

3.2.12   System Message Data

3.2.12.1   Response Status Aggregate <Status>

The <Status> aggregate is used in each response to indicate the status of the message. See Appendix A for information on Response Status Codes.

When <Severity> = Info or Warn and the server is responding synchronously (i.e., not requiring the client to pick up the response at a later time, as described below), the server must generate the complete response message. When <Severity> = Error, the server is only expected to include the <Status> aggregate (required), the <MsgRsHdr> aggregate (optional), and echo the <RqUID>, <AsyncRqUID>, and <CustId> fields. The server must include these fields unless a system error prevents the server from reading the request message contents. In the latter situation, it may only be possible for the server to generate a response message containing the <Status> aggregate by itself. Other actions to be taken in the case of severe errors are specified for each implementation.

> *Note: The explanatory text provided in the <StatusDesc> is freeform text that is not strictly dictated by the status code. Appendix A provides default <StatusDesc> text, but this text may be replaced or enhanced by particular implementations.*

The Asynchronous Response Information Aggregate <AsyncRsInfo> is used when the server cannot generate a complete response in a reasonable time (such as less than five minutes). In such a case, the server may return a <StatusCode> of 900 and <Severity> of Warn, with an <AsyncRsInfo> aggregate optionally containing an Available Date <AvailDt> and/or Expiration Date <ExpDt>. These two elements indicate to the client the window of time at which the response data is expected to be available. If the server does provide an asynchronous response, the client must send the *exact same request, but with a new <RqUID>, providing the <RqUID> from the original request in the <AsyncRqUID> element,* at a later time (generally within the availability window). The server then must respond with either the cached and built response, or another <AsyncRsInfo> indicating a new projected availability time.

The server may choose to reject, with a <StatusCode> of 910 and <Severity> of Error, a subsequent asynchronous request if the elements provided by the client don't match the corresponding elements from the original request.

Info status means that this status did not prevent the server from accepting message, service, or document. The status provided additional information related to the message, service, or document.

Warn status means that this status did not prevent the server from accepting the message, service, or document; however, not all information may have been returned to the client. For example, too many records requested.

Error status means that the server declined the message, service, or document. The server may return a SPRefId and/or a CSPRefId, where present, for tracking purposes; however, the return of these elements are not guaranteed.

Tag	Type	Usage	Description
<StatusCode>	Long	Required	Response Status Code. Valid values depend on context. See Appendix A.
<ServerStatusCode>	C-20	Optional	Server Status Code. The value placed here is used to allow the client to display the status code to the user. This allows the user to read the code to a customer service representative for debugging purposes.
<Severity>	Closed Enum	Required	Severity. Valid values: Error, Warn, Info
<StatusDesc>	C-255	Optional	Status Description. Explanatory text associated with the status code. The Status Description may be default text or Service Provider specific.
<AdditionalStatus>	Aggregate	Optional Repeating	Additional statuses. The <StatusCode> above must contain the primary response code. This aggregate may appear once for each additional status that the server intends to provide. For example, if both an account number and a date are invalid, the <StatusCode> above may contain one of the two errors, and this aggregate may contain the other.
<StatusCode>	Long	Optional	Response Status Code. Valid values depend on context. See Appendix A.
<ServerStatusCode>	C-20	Optional	Server Status Code. The value placed here is used to allow the client to display the status code to the user. This allows the user to read the code to a customer service representative for debugging purposes.
<Severity>	Closed Enum	Required	Severity. Valid values: Error, Warn, Info
<StatusDesc>	C-255	Optional	Status Description. Explanatory text associated with the status code. The Status Description may be default text or Service Provider specific.
</AdditionalStatus>			
<AsyncRsInfo>	Aggregate	Optional	Asynchronous Response Information.
<AvailDt>	DateTime	Optional	Available DateTime. The projected time at which the response will be available.
<ExpDt>	DateTime	Optional	Expiration DateTime. The time at which the cached response will be purged at the server.
</AsyncRsInfo>			

3.2.12.2 Records Control

Some response messages, typically associated with the inquiry/audit/sync request messages, may be quite large. The size of the response message may create problems for both the client and server. The problems are specific to the implementation. Some examples of these problems are:

- The client may not be able to allocate enough storage for the entire response.
- A network timeout may occur due to the time to return the complete HTTP response.

- Mixing very large with small messages within the message manager (OLTP) middleware may create serious performance problems for the small messages.

In order to address these potential problems, IFX allows either the client or server to constrain the size of a response message. The client initiates all IFX messages, so the server cannot "push" the additional information to the client. A mechanism is provided for the server to indicate that additional information is available and for the client to request the additional information

The Record Control value within <OptSupt> in the Service Profile indicates whether a server supports this capability.

3.2.12.2.1 Records Control Input <RecCtrlIn>

The <RecCtrlIn> aggregate is used in inquiry/audit/sync request messages to allow the client to specify a maximum number of records that it is capable of processing within a single response. The initial request omits the <Cursor> element.

Subsequent requests set the <Cursor> to the value returned by the server in the previous response (Rs) message. The <Cursor> value is meaningful to the server and opaque to the client. The <Cursor> is intended to be a pointer to a position within the answer set for the inquiry/audit/sync request.

The client should send the same selection criteria in the initial and subsequent requests. The server may verify that the cursor returned is valid by comparing the search criteria on subsequent requests with the search criteria on the initial request. If they differ, the cursor may be considered invalid and an error returned.

Similarly, if the time between the initial and subsequent requests is too long, the cursor may be rejected on the subsequent request. In this situation, the client may reinitiate the initial request.

Tag	Type	Usage	Description
<MaxRec>	Long	Required	Maximum Records Requested.
<Cursor>	Binary	Optional *but see Description*	Cursor Handle. Must be omitted in initial request. May be included in subsequent requests if server provided as a pointer to more records in previous <RecCtrlOut> in response.

3.2.12.2.2 Records Control Output <RecCtrlOut>

The <RecCtrlOut> aggregate is used in inquiry/audit/sync responses to provide a client that used the <RecCtrlIn> aggregate in the request with information it needs to retrieve the rest of the records that matched the specified selection criteria.

The <RecCtrlOut> aggregate must be sent as part of the Rs if the server indicated support for Records Control in the Service Profile by setting <OptSupt> = RecCtrl and one or both of the following are true:

- The Rq contains an <RecCtrlIn> aggregate or
- The server is returning a subset of the available output records in an Rs.

Tag	Type	Usage	Description
<MatchedRec>	Long	Required	Matched Records. Total number of records matching the selection criteria.
<SentRec>	Long	Required	Sent Records. Number of records matching the selection criteria that are included in this message.
<Cursor>	Binary	Optional *but see Description*	Cursor. Included in the response only if additional records are available. When <Cursor> is present in the response, the Status code 1110 must be sent.
			Assume server retains cursor of matching records for some time in case client requests more.
			Used to allow the client to issue another request to fetch more matching records.

3.2.12.2.3 Example

Assume:

- The server sets <OptSupt>=RecCtrl in Banking section of Service Profile.
- The client issues an Account History message.
- The client is capable of processing only process 10 records (history messages) within a single Rs.
- There are 25 messages in the answer set.

Initial request from client:

```
<MaxRec>=10
```
Note that the <Cursor> element is omitted.

Initial response:

```
<MatchedRec>=25
<SentRec>=10
<Cursor>=binary_value_1
```
Second request:

```
<MaxRec>=10
<Cursor>=binary_value_1
```
Second response:

```
<MatchedRec>=25
<SentRec>=10
<Cursor>=binary_value_2
```
Third request:

```
<MaxRec>=10
<Cursor>=binary_value_2
```
Third response:

```
<MatchedRec>=25
<SentRec>=5
```
<Cursor> element is omitted

3.2.12.3 Client Application <ClientApp> and Proxy Client <ProxyClient>

The primary use of the <ClientApp> aggregate is to inform a customer service representative about what software was used to create or change an object. <ClientApp> conveys the application that is acting as the client endpoint of the IFX message. This may be an application on the customer's PC if it is aware of Interactive Financial Exchange, or it may be an intermediate server that is composing IFX requests on behalf of a client that has connected to it using some other protocol (e.g., HTTP).

Tag	Type	Usage	Description
<Org>	Identifier	Required	Organization. This identifies the client application manufacturer.
<Name>	C-40	Required	Client Application Name.
<Version>	NC-12	Required	Client Application Version.

3.2.13 Industry Identifier Aggregate <IndustId>

The <IndustId> aggregate provides standard codes by which industries are classified. <IndustId> is used in payments and other messages as information for reporting and accounting by industry code.

Tag	Type	Usage	Description

Tag	Type	Usage	Description
<Org>	Identifier	Required	Organization. Identifies the organization assigning numbers to different industries.
			"SIC" identifies Standard Industrial Codes. "NAICS" identifies North American Industrial Code System.
<IndustNum>	NC-6	Required	Number identifying the industry.

3.2.14 Secrets Aggregates

3.2.14.1 Secret Prompt Aggregate <SecretPrompt>

The <SecretPrompt> aggregate is used to provide enough information to prompt a user for secrets used for initial authentication. For example, the server may request the user's tax ID, mother's maiden name or some other secret already shared between the user and the service provider operating the server.

The <SecretPrompt> may be used by CSPs during enrollment of customers for a specific service such as Bill Presentment. For example, a customer new to a specific biller may be authenticated against existing customer records at the biller. If omitted, the biller does not require the customer to enter any secrets for client enrollment.

Another use is to prompt for a security token or password that has been provided to the customer by some other means, possibly during an interactive identity verification session with the customer.

Tag	Type	Usage	Description
<SecretId>	Identifier	Required	The ID of this secret. The client must return the ID along with the secret itself in <SecretList>.
<Prompt>	C-20	Required	Prompt. The short prompt to be displayed to the user, explaining the value to be returned in <SecretList> for this secret.
<Memo>	C-255	Optional	Memo. This is a longer description of the meaning of <Prompt> above. This may be displayed to the user if the user requests more information regarding the intended meaning of <Prompt>.
<SecretOptional>	Boolean	Optional	If *True*, the secret *does not* need to be provided in the <SecretList> sent to the server in order for the server to process the requested message. If *False* or omitted, the secret *does* need to be provided in the <SecretList> sent to the server in order for the server to process the requested message.
<SecretFormat>	NC-1024	Optional	Secret Format.
			Regular expression describing the secret format. The definition and behavior of "Regular Expression" is per IEEE Std 1003.2-1992 (POSIX.2). General definition may be found at http://www.ciser.cornell.edu/info/regex.html
<SecretMask>	C-80	Optional	Secret Mask. Edit Mask for the user input for <Secret> in <SecretList>. EditMask, in Section 3.1.1,specifies the format for <SecretMask>.

3.2.14.2 Secret List Aggregate <SecretList>

The <SecretList> aggregate provides a way for users to input shared secrets in response to a <SecretPrompt>.

Tag	Type	Usage	Description
<SecretId>	Identifier	Required	ID of Secret. The client must use the value of <SecretId> sent by the server in <SecretPrompt>.

Tag	Type	Usage	Description
<CryptType>	Open Enum	Required Profiled values	Encryption Type.
			Defined values: None, PKCS#1 , PKCS#7, PKCS#10, ECB, CBC, CFB, OFB, TECB, TCBC, TCBC-I, TCFB, TCFB-P, TOFB, TOFB-I, AECB, ACBC, ACFB, AOFB, ACM
			Must be supported in list of encryption types in the SP's Service Profile.
<Secret>	C-80	Required XOR	Secret. The secret itself as entered by the user in response to the <Prompt> and/or <Memo> displayed in <SecretPrompt>.
			<Secret> is used when <CryptType> = None
<CryptSecret>	Binary, 128	Required XOR	Encrypted Secret. Positional list of customer-entered data corresponding to each secret prompt provided through the <BillerRec> aggregate. Typical uses are mother's maiden name, tax id, etc.
			<CryptSecret> is used when <CryptType> ≠ None

3.2.15   Payment Instrument Aggregate <PmtInst>

The <PmtInst> aggregate is used to communicate the types of payment that the biller is capable of accepting via the Pay provider. This is used to restrict the methods by which the customer may pay for the bill. The Service Provider indicates which Payment Instruments are supported via the Biller profile in <BillerPayInfo>. The Biller may use the <PmtInst> in <BillRec> to provide a subset of the types of payments the Biller accepts for a particular bill. Note that errors may occur if this aggregate contains types of payments different from those communicated through the Biller Profile.

Tag	Type	Usage	Description
<PmtInstType>	Closed Enum	Required	Payment type. Specifies the type of payment instruments that the biller is capable of accepting for electronic payment.
			Valid values: CheckAcct, CreditCard, ElectronicCash
<Brand>	Open Enum	Optional	Accepted brand for a given payment type. If not specified, the client assumes that all brands of the given <PmtInstType> are acceptable.
			Defined values: AmEx, CarteBlanche, DinersClub, Discover, Europay, JCB, MasterCard.
<SettlementInfo>	Aggregate *see section 3.2.15.1*	Optional Repeating	Settlement Information. May contain complete or partial information for use by the CPP in transferring funds to the Biller or BPP. One or more options may be supported for each Payment instrument and brand.

3.2.15.1   Settlement Information <SettlementInfo>

The <SettlementInfo> aggregate is used to communicate the settlement instruction of payment to the CPP. This aggregate may contain complete or partial information for use by the CPP in transferring funds to the Biller or BPP.

Tag	Type	Usage	Description

Tag	Type	Usage	Description
\<SettlementMethod>	Open Enum	Required	Settlement Method. Either the method accepted by the Biller or BPP for settling payments, or the payment/settlement method, provided as the general method for settling payment.
			Defined values: RPS, EPay, ACH, Concentrator, FedNet, SWIFT, CHIPS, CHAPS, BookEntry, Draft, OutsourcedCheck, DirDebACH
\<DepAcctId>	Aggregate *see section 3.2.7.1.2*	Required OR	Deposit Account Identification Aggregate.
\<SettlementId>	C-20	Required OR	Settlement Identification. May contain the RPS or Epay ID as applicable, depending on the Settlement Type.
\<OrgContact>	Aggregate *see section 3.2.5.2*	Required OR	Organization Contact. The Biller's Concentrator's contact name and telephone number for the CPP to find out additional information needed to complete payments or for clarification.
\<ContactInfo>	Aggregate *see section 3.2.3.7*	Required OR	Contact Information Aggregate.
\<Memo>	C-255	Required OR	Additional human-readable information that may be needed to complete payments or for clarification.
\<PmtInstruction>	Aggregate	Optional	Detail payment and settlement instruction.
\<Country>	NC-3	Optional	Country. Country of the payment system in which the payment is processed.
\<PmtFormat>	Open Enum	Optional	Payment format. Payment format for the specified payment method above. For example, in ACH electronic transfer, the format can be CTX, CCD, CCD+, PPD, etc.
\<RefInfo>	Aggregate *see section 3.2.17*	Optional	Reference information associated with payment information.
\<IntermediaryDepAcct>	Aggregate *see section 3.2.7.1.2.1*	Optional Repeating	Intermediary deposit account. Used to identify deposit accounts for intermediary banks that route the payment.
\<FeeChargeAlloc>	Aggregate	Optional	Bank Fee Charge Allocation. Used to communicate the fee charges on the payment by the CPP. This specifies who and how the fee is being distributed between the Payer and the Payee.
\<ChargeRegulation>	Open Enum	Required	Specify how the fee is distributed.
			Defined Values: Payer, Payee, Share Equally.
\<Fee>	Aggregate *see section 3.2.16.2*	Required	Fee Charge info and amount.
\<DepAcctId>	Aggregate *see section 3.2.7.1.2*	Optional	Account from which the bank fee is drawn.
\</FeeChargeAlloc>			
\</PmtInstruction>			

3.2.16   Fees and Amounts

Up to and including IFX 1.1, a fee was only allowed to be a currency amount. This is adequate for fees that are a fixed amount (e.g. surcharge fee of $1.50). For some transactions the associated fee is based on a percentage

of the transaction amount or a combination of a fixed amount and a percentage. The following are some fee scenarios:

- fee is a fixed currency amount (e.g. surcharge fee of $1.50)
- fee is a percentage of the transaction amount (e.g. check cashing fee of 1% of the check amount)
- fee is a percentage of the transaction amount with a minimum (e.g. check cashing fee of 1% of the check amount, with a minimum fee of $2.00)
- fee is a percentage of the transaction amount with a maximum (e.g. check cashing fee of 1% of the check amount, with a maximum fee of $5.00)
- fee is a percentage of the transaction amount with a minimum and maximum (e.g. check cashing fee of 1% of the check amount, with a minimum fee of $2.00 and maximum fee of $5.00)
- fee is a fixed currency amount plus a percentage of the transaction amount (e.g. a $1.00 fee to cash a check plus 2% of the check amount)

IFX 1.1 and earlier only allowed the first scenario. Starting with IFX 1.2, fees are permitted to be any of the five. Following are descriptions of which fields are necessary to describe each of those scenarios:

- fee is a fixed currency amount (e.g. surcharge fee of $1.50)

<CurAmt> = 1.50

- fee is a percentage of the transaction amount (e.g. check cashing fee of 1% of the check amount)

<Rate> = 1.00

- fee is a percentage of the transaction amount with a minimum (e.g. check cashing fee of 1% of the check amount, with a minimum fee of $2.00)

<Rate> = 1.00
<MinCurAmt> = 2.00

- fee is a percentage of the transaction amount with a maximum (e.g. check cashing fee of 1% of the check amount, with a maximum fee of $5.00)

<Rate> = 1.00
<MaxCurAmt> = 5.00

- fee is a percentage of the transaction amount with a minimum and maximum (e.g. check cashing fee of 1% of the check amount, with a minimum fee of $2.00 and maximum fee of $5.00)

<Rate> = 1.00
<MinCurAmt> = 2.00
<MaxCurAmt> = 5.00

- fee is a fixed currency amount plus a percentage of the transaction amount (e.g. a $1.00 fee to cash a check plus 2% of the check amount)

<CurAmt> = 1.00
<Rate> = 2.00

3.2.16.1 Composite Currency Amount <CompositeCurAmt>

The Composite Currency Amount aggregate <CompositeCurAmt> contains information about the amount and timing of float availability of funds for a single transaction. This aggregate is also used (usually repeating) where a currency amount identified in a message consists of multiple composites, e.g. a debit message which identifies both the customer requested debit amount plus one or more fees to be debited. The composite currency amount aggregate can also be used within a <DebitAdd> to charge a fee (e.g. for a printed statement). The <CompositeCurAmtId> is used to identify a specific component of a currency amount in communications between client and server (e.g. to override a specific Fee component).

Tag	Type	Usage	Description

Tag	Type	Usage	Description
<CompositeCurAmtId>	Identifier	Optional *but see Description*	Composite Currency Amount Id. Required if the aggregate is repeating within a debit or credit message.
<CompositeCurAmtType>	Open Enum	Required	Composite Currency Amount Type. Description of Float Category or Currency Amount Type. Defined Values: Immediate, 1DayFloat, 2DayFloat, 3DayFloat, 4DayFloat, 5DayFloat, 6DayFloat, OnePlusDay, TwoPlusDay, ThreePlusDay, Standard, Debit, Credit, ForExFee, StopChkFee, LateFee, TransactionFee, InterchangeFee, Surcharge, StatementFee, Tax, MerchandisePurchase, MerchandiseReturn.
<CurAmt>	Currency Amount	Required OR	Currency Amount. When used for float, the fractional amount of the transaction amount assigned to this float category. When used for a fixed amount fee, this is the fixed amount that will be charged. For fees that are calculated as a fixed amount plus a percentage of the transaction, this is the fixed amount and the percentage is in Rate.
<Rate>	Decimal	Required OR 1.2+	Fee Rate. Usage is a percentage (e.g. a value of 2.5 = 2.5%).
<MinCurAmt>	Currency Amount	Optional	Minimum Currency Amount. The minimum amount to charge for this fee.
<MaxCurAmt>	Currency Amount	Optional	Maximum Currency Amount. The maximum amount to charge for this fee.
<SpecialHandling>	Open Enum	Optional	Special Handling requests. Defined Values: ClientOverride
<Memo>	C-255	Optional	Memo. Additional information about the transaction (e.g. reason for modifying the fees)

3.2.16.2 Fee <Fee>

The <Fee> aggregate identifies the type and amount of fee being charged.

Tag	Type	Usage	Description
<FeeType>	Open Enum	Required	Fee Type. Defined values: ForEx, StopChk, Late, Transaction, Interchange, Surcharge
<CurAmt>	Currency Amount	Required OR	Currency Amount. When used for a fixed amount fee, this is the fixed amount that will be charged. For fees that are calculated as a fixed amount plus a percentage of the transaction, this is the fixed amount and the percentage is in Rate.
<Rate>	Decimal	Required OR	Fee Rate. Usage is a percentage (e.g. a value of 2.5 = 2.5%).
<MinCurAmt>	Currency Amount	Optional	Minimum Currency Amount. The minimum amount to charge for this fee.
<MaxCurAmt>	Currency Amount	Optional	Maximum Currency Amount. The maximum amount to charge for this fee.

3.2.16.3 Tax Information <TaxInfo>

The <TaxInfo> aggregate identifies a levied tax.

Tag	Type	Usage	Description
\<Org\>	Identifier	Required	Organization. Organization defining this name space. Usage is Tax Authority (e.g., state or country).
\<TaxType\>	Open Enum	Required	Tax Type. Qualified by \<Org\>. Defined values: WithHoldingTax, DebitsTax, FIDuty
\<CurAmt\>	Currency Amount	Required OR	Tax Amount (+ paid by customer, – earned by customer).
\<Rate\>	Decimal	Required OR	Tax Rate. Usage is a percentage (e.g., a value of 5.2 = 5.2%). (+ paid by customer, – earned by customer)

3.2.17   Reference Information \<RefInfo\>

The \<RefInfo\> aggregate provides information typically related to an IFX object. It is generally expected that the reference information is to be used by the message recipient to relate information within the associated IFX object with other data obtained via a different channel.

Tag	Type	Usage	Description
\<RefType\>	Open Enum	Required	Reference Type. Description of the type of Reference Id. Defined Values depend on the specific usage.
\<RefId\>	Identifier	Required	Reference Identifier.

3.2.18   Network Transaction Information Aggregate \<NetworkTrnInfo\>

The Network Transaction Information aggregate \<NetworkTrnInfo\> contains information regarding the network processing the transaction, i.e., owner, location, bank ID and reference number assigned by the network when processing the transaction. Information in this aggregate is also used for providing the physical location of an automated terminal at which a customer conducts a transaction. The physical location information may be required to be printed on statements and receipts for regulatory purposes.

> *Note: This aggregate is used in IFX 1.1 and later versions as a replacement for the \<ATMTrnInfo\> and \<USA.ACHTrnInfo\> aggregates within the \<DepAcctTrnRec\> aggregate. The \<ATMTrnInfo\> and \<USA.ACHTrnInfo\> aggregates will be deprecated in IFX 2.0.*

Tag	Type	Usage	Description
\<NetworkOwner\>	Open Enum	Required	Network Owner. The name of the individual or organization that owns the source network. Defined Values: ATM, ACH, FedNet, SWIFT
\<OriginatorName\>	C-40	Optional	Name. Name of originator of the Message.
\<TerminalId\>	Identifier	Optional *but see Description*	Terminal Identifier. Identification of terminal, such as terminal code or terminal number of ATM. Required if \<NetworkOwner\> is ATM.
\<PostAddr\>	Aggregate *see section 3.2.3.7.1*	Optional *but see Description* XOR	Postal Address. Physical Address of the terminal. Required if \<NetworkOwner\> is ATM.
\<Desc\>	C-80	Optional *but see Description* XOR	Description. A generally accepted name for the location of the terminal. Required if \<NetworkOwner\> is ATM.

Tag	Type	Usage	Description
<Name>	C-40	Optional *but see Description* XOR	Name. Name of the owner or operator of the terminal. Required if <NetworkOwner> is ATM.
<BankIdType>	Open Enum	Optional	Bank Identifier Type. Type of bank identifier. Defined Values: SWIFT, ABA, FedWire, CHIP, CHAP, SortCode. Default value is ABA.
<BankId>	NC-34	Optional *but see Description*	Bank Identifier. Qualifies account number if known by the customer/client. Usage is expected to be routing and transit number in the US or the equivalent in an international implementation. This is required in USA when <NetworkOwner> is ACH.
<NetworkRefId>	Identifier	Optional	Network Reference Identifier. Reference number assigned by network due to the processing of a message at the respective network. Example: Federal Reserve wire network assigns a Fed Reference Number or ACH trace number.

4 Security

4.1 Channel Level Security vs. Application Level Security

The IFX Specification is designed to provide messages and data elements necessary to provide Application Level Security on top of an existing secure communications channel between the client and server. This Channel Level Security is not specified in this document, which is intentionally transport-independent. The mechanism used to provide this secure communications channel will be specified in each of the IFX Implementation Specifications associated with this document.

4.2 Application Level Security in the IFX Specification

The IFX Specification supports a Customer Login ID and Password for each Signon Realm that the customer must access for their services. Each Signon Realm may have different rules regarding Customer Password length and composition; the rules may be found in the <SignonInfo> aggregate returned to the client in the <*xxx*SvcProfInfo>. Each CSP may also independently set its requirements for Customer Password Encryption.

> *Note:* *At present, The IFX Specification supports several values for encryption of passwords and other authentication information. The first defined value is NONE, which relies entirely on channel level security and sends passwords as clear text. The other defined values enable a password or similar authentication information (up to 117 bytes) to be encrypted by the client or server and passed as a 128-byte binary element that may be decrypted by the receiver. Encryption keys can be generated with a 1024-bit modulus and distributed in an "out of band" process. Alternatively, with the use of the Security Object, encryption keys can be generated and distributed in an "in-band" process.*

Once a client authenticates with a CSP, the client may perform messages for any of the Services enabled by the CSP. If the client wishes to perform a message from a Service that has not been enabled for the current CSP, the client must authenticate with the CSP for which the desired Service has been enabled.

4.3 Security Common Elements and Aggregates

4.3.1 Customer Password Aggregate <CustPswd>

Tag	Type	Usage	Description
<CryptType>	Open Enum	Required Profiled values	Encryption Type. Defined values: None, PKCS#1. , PKCS#7, PKCS#10, ECB, CBC, CFB, OFB, TECB, TCBC, TCBC-I, TCFB, TCFB-P, TOFB, TOFB-I, AECB, ACBC, ACFB, AOFB, ACM Must be supported in list of encryption types in <SignonInfo> in the CSP's Service Profile.
<SecObjId>	Identifier	Optional *but see Description*	Identifier of Security Object on which the <CryptPswd> is encrypted Note: If security objects are managed in band, this element should be included in order to know how to decrypt <CryptPswd>
<Pswd>	C-32	Required XOR	Clear text Password. Used only when <CryptType> = None

<CryptPswd>	Binary, 128	Required XOR	Encrypted Password. Used when <CryptType> ≠ None.

4.3.2   Signon Password Aggregate <SignonPswd>

Tag	*Type*	*Usage*	*Description*
<SignonRole>	Open Enum	Optional	Signon role. Defined values: Customer, CSR, Agent Default value is Customer
<CustId>	Aggregate *see section 3.2.3.1*	Required	Customer Identification Aggregate. This is the identifier of the user signing on.
<CustPswd>	Aggregate *see section 4.3.1*	Required	Customer Password Aggregate.
<GenSessKey>	Boolean	Optional	Session Key Requested Indicator. *True* indicates that the client is requesting a Session Key in the response. *False* or absent indicates that the client is not requesting a Session Key in the response.

4.3.3   Signon Certificate Aggregate <SignonCert>

Tag	*Type*	*Usage*	*Description*
<SignonRole>	Open Enum	Optional	Signon role. Defined values: Customer, CSR, Agent Default value is Customer
<CustId>	Aggregate *see section 3.2.3.1*	Required	Customer Identification Aggregate. This is the identifier of the user signing on.
<Certificate>	Binary	Required	
<GenSessKey>	Boolean	Optional	Session Key Requested Indicator. *True* indicates that the client is requesting a Session Key in the response. *False* or absent indicates that the client is not requesting a Session Key in the response.

4.3.4   Signon Transport Aggregate <SignonTransport>

Tag	*Type*	*Usage*	*Description*
<SignonRole>	Open Enum	Optional	Signon role. Defined values: Customer, CSR, Agent Default value is Customer
<CustId>	Aggregate *see section 3.2.3.1*	Required	Customer Identification Aggregate. This is the identifier of the user signing on.
<GenSessKey>	Boolean	Optional	Session Key Requested Indicator. *True* indicates that the client is requesting a Session Key in the response. *False* or absent indicates that the client is not requesting a Session Key in the response.

4.3.5   Signon Magnetic Stripe/PIN Aggregate <SignonMagPIN>

Tag	Type	Usage	Description
<CardMagData>	Aggregate *see section 3.2.7.1.3.1*	Required	Card Magnetic Stripe Data
<CryptType>	Open Enum	Optional *but see desciption*	Encryption Type. <CryptType> will be required in IFX 2.0.
			Defined values: None, PKCS#1, PKCS#7, PKCS#10, ECB, CBC, CFB, OFB, TECB, TCBC, TCBC-I, TCFB, TCFB-P, TOFB, TOFB-I, AECB, ACBC, ACFB, AOFB, ACM
			Must be supported in list of encryption types in <SignonInfo> in the CSP's Service Profile.
			Note: If security objects are managed in band, this element should be included in order to know how to decrypt <PINBlock>
<PINBlock>	NC-32	Required	PIN pad data
<SecObjId>	Identifier	Optional Repeating *but see Description*	Identifier of Security Object on which the <PINBlock> is encrypted. <SecObjId> will be required in IFX 2.0
			Note: If security objects are managed in band, this element should be included in order to know how to decrypt <PINBlock>
<GenSessKey>	Boolean	Optional	Session Key Requested Indicator. *True* indicates that the client is requesting a Session Key in the response. *False* or absent indicates that the client is not requesting a Session Key in the response.

4.3.6   Session Key <SessKey>

As an option, a client may request that the server generate and return a Session Key <SessKey> during a <SignonRq>. A Session Key may have an associated Expiration DateTime, after which it may not be used. If a client has a Session Key that has not expired, it may use the <SessKey> in subsequent <SignonRq>s.

This mechanism addresses several requirements:

- It allows a server to efficiently process multiple <SignonRq>s from the same client over a relatively short time period. A batch-oriented client may group messages into a number of batches so that the results of the preceding batch are known when the next is submitted. The <SignonRq> in the first batch must contain a <CustId> and/or any other required authentication information. The <SignonRq> within subsequent batches may then contain the <SessKey> returned by the first <SignonRs>, in lieu of other authentication mechanisms.

- It allows a server to efficiently process multiple <SignonRq>s for the same user, but not necessarily the same client, over a short period of time. For example, a PFM client may invoke a web browser to support a CSP function that is not directly supported by the PFM client. The PFM client may use the <SessKey> obtained from the <SignonRs> to issue a <SignonRq> for the web browser session.

- A batch or interactive client may process multiple Services with different Service Providers; e.g., VRU sends Pay messages to a different SP than Banking. The CSP and other SPs have a shared authentication mechanism. The client may use the <SessKey> obtained from the <SignonRs> from the CSP as part of the <SignonRq> to the other SPs, rather than having to either store the user password or prompt the user to enter his/her password a second time.

A Session Key does *not* include the credentials of the entity specified in the <SignonOverride> aggregate.

Tag	Type	Usage	Description

Tag	Type	Usage	Description
<SessKey>	NC-64		Session Key.

4.4 Authentication

4.4.1 Description

Customer Authentication Messages are used to authenticate customers and address security at the application level.

The Authentication messages are the only IFX messages without <RqUID>s, as there may be only one per IFX document.

4.4.2 Signon

The Signon message is used to authenticate an IFX client. Unlike other IFX messages, the <SignonRs> does not echo all elements of the request, for security reasons. Also, because there must be only one <SignonRq> per IFX block, there is no <RqUID> in either the request or the response.

If none of the optional signon methods are provided, the signon is anonymous.

Customer authentication is between the IFX client and the IFX server. Other providers behind the IFX server are expected to rely on the authentication performed by the initial IFX server (note this does not apply if the client is redirected, or handed off, to another CSP). Intermediate elements, such as HTTP proxies, do not participate in the customer authentication exchange. The IFX server may rely on a different server to actually verify the customer password, but this is a local matter.

> *Note: Signon and Signoff are the only two messages to occur at the top level of IFX, rather than being contained within a service wrapper.*

4.4.2.1 Request <SignonRq>

Tag	Type	Usage	Description
<SignonPswd>	Aggregate *see section 4.3.2*	Optional XOR	Signon with <CustId>/<CustPswd> authentication method.
<SignonCert>	Aggregate *see section 4.3.3*	Optional XOR	Signon with embedded certificate.
<SignonTransport>	Aggregate *see section 4.3.4*	Optional XOR	Signon with transport certificate (i.e., SSL) authentication method.
<SignonMagPIN>	Aggregate *see section 4.3.5*	Optional XOR	Signon with a magnetic stripe card and a PIN pad
<SessKey>	NC-64	Optional XOR	Authentication Key. Issued previously based on Password. Valid only for a fixed interval, as set by the server.
<SignonOverride>	Aggregate	Optional	Authentication used for overriding functions that could not normally be performed by the signed on user.
<OverrideType>	Open Enum	Required	Type of override being performed. Defined values: Teller, Supervisor
<SignonPswd>	Aggregate *see section 4.3.2*	Required XOR	Signon with <CustId>/<CustPswd> authentication method.

Tag	Type	Usage	Description
<SignonCert>	Aggregate *see section 4.3.3*	Required XOR	Signon with embedded certificate.
<SignonMagPIN>	Aggregate *see section 4.3.5*	Required XOR	Signon with a magnetic stripe card and a PIN block.
</SignonOverride>			
<ClientDt>	DateTime	Required	Client DateTime. Time according to the client.
			Note: *This is typically used by customer service to resolve problems regarding cut-off time or timeliness of customer messages. It may be compared with the server time to determine whether there is a discrepancy.*
<CustLangPref>	NC-17	Required Profiled values	The Client Language Preference has the same valid values as all other language elements.
			See section 2.6.4 for more information on the format of this element.
			Must be one of those supported in the Service Profile.
<ClientApp>	Aggregate *see section 3.2.12.3*	Required	Client Application Aggregate. Manufacture, name, and version of the client endpoint of the IFX message.
<ProxyClient>	Aggregate *see section 3.2.12.3*	Optional	Proxy Client Aggregate. In the case where the IFX client is composing requests for some other application that the user is running (e.g., a Web browser), the details of the customer's interface application should be included here. It has the same structure as the <ClientApp> aggregate.
<EU.Cur>	Closed Enum	Optional	Euro currency selection. Used only in countries supporting the euro currency.
			Valid Values: Local, Euro.
<SuppressEcho>	Boolean	Optional Profiled support	If True, the client requests echoed fields to *not* appear in responses. If False or absent, the client requests echoed fields to appear in responses.
			The SP may choose to not honor the request and therefore send echoed fields in responses.

4.4.2.2   Response <SignonRs>

Tag	Type	Usage	Description
<Status>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, <StatusCode> defaults to 0 (zero).
<SignonRole>	Open Enum	Optional Echoed	Signon role.
<CustId>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
			Note: *In cases where <Status> <Severity> is* Error, *echoing <CustId> is optional.*

Tag	Type	Usage	Description
\<GenSessKey\>	Boolean	Optional Echoed	User Key Requested Indicator.
\<ClientDt\>	DateTime	Required Echoed	Client DateTime.
\<CustLangPref\>	NC-17	Required Echoed	Customer Language Preference Aggregate.
\<ClientApp\>	Aggregate *see section 3.2.12.3*	Required Echoed	Client Application Aggregate.
\<ProxyClient\>	Aggregate *see section 3.2.12.3*	Optional Echoed	Proxy Client Aggregate.
\<EU.Cur\>	Closed Enum	Optional Echoed	Euro Currency Selection.
\<ServerDt\>	DateTime	Required	Server DateTime. System time according to the server.
\<SessKey\>	NC-64	Optional	Session Key.
			Included only at client request in \<SignonRq\>.
\<ExpDt\>	DateTime	Optional	Session Key Expiration DateTime. Included only if \<SessKey\> is returned.
\<Language\>	NC-17	Required	Language.
			See section 2.6.4 for more information on the format of this element.
			May or may not be the same as the Customer Preferred Language in \<SignonRq\>.

4.4.3   Signoff

A client that has no further messages to perform should send a \<SignoffRq\> to indicate to the CSP that no more messages will be sent in this session and server resources may be freed and reallocated to other customers. The client may begin another session whenever it has new messages to perform by using the \<SignonRq\>. Because there may be only one \<SignoffRq\> per IFX block, there is no \<RqUID\> in either the request or the response.

> *Note: Signon and Signoff are the only two messages to occur at the top level of IFX, rather than being contained within a service wrapper.*

4.4.3.1   Request \<SignoffRq\>

Tag	Type	Usage	Description
\<CustId\>	Aggregate *see section 3.2.3.1*	Optional	Customer ID. Provided only in the case where a CSR is forcing a Signoff of a customer.

4.4.3.2   Response \<SignoffRs\>

Tag	Type	Usage	Description
\<Status\>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, \<StatusCode\> defaults to 0 (zero).
\<CustId\>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer ID.

5 The Base Service <BaseSvc>

The Base Service in IFX includes messages that perform basic communication functions that must take place between parties (client and server or between different service providers). These messages are grouped into functions including Service Profile, Customer Authentication, Customer Profile, General Email, and Customer Service.

IFX uses Service Profile messages to communicate information about a Service Provider's services and configuration to a client.

Customer Profile Messages provide for the registration of Customers with Service Providers, and the maintenance of that information.

Customer Service messages provide administrative functions to assist Customer Service Representatives in serving customers.

5.1 Base Service Message Summary

Function / Message Name	Required	Comments
Service Profile Inquiry **<SvcProfInqRq>** **<SvcProfInqRs>**	Yes	Allows the client to retrieve a current view of the services supported by the SP, the Authentication Realms that must be used for authentication, and basic information about the SP.
Holiday Inquiry **<HolInqRq>** **<HolInqRs>**		Allows a client to retrieve a list of Bank Holidays observed by the Service Provider.
Customer Add **<CustAddRq>** **<CustAddRs>**		Allows client to provide customer personal details and indicate desire to use IFX Specification to communicate with a Financial Institution. SP may respond with Customer Login ID and Password or may provide these to the customer through another channel.
Customer Modify **<CustModRq>** **<CustModRs>**		Allows client to modify previously provided customer name, address, and contact information.
Customer Password Modify **<CustPswdModRq>** **<CustPswdModRs>**		Allows a client to modify a Customer Password. **Note:** *this message will be deprecated in IFX 2.0, replaced by the Customer Authentication Modify message.*
Customer Authentication Modify **<CustAuthModRq>** **<CustAuthModRs>**		Allows a client to modify a customer's authentication credentials. **Note:** *this message replaces the Customer Password Modify message, which will be deprecated in IFX 2.0*
Customer Authentication Reversal **<CustAuthRev>**		Allows a client to reverse a previous customer authentication modification
Customer Status Modify **<CustStatusModRq>** **<CustStatusModRs>**		The Customer Status Modify Message is used primarily by CSRs to reset the status and counters that were set when a user exceeded the invalid password threshold.
Customer Delete **<CustDelRq>** **<CustDelRs>**		Allows client to unenroll, deleting previously provided customer name, address, and contact information.

Function / Message Name	Required	Comments
Customer Inquiry **\<CustInqRq>** **\<CustInqRs>**	Yes	Allows client to retrieve current customer profile, including customer name, address, and contact information.
Customer Identifier Inquiry **\<CustIdInqRq>** **\<CustIdInqRs>**		Allows client to retrieve either the login id corresponding to a permanent id, or the permanent id corresponding to a login id.
Customer Audit **\<CustAudRq>** **\<CustAudRs>**		Allows client to audit Customer messages associated with the current customer.
Customer Synchronize **\<CustSyncRq>** **\<CustSyncRs>**		Allows client to synchronize Customer messages associated with the current customer.
Customer/Service Link Add **\<CustSvcAddRq>** **\<CustSvcAddRs>**		Allows client to request the enabling of a service for a customer. Optionally allows an SP to return terms and conditions to the customer for acceptance.
Customer/Service Link Modify **\<CustSvcModRq>** **\<CustSvcModRs>**		Allows the client to modify the fee account for a service.
Customer/Service Link Status Modify **\<CustSvcStatusModRq>** **\<CustSvcStatusModRs>**		Allows the client to modify the customer/service link status.
Customer/Service Link Delete **\<CustSvcDelRq>** **\<CustSvcDelRs>**		Allows client to request the disabling of a service for the customer.
Customer/Service Link Audit **\<CustSvcAudRq>** **\<CustSvcAudRs>**		Allows client to audit Service Enable and Disable messages associated with the current customer.
Customer/Service Link Synchronize **\<CustSvcSyncRq>** **\<CustSvcSyncRs>**		Allows client to synchronize Service Enable and Disable messages associated with the current customer.
Service/Account Link Add **\<SvcAcctAddRq>** **\<SvcAcctAddRs>**		Allows client to request activation of an account for one of the customer's services. Optionally allows an SP to return terms and conditions to the customer for acceptance.
ServiceAccount Link Modify **\<SvcAcctModRq>** **\<SvcAcctModRs>**		Allows client to request modification of an account for one of the customer's services. Optionally allows an SP to return terms and conditions to the customer for acceptance.
ServiceAccount Link Status Modify **\<SvcAcctStatusModRq>** **\<SvcAcctStatusModRs>**		Allows client to request modification of the status of an account for one of the customer's services.

Function / Message Name	Required	Comments
Service Account Reversal **\<SvcAcctRev\>**		Allows a client to reverse a previous service account request message.
ServiceAccount Identifier Link Modify **\<SvcAcctIdModRq\>** **\<SvcAcctIdModRs\>**		Allows client to request replacement of one account for another within one of the customer's services.
Service/Account Link Delete **\<SvcAcctDelRq\>** **\<SvcAcctDelRs\>**		Allows client to request deactivation of an account for one of the customer's services.
Service/Account Link Inquiry **\<SvcAcctInqRq\>** **\<SvcAcctInqRs\>**		Allows client to retrieve account and service current state, and account balances.
Service/Account Link Audit **\<SvcAcctAudRq\>** **\<SvcAcctAudRs\>**		Allows client to audit Account Activation and Deactivation messages associated with the current customer.
Service/Account Link Synchronize **\<SvcAcctSyncRq\>** **\<SvcAcctSyncRs\>**		Allows client to synchronize Account Activation and Deactivation messages associated with the current customer.
Disclosure Inquiry **\<DiscInqRq\>** **\<DiscInqRs\>**		Allows client to retrieve disclosures.
Customer/Disclosure Link Status Modify **\<CustDiscStatusModRq\>** **\<CustDiscStatusModRs\>**		Allows client to accept a disclosure received as part of the service enable/disable/modification and account activation/deactivation/modification process.
Customer/Disclosure Link Inquiry **\<CustDiscInqRq\>** **\<CustDiscInqRs\>**		Allows a customer to inquire about his/her associated disclosures for accounts or services.
Security Object Add **\<SecObjAddRq\>** **\<SecObjAddRs\>**		Allows a client to create a security object. A security object is used for distributing encryption keys with IFX business messages using an "in-band" process.
Security Object Modify **\<SecObjModRq\>** **\<SecObjModRs\>**		Allows a client to modify a security object.
Secuity Object Inquiry **\<SecObjInqRq\>** **\<SecObjInqRs\>**		Allows a client to inquiry on a security object.
Security Object Delete **\<SecObjDelRq\>** **\<SecObjDelRs\>**		Allows a client to delete an instance of a security object.

Function / Message Name	Required	Comments
Security Object Advise **<SecObjAdvRq>** **<SecObjAdvRs>**		This message is sent to a terminal (e.g. ATM or POS) to trigger a remote key load process. This message can also be sent by a terminal to provide information on a security object that was loaded using an "out-of-band" process (e.g. manually loaded, injected from another system, etc.).
Security Object Synchronize **<SecObjSyncRq>** **<SecObjSyncRs>**		Allows a client a means to synchronize the security objects known by both the client and server.
Party to Account Relationship Add **<PartyAcctRelAddRq>** **<PartyAcctRelAddRs>**		The Party Account Relationship Add function relates parties to accounts with relationship types. This function will also be used to add parties, owners and non-owners, to existing accounts that already have at least one party to account relationship.
Party to Account Relationship Delete **<PartyAcctRelDelRq>** **<PartyAcctRelDelRs>**		The Party Account Relationship Delete Function will delete related parties from a specific account.
Party to Account Relationship Inquiry **<PartyAcctRelInqRq>** **<PartyAcctRelInqRs>**		The Party Account Relationship Inquiry function returns parties related to a specific account, and can also be used to return the account associated with a particular party. This function will return base account ownership information such as Relationship Type, Primary Contact, and Tax Responsible Indicators. In addition, limited high level account information such as Account Status and Product will be returned.
Party to Account Relationship Modify **<PartyAcctRelModRq>** **<PartyAcctRelModRs>**		The Party Account Relationship Modify function is used to modify the relationship a party has with an account.

5.2 Base Service Common Elements and Aggregates

5.2.1 Service Profile Common Elements

5.2.1.1 Messages Supported Element <MsgSupt>

The <MsgSupt> element is included in the Service Profile for each service. It provides the client with a list of supported messages for that service.

Tag	Type	Usage	Description
<MsgSupt>	Open Enum		Supported Messages. This is a list of messages that are supported for the current service. The convention is to use the name of the message without the Rq or Rs so that each message is only listed once.

5.2.1.2 Options Supported Element <OptSupt>

The <OptSupt> element is included in each Service Profile where there are options that the client should know are supported or not supported by the server.

Tag	Type	Usage	Description
<OptSupt>	Open Enum		Options Supported.
			Defined values: AcctNickname, BillerPayee, BillStatus, CustPayeeNickname, ForEx, ForExCommit, FSPayee, ImmediateXfer, LineItem, RecCtrl, RecFinalCurAmt, RecInitialCurAmt, RecPend, SchedXfer, SkipInst, StopChkRangeSC, SuppressEcho, XferPayee

The following table identifies the services that use each option support value.

<OptSupt> value	Base	Bank	Pay	Pres
AcctNickname		×	×	×
BillerPayee			×	
BillStatus				×
CustPayee			×	
CustPayeeNickname			×	
ForEx		×	×	
ForExCommit		×		
FSPayee			×	
ImmediateXfer		×		
LineItem			×	
RecCtrl	×	×	×	×
RecFinalCurAmt		×	×	
RecInitialCurAmt		×	×	
SchedXfer		×		
SkipInst		×	×	
StopChkRangeSC		×		
XferPayee			×	

5.2.2   *Service Profile Common Aggregates*

5.2.2.1   **Service Core Aggregate <SvcCore>**

The <SvcCore> aggregate provides information about the service that is critical for the correct processing of that service.

Tag	Type	Usage	Description
<Version>	NC-12	Required	The version of IFX that the server supports. May be used by the server to infer that any tags added beyond this level are not supported.
<SPName>	Identifier	Required	Service Provider Name.
<LegalName>	C-96	Required	SP Legal Name.
<Nickname>	C-40	Optional	SP Nickname.
<HoldCoId>	Identifier	Optional	SP Holding Company Identifier. A unique identifier assigned by a service provider to identify the holding company that owns the SP branding company.
<USA.RTN>	NC-9	Optional	Routing and Transit Number. This is the default RTN for a customer account associated with the SPName above.

Tag	Type	Usage	Description
<OrgContact>	Aggregate *see section 3.2.5.2*	Required XOR Repeating	Service Provider Contact Aggregate. Usage is contact information for reaching relevant organizations at the SP (for example, the customer services area).
<ContactInfo>	Aggregate *see section 3.2.3.7*	Required XOR Repeating	Contact Information Aggregate.
<Language>	NC-17	Required Repeating	Language. See section 2.6.4 for more information on the format of this element. If the server supports more than one language for this service, multiple Language aggregates may be listed.
<URL>	URL	Optional	URL. <URL> identifies where the client should send the request for that service.

5.2.2.2   Service Profile Information Aggregate (xxx**SvcProfInfo**)

The Service Profile Information aggregate is used to express profile options for particular services. A list of these aggregates comprises part of the Service Profile Inquiry Response message <SvcProfInqRs> that is used to communicate server capabilities to a client.

Tag	Type	Usage	Description
<SvcCore>	Aggregate *see section 5.2.2.1*	Required	Service Core Aggregate. Information specified for every service.
Service specific elements and aggregates			Other elements and aggregates as appropriate for profiling options for the specific service.

The definition of this aggregate is in the last section of each service chapter.

5.2.2.3   Processing Schedule Aggregate <PrcSched>

The <PrcSched> aggregate appears wherever information about a Service Provider's processing schedule is needed. Note that this aggregate does not take holidays into account.

Tag	Type	Usage	Description
<PrcDaysOff>	Closed Enum	Optional Repeating	Processing Days Off. Days of the week on which no processing occurs. Valid values: Monday, Tuesday, Wednesday, Thursday, Friday, Saturday, Sunday, Holiday
<CutoffTm>	Time	Required	Cutoff Time for Daily Processing. This is the latest time a customer may submit a request for processing today.
<PrcDtAdj>	Open Enum	Required	Processing Date Adjustment. Algorithm used for adjustment of processing for requests that fall on non-processing days or holidays. Defined values: Earlier, Later.

5.2.3   Customer Profile Common Aggregates

5.2.3.1   Customer Record Aggregate <CustRec>

Tag	Type	Usage	Description

Tag	Type	Usage	Description
\<CustId\>	Aggregate *see section 3.2.3.1*	Required	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in \<SignonRq\>. For example, if a CSR or SP issues the request on behalf of the user, then \<CustId\> is required, and must contain the value of the user whose request is being issued.
\<CustInfo\>	Aggregate *see section 5.2.3.1.1*	Optional	Customer Information Aggregate.
\<CustStatus\>	Aggregate *see section 5.2.3.1.2*	Required	Customer Status Aggregate.
\<UpDt\>	Timestamp	Optional	Customer Record Update DateTime. The value is for the last update to \<CustInfo\> on the server.

5.2.3.1.1 Customer Information Aggregate \<CustInfo\>

The Customer Information aggregate \<CustInfo\> contains information about the customer, such as name, address, and contact information.

Tag	Type	Usage	Description
\<CustName\>	Aggregate *see section 3.2.3.2*	Required XOR	Customer Name Aggregate.
\<PostAddr\>	Aggregate *see section 3.2.3.7.1*	Optional Profiled requirement	Customer Postal Address Aggregate.
\<CustContact\>	Aggregate *see section 3.2.3.3*	Optional Profiled requirement	Customer Contact Aggregate.
\<ContactInfo\>	Aggregate *see section 3.2.3.7*	Optional	Contact Information Aggregate.
\<CustPref\>	Aggregate *see section 3.2.3.5*	Optional	Customer Preference Aggregate. An aggregate used to store preferences the customer has selected or for the service provider to store values to be used specifically for this customer.
\<CustType\>	Open Enum	Optional	Customer Type. The type of relationship established between the CSP and the customer. Valid values: Retail, Business.
\<PersonInfo\>	Aggregate *see section 3.2.4.1*	Required XOR 1.1+	Person Information Aggregate
\<OrgInfo\>	Aggregate *see section 3.2.5.1.2*	Required XOR 1.1+	Organization Information Aggregate

5.2.3.1.2 Customer Status Aggregate \<CustStatus\>

Tag	Type	Usage	Description

Tag	Type	Usage	Description
<CustStatusCode>	Open Enum	Required	Customer Status Code.
			Defined values: Enrolled, EnrollPend, Suspended, UnenrollPend, Unenrolled
<StatusDesc>	C-255	Optional	Status Description. Explanatory text associated with this service status.
<EffDt>	DateTime	Optional	Effective Date/Time. The date/time the <CustStatusCode> took effect.
<StatusModBy>	Open Enum	Optional	Status Modified By. If present, indicates who modified the Customer Status Code.
			Defined values: Customer, CSP, CSPSR, BSP, BSPSR, CPP, CPPSR, BPP, and BPPSR.
<ResolvePendingDt>	Date	Optional	Resolve Pending Date. The date that the *xxx*Pend status in <CustStatusCode> is expected to change to a final status; e.g., Enrolled, Unenrolled. This date should be returned when the <CustStatusCode> is set to *xxx*Pend.

5.2.3.2   Disclosure Record Aggregate <DiscRec>

The <DiscRec> aggregate allows the CSP/SP to send Terms and Conditions, related to the IFX Service, to the customer. The CSP/SP may optionally require acceptance of these terms and conditions prior to enabling a service or activating/modifying an account for a service.

> *Note:*   *Customer acceptance of a disclosure is associated with the Disclosure Identifier <DiscId> for that disclosure. If it is necessary to track the version of a disclosure that a customer has accepted, it is recommended that a service provider use a new <DiscId> any time that they modify their disclosure. The <DiscId> itself is the versioning mechanism for disclosures within IFX (see Section 5.4.1.2 for additional information on Disclosures).*

Tag	Type	Usage	Description
<DiscId>	UUID	Required	Disclosure Identifier
<DiscInfo>	Aggregate *see section 5.2.3.2.1*	Required	Disclosure Information

5.2.3.2.1   Disclosure Information Aggregate <DiscInfo>

Tag	Type	Usage	Description
<LongText>	C-∞	Required XOR	Disclosure Text.
<DiscURL>	URL	Required XOR	Disclosure URL. The address for obtaining human-readable disclosure information.
<AcceptReqd>	Boolean	Optional	Disclosure Accept Required—The server indicates whether the acceptance of this disclosure is required to proceed. If set to *True*, acceptance is required.

5.2.3.3   Customer/Disclosure Link Record Aggregate <CustDiscRec>

Tag	Type	Usage	Description
<CustId>	Aggregate *see section 3.2.3.1*	Required	Customer Identifier
<DiscId>	UUID	Required	Disclosure Identifier

Tag	Type	Usage	Description
<DiscInfo>	Aggregate *see section 5.2.3.2.1*	Optional	Disclosure Information. Provided if requested by the client.
<CustDiscStatus>	Aggregate *see section 5.2.3.3.1*	Required	Customer/Disclosure Link Status

5.2.3.3.1 Customer/Disclosure Link Status Aggregate <CustDiscStatus>

Tag	Type	Usage	Description
<CustDiscStatusCode>	Closed Enum	Required	Customer/Disclosure Link Status Code Valid values: None, Accepted, Rejected, Viewed.
<StatusDesc>	C-255	Optional	Status Description. Explanatory text associated with this customer/disclosure link status.
<EffDt>	DateTime	Optional	Effective Date/Time. The date/time the <CustDiscStatusCode> took effect.
<StatusModBy>	Open Enum	Optional	Status Modified By. If present, indicates who modified the Status Code. Defined values: Customer, FI, CSP, CSPSR, BSP, BSPSR, CPP, CPPSR, BPP, and BPPSR.

5.2.3.4 Customer/Service Link Record Aggregate <CustSvcRec>

Tag	Type	Usage	Description
<CustSvcId>	Aggregate *see section 5.2.3.4.1*	Required	Customer/Service Link Identification Aggregate.
<CustSvcInfo>	Aggregate *see section 5.2.3.4.2*	Required	Customer/Service Link Information Aggregate.
<CustSvcStatus>	Aggregate *see section 5.2.3.4.3*	Required	Customer/Service Status Aggregate.

5.2.3.4.1 Customer/Service Link Identification Aggregate <CustSvcId>

Tag	Type	Usage	Description
<SvcName>	Open Enum	Required	Service Name. Defined values: Bank, Pay, Pres
<SPName>	Identifier	Required	Service Provider Name.

5.2.3.4.2 Customer/Service Link Information Aggregate <CustSvcInfo>

Tag	Type	Usage	Description
<DepAcctId>	Aggregate *see section 3.2.7.1.2*	Optional XOR	Deposit Account Aggregate. Used to specify a deposit account to be used for charging fees for the service.
<CardAcctId>	Aggregate *see section 3.2.7.1.3*	Optional XOR	Card Account Identifier Aggregate. Used to specify a card account to be used for charging fees for the service.

5.2.3.4.3 Customer/Service Link Status Aggregate <CustSvcStatus>

Tag	Type	Usage	Description
<CustSvcStatusCode>	Closed Enum	Required	Customer/Service Status Code.
			Valid values: Enabled, EnablePend, Disabled, DisablePend, Rejected
<StatusDesc>	C-255	Optional *but see Description*	Status Description. Explanatory text associated with this customer/service link status.
			Required if service change is rejected.
<EffDt>	DateTime	Optional	Effective Date/Time. The date/time the <CustSvcStatusCode> took effect.
<StatusModBy>	Open Enum	Optional	Status Modified By. If present, indicates who modified the Status Code.
			Defined values: Customer, FI, CSP, CSPSR, BSP, BSPSR, CPP, CPPSR, BPP, and BPPSR.
<ResolvePendingDt>	Date	Optional	Resolve Pending Date. The date that the *xxx*Pend status in <CustSvcStatusCode> is expected to change to a final status; e.g., Enabled, Rejected. This date should be returned when the <CustSvcStatusCode> is set to *xxx*Pend.

5.2.3.5 Service/Account Link Record Aggregate <SvcAcctRec>

Tag	Type	Usage	Description
<SvcAcctId>	Aggregate *see section 5.2.3.5.1*	Required	Service/Account Link Identifier Aggregate.
<SvcAcctInfo>	Aggregate *see section 5.2.3.5.2*	Required	Service/Account Link Information Aggregate.
<SvcAcctStatus>	Aggregate *see section 5.2.3.5.3*	Required	Service/Account Status Aggregate.
<EffDt>	DateTime	Optional	Effective Date. The date that the first account information (i.e. statement, bill, etc.) is expected to be available. This date should be returned when the <SvcAcctStatusCode> for an account is set to Activated. For billing accounts, this is the date of the first available bill.
			If absent, the semantic meaning is that information is available immediately.

5.2.3.5.1 Service/Account Link Identification Aggregate <SvcAcctId>

Tag	Type	Usage	Description
<CustSvcId>	Aggregate *see section 5.2.3.4.1*	Required	Customer/Service Link Identifier Aggregate.
<DepAcctId>	Aggregate *see section 3.2.7.1.2*	Required XOR	Deposit Account Identifier Aggregate.

Tag	Type	Usage	Description
<CardAcctId>	Aggregate *see section 3.2.7.1.3*	Required XOR	Card Account Identifier Aggregate.
<LoanAcctId>	Aggregate *see section 3.2.7.1.4*	Required XOR	Loan Account Identifier Aggregate.
<PresAcctId>	Aggregate *see section 3.2.7.2.1*	Required XOR	Presentment Account Identifier Aggregate.
<PresAcctInfo>	Aggregate *see section 3.2.7.2.2*	Required	Presentment Account Information Aggregate.

5.2.3.5.2   Service/Account Link Information Aggregate <SvcAcctInfo>

Tag	Type	Usage	Description
<Nickname>	C-40	Optional	Account Nickname, as held by the Service Provider.

5.2.3.5.3   Service/Account Link Status Aggregate <SvcAcctStatus>

Tag	Type	Usage	Description
<SvcAcctStatusCode>	Closed Enum	Required	Service/Account Status Code. Valid values: Activated, ActivatePend, Deactivated, DeactivatePend, Rejected, Suspended
<StatusDesc>	C-255	Optional *but see Description*	Status Description. Explanatory text associated with this customer/service link status. Required if service change is rejected.
<EffDt>	DateTime	Optional	Effective Date/Time. The date/time the <SvcAcctStatusCode> took effect.
<StatusModBy>	Open Enum	Optional	Status Modified By. Defined values: CSP, CPP, BSP, BPP, FI
<ResolvePendingDt>	Date	Optional	Resolve Pending Date. The date that the *xxx*Pend value in <SvcAcctStatusCode> is expected to change to a final status; e.g., Activated, Rejected. This date should be returned when the <SvcAcctStatusCode> is set to *xxx*Pend.

5.2.3.6   Security Object Record Aggregate <SecObjRec>

Note: For security reasons, <SecObjValue> is not to be included in the Security Object Record Aggregate unless the object is not of sensitive nature.

Tag	Type	Usage	Description
<SecObjId>	Identifier	Required	Security Object Identifier used to reference this security object in future request
<SecObjKeyChkValue>	HexBin-∞	Optional	Key Check value for those protocols that can generate a key Check value (i.e. DEA, AES)
<SecObjName>	C-64	Optional But See Description	Human readable object description. If there are two or more Security Objects of with the same <SecObjType>/<SecObjPurpose>, then this field is required in order to differentiate between the usages of the key.

<SecObjType>	Open Enum	Optional	Type of Security Object.
			Valid Value: 1KeyDES, 2KeyDES, 3KeyDES, 128BitAES, 192BitAES, 256BitAES, DigitalCertificate, PublicKey, RandomNumber, ModuleID
<SecObjPurpose>	Open Enum	Optional	Purpose (use) of the Security Object
			Note: The Algorithm that employs this key is agreed to out of band.
			Valid Values: PINEncryptingKey, KeyEncryptingKey, PasswordEncryptingKey, DataEncryptingKey, Identification, MAC, Uniqueness, VISAPINVerification
<CryptType>	Open Enum	Optional	Type of encryption used to encrypt <SecObjValue>. Depending on the Security Object, this represents how the <SecObjValue> data is formatted. Note, must be present is <SecObjValue> is present.
			The only <CryptType> values that make logical sense for use within the <SecObjRec> are the following:
			Valid Values: PKCS#1, PKCS#7, PKCS#10, None
<SecObjValue>	HexBin-∞	Optional	For an Inquiry message, the security object is returned in the case of a digital signature or public key since there are not security concerns associated with transporting this key
			For an add message in which the KeyEncryptingKey is loaded via asymmetric cryptography, this is the ATM verification of the key load (Host Random Number, ATM Random Number signed by ATM Identification Key
<SecObjStatus>	Aggregate *see section 5.2.3.6.2*	Required	Security Object Status

5.2.3.6.1   Security Object Information Aggregate <SecObjInfo>

Tag	Type	Usage	Description
<SecObjName>	C-64	Optional But See Description	Human readable object description. If there are two or more Security Objects of with the same <SecObjType>/<SecObjPurpose>, then this field is required in order to differentiate between the usages of the key.
<SecObjType>	Open Enum	Required	Type of Security Object.
			Valid Value: 1KeyDES, 2KeyDES, 3KeyDES, 128BitAES, 192BitAES, 256BitAES, DigitalCertificate, PublicKey, RandomNumber. ModuleID
<SecObjPurpose>	Open Enum	Required	Purpose (use) of the Security Object
			Note: The Algorithm that employs this key is agreed to out of band.
			Valid Values: PINEncryptingKey, KeyEncryptingKey, PasswordEncryptingKey, DataEncryptingKey, Identification, MAC, Uniqueness, VISAPINVerification
<CryptType>	Open Enum	Required	Type of encryption used to encrypt <SecObjValue>. Depending on the Security Object, this represents how the <SecObjValue> data is formatted
			Valid Values: PKCS#1, PKCS#7, PKCS#10, ECB, CBC, CFB, OFB, TECB, TCBC, TCBC-I, TCFB, TCFB-P, TOFB, TOFB-I, AECB, ACBC, ACFB, AOFB, ACM, None

\<SecEncryptId\>	Identifier	Optional But see Description	Identifier of Security Object on which the \<SecObjValue\> is encrypted. \<SecEncryptjId\> is required if \<SecObjValue\> is encrypted.
\<SecObjValue\>	HexBin-∞	Required	Actual Security Object being transported (encryption key, digital certificate, etc.)

5.2.3.6.2 *Security Object Status \<SecObjStatus\>*

Tag	Type	Usage	Description
\<SecObjStatusCode\>	Closed Enum	Required	Security Object Status Code. This identifies the security object status.
			Valid Values: Active, Inactive, Expired, OutOfSync
\<StatusDesc\>	C-255	Optional	Status Description. Explanatory text associated with this status
\<EffDt\>	DateTime	Optional	Security Object Status date. The date associated with the state change to the current status
\<StatusModBy\>	Open Enum	Optional	Indicates the entity to last modify the object status
			Valid Values: Host, Terminal, Maintenance

5.2.4   *Party Account Relationship Info \<PartyAcctRelInfo\>*

Tag	Type	Usage	Description
\<CustId\>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate.
\<DepAcctId\>	Aggregate *see section 3.2.7.1.2*	Required XOR	Deposit Account Identification Aggregate.
\<CardAcctId\>	Aggregate *see section 3.7.1.3*	Required XOR	Card Account Identification Aggregate.
\<LoanAcctId\>	Aggregate *see section 3.2.7.1.4*	Required XOR	Loan Account Identification Aggregate.
\<PartyAcctRel\>	Aggregate *see section 5.2.4.3*	Required Repeating	Party to account relationship aggregate.
\<OwnerPercent\>	Decimal	Optional	Percentage of account ownership. Used primarily for Loan Accounts. Usage is a percentage. (e.g. – a value of 5.2 = 5.2%)

5.2.4.1   **Party Account Relationship Record \<PartyAcctRelRec\>**

Tag	Type	Usage	Description
\<PartyAcctRelId\>	Identifier	Required	Party Account Relationship Identifier.
\<PartyAcctRelInfo\>	Aggregate *see section 5.2.4*	Required	Party Account Relationship Info Aggregate.
\<FullName\>	C-96	Optional	Full name of the person or business.

Tag	Type	Usage	Description
<TINInfo>	Aggregate *see section 3.2.4.1.3*	Optional	Tax Identifier Number aggregate.
<OwnerInd>	Boolean	Optional	Indicates if the relationship is of the owner or non-owner type.
<BankAcctStatus>	Aggregate *see section 3.2.7.1.6*	Optional	Bank Account Status Aggregate. Intended use within the PartyAcctRelRec is when the accounts associated with a customer are being returned for the PartyAcctRelInqRs.
<OpenDt>	Date	Optional	Indicates the date the account opened.
<ClosedDt>	Date	Optional	Indicates the date the account closed
<PrimaryAcct>	Boolean	Optional	Primary Account Indicator. If *True*, used as default account for a class of accounts (i.e., checking). Typically used in ATM networks. May be used for other applications by some FIs.
<NickName>	C-40	Optional	Used to represent an account nickname applied by the customer of FI.
<AcctBal>	Aggregate *see section 3.2.8.1*	Optional Repeating	Account balance aggregate.
<PartyAcctRelStatus>	Aggregate *see section 5.2.4.2*	Optional	Party Account Relationship Status.

5.2.4.2 Party Account Relationship Status <PartyAcctRelStatus>

Tag	Type	Usage	Description
<PartyAcctRelStatusCode>	Closed Enum	Required	Party/Account Status Code. Valid values: Active, InActive, PendDelete
<StatusDesc>	C-255	Optional *but see Description*	Status Description. Explanatory text associated with this party/account link status. Required if service change is rejected.
<EffDt>	DateTime	Optional	Effective Date/Time. The date/time the <PartyAcctStatusCode> took effect.
<StatusModBy>	Open Enum	Optional	Status Modified By. Defined values: CSP, CPP, BSP, BPP, FI

5.2.4.3 Party Account Relationship <PartyAcctRel>

Tag	Type	Usage	Description

Tag	Type	Usage	Description
<PartyAcctRelType>	Open Enum	Required Repeating	Party Account Relationship Type. Defined values are:
			Tax Responsible
			Primary Contact
			Doing Business As
			Individual
			Individual with Rights
			Joint Tenancy
			Joint Tenants in Common
			Joint with Rights
			Multiple Signer
			Custodian
			Guardian
			Executor
			Administrator
			Guarantor
			Comaker
			Trustee
			Payee
			Power of Attorney
			Pay On Death
<PartyAcctRelDesc>	C-80	Optional	Party Account Relationship Description.

5.3   Service Profile

5.3.1   Description

The Service Profile has two messages, a Service Profile Inquiry message, and a Holiday Inquiry message. The Holiday Inquiry Message allows a client to retrieve a list of Bank Holidays observed by a Financial Institution or Service Provider.

The Service Profile Inquiry Message allows a client to retrieve the Service Profile from a Financial Institution, CSP, or other Service Provider.

The Service Profile contains the following information about the Customer's Service Provider:

- Services Supported—Services are collections of messages that are functionally related. Each Service is designed with a corresponding section in the Service Profile to allow each Service Provider to inform clients as to which messages, functions, and options of the service supported.

- Signon Realms—Signon Realms provide a mechanism for a Service Provider to manage customer authentication across services. Service Providers may choose not to support IFX authentication when communicating to other service providers. Server-to-server authentication may take place in some other way.

- Service Provider Information—Provides the legal name of the SP, the SP address, the SP's Customer Service telephone number, and other useful information for that service provider.

Each time a client authenticates with a Signon Realm, the time and date of the current Service Profile is returned by the server. If the client does not have a copy of the current Service Profile, it should perform a

Service Profile Inquiry message to retrieve the current Service Profile. This mechanism is used to inform clients of changes in server capabilities.

Note: All IFX servers must support the Service Profile Inquiry message.

5.3.2   Service Profile Inquiry Message

5.3.2.1   Request <SvcProfInqRq>

The client submits a <SvcProfInqRq> that optionally includes a Timestamp value <UpDt> for the last Service Profile received from the server. <UpDt> indicates when the Service Profile was last updated on the server. If <CustId> is not available (because of an anonymous login, for example), a generic profile must be returned. If <CustId> is available, then a custom profile for that customer may be returned. This capability allows the SP to support capabilities such as: (1) providing different contact information for preferred customers or (2) providing information specific to service providers for this particular customer when the SP has relationships with multiple service providers for the same service.

Tag	Type	Usage	Description
<RqUID>	UUID	Required	Request Identifier. Sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header aggregate.
<AsyncRqUID>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously.
			For more information, see section 3.2.12.1.
<CustId>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<DeliveryMethod>	Open Enum	Optional Profiled values	Delivery Method. Default is Channel.
			Value must be supported in Service Profile.
<UpDt>	Timestamp	Optional	Timestamp when the Service Profile was Last Updated on the Server.
			This is the timestamp of the last Service Profile received from the server.

5.3.2.2   Response <SvcProfInqRs>

The Service Profile Inquiry Response Message returns the following information in the following order:

- Sections for each service that the SP supports, including the parameters for that service, and the associated signon realm; Information about the SP, e.g. name, address, contact information, is included in <SvcCore> for the Base service, and
- Information about the signon realms.

If there have been no updates to the Service profile, the server should return <StatusCode> 1 to indicate that the client has the most current Service Profile. In this case, the server should not return the <*xxx*SvcProfInfo> aggregates.

Tag	Type	Usage	Description

Tag	Type	Usage	Description
\<Status\>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, \<StatusCode\> defaults to 0 (zero).
\<RqUID\>	UUID	Required Echoed	Request Identifier.
\<MsgRqHdr\>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header aggregate.
\<MsgRsHdr\>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header aggregate.
\<AsyncRqUID\>	UUID	Optional Echoed	Asynchronous Request Identifier.
\<CustId\>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in \<SignonRq\>. For example, if a CSR or SP issues the request on behalf of the user, then \<CustId\> is required, and must contain the value of the user whose request is being issued.
\<DeliveryMethod\>	Open Enum	Optional Echoed	Delivery Method.
\<UpDt\>	Timestamp	Optional Echoed	Timestamp when the Service Profile was Last Updated on the Server. The last timestamp received by the client.
\<NewUpDt\>	Timestamp	Required	New Service Profile Update Date Time. The value is for the last update to the most recent \<*xxx*SvcProfInfo\> on the Server.
\<*xxx*SvcProfInfo\>	Aggregate	Optional Repeating	Service Profile Information Aggregate. At least one per service supported.

5.3.3   Holiday Inquiry Message

The Holiday Inquiry message allows a client to retrieve a list of Bank Holidays observed by the Service Provider. The list of holidays is typically used to adjust the processing date for a scheduled payment or transfer.

5.3.3.1   Request \<HolInqRq\>

The client submits a \<HolInqRq\> in order to request a list of the Service Provider's observed holidays.

Tag	Type	Usage	Description
\<RqUID\>	UUID	Required	Request Identifier. Sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.
\<MsgRqHdr\>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header aggregate.
\<AsyncRqUID\>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously. For more information, see section 3.2.12.1.

Tag	Type	Usage	Description
<SPName>	Identifier	Optional *but see Description*	Service Provider Name. If omitted, the receiver of the message must determine (1) that the message is unambiguous and provide a successful response or (2) that it is ambiguous and provide an error response.
<SvcName>	Open Enum	Optional *but see Description*	Service Name. If omitted, the server must determine (1) that the message is unambiguous and provide a successful response or (2) that it is ambiguous and provide an error response.
			Defined values: Bank, Pay, Pres.
<DeliveryMethod>	Open Enum	Optional Profiled values	Delivery Method. Default is Channel.
			Value must be supported in Service Profile.

5.3.3.2   Response <HolInqRs>

The server at the Financial Institution or Service Provider responds with a list of observed Bank Holidays if available.

Tag	Type	Usage	Description
<Status>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, <StatusCode> defaults to 0 (zero).
<RqUID>	UUID	Required Echoed	Request Identifier.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header aggregate.
<MsgRsHdr>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header aggregate.
<AsyncRqUID>	UUID	Optional Echoed	Asynchronous Request Identifier.
<SPName>	Identifier	Optional Echoed	Service Provider Name.
<SvcName>	Open Enum	Optional Echoed	Service Name.
			Defined values: Bank, Pay, Pres
<DeliveryMethod>	Open Enum	Optional Echoed	Delivery Method.
<HolInfo>	Aggregate	Optional Repeating	Holiday Information Aggregate.
<Name>	C-40	Required	Holiday Name.
<HolDt>	Date	Required	Holiday Date.
</HolInfo>			

5.4   Customer Profile

5.4.1   Description

Customer Profile Messages provide the following functions:

- *Enrollment and Acquisition of Login ID and Password.* An Enrollment message that allows an individual to submit enough personal information to be identified to a Service Provider is provided. If the SP supports customer self-enrollment using the IFX Specification, it may elect to return a login ID and password to the customer. Alternately, the SP may collect the customer's information and use a separate process (possibly a mailing or a customer service telephone call) to provide the customer with a login and password.

- *Updating of Customer Name, Address, and Contact Information.* After the customer initially provides his or her personal information through the enrollment process, he or she may need to update this information periodically. The Modify Customer Information message is provided for this purpose.

- *Customer Profile*. The Customer Profile message provides a current view of the customer's information, accounts balances, and services. It is useful both in initial setup and on an ongoing basis.

- *Service Enabling and Disabling*. Services may be enabled for a customer, independently of any account. The customer and CSP may use the Customer/Service Link Add and Customer/Service Link Delete messages to manage which services are currently activated for the customer. A fee account may optionally be specified. Any fees attributable to the use of the service are charged to the fee account.

- *Service Modification.* The Customer/Service Link Modify message is used to change the fee account.

- *Account Activation and Deactivation.* Accounts may be activated for use by a specific service. The customer and CSP may use the Service/Account Add and Service/Account Delete messages to manage which accounts are currently activated for each service.

- *Account Modification.* The Service/Account Modify message is used to modify a funding account for the Pay service, or to change an account nickname.

5.4.1.1   Account Activation and Deactivation

Accounts are established with an SP and may be used electronically if activated for a service; e.g., a checking account may be activated for electronic banking (Bank service). Another CSP/ SP may reference the account; e.g., the Pay service provider may use it as a funding account. Thus, the meaning of activating an account for a service is specific to the service.

The use of an account may be constrained based on a combination of the type of account, its tax status, and CSP/SP policy. For example:

- A certificate of deposit (CD) account may not be a funding or fee account for Pay.
- A CSP might not allow interbank transfers from a savings (SDA) account.

The allowable usage for a bank account is specified by a series of aggregates <XferFromSupt>, <XferToSupt>, and <PaySupt> within the Bank Account Information Aggregate <BankAcctRec>. These tags are sometimes referred to as the account's message authority.

The following table defines the meaning of activating an account for a service.

Service	Activation Means	Comments
Base	Not applicable	
Bank	Banking Accounts	Allows all banking services subject to message authority.
Pay	Funding Accounts	
Pres	Biller Accounts	Customer's account at the biller.

Services that normally require an account to be associated with them may remain enabled even when no account is activated. These situations are typically temporary conditions. For example, a customer's checkbook for the sole Pay funding account may be stolen. The checking account is closed immediately to reduce the risk of loss, and there is a processing delay while a replacement account number is assigned. The Pay service must remain enabled during the processing delay, even though there was no funding account specified, in order to

maintain the customer payee list, pending payments, and payment models that are defined. The Modify Account message is used to replace an existing account (e.g., change a funding account for Pay), change information for a Bill Presentment account, and modify the account nickname

> *Note: The Customer/Service Link and Service/Account Link messages are intentionally limited to a single link per message to simplify error reporting when the request is rejected by the SP.*

5.4.1.2   Disclosures

IFX Services may have terms and conditions associated with them. These terms and conditions may be a combination of fees, terms mandated by statute, and terms required by the SP. Banking regulations and contract law may restrict the acceptable means of notification; e.g., US Mail to the address of record in the US.

Some services, notably Bill Presentment, may require disclosures at the account level. In Bill Presentment, for example, individual Billers may need to disclose their specific terms and conditions for the Bill Presentment Service. Thus, disclosures may be associated with account-level messages as well as with service-level messages. Note that the disclosure-related structures within IFX were intended for exchange of disclosures related to the electronic service being facilitated by IFX (e.g., online banking, bill payment service, bill presentment service). Disclosures that relate to the underlying product or service, which is being billed in bill presentment, for example, are likely to be customer specific and can be delivered within a bill summary with <BillType> = Notice, using the <Memo> field, or can be displayed within the bill detail. An example would be a gas and electric company who must send notification of rate changes related to gas service. They would use the Bill Summary structure within IFX to send this notification, rather than the Disclosure structure.

Disclosures are presented when a service is enabled or an account is activated, and can be obtained in advance using a disclosure inquiry (typically used by an SP to query another SP). Changes to the terms and conditions that were presented to the customer may also need to be presented. These may be considered to be "server-initiated changes" and are conveyed to the client through standard IFX mechanisms described below.

Disclosures are sent to the client as part of <DiscInqRs>, <CustSvcAddRs>, <CustSvcModRs>, <SvcAcctAddRs>, or <SvcAcctModRs>, and the client sends confirmations using <CustDiscStatusModRq>.

The SP must indicate in the appropriate Rs message whether a disclosure should be presented to the customer for information only or whether it requires a confirmation.

> *Note: Customer acceptance of a disclosure is associated with the Disclosure Identifier <DiscId> for that disclosure. If it is necessary to track the version of a disclosure that a customer has accepted, it is recommended that a service provider use a new <DiscId> any time that they modify their disclosure. The <DiscId> itself is the versioning mechanism for disclosures within IFX. Within the Bill Presentment Service, in the event that a Biller needs to republish disclosure information after the initial Bill Presentment account activation, they may choose to send a Bill Summary with the <BillType>=Notice. The Notice provides the ability to send any type of notification information to a customer using the <Memo> field. For disclosure-type notices, the <Memo> field may contain the disclosure text or may give instructions for accessing a URL. Note that this process of utilizing the Bill Summary for the delivery of disclosure information does not provide the Biller with the ability to receive a customer acceptance response. See Section 8.1.1 for additional information.*

5.4.1.3   Customer Profile Retrieval via Service/Account Link

IFX allows a Service Provider to generate a customer profile, containing a list of the services for which the customer has access and a list of accounts associated with those services, for each customer who identifies himself or herself through an enrollment process. Once a customer is identified, the profile may be automatically generated to include information about the customer and his or her accounts.

The Service/Account Link Inquiry message provides an efficient mechanism for client software to determine the current state of the customer's accounts and what services are activated. This capability is especially important for customers who use multiple channels or multiple software clients to interact with an SP.

The format of the response to a Service/Account Link Inquiry is constant regardless of which SP generates it; however, the content may differ. For example, a CSP may return information about all of the checking accounts

for a customer, while the Pay Service Provider may choose to return only the identifiers of the checking accounts that have been identified to it as funding accounts.

5.4.1.4 Using the Customer Profile

A list of available accounts and services is often displayed to the end user so that he or she may select from them. This technique is preferred over asking the user to enter the information since it eliminates the possibility of errors when the user keys in account identifiers; e.g., account number, account type, routing and transit numbers.

The list of services available to the customer, or being used by the customer, is contained within <CustSvcRec>. A service is *Available* if the SP offers the service and the specific customer may enable it.

The list of available accounts is contained within <SvcAcctRec>. This aggregate may be used for displaying appropriate candidates for use within a specific service.

5.4.2 Customer

5.4.2.1 Customer Add Message

The Customer Add message is used to add a Customer record at a Service Provider. It also allows an individual to submit enough personal information to be identified to a Customer Service Provider. If the CSP supports customer self-enrollment using IFX, it may elect to return a login ID and password to the customer. Alternatively, the CSP may collect the customer's information and use a separate process (possibly a mailing or a customer service telephone call) to provide the customer with a login and password.

A CSP may choose not to support the Customer Add functionality defined in IFX for enrollment, and may provide a customized World Wide Web (WWW) page or some other means to support its own enrollment process. In this case, the CSP may communicate the URL for this page in the Service Profile. A short note about an alternative enrollment process may also be provided to the client through the Service Profile.

5.4.2.1.1 Request <CustAddRq>

If the CSP supports customer self-enrollment using this message, the CSP may specify what information it must receive to authenticate the customer using the <SecretPrompt> aggregate within the <EnrollProf> aggregate in the Service Profile. Information requested through this mechanism typically includes things like Tax ID, Date of Birth, or Mother's Maiden Name. The customer must provide this information in the <CustAddRq> to prove that the enrollment is valid against existing customer information stored in the SP customer database.

The secret list is provided so that a customer new to this channel may be authenticated against existing customer records at the SP. Note that this is a variable-length list, depending on the requirements of the SP as expressed in <SecretPrompt> in the Service Profile. The client should display each <SecretPrompt> and collect customer input. This information is returned in the <SecretList> either in clear text (relying on channel security) or encrypted.

> *Note:* *The Customer Add message requires special handling because it does not assume that a client is authenticated. By definition, a client that is performing an enrollment using the Customer Add message does not have a Customer Login ID or Password. Therefore, the Signon is likely anonymous under these circumstances.*

Tag	Type	Usage	Description
<RqUID>	UUID	Required	Request Identifier. Sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header aggregate.

Tag	Type	Usage	Description
<AsyncRqUID>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously. For more information, see section 3.2.12.1.
<CustInfo>	Aggregate *see section 5.2.3.1.1*	Optional Profiled requirement	Customer Information Aggregate.
<SecretList>	Aggregate *see section 3.2.14.2*	Optional Repeating AND	Secret List. This aggregate is for a customer to input the secret required for client enrollment.
<CryptType>	Open Enum	Optional AND Profiled values	Encryption Type. Indicates the encryption algorithm used for the secrets within the <SecretList>. Must be one of the supported types such as the ones defined in the <EnrollProf> aggregate in <BaseSvcProfInfo>. Defined values: None, PKCS#1, PKCS#7, PKCS#10, ECB, CBC, CFB, OFB, TECB, TCBC, TCBC-I, TCFB, TCFB-P, TOFB, TOFB-I, AECB, ACBC, ACFB, AOFB, ACM
<SecObjId>	Identifier	Optional Repeating AND	Identifier of Security Object on which the <SecretList> is encrypted Note: If security objects are managed in band, this element should be included in order to know how to decrypt <CryptPswd>
<CustId>	Aggregate *see section 3.2.3.1*	Optional	Customer ID. Usage is actual <CustId> if preassigned, or preference for SPs that allow customer to pick.
<AuthInfo>	Aggregate	Optional	Authentication Information. The client may provide information within this aggregate to send to the server the authentication tokens (such as password; other mechanisms will be defined in future releases of IFX) that may be used in future communications with the server.
<CustPswd>	Aggregate *see section 4.3.1*	Optional	Customer Password Aggregate.
</AuthInfo>			

5.4.2.1.2 Response <CustAddRs>

The <CustAddRs> message acknowledges the client's Enroll request. If the SP chooses to deliver a Customer Login ID and a password in the response, they are also included. Otherwise, the SP may provide the Customer Login ID and password through the postal mail, electronic mail, or over the telephone.

> **Note:** When the <CustAddRs> message is returned in an Audit Response <CustAudRs> or Sync Response <CustSyncRs>, <CryptType> and <PswdDelivery> may not be included, for security reasons.

Tag	Type	Usage	Description
<Status>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, <StatusCode> defaults to 0 (zero).
<RqUID>	UUID	Required Echoed	Request Identifier. Optionally sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.

Tag	Type	Usage	Description
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header aggregate.
<MsgRsHdr>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header aggregate.
<AsyncRqUID>	UUID	Optional Echoed	Asynchronous Request Identifier.
<CustInfo>	Aggregate *see section 5.2.3.1.1*	Optional Echoed	Customer Information Aggregate.
<CustId>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Login ID.
<CustRec>	Aggregate *see section 5.2.3.1*	Required	Customer Record Aggregate.
<PswdDelivery>	Aggregate	Optional	Password Delivery Aggregate.
<CustPswd>	Aggregate *see section 4.3.1*	Required	Customer Password Aggregate.
<ExpDt>	DateTime	Optional	Password Expiration date and time. If absent, the password never expires.
</PswdDelivery>			
<SPRefId>	Identifier	Optional	Service Provider Reference Identifier.

5.4.2.2   Customer Modify Message

A client may use the Customer Modify message to update his or her name, address, or contact details in the SP's records. For information on Conventions for Modification of Server-Based Data, see Section 2.4.3.2.

5.4.2.2.1   Request <CustModRq>

Tag	Type	Usage	Description
<RqUID>	UUID	Required	Request Identifier. Optionally sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header aggregate.
<AsyncRqUID>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously. For more information, see section 3.2.12.1.
<CustId>	Aggregate *see section 3.2.3.1*	Optional	Customer ID.
<CustInfo>	Aggregate *see section 5.2.3.1.1*	Required	Customer Information Aggregate.

5.4.2.2.2 Response <CustModRs>

Tag	Type	Usage	Description
<Status>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, <StatusCode> defaults to 0 (zero).
<RqUID>	UUID	Required Echoed	Request Identifier.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header aggregate.
<MsgRsHdr>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header aggregate.
<AsyncRqUID>	UUID	Optional Echoed	Asynchronous Request Identifier.
<CustId>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer ID.
<CustInfo>	Aggregate *see section 5.2.3.1.1*	Required Echoed	Customer Information Aggregate.
<CustRec>	Aggregate *see section 5.2.3.1*	Required	Customer Record Aggregate.
<SPRefId>	Identifier	Optional	Service Provider Reference Identifier.

5.4.2.3 Customer Password Modify Message

A client may use the Modify Customer Password message to change the password. The new Customer Password must follow the password rules established in the <SignonInfo> aggregate, and must be encrypted if the Signon Realm requires it.

> *Note:* *This message will be deprecated in IFX 2.0, replaced by the Customer Authentication Modify (CustAuthMod) message.*

5.4.2.3.1 Request <CustPswdModRq>

Tag	Type	Usage	Description
<RqUID>	UUID	Required	Request Identifier. Optionally sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header aggregate.
<AsyncRqUID>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously. For more information, see section 3.2.12.1.
<CustId>	Aggregate *see section 3.2.3.1*	Optional	Customer ID.
<CustPswd>	Aggregate *see section 4.3.1*	Required	Customer Password Aggregate.

5.4.2.3.2 Response <CustPswdModRs>

The <CustPswdModRs> message provides the client with an acknowledgement that the Customer Password has been changed. If an error causes a client not to receive a response for a Modify Customer Password message, the server is in an indeterminate state and the client should attempt to establish a session with the new Customer Password. If that attempt fails, the client should attempt to establish a session with the old Customer Password.

Tag	Type	Usage	Description
<Status>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, <StatusCode> defaults to 0 (zero).
<RqUID>	UUID	Required	Request Identifier. Optionally sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header aggregate.
<MsgRsHdr>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header aggregate.
<AsyncRqUID>	UUID	Optional Echoed	Asynchronous Request Identifier.
<CustId>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer ID.
<ExpDt>	DateTime	Optional	Password Expiration date and time. If absent, the password never expires.
<SPRefId>	Identifier	Optional	Service Provider Reference Identifier.

5.4.2.4 Customer Authentication Modify Message

A client may use the Customer Authentication Modify message to change the authentication value (e.g. password, certificate, PIN). This may occur with a customer login or a CSR login, as a CSR may need to change the customer's password (e.g., when the customer has forgotten the password).

> *Note: This message replaces Customer Password Modify (CustPswdMod), which will be deprecated in IFX 2.0.*

5.4.2.4.1 Request <CustAuthModRq>

Tag	Type	Usage	Description
<RqUID>	UUID	Required	Request Identifier.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header aggregate.
<AsyncRqUID>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously. For more information, see section 3.2.12.1.

Tag	Type	Usage	Description
<RqUID>	UUID	Required	Request Identifier.
<CustId>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<CustPswd>	Aggregate *see section 4.3.1*	Required XOR	Customer Password Aggregate
<Certificate>	Binary	Required XOR	Certificate
<PINBlock>	NC-32	Required XOR	New PIN pad data
<CryptType>	Open Enum	Optional *but see desciption*	Encryption Type. <CryptType> will be required in IFX 2.0. Defined values: None, PKCS#1, PKCS#7, PKCS#10, ECB, CBC, CFB, OFB, TECB, TCBC, TCBC-I, TCFB, TCFB-P, TOFB, TOFB-I, AECB, ACBC, ACFB, AOFB, ACM Must be supported in list of encryption types in <SignonInfo> in the CSP's Service Profile. Note: If security objects are managed in band, this element should be included in order to know how to decrypt <PINBlock>
<SecObjId>	Identifier	Optional Repeating *but see Description*	Identifier of Security Object on which the <PINBlock> is encrypted. <SecObjId> will be required in IFX 2.0 Note: If security objects are managed in band, this element should be included in order to know how to decrypt <PINBlock>

5.4.2.4.2 Response <CustAuthModRs>

The <CustAuthModRs> message provides the client with an acknowledgement that the authentication has been changed. If an error causes a client not to receive a response for a Customer Authentication Modify message, the server is in an indeterminate state and the client should attempt to establish a session with the new authentication value. If that attempt fails, the client should attempt to establish a session with the old authentication value.

> *Note:* *The authentication credentials are not echoed in this message response in order to reduce the potential for authentication credentials to be stolen.*

Tag	Type	Usage	Description
<Status>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, <StatusCode> defaults to 0 (zero).
<RqUID>	UUID	Required Echoed	Request Identifier.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header aggregate.
<MsgRsHdr>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header aggregate.

Tag	Type	Usage	Description
\<Status\>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, \<StatusCode\> defaults to 0 (zero).
\<RqUID\>	UUID	Required Echoed	Request Identifier.
\<AsyncRqUID\>	UUID	Optional Echoed	Asynchronous Request Identifier.
\<CustId\>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer ID.
\<ExpDt\>	DateTime	Optional	Customer Authentication Expiration date and time. If absent, the authentication value never expires.
\<SPRefId\>	Identifier	Optional	Service Provider Reference Identifier.

5.4.2.5   Customer Authentication Reversal Message

5.4.2.5.1   Request \<CustAuthRevRq\>

Tag	Type	Usage	Description
\<RqUID\>	UUID	Required	Request Identifier.
\<MsgRqHdr\>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header aggregate.
\<AsyncRqUID\>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously. For more information, see section **3.2.12.1**.
\<CustId\>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in \<SignonRq\>. For example, if a CSR or SP issues the request on behalf of the user, then \<CustId\> is required, and must contain the value of the user whose request is being issued.
\<RevReasonCode\>	Open Enum	Required	Reversal reason code. The reason why the client is requesting a reversal.
\<Desc\>	C-80	Optional	Description, text describing the reason for the reversal.
\<CustAuthMsgRqInfo\>	Aggregate	Required	Customer Authentication Message Request Information Aggregate. This aggregate contains the message to reverse. See RevOptSupt in the service profile for specific reversal messages supported.
\<CustAuthModRq\>	Aggregate *see section 5.4.2.4.1*	Optional	Customer Authentication Modify Request Message Aggregate.
\</CustAuthMsgRqInfo\>			

5.4.2.5.2   Response \<CustAuthRevRs\>

Tag	Type	Usage	Description

Tag	Type	Usage	Description
\<Status\>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, \<StatusCode\> defaults to 0 (zero).
\<RqUID\>	UUID	Required Echoed	Request Identifier.
\<MsgRqHdr\>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header aggregate.
\<MsgRsHdr\>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header aggregate.
\<AsyncRqUID\>	UUID	Optional Echoed	Asynchronous Request Identifier.
\<CustId\>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Identification Aggregate.
\<RevReasonCode\>	Open Enum	Required Echoed	Reversal reason code. The reason why the client is requesting a reversal.
\<Desc\>	C-80	Optional Echoed	Description, text describing the reason for the reversal.
\<CustAuthMsgRqInfo\>	Aggregate	Required Echoed	Customer Authentication Message Request Information Aggregate. This aggregate contains the message to reverse. See RevOptSupt in the service profile for specific reversal messages supported.
\<CustAuthModRq\>	Aggregate *see section 5.4.2.4.1*	Optional	Customer Authentication Modify Request Message Aggregate.
\</CustAuthMsgRqInfo\>			
\<CSPRefId\>	Identifier	Optional	Customer Service Provider Reference Identifier.
\<SPRefId\>	Identifier	Optional	Service Provider Reference Identifier.

5.4.2.6   Customer Status Modify Message

The Customer Status Modify Message may be used to modify the status.

5.4.2.6.1   Request \<CustStatusModRq\>

Tag	Type	Usage	Description
\<RqUID\>	UUID	Required	Request Identifier.
\<MsgRqHdr\>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header aggregate.
\<AsyncRqUID\>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously. For more information, see section 3.2.12.1.

Tag	Type	Usage	Description
<CustId>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued. This is the identifier of the customer whose status is being modified.
<CustStatus>	Aggregate *see section 5.2.3.1.2*	Required	Customer Status Aggregate.

5.4.2.6.2   Response <CustStatusModRs>

Tag	Type	Usage	Description
<Status>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, <StatusCode> defaults to 0 (zero).
<RqUID>	UUID	Required Echoed	Request Identifier.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header aggregate.
<MsgRsHdr>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header aggregate.
<AsyncRqUID>	UUID	Optional Echoed	Asynchronous Request Identifier.
<CustId>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<CustStatus>	Aggregate *see section 5.2.3.1.2*	Required Echoed	Customer Status Aggregate.
<CustRec>	Aggregate *see section 5.2.3.1*	Required	Customer Record Aggregate.

5.4.2.7   Customer Delete Message

A client may use the Customer Delete message to unenroll.

5.4.2.7.1   Request <CustDelRq>

Tag	Type	Usage	Description
<RqUID>	UUID	Required	Request Identifier. Optionally sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header aggregate.

Tag	Type	Usage	Description
\<AsyncRqUID\>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously. For more information, see section 3.2.12.1.
\<CustId\>	Aggregate *see section 3.2.3.1*	Optional	Customer ID.
\<CascadeDel\>	Boolean	Optional	Cascade Delete. If *True*, server must delete all dependent objects when this object is deleted. If *False* or omitted, the customer/service link must not be deleted if dependent objects exist. For a service, a dependent object is a service/account link, pending transactions (transfers or payments), recurring models, bills or payees, depending on the service.

5.4.2.7.2   Response \<CustDelRs\>

Tag	Type	Usage	Description
\<Status\>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, \<StatusCode\> defaults to 0 (zero).
\<RqUID\>	UUID	Required Echoed	Request Identifier.
\<MsgRqHdr\>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header aggregate.
\<MsgRsHdr\>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header aggregate.
\<AsyncRqUID\>	UUID	Optional Echoed	Asynchronous Request Identifier.
\<CustId\>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer ID.
\<CascadeDel\>	Boolean	Optional Echoed	Cascade Delete.
\<CustRec\>	Aggregate *see section 5.2.3.1*	Optional XOR	Customer Record Aggregate. This aggregate is provided in cases where the server keeps the customer's record on the server even after receiving a deletion request. This may occur if the deletion is not immediate, or if the server maintains the record with a status of *Unenrolled*.
\<DependentType\>	Open Enum	Optional XOR Repeating	Dependent object type that exists for the customer/service link. One element must be returned for each dependent. Defined values: CustSvc
\<SPRefId\>	Identifier	Optional	Service Provider Reference Identifier.

5.4.2.8   Customer Inquiry Message

If \<UpDt\> is equal the server's date of last update, a \<Status\> code of 1 is returned, and the profile is not returned in the response. This is the newest \<CustRec\> \<UpDt\> received by the client. If this information is absent, or does not match the server's date of last update, the server must send all data in the response.

5.4.2.8.1   Request <CustInqRq>

Tag	Type	Usage	Description
<RqUID>	UUID	Required	Request Identifier. Optionally sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header aggregate.
<AsyncRqUID>	UUID	Optional Echoed	Asynchronous Request Identifier.
<CustId>	Aggregate *see section 3.2.3.1*	Optional Repeating	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<CardMagData>	Aggregate *see section 3.2.7.1.3.1*	Optional Repeating	Card Magnetic Stripe Data aggregate. This is the card data identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if an ATM issues the request on behalf of the user, then <CardMagData> is required, and must contain the value of the user whose request is being issued.
<DeliveryMethod>	Open Enum	Optional Profiled values	Delivery Method. Default is Channel. Value must be supported in Service Profile.
<UpDt>	Timestamp	Optional	Customer Record Update DateTime. The value is for the last update to <CustRec> on the server that is known to the client.
<IncToken>	Boolean	Optional	Include Token. If *True*, a <Token> should be included in the response to set a base for future Service/Account Sync messages. If *False* or omitted, no <Token> is returned.

5.4.2.8.2   Response <CustInqRs>

Tag	Type	Usage	Description
<Status>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, <StatusCode> defaults to 0 (zero).
<RqUID>	UUID	Required Echoed	Request Identifier. Optionally sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header aggregate.
<MsgRsHdr>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header aggregate.
<AsyncRqUID>	UUID	Optional Echoed	Asynchronous Request Identifier.

Tag	Type	Usage	Description
<CustId>	Aggregate *see section 3.2.3.1*	Optional Repeating Echoed	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<CardMagData>	Aggregate *see section 3.2.7.1.3.1*	Optional Repeating Echoed	Card Magnetic Stripe Data aggregate.
<DeliveryMethod>	Open Enum	Optional Echoed	Delivery Method.
<UpDt>	Timestamp	Optional Echoed	Customer Record Update DateTime.
<IncToken>	Boolean	Optional Echoed	Include Token. If *True*, a <Token> should be included in the response to set a base for future Service/Account Sync messages. If *False* or omitted, no <Token> is returned.
<NewUpDt>	Timestamp	Required	New Customer Record Update Date Time. The value is for the last update to <CustRec> on the Server.
<CustRec>	Aggregate *see section 5.2.3.1*	Optional Repeating	Customer Record Aggregate. One record is returned for each customer matching the selection criteria in the request.
<Token>	Identifier	Optional	Token. Server Assigned. Client should make no assumptions about the value of this token relative to others that it may have received. The token is significant only to the server that originally assigned it.

5.4.2.9   Customer Identifier Inquiry Message

A client, typically a CSR, may send a request to the Server to retrieve either the Login Id corresponding to a Permanent Id, or the Permanent Id corresponding to a Login Id.

5.4.2.9.1   Request <CustIdInqRq>

The client must specify either the Customer Permanent identifier, or the Customer Login identifier.

Tag	Type	Usage	Description
<RqUID>	UUID	Required	Request Identifier. Optionally sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header aggregate.
<AsyncRqUID>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously. For more information, see section 3.2.12.1.
<CustPermId>	Identifier	Required OR	Customer Permanent ID. Used as a database key to uniquely identify a Service Provider's customer. Cannot be changed by the customer.
<CustLoginId>	Identifier	Required OR	Customer Login ID. Used as a user-friendly name for the customer for authentication purposes. Maps directly to Customer Permanent ID. Some implementations may allow a user to change his or her Login ID.

5.4.2.9.2 Response <CustIdInqRs>

The SP Server responds with a code indicating the processing status and the client identifiers.

Tag	Type	Usage	Description
<Status>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, <StatusCode> defaults to 0 (zero).
<RqUID>	UUID	Required Echoed	Request Identifier. Optionally sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header aggregate.
<MsgRsHdr>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header aggregate.
<AsyncRqUID>	UUID	Optional Echoed	Asynchronous Request Identifier.
<CustId>	Aggregate *see section 3.2.3.1*	Required	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued. **Note:** *in cases where there is a <Status> <Severity> of* Error*, <CustId> is* not *required in the response. This allows for graceful failures.*

5.4.2.10   Customer Audit Message

The Customer Audit Message allows the client to audit Customer Add/Modify/Delete messages. When the Cust object changes, the server must generate an Rs message to the Rq that created the pending state.

5.4.2.10.1 Request <CustAudRq>

Tag	Type	Usage	Description
<RqUID>	UUID	Required	Request Identifier. Optionally sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header aggregate.
<AsyncRqUID>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously. For more information, see section 3.2.12.1.
<CustId>	Aggregate *see section 3.2.3.1*	Optional Repeating	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.

Tag	Type	Usage	Description
\<SPName\>	Identifier	Optional	Service Provider Name.
\<RecCtrlIn\>	Aggregate *see section 3.2.12.2.1*	Optional	Records Control Input Aggregate.
\<SelRangeDt\>	Aggregate *see section 3.2.10.1*	Optional	Selection Range Date Time Aggregate.
\<Method\>	Closed Enum	Optional Repeating	Customer/Service Link Method. This field is used as a selection criterion.
			Valid values: Add, Mod, Del.

5.4.2.10.2 Response \<CustAudRs\>

Tag	Type	Usage	Description
\<Status\>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, \<StatusCode\> defaults to 0 (zero).
\<RqUID\>	UUID	Required Echoed	Request Identifier. Optionally sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.
\<MsgRqHdr\>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header aggregate.
\<MsgRsHdr\>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header aggregate.
\<AsyncRqUID\>	UUID	Optional Echoed	Asynchronous Request Identifier.
\<CustId\>	Aggregate *see section 3.2.3.1*	Optional Repeating Echoed	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in \<SignonRq\>. For example, if a CSR or SP issues the request on behalf of the user, then \<CustId\> is required, and must contain the value of the user whose request is being issued.
\<SPName\>	Identifier	Optional Echoed	Service Provider Name.
\<RecCtrlOut\>	Aggregate *see section 3.2.12.2.2*	Optional *but see Description*	Records Control Output Aggregate. Required if client included \<RecCtrlIn\> in request and the server supports Records Control.
\<SelRangeDt\>	Aggregate *see section 3.2.10.1*	Optional Echoed	Selection Range Date Aggregate.
\<Method\>	Closed Enum	Optional Repeating Echoed	Customer/Service Link Method. This field is used as a selection criterion.
\<CustMsgRec\>	Aggregate	Optional Repeating	Customer Message Record

Tag	Type	Usage	Description
<CustId>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the entity who initiated the request (e.g., the one whose identifier appears in the <SignonRq> of the original issued request). For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the CSR or SP.
<MsgRecDt>	DateTime	Optional	Audit Record Creation Date. Date/time at which the audit record was stored/created by the service provider.
<CustAddRs>	Aggregate *see section 5.4.2.1.2*	Required XOR	Customer Add Response Record Aggregate. One record is returned for each Customer Add message for this customer, subject to selection criteria or token.
<CustModRs>	Aggregate *see section 5.4.2.2.2*	Required XOR	Customer Modify Response Record Aggregate. One record is returned for each Customer Modify message for this customer, subject to selection criteria or token.
<CustStatusModRs>	Aggregate *see section 5.4.2.6.2*	Required XOR	Customer Status Modify Response Record Aggregate. One record is returned for each Customer Status Modify message for this customer, subject to selection criteria or token.
<CustDelRs>	Aggregate *see section 5.4.2.7.2*	Required XOR	Customer Delete Response Record Aggregate. One record is returned for each Customer Delete message for this customer, subject to selection criteria or token.
</CustMsgRec>			

5.4.2.11 Customer Sync Message

The Customer Sync Message allows the client to synchronize on Customer Add/Modify/Delete messages. When the Cust object changes, the server must generate an Rs message to the Rq that created the pending state.

5.4.2.11.1 Request <CustSyncRq>

Tag	Type	Usage	Description
<RqUID>	UUID	Required	Request Identifier. Optionally sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header aggregate.
<AsyncRqUID>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously. For more information, see section 3.2.12.1.
<CustId>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<RecCtrlIn>	Aggregate *see section 3.2.12.2.1*	Optional	Records Control Input Aggregate.

Tag	Type	Usage	Description
<Token>	Identifier	Required	Token. Server assigned. Client should make no assumptions about the value of this token relative to others it may have received. The token is only significant to the server that originally assigned it.
			This is a token that has been previously sent by the server, or zero for first time requests.

5.4.2.11.2 Response <CustSyncRs>

Tag	Type	Usage	Description
<Status>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, <StatusCode> defaults to 0 (zero).
<RqUID>	UUID	Required	Request Identifier. Optionally sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header aggregate.
<MsgRsHdr>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header aggregate.
<AsyncRqUID>	UUID	Optional	Asynchronous Request Identifier.
<CustId>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<RecCtrlOut>	Aggregate *see section 3.2.12.2.2*	Optional *but see Description*	Records Control Output Aggregate. Required if client included <RecCtrlIn> in request and server supports Records Control.
<Token>	Identifier	Required Echoed	Token.
<NewToken>	Identifier	Required	New Token. Server assigned. Client should make no assumptions about the value of this token relative to others it may have received. The token is only significant to the server that originally assigned it.
<CustMsgRec>	Aggregate	Optional Repeating	Customer Message Record
<CustId>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the entity who initiated the request (e.g., the one whose identifier appears in the <SignonRq> of the original issued request). For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the CSR or SP.
<MsgRecDt>	DateTime	Optional	Audit Record Creation Date. Date/time at which the audit record was stored/created by the service provider.
<CustAddRs>	Aggregate *see section 5.4.2.1.2*	Required XOR	Customer Add Response Record Aggregate. One record is returned for each Customer Add message for this customer, subject to selection criteria or token.

<CustModRs>	Aggregate *see section 5.4.2.2.2*	Required XOR	Customer Modify Response Record Aggregate. One record is returned for each Customer Modify message for this customer, subject to selection criteria or token.
<CustStatusModRs>	Aggregate *see section 5.4.2.6.2*	Required XOR	Customer Status Modify Response Record Aggregate. One record is returned for each Customer Status Modify message for this customer, subject to selection criteria or token.
<CustDelRs>	Aggregate *see section 5.4.2.7.2*	Required XOR	Customer Delete Response Record Aggregate. One record is returned for each Customer Delete message for this customer, subject to selection criteria or token.
</CustMsgRec>			

5.4.3   *Customer/Service Link*

5.4.3.1   **Customer/Service Link Add**

5.4.3.1.1   *Request <CustSvcAddRq>*

A client uses the <CustSvcAddRq> message to request that a service be enabled. If the <SvcName> value is Pay, then this message indicates enabling the Pay service, and should be sent from the CSP to the CPP; likewise, if the <SvcName> is Pres, then it should be sent to a BSP. The fee account should be passed to Service Providers who are actually charging for the service.

Tag	Type	Usage	Description
<RqUID>	UUID	Required	Request Identifier. Optionally sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header aggregate.
<AsyncRqUID>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously. For more information, see section 3.2.12.1.
<CustId>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<CustSvcId>	Aggregate *see section 5.2.3.4.1*	Required	Customer/Service Link Identification Aggregate
<CustSvcInfo>	Aggregate *see section 5.2.3.4.2*	Required	Customer/Service Link Information Aggregate

5.4.3.1.2   *Response <CustSvcAddRs>*

The Customer/Service Link Add Response Message acknowledges receipt of the enabling request. It also optionally allows the SP to return disclosures associated with the activation request.

> *Note:* *<CustSvcAddRs> may be returned before enabling a service if several steps must be performed in enabling a service. Therefore, a <CustSvcStatusCode> of xxxPending and the Resolve Pending Date may be returned to the CSP and/or Customer to provide an estimated date for completion of the request. This may be used, for example, when a CPP does a prenotification for a fee account, since it might take several days to obtain results.*

Tag	Type	Usage	Description
<Status>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, <StatusCode> defaults to 0 (zero).
<RqUID>	UUID	Required Echoed	Request Identifier.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header aggregate.
<MsgRsHdr>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header aggregate.
<AsyncRqUID>	UUID	Optional Echoed	Asynchronous Request Identifier.
<CustId>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<CustSvcId>	Aggregate *see section 5.2.3.4.1*	Required Echoed	Customer/Service Link Identification Aggregate.
<CustSvcInfo>	Aggregate *see section 5.2.3.4.2*	Required Echoed	Customer/Service Link Information Aggregate.
<CustSvcRec>	Aggregate *see section 5.2.3.4*	Required	Customer/Service Record Aggregate
<DiscRec>	Aggregate *see section 5.2.3.2*	Optional Repeating	Disclosure Record Aggregate.
<CSPRefId>	Identifier	Optional	Customer Service Provider Reference Identifier.
<SPRefId>	Identifier	Optional	Service Provider Reference Identifier.

5.4.3.2   Customer/Service Link Modify

The <CustSvcModRq> allows a customer to modify specific information about a customer/service link. This capability is currently limited to modifying the fee account.

5.4.3.2.1   Request <CustSvcModRq>

Tag	Type	Usage	Description
<RqUID>	UUID	Required	Request Identifier. Optionally sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header aggregate.

Tag	Type	Usage	Description
<AsyncRqUID>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously.
			For more information, see section 3.2.12.1.
<CustId>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<CustSvcId>	Aggregate *see section 5.2.3.4.1*	Required	Customer/Service Link Identification Aggregate.
<CustSvcInfo>	Aggregate *see section 5.2.3.4.2*	Required	Customer/Service Link Information Aggregate

5.4.3.2.2   Response <CustSvcModRs>

Tag	Type	Usage	Description
<Status>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, <StatusCode> defaults to 0 (zero).
<RqUID>	UUID	Required Echoed	Request Identifier.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header aggregate.
<MsgRsHdr>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header aggregate.
<AsyncRqUID>	UUID	Optional Echoed	Asynchronous Request Identifier.
<CustId>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<CustSvcId>	Aggregate *see section 5.2.3.4.1*	Required Echoed	Customer/Service Link Identification Aggregate.
<CustSvcInfo>	Aggregate *see section 5.2.3.4.2*	Required Echoed	Customer/Service Link Information Aggregate.
<CustSvcRec>	Aggregate *see section 5.2.3.4*	Required	Customer/Service Link Record Aggregate
<DiscRec>	Aggregate *see section 5.2.3.2*	Optional Repeating	Disclosure Record Aggregate.

Tag	Type	Usage	Description
<CSPRefId>	Identifier	Optional	Customer Service Provider Reference Identifier.
<SPRefId>	Identifier	Optional	Service Provider Reference Identifier.

5.4.3.3   Customer/Service Link Status Modify

The <CustSvcStatusModRq> allows a Service Provider to temporarily change the customer's service status. For example, this function is used in the Presentment Service to notify a BSP that the CSP is unable to deliver bills or other material to a customer and to notify the BSP that reverse sthe notification. This may occur when there is a potential security/fraud problem or when a technical problem precludes the customer's access to his/her account with the CSP. The BSP may need to notify the Biller to send paper bills, depending on regulatory or business practices, to ensure the customer's billing is not interrupted.

5.4.3.3.1   Request <CustSvcStatusModRq>

Tag	Type	Usage	Description
<RqUID>	UUID	Required	Request Identifier. Optionally sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header aggregate.
<AsyncRqUID>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously.
			For more information, see section 3.2.12.1.
<CustId>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<CustSvcId>	Aggregate *see section 5.2.3.4.1*	Required	Customer/Service Link Identification Aggregate
<CustSvcStatus>	Aggregate *see section 5.2.3.4.3*	Required	Customer/Service Link Status Aggregate

5.4.3.3.2   Response <CustSvcStatusModRs>

Tag	Type	Usage	Description
<Status>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, <StatusCode> defaults to 0 (zero).
<RqUID>	UUID	Required Echoed	Request Identifier.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header aggregate.
<MsgRsHdr>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header aggregate.

Tag	Type	Usage	Description
<AsyncRqUID>	UUID	Optional Echoed	Asynchronous Request Identifier.
<CustId>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<CustSvcId>	Aggregate *see section 5.2.3.4.1*	Required Echoed	Customer/Service Link Identification Aggregate.
<CustSvcStatus>	Aggregate *see section 5.2.3.4.3*	Required Echoed	Customer/Service Link Status Aggregate
<CustSvcRec>	Aggregate *see section 5.2.3.4*	Required	Customer/Service Link Record Aggregate
<CSPRefId>	Identifier	Optional	Customer Service Provider Reference Identifier.
<SPRefId>	Identifier	Optional	Service Provider Reference Identifier.

5.4.3.4   Customer/Service Link Delete

5.4.3.4.1   Request <CustSvcDelRq>

The <CustSvcDelRq> allows a customer to disable a service offered by a specified service provider. A customer may initiate the process by sending the <CustSvcDelRq> to the CSP. The CSP may then send a <CustSvcDelRq> message to a BSP to disable, for example, Bill Presentment Service for a particular customer at the BSP.

Tag	Type	Usage	Description
<RqUID>	UUID	Required	Request Identifier. Optionally sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header aggregate.
<AsyncRqUID>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously. For more information, see section 3.2.12.1.
<CustId>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<CustSvcId>	Aggregate *see section 5.2.3.4.1*	Required	Customer/Service Identification Aggregate

Tag	Type	Usage	Description
<CascadeDel>	Boolean	Optional	Cascade Delete. If *True*, server must delete all dependent objects when this object is deleted. If *False* or omitted, the customer/service link must not be deleted if dependent objects exist. For a service, a dependent object is a service/account link, pending transactions (transfers or payments), recurring models, bills or payees, depending on the service.

5.4.3.4.2 Response <CustSvcDelRs>

Tag	Type	Usage	Description
<Status>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, <StatusCode> defaults to 0 (zero).
<RqUID>	UUID	Required Echoed	Request Identifier.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header aggregate.
<MsgRsHdr>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header aggregate.
<AsyncRqUID>	UUID	Optional Echoed	Asynchronous Request Identifier.
<CustId>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<CustSvcId>	Aggregate *see section 5.2.3.4.1*	Required Echoed	Customer/Service Identification Aggregate
<CascadeDel>	Boolean	Optional Echoed	Cascade Delete.
<CustSvcRec>	Aggregate *see section 5.2.3.4*	Optional XOR	Customer/Service Link Record Aggregate. This aggregate is provided in cases where the server keeps the customer/service link's record on the server even after receiving a deletion request. This may occur if the deletion is not immediate, or if the server maintains the record with a status of Disabled.
<DependentType>	Open Enum	Optional XOR Repeating	Dependent object type that exists for the customer/service link. One element must be returned for each dependent. Defined values: SvcAcct, Bill, CustPayee, Pmt, RecPmt, RecXfer, Xfer
<CSPRefId>	Identifier	Optional	Customer Service Provider Reference Identifier.
<SPRefId>	Identifier	Optional	Service Provider Reference Identifier.

5.4.3.5   Customer/Service Link Audit

Allows client to audit Customer/Service Link Add/Modify/Delete messages associated with the current customer. When the <CustSvcStatus> changes, the server must generate an Rs message corresponding to the Rq

that created the pending state. The <Status> <Severity> must always be Info. The <StatusDesc> is used to supply the details of a rejection.

5.4.3.5.1 Request <CustSvcAudRq>

Tag	Type	Usage	Description
<RqUID>	UUID	Required	Request Identifier. Sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header aggregate.
<AsyncRqUID>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously. For more information, see section 3.2.12.1.
<CustId>	Aggregate *see section 3.2.3.1*	Optional Repeating	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<RecCtrlIn>	Aggregate *see section 3.2.12.2.1*	Optional	Records Control Input Aggregate.
<SelRangeDt>	Aggregate *see section 3.2.10.1*	Optional	Selection Range Date Time Aggregate.
<Method>	Closed Enum	Optional Repeating	Customer/Service Link Method. This field is used as a selection criterion. Valid values: Add, Mod, Del.
<SvcName>	Open Enum	Optional Repeating	Service Name Valid values: Bank, Pay, Pres.
<SPName>	Identifier	Optional Repeating	Service Provider Name.

5.4.3.5.2 Response <CustSvcAudRs>

Tag	Type	Usage	Description
<Status>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, <StatusCode> defaults to 0 (zero).
<RqUID>	UUID	Required Echoed	Request Identifier. Sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header aggregate.
<MsgRsHdr>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header aggregate.

Tag	Type	Usage	Description
\<AsyncRqUID\>	UUID	Optional Echoed	Asynchronous Request Identifier.
\<CustId\>	Aggregate *see section 3.2.3.1*	Optional Repeating Echoed	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in \<SignonRq\>. For example, if a CSR or SP issues the request on behalf of the user, then \<CustId\> is required, and must contain the value of the user whose request is being issued.
\<RecCtrlOut\>	Aggregate *see section 3.2.12.2.2*	Optional *but see Description*	Records Control Output Aggregate. Required if client included \<RecCtrlIn\> in request and server supports Records Control.
\<SelRangeDt\>	Aggregate *see section 3.2.10.1*	Optional Echoed	Selection Range Date Aggregate.
\<Method\>	Closed Enum	Optional Repeating Echoed	Customer/Service Link Action. Valid values: Add, Mod, Del
\<SvcName\>	Open Enum	Optional Repeating Echoed	Service Name Valid values: Bank, Pay, Pres
\<SPName\>	Identifier	Optional Repeating Echoed	Service Provider Name.
\<CustSvcMsgRec\>	Aggregate	Optional Repeating	Customer/Service Message Record.
\<CustId\>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the entity who initiated the request (e.g., the one whose identifier appears in the \<SignonRq\> of the original issued request). For example, if a CSR or SP issues the request on behalf of the user, then \<CustId\> is required, and must contain the value of the CSR or SP.
\<MsgRecDt\>	DateTime	Optional	Audit Record Creation Date. Date/time at which the audit record was stored/created by the service provider.
\<CustSvcAddRs\>	Aggregate *see section 5.4.3.1.2*	Required XOR	Customer/Service Link Add Response Record Aggregate. One record is returned for each Customer/Service Link Add message for this customer, subject to selection criteria or token.
\<CustSvcModRs\>	Aggregate *see section 5.4.3.2.2*	Required XOR	Customer/Service Link Modify Response Record Aggregate. One record is returned for each Customer/Service Link Modify message for this customer, subject to selection criteria or token.
\<CustSvcStatusModRs\>	Aggregate *see section 5.4.3.3.2*	Required XOR	Customer/Service Link Status Modify Response Record Aggregate. One record is returned for each Customer/Service Link Status Modify message for this customer, subject to selection criteria or token.
\<CustSvcDelRs\>	Aggregate *see section 5.4.3.4.2*	Required XOR	Customer/Service Link Delete Response Record Aggregate. One record is returned for each Customer/Service Link Delete message for this customer, subject to selection criteria or token.
\</CustSvcMsgRec\>			

5.4.3.6 Customer/Service Link Sync

Allows client to synchronize Customer/Service Link Add/Modify/Delete messages associated with the current customer. When the <CustSvcStatus> changes, the server must generate an Rs message corresponding to the Rq that created the pending state. The <Status> <Severity> must always be Info. The <StatusDesc> is used to supply the details of a rejection.

5.4.3.6.1 Request <CustSvcSyncRq>

Tag	Type	Usage	Description
<RqUID>	UUID	Required	Request Identifier. Sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header aggregate.
<AsyncRqUID>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously.
			For more information, see section 3.2.12.1.
<CustId>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<RecCtrlIn>	Aggregate *see section 3.2.12.2.1*	Optional	Records Control Input Aggregate.
<Token>	Identifier	Required	Token. Server assigned. Client should make no assumptions about the value of this token relative to others it may have received. The token is only significant to the server that originally assigned it.
			This is a token that has been previously sent by the server, or zero for first time requests.

5.4.3.6.2 Response <CustSvcSyncRs>

Tag	Type	Usage	Description
<Status>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, <StatusCode> defaults to 0 (zero).
<RqUID>	UUID	Required Echoed	Request Identifier. Sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header aggregate.
<MsgRsHdr>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header aggregate.

Tag	Type	Usage	Description
<AsyncRqUID>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously.
			For more information, see section 3.2.12.1.
<CustId>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<RecCtrlOut>	Aggregate *see section 3.2.12.2.2*	Optional *but see Description*	Records Control Output Aggregate. Required if client included <RecCtrlIn> in request and server supports Records Control.
<Token>	Identifier	Required Echoed	Token.
<NewToken>	Identifier	Required	New Token. Server assigned. Client should make no assumptions about the value of this token relative to others it may have received. The token is only significant to the server that originally assigned it.
<CustSvcMsgRec>	Aggregate	Optional Repeating	Customer/Service Message Record.
<CustId>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the entity who initiated the request (e.g., the one whose identifier appears in the <SignonRq> of the original issued request). For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the CSR or SP.
<MsgRecDt>	DateTime	Optional	Audit Record Creation Date. Date/time at which the audit record was stored/created by the service provider.
<CustSvcAddRs>	Aggregate *see section 5.4.3.1.2*	Required XOR	Customer/Service Link Add Response Record Aggregate. One record is returned for each Customer/Service Link Add message for this customer, subject to selection criteria or token.
<CustSvcModRs>	Aggregate *see section 5.4.3.2.2*	Required XOR	Customer/Service Link Modify Response Record Aggregate. One record is returned for each Customer/Service Link Modify message for this customer, subject to selection criteria or token.
<CustSvcStatusModRs>	Aggregate *see section 5.4.3.3.2*	Required XOR	Customer/Service Link Status Modify Response Record Aggregate. One record is returned for each Customer/Service Link Status Modify message for this customer, subject to selection criteria or token.
<CustSvcDelRs>	Aggregate *see section 5.4.3.4.2*	Required XOR	Customer/Service Link Delete Response Record Aggregate. One record is returned for each Customer/Service Link Delete message for this customer, subject to selection criteria or token.
</CustSvcMsgRec>			

5.4.4 Service/Account Link

5.4.4.1 Bill Presentment

When activating an account for Bill Presentment Service, the customer and/or CSP must send secrets as requested by the Biller, names and addresses as known to the Biller to be associated with the account, and the customer account with the Biller <BillingAcct>, in order for a BSP or a Biller to correctly identify which customer and account to activate. The Customer name and address at the CSP may be used to assist a BSP or Biller in matching the correct customer with the account to be activated.

5.4.4.2 Account specific disclosures

Note that if the Service Provider (e.g., CSP, CPP, or BSP) has account-specific disclosures that were not displayed in the service activation process, the Service Provider must send disclosure text with the <SvcAcctAddRs> when an account is being activated.

If presentation of terms and conditions is required prior to account activation, Disclosure Record Aggregates must be returned in the response. If acceptance is required, and the customer does not accept, the account activation may not be sent to the BSP. If the customer does accept, the client may use <CustDiscStatusModRq>.

5.4.4.3 Service/Account Link Add Message

5.4.4.3.1 Request <SvcAcctAddRq>

A client uses the <SvcAcctAddRq> message to request that a particular account be activated for an enabled service, e.g., a long distance telephone account for a bill presentment service. Multiple accounts may be activated by sending a separate <SvcAcctAddRq> for each account to be activated.

Tag	Type	Usage	Description
<RqUID>	UUID	Required	Request Identifier. Sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header aggregate.
<AsyncRqUID>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously. For more information, see section 3.2.12.1.
<CustId>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<SvcAcctId>	Aggregate *see section 5.2.3.5.1*	Required	Service/Account Link Identification Aggregate
<SvcAcctInfo>	Aggregate *see section 5.2.3.5.2*	Required	Service/Account Link Information Aggregate
<SecretList>	Aggregate *see section 3.2.14.2*	Optional Repeating AND	Secret List. This is the list of responses to <SecretPrompt> provided for account activation in aggregates such as <BillerRec>in Bill Presentment service.

Tag	Type	Usage	Description
\<CryptType\>	Open Enum	Optional Profiled values AND	Encryption Type. Indicates the encryption algorithm used for the secrets within the \<SecretList\>. Must be one of the supported types such as the ones defined in the Biller directory entry \<BillerRec\> for Bill Presentment service. Defined values: None, PKCS#1, PKCS#7, PKCS#10, ECB, CBC, CFB, OFB, TECB, TCBC, TCBC-I, TCFB, TCFB-P, TOFB, TOFB-I, AECB, ACBC, ACFB, AOFB, ACM
\<SecObjId\>	Identifier	Optional Repeating AND	Identifier of Security Object on which the \<SecretList\> is encrypted Note: If security objects are managed in band, this element should be included in order to know how to decrypt \<CryptPswd\>
\<CustNameAddr\>	Aggregate *see section 3.2.3.4*	Optional XOR Repeating	Customer Name/Address aggregate. This may be required for account activation, such as when adding a presentment account.
\<PersonInfo\>	Aggregate *see section 3.2.4.1*	Optional XOR Repeating	Person Information Aggregate.

5.4.4.3.2   Response \<SvcAcctAddRs\>

The Activate Account Response Message acknowledges receipt of the activate request and contains as much status information as is available.

> *Note:*   *When a CSP activates an account on behalf of a customer, the xSP may return a status description of PendAdd. Upon subsequent Service/Account Link Audit Requests \<SvcAcctAudRq\> and Service/Account Link Sync Requests \<SvcAcctSyncRq\>, the Effective Date may be filled in when the account is activated. The \<EffDt\> is advisory information coming from the xSP.*

Tag	Type	Usage	Description
\<Status\>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, \<StatusCode\> defaults to 0 (zero).
\<RqUID\>	UUID	Required Echoed	Request Identifier.
\<MsgRqHdr\>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header aggregate.
\<MsgRsHdr\>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header aggregate.
\<AsyncRqUID\>	UUID	Optional Echoed	Asynchronous Request Identifier.
\<CustId\>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in \<SignonRq\>. For example, if a CSR or SP issues the request on behalf of the user, then \<CustId\> is required, and must contain the value of the user whose request is being issued.
\<SvcAcctId\>	Aggregate *see section 5.2.3.5.1*	Required Echoed	Service/Account Link Identification Aggregate.

Tag	Type	Usage	Description
\<SvcAcctInfo>	Aggregate *see section 5.2.3.5.2*	Required Echoed	Service/Account Link Information Aggregate.
\<SvcAcctRec>	Aggregate *see section 5.2.3.5*	Required	Service/Account Record Aggregate.
\<DiscRec>	Aggregate *see section 5.2.3.2*	Optional Repeating	Disclosure Record Aggregate.
\<CSPRefId>	Identifier	Optional	Customer Service Provider Reference Identifier.
\<SPRefId>	Identifier	Optional	Service Provider Reference Identifier.

5.4.4.4   Service/Account Link Modify Message

5.4.4.4.1   Request \<SvcAcctModRq>

The \<SvcAcctModRq> allows a customer to modify specific information about an account. This message is currently limited to modifying (1) a Pay funding account, and (2) the account Nickname.

Tag	Type	Usage	Description
\<RqUID>	UUID	Required	Request Identifier.
\<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header aggregate.
\<AsyncRqUID>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously. For more information, see section 3.2.12.1.
\<CustId>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in \<SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then \<CustId> is required, and must contain the value of the user whose request is being issued.
\<SvcAcctId>	Aggregate *see section 5.2.3.5.1*	Required	Service/Account Link Identifier.
\<SvcAcctInfo>	Aggregate *see section 5.2.3.5.2*	Required	Service/Account Link Information Aggregate.
\<SecretList>	Aggregate *see section 3.2.14.2*	Optional Repeating AND	Secret List. This is the list of responses to \<SecretPrompt> provided for account activation in aggregates such as \<BillerRec>in Bill Presentment service.
\<CryptType>	Open Enum	Optional Profiled values AND	Encryption Type. Indicates the encryption algorithm used for the secrets within the \<SecretList>. Must be one of the supported types such as the ones defined in the Biller directory entry \<BillerRec> for Bill Presentment service. Defined values: None, PKCS#1, PKCS#7, PKCS#10, ECB, CBC, CFB, OFB, TECB, TCBC, TCBC-I, TCFB, TCFB-P, TOFB, TOFB-I, AECB, ACBC, ACFB, AOFB, ACM

Tag	Type	Usage	Description
<SecObjId>	Identifier	Optional Repeating AND	Identifier of Security Object on which the <CryptPswd> is encrypted Note: If security objects are managed in band, this element should be included in order to know how to decrypt <CryptPswd>
<CustNameAddr>	Aggregate *see section 3.2.3.4*	Optional XOR Repeating	Customer Name/Address aggregate. This may be required for account modification, such as with a presentment account.
<PersonInfo>	Aggregate *see section 3.2.4.1*	Optional XOR Repeating	Person Information Aggregate.

5.4.4.4.2   Response <SvcAcctModRs>

Tag	Type	Usage	Description
<Status>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, <StatusCode> defaults to 0 (zero).
<RqUID>	UUID	Required Echoed	Request Identifier.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header aggregate.
<MsgRsHdr>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header aggregate.
<AsyncRqUID>	UUID	Optional Echoed	Asynchronous Request Identifier.
<CustId>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<SvcAcctId>	Aggregate *see section 5.2.3.5.1*	Required Echoed	Service/Account Link Identification Aggregate.
<SvcAcctInfo>	Aggregate *see section 5.2.3.5.2*	Required Echoed	Service/Account Link Information Aggregate.
<SvcAcctRec>	Aggregate *see section 5.2.3.5*	Required	Service/Account Record Aggregate.
<CustDiscRec>	Aggregate *see section 5.2.3.3*	Optional Repeating	Disclosure Record Aggregate.
<CSPRefId>	Identifier	Optional	Financial Institution Reference Number.
<SPRefId>	Identifier	Optional	Service Provider Reference Number.

5.4.4.5 Service/Account Link Status Modify

The <SvcAcctStatusModRq> allows a Service Provider to temporarily change the customer's service/account link status.

5.4.4.5.1 Request <SvcAcctStatusModRq>

Tag	Type	Usage	Description
<RqUID>	UUID	Required	Request Identifier. Optionally sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header aggregate.
<AsyncRqUID>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously. For more information, see section 3.2.12.1.
<CustId>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<SvcAcctId>	Aggregate *see section 5.2.3.5.1*	Required	Service/Account Link Identification Aggregate
<SvcAcctStatus>	Aggregate *see section 5.2.3.5.3*	Required	Service/Account Link Status Aggregate

5.4.4.5.2 Response <SvcAcctStatusModRs>

Tag	Type	Usage	Description
<Status>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, <StatusCode> defaults to 0 (zero).
<RqUID>	UUID	Required Echoed	Request Identifier.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header aggregate.
<MsgRsHdr>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header aggregate.
<AsyncRqUID>	UUID	Optional Echoed	Asynchronous Request Identifier.
<CustId>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.

Tag	Type	Usage	Description
<SvcAcctId>	Aggregate *see section 5.2.3.5.1*	Required Echoed	Service/Account Link Identification Aggregate.
<SvcAcctStatus>	Aggregate *see section 5.2.3.5.3*	Required Echoed	Service/Account Link Status Aggregate
<SvcAcctRec>	Aggregate *see section 5.2.3.5*	Required	Service/Account Link Record Aggregate
<CSPRefId>	Identifier	Optional	Customer Service Provider Reference Identifier.
<SPRefId>	Identifier	Optional	Service Provider Reference Identifier.

5.4.4.6   Service/Account Link Identifier Modify Message

5.4.4.6.1   Request <SvcAcctIdModRq>

The <SvcAcctIdModRq> allows a customer or CSP to change the identifier associated with a particular Service Account Link. For example, in the Pay service, the <SvcAcctIdModRq> would change all pending payments against a particular funding account to be applied against a new funding account.

Tag	Type	Usage	Description
<RqUID>	UUID	Required	Request Identifier. Sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header aggregate.
<AsyncRqUID>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously. For more information, see section 3.2.12.1.
<CustId>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<SvcAcctId>	Aggregate *see section 5.2.3.5.1*	Required	Service/Account Link Identifier. This is the identifier of the currently linked account to the service.
<DepAcctId>	Aggregate *see section 3.2.7.1.2*	Required XOR	Deposit Account Identifier. This is the identifier of the new account to be linked.
<CardAcctId>	Aggregate *see section 3.2.7.1.3*	Required XOR	Card Account Identifier. This is the identifier of the new account to be linked.
<LoanAcctId>	Aggregate *see section 3.2.7.1.4*	Required XOR	Loan Account Identifier. This is the identifier of the new account to be linked.
<PresAcctId>	Aggregate *see section 3.2.7.2.1*	Required XOR	Presentment Account Identifier. This is the identifier of the new account to be linked.

Tag	Type	Usage	Description
\<PresAcctInfo\>	Aggregate *see section 3.2.7.2.2*	Optional	Presentment Account Information.

5.4.4.6.2 Response \<SvcAcctIdModRs\>

Tag	Type	Usage	Description
\<Status\>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate.
\<RqUID\>	UUID	Required Echoed	Request Identifier.
\<MsgRqHdr\>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header aggregate.
\<MsgRsHdr\>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header aggregate.
\<AsyncRqUID\>	UUID	Optional Echoed	Asynchronous Request Identifier.
\<CustId\>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Identification Aggregate.
\<SvcAcctId\>	Aggregate *see section 5.2.3.5.1*	Required Echoed	Service/Account Link Identifier.
\<DepAcctId\>	Aggregate *see section 3.2.7.1.2*	Required XOR Echoed	Deposit Account Identifier.
\<CardAcctId\>	Aggregate *see section 3.2.7.1.3*	Required XOR Echoed	Card Account Identifier.
\<LoanAcctId\>	Aggregate *see section 3.2.7.1.4*	Required XOR Echoed	Loan Account Identifier.
\<PresAcctId\>	Aggregate *see section 3.2.7.2.1*	Required XOR Echoed	Presentment Account Identifier.
\<PresAcctInfo\>	Aggregate *see section 3.2.7.2.2*	Optional Echoed	Presentment Account Information.
\<SvcAcctRec\>	Aggregate *see section 5.2.3.5*	Required	Service/Account Link Record.

5.4.4.7 Service/Account Link Delete Message

5.4.4.7.1 Request \<SvcAcctDelRq\>

The \<SvcAcctDelRq\> allows a customer or CSP to deactivate a specific account for the specified service. Note that deactivating an account in IFX refers to a service provider removing the linkage between the account and the service, and does not impact the status of the actual account, e.g. suspending an electronic billing account should not be confused with halting the actual service, such as turning off the electricity.

Tag	Type	Usage	Description
\<RqUID\>	UUID	Required	Request Identifier. Sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.
\<MsgRqHdr\>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header aggregate.
\<AsyncRqUID\>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously. For more information, see section 3.2.12.1.
\<CustId\>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in \<SignonRq\>. For example, if a CSR or SP issues the request on behalf of the user, then \<CustId\> is required, and must contain the value of the user whose request is being issued.
\<SvcAcctId\>	Aggregate *see section 5.2.3.5.1*	Required	Service/Account Link Identification Aggregate
\<CascadeDel\>	Boolean	Optional	Cascade Delete. If *True*, server must delete all dependent objects when this object is deleted. If *False* or omitted, the service/account link must not be deleted if dependent objects exist. For a service/account link, a dependent object is pending transactions (transfers or payments), recurring models, bills or payees, depending on the service.

5.4.4.7.2   Response \<SvcAcctDelRs\>

Tag	Type	Usage	Description
\<Status\>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, \<StatusCode\> defaults to 0 (zero).
\<RqUID\>	UUID	Required Echoed	Request Identifier.
\<MsgRqHdr\>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header aggregate.
\<MsgRsHdr\>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header aggregate.
\<AsyncRqUID\>	UUID	Optional Echoed	Asynchronous Request Identifier.
\<CustId\>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in \<SignonRq\>. For example, if a CSR or SP issues the request on behalf of the user, then \<CustId\> is required, and must contain the value of the user whose request is being issued.
\<SvcAcctId\>	Aggregate *see section 5.2.3.5.1*	Required Echoed	Service/Account Link Identification Aggregate.

Tag	Type	Usage	Description
<CascadeDel>	Boolean	Optional Echoed	Cascade Delete.
<SvcAcctRec>	Aggregate *see section 5.2.3.5*	Optional XOR	Service/Account Link Record Aggregate. This aggregate is provided in cases where the server keeps the service/account link's record on the server even after receiving a deletion request. This may occur if the deletion is not immediate, or if the server maintains the record with a status of Deactivated.
<DependentType>	Open Enum	Optional XOR Repeating	Dependent object type that exists for the customer/service link. One element must be returned for each dependent. Defined values: SvcAcct, Bill, CustPayee, Pmt, RecPmt, RecXfer, Xfer
<CSPRefId>	Identifier	Optional	Customer Service Provider Reference Identifier.
<SPRefId>	Identifier	Optional	Service Provider Reference Identifier.

5.4.4.8   Service/Account Link Inquiry Message

If <UpDt> is equal the server's date of last update, a <Status> code of 1 is returned, and the profile is not returned in the response. This is the last <NewUpDt> received by the client. If this information is absent, or does not match the server's date of last update, the server must send all data in the response.

5.4.4.8.1   Request <SvcAcctInqRq>

Tag	Type	Usage	Description
<RqUID>	UUID	Required	Request Identifier. Sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header aggregate.
<AsyncRqUID>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously. For more information, see section 3.2.12.1.
<CustId>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<CardMagData>	Aggregate *see section 3.2.7.1.3.1*	Optional	Card Magnetic Stripe Data aggregate. This is the card data identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if an ATM issues the request on behalf of the user, then <CardMagData> is required, and must contain the value of the user whose request is being issued.
<RecCtrlIn>	Aggregate *see section 3.2.12.2.1*	Optional	Records Control Input Aggregate.

Tag	Type	Usage	Description
<SvcName>	Open Enum	Optional Repeating	Service Name. This field identifies the specific set of services for which the server should provide customer profile information. This allows the client to scope the profile request to provide account information only for a specific set of services.
			Defined values: Bank, Pay, Pres
<IncDetail>	Boolean	Optional	Include Detail Indicator. If *True*, the response should include the detail account records, <xxxAcctRec>, in the aggregates returned. If *False*, or omitted, the detail account records should not be included.
<IncBal>	Boolean	Optional	Include Balances Indicator. If *True*, the <AcctBal> aggregate within <BankAcctRec> is being requested. If *False* or omitted, the <AcctBal> aggregate within <BankAcctRec> is not being requested. This indicator is only meaningful whenever <IncDetail> is *True*.
<IncToken>	Boolean	Optional	Include Token. If *True*, a <Token> should be included in the response to set a base for future Service/Account Sync messages. If *False* or omitted, no <Token> is returned.
<DeliveryMethod>	Open Enum	Optional Profiled values	Delivery Method. Default is Channel.
			Value must be supported in Service Profile.
<UpDt>	Timestamp	Optional	Service/Account Information Update Timestamp. The value is for the last (i.e., latest) update made to either <CustSvcRec>or <SvcAcctRec> on the server that is known to the client.

5.4.4.8.2 Response <SvcAcctInqRs>

Tag	Type	Usage	Description
<Status>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, <StatusCode> defaults to 0 (zero).
<RqUID>	UUID	Required Echoed	Request Identifier. Sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header aggregate.
<MsgRsHdr>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header aggregate.
<AsyncRqUID>	UUID	Optional Echoed	Asynchronous Request Identifier.
<CustId>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<CardMagData>	Aggregate *see section 3.2.7.1.3.1*	Optional Echoed	Card Magnetic Stripe Data aggregate.

Tag	Type	Usage	Description
<RecCtrlOut>	Aggregate *see section 3.2.12.2.2*	Optional *but see Description*	Records Control Output Aggregate. Required if <RecCtrlIn> was provided in the request and the server supports Records Control.
<SvcName>	Open Enum	Optional Repeating Echoed	Service Name. Defined values: Bank, Pay, Pres
<IncDetail>	Boolean	Optional Echoed	Include Detail Indicator. If *True*, the response should include the detail account records, <xxxAcctRec>, in the aggregates returned. If *False*, or omitted, the detail account records should not be included.
<IncBal>	Boolean	Optional Echoed	Include Balances Indicator. If *True*, the <AcctBal> aggregate within <BankAcctRec> is being requested. If *False* or omitted, the <AcctBal> aggregate within <BankAcctRec> is not being requested.
<IncToken>	Boolean	Optional Echoed	Include Token. If *True*, a <Token> should be included in the response to set a base for future Service/Account Sync messages. If *False* or omitted, no <Token> is returned.
<DeliveryMethod>	Open Enum	Optional Echoed	Delivery Method.
<UpDt>	Timestamp	Optional Echoed	Service Information Update DateTime.
<NewUpDt>	Timestamp	Required	New Service/Account Link Update DateTime. The value is for the last (i.e., latest) update made to either <CustSvcRec>or <SvcAcctRec> on the server.
<*xxx*AcctRec>	Aggregate	Optional Repeating	Various Account Record Aggregates, as defined in Section 3.2.7. (e.g., <BankAcctRec> and/or <PresAcctRec>)
<CustSvcRec>	Aggregate *see section 5.2.3.4*	Optional Repeating	Customer/Service Link Record Aggregate. This aggregate contains the status of a specific service and, if applicable, fee account information associated with the service.
<SvcAcctRec>	Aggregate *see section 5.2.3.5*	Optional Repeating	Service/ Account Link Record Aggregate. This aggregate identifies an account and its status with respect to a specific service.
<Token>	Identifier	Optional	Token. Server Assigned. Client should make no assumptions about the value of this token relative to others that it may have received. The token is significant only to the server that originally assigned it.

5.4.4.9 Service/Account Link Audit Message

Allows client to audit Service/Account Link Add/Modify/Delete messages associated with the current customer and the current customer's accounts. When the <SvcAcctStatus> changes, the server must generate an Rs message corresponding to the Rq that created the pending state. The <Status> <Severity> must always be Info. The <StatusDesc> is used to supply the details of a rejection.

5.4.4.9.1 Request <SvcAcctAudRq>

Tag	Type	Usage	Description
<RqUID>	UUID	Required	Request Identifier. Sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header aggregate.

Tag	Type	Usage	Description
\<AsyncRqUID\>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously. For more information, see section 3.2.12.1.
\<CustId\>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in \<SignonRq\>. For example, if a CSR or SP issues the request on behalf of the user, then \<CustId\> is required, and must contain the value of the user whose request is being issued.
\<RecCtrlIn\>	Aggregate *see section 3.2.12.2.1*	Optional	Records Control Input Aggregate.
\<SelRangeDt\>	Aggregate *see section 3.2.10.1*	Optional	Selection Range Date Time Aggregate.
\<Method\>	Closed Enum	Optional Repeating	Service/Account Link Method.This field is used as a selection criterion. Valid values: Add, Mod, StatusMod, Del

5.4.4.9.2   Response \<SvcAcctAudRs\>

Tag	Type	Usage	Description
\<Status\>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, \<StatusCode\> defaults to 0 (zero).
\<RqUID\>	UUID	Required Echoed	Request Identifier. Sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.
\<MsgRqHdr\>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header aggregate.
\<MsgRsHdr\>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header aggregate.
\<AsyncRqUID\>	UUID	Optional Echoed	Asynchronous Request Identifier.
\<CustId\>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in \<SignonRq\>. For example, if a CSR or SP issues the request on behalf of the user, then \<CustId\> is required, and must contain the value of the user whose request is being issued.
\<RecCtrlOut\>	Aggregate *see section 3.2.12.2.2*	Optional *but see Description*	Records Control Output Aggregate. Required if client included \<RecCtrlIn\> in request and server supports Records Control.
\<SelRangeDt\>	Aggregate *see section 3.2.10.1*	Optional Echoed	Selection Range Date Aggregate.

Tag	Type	Usage	Description
<Method>	Closed Enum	Optional Repeating Echoed	Service/Account Link Method.
<SvcAcctMsgRec>	Aggregate	Optional Repeating	Service/Account Link Message Record.
<CustId>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the entity who initiated the request (e.g., the one whose identifier appears in the <SignonRq> of the original issued request). For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the CSR or SP.
<MsgRecDt>	Timestamp	Optional	Audit Record Creation Timestamp. Timestamp at which the audit record was stored/created by the service provider.
<SvcAcctAddRs>	Aggregate *see section 5.4.4.3.2*	Required XOR	Service/Account Link Add Response Record Aggregate. One record is returned for each Service/Account Link Add message for this customer, subject to selection criteria or token.
<SvcAcctModRs>	Aggregate *see section 5.4.4.4.2*	Required XOR	Service/Account Link Modify Response Record Aggregate. One record is returned for each Service/Account Link Modify message for this customer, subject to selection criteria or token.
<SvcAcctStatusModRs>	Aggregate *see section 5.4.4.5.2*	Required XOR	Service/Account Link Status Modify Response Record Aggregate. One record is returned for each Service/Account Link Status Modify message for this customer, subject to selection criteria or token.
<SvcAcctIdModRs>	Aggregate *see section 5.4.4.6.2*	Required XOR 1.1+	Service/Account Link Identifier Modify Response Record Aggregate. One record is returned for each Service/Account Link Identifier Modify message for this customer, subject to selection criteria or token.
<SvcAcctDelRs>	Aggregate *see section 5.4.4.7.2*	Required XOR	Service/Account Link Delete Response Record Aggregate. One record is returned for each Service/Account Link Delete message for this customer, subject to selection criteria or token.
<SvcAcctRevRs>	Aggregate *see section 5.4.4.11.2*	Required XOR	Service/Account Link Reveral Response Record Aggregate. One record is returned for each Service/Account Link Reversal message for this customer, subject to selection criteria or token.
</SvcAcctMsgRec>			

5.4.4.10 Service/Account Link Sync Message

Allows client to synchronize Service/Account Link Add/Modify/Delete messages associated with the current customer and the current customer's accounts. When the <SvcAcctStatus> changes, the server must generate an Rs message to the Rq that created the pending state. The <Status> <Severity> must be Info. The <StatusDesc> is used to supply the details of a rejection.

5.4.4.10.1 Request <SvcAcctSyncRq>

Tag	Type	Usage	Description
<RqUID>	UUID	Required	Request Identifier. Sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.

Tag	Type	Usage	Description
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header aggregate.
<AsyncRqUID>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously.
			For more information, see section 3.2.12.1.
<CustId>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<RecCtrlIn>	Aggregate *see section 3.2.12.2.1*	Optional	Records Control Input Aggregate.
<Token>	Identifier	Required	Token. Server assigned. Client should make no assumptions about the value of this token relative to others it may have received. The token is only significant to the server that originally assigned it.
			This is a token that has been previously sent by the server, or zero for first time requests.

5.4.4.10.2 Response <SvcAcctSyncRs>

Tag	Type	Usage	Description
<Status>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, <StatusCode> defaults to 0 (zero).
<RqUID>	UUID	Required Echoed	Request Identifier. Sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header aggregate.
<MsgRsHdr>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header aggregate.
<AsyncRqUID>	UUID	Optional Echoed	Asynchronous Request Identifier.
<CustId>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<RecCtrlOut>	Aggregate *see section 3.2.12.2.2*	Optional *but see Description*	Records Control Output Aggregate.
			Required if client included <RecCtrlIn> in request and server supports Records Control.
<Token>	Identifier	Required Echoed	Token.

Tag	Type	Usage	Description
\<NewToken\>	Identifier	Required	New Token. Server assigned. Client should make no assumptions about the value of this token relative to others it may have received. The token is only significant to the server that originally assigned it.
\<SvcAcctMsgRec\>	Aggregate	Optional Repeating	
\<CustId\>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the entity who initiated the request (e.g., the one whose identifier appears in the \<SignonRq\> of the original issued request). For example, if a CSR or SP issues the request on behalf of the user, then \<CustId\> is required, and must contain the value of the CSR or SP.
\<MsgRecDt\>	DateTime	Optional	Audit Record Creation Date. Date/time at which the audit record was stored/created by the service provider.
\<SvcAcctAddRs\>	Aggregate *see section 5.4.4.3.2*	Required XOR	Service/Account Link Add Response Record Aggregate. One record is returned for each Service/Account Link Add message for this customer, subject to selection criteria or token.
\<SvcAcctModRs\>	Aggregate *see section 5.4.4.4.2*	Required XOR	Service/Account Link Modify Response Record Aggregate. One record is returned for each Service/Account Link Modify message for this customer, subject to selection criteria or token.
\<SvcAcctStatusModRs\>	Aggregate *see section 5.4.4.5.2*	Required XOR	Service/Account Link Status Modify Response Record Aggregate. One record is returned for each Service/Account Link Status Modify message for this customer, subject to selection criteria or token.
\<SvcAcctIdModRs\>	Aggregate *see section 5.4.4.6.2*	Required XOR 1.1+	Service/Account Link Identifier Modify Response Record Aggregate. One record is returned for each Service/Account Link Identifier Modify message for this customer, subject to selection criteria or token.
\<SvcAcctDelRs\>	Aggregate *see section 5.4.4.7.2*	Required XOR	Service/Account Link Delete Response Record Aggregate. One record is returned for each Service/Account Link Delete message for this customer, subject to selection criteria or token.
\</SvcAcctMsgRec\>			

5.4.4.11 Service Account Reversal

5.4.4.11.1 Request \<SvcAcctRevRq\>

Tag	Type	Usage	Description
\<RqUID\>	UUID	Required	Request Identifier.
\<MsgRqHdr\>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header aggregate.
\<AsyncRqUID\>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously. For more information, see section **3.2.12.1**.

Tag	Type	Usage	Description
<CustId>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<RevReasonCode>	Open Enum	Required	Reversal reason code. The reason why the client is requesting a reversal.
<Desc>	C-80	Optional	Description, text describing the reason for the reversal.
<SvcAcctMsgRqInfo>	Aggregate	Required	Service Account Message Request Information Aggregate. This aggregate contains the message to reverse. See RevOptSupt in the service profile for specific reversal messages supported.
<SvcAcctAddRq>	Aggregate *see section 5.4.4.3.1*	Required XOR	Service Account Add Request Message Aggregate.
<SvcAcctModRq>	Aggregate *see section 5.4.4.4.1*	Required XOR	Service Account Modify Request Message Aggregate.
<SvcAcctStatusModRq>	Aggregate *see section 5.4.4.5.1*	Required XOR	Service Account Stautus Modify Request Message Aggregate.
<SvcAcctDelRq>	Aggregate *see section 5.4.4.7.1*	Required XOR	Service Account Delete Request Message Aggregate.
<SvcAcctInqRq>	Aggregate *see section 5.4.4.8.1*	Required XOR	Service Account Inquiry Request Message Aggregate.
<SvcAcctAudRq>	Aggregate *see section 5.4.4.9.1*	Required XOR	Service Account Audit Request Message Aggregate.
<SvcAcctSyncRq>	Aggregate *see section 5.4.4.10.1*	Required XOR	Service Account Synchronization Request Message Aggregate.
</SvcAcctMsgRqInfo>			

5.4.4.11.2 Response <SvcAcctRevRs>

Tag	Type	Usage	Description
<Status>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, <StatusCode> defaults to 0 (zero).
<RqUID>	UUID	Required Echoed	Request Identifier.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header aggregate.
<MsgRsHdr>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header aggregate.
<AsyncRqUID>	UUID	Optional Echoed	Asynchronous Request Identifier.

Tag	Type	Usage	Description
<CustId>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Identification Aggregate.
<RevReasonCode>	Open Enum	Required Echoed	Reversal reason code. The reason why the client is requesting a reversal.
<Desc>	C-80	Optional Echoed	Description, text describing the reason for the reversal.
<SvcAcctMsgRqInfo>	Aggregate	Required Echoed	Service Account Message Request Information Aggregate. This aggregate contains the message to reverse. See RevOptSupt in the service profile for specific reversal messages supported.
<SvcAcctAddRq>	Aggregate *see section 5.4.4.3.1*	Required Echoed XOR	Service Account Add Request Message Aggregate.
<SvcAcctModRq>	Aggregate *see section 5.4.4.4.1*	Required Echoed XOR	Service Account Modify Request Message Aggregate.
<SvcAcctStatusModRq>	Aggregate *see section 5.4.4.5.1*	Required Echoed XOR	Service Account Stautus Modify Request Message Aggregate.
<SvcAcctDelRq>	Aggregate *see section 5.4.4.7.1*	Required Echoed XOR	Service Account Delete Request Message Aggregate.
<SvcAcctInqRq>	Aggregate *see section 5.4.4.8.1*	Required Echoed XOR	Service Account Inquiry Request Message Aggregate.
<SvcAcctAudRq>	Aggregate *see section 5.4.4.9.1*	Required Echoed XOR	Service Account Audit Request Message Aggregate.
<SvcAcctSyncRq>	Aggregate *see section 5.4.4.10.1*	Required Echoed XOR	Service Account Synchronization Request Message Aggregate.
</SvcAcctMsgRqInfo>			
<CSPRefId>	Identifier	Optional	Customer Service Provider Reference Identifier.
<SPRefId>	Identifier	Optional	Service Provider Reference Identifier.

5.4.5   Disclosure

5.4.5.1   Disclosure Inquiry Message

A customer or CSP may make a query about disclosure terms and conditions for a given service.

5.4.5.1.1   Request <DiscInqRq>

Tag	Type	Usage	Description
<RqUID>	UUID	Required	Request Identifier.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header aggregate.

Tag	Type	Usage	Description
\<AsyncRqUID>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously.
			For more information, see section 3.2.12.1.
\<CustId>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in \<SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then \<CustId> is required, and must contain the value of the user whose request is being issued.
\<SvcName>	Open Enum	Optional Repeating	Service Name.
			Defined values: Bank, Pay, Pres
			This element is used as a selection criterion.
\<DiscId>	UUID	Optional Repeating	Disclosure ID. This element is used as a selection criterion.
\<DeliveryMethod>	Open Enum	Optional Profiled values	Delivery Method. Default is Channel.
			Value must be supported in Service Profile.
\<UpDt>	Timestamp	Optional	Timestamp when the Disclosure list was Last Updated on the Server.
			This is the timestamp of the last Disclosure list received from the server.

5.4.5.1.2   Response \<DiscInqRs>

The SP Server responds with disclosure details.

Tag	Type	Usage	Description
\<Status>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, \<StatusCode> defaults to 0 (zero).
\<RqUID>	UUID	Required Echoed	Request Identifier.
\<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header aggregate.
\<MsgRsHdr>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header aggregate.
\<AsyncRqUID>	UUID	Optional Echoed	Asynchronous Request Identifier.
\<CustId>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in \<SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then \<CustId> is required, and must contain the value of the user whose request is being issued.
\<SvcName>	Open Enum	Optional Repeating Echoed	Service Name.

Tag	Type	Usage	Description
<DiscId>	UUID	Optional Repeating Echoed	Disclosure ID. This element is used as a selection criterion.
<DeliveryMethod>	Open Enum	Optional Echoed	Delivery Method.
<UpDt>	Timestamp	Optional Echoed	Timestamp when the Disclosure list was last updated on the Server. The last timestamp received by the client.
<NewUpDt>	Timestamp	Required	New Disclosure list update timestamp. The value is for the last update to the most recent <DiscRec> on the Server.
<DiscRec>	Aggregate *see section 5.2.3.2*	Optional Repeating	Disclosure Record Aggregate. One aggregate is returned for each record matching the selection criteria in the request.

5.4.6   Customer/Disclosure Link

5.4.6.1   Customer/Disclosure Link Status Modify Message

A customer uses the Customer/Disclosure Link Status Modify Message to accept or reject a disclosure received via various messages.

5.4.6.1.1   Request <CustDiscStatusModRq>

Tag	Type	Usage	Description
<RqUID>	UUID	Required	Request Identifier. Sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header aggregate.
<AsyncRqUID>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously. For more information, see section 3.2.12.1.
<CustId>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<DiscId>	UUID	Required	Disclosure Identifier.
<CustDiscStatusCode>	Closed Enum	Required	Customer/Disclosure Link Status Code. Valid values: None, Accepted, Rejected, Viewed

5.4.6.1.2   Response <CustDiscStatusModRs>

Tag	Type	Usage	Description
<Status>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, <StatusCode> defaults to 0 (zero).

Tag	Type	Usage	Description
<RqUID>	UUID	Required Echoed	Request Identifier.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header aggregate.
<MsgRsHdr>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header aggregate.
<AsyncRqUID>	UUID	Optional Echoed	Asynchronous Request Identifier.
<CustId>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<DiscId>	UUID	Required Echoed	Disclosure Identifier.
<CustDiscStatusCode>	Closed Enum	Required Echoed	Customer/Disclosure Link Status Code
<CSPRefId>	Identifier	Optional	Customer Service Provider Reference Identifier.
<SPRefId>	Identifier	Optional	Service Provider Reference Identifier.

5.4.6.2   Customer/Disclosure Link Inquiry Message

A customer or CSP may make a query about the status of a disclosure for a given consumer, and optionally include the associated disclosure itself.

5.4.6.2.1   Request <CustDiscInqRq>

Tag	Type	Usage	Description
<RqUID>	UUID	Required	Request Identifier.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header aggregate.
<AsyncRqUID>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously. For more information, see section 3.2.12.1.
<CustId>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<SvcName>	Open Enum	Required	Service Name. Defined values: Bank, Pay, Pres
<DiscId>	UUID	Optional	Disclosure ID. This element is used as a selection criterion.

Tag	Type	Usage	Description
<CustDiscStatusCode>	Closed Enum	Optional	Customer/Disclosure Link Status Code. This element is used as a selection criterion
			Valid values: None, Accepted, Rejected, Viewed
<IncDisc>	Boolean	Optional	Include Disclosure Boolean. If *True*, the client is requesting that <DiscInfo> be included with each record in the response. If *False* or omitted, the client is requesting that <DiscInfo> *not* be included with each record in the response.
<DeliveryMethod>	Open Enum	Optional Profiled values	Delivery Method. Default is Channel.
			Value must be supported in Service Profile.

5.4.6.2.2   Response <CustDiscInqRs>

The SP Server responds with disclosure details.

Tag	Type	Usage	Description
<Status>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, <StatusCode> defaults to 0 (zero).
<RqUID>	UUID	Required Echoed	Request Identifier.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header aggregate.
<MsgRsHdr>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header aggregate.
<AsyncRqUID>	UUID	Optional Echoed	Asynchronous Request Identifier.
<CustId>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<SvcName>	Open Enum	Required Echoed	Service Name.
<DiscId>	UUID	Optional Echoed	Disclosure ID. This element is used as a selection criterion.
<CustDiscStatusCode>	Closed Enum	Optional Echoed	Customer/Disclosure Link Status Code.
<IncDisc>	Boolean	Optional Echoed	Include Disclosure Boolean.
<DeliveryMethod>	Open Enum	Optional Echoed	Delivery Method.
<CustDiscRec>	Aggregate *see section 5.2.3.3*	Optional Repeating	Customer/Disclosure Record Aggregate. One aggregate is returned for each record matching the selection criteria in the request. <CustDiscRec> <DiscInfo> may also be returned if <IncDisc> is *True*.

5.5 Security Object

The Security Object messages are intended to allow security objects to be transported for use within IFX transactions. Security objects include items such as DES/3DES keys used for PIN encryption, Digital Certificates used for identification and public keys used for data encryption using asynchronous encryption techniques.

A unique feature of the security object is the requirement for the object to physically exist at each end-point involved in the encryption process. Unlike a transaction (withdrawal, bill pay, etc) or customer information for which the system of record is in a single location, security objects must exist and be tracked in more than one location and be kept in sync.

Note: None of the security object messages contain any of the IFX asynchronous element (AsyncRqUID). This is due to the fact that the exchange of security object information is expected to occur in a synchronous manner.

5.5.1 Security Object Add

The Security Object Add message is used to create an instance of a security object.

5.5.1.1 Request <SecObjAddRq>

Tag	Type	Usage	Description
<RqUId>	UUID	Required	Request Identifier.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header Aggregate
<SecObjInfo>	Aggregate *see section 5.2.3.6.1*	Required	Security Object Information Aggregate

5.5.1.2 Response <SecObjAddRs>

> *Note:* *The encryption keys located in <SecObjInfo> are not echoed in this message response in order to reduce the potential for the encryption keys to be stolen.*

Tag	Type	Usage	Description
<Status>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, <StatusCode> defaults to 0 (zero).
<RqUId>	UUID	Required Echoed	Request Identifier.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header Aggregate
<MsgRsHdr>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header Aggregate
<SecObjInfo>	Aggregate *see section 5.2.3.6.1*	Required Echoed	Security Object Information Aggregate
<SecObjRec>	Aggregate *see section 5.2.3.6*	Required	Security Object Record Aggregate
<CSPRefId>	Identifier	Optional	Customer Service Provider Reference Identifier
<SPRefId>	Identifier	Optional	Service Provider Reference Identifier

5.5.2 *Security Object Modify*

The Security Object Modify message provides a mechanism to modify a particular instance of a security object.

5.5.2.1 Request <SecObjModRq>

Tag	Type	Usage	Description
<RqUID>	UUID	Required	Request Identifier.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header Aggregate
<SecObjId>	Identifier	Required	Security Object Identifier of security object to be modified
<SecObjInfo>	Aggregate *see section 5.2.3.6.1*	Required	Security Object Information Aggregate

5.5.2.2 Response <SecObjModRs>

> *Note:* *The encryption keys located in <SecObjInfo> are not echoed in this message response in order to reduce the potential for the encryption keys to be stolen.*

Tag	Type	Usage	Description
<Status>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, <StatusCode> defaults to 0 (zero).
<RqUID>	UUID	Required Echoed	Request Identifier.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header Aggregate
<MsgRsHdr>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header Aggregate
<SecObjInfo>	Aggregate *see section 5.2.3.6.1*	Required Echoed	Security Object Information Aggregate
<SecObjRec>	Aggregate *see section 5.2.3.6*	Required	Security Object Record Aggregate
<CSPRefId>	Identifier	Optional	Customer Service Provider Reference Identifier
<SPRefId>	Identifier	Optional	Service Provider Reference Identifier

5.5.3 *Security Object Inquiry*

This message is sent to request information on a particular security object. Information returned is based on the type of security object. For symmetric keys, the Key Check information is returned. For Public Keys or Digital certificate, the actual Public Key or Digital Certificate is returned.

5.5.3.1 Request <SecObjInqRq>

Tag	Type	Usage	Description
<RqUId>	UUID	Required	Request Identifier.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header Aggregate

Tag	Type	Usage	Description
<SecObjId>	Identifier	Optional	Security Object Identifier of security object information is requested on.
<SecObjName>	C-64	Optional	Security Object Identifier used to reference this security object in future request
<SecObjPurpose>	Open Enum	Optional	Purpose (use) of the Security Object
			Note: The Algorithm that employs this key is agreed to out of band.
			Valid Values: PINEncryptingKey, KeyEncryptingKey, PasswordEncryptingKey, DataEncryptingKey, Identification, MAC, Uniqueness, VISAPINVerification
<SecObjType>	Open Enum	Optional	Type of Security Object.
			Valid Value: 1KeyDES, 2KeyDES, 3KeyDES, 128BitAES, 192BitAES, 256BitAES, DigitalCertificate, PublicKey, RandomNumber, ModuleID
<CSPRefId>	Identifier	Optional	Customer Service Provider Reference Identifier of the object information is requested on
<SPRefId>	Identifier	Optional	Service Provider Reference Identifier of the object information is requested on.
<RecCtrlIn>	Aggregate *see section 3.2.12.2.1*	Optional	Record control input aggregate
<SecObjStatusCode>	Closed Enum	Optional Repeating	Returns all security objects that match the status
			Valid Values: Active, Inactive, Expired, OutofSync

5.5.3.2   Response <SecObjInqRs>

Tag	Type	Usage	Description
<Status>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, <StatusCode> defaults to 0 (zero).
<RqUID>	UUID	Required Echoed	Request Identifier.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header Aggregate
<MsgRsHdr>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header Aggregate
<SecObjRec>	Aggregate *see section 5.2.3.6*	Required Repeating	Security Object Record Aggregate
<CSPRefId>	Identifier	Optional Echoed	Customer Service Provider Reference Identifier
<SPRefId>	Identifier	Optional Echoed	Service Provider Reference Identifier

5.5.4   *Security Object Delete*

This message is sent to delete a particular instance of a security object.

5.5.4.1   Request <SecObjDelRq>

Tag	Type	Usage	Description
<RqUID>	UUID	Required	Request Identifier.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header Aggregate
<SecObjId>	Identifier	Required	Security Object Identifier of security object to be modified

5.5.4.2   Response <SecObjDelRs>

Tag	Type	Usage	Description
<Status>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, <StatusCode> defaults to 0 (zero).
<RqUID>	UUID	Required Echoed	Request Identifier.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header Aggregate
<MsgRsHdr>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header Aggregate
<CSPRefId>	Identifier	Optional	Customer Service Provider Reference Identifier
<SPRefId>	Identifier	Optional	Service Provider Reference Identifier

5.5.5   *Security Object Advise*

The Advise method can be used as a means of either triggering the remote key load process in an ATM or POS System or as a means of providing the server with information on a security object that was loaded into the client via an out of band method (manually loaded, injected from another system, etc.).

This method is sent from the Server to the ATM with the <SecObjInitRemKey> tag set to "True" to request the client to initiate the Remote Key Load process. If the ATM is in a state that would allow it to initiate a remote key load process, the Advise will be responded to with a status code of "0" (OK). If the device is not in a state to initiate a remote key load process, then the Advise will be responded to with an information status code of 3760 and will initiate Remote Key Load process as soon as the state of the device will allow. If the device does not support Remote Key Load, then an error code of 3770 will be returned.

If a security object is loaded into the client via an out-of-band method, the client can notify the server of the security object's <SecObjId> as well as the object's characteristics via the inclusion of the <SecObjRec> aggregate.

5.5.5.1   Request <SecObjAdviseRq>

Tag	Type	Usage	Description
<RqUID>	UUID	Required	Request Identifier.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header Aggregate
<SecObjRec>	Aggregate *see section 5.2.3.6*	Required XOR	Information on security object applied to message initiator out-of-band

Tag	Type	Usage	Description
<SecObjInitRemKey>	Boolean	Required XOR	Request for recipient to initiate a Remote Key Load session

Valid Values: True, False |

5.5.5.2   Response <SecObjAdviseRs>

Tag	Type	Usage	Description
<Status>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, <StatusCode> defaults to 0 (zero).
<RqUID>	UUID	Required Echoed	Request Identifier.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header Aggregate
<MsgRsHdr>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header Aggregate
<SecObjRec>	Aggregate *see section 5.2.3.6*	Required Echoed XOR	Information on security object applied to message initiator out-of-band
<SecObjInitRemKey>	Boolean	Required XOR Echoed	Request for recipient to initiate a Remote Key Load session

Valid Values: True, False |

5.5.6   *Security Object Sync*

The sync method can be used as a means of synchronization of the security objects known by either party. Note that actual security objects are not sent as part of this exchange unless the security object is not of a sensitive nature (Module ID, Public Key, etc.). If during this process, it is discovered that one party is missing a security object, an appropriate Security Object Add request should be issued.

5.5.6.1   Request <SecObjSyncRq>

Tag	Type	Usage	Description
<RqUID>	UUID	Required	Request Identifier. Sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header Aggregate
<SecObjId>	Identifier	Optional Repeating	Optional list of security IDs that the issuer is aware of

Note: This is not a section criteria for the synchronization |
| **<RecCtrlIn>** | Aggregate *see section 3.2.12.2.1* | Optional | Records Control Input Aggregate. |
| **<Token>** | Identifier | Required | Token. Server assigned. Client should make no assumptions about the value of this token relative to others it may have received. The token is only significant to the server that originally assigned it.

This is a token that has been previously sent by the server, or zero for first time requests. |

5.5.6.2 Response <SecObjSyncRs>

Tag	Type	Usage	Description
<Status>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, <StatusCode> defaults to 0 (zero).
<RqUID>	UUID	Required Echoed	Request Identifier. Sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header Aggregate
<MsgRsHdr>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header Aggregate
<SecObjId>	Identifier	Optional Repeating Echoed	Optional list of security IDs that the issuer is aware of

Note: This is not a section criteria for the synchronization |
| **<RecCtrlOut>** | Aggregate *see section 3.2.12.2.2* | Optional *but see Description* | Records Control Output Aggregate.

Required if client included <RecCtrlIn> in request and server supports Records Control. |
<Token>	Identifier	Required Echoed	Token.
<NewToken>	Identifier	Required	New Token. Server assigned. Client should make no assumptions about the value of this token relative to others it may have received. The token is only significant to the server that originally assigned it.
<SecObjRec>	Aggregate *see section 5.2.3.6*	Optional Repeating	Optional security object record that contains information on security objects that the server is aware of

5.6 Party to Account Relationship

The party to account relationship messages are used to associate a party (i.e., customer, prospect) to and account. This association could be as an owner or non-owner, and is also used to indicate the primary contact and tax responsible parties.

5.6.1 Party to Account Relationship Add

This message is used to add a relationship between a pre-existing party (i.e., customer or prospect) and a pre-existing account. This message may be used to add additional "owners" to an account to satisfy the requirements of a legal decision (power of attorney), or to instantiate a new "joint" owner to an account that has been modified to allow joint accounts.

5.6.1.1 Request <PartyAcctRelAddRq>

Tag	Type	Usage	Description
<RqUID>	UUID	Required	Request Identifier.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header Aggregate
<AsyncRqUID>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously.

Tag	Type	Usage	Description
<CustId>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<DepAcctId>	Aggregate *see section 3.2.7.1.2*	Required XOR	Deposit Account Identification Aggregate.
<CardAcctId>	Aggregate *see section 3.2.7.1.3*	Required XOR	Card Account Identification Aggregate.
<LoanAcctId>	Aggregate *see section 3.2.7.1.4*	Required XOR	Loan Account Identification Aggregate.
<PartyAcctRelInfo>	Aggregate *see section 5.2.4*	Required Repeating	Party to Account Relationship Aggregate.

5.6.1.2   Response <PartyAcctRelAddRs>

Tag	Type	Usage	Description
<Status>	Aggregate *see section 3.2.12.1*	Optional	Status Aggregate
<RqUID>	UUID	Required Echoed	Request Identifier.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header Aggregate
<MsgRsHdr>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header Aggregate
<AsyncRqUID>	UUID	Optional Echoed	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously. For more information, see section **3.2.12.1**.
<CustId>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<DepAcctId>	Aggregate *see section 3.2.7.1.2*	Required XOR Echoed	Deposit Account Identification Aggregate.
<CardAcctId>	Aggregate *see section 3.2.7.1.3*	Required XOR Echoed	Card Account Identification Aggregate.

Tag	Type	Usage	Description
<LoanAcctId>	Aggregate *see section 3.2.7.1.4*	Required XOR Echoed	Loan Account Identification Aggregate.
<PartyAcctRelInfo>	Aggregate *see section 5.2.4*	Required Repeating Echoed	Party to Account Relationship Information Aggregate.
<PartyAcctRelRec>	Aggregate *see section 5.2.4.1*	Required Repeating	Party to Account Relationship Record Aggregate
<CSPRefId>	Identifier	Optional	Customer Service Provider Reference Identifier.
<SPRefId>	Identifier	Optional	Service Provider Reference Identifier.

5.6.2   Party to Account Relationship Delete

The PartyAcctRelDel will delete the relationship between the party and the account. This will not delete either the party or the account from the system. Under no circumstances will the deletion request delete the last party to account relationship.

5.6.2.1   Request <PartyAcctRelDelRq>

Tag	Type	Usage	Description
<RqUID>	UUID	Required	Request Identifier.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header Aggregate
<AsyncRqUID>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously.
			For more information, see section **3.2.12.1**.
<CustId>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<EffDt>	DateTime	Optional	Effective Date. The date this activity will take effect.
<PartyAcctRelId>	Aggregate	Required Repeating	Party to Account Relationship Id.

5.6.2.2   Response <PartyAcctRelDelRs>

Tag	Type	Usage	Description
<Status>	Aggregate *see section 3.2.12.1*	Optional	Status Aggregate
<RqUID>	UUID	Required Echoed	Request Identifier.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header Aggregate

Tag	Type	Usage	Description
<MsgRsHdr>	Aggregate *see section 3.2.1.2*	Optional	Message Respnse Header Aggregate
<AsyncRqUID>	UUID	Optional Echoed	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously. For more information, see section **3.2.12.1**.
<CustId>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<EffDt>	DateTime	Optional Echoed	Effective Date. The date this activity will take effect.
<PartyAcctRelId>	Aggregate	Required Repeating Echoed	Party to Account Relationship Information Aggregate.
<PartyAcctRelRec>	Aggregate *see section 5.2.4.1*	Optional Repeating	Party to Account Relationship Record Aggregate
<CSPRefId>	Identifier	Optional	Customer Service Provider Reference Identifier.
<SPRefId>	Identifier	Optional	Service Provider Reference Identifier.

5.6.3   *Party to Account Relationship Inquiry*

The PartyAcctRelInq will inquire on all parties related to a specific account, or inquire on all accounts related to a specific party. If the <DepAcctId>, <CardAcctId>, or the <LoanAcctId> is used in the request, the return base will be all the relationships associated to the account. If none of these three aggregates are present, the return base will be all the accounts related to the party represented within the <CustId>. If both are present, the only search criteria used will be the account information, and the customer Id will represent the customer the request is being acted on behalf of.

5.6.3.1   Request <PartyAcctRelInqRq>

Tag	Type	Usage	Description
<RqUID>	UUID	Required	Request Identifier.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header Aggregate
<AsyncRqUID>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously. For more information, see section **3.2.12.1**.
<CustId>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.

Tag	Type	Usage	Description
<DepAcctId>	Aggregate *see section 3.2.7.1.2*	Optional XOR	Deposit Account Identification Aggregate.
<CardAcctId>	Aggregate *see section 3.2.7.1.3*	Optional XOR	Card Account Identification Aggregate.
<LoanAcctId>	Aggregate *see section 3.2.7.1.4*	Optional XOR	Loan Account Identification Aggregate.
<SelRangeDt>	Aggregate *see section 3.2.10.1*	Optional	Selection Range Date Aggregate.
<RecCtrlIn>	Aggregate *see section 3.2.12.2.1*	Optional	Records Control Input Aggregate.
<OpenAcctInd>	Boolean	Optional	Option for returning only Open Accounts. If Open Account Indicator = True, only return open accounts.

5.6.3.2   Response <PartyAcctRelInqRs>

Tag	Type	Usage	Description
<Status>	Aggregate *see section 3.2.12.1*	Optional	Status Aggregate
<RqUID>	UUID	Required Echoed	Request Identifier.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header Aggregate
<MsgRsHdr>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header Aggregate
<AsyncRqUID>	UUID	Optional Echoed	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously. For more information, see section **3.2.12.1**.
<CustId>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<DepAcctId>	Aggregate *see section 3.2.7.1.2*	Optional XOR Echoed	Deposit Account Identification Aggregate.
<CardAcctId>	Aggregate *see section 3.2.7.1.3*	Optional XOR Echoed	Card Account Identification Aggregate.
<LoanAcctId>	Aggregate *see section 3.2.7.1.4*	Optional XOR Echoed	Loan Account Identification Aggregate.

Tag	Type	Usage	Description
<RecCtrlOut>	Aggregate *see section 3.2.12.2.2*	Optional *but see Description*	Records Control Output Aggregate. Required if <RecCtrlIn> was provided in the request and the server supports Records Control.
<OpenAcctInd>	Boolean	Optional Echoed	Option for returning only Open Accounts. If Open Account Indicator =1, only return account with account status ="Open" Default is "0"(No)
<PartyAcctRelRec>	Aggregate *see section 5.2.4.1*	Required Repeating	Party to Account Relationship Record Aggregate.
<CSPRefId>	Identifier	Optional	Customer Service Provider Reference Identifier.
<SPRefId>	Identifier	Optional	Service Provider Reference Identifier.

5.6.4   Party to Account Relationship Modify

The PartyAcctRelMod will modify existing relationships, but not add or delete relationships, associated with a specific account. Multiple relationships for one party may be modified with one PartyAcctRelMod message. However, in order to modify relationships for multiple parties, one PartyAcctRelMod message will need to be sent for each party.

5.6.4.1   Request <PartyAcctRelModRq>

Tag	Type	Usage	Description
<RqUID>	UUID	Required	Request Identifier.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header Aggregate
<AsyncRqUID>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously. For more information, see section **3.2.12.1**.
<CustId>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<EffDt>	DateTime	Optional	Effective Date. The date this activity will take effect.
<PartyAcctRelId>	Aggregate	Required	Party Account Relationship Identifier.
<PartyAcctRelInfo>	Aggregate *see section 5.2.4*	Required	Party to Account Relationship Aggregate.

5.6.4.2   Response <PartyAcctRelModRs>

Tag	Type	Usage	Description
<Status>	Aggregate *see section 3.2.12.1*	Optional	Status Aggregate

Tag	Type	Usage	Description
<RqUID>	UUID	Required Echoed	Request Identifier.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header Aggregate
<MsgRsHdr>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header Aggregate
<AsyncRqUID>	UUID	Optional Echoed	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously. For more information, see section **3.2.12.1**.
<CustId>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<EffDt>	DateTime	Optional Echoed	Effective Date. The date this activity will take effect.
<PartyAcctRelId>	Aggregate	Required Echoed	Party Account Relationship Identifier.
<PartyAcctRelInfo>	Aggregate *see section 5.2.4*	Required Echoed	Party to Account Relationship Information Aggregate.
<PartyAcctRelRec>	Aggregate *see section 5.2.4.1*	Required	Party to Account Relationship Record Aggregate
<CSPRefId>	Identifier	Optional	Customer Service Provider Reference Identifier.
<SPRefId>	Identifier	Optional	Service Provider Reference Identifier.

5.7 Base Service Profile <BaseSvcProfInfo>

The Base Service Profile functions the same way as all other services; therefore, it contains a Profile description for that service.

Profiles for other services may be found at the end of each chapter of this document. A SP must return a complete set of Profiles for the services that it supports in response to a customer-initiated Service Profile Inquiry <SvcProfInqRq>.

The Base Service Profile is defined below. This information is returned to the client in the Service Profile Inquiry message and provides information on how the client should use the Base Service.

Tag	Type	Usage	Description
<SvcCore>	Aggregate *see section 5.2.2.1*	Required	Service Core Aggregate. Information specified for every service.

Tag	Type	Usage	Description
<MsgSupt>	Open Enum	Required Repeating	MessagesSupported.
			Defined values: SvcProfInq, HolInq, CustAdd, CustMod, CustPswdMod, CustAuthMod, CustStatusMod, CustDel, CustInq, CustIdInq, CustAud, CustSync, CustSvcAdd, CustSvcMod, CustSvcStatusMod, CustSvcDel, CustSvcAud, CustSvcSync, SvcAcctAdd, SvcAcctMod, SvcAcctStatusMod, SvcAcctIdMod, SvcAcctDel, SvcAcctInq, SvcAcctAud, SvcAcctSync, CustDiscStatusMod, CustDiscInq, SvcAcctStatusMod, DiscInq, SecObjAdd, SecObjMod, SecObjInq, SecObjAdvise, SecObjDel, SecObjSync, PartyAcctRelAdd, PartyAcctRelDel, PartyAcctRelInq, PartyAcctRelMod
<OptSupt>	Open Enum	Optional Repeating	Options Supported.
			Defined values: RecCtrl, SuppressEcho
<SignonInfo>	Aggregate	Required	
<AuthSupt>	Open Enum	Optional Repeating	Authentication methods supported.
			Defined values: Password, EmbedCert, TranspCert, MagPIN, SessKey
<CryptType>	Open Enum	Optional *but see Description* Repeating	Encryption Type. Required if <AuthSupt> has a value of Password.
			Defined values: None, PKCS#1, PKCS#7, PKCS#10, ECB, CBC, CFB, OFB, TECB, TCBC, TCBC-I, TCFB, TCFB-P, TOFB, TOFB-I, AECB, ACBC, ACFB, AOFB, ACM
<ModPswdFirst>	Boolean	Optional	Password Modification Required First. If set to *True*, indicates server requires customer to perform Modify Password message prior to executing any messages other than signon in the first session.

The following five elements provide options of allowable password characters. They are provided to allow the client to validate a customer-entered password during password creation

Tag	Type	Usage	Description
<Min>	Long	Optional	Minimum Number of Password Characters.
<Max>	Long	Optional	Maximum Number of Password Characters.
<CharType>	Open Enum	Optional	Character Type Code.
			Defined values: AlphaOnly, NumericOnly, AlphaOrNumeric, AlphaAndNumeric, Special
<CaseSen>	Boolean	Optional	Case Sensitive Flag. If set to *True*, indicates that the password is case-sensitive.
<SpaceAlwd>	Boolean	Optional	Spaces Allowed. If set to *True*, indicates that the password allows spaces.
<Memo>	C-255	Optional	Memo Text. The text to be displayed to the user to indicate password-formatting rules.
</SignonInfo>			
<EnrollProf>	Aggregate	Optional	Enroll Profile Aggregate.
<WebEnrollURL>	URL	Optional	Web URL to start Enrollment Process in lieu of using the <CustAddRq> message.
<EnrollDesc>	C-255	Optional	Enrollment Description. Instructions to customer for other enrollment process. For example, the customer services telephone number.

Tag	Type	Usage	Description
\<CryptType\>	Open Enum	Optional *but see Description* Repeating	Encryption Type. Indicates which encryption types are supported for the secret list in the Enroll message. Defined values: None, PKCS#1 , PKCS#7, PKCS#10, ECB, CBC, CFB, OFB, TECB, TCBC, TCBC-I, TCFB, TCFB-P, TOFB, TOFB-I, AECB, ACBC, ACFB, AOFB, ACM Required if IFX in-band enrollment is supported.
\<SecretPrompt\>	Aggregate *see section 3.2.14.1*	Optional Repeating	Secret Prompt. If omitted, the SP does not require the customer to enter any secrets for client enrollment.
\<CustNameReqd\>	Boolean	Optional	Customer Name Required Indicator. If set to *True*, SP requires that \<CustName\> or \<PersonName\> be included in \<CustAddRq\>.
\<PostAddrReqd\>	Boolean	Optional	Customer Postal Address Required Indicator. If set to *True*, SP requires that \<PostAddr\> be included in \<CustAddRq\>.
\</EnrollProf\>			

6 The Banking Service <BankSvc>

The IFX Specification provides core-banking capabilities through the Banking Service. The Banking Service includes functions such as Statements and Account Inquiries, Transfers, Recurring Transfers, Customer Communications, and Bank Mail. These banking functions may be applied to either deposit accounts, loan accounts, or credit card accounts.

6.1 Description

The Banking Service allows clients to perform the following functions:

- Statements and Account Inquiries, including Account Balances, Account Detail Inquiry, Account Closing Statement, Account History, and Interest Rate, Account Taxation and Exchange Rate Inquiries.
- Add, Modify and Delete a Stop Check
- Add, Modify, and Delete and Interbank Transfers (may be immediate or scheduled), and request Line of Credit Advance and loan repayment
- Add, Modify and Delete Recurring and Interbank Transfer Models for scheduled recurring transfers
- Customer Communications such as order a copy of a statement, order check copies, reorder checkbooks, open a term deposit account, order a deposit book, report a lost card, request a copy of a credit card sales slip, report a credit card dispute, report a lost credit card, change the credit limit on a credit card, and send and retrieve banking Email.
- Debits and Credits with the use of the Add, Modify, Delete a Debit Authorization, and Add Debit. Add, Modify, Delete a Credit Authorization, and Add a Credit.

6.1.1 Accounts

The Banking Service specifies accounts using the Account Identification aggregates <DepAcctId>, <LoanAcctId>, and <CardAcctId>

6.1.2 Client-Initiated Actions

6.1.2.1 Account Inquiries

The Banking Service provides messages that allow a client to retrieve detailed information about bank accounts, such as balances or previously initiated transactions.

6.1.2.2 Bank Statement View

The Banking Service provides messages that allow a client to retrieve Bank Account Statements for one or more given periods, comparable to traditional paper statements. The Statement includes various balances, and optionally the transaction detail for the statement period.

6.1.2.3 Stop Checks

The Banking Service provides messages that allow a client to stop payment on a check. Messages are also provided to inquire about stopped checks and to play back messages associated with stopped checks.

6.1.2.4 Order Checks

A customer may use the Check order messages to request that an order of checks be sent. The customer may specify the total quantity of checks to be sent, starting number of the checks, and style of the checks. The customer can also issue messages to establish recurring check orders.

6.1.2.5 Order Deposit Books

A customer may use the Deposit Book Order message to request that deposit book be sent.

6.1.2.6 Retrieve Interest Rates

A customer may request quotes on interest rates for a variety of financial products using the Interest Rate Inquiry message. The customer enters the Account Type, Tax Status, Amount, and Term, and the message returns the current interest rates.

6.1.2.7 Add, Modify, and Delete Transfers

A customer may add, modify, or delete a non-recurring funds transfer.

6.1.2.8 Add, Modify and Delete Recurring Transfer Models

Messages are also provided in the Banking Service to allow a client to manage models for fixed amount recurring funds transfers. The customer enters parameters, such as the transfer amount, the date of the first transfer, the frequency of the transfers, and the term of the model, and the Financial Institution automatically generates transfers based on this model at the requested frequency for the model term. Recurring transfer models may be closed-ended or open-ended (if no term is provided).

6.1.2.9 Credit Card Messages

The Banking Service provides customer service messages related to credit card accounts, including client request for a sales slip copy or report of a credit card dispute.

6.1.2.10 Debit and Credit Messages

The Banking Service provides debit messages to allow cash withdrawals, stamp dispensing, point-of-sale purchases, and similar transactions that incur a debit to an account. Credit messages are provided for cash and check deposits, which involve a credit to an account.

6.1.2.11 Deposit Account Application

A customer may use the Deposit Account Application messages to apply for a deposit account with a financial institution.

6.2 Banking Service Message Summary

Function / Message Name	Required	Comments
Balance Inquiry **\<BalInqRq\>** **\<BalInqRs\>**	Yes	Allows client to retrieve appropriate balances as of the time the message is executed based on the type of Banking Account.
Balance Reversal **\<BalRevRq\>** **\<BalRevRs\>**		Allows a client to reverse a previous request for a balance inquiry.
Account Inquiry **\<AcctInqRq\>** **\<AcctInqRs\>**		Allows client to retrieve current information about the Account that varies by account type. This inquiry includes balances, but does not include message detail.
Account Reversal **\<AcctRevRq\>** **\<AcctRevRs\>**		Allows a client to reverse a previous request for an account inquiry.
Deposit Account Statement Inquiry **\<DepAcctStmtInqRq\>** **\<DepAcctStmtInqRs\>**		Allows client to retrieve a Deposit Account Closing Statement for one or more given periods, comparable to a traditional paper statement. The Closing Statement includes Opening, Closing, and Minimum Ledger Balances and optionally the messages detail for the statement period

Function / Message Name	Required	Comments
Deposit Account Statement Reversal **\<DepAcctStmtRevRq\>** **\<DepAcctStmtRevRs\>**		Allows a client to reverse a previous request for a deposit account statement inquiry.
Deposit Account Statement Advise **\<DepAcctStmtAdviseRq\>** **\<DepAcctStmtAdviseRs\>**		Allows a client to advise an entity of account information for a statement period.
Credit Card Statement Inquiry **\<CCAcctStmtInqRq\>** **\<CCAcctStmtInqRs\>**		Allows client to retrieve a Credit Card Account Closing Statement for one or more given periods, comparable to a traditional paper statement. The Closing Statement includes Opening and Closing Credit Card Balances, as well as Date Due and a number of summary level amounts and optionally the messages detail for the statement period
Credit Card Statement Reversal **\<CCAcctStmtRevRq\>** **\<CCAcctStmtRevRs\>**		Allows a client to reverse a previous request for a credit card statement inquiry.
Deposit Account Transaction Inquiry **\<DepAcctTrnInqRq\>** **\<DepAcctTrnInqRs\>**		Allows client to retrieve the financial messages posted against an account over a specified time period.
Deposit Account Transaction Reversal **\<DepAcctTrnRevRq\>** **\<DepAcctTrnRevRs\>**		Allows a client to reverse a previous request for a deposit account transaction inquiry.
Card Account Transaction Inquiry **\<CCAcctTrnInqRq\>** **\<CCAcctTrnInqRs\>**		Allows a client to retrieve the financial messages posted against an account over a specified time period.
Card Account Transaction Reversal **\<CCAcctTrnRevRq\>** **\<CCAcctTrnRevRs\>**		Allows a client to reverse a previous request for a card account transaction inquiry.
Bank Account Transaction Image Inquiry **\<BankAcctTrnImgInqRq\>** **\<BankAcctTrnImgInqRs\>**		Allows a client to request a copy of a check or sales slip.
Bank Account Transaction Image Reversal **\<BankAcctTrnImgRevRq\>** **\<BankAcctTrnImgRevRs\>**		Allows a client to reverse a previous request for a bank account transaction image inquiry.
Interest Rate Inquiry **\<IntRateInqRq\>** **\<IntRateInqRs\>**		Allows client to retrieve a list of financial products and their associated interest rates.
Interest Rate Reversal **\<IntRateRevRq\>** **\<IntRateRevRs\>**		Allows a client to reverse a previous request for an interest rate inquiry.
Bank Account Taxation Inquiry **\<BankAcctTaxInqRq\>** **\<BankAcctTaxInqRs\>**		Allows client to request details of taxation on a specific bank account.

Function / Message Name	Required	Comments
Foreign Exchange Rate Inquiry **<ForExRateInqRq>** **<ForExRateInqRs>**		Allows client to inquire about an exchange rate from a Financial Institution, and to optionally request an exchange rate commitment
Foreign Exchange Rate Reversal **<ForExRateRevRq>** **<ForExRateRevRs>**		Allows a client to reverse a previous request for a foreign exchange rate inquiry.
Stop Check Add **<StopChkAddRq>** **<StopChkAddRs>**		Allows a client to stop a check or a range of checks.
Stop Check Cancel **<StopChkCanRq>** **<StopChkCanRs>**		Allows client to cancel a previous Stop Check request with a Financial Institution.
Stop Check Inquiry **<StopChkInqRq>** **<StopChkInqRs>**		Allows client to view current Stopped Check records.
Stop Check Audit **<StopChkAudRq>** **<StopChkAudRs>**		Allows client to play back Stopped Check messages associated with the current customer since some past point in time.
Stop Check Synchronization **<StopChkSyncRq>** **<StopChkSyncRs>**		Allows client to play back Stopped Check messages associated with the current customer since some past point in time.
Stop Check Reveresal **<StopChkRevRq>** **<StopChkRevRs>**		Allows a client to reverse a previous stop check request message.
Funds Transfer Add **<XferAddRq>** **<XferAddRs>**		Allows client to schedule a Funds Transfer. Includes immediate transfer.
Funds Transfer Modify **<XferModRq>** **<XferModRs>**		Allows client to modify a pending Transfer.
Funds Transfer Status Modify **<XferStatusModRq>** **<XferStatusModRs>**		Allows client to modify Funds Transfer Status.
Funds Transfer Cancel **<XferCanRq>** **<XferCanRs>**		Allows client to cancel a pending Funds Transfer.
Funds Transfer Inquiry **<XferInqRq>** **<XferInqRs>**		Allows client to view Funds Transfer records.
Funds Transfer Audit **<XferAudRq>** **<XferAudRs>**		Allows client to play back the Funds Transfer messages associated with the current customer since some past point in time.

Function / Message Name	Required	Comments
Funds Transfer Reversal **<XferRevRq>** **<XferRevRs>**		Allows a client to reverse a previous funds transfer request message.
Funds Transfer Synchronization **<XferSyncRq>** **<XferSyncRs>**		Allows client to play back the Funds Transfer messages associated with the current customer since some past point in time.
Recurring Transfer Model Add **<RecXferAddRq>** **<RecXferAddRs>**		Allows client to create a Recurring Transfer Model.
Recurring Transfer Model Modify **<RecXferModRq>** **<RecXferModRs>**		Allows client to modify an existing Recurring Transfer Model.
Recurring Transfer Model Cancel **<RecXferCanRq>** **<RecXferCanRs>**		Allows client to cancel an existing Recurring Transfer Model.
Recurring Transfer Model Inquiry **<RecXferInqRq>** **<RecXferInqRs>**		Allows client to retrieve current Recurring Transfer Model records.
Recurring Transfer Model Audit **<RecXferAudRq>** **<RecXferAudRs>**		Allows client to play back the messages associated with the model itself, as opposed to messages that are created by it, for the current customer since some past point in time.
Recurring Transfer Model Synchronization **<RecXferSyncRq>** **<RecXferSyncRs>**		Allows client to play back the messages associated with the model itself, as opposed to messages that are created by it, for the current customer since some past point in time.
Recurring Transfer Model Reversal **<RecXferRevRq>** **<RecXferRevRs>**		Allows a client to reverse a previous recurring transfer model request message.
Check Order Add **<ChkOrdAddRq>** **<ChkOrdAddRs>**		Allows client to reorder checks.
Check Order Modify **<ChkOrdModRq>** **<ChkOrdModRs>**		Allows client to modify an existing check order.
Check Order Cancel **<ChkOrdCanRq>** **<ChkOrdCanRs>**		Allows client to cancel an existing check order.
Check Order Inquiry **<ChkOrdInqRq>** **<ChkOrdInqRs>**		Allows client to retrieve current check order records.

Function / Message Name	Required	Comments
Check Order Audit **<ChkOrdAudRq>** **<ChkOrdAudRs>**		Allows client to play back the Check Order messages associated with the current customer since some past point in time.
Check Order Synchronization **<ChkOrdSyncRq>** **<ChkOrdSyncRs>**		Allows client to play back the Check Order messages associated with the current customer since some past point in time.
Check Order Reversal **<ChkOrdRevRq>** **<ChkOrdRevRs>**		Allows a client to reverse a previous check order request message.
Recurring Check Order Add **<RecChkOrdAddRq>** **<RecChkOrdAddRs>**		Allows client to create a Recurring Check Order Model.
Recurring Check Order Model Modify **<RecChkOrdModRq>** **<RecChkOrdModRs>**		Allows client to modify an existing Recurring Check Order Model.
Recurring Check Order Model Cancel **<RecChkOrdCanRq>** **<RecChkOrdCanRs>**		Allows client to cancel an existing Recurring Check Order Model.
Recurring Check Order Model Inquiry **<RecChkOrdInqRq>** **<RecChkOrdInqRs>**		Allows client to retrieve current Recurring Check Order Model records.
Recurring Check Order Model Audit **<RecChkOrdAudRq>** **<RecChkOrdAudRs>**		Allows client to play back the messages associated with the model itself, as opposed to messages that are created by it, for the current customer since some past point in time.
Recurring Check Order Model Synchronization **<RecChkOrdSyncRq>** **<RecChkOrdSyncRs>**		Allows client to play back the messages associated with the model itself, as opposed to messages that are created by it, for the current customer since some past point in time.
Recurring Check Order Instance Add **<RecChkOrdInstAddRq>** **<RecChkOrdInstAddRs>**		Allows a client to request a one-off instance of a recurring check order.
Deposit Book Order Add **<DepBkOrdAddRq>** **<DepBkOrdAddRs>**		Allows client to order a Deposit Book from a Financial Institution.
Deposit Book Order Reversal **<DepBkOrdRevRq>** **<DepBkOrdRevRs>**		Allows a client to reverse a previous deposit book order request message.
Debit Authorization Add **<DebitAuthAddRq>** **<DebitAuthAddRs>**		Allows a client to request authorization for a debit to a specified account.

Function / Message Name	Required	Comments
Debit Authorization Modify **<DebitAuthModRq>** **<DebitAuthModRs>**		Allows a client to modify the details of a debit authorization
Debit Authorization Status Modify ***<DebitAuthStatusModRevRq>*** ***<DebitAuthStatusModRevRs>***		Allows a client to change the status of a debit authorization.
Debit Authorization Cancel **<DebitAuthCanRq>** **<DebitAuthCanRs>**		Allows a client to cancel a debit authorization.
Debit Authorization Inquiry **<DebitAuthInqRq>** **<DebitAuthInqRs>**		Allows a client to view debit authorization records.
Debit Authorization Audit **<DebitAuthAudRq>** **<DebitAuthAudRs>**		Allows a client to play back the Debit authorization messages associated with the current customer since some past point in time.
Debit Authorization Reversal **<DebitAuthRevRq>** **<DebitAuthRevRs>**		Allows a client to reverse a previous debit authorization request message.
Debit Authorization Synchronization **<DebitAuthSyncRq>** **<DebitAuthSyncRs>**		Allows a client to synchronize the Debit authorization messages associated with the current customer since some past point in time.
Debit Add **<DebitAddRq>** **<DebitAddRs>**		Allows a client to debit a specified account.
Debit Reversal **<DebitRevRq>** **<DebitRevRs>**		Allows a client to reverse a previous debit request message.
Debit Modify **<DebitModRq>** **<DebitModRs>**		Allows a client to modify the details of a debit.
Debit Cancel **<DebitCanRq>** **<DebitCanRs>**		Allows a client to cancel a debit.
Debit Inquiry **<DebitInqRq>** **<DebitInqRs>**		Allows a client to retrieve debit records.
Debit Audit **<DebitAudRq>** **<DebitAudRs>**		Allows a client to play back the debit messages associated with the current customer since some past point in time.
Debit Synchronize **<DebitSynchRq>** **<DebitSynchRs>**		Allows a client to synchronize the debit messages associated with the current customer since some past point in time.

Function / Message Name	Required	Comments
Debit Status Modify **<DebitStatusModRq>** **<DebitStatusModRs>**		Allows a client to change the status of a debit.
Credit Authorization Add **<CreditAuthAddRq>** **<CreditAuthAddRs>**		Allows a client to request authorization for a credit to a specified account.
Credit Authorization Modify **<CreditAuthModRq>** **<CreditAuthModRs>**		Allows a client to modify the details of a credit authorization
Credit Authorization Cancel **<CreditAuthCanRq>** **<CreditAuthCanRs>**		Allows a client to cancel a credit authorization
Credit Authorization Inquiry **<CreditAuthInqRq>** **<CreditAuthInqRs>**		Allows a client to view credit authorization records
Credit Authorization Audit **<CreditAuthAudRq>** **<CreditAuthAudRs>**		Allows a client to play back the credit authorization messages associated with the current customer since some past point in time.
Credit Authorization Synchronization **<CreditAuthSyncRq>** **<CreditAuthSyncRs>**		Allows a client to synchronize the credit authorization messages associated with the current customer since some past point in time.
Credit Authorization Reversal **<CreditAuthRevRq>** **<CreditAuthRevRs>**		Allows a client to reverse a previous credit authorization request message.
Credit Authorization Status Modify **<CreditAuthStatusModRq>** **<CreditAuthStatusModRs>**		Allows a client to change the status of a credit authorization.
Credit Add **<CreditAddRq>** **<CreditAddRs>**		Allows a client to credit a specified account.
Credit Modify **<CreditModRq>** **<CreditModRs>**		Allows a client to modify the details of a credit.
Credit Cancel **<CreditCanRq>** **<CreditCanRs>**		Allows a client to cancel a credit.
Credit Inquiry **<CreditInqRq>** **<CreditInqRs>**		Allows a client to retrieve credit records.
Credit Audit **<CreditAudRq>** **<CreditAudRs>**		Allows a client to play back the credit messages associated with the current customer since some past point in time.

Function / Message Name	Required	Comments
Credit Synchronize **\<CreditSyncRq\>** **\<CreditSyncRs\>**		Allows a client to synchronize the credit messages associated with the current customer since some past point in time.
Credit Status Modify **\<CreditStatusModRq\>** **\<CreditStatusModRs\>**		Allows a client to change the status of a credit.
Credit Reversal **\<CreditRevRq\>** **\<CreditRevRs\>**		Allows a client to reverse a previous credit request message.
Deposit Account Application Add **\<DepAppAddRq\>** **\<DepAppAddRs\>**		Allows client to apply for a Deposit Account.
Deposit Account Application Modify **\<DepAppModRq\>** **\<DepAppModRs\>**		Allows client to modify a pending Deposit Account Application.
Deposit Account Application Cancel **\<DepAppCanRq\>** **\<DepAppCanRs\>**		Allows client to cancel a pending Deposit Account Application.
Deposit Account Application Inquiry **\<DepAppInqRq\>** **\<DepAppInqRs\>**		Allows client to view Deposit Account Application records.
Deposit Account Application Audit **\<DepAppAudRq\>** **\<DepAppAudRs\>**		Allows client to play back the Deposit Account Application messages associated with the current customer since some past point in time.
Deposit Account Application Synchronization **\<DepAppSyncRq\>** **\<DepAppSyncRs\>**		Allows client to play back the Deposit Account Application messages associated with the current customer since some past point in time.
Bank Account Statement Image Inquiry **\<BankAcctStmtImgInqRq\>** **\<BankAcctStmtImginqRs\>**		Allows a client to retrieve pre-rendered bank account statements.
Bank Account Statement Image Reversal **\<BankAcctStmtImgRevRq\>** **\<BankAcctStmtImgRevRs\>**		Allows a client to reverse a previous request for a bank account statement image.
Payment Enclosed Add **\<PmtEnclAddRq\>** **\<PmtEnclAddRs\>**		Allows a client to make a payment towards an outstanding debt. The payment may be in an envelope or the payment and remittance slip may be scanned.
Payment Enclosed Modify **\<PmtEnclModRq\>** **\<PmtEnclModRs\>**		Allows a client to modify a payment enclosed.

Function / Message Name	Required	Comments
Payment Enclosed Inquiry **<PmtEnclInqRq>** **<PmtEnclInqRs>**		Allows a client to retrieve payment enclosed records.
Payment Enclosed Cancel **<PmtEnclCanRq>** **<PmtEnclCanRs>**		Allows a client to cancel a payment enclosed.
Payment Enclosed Reversal **<PmtEnclRevRq>** **<PmtEnclRevRs>**		Allows a client to reverse a previous payment enclosed request message.
Check Issue Add **<ChkIssueAddRq>** **<ChkIssueAddRs>**		Allows a client to request check issue information to be added.
Chieck Issue Modify **<ChkIssueModR>** **<ChkIssueModRs>**		Allows a client to modify the details of check issue information.
Check Issue Status Modify **<ChkIssueStatusModRq>** **<ChkIssueStatusModRs>**		Allows a client to modify the status of the details of check issue information.
Check Issues Delete **<ChkIssueDelRq>** **<ChkIssueDelRs>**		Allows a client to delete the details of check issue information.
Check Issue Inquiry **<ChkIssueInqRq>** **<ChkIssueInqRs>**		Allows a client to view the details of check issue information.
Check Issue Audit **<ChkIssueAudRq>** **<ChkIssueAudRs>**		Allows a client to play back the check issue messages associated with the current customer since some past point in time.
Check Issue Sync **<ChkIssueSyncRq>** **<ChkIssueSyncRs>**		Allows a client to synchronize the check issue messages associated with the current customer since some past point in time.
Bank Service Checksum Add **<BankSvcChkSumAddRq>** **<BankSvcChkSumAddRs>**		Allows a client to request bank service Checksum information to be added.
Bank Service Checksum Modify **<BankSvcChkSumModRq>** **<BankSvcChkSumModRs>**		Allows a client to modify bank service Checksum information.
Bank Serice Checksum Status Modify **<BankSvcChkSumStatusModRq>** **<BankSvcChkSumStatusModRs>**		Allows a client to modify the status of bank service Checksum information.
Bank Service Checksum Delete **<BankSvcChkSumDelRq>** **<BankSvcChkSumDelRs>**		Allows a client to delete bank service Checksum information.

Function / Message Name	Required	Comments
Bank Service Checksum Inquiry **<BankSvcChkSumInqRq>** **<BankSvcChkSumInqRs>**		Allows a client to view the bank service Checksum information.
Bank Service Checksum Audit **<BankSvcChkSumAudRq>** **<BankSvcChkSumAudRs>**		Allows a client to play back the bank service Checksum messages associated with the current customer since some past point in time.
Bank Service Checksum Synchronize **<BankSvcChkSumSyncRq>** **<BankSvcChkSumSyncRs>**		Allows a client to synchronize the bank service Checksum messages associated with the current customer since some past point in time.

6.3 Banking Service Common Aggregates

6.3.1 Bank Service Checksum Record Aggregate <BankSvcChkSumRec>

Tag	Type	Usage	Description
<BankSvcChkSumId>	Identifier	Required	Bank Service Checksum Identifier. Assigned by the server at the time the bank service Checksum is first added.
<BankSvcChkSumInfo>	Aggregate *see section 6.3.1.1*	Required	Bank Service Checksum Information Aggregate.
<BankSvcChkSumStatus>	Aggregate *see section 6.3.1.2*	Required	Bank Service Checksum Status Aggregate.

6.3.1.1 Bank Service Checksum Information Aggregate <BankSvcChkSumInfo>

The Bank Service Checksum Information <BankSvcChkSumInfo> aggregate provides information about the check file and its summary control totals. These control totals provide information necessary for the server to verify the data contained in a Pay Service Request <PaySvcRq>.

Tag	Type	Usage	Description
<RefInfo>	Aggregate *see section 3.2.17*	Optional	Issuance transaction reference Identifier number. Identifier for the transaction file generated by the client.
<MsgType>	Open Enum	Required	Bank Service Checksum Message Type. Message type of transaction being calculated for the Bank Service Checksum. Defined values: All IFX Add and Mod Messages in Banking Services. Example: XferModRq, StopChkAddRq, etc.
<Count>	Long	Required	Transaction count. This count indicates how many request messages should be contained in the file or <BankSvcRq>.
<BankSvcChkSum>	Aggregate	Optional Repeating	Bank Service Checksum Aggregate. Defines different types of Checksum and values for verification of control Checksum totals with file transactions. Example: transaction amount sum, account Id sum or bank Id sum (foreign payment transactions).

Tag	Type	Usage	Description
<BankSvcChkSumType>	Open Enum	Required	Bank Service Checksum Type. Identifies the type of Checksum that is contained in <BankSvcChkSumValue>.
			Defined values: Sum, Hash, AcctId, BankId
<BankSvcChkSumValue>	Decimal	Required	Bank Service Checksum Value. Value of Bank Service Checksum type <BankSvcChkSumType>. Example, if used for the sum, <BankSvcChkSumValue> contains the total amount of all transactions of the file, without currency implication. Thus it contains the total of all <CurAmt>s in all <RemitInfo>s of the associated <BankSvcRq>, reflecting all items in the file.
</BankSvcChkSum>			

6.3.1.2 Bank Service Checksum Status Aggregate <BankSvcChkSumStatus>

The Bank Service Checksum Status <BankSvcChkSumStatus> is returned in responses to Add or Modify BankSvcChkSum.

Tag	Type	Usage	Description
<BankSvcChkSumStatusCode>	Open Enum	Required	Bank Service Checksum Status Code. This identifies the Bank Service Checksum status.
			Defined values: Pending, Posted, Refused, Rejected, Returned, DelPend, Deleted
<StatusDesc>	C-255	Optional	Status Description. Explanatory text associated with this status.
<EffDt>	DateTime	Required	Effective Date. The date associated with the change to the current status.
<StatusModBy>	Open Enum	Optional	Status Modified By. If present, indicates who modified the Bank Service Checksum Status Code.
			Defined values: Customer, CSP, CSPSR, BSP, BSPSR, CPP, CPPSR, BPP, and BPPSR.
<Memo>	C-255	Optional	Memo. Intended to be optionally used by a BPP to store any necessary text associated with the Bank Service Checksum Control status.

6.3.2 Credit Record Aggregate <CreditRec>

The <CreditRec> aggregate contains the credit record.

Tag	Type	Usage	Description
<CreditId>	Identifier	Required	Credit Identifier. Assigned by the server at the time the Credit is first added. Cannot be changed by the client.
<CreditInfo>	Aggregate *see section 6.3.29*	Required	Credit Information aggregate.
<CreditStatus>	Aggregate *see section 6.3.3*	Required	Credit Status aggregate.

6.3.3   Credit Status Aggregate <CreditStatus>

The <CreditStatus> aggregate contains details of the status of a credit object.

Tag	Type	Usage	Description
<CreditStatusCode>	Open Enum	Required	The current status of the credit object.
			Defined Values:
			Authorized, Rejected, Posted, Held
<EffDt>	DateTime	Required	Credit Status Date. The date associated with the state change to the current state.
<StatusModBy>	Open Enum	Optional	Status Modified By.
			Defined values: BPP, BPPSR, BSP, BSPSR, CPP, CSP, CSPSR, Customer, FI

6.3.4   Credit Media Item Aggregate <CreditMediaItem>

The <CreditMediaItem> aggregate contains details of the media item (e.g. check, cash) being deposited.

Tag	Type	Usage	Description
<CreditMediaItemId>	Identifier	Optional *but see description*	Credit Media Item Identifier. If more than one <CreditMediaItem> aggregate is exists then the <CreditMediaItemId> is required.
<CreditMediaCashInfo>	Aggregate *see section 6.3.5*	Required XOR	Credit Media Cash Information aggregate. This aggregate contains details about cash being deposited.
<CreditMediaChkInfo>	Aggregate *see section 6.3.6*	Required XOR	Credit Media Check Information aggregate. This aggregate contains details about the check being deposited.

6.3.5   Credit Media Cash Information Aggregate <CreditMediaCashInfo>

The <CreditMediaCashInfo> aggregate contains details about a cash media item being deposited.

Tag	Type	Usage	Description
<CashAmt>	CurAmt	Required	Cash Amount. The cash denomination being deposited.
<CashValidity>	Open Enum	Optional	Cash Validity. The validity of the cash being deposited.
			Defined Values:
			Genuine, Counterfeit, Suspect
<CashQuality>	Open Enum	Optional	Cash Quality. The quality of the cash being deposited.
			Defined Values:
			Fit, Unfit

6.3.6   Credit Media Check Information Aggregate <CreditMediaChkInfo>

The <CreditMediaChkInfo> aggregate contains details about a check media item being deposited.

Tag	Type	Usage	Description
<ChkEnteredAmt>	CurAmt	Required	Check Entered Amount. The amount the customer entered as the check amount being deposited.

Tag	Type	Usage	Description
<ChkCourtesyAmt>	CurAmt	Optional	Check Courtesy Amount. The amount read from the courtesy amount of the check being deposited.
<MicrReadIntegrity>	Open Enum	Optional	MICR Read Integrity. The integrity of the MICR data read from the check being deposited.
			Defined Values:
			Good, Partial, Bad
<MicrUnformatted>	C-64	Optional	MICR Unformatted. The unformatted MICR data read from the check being deposited.

6.3.7   Deposit Account Record Aggregate <DepAcctRec>

Tag	Type	Usage	Description
<StopCount>	Long	Optional	Number of Stopped Payments.
<HoldCount>	Long	Optional	Number of Holds outstanding.
<LastDepDt>	Date	Optional	Last Deposit Date.
<LastDepCurAmt>	Currency Amount	Optional	Last Deposit Amount.
<DepMatureDt>	Date	Optional	Deposit Maturity Date.

6.3.8   Credit Card Account Record Aggregate <CCAcctRec>

Tag	Type	Usage	Description
<DueDt>	Date	Optional	Payment Due Date.
<ExpDt>	DateTime	Optional	Expiration date for card. If the card has only month and year expiration, the *last* day of the month must be specified here.
<LastPmtDt>	Date	Optional	Last Payment Date.
<LastPmtCurAmt>	Currency Amount	Optional	Last Payment Amount.

6.3.9   Common Loan and LOC Account Aggregate <LoanInfoCommon>

This aggregate contains information that is common to both the <LoanInfo> and <LocInfo> aggregates.

Tag	Type	Usage	Description
<OrigDt>	Date	Optional	Original Date.
<Freq>	Open Enum	Optional	Frequency. Repayment Frequency.
<RemainingPmtCount>	Long	Optional	Number of Payments Remaining.
<PurposeDesc>	C-80	Optional	Purpose Description—Purpose for Loan or LOC
<CollateralDesc>	C-80	Optional	Collateral Description.
<CompletedPmtCount>	Long	Optional	Number of Payments Completed.
<RegPmtCurAmt>	Currency Amount	Optional	Regular Payment Amount
<DueDt>	Date	Optional	Date Next Payment Due.

Tag	Type	Usage	Description
<NextPmtCurAmt>	Currency Amount	Optional	Next Payment Amount, including any past due amounts that are now due. A first or last payment may also differ from the regular payment amount, even if there is no past due amount.
<LastPmtDt>	Date	Optional	Last Payment Date.
<LastPmtCurAmt>	Currency Amount	Optional	Last Payment Amount.
<PastDuePmtCount>	Long	Optional	Number of Past Due Payments.
<PrepayPenalty>	Currency Amount	Optional	Prepay Penalty. Amount charged for paying off a loan prior to the specified term.

6.3.10   Loan Account Record Aggregate <LoanAcctRec>

Tag	Type	Usage	Description
<Term>	Aggregate *see section 3.2.9.1*	Optional	Term Aggregate.
<MatDt>	Date	Optional	Loan Maturity Date.
<LoanInfoCommon>	Aggregate *see section 6.3.9*	Optional	Loan and LOC common information aggregate

6.3.11   Line of Credit Account Record Aggregate <LOCAcctRec>

Tag	Type	Usage	Description
<LOCLimit>	Currency Amount	Optional	LOC Limit—The maximum authorized amount for the line of credit.
<MinPmtCurAmt>	Currency Amount	Optional	Minimum Payment Amount.
<ExpDt>	DateTime	Optional	Expiration Date. If omitted, the LOC is open ended.
<LoanInfoCommon>	Aggregate *see section 6.3.9*	Optional	Loan and LOC common information aggregate

6.3.12   Mortgage Account Record Aggregate <MortAcctRec>

Tag	Type	Usage	Description
<OrigDt>	Date	Optional	Origination Date.
<LoanInfoCommon>	Aggregate *see section 6.3.9*	Optional	Loan Information.
<LastPmtDt>	Date	Optional	Last Payment Date.
<TaxPaidDt>	Date	Optional	Tax Paid Date.
<TaxPaidCurAmt>	Currency Amount	Optional	Tax Paid Amount.
<LastYrTaxPaidDt>	Date	Optional	Last Year Tax Paid Date.

Tag	Type	Usage	Description
<LastYrTaxPaidCurAmt>	Currency Amount	Optional	Last Year Tax Paid Amount.
<InsPaidDt>	Date	Optional	Mortgage Insurance Paid Date.
<InsPaidCurAmt>	Currency Amount	Optional	Mortgage Insurance Paid Amount.
<LastYrInsPaidDt>	Date	Optional	Mortgage Last Year Insurance Paid Date.
<LastYrInsPaidCurAmt>	Currency Amount	Optional	Mortgage Last Year Insurance Paid Amount.

6.3.13   Bank Account Transaction Record Aggregate <BankAcctTrnRec>

The <BankAcctTrnRec> aggregate describes the data common to all banking account transaction types.

Tag	Type	Usage	Description
<TrnType>	Open Enum	Optional	Transaction Type.
			Defined values: Debit, Credit, Withdrawal, Check, Deposit, Transfer, Payment, Interest, Dividend, DirectDeposit, DirectDebit, RepeatPayment, Fee, ServiceCharge, Adjustment, BAI:*xxx* or TMA:*xxx* (Reference BAI Code List at http://www.bai.org/operations/bai_codes.html, or TMA Code List at http://www.AFPonline.org/Information_Center/Publications/Bookstore/afpbooks/afpbooks.html as maintained by Association of Financial Professionals (AFP)) Example: BAI:010 (Beginning Ledger Balance), BAI:140 (ACH Credits), etc.
<TrnSrc>	Open Enum	Optional	Transaction Source. See Data Dictionary for details.
			Defined values: ATM, Teller, POS, ARU, HomeBank, ACH.
<PostedDt>	Date	Required	Posted Date. For banking, the date on which the transaction was recorded against the account. For transfers, the date on which the entries were made on the books of the receiving Financial Institution.
<OrigDt>	Date	Optional	Origination Date. The date on which the customer originated the transaction.
<EffDt>	DateTime	Optional	Effective Date. Bank offset date for a correcting transaction; date of previous transaction
<CurAmt>	Currency Amount	Required	Currency Amount. Always in the currency of the account.
<OrigCurAmt>	Currency Amount	Optional	Original Currency Amount. Indicates the transaction was initiated in a currency other than the default for the account (funding account in the case of transfers or payments). The currency rate, if provided, must be included here.
<CompositeCurAmt>	Aggregate *see section 3.2.16.1*	Optional	Composite Currency Amount Aggregate. Bank charges on transaction.
<IndustId>	Aggregate *see section 3.2.13*	Optional	Standard Industrial ID Aggregate. For categorization purposes.
<Memo>	C-255	Optional Repeating	Memo. Additional information about the transaction.
<Name>	C-40	Optional	Name of payee or merchant.

Tag	Type	Usage	Description
\<CounterpartyInfo>	Aggregate *see section 7.3.6.1*	Optional	Counterparty Information Aggregate
\<RefInfo>	Aggregate *see section 3.2.17*	Optional Repeating	Reference Information Aggregate. Identification information, typically supplied by the customer, used to correlate the bank account transaction to an associated \<XferInfo> or \<RemitInfo> sent by the client.
\<BillRefInfo>	NC-80	Optional	Biller Reference Information. Biller-defined text from \<BillRec>, for the biller's Accounts Receivable reconciliation. Depending on the Payment Provider and the payment network and message used to make the payment, \<BillRefInfo> may be truncated to 22 characters.
\<RemitAdviceRefId>	Identifier	Optional	Remittance Advice Reference Identification. Remit advice identification for reconciliation process to link payments and remittance advice. This is also used to reconcile two information streams when remittance and payment travel separately.
\<CustPayeeInfo>	Aggregate *see section 7.3.2.1*	Optional	Customer Payee Information Aggregate.
\<InvoiceInfo>	Aggregate *see section 7.3.6*	Optional	Invoice Information Aggregate. Structured advice information on the transaction
\<CSPRefId>	Identifier	Optional	CSP Reference Identifier.
\<SPRefId>	Identifier	Optional	Service Provider Reference Identifier.
\<SPRefIdCorrect>	Aggregate	Optional	SP Reference Identifier Correction. Either replaces or deletes a previous transaction depending on the value of \<CorrectAction>.
\<SPRefId>	Identifier	Required	Old Message Identifier. This is the \<SPRefId> of the transaction that is being replaced or deleted.
\<CorrectAction>	Closed Enum	Required	Correction Action.
\</SPRefIdCorrect>			

6.3.14   Deposit Account Transaction Record Aggregate \<DepAcctTrnRec>

The \<DepAcctTrnRec> aggregate describes a single transaction posted against a Deposit Account. It is used in messages that provide transaction detail.

The sign convention for the statement message aggregate is as follows: a positive \<CurAmt> is added to the account balance and a negative \<CurAmt> is subtracted from the account balance

Sign of \<CurAmt>	Meaning
Positive	Increase customer asset.
Negative	Decrease customer asset.

Tag	Type	Usage	Description
\<BankAcctTrnRec>	Aggregate *see section 6.3.13*	Required	Banking Transaction Record.

Tag	Type	Usage	Description
\<AvailDt\>	Date	Optional	Available Date. The date on which funds are available.
\<CompositeCurAmt\>	Aggregate *see section 3.2.16.1*	Optional Repeating	Composite Currency Amount Aggregate. This aggregate is generally used here to contain float amounts.
\<ChkNum\>	NC-12	Optional	Check Number.
\<XferId\>	Identifier	Optional	Funds Transfer Identifier. Assigned by the server at the time the Funds Transfer is first added. Cannot be changed by the client.
\<PmtId\>	Identifier	Optional	Payment Identifier. Assigned by the server at the time the Payment is first added. Cannot be modified by the client.
\<DepAcctIdTo\>	Aggregate	Optional XOR	Deposit Account Detail. Used for transfer messages.
\<CardAcctIdTo\>	Aggregate	Optional XOR	Card Account Detail. Used for transfer messages.
\<LoanAcctIdTo\>	Aggregate	Optional XOR	Loan Account Detail. Used for transfer messages.
\<NetworkTrnInfo\>	Aggregate *see section 3.2.18*	Optional XOR 1.1+	Network Transaction Information Aggregate. Contains information regarding the network that processed the transaction.
			This aggregate replaces \<ATMTrnInfo\> and \<USA.ACHTrnInfo\>, both of which will be deprecated in IFX 2.0.
\<ATMTrnInfo\>	Aggregate	Optional XOR	ATM (Automated Teller Machine) Message Information.
			This aggregate will be deprecated in IFX 2.0, being replaced by \<NetworkTrnInfo\>
\<ATMOwner\>	C-14	Optional	The name of the individual or organization which owns the ATM.
\<ATMLocation\>	C-18	Optional	Alphanumeric string assigned by the ATM owner that specifies the ATM location.
\<ATMCityState\>	C-15	Optional	Alphanumeric string assigned by the ATM owner that specifies the ATM city and state or province.
\</ATMTrnInfo\>			
\<USA.ACHTrnInfo\>	Aggregate	Optional	ACH Message Information Aggregate.
			This aggregate will be deprecated in IFX 2.0, being replaced by \<NetworkTrnInfo\>
\<OriginatorName\>	C-40	Required	Name of originator of the ACH Message.
\<USA.RTN\>	NC-9	Optional	Transit Routing Number. A number uniquely identifying an organization providing products and services of a monetary or financial nature within the United States.
\</USA.ACHTrnInfo\>			

6.3.15 Credit Card Account Transaction Record Aggregate \<CCAcctTrnRec\>

The sign convention for the statement transaction aggregate is as follows: a positive \<CurAmt\> is added to the account balance and a negative \<CurAmt\> is subtracted from the account balance.

Sign of \<CurAmt\>	Meaning
Positive	Increase customer liability.

Negative	Decrease customer liability.

Tag	Type	Usage	Description
<BankAcctTrnRec>	Aggregate *see section 6.3.13*	Required	Banking Transaction Record.
<SalesSlipRefNum>	NC-23	Required	Sales Slip Reference Number.
<Memo>	C-255	Optional	Memo. Additional information about the message.

6.3.16 Selection Range Check Number Aggregate <ChkRange>

Tag	Type	Usage	Description
<ChkNumStart>	NC-12	Required OR	Selection Low Check Number.
<ChkNumEnd>	NC-12	Required OR	Selection High Check Number.

6.3.17 Stop Check Record Aggregate <StopChkRec>

The <StopChkRec> aggregate is generally used in response messages related to Stopped Checks.

Tag	Type	Usage	Description
<StopChkInfo>	Aggregate *see section 6.3.17.1*	Required	Stop Check Information.
<PostedDt>	Date	Optional	Check Posting Date. The date on which the check is posted against the customer's account.
<StopChkStatusCode>	Closed Enum	Required	Stop Check Status Code. The status of a Stop Check request that is returned as part of a Stop Check inquiry. Valid values: Pending, Stopped, Returned, Cleared, Rejected, Cancelled
<StatusDesc>	C-255	Optional	Check Error Description.

6.3.17.1 Stop Check Information Aggregate <StopChkInfo>

The <StopChkInfo> aggregate is used in messages related to Stop Checks. It is generally used in the request messages.

Tag	Type	Usage	Description
<ChkNum>	NC-12	Required	Check Number.
<Name>	C-40	Optional	Check Pay-To Name.
<CurAmt>	Currency Amount	Optional	Check Currency Amount.
<OrigDt>	Date	Optional	Check Origination Date. The date on which the customer originates the check.
<Desc>	C-80	Optional	Description. Additional information, such as why the check is being stopped.

6.3.18   Foreign Exchange Rate Information Aggregate <ForExRateInfo>

A customer may request a foreign exchange rate commitment from a Financial Institution using the Foreign Exchange Rate Message. This commitment may be stored by the client and may be referenced by an Intra- or Interbank transfer until the expiration date/time.

Tag	Type	Usage	Description
<CurAmt>	Currency Amount	Optional *but see Description*	Message Amount. This is the amount in the TO currency at the quoted exchange rate. Must be included when the Financial Institution quotes the commitment in the Foreign Exchange Response Message. Included by the client in a subsequent transfer request only if the transfer amount = the commitment amount.
<CurRate>	Decimal	Required	Exchange Rate.
<CurConvertRule>	Closed Enum	Required	Currency Conversion Indicator. Valid values: Direct, Indirect. See Section 2.3.8 for more information.
<Fee>	Aggregate *see section 3.2.16.2*	Optional Repeating	Fee. The fee(s) required to execute the exchange may be provided to the customer for informational purposes.
<ExpDt>	DateTime	Optional *but see Description*	Expiration Date/Time. Required if it is a committed rate. Explicit commitment expiration date and time. After the date/time specified in this field, the Financial Institution is under no obligation to honor the rate returned.
<CSPRefId>	Identifier	Optional	Customer Service Provider Reference Identifier.
<SPRefId>	Identifier	Optional	Service Provider Reference Identifier.

6.3.19   Transfer Record Aggregate <XferRec>

Tag	Type	Usage	Description
<XferId>	Identifier	Required	Transfer Identifier. Assigned by the server at the time the Transfer is first added.
<RecXferId>	Identifier	Optional	Recurring Transfer Model Identifier. Assigned by the server at the time the Recurring Transfer Model is first added. Cannot be changed by the client. Included only if the Banking provider generated the transfer instance from a recurring model.
<RecXferMod>	Boolean	Optional *but see Description*	Recurring Transfer Modified Indicator. Required if <XferModRq> has subsequently modified a transfer generated from a Recurring Transfer Model so it no longer matches the Recurring Transfer Model. This may be supplied only for recurring transfer instances.
<XferInfo>	Aggregate *see section 6.3.19.2*	Required	Transfer Information Aggregate.
<XferStatus>	Aggregate *see section 6.3.19.3*	Required	Transfer Status.

6.3.19.1   Recurring Transfer Record Aggregate <RecXferRec>

Tag	Type	Usage	Description
<RecXferId>	Identifier	Required	Recurring Transfer Model Identifier. Assigned by the server at the time the Recurring Transfer Model is first added. Cannot be changed by the client.
<XferInfo>	Aggregate *see section 6.3.19.2*	Required	Transfer Information Aggregate.
<RecModelInfo>	Aggregate *see section 3.2.11.1*	Required	Recurring Model Information Aggregate.
<RemainingInsts>	Long	Required	Remaining Instance Count. The server must calculate this number as the number of actual payments to be made plus the number of instances to skip. Server must calculate on <RecXferAddRq> and return in response. Server must recalculate in case of an <RecXferModRq> that changes <RecSeriesEnd> or when a payment is spawned.

6.3.19.2   Transfer Information Aggregate <XferInfo>

The <XferInfo> aggregate is used in messages related to Transfers, Interbank Transfers, and Recurring Transfer Models for Funds Transfers.

Transfers may be single currency transfers, (i.e. the source and target amount is the same) or may be foreign exchange transfer (i.e. source and target amounts have different ISO currency code). The Financial Institution indicates support of foreign exchange transfers by specifying ForEx within <OptSupt> within <XferProf> in the Banking Profile

In some cases, a customer may not be able to specify the exchange rate for a foreign exchange transfer Typically, the Financial Institution makes the transfer at the prevailing exchange rate at the time of the transfer, often considering such factors as the amount of transfer and the customer relationship with the Financial Institution, in addition to the interbank exchange rate. Some Financial Institutions make exchange rate commitments (see Exchange Rate message). Usually, these commitments expire after a short period of time. In this case, the customer may specify the exchange rate and reference the <CSPRefId> returned within the Exchange Rate Rs. Where a specific foreign exchange contract is referred to, the <ForExDealId> is used instead of the <ForExInfo> aggregate.

Tag	Type	Usage	Description
<DepAcctIdFrom>	Aggregate *see section 3.2.7.1.2*	Required XOR	Deposit Account Identification Aggregate. Usage is transfer source account.
<CardAcctIdFrom>	Aggregate *see section 3.2.7.1.3*	Required XOR	Card Account Identification Aggregate. Usage is transfer source account.
<LoanAcctIdFrom>	Aggregate *see section 3.2.7.1.4*	Required XOR	Loan Account Identification Aggregate. Usage is transfer source account.
<DepAcctIdTo>	Aggregate	Required XOR	Deposit Account Identification Aggregate. Usage is transfer destination account.
<CardAcctIdTo>	Aggregate	Required XOR	Card Account Identification Aggregate. Usage is transfer destination account.

Tag	Type	Usage	Description
\<LoanAcctIdTo\>	Aggregate	Required XOR	Loan Account Identification Aggregate. Usage is transfer destination account.
\<CurAmt\>	Currency Amount	Required	Currency Amount.
\<DueDt\>	Date	Optional	Due Date. If not specified, the transfer is to be scheduled as soon as possible.
\<Category\>	C-40	Optional	Category for this message, modified by the client.
\<ImmediateXfer\>	Boolean	Optional Profiled support	Immediate Transfer Indicator. If set to *True*, the transfer should be executed immediately, and not at end of day. Subject to support in Service Profile.
\<ForExRateInfo\>	Aggregate *see section 6.3.18*	Optional XOR	Foreign Exchange Rate Information Aggregate.
\<ForExDealId\>	Identifier	Optional XOR	Foreign Exchange Deal Identifier (Committed deals). Where a specific foreign exchange contract is referred to, the \<ForExDealId\> is used instead of the \<ForExInfo\> aggregate
\<RefInfo\>	Aggregate *see section 3.2.17*	Optional Repeating	Reference Information Aggregate. Identification information, used to correlate to transactions provided in bank statements.

6.3.19.3 Transfer Status Aggregate \<XferStatus\>

The \<XferStatus\> is returned in responses to Add or Modify Transfer and Add or Modify Recurring Transfer Model. Note that all elements within this aggregate are assigned by the server and cannot be assigned or modified by the client.

Tag	Type	Usage	Description
\<XferStatusCode\>	Closed Enum	Required	Transfer Status Code. This identifies the payment processing status. Valid values: Scheduled, Cancelled, FIHeld, RejNoFund, RejInactive, RejClosed, Returned, Failed, Processed, Posted, Cleared, Skipped
\<StatusDesc\>	C-255	Optional	Status Description. Explanatory text associated with this status.
\<EffDt\>	DateTime	Required	Transfer Status Date. The date associated with the state change to the current state.
\<StatusModBy\>	Open Enum	Optional	Status Modified By. Defined values: BPP, BPPSR, BSP, BSPSR, CPP, CSP, CSPSR, Customer, FI

6.3.20 Check Order Record Aggregate \<ChkOrdRec\>

Tag	Type	Usage	Description
\<ChkOrdId\>	Identifier	Required	Check Order Identifier. Assigned by the server at the time the Check Order is first added.
\<ChkOrdInfo\>	Aggregate *see section 6.3.20.1*	Required	Check Order Information Aggregate.
\<ChkOrdStatus\>	Aggregate *see section 6.3.20.2*	Required	Check Order Status.

6.3.20.1 Check Order Information Aggregate <ChkOrdInfo>

Tag	Type	Usage	Description
<DepAcctId>	Aggregate *see section 3.2.7.1.2*	Required XOR	Deposit Account Identification Aggregate.
<CardAcctId>	Aggregate *see section 3.2.7.1.3*	Required XOR	Card Account Identification Aggregate.
<LoanAcctId>	Aggregate *see section 3.2.7.1.4*	Required XOR	Loan Account Identification Aggregate.
<ChkPrint>	Aggregate *see section 6.3.20.1.1*	Optional XOR	Check Print Aggregate.
<ChkText>	C-80	Optional XOR Repeating	Freeform Check Text.
<CustNameAddr>	Aggregate *see section 3.2.3.4*	Optional Repeating XOR	Customer Name/Address Aggregate. Within this aggregate, <NameAddrType> is most likely set to Delivery.
<PersonInfo>	Aggregate *see section 3.2.4.1*	Optional Repeating XOR	Person Information Aggregate.
<DeliveryDestination>	Open Enum	Optional	Delivery destination. Specifies where to deliver the checks, such as Branch, Consumer, Alternate Address. Default value: Consumer
<Count>	Long	Optional	Number of Checks. If this is missing, the number used for the previous order must be used.
<ChkNumPrefix>	C-2	Optional	Check Number Prefix. Two characters printed immediately before the check number.
<ChkNumStart>	NC-12	Optional	Starting Check Number of the Order.
<ChkNumSuffix>	C-2	Optional	Check Number Suffix. Two characters printed immediately after the check number.
<ChkBkStyleId>	Identifier	Optional	Checkbook Style Identifier. Service Provider assigns defined values. This field indicates customer's choice.
<DeliveryMethod>	Open Enum	Optional Profiled values	Delivery Method. Default is Post. Value must be supported in Service Profile.
<BillingMethod>	Closed Enum	Optional	Standard and Non-Standard billing methods for product(s) ordered. Valid Values: Customer, Institution, CheckPrinter
<CoverMngrm>	C-3	Optional Repeating	Cover Monogram. Up to 3 characters affixed to the covers ordered.
<CoverStyleId>	Identifier	Optional Repeating	Cover Style. Checkbook cover style.
<ArtWorkId>	Identifier	Optional Repeating	For valid cuts and pridemarks. This is dependent on <ChkBkStyleId>.
<IncEndrsmntStmp>	Boolean	Optional	Endorsement Stamp. If *True* an endorsement stamp is included with the check order. If *False*, no endorsement stamp is included with the product.

Tag	Type	Usage	Description
<FontId>	Identifier	Optional	Font Type. The type of font used for the check personalization.
<OvrSigText>	C-40	Optional Repeating	Over Signature Text. Printed information residing above the signature line.
<IncSecSigLine>	Boolean	Optional	Secondary Signature Line. If *True*, a second signature line is printed below the first. If *False*, no secondary signature line is added.
<ClubAcctId>	Identifier	Optional	Identifier for bank club program discounts or special promotioins.
<VoucherTypeId>	Identifier	Optional	Voucher Code. Indicates the voucher apron type included with the order.

6.3.20.1.1 Check Print Aggregate <ChkPrint>

Tag	Type	Usage	Description
<CustName>	Aggregate *see section 3.2.3.2*	Required XOR Repeating	Customer Name Aggregate.
<PersonName>	Aggregate *see section 3.2.4.1.1*	Required XOR Repeating	Person Name Aggregate.
<PostAddr>	Aggregate *see section 3.2.3.7.1*	Required	Customer Address Aggregate.
<Phone>	Phone Number	Optional	Telephone Number. Provided if it is to be printed on checks.
<Desc>	C-80	Optional	Additional Information to be printed on checks.

6.3.20.2 Check Order Status Aggregate <ChkOrdStatus>

Tag	Type	Usage	Description
<ChkOrdStatusCode>	Open Enum	Required	Check Order Status Code.
			Defined values: Accepted, Processed, Sent, Pending, Rejected, Cancelled
<StatusDesc>	C-255	Optional	Status Description. Explanatory text associated with this status.
<EffDt>	DateTime	Required	Check Order Status Date. The date associated with the state change to the current state.
<StatusModBy>	Open Enum	Optional	Status Modified By.
			Defined values: BPP, BPPSR, BSP, BSPSR, CPP, CSP, CSPSR, Customer, FI
<ResolvePendingDt>	Date	Optional	Resolve Pending Date. The date that the Pending status in <ChkOrdStatusCode> is expected to change to a final status; e.g., Sent or Cancelled. This date should be returned when the <ChkOrdStatusCode> is set to Pending.

6.3.21 Recurring Check Order Record Aggregate <RecChkOrdRec>

Tag	Type	Usage	Description
<RecChkOrdId>	Identifier	Required	Recurring Check Order Identifier. Assigned by the server at the time the Recurring Check Order is first added.
<ChkOrdInfo>	Aggregate *see section 6.3.20.1*	Required	Check Order Information Aggregate.
<RecModelInfo>	Aggregate *see section 3.2.11.1*	Required	Recurring Model Information Aggregate.
<RemainingInsts>	Long	Optional	Remaining Instance Count. The server must calculate this number as the number of actual check orders to be made plus the number of instances to skip based on the customer-entered <RecSeriesEnd>.
			Server must calculate on <RecChkOrdAddRq> and return in response. Server must recalculate in case of an <RecChkOrdModRq> that changes <RecSeriesEnd> or when a payment is spawned.

6.3.22 Check Order Detail Aggregate <ChkOrdDetail>

Tag	Type	Usage	Description
<DepAcctId>	Identifier	Optional XOR	Deposit Account Identification Aggregate
<CardAcctId>	Aggregate *see section 3.2.7.1.3*	Optional XOR	Credit Account Identification Aggregate.
<LoanAcctId>	Aggregate *see section 3.2.7.1.4*	Optional XOR	Loan Account Identification Aggregate.
<ChkPrint>	Aggregate *see section 6.3.20.1.1*	Optional XOR	Check Print Aggregate
<ChkText>	C-80	Optional XOR Repeating	Freeform Check Text.
<CustNameAddr>	Aggregate *see section 3.2.3.4*	Optional XOR Repeating	Customer Name/Address Aggregate. Within this aggregate, <NameAddrType> is most likely set to Delivery. Contains the address for check order delivery.
<PersonInfo>	Aggregate *see section 3.2.4.1*	Optional Repeating XOR	Person Information Aggregate.
<DeliveryDestination>	Open Enum	Optional	Delivery destination. Specifies where to deliver the checks, such as Branch, Consumer, Alternate Address, etc.
			Defined values: Branch, Consumer, Alternate Address
			Default value: Consumer
<Count>	Long	Optional	Number of checks. If omitted, the number from the previous order is used.
<ChkNumPrefix>	C-2	Optional	Check Number Prefix. Two characters printed immediately before the check number.

Tag	Type	Usage	Description
<ChkNumStart>	NC-12	Optional	Starting Check Number of the Order.
<ChkNumSuffix>	C-2	Optional	Check Number Suffix. Two characters printed immediately after the check number.
<ChkBkStyleId>	Identifier	Optional	Checkbook Style Identifier. Service Provider assigns defined values. This field indicates customer's choice.
<DeliveryMethod>	Open Enum	Optional Profiled values	Delivery Method. Default is Post. Value must be supported in Service Profile.
<BillingMethod>	Closed Enum	Optional	Standard and Non-Standard billing methods for product(s) ordered. Valid Values: Customer, Institution, CheckPrinter
<CoverMngrm>	C-3	Optional Repeating	Cover Monogram. Up to 3 characters affixed to the covers ordered.
<CoverStyleId>	Identifier	Optional Repeating	Cover Style. Checkbook cover style.
<ArtWorkId>	Identifier	Optional Repeating	For valid cuts and pridemarks. This is dependent on <ChkBkStyleId>.
<IncEndrsmntStmp>	Boolean	Optional	Endorsement Stamp. If *True* an endorsement stamp is included with the check order. If *False*, no endorsement stamp is included with the product.
<FontId>	Identifier	Optional	Font Type. The type of font used for the check personalization.
<OvrSigText>	C-40	Optional Repeating	Over Signature Text. Printed information residing above the signature line.
<IncSecSigLine>	Boolean	Optional	Secondary Signature Line. If *True*, a second signature line is printed below the first. If *False*, no secondary signature line is added.
<ClubAcctId>	Identifier	Optional	Identifier for bank club program discounts or special promotioins.
<VoucherTypeId>	Identifier	Optional	Voucher Code. Indicates the voucher apron type included with the order.

6.3.23 Deposit Book Order Record Aggregate <DepBkOrdRec>

Tag	Type	Usage	Description
<DepBkOrdId>	Identifier	Required	Deposit Book Order Identifier. Assigned by the server at the time the Deposit Book Order is first added.
<DepBkOrdInfo>	Aggregate *see section 6.3.23.1*	Required	Deposit Book Order Information Aggregate.
<DepBkOrdStatus>	Aggregate *see section 6.3.23.2*	Required	Deposit Book Order Status.

6.3.23.1 Deposit Book Order Information Aggregate <DepBkOrdInfo>

Tag	Type	Usage	Description

Tag	Type	Usage	Description
\<DepAcctId>	Aggregate *see section 3.2.7.1.2*	Required	Deposit Account Identification Aggregate.
\<Count>	Long	Optional	Number of Deposit Slips. If omitted, the number used for the previous order must be used.
\<DepBkStyleId>	Identifier	Optional	Style of Deposit Book.
\<DeliveryMethod>	Open Enum	Optional Profiled values	Delivery Method. Used to request the delivery channel for requested information. See Data Dictionary for details. Default is Post.
			Value selected must be supported in Service profile.

6.3.23.2 Deposit Book Order Status Aggregate \<DepBkOrdStatus>

Tag	Type	Usage	Description
\<DepBkOrdStatusCode>	Open Enum	Required	Deposit Book Order Status Code.
			Defined values: Accepted, Processed, Sent, Pending, Rejected, Cancelled
\<StatusDesc>	C-255	Optional	Status Description. Explanatory text associated with this status.
\<EffDt>	DateTime	Required	Deposit Book Order Status Date. The date associated with the state change to the current state.
\<StatusModBy>	Open Enum	Optional	Status Modified By.
			Defined values: BPP, BPPSR, BSP, BSPSR, CPP, CSP, CSPSR, Customer, FI

6.3.24   Transfer Profile Aggregate \<XferProf>

The Transfer Profile Aggregate is used to convey information on the transfer.

Tag	Type	Usage	Description
\<OptSupt>	Open Enum	Optional Repeating	Options Supported.
			Defined values: ForEx, ImmediateXfer, RecCtrl, RecFinalAmt, RecInitialAmt, RecModelNickname, Skip, SchedXfer.
\<PrcSched>	Aggregate *see section 5.2.2.3*	Optional	Processing Schedule Aggregate. If omitted, the default processing schedule is assumed.
\<RecXferProf>	Aggregate	Optional	Recurring Transfer Profile Aggregate.
\<Freq>	Open Enum	Required Repeating	Recurring Model Frequency. Usage is a list of supported frequencies.
			Defined values: Daily, Weekly, Biweekly, TwiceMonthly, Monthly, EndOfMonth, FourWeeks, BiMonthly, Quarterly, SemiAnnually, Annually
\<ModPendingType>	Closed Enum	Required	Client Modify Pending Type.
			Valid values: Always, Never, IfRequested
\</RecXferProf>			

6.3.25 Debit Record Aggregate <DebitRec>

The <DebitRec> aggregate contains the debit record.

Tag	Type	Usage	Description
<DebitId>	Identifier	Required	Debit Identifier. Assigned by the server at the time the Debit is first added. Cannot be changed by the client.
<DebitInfo>	Aggregate *see section 6.3.27*	Required	Debit Information aggregate.
<DebitStatus>	Aggregate *see section 6.3.25.1*	Required	Debit Status aggregate.

6.3.25.1 Debit Status Aggregate <DebitStatus>

The <DebitStatus> aggregate contains details of the status of a debit object.

Tag	Type	Usage	Description
<DebitStatusCode>	Open Enum	Required	The current status of the debit object.
			Defined Values:
			Authorized, Rejected, Posted, Held
<EffDt>	DateTime	Required	Debit Status Date. The date associated with the state change to the current state.
<StatusModBy>	Open Enum	Optional	Status Modified By.
			Defined values: BPP, BPPSR, BSP, BSPSR, CPP, CSP, CSPSR, Customer, FI

6.3.26 Debit Authorization Record Aggregate <DebitAuthRec>

The <DebitAuthRec> aggregate contains the debit authorization record.

Tag	Type	Usage	Description
<DebitAuthId>	Identifier	Required	Debit Authorization Identifier. Assigned by the server at the time the Debit Authorization is first added. Cannot be changed by the client.
<DebitAuthInfo>	Aggregate *see section 6.3.26.1*	Required	Debit Authorization Information aggregate.
<DebitAuthStatus>	Aggregate *see section 6.3.26.2*	Required	Debit Authorization Status aggregate.

6.3.26.1 Debit Authorization Information Aggregate <DebitAuthInfo>

The <DebitAuthInfo> aggregate is used in messages related to debit authorizations. It is generally used in the request messages.

Tag	Type	Usage	Description

Tag	Type	Usage	Description
<DebitAuthType>	Open Enum	Required	The type of the debit authorization.
			Defined Values:
			CashWithdrawal, StampDispense, TicketDispense, ValueCouponDispense, CreditCardAdvance, MerchandisePurchase.
<CompositeCurAmt>	Aggregate *see section 3.2.16.1*	Required Repeating	The amounts of the debit transaction and appropriate fees.
<DepAcctId>	Aggregate *see section 3.2.7.1.2*	Required XOR	Deposit Account Identification Aggregate.
<CardAcctId>	Aggregate *see section 3.2.7.1.3*	Required XOR	Card Account Identification Aggregate.

6.3.26.2 Debit Authorization Status Aggregate <DebitAuthStatus>

The <DebitStatus> aggregate contains details of the status of a debit authorization object.

Tag	Type	Usage	Description
<DebitStatusCode>	Open Enum	Required	The current status of the debit authorization object.
			Defined Values:
			Authorized, Rejected, Posted, Held
<EffDt>	DateTime	Required	Debit Status Date. The date associated with the state change to the current state.
<StatusModBy>	Open Enum	Optional	Status Modified By.
			Defined values: BPP, BPPSR, BSP, BSPSR, CPP, CSP, CSPSR, Customer, FI

6.3.27 Debit Information Aggregate <DebitInfo>

The <DebitInfo> aggregate is used in messages related to debits. It is generally used in the request messages.

Tag	Type	Usage	Description
<DebitAuthType>	Open Enum	Required	The type of the debit authorization transaction.
			Defined Values:
			CashWithdrawal, StampDispense, TicketDispense, ValueCouponDispense, CreditCardAdvance, MerchandisePurchase.
			This element will be deprecated from the DebitInfo aggregate in IFX 2.0. The replacement is to use the DebitType element.

Tag	Type	Usage	Description
<DebitType>	Open Enum	Optional *but see Description*	The type of the debit transaction. Defined Values: CashWithdrawal, StampDispense, TicketDispense, ValueCouponDispense, CreditCardAdvance, CardVerification, TransactionFee, MerchandisePurchase. Since this element will be replacing DebitAuthType in the DebitInfo aggregate, this element will be required in IFX 2.0.
<CompositeCurAmt>	Aggregate *see section 3.2.16.1*	Required Repeating	The amounts of the debit transaction and appropriate fees.
<DepAcctId>	Aggregate *see section 3.2.7.1.2*	Required XOR	Deposit Account Identification Aggregate.
<CardAcctId>	Aggregate *see section 3.2.7.1.3*	Required XOR	Card Account Identification Aggregate.
<DebitAuthId>	Identifier	Optional	Debit Authorization Identifier. Assigned by the server at the time the Debit Authorization is first added. This is used to relate a debit to a previously authorized debit authorization object
<TrnRqUID>	UUID	Optional	Transaction RqUID. If this debit is used as a fee for a transaction (e.g. fee for a balance inquiry) then this element will contain the RqUID of the transaction (e.g. the balance inquiry).
<ClientChgCode>	Open Enum	Optional	Client Change Code. This indicates the reason why a Debit was for a different amount from the original authorization for the debit. Defined Values: None, ConsumerCancelled, TerminalExceptionAmountKnown, TerminalExceptionAmountUnknown, ConsumerExceptionAmountKnown, ConsumerExceptionAmountUnknown
<NetworkTrnInfo>	Aggregate *see section 3.2.18*	Optional	Network Transaction Information. Identification and location of the terminal from which the message originates. This aggregate will be removed from the DebitInfo aggregate in IFX 2.0. With IFX 1.4+, the <NetworkTrnInfo> aggregate is available in the <MsgRqHdr> aggregate.

6.3.28 Credit Authorization Record Aggregate <CreditAuthRec>

The <CreditAuthRec> aggregate contains the credit authorization record.

Tag	Type	Usage	Description
<CreditAuthId>	Identifier	Required	Credit Authorization Identifier. Assigned by the server at the time the Credit Authorization is first added. Cannot be changed by the client.

Tag	Type	Usage	Description
\<CreditAuthInfo\>	Aggregate *see section 6.3.28.1*	Required	Credit Authorization Information aggregate.
\<CreditAuthStatus\>	Aggregate *see section 6.3.28.2*	Required	Credit Authorization Status aggregate.

6.3.28.1 Credit Authorization Information Aggregate \<CreditAuthInfo\>

The \<CreditAuthInfo\> aggregate is used in messages related to credit authorizations. It is generally used in the request messages.

Tag	Type	Usage	Description
\<CreditAuthType\>	Open Enum	Required	The type of the credit authorization. Defined Values: UnverifiedCashDeposit, VerifiedCashDeposit, UnverifiedCheckDeposit, VerifiedCheckDeposit, EnvelopeDeposit, MultiDeposit, MerchandiseReturn.
\<CompositeCurAmt\>	Aggregate *see section 3.2.16.1*	Required Repeating	The amounts of the credit transaction and appropriate fees.
\<DepAcctId\>	Aggregate *see section 3.2.7.1.2*	Required XOR	Deposit Account Identification Aggregate.
\<CardAcctId\>	Aggregate *see section 3.2.7.1.3*	Required XOR	Card Account Identification Aggregate.

6.3.28.2 Credit Authorization Status Aggregate \<CreditAuthStatus\>

The \<CreditAuthStatus\> aggregate contains details of the status of a credit authorization object.

Tag	Type	Usage	Description
\<CreditStatusCode\>	Open Enum	Required	The current status of the credit authorization object. Defined Values: Authorized, Rejected, Posted, Held
\<EffDt\>	DateTime	Required	Credit Status Date. The date associated with the state change to the current state.
\<StatusModBy\>	Open Enum	Optional	Status Modified By. Defined values: BPP, BPPSR, BSP, BSPSR, CPP, CSP, CSPSR, Customer, FI

6.3.29   Credit Information Aggregate \<CreditInfo\>

The \<CreditInfo\> aggregate is used in messages related to credits. It is generally used in the request messages.

Tag	Type	Usage	Description

Tag	Type	Usage	Description
\<CreditAuthType\>	Open Enum	Required	The type of the debit authorization transaction.
			Defined Values:
			UnverifiedCashDeposit, VerifiedCashDeposit, UnverifiedCheckDeposit, VerifiedCheckDeposit, EnvelopeDeposit, MultiDeposit, MerchandiseReturn
			This element will be deprecated from the CreditInfo aggregate in IFX 2.0. The replacement is to use the CreditType element.
\<CreditType\>	Open Enum	Optional *but see Description*	The type of the credit transaction.
			Defined Values:
			UnverifiedCashDeposit, VerifiedCashDeposit, UnverifiedCheckDeposit, VerifiedCheckDeposit, EnvelopeDeposit, MultiDeposit, MerchandiseReturn.
			Since this element will be replacing CreditAuthType in the CreditInfo aggregate, this element will be required in IFX 2.0.
\<CompositeCurAmt\>	Aggregate *see section 3.2.16.1*	Required Repeating	The amounts of the credit transaction and appropriate fees.
\<DepAcctId\>	Aggregate *see section 3.2.7.1.2*	Required XOR	Deposit Account Identification Aggregate.
\<CardAcctId\>	Aggregate *see section 3.2.7.1.3*	Required XOR	Card Account Identification Aggregate.
\<CreditAuthId\>	Identifier	Optional	Credit Authorization Identifier. Assigned by the server at the time the Credit Authorization is first added. This is used to relate a credit to a previously authorized credit authorization object
\<ClientChgCode\>	Open Enum	Optional	Client Change Code. This indicates the reason why a Credit was for a different amount from the original authorization for the credit.
			Defined Values:
			None, ConsumerCancelled, TerminalExceptionAmountKnown, TerminalExceptionAmountUnknown, ConsumerExceptionAmountKnown, ConsumerExceptionAmountUnknown
\<CreditMediaItem\>	Aggregate *see section 6.3.4*	Optional Repeating	Credit Media Item aggregate. This aggregate contains details pertaining to the individual items in the deposit (e.g. check, cash).
\<NetworkTrnInfo\>	Aggregate *see section 3.2.18*	Optional	Network Transaction Information. Identification and location of the terminal from which the message originates.
			This aggregate will be removed from the CreditInfo aggregate in IFX 2.0. With IFX 1.4+, the \<NetworkTrnInfo\> aggregate is available in the \<MsgRqHdr\> aggregate.

6.3.30 Deposit Application Information Aggregate \<DepAppInfo\>

The \<DepAppInfo\> aggregate contains information about a deposit application.

Tag	Type	Usage	Description

Tag	Type	Usage	Description
<DepApplicant>	Aggregate *see section 6.3.30.2*	Required Repeating	Deposit Applicant Aggregate.
<DepAppAcctId>	Aggregate *see section 6.3.30.1*	Required	Deposit Account Identification Aggregate.
<BankAcctInfo>	Aggregate *see section 3.2.7.1.5*	Required	Bank Account Aggregate.
<CardType>	Open Enum	Optional	Card Type. Defined value: ATM, Debit
<OverdraftFunding>	Aggregate *see section 6.3.30.3*	Optional Repeating	Overdraft Funding Aggregate.
<CreatedDt>	DateTime	Optional	Application Creation Date.

6.3.30.1 Deposit Application Account Id Aggregate <DepAppAcctId>

The <DepAppAcctId> aggregate contains data about the assigned account number for the deposit application. The <DepAppAddRq> message may send an account number in this aggregate indicating that an account starter kit was issued, or the <DepAppAddRs> message may send an account number for the pending account.

Tag	Type	Usage	Description
<AcctId>	Identifier	Optional	Account Identifier. Sometimes known as account number but not restricted to numeric characters. **Note:** *An International Bank Account Number (IBAN) can be used here to replace the domestic account number.*
<AcctType>	Open Enum	Required	Account Type. See Data Dictionary for details. Defined values: DDA, SDA, CDA, MMA, CMA
<AcctKey>	NC-22	Optional	Account Key. Checksum for international banks.
<AcctCur>	NC-22	Optional	Account Currency. Currency of the account. This may be necessary to uniquely identify the account, as many countries allow for a single account containing multiple subaccounts, each in a different currency.
<BankInfo>	Aggregate *see section 3.2.7.1.1*	Optional	Bank Information aggregate.

6.3.30.2 Deposit Applicant Aggregate <DepApplicant>

Information about the applicant applying for a a deposit account.

Tag	Type	Usage	Description
<CustId>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. Use this ID if the applicant is an existing bank customer. This information supersedes the CustId located in the message header.
<CustInfo>	Aggregate *see section 5.2.3.1.1*	Optional	Customer Information Aggregate.

Tag	Type	Usage	Description
<DepApplicantAcctRel>	Open Enum	Optional Repeating	Deposit Applicant Account Relationship. Relationship the applicant has the the account.
			Defined values:
			Primary, Secondary, Tax Responsible, Statement Mailing, Phone Contact, Custodial, Trust, Doing Business As.

6.3.30.3 Overdraft Funding Aggregate <OverdraftFunding>

Funding account used for overdraft protection.

Tag	Type	Usage	Description
<DepAcctId>	Aggregate *see section 3.2.7.1.2*	Required XOR	Deposit Account Identification Aggregate.
<LoanAcctId>	Aggregate *see section 3.2.7.1.4*	Required XOR	Loan Account Identification Aggregate.
<FundingPriority>	NC-3	Required	Funding Priority. Funding Priority for the account.

6.3.30.4 Interest Rate Information Aggregate <IntRateInfo>

Interest rate information.

Tag	Type	Usage	Description
<Rate>	Decimal	Required	Interest Rate. Usage is a percentage. (e.g. – a value of 5.2 = 5.2%)
<Desc>	C-80	Optional	Short Description. Explanatory text associated with the interest rate code. Assigned by the financial institution.
<IntAPY>	Decimal	Optional	Annual Percentage Yield. Usage is a percentage of yield on an annualized basis.
<Term>	Aggregate *see section 3.2.9.1*	Optional	Term Aggregate
<LowCurAmt>	Currency Amount	Optional	Interest Rate Low Amount.
<HighCurAmt>	Currency Amount	Optional	Interest Rate High Amount.

6.3.31   Deposit Application Record Aggregate <DepAppRec>

The <DepAppRec> aggregate contains the deposit application record.

Tag	Type	Usage	Description
<DepAppId>	Identifier	Required	Deposit Application Identifier. Assigned by the server at the time the application is added.
<DepAppInfo>	Aggregate *see section 6.3.30*	Required	Deposit Application Information Aggregate.
<DepAppStatus>	Aggregate *see section 6.3.31.1*	Optional	Deposit Application Status Aggregagte.

6.3.31.1 Deposit Application Status <DepAppStatus>

The <DepAppStatus> contains the status of the Deposit Application Record.

Tag	Type	Usage	Description
<DepAppStatusCode>	Closed Enum	Required	Deposit Application Status Code.
			Valid values:
			Pending, Complete, Cancelled, Feature Pending, In Work, Incomplete, New Customer Pending, Post Failed, Transmitting
<StatusDesc>	C-255	Optional	Status Description. Explanatory text associated with this status.
<EffDt>	DateTime	Required	Deposit Application Status Date. The date associated with the state change to the current state.
<StatusModBy>	Open Enum	Optional	Status Mod By.
			Defined values:
			BPP, BPPSR, BSP, BSPSR, CPP, CSP, CSPSR, Customer, FI

6.3.32 Payment Enclosed Information Aggregate <PmtEnclInfo>

The <PmtEnclInfo> aggregate is used in Payment Enclosed transaction messages. It is generally used in the request messages.

Tag	Type	Usage	Description
<PmtEnclType>	Open Enum	Required	The type of the Payment Enclosed transaction.
			Defined Values:
			UnverifiedCashDeposit, VerifiedCashDeposit, UnverifiedCheckDeposit, VerifiedCheckDeposit, EnvelopeDeposit, MultiDeposit
<PmtEnclPayeeInfo>	Aggregate *see section 6.3.32.1*	Optional *but see description*	Payment Enclosed Payee Information Aggregate.
			This aggregate is required if the <PmtEnclType> is other than "EnvelopeDeposit".
<CreditMediaItem>	Aggregate *see section 6.3.4*	Optional Repeating	Credit Media Item aggregate. This aggregate contains details pertaining to the individual items in the enclosed payment (e.g. check, cash).
<CompositeCurAmt>	Aggregate *see section 3.2.16.1*	Required Repeating	The amounts of the Payment Enclosed transaction and any appropriate fees.
<Category>	C-40	Optional	Payment Category. The payment category selected by the customer.
<PrcDt>	Date	Optional XOR Profiled support	Payment Processing Date. The customer enters this date as the original requested processing date, not corrected for holidays and non-processing days. May be modified by the customer.
<DueDt>	Date	Optional XOR Profiled support	Payment Due Date. The customer enters this date to represent the date the payment is due to arrive at the payee. May be modified by the customer.

6.3.32.1 Payment Enclosed Payee Information Aggregate <PmtEnclPayeeInfo>

The <PmtEnclPayeeInfo> aggregate is used in <PmtEnclInfo> aggregate to identify the payee (when the <PmtEnclType> is other than "EnvelopeDeposit").

Tag	Type	Usage	Description
<StdPayeeId>	Aggregate *see section 7.3.1.1*	Required XOR	Customer's Payee Identifier. The Standard Payee as known to the CPP.
<PmtEnclPayee>	Aggregate	Required XOR	Payee's Name and Address Aggregate.
<Name>	C-40	Required	Payee Name.
<PostAddr>	Aggregate *see section 3.2.3.7.1*	Required	Payee Address Aggregate.
</PmtEnclPayee>			
<IndustId>	Aggregate *see section 3.2.13*	Optional	Industry Identifier. It provides standard codes by which industries are classified.

6.3.32.2 Payment Enclosed Record Aggregate <PmtEnclRec>

The <PmtEnclRec> aggregate is used in Payment Enclosed transaction messages. It is generally used in the response messages.

Tag	Type	Usage	Description
<PmtEnclId>	Identifier	Required	Payment Enclosed Identifier. Assigned by the server at the time the Payment Enclosed is first added.
<PmtEnclInfo>	Aggregate *see section 6.3.32*	Required	Payment Enclosed Information aggregate.
<PmtEnclStatus>	Aggregate *see section 6.3.32.3*	Required	Payment Enclosed status aggregate.

6.3.32.3 Payment Enclosed Status Aggregate <PmtEnclStatus>

The <PmtEnclStatus> is returned in responses to Add or Modify Payment Enclosed Model. Note that all elements within this aggregate are assigned by the server and cannot be assigned or modified by the client.

Tag	Type	Usage	Description
<PmtEnclStatusCode>	Closed Enum	Required	Payment Enclosed Status Code. This identifies the payment enclosed processing status.
			Valid values: Cancelled, Failed, AcceptedToVerify, Verified
<StatusDesc>	C-255	Optional	Status Description. Explanatory text associated with this status.
<EffDt>	DateTime	Optional	Payment Enclosed Status Date. The date associated with the state change to the current state.
<StatusModBy>	Open Enum	Optional	Status Modified By.
			Defined values: BPP, BPPSR, BSP, BSPSR, CPP, CSP, CSPSR, Customer, FI

6.3.33 Deposit Acount Statement Record <DepAcctStmtRec>

The <DepAcctStmtRec> contains deposit account statement information.

Tag	Type	Usage	Description
<DepAcctStmtRec>	Aggregate *see section 6.3.33*	Optional Repeating	Deposit Account Statement Record.
<SPRefId>	Identifier	Optional	Service Provider Reference Identifier.
			This is a reference number for the statement.
<DepAcctId>	Aggregate *see section 3.2.7.1.2*	Optional	Deposit Account Identification Aggregate. This aggregate refers to an account that is a child of the account referenced in the request.
<DepAcctStmtId>	Identifier	Optional	Deposit Account Statement Identifier. Statement number; unique number assigned to a statement per account per customer.
<EffDt>	DateTime	Optional	Effective Date. Date of statement; date when statement was generated.
<NextDt>	DateTime	Optional	Next Closing DateTime.
<AcctBal>	Aggregate *see section 3.2.8.1*	Required Repeating	Account Balance Aggregate.
			May be used to provide the opening, closing and minimum ledger balance and any other balances to be provided on the statement. The ClosingLedger must be included.
<StmtSummAmt>	Aggregate	Optional Repeating	Statement Summary Amount Aggregate. Used to return all the statement summary totals for this closing statement period.
<StmtSummType>	Open Enum	Required	Summary Types. Used to identify the type of summary data.
			Defined Values: Deposits, OthCredits, Checks, OthDebits, Fees, IntCharged, IntEarned, ATM, Electronic, CreditsOnly, DebitsOnly, BAI:xxx or TMA:xxx (Reference BAI Code List at http://www.bai.org/operations/bai_codes.html, or TMA Code List at http://www.AFPonline.org/Information_Center/Publications/Bookstore/afpbooks/afpbooks.html as maintained by Association of Financial Professionals (AFP)) Example: BAI:010 (Beginning Ledger Balance), BAI:140 (ACH Credits), etc.
<CurAmt>	Currency Amount	Required	Summary Amount.
<Count>	Long	Optional	Count. Summary Item Count. Example: This count would indicate how many checks should be contained in the detail record <DepAcctTrnRec>.
</StmtSummAmt>			
<StartDt>	DateTime	Required	Selection Start Date for this statement.
<EndDt>	DateTime	Required	End for this statement.
<MktgInfo>	C-255	Optional	Marketing Information.
<URL>	URL	Optional	Contains bank-rendered statement copy.
<DepAcctTrnRec>	Aggregate *see section 6.3.14*	Optional Repeating	Deposit Account Transaction Record aggregate. Included if the <IncDetail> Boolean is set to True in the request.
			One record per message for this closing statement period.

6.3.34   Check Issue Information Aggregate <ChkIssueInfo>

The <ChkIssueInfo> aggregate contains the details of an issued check.

Tag	Type	Usage	Description
<DepAcctId>	Aggregate *see section 3.2.7.1.2*	Required	Deposit Account Identification Aggregate.
<EffDt>	DateTime	Optional	Effective Date. The date associated with the origination of the Check Issue file.
<RefInfo>	Aggregate *see section 3.2.17*	Optional Repeating	Additional Reference information to uniquely identify the bank. Defined values for contained <RefType>: CountrySpecific, BankSpecific
<ChkInfo>	Aggregate	Optional	Check information.
<ChkNum>	C-40	Optional	Check number. Assigned by the Payer or CSP or CPP if payment is by paper check.
<OrigDt>	Date	Optional	Check origination date. The date on which the check is originated.
<Name>	C-40	Optional	Name of payer to appear on check (as originator). This can be Paying organization (Payer or Invoice Receiver) or check originator (CSP or CPP).
<PostAddr>	Aggregate *see section 3.2.3.7.1*	Optional	Postal address of the Compositeany to appear on check (as originator). This can be Paying Compositeany (Payer or Invoice Receiver) or check originator (CSP or CPP) postal address.
<OrgPhone>	Phone Number	Optional	Phone number of the organization to appear on check (as originator). This can be Paying organization (Payer or Invoice Receiver) or check originator (CSP or CPP) phone number.
<Memo>	C-255	Optional	Text description on the check.
</ChkInfo>			
<CurAmt>	Currency Amount	Required	Currency Amount. Check currency amount
<DueDt>	Date	Optional	Check Due Date. The date on which the check is due to pay.
<PaidDt>	Date	Optional	Check Origination Date. The date on which the customer originates the check.
<ImageURL>	URL	Optional	URL address for retrieving an image of the Compositelete bill encoded as HTML. This may be cached by the client for later display, or it may be viewed live directly from the Web.

6.3.34.1   Check Issue Status Code Aggregate <ChkIssueStatus>

Tag	Type	Usage	Description
<ChkIssueStatusCode>	Closed Enum	Required	Check Issue Status Code. Values: Pending, Paid, Rejected, Pay, NoPay
<StatusDesc>	C-255	Optional	Status Description. Explanatory text associated with this check issue status.
<EffDt>	DateTime	Optional	Effective Date/Time. The date/time the <ChkIssueStatusCode> took effect.

Tag	Type	Usage	Description
<StatusModBy>	Open Enum	Optional	Status Mod By. If present, indicates who modified the Check Issue Status Code.
			Values:
			CSP, FI

6.3.34.2   Check Issue Record Aggregate <ChkIssueRec>

Tag	Type	Usage	Description
<ChkIssueId>	Aggregate	Required	Check Issue Identifier. Typically a combination of the deposit account ID and the check number.
<ChkNum>	C-40	Optional	Check number. Assigned by the Payer or CSP or CPP if payment is by paper check.
<OrigDt>	Date	Optional	Check origination date. The date on which the check is originated.
<AcctId>	Identifier	Optional	Account Identifier. Sometimes known as account number, but not restricted to numeric characters.
			Note: *An International Bank Account Number (IBAN) can be used here to replace the domestic account number.*
</ChkIssueId>			
<ChkIssueInfo>	Aggregate *see section 6.3.34*	Required	Check Issue Information Aggregate
<ChkIssueStatus>	Aggregate *see section 6.3.34.1*	Required	Check Issue Status Aggregate

6.3.35   Reversal Request Information Aggregate <RevRqInfo>

Tag	Type	Usage	Description
<OrigRqUID>	UUID	Required	Original Request Identifier of the message to reverse.
<RevReasonCode>	Open Enum	Optional	Reversal reason code.
<Desc>	C-80	Optional	Description, text describing the reason for the reversal.

6.4   Statement and Account Inquiry

6.4.1   Balance Inquiry

A client may use the <BalInqRq> message to retrieve account balances without requesting message detail or a statement.

6.4.1.1   Request <BalInqRq>

The client specifies only the account for which to retrieve balances.

Tag	Type	Usage	Description
<RqUID>	UUID	Required	Request Identifier.

Tag	Type	Usage	Description
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header aggregate.
<AsyncRqUID>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously. For more information, see section 3.2.12.1.
<CustId>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<DepAcctId>	Aggregate *see section 3.2.7.1.2*	Required XOR	Deposit Account Identification Aggregate.
<CardAcctId>	Aggregate *see section 3.2.7.1.3*	Required XOR	Card Account Identification Aggregate.
<LoanAcctId>	Aggregate *see section 3.2.7.1.4*	Required XOR	Loan Account Identification Aggregate.
<IncExtBal>	Boolean	Optional	Include Extended Balances Indicator. If *True*, the response should also include the <ExtAcctBal> aggregate and return all available balances for the type of account. If *False* or omitted, the response should only include the standard balances for the account in <AcctBal>.
<DeliveryMethod>	Open Enum	Optional Profiled values	Delivery Method. Default is Channel. Value must be supported in Service Profile.

6.4.1.2   Response <BalInqRs>

Tag	Type	Usage	Description
<Status>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, <StatusCode> defaults to 0 (zero).
<RqUID>	UUID	Required Echoed	Request Identifier.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header aggregate.
<MsgRsHdr>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header aggregate.
<AsyncRqUID>	UUID	Optional Echoed	Asynchronous Request Identifier.

Tag	Type	Usage	Description
<CustId>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<DepAcctId>	Aggregate *see section 3.2.7.1.2*	Required XOR Echoed	Deposit Account Identification Aggregate.
<CardAcctId>	Aggregate *see section 3.2.7.1.3*	Required XOR Echoed	Card Account Identification Aggregate.
<LoanAcctId>	Aggregate *see section 3.2.7.1.4*	Required XOR Echoed	Loan Account Identification Aggregate.
<IncExtBal>	Boolean	Optional Echoed	Include Extended Balances Indicator.
<DeliveryMethod>	Open Enum	Optional Echoed	Delivery Method.
<AcctBal>	Aggregate *see section 3.2.8.1*	Required Repeating	Account Balance Aggregate.
<ExtAcctBal>	Aggregate *see section 3.2.8.2*	Optional Repeating	Extended Account Balance Aggregate.
<MktgInfo>	C-255	Optional	Marketing Information.

6.4.2 Balance Reversal

6.4.2.1 Request <BalRevRq>

Tag	Type	Usage	Description
<RqUID>	UUID	Required	Request Identifier.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header aggregate.
<AsyncRqUID>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously. For more information, see section 3.2.12.1.
<CustId>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<RevReasonCode>	Open Enum	Required	Reversal reason code. The reason why the client is requesting a reversal.
<Desc>	C-80	Optional	Description, text describing the reason for the reversal.

Tag	Type	Usage	Description
<BalMsgRqInfo>	Aggregate	Required	Balance Message Request Information Aggregate. This aggregate contains the message to reverse. See RevOptSupt in the service profile for specific reversal messages supported.
<BalInqRq>	Aggregate *see section 6.4.1.1*	Optional	Balance Inquiry Request Message Aggregate.
</BalMsgRqInfo>			

6.4.2.2   Response <BalRevRs>

Tag	Type	Usage	Description
<Status>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, <StatusCode> defaults to 0 (zero).
<RqUID>	UUID	Required Echoed	Request Identifier.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header aggregate.
<MsgRsHdr>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header aggregate.
<AsyncRqUID>	UUID	Optional Echoed	Asynchronous Request Identifier.
<CustId>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Identification Aggregate.
<RevReasonCode>	Open Enum	Required Echoed	Reversal reason code. The reason why the client is requesting a reversal.
<Desc>	C-80	Optional Echoed	Description, text describing the reason for the reversal.
<BalMsgRqInfo>	Aggregate	Required Echoed	Balance Message Request Information Aggregate. This aggregate contains the message to reverse. See RevOptSupt in the service profile for specific reversal messages supported.
<BalInqRq>	Aggregate *see section 6.4.1.1*	Optional	Balance Inquiry Request Message Aggregate.
</BalMsgRqInfo>			
<CSPRefId>	Identifier	Optional	Customer Service Provider Reference Identifier.
<SPRefId>	Identifier	Optional	Service Provider Reference Identifier.

6.4.3   Account Inquiry

A client uses the Account Inquiry Message to retrieve more detailed information about a bank account than that provided by the Balance Inquiry Message. Much of the information is returned in an aggregate that is specific to each Account Type.

6.4.3.1   Request <AcctInqRq>

Tag	Type	Usage	Description

Tag	Type	Usage	Description
<RqUID>	UUID	Required	Request Identifier.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header aggregate.
<AsyncRqUID>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously. For more information, see section 3.2.12.1.
<CustId>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<DepAcctId>	Aggregate *see section 3.2.7.1.2*	Required XOR	Deposit Account Identification Aggregate.
<CardAcctId>	Aggregate *see section 3.2.7.1.3*	Required XOR	Card Account Identification Aggregate.
<LoanAcctId>	Aggregate *see section 3.2.7.1.4*	Required XOR	Loan Account Identification Aggregate.
<IncExtBal>	Boolean	Optional	Include Extended Balances Indicator. If *True*, the response should include the <ExtAcctBal> aggregate and return all available balances for the type of account. If *False* or omitted, the response should not include balances.
<IncBal>	Boolean	Optional	Include Balances Indicator. If *True*, the response should include the <AcctBal> and <ExtAcctBal> aggregate and return all available balances for the type of account. If *False* or omitted, the response should not include balances.
<DeliveryMethod>	Open Enum	Optional Profiled values	Delivery Method. Default is Channel. Value must be supported in Service Profile.

6.4.3.2   Response <AcctInqRs>

Tag	Type	Usage	Description
<Status>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, <StatusCode> defaults to 0 (zero).
<RqUID>	UUID	Required Echoed	Request Identifier.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header aggregate.
<MsgRsHdr>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header aggregate.
<AsyncRqUID>	UUID	Optional Echoed	Asynchronous Request Identifier.

Tag	Type	Usage	Description
<CustId>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<DepAcctId>	Aggregate *see section 3.2.7.1.2*	Required XOR Echoed	Deposit Account Identification Aggregate.
<CardAcctId>	Aggregate *see section 3.2.7.1.3*	Required XOR Echoed	Card Account Identification Aggregate.
<LoanAcctId>	Aggregate *see section 3.2.7.1.4*	Required XOR Echoed	Loan Account Identification Aggregate.
<IncExtBal>	Boolean	Optional Echoed	Include Extended Balances Indicator.
<IncBal>	Boolean	Optional Echoed	Include Balances Indicator.
<DeliveryMethod>	Open Enum	Optional Echoed	Delivery Method.
<UpDt>	Timestamp	Required	Account Update Timestamp.
<LastTrnDt>	Date	Optional	Last Transaction Date.
<LastStmtDt>	Date	Optional	Last Statement Cycle Date.
<AcctBal>	Aggregate *see section 3.2.8.1*	Optional Repeating	Account Balance Aggregate
<ExtAcctBal>	Aggregate *see section 3.2.8.2*	Optional Repeating	Extended Account Balance Aggregate.
<DepAcctRec>	Aggregate *see section 6.3.7*	Optional XOR	Deposit Account Record Aggregate.
<CCAcctRec>	Aggregate *see section 6.3.8*	Optional XOR	Credit Card Account Record Aggregate.
<LoanAcctRec>	Aggregate *see section 6.3.10*	Optional XOR	Loan Account Record Aggregate.
<LOCAcctRec>	Aggregate *see section 6.3.11*	Optional XOR	Line of Credit Account Record Aggregate.
<MortAcctRec>	Aggregate *see section 6.3.12*	Optional XOR	Mortgage Account Record Aggregate.

6.4.4 Account Reversal

6.4.4.1 Request <AcctRevRq>

Tag	Type	Usage	Description

Tag	Type	Usage	Description
<RqUID>	UUID	Required	Request Identifier.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header aggregate.
<AsyncRqUID>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously.
			For more information, see section 3.2.12.1.
<CustId>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<RevReasonCode>	Open Enum	Required	Reversal reason code. The reason why the client is requesting a reversal.
<Desc>	C-80	Optional	Description, text describing the reason for the reversal.
<AcctMsgRqInfo>	Aggregate	Required	Account Message Request Information Aggregate. This aggregate contains the message to reverse. See RevOptSupt in the service profile for specific reversal messages supported.
<AcctInqRq>	Aggregate *see section 6.4.3.1*	Optional	Account Inquiry Request Message Aggregate.
</AcctMsgRqInfo>			

6.4.4.2   Response <AcctRevRs>

Tag	Type	Usage	Description
<Status>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, <StatusCode> defaults to 0 (zero).
<RqUID>	UUID	Required Echoed	Request Identifier.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header aggregate.
<MsgRsHdr>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header aggregate.
<AsyncRqUID>	UUID	Optional Echoed	Asynchronous Request Identifier.
<CustId>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Identification Aggregate.
<RevReasonCode>	Open Enum	Required Echoed	Reversal reason code.
<Desc>	C-80	Optional Echoed	Description, text describing the reason for the reversal.

Tag	Type	Usage	Description
<AcctMsgRqInfo>	Aggregate	Required Echoed	Account Message Request Information Aggregate. This aggregate contains the message to reverse. See RevOptSupt in the service profile for specific reversal messages supported.
<AcctInqRq>	Aggregate *see section 6.4.3.1*	Optional	Account Inquiry Request Message Aggregate.
</AcctMsgRqInfo>			
<CSPRefId>	Identifier	Optional	Customer Service Provider Reference Identifier.
<SPRefId>	Identifier	Optional	Service Provider Reference Identifier.

6.4.5   Deposit Account Statement Inquiry

A client may use <DepAcctStmtInqRq> to retrieve account information for a statement period or multiple statement periods. A client may optionally provide a date range to limit the number of Account Statement aggregates that are returned in the response. Note that the statement is based on the normal cutoff cycles and the date range will return statements that *ended* (cutoff or cycled) during the specified range. The client may specify a date range that results in no statement, because there were no statement end cutoffs within the range. If the client does not specify a date range, the server returns as many statements as possible.

The client may request the detail messages associated with each returned Closing Statement by specifying *True* in the <IncDetail> Boolean.

6.4.5.1   Request <DepAcctStmtInqRq>

Tag	Type	Usage	Description
<RqUID>	UUID	Required	Request Identifier.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header aggregate.
<AsyncRqUID>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously. For more information, see section 3.2.12.1.
<CustId>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<RecCtrlIn>	Aggregate *see section 3.2.12.2.1*	Optional	Records Control Input Aggregate.
<DepAcctId>	Aggregate *see section 3.2.7.1.2*	Required XOR	Deposit Account Identification Aggregate.
<CardAcctId>	Aggregate *see section 3.2.7.1.3*	Required XOR	Card Account Identification Aggregate.
<SelRangeDt>	Aggregate *see section 3.2.10.1*	Optional	Selection Range Date Aggregate.

Tag	Type	Usage	Description
<IncDetail>	Boolean	Optional	Include Detail Indicator. If *True*, the response should include the detail statement messages <DepStmtTrnRec> for the statements returned. If *False* or omitted, the detail messages should not be included.
<DeliveryMethod>	Open Enum	Optional Profiled values	Delivery Method. Default is Channel. Value must be supported in Service Profile.
<StmtType>	Open Enum	Optional	Statement type. Defined values: Partial, Legal

6.4.5.2   Response <DepAcctStmtInqRs>

Tag	Type	Usage	Description
<Status>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, <StatusCode> defaults to 0 (zero).
<RqUID>	UUID	Required Echoed	Request Identifier.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header aggregate.
<MsgRsHdr>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header aggregate.
<AsyncRqUID>	UUID	Optional Echoed	Asynchronous Request Identifier.
<CustId>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<RecCtrlOut>	Aggregate *see section 3.2.12.2.2*	Optional *but see Description*	Records Control Output Aggregate. Required if <RecCtrlIn> was provided in the request and the server supports Records Control.
<DepAcctId>	Aggregate *see section 3.2.7.1.2*	Required XOR Echoed	Deposit Account Identification Aggregate.
<CardAcctId>	Aggregate *see section 3.2.7.1.3*	Required XOR Echoed	Card Account Identification Aggregate.
<SelRangeDt>	Aggregate *see section 3.2.10.1*	Optional Echoed	Selection Range Date Aggregate.
<IncDetail>	Boolean	Optional Echoed	Include Detail Indicator.
<StmtType>	Open Enum	Optional Echoed	Statement type
<DeliveryMethod>	Open Enum	Optional Echoed	Delivery Method.

Tag	Type	Usage	Description
<Fee>	Aggregate *see section 3.2.16.2*	Optional Repeating	The fees charged for this transaction.
<DepAcctStmtRec>	Aggregate *see section 6.3.33*	Optional Repeating	Deposit Account Statement Record.

6.4.6   Deposit Account Statement Reversal

6.4.6.1   Request <DepAcctStmtRevRq>

Tag	Type	Usage	Description
<RqUID>	UUID	Required	Request Identifier.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header aggregate.
<AsyncRqUID>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously.
			For more information, see section **3.2.12.1**.
<CustId>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<RevReasonCode>	Open Enum	Required	Reversal reason code. The reason why the client is requesting a reversal.
<Desc>	C-80	Optional	Description, text describing the reason for the reversal.
<DepAcctStmtMsgRqInfo>	Aggregate	Required	Deposit Account Statement Message Request Information Aggregate. This aggregate contains the message to reverse. See RevOptSupt in the service profile for specific reversal messages supported.
<DepAcctStmtInqRq>	Aggregate *see section 6.4.5.1*	Optional	Deposit Account Statement Inquiry Request Message Aggregate.
</DepAcctStmtMsgRqInfo>			

6.4.6.2   Response <DepAcctStmtRevRs>

Tag	Type	Usage	Description
<Status>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, <StatusCode> defaults to 0 (zero).
<RqUID>	UUID	Required Echoed	Request Identifier.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header aggregate.

Tag	Type	Usage	Description
<MsgRsHdr>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header aggregate.
<AsyncRqUID>	UUID	Optional Echoed	Asynchronous Request Identifier.
<CustId>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Identification Aggregate.
<RevReasonCode>	Open Enum	Required Echoed	Reversal reason code. The reason why the client is requesting a reversal.
<Desc>	C-80	Optional Echoed	Description, text describing the reason for the reversal.
<DepAcctStmtMsgRqInfo>	Aggregate	Required Echoed	Deposit Account Statement Message Request Information Aggregate. This aggregate contains the message to reverse. See RevOptSupt in the service profile for specific reversal messages supported.
<DepAcctStmtInqRq>	Aggregate *see section 6.4.5.1*	Optional	Deposit Account Statement Inquiry Request Message Aggregate.
</DepAcctStmtMsgRqInfo>			
<CSPRefId>	Identifier	Optional	Customer Service Provider Reference Identifier.
<SPRefId>	Identifier	Optional	Service Provider Reference Identifier.

6.4.7   *Deposit Account Statement Advise*

DepAcctStmtAdvise message is used to advise an entity of account information for a statement period.

6.4.7.1   Request < DepAcctStmtAdviseRq >

Tag	Type	Usage	Description
<RqUID>	UUID	Required	Request Identifier. Sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header Aggregate.
<CustId>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<DepAcctStmtRec>	Aggregate *see section 6.3.33*	Optional Repeating	Deposit Account Statement Record.

6.4.7.2   Response < DepAcctStmtAdviseRs >

Tag	Type	Usage	Description
<Status>	Aggregate *see section 3.2.12.1*	Required	Response Status Aggregate. If this aggregate is absent, <StatusCode> defaults to 0 (zero).

\<RqUID\>	UUID	Required Echoed	Request Identifier. Sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.
\<MsgRqHdr\>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header Aggregate.
\<CustId\>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in \<SignonRq\>. For example, if a CSR or SP issues the request on behalf of the user, then \<CustId\> is required, and must contain the value of the user whose request is being issued.
\<DepAcctStmtRec\>	Aggregate *see section 6.3.33*	Required Echoed	Deposit Account Statement Record.
\<CSPRefId\>	Identifier	Optional Repeating	Customer Service Provider Reference Identifier. \<CSPRefId\> is used to inquire about a payment corresponding to a confirmation number that was returned to the client when the payment was added or modified. When a payment has been modified, only the \<CSPRefId\> received in the most recent \<DepAcctStmtAdvise\> is valid.
\<SPRefId\>	Identifier	Optional Repeating	Service Provider Reference Identifier. Same usage as \<CSPRefId\>.

6.4.8   *Credit Card Statement Inquiry*

A client may use \<CCAcctStmtInqRq\> to retrieve account information for a statement period or multiple statement periods. A client may optionally provide a date range to limit the number of Account Statement aggregates that are returned in the response. Note that the statement is based on the normal cutoff cycles and the date range must return statements that *ended* (cutoff or cycled) during the specified range. The client may specify a date range that results in no statement, because there were no statement end cutoffs within the range. If the client does not specify a date range, the server returns as many statements as possible.

The client may request the detail messages associated with each returned Statement by specifying *True* in the \<IncDetail\> Boolean.

6.4.8.1   Request \<CCAcctStmtInqRq\>

Tag	Type	Usage	Description
\<RqUID\>	UUID	Required	Request Identifier.
\<MsgRqHdr\>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header aggregate.
\<AsyncRqUID\>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously. For more information, see section 3.2.12.1.
\<CustId\>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in \<SignonRq\>. For example, if a CSR or SP issues the request on behalf of the user, then \<CustId\> is required, and must contain the value of the user whose request is being issued.

Tag	Type	Usage	Description
\<RecCtrlIn\>	Aggregate *see section 3.2.12.2.1*	Optional	Records Control Input Aggregate.
\<CardAcctId\>	Aggregate *see section 3.2.7.1.3*	Required	Card Account Identifier Aggregate.
\<SelRangeDt\>	Aggregate *see section 3.2.10.1*	Optional	Selection Range Date Aggregate.
\<IncDetail\>	Boolean	Optional	Include Detail Indicator. If *True*, the response should include the detail statement messages \<CCAcctTrnRec\> for the statements returned. If *False* or omitted, the detail messages should not be included.
\<DeliveryMethod\>	Open Enum	Optional Profiled values	Delivery Method. Default is Channel. Value must be supported in Service Profile.
\<StmtType\>	Open Enum	Optional	Statement type Defined values: Partial, Legal

6.4.8.2   Response \<CCAcctStmtInqRs\>

Tag	Type	Usage	Description
\<Status\>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, \<StatusCode\> defaults to 0 (zero).
\<RqUID\>	UUID	Required Echoed	Request Identifier.
\<MsgRqHdr\>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header aggregate.
\<MsgRsHdr\>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header aggregate.
\<AsyncRqUID\>	UUID	Optional Echoed	Asynchronous Request Identifier.
\<CustId\>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in \<SignonRq\>. For example, if a CSR or SP issues the request on behalf of the user, then \<CustId\> is required, and must contain the value of the user whose request is being issued.
\<RecCtrlOut\>	Aggregate *see section 3.2.12.2.2*	Optional *but see Description*	Records Control Output Aggregate. Required if \<RecCtrlIn\> was provided in the request and the server supports Records Control.
\<CardAcctId\>	Aggregate *see section 3.2.7.1.3*	Required Echoed	Card Account Identifier Aggregate.
\<SelRangeDt\>	Aggregate *see section 3.2.10.1*	Optional Echoed	Selection Range Date Aggregate.

Tag	Type	Usage	Description
\<IncDetail>	Boolean	Optional Echoed	Include Detail Indicator.
\<StmtType>	Open Enum	Optional Echoed	Statement type
\<DeliveryMethod>	Open Enum	Optional Echoed	Delivery Method.
\<CCAcctStmtRec>	Aggregate	Optional Repeating	Credit Card Account Statement Record
\<SPRefId>	Identifier	Optional	Service Provider Reference Identifier.
\<NextDt>	DateTime	Optional	Next Closing DateTime.
\<AcctBal>	Aggregate *see section 3.2.8.1*	Required Repeating	Account Balance Aggregate. The ClosingOutstanding balance is required.
\<ExtAcctBal>	Aggregate *see section 3.2.8.2*	Optional Repeating	Extended Account Balance Aggregate. Used to report other balances for this statement end, such as Period Fee, Cashline, Outstanding Cash Advance Total, Cash Available, and Over Limit Amount
\<DueDt>	Date	Optional	Payment Due Date.
\<MinAmtDue>	Currency Amount	Optional	Minimum Payment Due.
\<StmtSummAmt>	Aggregate	Optional Repeating	Statement Summary Amount Aggregate. Used to return all the statement summary totals for this closing statement period.
\<StmtSummType>	Open Enum	Required	Summary Types. Used to identify the type of summary data.
			Defined values: Deposits, OthCredits, Checks, OthDebits, Fees, IntCharged, IntEarned, ATM, Electronic, CreditsOnly, DebitsOnly.
\<CurAmt>	Currency Amount	Required	Summary Amount.
\<Count>	Long	Optional	Count. Summary Item Count.
			Example: This count would indicate how many debits or credits should be contained in the detail record \<CCAcctTrnRec>.
\</StmtSummAmt>			
\<DelinqAging>	Aggregate	Optional Repeating	Delinquency Aging Aggregate.
\<Aging>	Open Enum	Required	Number of Days Delinquent.
			Defined values: 0-30, 31-60, 61-90, 91-120, Over121.
\<CurAmt>	Currency Amount	Required	Amount Delinquent.
\</DelinqAging>			
\<StartDt>	DateTime	Required	Selection Start DateTime Provided to allow client to use these dates to retrieve message detail corresponding to this statement.
\<EndDt>	DateTime	Required	End DateTime. Provided to allow client to use these dates to retrieve message detail corresponding to this statement.
\<MktgInfo>	C-255	Optional Repeating	Marketing Information.
\<URL>	URL	Optional	Contains bank-rendered statement copy.

Tag	Type	Usage	Description
<CCAcctTrnRec> **</CCAcctStmtRec>**	Aggregate *see section 6.3.15*	Optional Repeating	Credit Card Transaction Record Aggregate. Included if the <IncDetail> Boolean is set to *True* in the request.

6.4.9 *Credit Card Statement Inquiry Reversal*

6.4.9.1 Request <CCAcctStmtRevRq>

Tag	Type	Usage	Description
<RqUID>	UUID	Required	Request Identifier.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header aggregate.
<AsyncRqUID>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously. For more information, see section **3.2.12.1**.
<CustId>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<RevReasonCode>	Open Enum	Required	Reversal reason code. The reason why the client is requesting a reversal.
<Desc>	C-80	Optional	Description, text describing the reason for the reversal.
<CCAcctStmtMsgRqInfo>	Aggregate	Required	Credit Card Statement Message Request Information Aggregate. This aggregate contains the message to reverse. See RevOptSupt in the service profile for specific reversal messages supported.
<CCAcctStmtInqRq> **</CCAcctStmtMsgRqInfo>**	Aggregate *see section 6.4.8.1*	Optional	Credit Card Statement Inquiry Request Message Aggregate.

6.4.9.2 Response <CCAcctStmtRevRs>

Tag	Type	Usage	Description
<Status>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, <StatusCode> defaults to 0 (zero).
<RqUID>	UUID	Required Echoed	Request Identifier.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header aggregate.

Tag	Type	Usage	Description
<MsgRsHdr>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header aggregate.
<AsyncRqUID>	UUID	Optional Echoed	Asynchronous Request Identifier.
<CustId>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Identification Aggregate.
<RevReasonCode>	Open Enum	Required Echoed	Reversal reason code. The reason why the client is requesting a reversal.
<Desc>	C-80	Optional Echoed	Description, text describing the reason for the reversal.
<CCAcctStmtMsgRqInfo>	Aggregate	Required Echoed	Credit Card Statement Message Request Information Aggregate. This aggregate contains the message to reverse. See RevOptSupt in the service profile for specific reversal messages supported.
<CCAcctStmtInqRq>	Aggregate *see section 6.4.8.1*	Optional	Credit Card Statement Inquiry Request Message Aggregate.
</CCAcctStmtMsgRqInfo>			
<CSPRefId>	Identifier	Optional	Customer Service Provider Reference Identifier.
<SPRefId>	Identifier	Optional	Service Provider Reference Identifier.

6.4.10   Deposit Account Transaction Inquiry

A client uses the Deposit Account Transaction Inquiry to retrieve transaction detail for a given account. A variety of selection criteria are supported.

6.4.10.1   Request <DepAcctTrnInqRq>

Tag	Type	Usage	Description
<RqUID>	UUID	Required	Request Identifier.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header aggregate.
<AsyncRqUID>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously. For more information, see section 3.2.12.1.
<CustId>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<RecCtrlIn>	Aggregate *see section 3.2.12.2.1*	Optional	Records Control Input Aggregate.
<DepAcctId>	Aggregate *see section 3.2.7.1.2*	Required XOR	Deposit Account Identification Aggregate.

Tag	Type	Usage	Description
\<CardAcctId>	Aggregate *see section 3.2.7.1.3*	Required XOR	Card Account Identification Aggregate.
\<SelRangeDt>	Aggregate *see section 3.2.10.1*	Optional	Selection Range Date Aggregate. Selection criteria are based upon message posting date.
\<SelRangeCurAmt>	Aggregate *see section 3.2.10.2*	Optional	Selection Range Amount Aggregate.
\<ChkRange>	Aggregate *see section 6.3.16*	Optional	Selection Range Check Number Aggregate.
\<TrnType>	Open Enum	Optional Repeating	Transaction Type. Defined values: Debit, Credit, Withdrawal, Check, Deposit, Transfer, Payment, Interest, Dividend, DirectDeposit, DirectDebit, RepeatPayment, Fee, ServiceCharge
\<TrnSrc>	Open Enum	Optional Repeating	Transaction Source. Defined values: ATM, Teller, POS, VRU, HomeBank, ACH
\<CSPRefId>	Identifier	Optional Repeating	Customer Service Provider Reference Identifier. This field is used as a selection criterion.
\<SPRefId>	Identifier	Optional Repeating	Service Provider Reference Identifier. This field is used as a selection criterion.
\<DeliveryMethod>	Open Enum	Optional Profiled values	Delivery Method. Default is Channel. Value must be supported in Service Profile.

6.4.10.2   Response \<DepAcctTrnInqRs>

The Deposit Account Transaction Inquiry Response message contains a list of Deposit Account Transaction Records for those transactions that meet the selection criteria in the request.

Tag	Type	Usage	Description
\<Status>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, \<StatusCode> defaults to 0 (zero).
\<RqUID>	UUID	Required Echoed	Request Identifier.
\<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header aggregate.
\<MsgRsHdr>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header aggregate.
\<AsyncRqUID>	UUID	Optional Echoed	Asynchronous Request Identifier.
\<CustId>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in \<SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then \<CustId> is required, and must contain the value of the user whose request is being issued.

Tag	Type	Usage	Description
\<RecCtrlOut\>	Aggregate *see section 3.2.12.2.2*	Optional *but see Description*	Records Control Output Aggregate. Required if \<RecCtrlIn\> was provided in the request and the server supports Records Control.
\<DepAcctId\>	Aggregate *see section 3.2.7.1.2*	Required XOR Echoed	Deposit Account Identification Aggregate.
\<CardAcctId\>	Aggregate *see section 3.2.7.1.3*	Required XOR Echoed	Card Account Identification Aggregate.
\<SelRangeDt\>	Aggregate *see section 3.2.10.1*	Optional Echoed	Selection Range Date Aggregate.
\<SelRangeCurAmt\>	Aggregate *see section 3.2.10.2*	Optional Echoed	Selection Range Amount Aggregate.
\<ChkRange\>	Aggregate *see section 6.3.16*	Optional Echoed	Selection Range Check Number Aggregate.
\<TrnType\>	Open Enum	Optional Repeating Echoed	Transaction Type.
\<TrnSrc\>	Open Enum	Optional Repeating Echoed	Transaction Source.
\<CSPRefId\>	Identifier	Optional Repeating Echoed	Customer Service Provider Reference Identifier.
\<SPRefId\>	Identifier	Optional Repeating Echoed	Service Provider Reference Identifier.
\<DeliveryMethod\>	Open Enum	Optional Echoed	Delivery Method.
\<MktgInfo\>	C-255	Optional	Marketing Information.
\<Fee\>	Aggregate *see section 3.2.16.2*	Optional Repeating	The fees charged for this transaction.
\<DepAcctTrnRec\>	Aggregate *see section 6.3.14*	Optional Repeating	Deposit Message Record Aggregate. One record per message subject to selection criteria and message records control.

6.4.11   *Deposit Account Transaction Reversal*

6.4.11.1   Request \<DepAcctTrnRevRq\>

Tag	Type	Usage	Description
\<RqUID\>	UUID	Required	Request Identifier.
\<MsgRqHdr\>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header aggregate.
\<AsyncRqUID\>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated

Tag	Type	Usage	Description
			by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously.
			For more information, see section **3.2.12.1**.
<CustId>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<RevReasonCode>	Open Enum	Required	Reversal reason code. The reason why the client is requesting a reversal.
<Desc>	C-80	Optional	Description, text describing the reason for the reversal.
<DepAcctTrnMsgRqInfo>	Aggregate	Required	Deposit Account Transaction Message Request Information Aggregate. This aggregate contains the message to reverse. See RevOptSupt in the service profile for specific reversal messages supported.
<DepAcctTrnInqRq>	Aggregate *see section 6.4.10.1*	Optional	Deposit Account Transaction Inquiry Request Message Aggregate.
</DepAcctTrnMsgRqInfo>			

6.4.11.2   Response <DepAcctTrnRevRs>

Tag	Type	Usage	Description
<Status>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, <StatusCode> defaults to 0 (zero).
<RqUID>	UUID	Required Echoed	Request Identifier.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header aggregate.
<MsgRsHdr>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header aggregate.
<AsyncRqUID>	UUID	Optional Echoed	Asynchronous Request Identifier.
<CustId>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Identification Aggregate.
<RevReasonCode>	Open Enum	Required Echoed	Reversal reason code. The reason why the client is requesting a reversal.
<Desc>	C-80	Optional Echoed	Description, text describing the reason for the reversal.
<DepAcctTrnMsgRqInfo>	Aggregate	Required Echoed	Deposit Account Transaction Message Request Information Aggregate. This aggregate contains the message to reverse. See RevOptSupt in the service profile for specific reversal messages supported.

Tag	Type	Usage	Description
<DepAcctTrnInqRq>	Aggregate *see section 6.4.10.1*	Optional	Deposit Account Transaction Inquiry Request Message Aggregate.
</DepAcctTrnMsgRqInfo>			
<CSPRefId>	Identifier	Optional	Customer Service Provider Reference Identifier.
<SPRefId>	Identifier	Optional	Service Provider Reference Identifier.

6.4.12   Credit Card Account Transaction Inquiry

A client uses the Credit Card Account Transaction Inquiry to retrieve transaction detail for a given account. A variety of selection criteria are supported.

6.4.12.1   Request <CCAcctTrnInqRq>

Tag	Type	Usage	Description
<RqUID>	UUID	Required	Request Identifier.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header aggregate.
<AsyncRqUID>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously. For more information, see section 3.2.12.1.
<CustId>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<RecCtrlIn>	Aggregate *see section 3.2.12.2.1*	Optional	Records Control Input Aggregate.
<CardAcctId>	Aggregate *see section 3.2.7.1.3*	Required	Card Account Identifier Aggregate.
<SelRangeDt>	Aggregate *see section 3.2.10.1*	Optional	Selection Range DateTime Aggregate.
<SelRangeCurAmt>	Aggregate *see section 3.2.10.2*	Optional	Selection Range Amount Aggregate.
<TrnType>	Open Enum	Optional Repeating	Transaction Type. See Data Dictionary for details. This field is used as a selection criterion. Defined values: Debit, Credit, Withdrawal, Check, Deposit, Transfer, Payment, Interest, Dividend, DirectDeposit, DirectDebit, RepeatPayment, Fee, ServiceCharge, Adjustment

Tag	Type	Usage	Description
<TrnSrc>	Open Enum	Optional Repeating	Message Source. Defined values: ATM, Teller, POS, VRU, HomeBank, ACH This field is used as a selection criterion.
<CSPRefId>	Identifier	Optional Repeating	Customer Service Provider Reference Identifier. This field is used as a selection criterion.
<SPRefId>	Identifier	Optional Repeating	Service Provider Reference Identifier. This field is used as a selection criterion.
<DeliveryMethod>	Open Enum	Optional Profiled values	Delivery Method. Default is Channel. Value must be supported in Service Profile.

6.4.12.2 Response <CCAcctTrnInqRs>

Tag	Type	Usage	Description
<Status>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, <StatusCode> defaults to 0 (zero).
<RqUID>	UUID	Required Echoed	Request Identifier.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header aggregate.
<MsgRsHdr>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header aggregate.
<AsyncRqUID>	UUID	Optional Echoed	Asynchronous Request Identifier.
<CustId>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<RecCtrlOut>	Aggregate *see section 3.2.12.2.2*	Optional *but see Description*	Message Records Control Output Aggregate. Required if <RecCtrlIn> was provided in the request, and the server supports Records Control.
<CardAcctId>	Aggregate *see section 3.2.7.1.3*	Required Echoed	Card Account Identifier Aggregate.
<SelRangeDt>	Aggregate *see section 3.2.10.1*	Optional Echoed	Selection Range DateTime Aggregate.
<SelRangeCurAmt>	Aggregate *see section 3.2.10.2*	Optional Echoed	Selection Range Amount Aggregate.
<TrnType>	Open Enum	Optional Repeating Echoed	Message Type.

Tag	Type	Usage	Description
<TrnSrc>	Open Enum	Optional Repeating Echoed	Message Source.
<CSPRefId>	Identifier	Optional Repeating Echoed	Customer Service Provider Reference Identifier.
<SPRefId>	Identifier	Optional Repeating Echoed	Service Provider Reference Identifier. This field is used as a selection criterion.
<DeliveryMethod>	Open Enum	Optional Echoed	Delivery Method.
<MktgInfo>	C-255	Optional	Marketing Information.
<CCAcctTrnRec>	Aggregate *see section 6.3.15*	Optional Repeating	Credit Card Transaction Record Aggregate. One record per message subject to selection criteria and message records control.

6.4.13   Credit Card Account Transaction Reversal

6.4.13.1   Request <CCAcctTrnRevRq>

Tag	Type	Usage	Description
<RqUID>	UUID	Required	Request Identifier.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header aggregate.
<AsyncRqUID>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously. For more information, see section **3.2.12.1**.
<CustId>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<RevReasonCode>	Open Enum	Required	Reversal reason code. The reason why the client is requesting a reversal.
<Desc>	C-80	Optional	Description, text describing the reason for the reversal.
<CCAcctTrnMsgRqInfo>	Aggregate	Required	Credit Card Account Transaction Message Request Information Aggregate. This aggregate contains the message to reverse. See RevOptSupt in the service profile for specific reversal messages supported.
<CCAcctTrnInqRq>	Aggregate *see section 6.4.12.1*	Optional	Credit Card Account Transaction Inquiry Request Message Aggregate.
</CCAcctTrnMsgRqInfo>			

6.4.13.2 Response <CCAcctTrnRevRs>

Tag	Type	Usage	Description
<Status>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, <StatusCode> defaults to 0 (zero).
<RqUID>	UUID	Required Echoed	Request Identifier.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header aggregate.
<MsgRsHdr>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header aggregate.
<AsyncRqUID>	UUID	Optional Echoed	Asynchronous Request Identifier.
<CustId>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Identification Aggregate.
<RevReasonCode>	Open Enum	Required Echoed	Reversal reason code. The reason why the client is requesting a reversal.
<Desc>	C-80	Optional Echoed	Description, text describing the reason for the reversal.
<CCAcctTrnMsgRqInfo>	Aggregate	Required Echoed	Credit Card Account Transaction Message Request Information Aggregate. This aggregate contains the message to reverse. See RevOptSupt in the service profile for specific reversal messages supported.
<CCAcctTrnInqRq>	Aggregate *see section 6.4.12.1*	Optional	Credit Card Account Transaction Inquiry Request Message Aggregate.
</CCAcctTrnMsgRqInfo>			
<CSPRefId>	Identifier	Optional	Customer Service Provider Reference Identifier.
<SPRefId>	Identifier	Optional	Service Provider Reference Identifier.

6.4.14   Bank Account Transaction Image Inquiry

The Bank Account Transaction Image Order Message allows a client to request a copy of a check or sales slip.

> *Note:*   *The client may specify a delivery method picked from those supported in the Service Profile.*

6.4.14.1 Request <BankAcctTrnImgInqRq>

Tag	Type	Usage	Description
<RqUID>	UUID	Required	Request Identifier. Sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header aggregate.

Tag	Type	Usage	Description
<AsyncRqUID>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously.
			For more information, see section 3.2.12.1.
<CustId>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<RecCtrlIn>	Aggregate *see section 3.2.12.2.1*	Optional	Records Control Input Aggregate.
<DeliveryMethod>	Open Enum	Optional Profiled values	Delivery Method. Used to request the delivery channel for requested information.
			Defined values: Channel, Courier, Email, Fax, HomeBank, InBand, Overnight, Post, TwoDay, UPS.
			Value selected must be supported in Service Profile.
<DeliveryMedia>	Open Enum	Optional	Delivery Media
			Defined values: CD, Diskette, DVD, Paper. Default value is Paper.
			Note: If media is specified, <DeliveryMethod> should be a physical method (i.e. Post, UPS)
<DepAcctId>	Aggregate *see section 3.2.7.1.2*	Required XOR	Deposit Account Identification Aggregate.
<CardAcctId>	Aggregate *see section 3.2.7.1.3*	Required XOR	Card Account Identification Aggregate.
<LoanAcctId>	Aggregate *see section 3.2.7.1.4*	Required XOR	Loan Account Identification Aggregate.
<StopChkInfo>	Aggregate *see section 6.3.17.1*	Optional Repeating XOR	Check Description Aggregate.
			This field is used as a selection criterion.
			Note: This aggregate will be deprecated from this message in IFX 2.0, to be replaced by <ChkNum>.
<ChkNum>	NC-12	Optional Repeating XOR 1.2+	Check Number. This field is used as a selection criterion.
<SelRangeCurAmt>	Aggregate *see section 3.2.10.2*	Optional	Selection Range Currency Amount. This field is used as a selection criterion.
<SelRangeDt>	Aggregate *see section 3.2.10.1*	Optional	Selection Range Date. This field is used as a selection criterion.
<TrnType>	Open Enum	Optional Repeating	Transaction Type. See Data Dictionary for details. This field is used as a selection criterion.

Tag	Type	Usage	Description
<IncAllItems>	Boolean	Optional	Indicate whether all items should be included in a deposit request. If set to *true*, all items will be included with a deposit copy request. If *false* or omitted, include only the deposit image.
<CustContact>	Aggregate *see section 3.2.3.3*	Optional XOR	Customer Contact Aggregate.
<ContactInfo>	Aggregate *see section 3.2.3.7*	Optional XOR 1.1+	Contact Information Aggregate.

6.4.14.2   Response <BankAcctTrnImgInqRs>

Tag	Type	Usage	Description
<Status>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, <StatusCode> defaults to 0 (zero).
<RqUID>	UUID	Required Echoed	Request Identifier.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header aggregate.
<MsgRsHdr>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header aggregate.
<AsyncRqUID>	UUID	Optional Echoed	Asynchronous Request Identifier.
<CustId>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<RecCtrlOut>	Aggregate *see section 3.2.12.2.2*	Optional *but see Description*	Records Control Output Aggregate. Required if <RecCtrlIn> was provided in the request and the server supports Records Control.
<DeliveryMethod>	Open Enum	Optional Profiled values Echoed	Delivery Method.
<DeliveryMedia>	Open Enum	Optional Echoed	Delivery Media
<DepAcctId>	Aggregate *see section 3.2.7.1.2*	Required XOR Echoed	Deposit Account Identification Aggregate.
<CardAcctId>	Aggregate *see section 3.2.7.1.3*	Required XOR Echoed	Card Account Identification Aggregate.
<LoanAcctId>	Aggregate *see section 3.2.7.1.4*	Required XOR Echoed	Loan Account Identification Aggregate.

Tag	Type	Usage	Description
<StopChkInfo>	Aggregate *see section 6.3.17.1*	Optional Repeating XOR Echoed	Check Description Aggregate.
<ChkNum>	NC-12	Optional Repeating XOR Echoed 1.2+	Check Number.
<SelRangeCurAmt>	Aggregate *see section 3.2.10.2*	Optional Echoed	Selection Range Currency Amount.
<SelRangeDt>	Aggregate *see section 3.2.10.1*	Optional Echoed	Selection Range Date.
<TrnType>	Open Enum	Optional Repeating Echoed	Transaction Type. See Data Dictionary for details.
<IncAllItems>	Boolean	Optional Echoed	Indicate whether all items should be included in a deposit request. If set to *true*, all items will be included with a deposit copy request. If *false* or omitted, include only the deposit image.
<CustContact>	Aggregate *see section 3.2.3.3*	Optional XOR Echoed	Customer Contact Aggregate.
<ContactInfo>	Aggregate *see section 3.2.3.7*	Optional XOR Echoed 1.1+	Contact Information Aggregate.
<BankAcctTrnImgRec>	Aggregate	Optional Repeating	Bank Account Transaction Image Record Aggregate. One aggregate is returned for each record matching the selection criteria, if the <DeliveryMethod> is InBand.
<TrnType>	Open Enum	Optional	Transaction Type.
<CurAmt>	Currency Amount	Optional	Currency Amount. The amount of the transaction.
<PrcDt>	Date	Optional	Transaction Processing Date.
<TrnImg>	Binary	Optional OR	Transaction Image.
<ImageURL>	URL	Optional OR	URL for retrieving an image of the requested transaction.
</BankAcctTrnImgRec>			
<CSPRefId>	Identifier	Optional	Customer Service Provider Reference Identifier.
<SPRefId>	Identifier	Optional	Service Provider Reference Identifier.

6.4.15   Bank Account Transaction Image Reversal

6.4.15.1   Request <BankAcctTrnImgRevRq>

Tag	Type	Usage	Description
<RqUID>	UUID	Required	Request Identifier.

Tag	Type	Usage	Description
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header aggregate.
<AsyncRqUID>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously.
			For more information, see section 3.2.12.1.
<CustId>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<RevReasonCode>	Open Enum	Required	Reversal reason code. The reason why the client is requesting a reversal.
<Desc>	C-80	Optional	Description, text describing the reason for the reversal.
<BankAcctTrnImgMsgRqInfo>	Aggregate	Required	Bank Account Transaction Image Reversal Message Request Information Aggregate. This aggregate contains the message to reverse. See RevOptSupt in the service profile for specific reversal messages supported.
<BankAcctTrnImgInqRq>	Aggregate *see section 6.4.14.1*	Optional	Bank Account Transaction Image Reversal Request Message Aggregate.
</BankAcctTrnImgMsgRqInfo>			

6.4.15.2   Response <BankAcctTrnImgRevRs>

Tag	Type	Usage	Description
<Status>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, <StatusCode> defaults to 0 (zero).
<RqUID>	UUID	Required Echoed	Request Identifier.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header aggregate.
<MsgRsHdr>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header aggregate.
<AsyncRqUID>	UUID	Optional Echoed	Asynchronous Request Identifier.
<CustId>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Identification Aggregate.
<RevReasonCode>	Open Enum	Required Echoed	Reversal reason code. The reason why the client is requesting a reversal.

Tag	Type	Usage	Description
\<Desc\>	C-80	Optional Echoed	Description, text describing the reason for the reversal.
\<BankAcctTrnImgMsgRqInfo\>	Aggregate	Required Echoed	Bank Account Transaction Image Reversal Aggregate. This aggregate contains the message to reverse. See RevOptSupt in the service profile for specific reversal messages supported.
\<BankAcctTrnImgInqRq\>	Aggregate *see section 6.4.14.1*	Optional	Bank Account Transaction Image Reversal Message Aggregate.
\</BankAcctTrnImgMsgRqInfo\>			
\<CSPRefId\>	Identifier	Optional	Customer Service Provider Reference Identifier.
\<SPRefId\>	Identifier	Optional	Service Provider Reference Identifier.

6.4.16   Interest Rate Inquiry

The Interest Rate Inquiry is used to retrieve current interest rates on various account types of particular balances.

> *Note:*   *This is not used to obtain interest rates on open accounts, but rather for possible new accounts.*

6.4.16.1   Request \<IntRateInqRq\>

Tag	Type	Usage	Description
\<RqUID\>	UUID	Required	Request Identifier.
\<MsgRqHdr\>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header aggregate.
\<AsyncRqUID\>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously.
			For more information, see section 3.2.12.1.
\<CustId\>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in \<SignonRq\>. For example, if a CSR or SP issues the request on behalf of the user, then \<CustId\> is required, and must contain the value of the user whose request is being issued.
\<RecCtrlIn\>	Aggregate *see section 3.2.12.2.1*	Optional	Records Control Input Aggregate.
\<AcctType\>	Open Enum	Required	Account Type.
			Defined values: DDA, SDA, CCA, ILA, CLA, CDA, LOC, MLA, MMA, CMA
\<AcctTaxStatus\>	Open Enum	Optional	Account Tax Status.
			Defined values: TaxDeferred, Standard
\<CurAmt\>	Currency Amount	Optional	Currency Amount.
\<Term\>	Aggregate *see section 3.2.9.1*	Optional	Term Aggregate.

Tag	Type	Usage	Description
<DeliveryMethod>	Open Enum	Optional Profiled values	Delivery Method. Default is Channel. Value must be supported in Service Profile.

6.4.16.2 Response <IntRateInqRs>

Tag	Type	Usage	Description
<Status>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, <StatusCode> defaults to 0 (zero).
<RqUID>	UUID	Required Echoed	Request Identifier.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header aggregate.
<MsgRsHdr>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header aggregate.
<AsyncRqUID>	UUID	Optional Echoed	Asynchronous Request Identifier.
<CustId>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<RecCtrlOut>	Aggregate *see section 3.2.12.2.2*	Optional *but see Description*	Records Control Output Aggregate. Required if <RecCtrlIn> was provided in the request and the server supports Records Control.
<AcctType>	Open Enum	Required Echoed	Account Type.
<AcctTaxStatus>	Open Enum	Optional Echoed	Account Tax Status.
<CurAmt>	Currency Amount	Optional Echoed	Currency Amount.
<Term>	Aggregate *see section 3.2.9.1*	Optional Echoed	Term Aggregate. Echo of request.
<DeliveryMethod>	Open Enum	Optional Echoed	Delivery Method.
<IntRateInfo>	Aggregate *see section 6.3.30.4*	Optional Repeating	Interest Rate Aggregate.

6.4.17   Interest Rate Inquiry Reversal

6.4.17.1 Request <IntRateRevRq>

Tag	Type	Usage	Description
<RqUID>	UUID	Required	Request Identifier.

Tag	Type	Usage	Description
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header aggregate.
<AsyncRqUID>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously.
			For more information, see section **3.2.12.1**.
<CustId>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<RevReasonCode>	Open Enum	Required	Reversal reason code. The reason why the client is requesting a reversal.
<Desc>	C-80	Optional	Description, text describing the reason for the reversal.
<IntRateMsgRqInfo>	Aggregate	Required	Interst Rate Message Request Information Aggregate. This aggregate contains the message to reverse. See RevOptSupt in the service profile for specific reversal messages supported.
<IntRateInqRq>	Aggregate *see section 6.4.16.1*	Optional	Interst Rate Inquiry Request Message Aggregate.
</IntRateMsgRqInfo>			

6.4.17.2   Response <IntRateRevRs>

Tag	Type	Usage	Description
<Status>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, <StatusCode> defaults to 0 (zero).
<RqUID>	UUID	Required Echoed	Request Identifier.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header aggregate.
<MsgRsHdr>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header aggregate.
<AsyncRqUID>	UUID	Optional Echoed	Asynchronous Request Identifier.
<CustId>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Identification Aggregate.
<RevReasonCode>	Open Enum	Required Echoed	Reversal reason code. The reason why the client is requesting a reversal.
<Desc>	C-80	Optional Echoed	Description, text describing the reason for the reversal.
<IntRateMsgRqInfo>	Aggregate	Required Echoed	Interst Rate Message Request Information Aggregate. This aggregate contains the message to reverse. See

Tag	Type	Usage	Description
			RevOptSupt in the service profile for specific reversal messages supported.
\<IntRateInqRq>	Aggregate *see section 6.4.16.1*	Optional	Interst Rate Inquiry Request Message Aggregate.
\</IntRateMsgRqInfo>			
\<CSPRefId>	Identifier	Optional	Customer Service Provider Reference Identifier.
\<SPRefId>	Identifier	Optional	Service Provider Reference Identifier.

6.4.18 *Bank Account Taxation Inquiry*

6.4.18.1 Request \<BankAcctTaxInqRq>

The client must specify a client identifier, the account identifier, and tax details.

Tag	Type	Usage	Description
\<RqUID>	UUID	Required	Request Identifier.
\<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header aggregate.
\<AsyncRqUID>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously. For more information, see section 3.2.12.1.
\<CustId>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in \<SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then \<CustId> is required, and must contain the value of the user whose request is being issued.
\<DepAcctId>	Aggregate *see section 3.2.7.1.2*	Required XOR	Deposit Account Identification Aggregate.
\<CardAcctId>	Aggregate *see section 3.2.7.1.3*	Required XOR	Card Account Identification Aggregate.
\<LoanAcctId>	Aggregate *see section 3.2.7.1.4*	Required XOR	Loan Account Identification Aggregate.
\<TaxYear>	Long	Required	Tax year.
\<DeliveryMethod>	Open Enum	Optional Profiled values	Delivery Method. Default is Channel. Value must be supported in Service Profile.

6.4.18.2 Response \<BankAcctTaxInqRs>

If research is required, requests for years other than current and prior years may require \<Status> to say that it is accepted for asynchronous processing.

Tag	Type	Usage	Description

Tag	Type	Usage	Description
\<Status\>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, \<StatusCode\> defaults to 0 (zero).
\<RqUID\>	UUID	Required Echoed	Request Identifier.
\<MsgRqHdr\>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header aggregate.
\<MsgRsHdr\>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header aggregate.
\<AsyncRqUID\>	UUID	Optional Echoed	Asynchronous Request Identifier.
\<CustId\>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in \<SignonRq\>. For example, if a CSR or SP issues the request on behalf of the user, then \<CustId\> is required, and must contain the value of the user whose request is being issued.
\<DepAcctId\>	Aggregate *see section 3.2.7.1.2*	Required XOR Echoed	Deposit Account Identification Aggregate.
\<CardAcctId\>	Aggregate *see section 3.2.7.1.3*	Required XOR Echoed	Card Account Identification Aggregate.
\<LoanAcctId\>	Aggregate *see section 3.2.7.1.4*	Required XOR Echoed	Loan Account Identification Aggregate.
\<TaxYear\>	Long	Required Echoed	Tax year.
\<DeliveryMethod\>	Open Enum	Optional Echoed	Delivery Method.
\<TaxId\>	NC-12	Required	Customer Tax Identifier.
\<AcctTaxInfo\>	Aggregate	Optional Repeating	Account Tax Information Aggregate.
\<Org\>	Identifier	Required	Organization. Organization defining this name space. Usage is Tax Authority (e.g., state or country).
\<TaxType\>	Open Enum	Optional	Tax Type. Qualified by \<Org\>.
			Defined values: WithHoldingTax, DebitsTax, FIDuty
\<CurAmt\>	Currency Amount	Optional	Tax Amount (+ paid by customer, − earned by customer).
\<PrevYrCurAmt\>	Currency Amount	Optional	Previous Year Tax Amount (+ paid by customer, − earned by customer).
\<Rate\>	Decimal	Optional	Tax Rate. Usage is a percentage (e.g., a value of 5.2 = 5.2%). (+ paid by customer, − earned by customer)
\</AcctTaxInfo\>			

6.4.19 *Foreign Exchange Rate Inquiry*

A client may request an exchange rate or a committed exchange rate for a future message. The Financial Institution may quote the current rate or may commit a rate for some time period. If a commitment is made to honor a rate until a future date, the Financial Institution must return a commitment identifier, which is provided in the <SPRefId> in the <ForExRateInfo>.

6.4.19.1 Request <ForExRateInqRq>

Tag	Type	Usage	Description
<RqUID>	UUID	Required	Request Identifier.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header aggregate.
<AsyncRqUID>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously.
			For more information, see section 3.2.12.1.
<CustId>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<DepAcctId>	Aggregate *see section 3.2.7.1.2*	Required XOR	Deposit Account Identification Aggregate.
<CardAcctId>	Aggregate *see section 3.2.7.1.3*	Required XOR	Card Account Identification Aggregate.
<CurAmt>	Currency Amount	Required	Message Amount. Always in the currency of the From account.
<CurCode>	NC-3	Required	To ISO Currency Code. From currency is <DepAcctIdFrom> or <CardAcctIdFrom> currency.
<ForExRateType>	Closed Enum	Optional	Requested Rate Type.
			Valid values: Indicated, Committed
<ForExRateDealType>	Closed Enum	Optional	Requested Deal Type.
			Valid values: Buy, Sell
			If not present, the default value is Buy.
<ForExValDtType>	Closed Enum	Optional	Value date type of the rate.
			Valid values: Spot, Future
<DeliveryMethod>	Open Enum	Optional Profiled values	Delivery Method. Default is Channel.
			Value must be supported in Service Profile.

6.4.19.2 Response <ForExRateInqRs>

Tag	Type	Usage	Description
<Status>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, <StatusCode> defaults to 0 (zero).

Tag	Type	Usage	Description
<RqUID>	UUID	Required Echoed	Request Identifier.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header aggregate.
<MsgRsHdr>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header aggregate.
<AsyncRqUID>	UUID	Optional Echoed	Asynchronous Request Identifier.
<CustId>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<DepAcctId>	Aggregate *see section 3.2.7.1.2*	Required XOR Echoed	Deposit Account Identification.
<CardAcctId>	Aggregate *see section 3.2.7.1.3*	Required XOR Echoed	Card Account Identification Aggregate.
<CurAmt>	Currency Amount	Required Echoed	Message Amount. Always in the currency of the FROM account.
<CurCode>	NC-3	Required Echoed	TO ISO Currency Code. FROM currency is <DepAcctIdFrom> or <CardAcctIdFrom> currency.
<ForExRateType>	Closed Enum	Optional Echoed	Requested Rate Type. Valid values: Indicated, Committed,
<ForExRateDealType>	Closed Enum	Optional Echoed	Requested Deal Type. Valid values: Buy, Sell
<ForExValDtType>	Closed Enum	Optional Echoed	Value date type of the rate. Valid values: Spot, Future
<DeliveryMethod>	Open Enum	Optional Echoed	Delivery Method.
<ForExRateRec>	Aggregate	Required	Foreign Exchange Rate Record Aggregate
<ForExRateId>	Identifier	Optional *but see Description*	Foreign Exchange Rate Identifier. This is required if the server is committing to an exchange rate.
<ForExRateInfo>	Aggregate *see section 6.3.18*	Required	Foreign Exchange Rate Information Aggregate. This aggregate provides a Financial Institution commitment to honor this rate for the period of time specified.
<ForExValDtType>	Closed Enum	Optional Echoed	Value date type of the rate. Valid values: Spot, Future
</ForExRateRec>			

6.4.20   Foreign Exchange Rate Reversal

6.4.20.1   Request <ForExRateRevRq>

Tag	Type	Usage	Description
<RqUID>	UUID	Required	Request Identifier.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header aggregate.
<AsyncRqUID>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously.
			For more information, see section **3.2.12.1**.
<CustId>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<RevReasonCode>	Open Enum	Required	Reversal reason code. The reason why the client is requesting a reversal.
<Desc>	C-80	Optional	Description, text describing the reason for the reversal.
<ForExRateMsgRqInfo>	Aggregate	Required	Foreign Exchange Rate Message Request Information Aggregate. This aggregate contains the message to reverse. See RevOptSupt in the service profile for specific reversal messages supported.
<ForExRateInqRq>	Aggregate *see section 6.4.19.1*	Optional	Foreign Exchange Rate Inquiry Request Message Aggregate.
</ForExRateMsgRqInfo>			

6.4.20.2   Response <ForExRateRevRs>

Tag	Type	Usage	Description
<Status>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, <StatusCode> defaults to 0 (zero).
<RqUID>	UUID	Required Echoed	Request Identifier.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header aggregate.
<MsgRsHdr>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header aggregate.
<AsyncRqUID>	UUID	Optional Echoed	Asynchronous Request Identifier.
<CustId>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Identification Aggregate.

Tag	Type	Usage	Description
\<RevReasonCode>	Open Enum	Required Echoed	Reversal reason code. The reason why the client is requesting a reversal.
\<Desc>	C-80	Optional Echoed	Description, text describing the reason for the reversal.
\<ForExRateMsgRqInfo>	Aggregate	Required Echoed	Foreign Exchange Rate Request Information Aggregate. This aggregate contains the message to reverse. See RevOptSupt in the service profile for specific reversal messages supported.
\<ForExRateInqRq>	Aggregate *see section 6.4.19.1*	Optional	Foreign Exchange Rate Inquiry Request Message Aggregate.
\</ForExRateMsgRqInfo>			
\<CSPRefId>	Identifier	Optional	Customer Service Provider Reference Identifier.
\<SPRefId>	Identifier	Optional	Service Provider Reference Identifier.

6.5 Foreign Exchange Deal Elements

6.5.1 Foreign Exchange Deal Record Aggregate \<ForExDealRec>

Tag	Type	Usage	Description
\<ForExDealId>	Identifier	Required	Foreign Exchange Deal Identifier (Committed deals).
\<ForExDealInfo>	Aggregate *see section 6.5.1.1*	Required	Foreign Exchange Deal Information Aggregate
\<ForExDealStatus>	Aggregate *see section 6.5.1.2*	Optional	Foreign Exchange Deal Status Aggregate

6.5.1.1 Foreign Exchange Deal Information \<ForExDealInfo>

Tag	Type	Usage	Description
\<ForExRateDealType>	Closed Enum	Optional	Requested Deal Type. Valid values: Buy, Sell. If not present, the default value is Buy.
\<ExpDt>	DateTime	Optional	When part of \<ForExDealAddRq>, it is an echo of the expiry date/time quoted by financial institution for the rate offer. When part of \<ForExDealAddrs>, it is the date/time before which all payments or funds transfers referencing this deal must be completed.
\<CurAmt>	Currency Amount	Required	Message Amount. Always in the currency of the From account.
\<CurCode>	NC-3	Required XOR	ISO Currency Code.
\<SPRefId>	Identifier	Required XOR	Service Provider Reference Identifier. In the usage of this message, it is the unique Id assigned to the Foreign exchange deal quote.

Tag	Type	Usage	Description
<CurAmt>	Currency Amount	Required	Message Amount. Always in the currency of the From account.
<CurCode>	NC-3	Required	ISO Currency Code.
<SPRefId>	Identifier	Required XOR	Service Provider Reference Identifier. In the usage of this message, it is the unique Id assigned to the Foreign exchange deal quote.

6.5.1.2 Foreign Exchange Deal Status Aggregate <ForExDealStatus>

Tag	Type	Usage	Description
<ForExDealStatusCode>	Closed Enum	Required	Foreign Exchange Deal Status Code. Valid values: Pending, Rejected, Completed
<StatusDesc>	C-255	Optional	Status Description. Explanatory text associated with this customer/disclosure link status.
<EffDt>	DateTime	Optional	Effective Date/Time. The date/time the <ForExDealStatusCode> took effect.
<StatusModBy>	Open Enum	Optional	Status Modified By. If present, indicates who modified the Status Code. Defined values: Customer, FI, CSP, CSPSR, BSP, BSPSR, CPP, CPPSR, BPP, and BPPSR.

6.5.2 Foreign Exchange Deal Add

A client may purchase foreign exchange currency using the <ForExDealAddRq> and later make a single or installment payment on this purchase using the associated <SPRefId> or <ForExDealId> as a reference identifier in a payment or funds transfer message.

6.5.2.1 Request <ForExDealAddRq>

Tag	Type	Usage	Description
<RqUID>	UUID	Required	Request Identifier. Sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header Aggregate.
<AsyncRqUID>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously.
<CustId>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<ForExDealInfo>	Aggregate *see section 6.5.1.1*	Required	Foreign Exchange Deal information aggregate.

6.5.2.2 Response <ForExDealAddRs>

Tag	Type	Usage	Description
<Status>	Aggregate *see section 3.2.12.1*	Optional	Status Aggregate
<RqUID>	UUID	Required Echoed	Request Identifier. Sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header Aggregate.
<MsgRsHdr>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header Aggregate.
<AsyncRqUID>	UUID	Optional Echoed	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously.
<CustId>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<ForExDealInfo>	Aggregate *see section 6.5.1.1*	Required Echoed	Foreign Exchange Deal information aggregate.
<ForExDealRec>	Aggregate *see section 6.5.1*	Required	Foreign Exchange Deal Record Aggregate.
<CSPRefId>	Identifier	Optional	Customer Service Provider Reference Identifier.
<SPRefId>	Identifier	Optional	Service Provider Reference Identifier.

6.5.3 Foreign Exchange Deal Modify

The Foreign Exchange Deal Modify messages provide support to modify a particular Foreign Exchange Deal.

6.5.3.1 Request <ForExDealModRq>

Tag	Type	Usage	Description
<RqUID>	UUID	Required	Request Identifier. Sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header Aggregate.
<AsyncRqUID>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously.

<CustId>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<ForExDealId>	Identifier	Required	Foreign Exchange Deal Identifier.
<ForExDealInfo>	Aggregate *see section 6.5.1.1*	Required	Foreign Exchange Deal Information aggregate.

6.5.3.2 Response <ForExDealModRs>

Tag	Type	Usage	Description
<Status>	Aggregate *see section 3.2.12.1*	Optional	Status Aggregate
<RqUID>	UUID	Required Echoed	Request Identifier. Sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header Aggregate.
<MsgRsHdr>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header Aggregate.
<AsyncRqUID>	UUID	Optional Echoed	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously.
<CustId>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<ForExDealId>	Identifier	Required Echoed	Foreign Exchange Deal Identifier.
<ForExDealInfo>	Aggregate *see section 6.5.1.1*	Required Echoed	Foreign Exchange Deal information aggregate.
<ForExDealRec>	Aggregate *see section 6.5.1*	Required	Foreign Exchange Deal Record Aggregate.
<CSPRefId>	Identifier	Optional	Customer Service Provider Reference Identifier.
<SPRefId>	Identifier	Optional	Service Provider Reference Identifier.

6.5.4   Foreign Exchange Deal Inquiry

The Foreign Exchange Deal Inquiry allows for a client to inquire on foreign exchange deal(s) based upon the supplied search criteria.

6.5.4.1 Request <ForExDealInqRq>

Tag	Type	Usage	Description
<RqUID>	UUID	Required	Request Identifier. Sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header Aggregate.
<AsyncRqUID>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously.
<CustId>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<ForExDealId>	Identifier	Optional Repeating	ForExDeal Identifier.
<SPRefId>	Identifier	Optional Repeating	Service Provider Reference Identifier
<ForExRateDealType>	Closed Enum	Optional Repeating	Requested Deal Type. Valid values: Buy, Sell If not present, the default value is Buy.

6.5.4.2 Response <ForExDealInqRs>

Tag	Type	Usage	Description
<Status>	Aggregate *see section 3.2.12.1*	Optional	Status Aggregate
<RqUID>	UUID	Required Echoed	Request Identifier. Sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header Aggregate.
<MsgRsHdr>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header Aggregate.
<AsyncRqUID>	UUID	Optional Echoed	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously.
<CustId>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.

<ForExDealId>	Identifier	Optional Repeating Echoed	ForExDeal Identifier.
<SPRefId>	Identifier	Optional Repeating Echoed	Service Provider Reference Identifier
<ForExRateDealType>	Closed Enum	Optional Repeating Echoed	Requested Deal Type. Valid values: Buy, Sell If not present, the default value is Buy.
<ForExDealRec>	Aggregate *see section 6.5.1*	Optional Repeating	ForExDeal Record Aggregate.

6.5.5   Foreign Exchange Deal Cancel

The Foreign Exchange Deal Cancel message will cancel an existing pending deal.

6.5.5.1   Request <ForExDealCanRq>

Tag	Type	Usage	Description
<RqUID>	UUID	Required	Request Identifier. Sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header Aggregate.
<AsyncRqUID>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously.
<CustId>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<ForExDealId>	Identifier	Required	Foreign Exchange Deal Identifier.

6.5.5.2   Response <ForExDealCanRs>

Tag	Type	Usage	Description
<Status>	Aggregate *see section 3.2.12.1*	Optional	Status Aggregate
<RqUID>	UUID	Required Echoed	Request Identifier. Sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header Aggregate.
<MsgRsHdr>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header Aggregate.

Tag	Type	Usage	Description
\<AsyncRqUID\>	UUID	Optional Echoed	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously.
\<CustId\>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in \<SignonRq\>. For example, if a CSR or SP issues the request on behalf of the user, then \<CustId\> is required, and must contain the value of the user whose request is being issued.
\<ForExDealId\>	Identifier	Required Echoed	Foreign Exchange Deal Identifier.
\<ForExDealRec\>	Aggregate *see section 6.5.1*	Optional	Foreign Exchange Deal Record Aggregate.
\<CSPRefId\>	Identifier	Optional	Customer Service Provider Reference Identifier.
\<SPRefId\>	Identifier	Optional	Service Provider Reference Identifier.

6.5.6   Foreign Exchange Deal Audit

The Foreign Exchange Deal Audit message supports the ability for the client to trace the message history for all changes impacting the specified foreign exchange deal.

6.5.6.1   Request \<ForExDealAudRq\>

Tag	Type	Usage	Description
\<RqUID\>	UUID	Required	Request Identifier. Sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.
\<MsgRqHdr\>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header Aggregate.
\<AsyncRqUID\>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously.
\<CustId\>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in \<SignonRq\>. For example, if a CSR or SP issues the request on behalf of the user, then \<CustId\> is required, and must contain the value of the user whose request is being issued.
\<RecCtrlIn\>	Aggregate *see section 3.2.12.2.1*	Optional	Records Control Input Aggregate.
\<SelRangeDt\>	Aggregate *see section 3.2.10.1*	Optional	Selection Range Date Aggregate.
\<Method\>	Closed Enum	Optional Repeating	Audit Selection Action. Used to identify actions associated with the object that is being audited (e.g. transfer). Valid values: Add, Mod, Can This field is used as a selection criterion.

Tag	Type	Usage	Description
<ForExDealId>	Identifier	Required Repeating	Foreign Exchange Deal Identifier.

6.5.6.2   Response <ForExDealAudRs>

Tag	Type	Usage	Description
<Status>	Aggregate *see section 3.2.12.1*	Optional	Status Aggregate
<RqUID>	UUID	Required Echoed	Request Identifier. Sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header Aggregate.
<MsgRsHdr>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header Aggregate.
<AsyncRqUID>	UUID	Optional Echoed	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously.
<CustId>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<RecCtrlOut>	Aggregate *see section 3.2.12.2.2*	Optional but see Description	Message Records Control Output Aggregate. Required if <RecCtrlIn> was provided in the request and the server supports Records Control.
<SelRangeDt>	Aggregate *see section 3.2.10.1*	Optional Echoed	Selection Range Date Aggregate.
<Method>	Closed Enum	Optional Repeating Echoed	Audit Selection Action. Used to identify actions associated with the object that is being audited (e.g. transfer). This field is used as a selection criterion.
<ForExDealId>	Identifier	Required Repeating Echoed	Foreign Exchange Deal Identifier.
<ForExDealMsgRec>	Aggregate	Optional Repeating	Message Record Aggregate.
<CustId>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the entity who initiated the request (e.g., the one whose identifier appears in the <SignonRq> of the original issued request). For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the CSR or SP.
<MsgRecDt>	Date/Time	Optional	Message Record Creation Date. The date/time at which the message record was stored/created by the service provider.

Tag	Type	Usage	Description
\<ForExDealAddRs>	Aggregate *see section 6.5.2.2*	Required XOR	Foreign Exchange Deal Add Response Message Aggregate.
\<ForExDealModRs>	Aggregate *see section 6.5.3.2*	Required XOR	Foreign Exchange Deal Modify Response Message Aggregate.
\<ForExDealCanRs>	Aggregate *see section 6.5.5.2*	Required XOR	Foreign Exchange Deal Cancel Response Message Aggregate.
\</ForExDealMsgRec>			

6.5.7 Foreign Exchange Deal Sync

The Foreign Exchange Deal Sync messages provide the ability for a client device to be brought up to date on changes made to the specified Foreign Exchange Deal object.

6.5.7.1 Request \<ForExDealSyncRq>

Tag	Type	Usage	Description
\<RqUID>	UUID	Required	Request Identifier. Sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.
\<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header Aggregate.
\<AsyncRqUID>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously.
\<CustId>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in \<SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then \<CustId> is required, and must contain the value of the user whose request is being issued.
\<RecCtrlIn>	Aggregate *see section 3.2.12.2.1*	Optional	Records Control Input Aggregate.
\<Token>	Identifier	Required Echoed	Token. Server assigned. Client should make no assumptions about the value of this token relative to others that it may have received. The token is only significant to the server that originally assigned it.\n\nThis is a token that has been previously sent by the server, or zero for first time requests.
\<ForExDealId>	Identifier	Required Repeating	Foreign Exchange Deal Identifier.

6.5.7.2 Response \<ForExDealSyncRs>

Tag	Type	Usage	Description
\<Status>	Aggregate *see section 3.2.12.1*	Optional	Status Aggregate

Tag	Type	Usage	Description
<RqUID>	UUID	Required Echoed	Request Identifier. Sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header Aggregate.
<MsgRsHdr>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header Aggregate.
<AsyncRqUID>	UUID	Optional Echoed	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously.
<CustId>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<RecCtrlOut>	Aggregate *see section 3.2.12.2.2*	Optional but see Description	Message Records Control Output Aggregate. Required if <RecCtrlIn> was provided in the request and the server supports Records Control.
<Token>	Identifier	Required Echoed	Token. Server assigned. Client should make no assumptions about the value of this token relative to others that it may have received. The token is only significant to the server that originally assigned it. This is a token that has been previously sent by the server, or zero for first time requests.
<NewToken>	Identifier	Required	New Token. Server assigned. Client should make no assumptions about the value of this token relative to others that it may have received. The token is only significant to the server that originally assigned it.
<ForExDealId>	Identifier	Required Repeating Echoed	Foreign Exchange Deal Identifier.
<ForExDealMsgRec>	Aggregate	Optional Repeating	Foreign Exchange Deal Message Record Aggregate.
<CustId>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the entity who initiated the request (e.g., the one whose identifier appears in the <SignonRq> of the original issued request). For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the CSR or SP.
<MsgRecDt>	Date/Time	Optional	Message Record Creation Date. The date/time at which the message record was stored/created by the service provider.
<ForExDealAddRs>	Aggregate *see section 6.5.2.2*	Required XOR	Foreign Exchange Deal Add Response Message Aggregate.
<ForExDealModRs>	Aggregate *see section 6.5.3.2*	Required XOR	Foreign Exchange Deal Modify Response Message Aggregate.

Tag	Type	Usage	Description
<ForExDealCanRs>	Aggregate *see section 6.5.5.2*	Required XOR	Foreign Exchange Deal Cancel Response Message Aggregate.
</ForExDealMsgRec>			
<CSPRefId>	Identifier	Optional	Customer Service Provider Reference Identifier.
<SPRefId>	Identifier	Optional	Service Provider Reference Identifier.

6.5.8   Foreign Exchange Deal Reversal

The Foreign Exchange Deal Reversal message supports only a reversal on a <ForExDealAddRq> message.

6.5.8.1   Request <ForExDealRevRq>

Tag	Type	Usage	Description
<RqUID>	UUID	Required	Request Identifier. Sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header Aggregate.
<AsyncRqUID>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously.
<CustId>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<RevReasonCode>	Open Enum	Required	Reversal reason code. The reason why the client is requesting a reversal.
			Defined values: CustomerCancellation, Unspecified-NoActionTaken, SuspectedMalfunction, FormatError-NoActionTaken, CompletedPartially, OriginalAmountIncorrect, ResponseReceivedTooLate, CardAcceptorDeviceUnableToCompleteTransaction, DepositOutOfBalance, NoCheckInEnvelope, PaymentOutOfBalance, DepositOutOfBalance-AppliedContents, PaymentOutOfBalance-AppliedContents, UnableToDeliverMessageToPointOfService, SuspectedMalfunction-CardRetained, SuspectedMalfunction-CardReturned, SuspectedMalfunction-Track3NotUpdated, SuspectedMalfunction-NoCashDispensed, TimedOutAtTakingMoney-NoCashDispensed, TimedOutAtTakingCard-CardRetainedAndNoCashDispensed, InvalidResponse-NoActionTaken, TimeoutWaitingForResponse
			(reference ISO 8583:1993 Annex A.11)
<Desc>	C-80	Optional	Description, text describing the reason for the reversal.
<ForExDealMsgRqInfo>	Aggregate	Required	Object Message Request Information Aggregate. This aggregate contains the message to reverse. See RevOptSupt in the service profile for specific reversal messages supported.

Tag	Type	Usage	Description
<ForExDealAddRq> **</ForExDealMsgRqInfo>**	Aggregate *see section 6.5.2.1*	Required	Foreign Exchange Deal Add Request Message Aggregate.

6.5.8.2 Response <ForExDealRevRs>

Tag	Type	Usage	Description
<Status>	Aggregate *see section 3.2.12.1*	Optional	Status Aggregate
<RqUID>	UUID	Required Echoed	Request Identifier. Sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header Aggregate.
<MsgRsHdr>	Aggregate *see section 3.2.1.2*	Optional	Message Request Header Aggregate.
<AsyncRqUID>	UUID	Optional Echoed	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously.
<CustId>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<RevReasonCode>	Open Enum	Required Echoed	Reversal reason code. The reason why the client is requesting a reversal. Defined values: CustomerCancellation, Unspecified-NoActionTaken, SuspectedMalfunction, FormatError-NoActionTaken, CompletedPartially, OriginalAmountIncorrect, ResponseReceivedTooLate, CardAcceptorDeviceUnableToCompleteTransaction, DepositOutOfBalance, NoCheckInEnvelope, PaymentOutOfBalance, DepositOutOfBalance-AppliedContents, PaymentOutOfBalance-AppliedContents, UnableToDeliverMessageToPointOfService, SuspectedMalfunction-CardRetained, SuspectedMalfunction-CardReturned, SuspectedMalfunction-Track3NotUpdated, SuspectedMalfunction-NoCashDispensed, TimedOutAtTakingMoney-NoCashDispensed, TimedOutAtTakingCard-CardRetainedAndNoCashDispensed, InvalidResponse-NoActionTaken, TimeoutWaitingForResponse (reference ISO 8583:1993 Annex A.11)
<Desc>	C-80	Optional Echoed	Description, text describing the reason for the reversal.
<ForExDealMsgRqInfo>	Aggregate	Required Echoed	Object Message Request Information Aggregate. This aggregate contains the message to reverse. See RevOptSupt in the service profile for specific reversal messages supported.

Tag	Type	Usage	Description
<ForExDealAddRq>	Aggregate *see section 6.5.2.1*	Required Echoed	Foreign Exchange Deal Add Request Message Aggregate.
</ForExDealMsgRqInfo>			
<CSPRefId>	Identifier	Optional	Customer Service Provider Reference Identifier.
<SPRefId>	Identifier	Optional	Service Provider Reference Identifier.

6.5.9   Foreign Exchange Deal Status Modify

The Foreign Exchange Deal Status Modify message allows for the modification of the status of a Foreign Exchange Deal.

6.5.9.1   Request <ForExDealStatusModRq>

Tag	Type	Usage	Description
<RqUID>	UUID	Required	Request Identifier. Sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header Aggregate.
<AsyncRqUID>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously.
<CustId>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<ForExDealId>	Identifier	Required	Foreign Exchange Deal Identifier.
<ForExDealStatus>	Aggregate *see section 6.5.1.2*	Required	Foreign Exchange Deal Status aggregate.

6.5.9.2 Response <ForExDealStatusModRs>

Tag	Type	Usage	Description
<Status>	Aggregate *see section 3.2.12.1*	Optional	Status Aggregate
<RqUID>	UUID	Required Echoed	Request Identifier. Sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header Aggregate.
<MsgRsHdr>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header Aggregate.

Tag	Type	Usage	Description
<AsyncRqUID>	UUID	Optional Echoed	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously.
<CustId>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<ForExDealId>	Identifier	Required Echoed	Foreign Exchange Deal Identifier.
<ForExDealStatus>	Aggregate *see section 6.5.1.2*	Required Echoed	Foreign Exchange Deal Status aggregate.
<CSPRefId>	Identifier	Optional	Customer Service Provider Reference Identifier.
<SPRefId>	Identifier	Optional	Service Provider Reference Identifier.

6.5.10   Foreign Exchange Deal Status Inquiry

The Foreign Exchange Deal Status Inquiry allows a user to inquire on Foreign Exchange Deal status.

6.5.10.1   Request <ForExDealStatusInqRq>

Tag	Type	Usage	Description
<RqUID>	UUID	Required	Request Identifier. Sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header Aggregate.
<AsyncRqUID>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously.
<CustId>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<ForExDealId>	Identifier	Required	Foreign Exchange Deal Identifier

6.5.10.2   Response <ForExDealStatusInqRs>

Tag	Type	Usage	Description
<Status>	Aggregate *see section 3.2.12.1*	Optional	Status Aggregate
<RqUID>	UUID	Required Echoed	Request Identifier. Sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.

\<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header Aggregate.
\<MsgRsHdr>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header Aggregate.
\<AsyncRqUID>	UUID	Optional Echoed	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously.
\<CustId>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in \<SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then \<CustId> is required, and must contain the value of the user whose request is being issued.
\<ForExDealId>	Identifier	Required Echoed	Foreign Exchange Deal Identifier
\<ForExDealStatus>	Aggregate *see section 6.5.1.2*	Required	Foreign Exchange Deal Status aggregate.
\<CSPRefId>	Identifier	Optional	Customer Service Provider Reference Identifier.
\<SPRefId>	Identifier	Optional	Service Provider Reference Identifier.

6.5.11 *Foreign Exchange Deal Advise*

The Foreign Exchange Deal Advise message advises a recipient of the deal or the deal status.

6.5.11.1 Request \<ForExDealAdviseRq>

Tag	Type	Usage	Description
\<RqUID>	UUID	Required	Request Identifier. Sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.
\<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header Aggregate.
\<MsgRsHdr>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header Aggregate.
\<AsyncRqUID>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously.
\<CustId>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in \<SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then \<CustId> is required, and must contain the value of the user whose request is being issued.
\<ForExDealRec>	Aggregate *see section 6.5.1*	Required XOR	Foreign Exchange Deal Record Aggregate.

Tag	Type	Usage	Description
<ForExDealId>	Identifier	Required XOR	Foreign Exchange Deal Identifier
<ForExDealStatus>	Aggregate *see section 6.5.1.2*	Required	Foreign Exchange Deal Status aggregate.

6.5.11.2   Response <ForExDealAdviseRs>

Tag	Type	Usage	Description
<Status>	Aggregate *see section 3.2.12.1*	Optional	Status Aggregate
<RqUID>	UUID	Required	Request Identifier. Sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header Aggregate.
<MsgRsHdr>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header Aggregate.
<AsyncRqUID>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously.
<CustId>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<ForExDealRec>	Aggregate *see section 6.5.1*	Required XOR	Foreign Exchange Deal Identifier
<ForExDealId>	Identifier	Required XOR	Foreign Exchange Deal Status aggregate.
<ForExDealStatus>	Aggregate *see section 6.5.1.2*	Required	Foreign Exchange Deal Record Aggregate.
<CSPRefId>	Identifier	Optional	Customer Service Provider Reference Identifier.
<SPRefId>	Identifier	Optional	Service Provider Reference Identifier.

6.6   Stop Check

6.6.1   Stop Check Add

The client sends a <StopChkAddRq> message to request that a check be stopped. The minimum function server must support identification of the check via check number <ChkNum> within <StopChkInfo>. The server stores the additional data elements within <StopChkInfo> (payee name and amount) to help identify the stop check request for future reference. They are not used to identify the check to be stopped.

6.6.1.1   Request <StopChkAddRq>

Tag	Type	Usage	Description
<RqUID>	UUID	Required	Request Identifier. Sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header aggregate.
<AsyncRqUID>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously. For more information, see section 3.2.12.1.
<CustId>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<DepAcctId>	Aggregate *see section 3.2.7.1.2*	Required XOR	Deposit Account Identification Aggregate.
<CardAcctId>	Aggregate *see section 3.2.7.1.3*	Required XOR	Card Account Identification Aggregate.
<LoanAcctId>	Aggregate *see section 3.2.7.1.4*	Required XOR	Loan Account Identification Aggregate.
<ChkRange>	Aggregate *see section 6.3.16*	Required XOR	Selection Range—Check Number Aggregate. <ChkRange> or <StopChkInfo>, but not both.
<StopChkInfo>	Aggregate *see section 6.3.17.1*	Required XOR	Check Description Aggregate. This field is used as a selection criterion.

6.6.1.2   Response <StopChkAddRs>

Tag	Type	Usage	Description
<Status>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, <StatusCode> defaults to 0 (zero).
<RqUID>	UUID	Required Echoed	Request Identifier.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header aggregate.
<MsgRsHdr>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header aggregate.
<AsyncRqUID>	UUID	Optional Echoed	Asynchronous Request Identifier.

Tag	Type	Usage	Description
<CustId>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<DepAcctId>	Aggregate *see section 3.2.7.1.2*	Required XOR Echoed	Deposit Account Identification Aggregate.
<CardAcctId>	Aggregate *see section 3.2.7.1.3*	Required XOR Echoed	Card Account Identification Aggregate.
<LoanAcctId>	Aggregate *see section 3.2.7.1.4*	Required XOR Echoed	Loan Account Identification Aggregate.
<ChkRange>	Aggregate *see section 6.3.16*	Required XOR Echoed	Selection Range—Check Number Aggregate.
<StopChkInfo>	Aggregate *see section 6.3.17.1*	Required XOR Echoed	Stop Check Information Aggregate.
<StopChkRec>	Aggregate *see section 6.3.17*	Required Repeating	Stop Check Record. This aggregate contains information about the server-stored stop check information and status.
<CSPRefId>	Identifier	Optional	Customer Service Provider Reference Identifier.
<SPRefId>	Identifier	Optional	Service Provider Reference Identifier.

6.6.2   Stop Check Cancel

A client may cancel a Stop Check Payment Request with a Financial Institution.

6.6.2.1   Request <StopChkCanRq>

Tag	Type	Usage	Description
<RqUID>	UUID	Required	Request Identifier.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header aggregate.
<AsyncRqUID>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously. For more information, see section 3.2.12.1.
<CustId>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<DepAcctId>	Aggregate *see section 3.2.7.1.2*	Required XOR	Deposit Account Identification Aggregate.

Tag	Type	Usage	Description
<CardAcctId>	Aggregate *see section 3.2.7.1.3*	Required XOR	Card Account Identification Aggregate.
<LoanAcctId>	Aggregate *see section 3.2.7.1.4*	Required XOR	Loan Account Identification Aggregate.
<ChkRange>	Aggregate *see section 6.3.16*	Required XOR	Selection Range Check Number Aggregate.
<StopChkInfo>	Aggregate *see section 6.3.17.1*	Required XOR	Check Description Aggregate. Usage is selection criteria.

6.6.2.2   Response <StopChkCanRs>

Tag	Type	Usage	Description
<Status>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, <StatusCode> defaults to 0 (zero).
<RqUID>	UUID	Required Echoed	Request Identifier.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header aggregate.
<MsgRsHdr>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header aggregate.
<AsyncRqUID>	UUID	Optional Echoed	Asynchronous Request Identifier.
<CustId>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<DepAcctId>	Aggregate *see section 3.2.7.1.2*	Required XOR Echoed	Deposit Account Identification Aggregate.
<CardAcctId>	Aggregate *see section 3.2.7.1.3*	Required XOR Echoed	Card Account Identification Aggregate.
<LoanAcctId>	Aggregate *see section 3.2.7.1.4*	Required XOR Echoed	Loan Account Identification Aggregate.
<ChkRange>	Aggregate *see section 6.3.16*	Required XOR Echoed	Selection Range Check Number Aggregate.
<StopChkInfo>	Aggregate *see section 6.3.17.1*	Required XOR Echoed	Stop Check Information Aggregate.

Tag	Type	Usage	Description
<StopChkRec>	Aggregate *see section 6.3.17*	Optional Repeating	Stop Check Record. This aggregate is provided to list all check stop payments that apply to the cancellation.
<CSPRefId>	Identifier	Optional	Customer Service Provider Reference Identifier.
			This is the <CSPRefId> for this Cancel Stop Check message.
<SPRefId>	Identifier	Optional	Service Provider Reference Identifier.
			This is the <SPRefId> for this Cancel Stop Check message.

6.6.3   Stop Check Inquiry

The Stop Check Inquiry message allows a client to retrieve records of both current and completed stopped checks. Since volume of stopped checks is expected to be relatively low on a per-customer basis, there is not a separate Stop Check History Inquiry.

6.6.3.1   Request <StopChkInqRq>

Tag	Type	Usage	Description
<RqUID>	UUID	Required	Request Identifier.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header aggregate.
<AsyncRqUID>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously.
			For more information, see section 3.2.12.1.
<CustId>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<RecCtrlIn>	Aggregate *see section 3.2.12.2.1*	Optional	Records Control Input Aggregate.
<DepAcctId>	Aggregate *see section 3.2.7.1.2*	Required XOR	Deposit Account Detail Aggregate.
<CardAcctId>	Aggregate *see section 3.2.7.1.3*	Required XOR	Card Account Identification Aggregate.
<LoanAcctId>	Aggregate *see section 3.2.7.1.4*	Required XOR	Loan Account Identification Aggregate.
<ChkRange>	Aggregate *see section 6.3.16*	Optional XOR	Selection Range—Check Number Aggregate.
<StopChkInfo>	Aggregate *see section 6.3.17.1*	Optional Repeating XOR	Check Description Aggregate. This field is used as a selection criterion.

Tag	Type	Usage	Description
<SelRangeDt>	Aggregate *see section 3.2.10.1*	Optional XOR	Selection Range Date Aggregate. Returns all stop check requests originated in this date range.
<IncToken>	Boolean	Optional	Include Token. If *True*, a <Token> should be included in the response to set a base for future Sync messages. If *False* or omitted, no <Token> is returned.
<DeliveryMethod>	Open Enum	Optional Profiled values	Delivery Method. Default is Channel. Value must be supported in Service Profile.

6.6.3.2   Response <StopChkInqRs>

Tag	Type	Usage	Description
<Status>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, <StatusCode> defaults to 0 (zero).
<RqUID>	UUID	Required Echoed	Request Identifier.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header aggregate.
<MsgRsHdr>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header aggregate.
<AsyncRqUID>	UUID	Optional Echoed	Asynchronous Request Identifier.
<CustId>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<RecCtrlOut>	Aggregate *see section 3.2.12.2.2*	Optional *but see Description*	Records Control Output Aggregate. Required if <RecCtrlIn> was provided in the request and the server supports Records Control.
<DepAcctId>	Aggregate *see section 3.2.7.1.2*	Required XOR Echoed	Deposit Account Detail Aggregate.
<CardAcctId>	Aggregate *see section 3.2.7.1.3*	Required XOR Echoed	Card Account Identification Aggregate.
<LoanAcctId>	Aggregate *see section 3.2.7.1.4*	Required XOR Echoed	Loan Account Identification Aggregate.
<ChkRange>	Aggregate *see section 6.3.16*	Optional XOR Echoed	Selection Range—Check Number Aggregate.
<StopChkInfo>	Aggregate *see section 6.3.17.1*	Optional XOR Repeating Echoed	Stopped Check Information Aggregate.

Tag	Type	Usage	Description
<SelRangeDt>	Aggregate *see section 3.2.10.1*	Optional XOR Echoed	Selection Range Date Aggregate.
<DeliveryMethod>	Open Enum	Optional Echoed	Delivery Method.
<StopChkRec>	Aggregate *see section 6.3.17*	Optional Repeating	Stopped Check Record Aggregate. One record is returned for each stopped check subject to selection criteria.
<Token>	Identifier	Optional	Token. Server Assigned. Client should make no assumptions about the value of this token relative to others that it may have received. The token is significant only to the server that originally assigned it. The token is only included if <IncToken>=*True* in the request. <Token>=0 is returned if no records are returned within the response.

6.6.4 Stop Check Audit

6.6.4.1 Request <StopChkAudRq>

Tag	Type	Usage	Description
<RqUID>	UUID	Required	Request Identifier.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header aggregate.
<AsyncRqUID>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously. For more information, see section 3.2.12.1.
<CustId>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<RecCtrlIn>	Aggregate *see section 3.2.12.2.1*	Optional	Records Control Input Aggregate.
<DepAcctId>	Aggregate *see section 3.2.7.1.2*	Required XOR	Deposit Account Identification Aggregate.
<CardAcctId>	Aggregate *see section 3.2.7.1.3*	Required XOR	Card Account Identification Aggregate.
<LoanAcctId>	Aggregate *see section 3.2.7.1.4*	Required XOR	Loan Account Identification Aggregate.
<SelRangeDt>	Aggregate *see section 3.2.10.1*	Optional	Selection Range Date Aggregate. Date of the Stop Check request.

Tag	Type	Usage	Description
<ChkRange>	Aggregate *see section 6.3.16*	Optional	Selection Range—Check Number Aggregate.

6.6.4.2 Response <StopChkAudRs>

Tag	Type	Usage	Description
<Status>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, <StatusCode> defaults to 0 (zero).
<RqUID>	UUID	Required Echoed	Request Identifier.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header aggregate.
<MsgRsHdr>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header aggregate.
<AsyncRqUID>	UUID	Optional Echoed	Asynchronous Request Identifier.
<CustId>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<RecCtrlOut>	Aggregate *see section 3.2.12.2.2*	Optional *but see Description*	Records Control Output Aggregate. Required if <RecCtrlIn> was provided in the request and the server supports Records Control.
<DepAcctId>	Aggregate *see section 3.2.7.1.2*	Required XOR Echoed	Deposit Account Identification Aggregate.
<CardAcctId>	Aggregate *see section 3.2.7.1.3*	Required XOR Echoed	Card Account Identification Aggregate.
<LoanAcctId>	Aggregate *see section 3.2.7.1.4*	Required XOR Echoed	Loan Account Identification Aggregate.
<SelRangeDt>	Aggregate *see section 3.2.10.1*	Optional Echoed	Stop Check Audit Selection Criteria Aggregate.
<ChkRange>	Aggregate *see section 6.3.16*	Optional Echoed	Selection Range Date Aggregate.
<StopChkMsgRec>	Aggregate	Optional Repeating	Stop Check Message Record Aggregate.
<CustId>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the entity who initiated the request (e.g., the one whose identifier appears in the <SignonRq> of the original issued request). For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the CSR or SP.

Tag	Type	Usage	Description
<MsgRecDt>	Date/Time	Optional	Audit Record Creation Date. Date/time at which the audit record was stored/created by the service provider.
<StopChkAddRs>	Aggregate *see section 6.6.1.2*	Required XOR	Stop Check Response Message Aggregate. One record is returned for each Stop Check message, which meets the selection criteria specified in the request.
<StopChkCanRs>	Aggregate *see section 6.6.2.2*	Required XOR	Cancel Stop Check Response Message Aggregate. One record is returned for each Cancel Stop Check message, which meets the selection criteria specified in the request.
<StopChkRevRs>	Aggregate *see section 6.6.6.2*	Required XOR	Cancel Stop Check Response Message Aggregate. One record is returned for each Reversal Stop Check message, which meets the selection criteria specified in the request.
</StopChkMsgRec>			

6.6.5   Stop Check Sync

6.6.5.1   Request <StopChkSyncRq>

Tag	Type	Usage	Description
<RqUID>	UUID	Required	Request Identifier.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header aggregate.
<AsyncRqUID>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously.\n\nFor more information, see section 3.2.12.1.
<CustId>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<RecCtrlIn>	Aggregate *see section 3.2.12.2.1*	Optional	Records Control Input Aggregate.
<DepAcctId>	Aggregate *see section 3.2.7.1.2*	Required XOR	Deposit Account Identification Aggregate.
<CardAcctId>	Aggregate *see section 3.2.7.1.3*	Required XOR	Card Account Identification Aggregate.
<LoanAcctId>	Aggregate *see section 3.2.7.1.4*	Required XOR	Loan Account Identification Aggregate.
<Token>	Identifier	Required	Token. Server assigned. Client should make no assumptions about the value of this token relative to others that it may have received. The token is significant only to the server that originally assigned it.\n\nThis is a token that has been previously sent by the server, or zero for first time requests.

6.6.5.2 Response <StopChkSyncRs>

Tag	Type	Usage	Description
<Status>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, <StatusCode> defaults to 0 (zero).
<RqUID>	UUID	Required Echoed	Request Identifier.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header aggregate.
<MsgRsHdr>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header aggregate.
<AsyncRqUID>	UUID	Optional Echoed	Asynchronous Request Identifier.
<CustId>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<RecCtrlOut>	Aggregate *see section 3.2.12.2.2*	Optional *but see Description*	Records Control Output Aggregate. Required if <RecCtrlIn> was provided in the request and the server supports Records Control.
<DepAcctId>	Aggregate *see section 3.2.7.1.2*	Required XOR Echoed	Deposit Account Identification Aggregate.
<CardAcctId>	Aggregate *see section 3.2.7.1.3*	Required XOR Echoed	Card Account Identification Aggregate.
<LoanAcctId>	Aggregate *see section 3.2.7.1.4*	Required XOR Echoed	Loan Account Identification Aggregate.
<Token>	Identifier	Required Echoed	Stop Check Synchronization Aggregate.
<NewToken>	Identifier	Required	New Token. Server assigned. Client should make no assumptions about the value of this token relative to others that it may have received. The token is significant only to the server that originally assigned it.
<StopChkMsgRec>	Aggregate	Optional Repeating	Stop Check Message Record Aggregate.
<CustId>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the entity who initiated the request (e.g., the one whose identifier appears in the <SignonRq> of the original issued request). For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the CSR or SP.
<MsgRecDt>	Date/Time	Optional	Audit Record Creation Date. Date/time at which the audit record was stored/created by the service provider.
<StopChkAddRs>	Aggregate *see section 6.6.1.2*	Required XOR	Stop Check Response Message Aggregate. One record is returned for each Stop Check message, which meets the selection criteria specified in the request.

Tag	Type	Usage	Description
<StopChkCanRs>	Aggregate *see section 6.6.2.2*	Required XOR	Cancel Stop Check Response Message Aggregate. One record is returned for each Cancel Stop Check message, which meets the selection criteria specified in the request.
</StopChkMsgRec>			

6.6.6   Stop Check Reversal

6.6.6.1   Request <StopChkRevRq>

Tag	Type	Usage	Description
<RqUID>	UUID	Required	Request Identifier.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header aggregate.
<AsyncRqUID>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously. For more information, see section **3.2.12.1**.
<CustId>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<RevReasonCode>	Open Enum	Required	Reversal reason code. The reason why the client is requesting a reversal.
<Desc>	C-80	Optional	Description, text describing the reason for the reversal.
<StopChkMsgRqInfo>	Aggregate	Required	Stop Check Message Request Information Aggregate. This aggregate contains the message to reverse. See RevOptSupt in the service profile for specific reversal messages supported.
<StopChkAddRq>	Aggregate *see section 6.6.1.1*	Required XOR	Stop Check Add Request Message Aggregate.
<StopChkCanRq>	Aggregate *see section 6.6.2.1*	Required XOR	Stop Check Cancel Request Message Aggregate.
<StopChkInqRq>	Aggregate *see section 6.6.3.1*	Required XOR	Stop Check Inquiry Request Message Aggregate.
<StopChkAudRq>	Aggregate *see section 6.6.4.1*	Required XOR	Stop Check Audit Request Message Aggregate.
<StopChkSyncRq>	Aggregate *see section 6.6.5.1*	Required XOR	Stop Check Synchronization Request Message Aggregate.
</StopChkMsgRqInfo>			

6.6.6.2 Response <StopChkRevRs>

Tag	Type	Usage	Description
<Status>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, <StatusCode> defaults to 0 (zero).
<RqUID>	UUID	Required Echoed	Request Identifier.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header aggregate.
<MsgRsHdr>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header aggregate.
<AsyncRqUID>	UUID	Optional Echoed	Asynchronous Request Identifier.
<CustId>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Identification Aggregate.
<RevReasonCode>	Open Enum	Required Echoed	Reversal reason code. The reason why the client is requesting a reversal.
<Desc>	C-80	Optional Echoed	Description, text describing the reason for the reversal.
<StopChkMsgRqInfo>	Aggregate	Required Echoed	Stop Check Message Request Information Aggregate. This aggregate contains the message to reverse. See RevOptSupt in the service profile for specific reversal messages supported.
<StopChkAddRq>	Aggregate *see section 6.6.1.1*	Required Echoed XOR	Stop Check Add Request Message Aggregate.
<StopChkCanRq>	Aggregate *see section 6.6.2.1*	Required Echoed XOR	Stop Check Cancel Request Message Aggregate.
<StopChkInqRq>	Aggregate *see section 6.6.3.1*	Required Echoed XOR	Stop Check Inquiry Request Message Aggregate.
<StopChkAudRq>	Aggregate *see section 6.6.4.1*	Required Echoed XOR	Stop Check Audit Request Message Aggregate.
<StopChkSyncRq>	Aggregate *see section 6.6.5.1*	Required Echoed XOR	Stop Check Synchronization Request Message Aggregate.
</StopChkMsgRqInfo>			
<CSPRefId>	Identifier	Optional	Customer Service Provider Reference Identifier.
<SPRefId>	Identifier	Optional	Service Provider Reference Identifier.

6.7 Funds Transfer

This section contains messages for transferring money between accounts within one Financial Institution, or among service providers.

6.7.1   Funds Transfer Add

6.7.1.1   Request <XferAddRq>

Tag	Type	Usage	Description
<RqUID>	UUID	Required	Request Identifier. Sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header aggregate.
<AsyncRqUID>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously.
			For more information, see section 3.2.12.1.
<CustId>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<XferInfo>	Aggregate *see section 6.3.19.2*	Required	Transfer Information Aggregate.
<DupChkOverride>	Boolean	Optional	Duplicate Check Override Flag. When set to *True*, requests that the server not perform duplicate checking if any is normally performed. The client is affirming that this is a new transfer being added.

6.7.1.2   Response <XferAddRs>

Tag	Type	Usage	Description
<Status>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, <StatusCode> defaults to 0 (zero).
<RqUID>	UUID	Required Echoed	Request Identifier.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header aggregate.
<MsgRsHdr>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header aggregate.
<AsyncRqUID>	UUID	Optional Echoed	Asynchronous Request Identifier.
<CustId>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.

Tag	Type	Usage	Description
<XferInfo>	Aggregate *see section 6.3.19.2*	Required Echoed	Transfer Information Aggregate.
<DupChkOverride>	Boolean	Optional Echoed	Duplicate Check Override Flag.
<XferRec>	Aggregate *see section 6.3.19*	Required	Transfer Record Aggregate.
<CSPRefId>	Identifier	Optional	Customer Service Provider Reference Identifier.
<SPRefId>	Identifier	Optional	Service Provider Reference Identifier.

6.7.2   Funds Transfer Modify

6.7.2.1   Request <XferModRq>

Tag	Type	Usage	Description
<RqUID>	UUID	Required	Request Identifier. Sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header aggregate.
<AsyncRqUID>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously. For more information, see section 3.2.12.1.
<CustId>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<XferId>	Identifier	Required	Transfer Identifier. Assigned by the server at the time the Transfer is first added. Cannot be changed by the client.
<XferInfo>	Aggregate *see section 6.3.19.2*	Required	Transfer Information Aggregate.
<DupChkOverride>	Boolean	Optional	Duplicate Check Override Flag. When set to *True*, requests that the server not perform duplicate checking if any is normally performed. The client is affirming that this is a new transfer being added.

6.7.2.2   Response <XferModRs>

Tag	Type	Usage	Description
<Status>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, <StatusCode> defaults to 0 (zero).
<RqUID>	UUID	Required Echoed	Request Identifier.

Tag	Type	Usage	Description
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header aggregate.
<MsgRsHdr>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header aggregate.
<AsyncRqUID>	UUID	Optional Echoed	Asynchronous Request Identifier.
<CustId>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<XferId>	Identifier	Required Echoed	Transfer Identifier.
<XferInfo>	Aggregate *see section 6.3.19.2*	Required Echoed	Transfer Information Aggregate.
<DupChkOverride>	Boolean	Optional Echoed	Duplicate Check Override Flag. When set to *True*, requests that the server not perform duplicate checking if any is normally performed. The client is affirming that this is a new transfer being added.
<XferRec>	Aggregate *see section 6.3.19*	Required	Transfer Record Aggregate.
<CSPRefId>	Identifier	Optional	Customer Service Provider Reference Identifier.
<SPRefId>	Identifier	Optional	Service Provider Reference Identifier.

6.7.3 Funds Transfer Status Modify

6.7.3.1 Request <XferStatusModRq>

Tag	Type	Usage	Description
<RqUID>	UUID	Required	Request Identifier.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header aggregate.
<AsyncRqUID>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously. For more information, see section 3.2.12.1.
<CustId>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<XferId>	UUID	Required	Transfer Identifier.

Tag	Type	Usage	Description
<XferStatus>	Aggregate *see section 6.3.19.3*	Required	Transfer Status Aggregate.

6.7.3.2   Response <XferStatusModRs>

Tag	Type	Usage	Description
<Status>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, <StatusCode> defaults to 0 (zero).
<RqUID>	UUID	Required Echoed	Request Identifier.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header aggregate.
<MsgRsHdr>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header aggregate.
<AsyncRqUID>	UUID	Optional Echoed	Asynchronous Request Identifier.
<CustId>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<XferId>	UUID	Required Echoed	Transfer Identifier.
<XferStatus>	Aggregate *see section 6.3.19.3*	Required Echoed	Transfer Status Aggregate.
<CSPRefId>	Identifier	Optional	Financial Institute Reference Identifier.
<SPRefId>	Identifier	Optional	Service Provider Reference Identifier.

6.7.4   Funds Transfer Cancel

6.7.4.1   Request <XferCanRq>

Tag	Type	Usage	Description
<RqUID>	UUID	Required	Request Identifier. Sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header aggregate.
<AsyncRqUID>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously. For more information, see section 3.2.12.1.

Tag	Type	Usage	Description
\<CustId\>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in \<SignonRq\>. For example, if a CSR or SP issues the request on behalf of the user, then \<CustId\> is required, and must contain the value of the user whose request is being issued.
\<XferId\>	Identifier	Required	Transfer Identifier. Assigned by the server at the time the Transfer is first added. Cannot be changed by the client.

6.7.4.2 Response \<XferCanRs\>

Tag	Type	Usage	Description
\<Status\>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, \<StatusCode\> defaults to 0 (zero).
\<RqUID\>	UUID	Required Echoed	Request Identifier.
\<MsgRqHdr\>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header aggregate.
\<MsgRsHdr\>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header aggregate.
\<AsyncRqUID\>	UUID	Optional Echoed	Asynchronous Request Identifier.
\<CustId\>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in \<SignonRq\>. For example, if a CSR or SP issues the request on behalf of the user, then \<CustId\> is required, and must contain the value of the user whose request is being issued.
\<XferId\>	Identifier	Required Echoed	Transfer Identifier.
\<XferRec\>	Aggregate *see section 6.3.19*	Optional	Transfer Record Aggregate. This aggregate is provided in cases where the server keeps the transfer's record on the server even after receiving a deletion request. This may occur if the deletion is not immediate, or if the server maintains the record with a status of Cancelled.
\<CSPRefId\>	Identifier	Optional	Customer Service Provider Reference Identifier.
\<SPRefId\>	Identifier	Optional	Service Provider Reference Identifier.

6.7.5 Funds Transfer Inquiry

6.7.5.1 Request \<XferInqRq\>

Tag	Type	Usage	Description
\<RqUID\>	UUID	Required	Request Identifier.
\<MsgRqHdr\>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header aggregate.

Tag	Type	Usage	Description
<AsyncRqUID>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously.
			For more information, see section 3.2.12.1.
<CustId>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<RecCtrlIn>	Aggregate *see section 3.2.12.2.1*	Optional	Records Control Input Aggregate.
<XferId>	Identifier	Optional Repeating	Transfer Identifier.
<RecXferId>	Identifier	Optional Repeating	Recurring Transfer Model Identifier.
			This field is used as a selection criterion to select only transfer instances generated by a Recurring Transfer Model. Inclusion of <RecXferId> must not cause this message to include the recurring transfer model itself. The model may be retrieved using the Recurring Transfer Inquiry.
<CSPRefId>	Identifier	Optional Repeating	Customer Service Provider Reference Identifier.
			This field is used as a selection criterion.
<SPRefId>	Identifier	Optional Repeating	Service Provider Reference Identifier.
			This field is used as a selection criterion.
<DepAcctIdFrom>	Aggregate *see section 3.2.7.1.2*	Optional XOR	Deposit Account Identification.
			This field is used as a selection criterion.
<CardAcctIdFrom>	Aggregate *see section 3.2.7.1.3*	Optional XOR	Card Account Identification.
			This field is used as a selection criterion.
<LoanAcctIdFrom>	Aggregate *see section 3.2.7.1.4*	Optional XOR	Loan Account Identification.
			This field is used as a selection criterion.
<DepAcctIdTo>	Aggregate	Optional XOR	Deposit Account Identification.
			This field is used as a selection criterion and is subject to server support in the Service profile.
<CardAcctIdTo>	Aggregate	Optional XOR	Card Account Identification.
			This field is used as a selection criterion and is subject to server support in the Service profile
<LoanAcctIdTo>	Aggregate	Optional XOR	Loan Account Identification.
			This field is used as a selection criterion and is subject to server support in the Service profile
<XferStatusCode>	Closed Enum	Optional Repeating	Transfer Status Code.
<SelRangeDueDt>	Aggregate *see section 3.2.10.1*	Optional	Selection Range Due Date Aggregate.

Tag	Type	Usage	Description
<SelRangeCurAmt>	Aggregate *see section 3.2.10.2*	Optional	Selection Range Amount Aggregate.
<IncToken>	Boolean	Optional	Include Token. If *True*, a <Token> should be included in the response to set a base for future Sync messages. If *False* or omitted, no <Token> is returned.
<DeliveryMethod>	Open Enum	Optional Profiled values	Delivery Method. Default is Channel.
			Value must be supported in Service Profile.

6.7.5.2   Response <XferInqRs>

Tag	Type	Usage	Description
<Status>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, <StatusCode> defaults to 0 (zero).
<RqUID>	UUID	Required Echoed	Request Identifier.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header aggregate.
<MsgRsHdr>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header aggregate.
<AsyncRqUID>	UUID	Optional Echoed	Asynchronous Request Identifier.
<CustId>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<RecCtrlOut>	Aggregate *see section 3.2.12.2.2*	Optional *but see Description*	Records Control Output Aggregate.
			Required if <RecCtrlIn> was provided in the request and the server supports Records Control.
<XferId>	Identifier	Optional Repeating Echoed	Transfer Identifier.
<RecXferId>	Identifier	Optional Repeating Echoed	Recurring Transfer Model Identifier.
			This field is used as a selection criterion to select only transfer instances generated by a Recurring Transfer Model. Inclusion of <RecXferId> must not cause this message to include the recurring transfer model itself. The model may be retrieved using the Recurring Transfer Inquiry.
<CSPRefId>	Identifier	Optional Repeating Echoed	Customer Service Provider Reference Identifier.
			This field is used as a selection criterion.
<SPRefId>	Identifier	Optional Repeating Echoed	Service Provider Reference Identifier.
			This field is used as a selection criterion.

Tag	Type	Usage	Description
<DepAcctIdFrom>	Aggregate *see section 3.2.7.1.2*	Optional XOR Echoed	Deposit Account Identification. This field is used as a selection criterion.
<CardAcctIdFrom>	Aggregate *see section 3.2.7.1.3*	Optional XOR Echoed	Card Account Identification. This field is used as a selection criterion.
<LoanAcctIdFrom>	Aggregate *see section 3.2.7.1.4*	Optional XOR Echoed	Loan Account Identification. This field is used as a selection criterion.
<DepAcctIdTo>	Aggregate	Optional XOR Echoed	Deposit Account Identification. This field is used as a selection criterion and is subject to server support in the Service profile.
<CardAcctIdTo>	Aggregate	Optional XOR Echoed	Card Account Identification. This field is used as a selection criterion and is subject to server support in the Service profile
<LoanAcctIdTo>	Aggregate	Optional XOR Echoed	Loan Account Identification. This field is used as a selection criterion and is subject to server support in the Service profile
<XferStatusCode>	Closed Enum	Optional Repeating Echoed	Processing Status.
<SelRangeDueDt>	Aggregate *see section 3.2.10.1*	Optional Echoed	Selection Range Due Date Aggregate.
<SelRangeCurAmt>	Aggregate *see section 3.2.10.2*	Optional Echoed	Selection Range Amount Aggregate.
<DeliveryMethod>	Open Enum	Optional Echoed	Delivery Method.
<XferRec>	Aggregate *see section 6.3.19*	Optional Repeating	Transfer Record Aggregate. These records are generated by the server and reflect the current state of the customer's Transfers. The records are filtered by the selection criteria specified in the request message. Also, note that transfers may have been generated by a client (using <XferAddRq>), or may have been generated by the server from one of the customer's Recurring Transfer Models.
<Token>	Identifier	Optional *but see Description*	Token. Server Assigned. Client should make no assumptions about the value of this token relative to others that it may have received. The token is significant only to the server that originally assigned it. The token is only included if <IncToken>=*True* in the request. <Token>=0 is returned if no records are returned within the response.

6.7.6   *Funds Transfer Audit*

6.7.6.1   Request <XferAudRq>

Tag	Type	Usage	Description
<RqUID>	UUID	Required	Request Identifier.

Tag	Type	Usage	Description
\<MsgRqHdr\>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header aggregate.
\<AsyncRqUID\>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously. For more information, see section 3.2.12.1.
\<CustId\>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in \<SignonRq\>. For example, if a CSR or SP issues the request on behalf of the user, then \<CustId\> is required, and must contain the value of the user whose request is being issued.
\<RecCtrlIn\>	Aggregate *see section 3.2.12.2.1*	Optional	Records Control Input Aggregate.
\<SelRangeDt\>	Aggregate *see section 3.2.10.1*	Optional	Selection Range Date Aggregate.
\<Method\>	Closed Enum	Optional Repeating	Audit Selection Action. Used to identify actions associated with the object that is being audited (e.g. transfer). Valid values: Add, Mod, Can This field is used as a selection criterion.
\<XferId\>	Identifier	Optional Repeating	Transfer Identifier. Assigned by the server at the time the Transfer is first added. Cannot be changed by the client. This field is used as a selection criterion.
\<RecXferId\>	Identifier	Optional Repeating	Recurring Transfer Model Identifier. Assigned by the server at the time the Recurring Transfer Model is first added. Cannot be changed by the client. This field is used as a selection criterion.

6.7.6.2   Response \<XferAudRs\>

Tag	Type	Usage	Description
\<Status\>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, \<StatusCode\> defaults to 0 (zero).
\<RqUID\>	UUID	Required Echoed	Request Identifier.
\<MsgRqHdr\>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header aggregate.
\<MsgRsHdr\>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header aggregate.
\<AsyncRqUID\>	UUID	Optional Echoed	Asynchronous Request Identifier.

Tag	Type	Usage	Description
\<CustId\>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in \<SignonRq\>. For example, if a CSR or SP issues the request on behalf of the user, then \<CustId\> is required, and must contain the value of the user whose request is being issued.
\<RecCtrlOut\>	Aggregate *see section 3.2.12.2.2*	Optional *but see Description*	Message Records Control Output Aggregate. Required if \<RecCtrlIn\> was provided in the request and the server supports Records Control.
\<SelRangeDt\>	Aggregate *see section 3.2.10.1*	Optional Echoed	Selection Range Date Aggregate.
\<Method\>	Closed Enum	Optional Repeating Echoed	Audit Selection Action. Used to identify actions associated with the object that is being audited (e.g. transfer). This field is used as a selection criterion.
\<XferId\>	Identifier	Optional Repeating Echoed	Transfer Identifier. Assigned by the server at the time the Transfer is first added. Cannot be changed by the client. This field is used as a selection criterion.
\<RecXferId\>	Identifier	Optional Repeating Echoed	Recurring Transfer Model Identifier. Assigned by the server at the time the Recurring Transfer Model is first added. Cannot be changed by the client. This field is used as a selection criterion.
\<XferMsgRec\>	Aggregate	Optional Repeating	Transfer Message Record Aggregate.
\<CustId\>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the entity who initiated the request (e.g., the one whose identifier appears in the \<SignonRq\> of the original issued request). For example, if a CSR or SP issues the request on behalf of the user, then \<CustId\> is required, and must contain the value of the CSR or SP.
\<MsgRecDt\>	Date/Time	Optional	Message Record Creation Date. The date/time at which the message record was stored/created by the service provider.
\<XferAddRs\>	Aggregate *see section 6.7.1.2*	Required XOR	Transfer Add Response Message Aggregate.
\<XferModRs\>	Aggregate *see section 6.7.2.2*	Required XOR	Transfer Modify Response Message Aggregate.
\<XferStatusModRs\>	Aggregate *see section 6.7.3.2*	Required XOR	Transfer Status Modify Response Message Aggregate.
\<XferCanRs\>	Aggregate *see section 6.7.4.2*	Required XOR	Transfer Cancel Response Message Aggregate.
\<XferRevRs\>	Aggregate *see section 6.7.8.2*	Required XOR	Transfer Reversal Response Message Aggregate.
\</XferMsgRec\>			

6.7.7   Funds Transfer Sync

6.7.7.1   Request <XferSyncRq>

Tag	Type	Usage	Description
<RqUID>	UUID	Required Echoed	Request Identifier.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header aggregate.
<AsyncRqUID>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously.
			For more information, see section 3.2.12.1.
<CustId>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<RecCtrlIn>	Aggregate *see section 3.2.12.2.1*	Optional	Records Control Input Aggregate.
<Token>	Identifier	Required	Token. Server assigned. Client should make no assumptions about the value of this token relative to others that it may have received. The token is significant only to the server that originally assigned it.
			This is a token that has been previously sent by the server, or zero for first time requests.
<DepAcctIdFrom>	Aggregate *see section 3.2.7.1.2*	Required XOR	Deposit Account Identification Aggregate.
<CardAcctIdFrom>	Aggregate *see section 3.2.7.1.3*	Required XOR	Card Account Identification Aggregate.
<LoanAcctIdFrom>	Aggregate *see section 3.2.7.1.4*	Required XOR	Loan Account Identification Aggregate.

6.7.7.2   Response <XferSyncRs>

Tag	Type	Usage	Description
<Status>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, <StatusCode> defaults to 0 (zero).
<RqUID>	UUID	Required Echoed	Request Identifier.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header aggregate.
<MsgRsHdr>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header aggregate.

Tag	Type	Usage	Description
\<AsyncRqUID\>	UUID	Optional Echoed	Asynchronous Request Identifier.
\<CustId\>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in \<SignonRq\>. For example, if a CSR or SP issues the request on behalf of the user, then \<CustId\> is required, and must contain the value of the user whose request is being issued.
\<RecCtrlOut\>	Aggregate *see section 3.2.12.2.2*	Optional *but see Description*	Message Records Control Output Aggregate. Required if \<RecCtrlIn\> was provided in the request and the server supports Records Control.
\<Token\>	Identifier	Required Echoed	Token. Server assigned. Client should make no assumptions about the value of this token relative to others that it may have received. The token is significant only to the server that originally assigned it. This is a token that has been previously sent by the server, or zero for first time requests.
\<DepAcctIdFrom\>	Aggregate *see section 3.2.7.1.2*	Required XOR Echoed	Deposit Account Identification Aggregate.
\<CardAcctIdFrom\>	Aggregate *see section 3.2.7.1.3*	Required XOR Echoed	Card Account Identification Aggregate.
\<LoanAcctIdFrom\>	Aggregate *see section 3.2.7.1.4*	Required XOR Echoed	Loan Account Identification Aggregate.
\<NewToken\>	Identifier	Required	New Token. Server assigned. Client should make no assumptions about the value of this token relative to others that it may have received. The token is significant only to the server that originally assigned it. If the client sent a token in the request, the server returns a new token based on this audit message.
\<XferMsgRec\>	Aggregate	Optional Repeating	Transfer Message Record Aggregate.
\<CustId\>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the entity who initiated the request (e.g., the one whose identifier appears in the \<SignonRq\> of the original issued request). For example, if a CSR or SP issues the request on behalf of the user, then \<CustId\> is required, and must contain the value of the CSR or SP.
\<MsgRecDt\>	Date/Time	Optional	Message Record Creation Date. The date/time at which the message record was stored/created by the service provider.
\<XferAddRs\>	Aggregate *see section 6.7.1.2*	Required XOR	Transfer Add Response Message Aggregate.
\<XferModRs\>	Aggregate *see section 6.7.2.2*	Required XOR	Transfer Modify Response Message Aggregate.
\<XferStatusModRs\>	Aggregate *see section 6.7.3.2*	Required XOR	Transfer Status Modify Response Message Aggregate.

Tag	Type	Usage	Description
<XferCanRs>	Aggregate *see section 6.7.4.2*	Required XOR	Transfer Cancel Response Message Aggregate.
</XferMsgRec>			

6.7.8   Funds Transfer Reversal

6.7.8.1   Request <XferRevRq>

Tag	Type	Usage	Description
<RqUID>	UUID	Required	Request Identifier.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header aggregate.
<AsyncRqUID>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously.
			For more information, see section **3.2.12.1**.
<CustId>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<RevReasonCode>	Open Enum	Required	Reversal reason code. The reason why the client is requesting a reversal.
<Desc>	C-80	Optional	Description, text describing the reason for the reversal.
<XferMsgRqInfo>	Aggregate	Required	Transfer Message Request Information Aggregate. This aggregate contains the message to reverse. See RevOptSupt in the service profile for specific reversal messages supported.
<XferAddRq>	Aggregate *see section 6.7.1.1*	Required XOR	Transfer Add Request Message Aggregate.
<XferModRq>	Aggregate *see section 6.7.2.1*	Required XOR	Transfer Modify Request Message Aggregate.
<XferStatusModRq>	Aggregate *see section 6.7.3.1*	Required XOR	Transfer Status Modify Request Message Aggregate.
<XferCanRq>	Aggregate *see section 6.7.4.1*	Required XOR	Transfer Cancel Request Message Aggregate.
<XferInqRq>	Aggregate *see section 6.7.5.1*	Required XOR	Transfer Inquiry Request Message Aggregate.
<XferAudRq>	Aggregate *see section 6.7.6.1*	Required XOR	Transfer Audit Request Message Aggregate.

Tag	Type	Usage	Description
<XferSyncRq>	Aggregate *see section 6.7.7.1*	Required XOR	Transfer Synchronization Request Message Aggregate.
</XferMsgRqInfo>			

6.7.8.2   Response <XferRevRs>

Tag	Type	Usage	Description
<Status>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, <StatusCode> defaults to 0 (zero).
<RqUID>	UUID	Required Echoed	Request Identifier.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header aggregate.
<MsgRsHdr>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header aggregate.
<AsyncRqUID>	UUID	Optional Echoed	Asynchronous Request Identifier.
<CustId>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Identification Aggregate.
<RevReasonCode>	Open Enum	Required Echoed	Reversal reason code. The reason why the client is requesting a reversal.
<Desc>	C-80	Optional Echoed	Description, text describing the reason for the reversal.
<XferMsgRqInfo>	Aggregate	Required Echoed	Transfer Message Request Information Aggregate. This aggregate contains the message to reverse. See RevOptSupt in the service profile for specific reversal messages supported.
<XferAddRq>	Aggregate *see section 6.7.1.1*	Required Echoed XOR	Transfer Add Request Message Aggregate.
<XferModRq>	Aggregate *see section 6.7.2.1*	Required Echoed XOR	Transfer Modify Request Message Aggregate.
<XferStatusModRq>	Aggregate *see section 6.7.3.1*	Required Echoed XOR	Transfer Status Modify Request Message Aggregate.
<XferCanRq>	Aggregate *see section 6.7.4.1*	Required Echoed XOR	Transfer Cancel Request Message Aggregate.
<XferInqRq>	Aggregate *see section 6.7.5.1*	Required Echoed XOR	Transfer Inquiry Request Message Aggregate.
<XferAudRq>	Aggregate *see section 6.7.6.1*	Required Echoed XOR	Transfer Audit Request Message Aggregate.

Tag	Type	Usage	Description
<XferSyncRq>	Aggregate *see section 6.7.7.1*	Required Echoed XOR	Transfer Synchronization Request Message Aggregate.
</XferMsgRqInfo>			
<CSPRefId>	Identifier	Optional	Customer Service Provider Reference Identifier.
<SPRefId>	Identifier	Optional	Service Provider Reference Identifier.

6.8 Recurring Transfer Model

6.8.1 Recurring Transfer Model Add

6.8.1.1 Request <RecXferAddRq>

Tag	Type	Usage	Description
<RqUID>	UUID	Required	Request Identifier. Sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header aggregate.
<AsyncRqUID>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously. For more information, see section 3.2.12.1.
<CustId>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<XferInfo>	Aggregate *see section 6.3.19.2*	Required	Transfer Information Aggregate.
<RecModelInfo>	Aggregate *see section 3.2.11.1*	Required	Recurring Model Information Aggregate.
<DupChkOverride>	Boolean	Optional	Duplicate Check Override Flag. When set to *True*, requests that the server not perform duplicate checking if any is normally performed. The client is affirming that this is a new transfer being added.

6.8.1.2 Response <RecXferAddRs>

Tag	Type	Usage	Description
<Status>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, <StatusCode> defaults to 0 (zero).
<RqUID>	UUID	Required Echoed	Request Identifier.

Tag	Type	Usage	Description
\<MsgRqHdr\>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header aggregate.
\<MsgRsHdr\>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header aggregate.
\<AsyncRqUID\>	UUID	Optional Echoed	Asynchronous Request Identifier.
\<CustId\>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in \<SignonRq\>. For example, if a CSR or SP issues the request on behalf of the user, then \<CustId\> is required, and must contain the value of the user whose request is being issued.
\<XferInfo\>	Aggregate *see section 6.3.19.2*	Required Echoed	Transfer Information Aggregate.
\<RecModelInfo\>	Aggregate *see section 3.2.11.1*	Required Echoed	Recurring Model Information Aggregate.
\<DupChkOverride\>	Boolean	Optional Echoed	Duplicate Check Override Flag.
\<RecXferRec\>	Aggregate *see section 6.3.19.1*	Required	Recurring Transfer Model Record Aggregate.
\<CSPRefId\>	Identifier	Optional	Customer Service Provider Reference Identifier.
\<SPRefId\>	Identifier	Optional	Service Provider Reference Identifier.

6.8.2 Recurring Transfer Model Modify

6.8.2.1 Request \<RecXferModRq\>

Tag	Type	Usage	Description
\<RqUID\>	UUID	Required	Request Identifier. Sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.
\<MsgRqHdr\>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header aggregate.
\<AsyncRqUID\>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously. For more information, see section 3.2.12.1.
\<CustId\>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in \<SignonRq\>. For example, if a CSR or SP issues the request on behalf of the user, then \<CustId\> is required, and must contain the value of the user whose request is being issued.

Tag	Type	Usage	Description
<RecXferId>	Identifier	Required	Recurring Transfer Model Identifier. Assigned by the server at the time the Recurring Transfer Model is first added. Cannot be changed by the client.
<XferInfo>	Aggregate *see section 6.3.19.2*	Required	Transfer Information.
<RecModelInfo>	Aggregate *see section 3.2.11.1*	Required	Recurring Model Information.
<ModPending>	Boolean	Optional Profiled support	Modify Pending Flag. If allowed by profile and set by client, any changes to the recurring model must be propagated to pending transfers previously spawned from the model. Regardless of this field, any instances spawned in the future must be based on the changed model.

6.8.2.2   Response <RecXferModRs>

Tag	Type	Usage	Description
<Status>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, <StatusCode> defaults to 0 (zero).
<RqUID>	UUID	Required Echoed	Request Identifier.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header aggregate.
<MsgRsHdr>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header aggregate.
<AsyncRqUID>	UUID	Optional Echoed	Asynchronous Request Identifier.
<CustId>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<RecXferId>	Identifier	Required Echoed	Recurring Transfer Model Identifier.
<XferInfo>	Aggregate *see section 6.3.19.2*	Required Echoed	Transfer Information.
<RecModelInfo>	Aggregate *see section 3.2.11.1*	Required Echoed	Recurring Model Information.
<ModPending>	Boolean	Optional Profiled support Echoed	Modify Pending Flag.
<RecXferRec>	Aggregate *see section 6.3.19.1*	Required	Recurring Transfer Model Record Aggregate.
<CSPRefId>	Identifier	Optional	Customer Service Provider Reference Identifier.

Tag	Type	Usage	Description
<SPRefId>	Identifier	Optional	Service Provider Reference Identifier.

6.8.3   Recurring Transfer Model Cancel

6.8.3.1   Request <RecXferCanRq>

Note: *Cancel Recurring Transfer Model always cancels pending transfers based on the model.*

Tag	Type	Usage	Description
<RqUID>	UUID	Required	Request Identifier. Sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header aggregate.
<AsyncRqUID>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously. For more information, see section 3.2.12.1.
<CustId>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<RecXferId>	Identifier	Required	Recurring Transfer Model Identifier. Assigned by the server at the time the Recurring Transfer Model is first added. Cannot be changed by the client.
<CascadeDel>	Boolean	Optional	Cascade Delete. If *True*, server must delete all dependent objects when this object is deleted. If *False* or omitted, the recurring model must not be deleted if dependent transfers exist.

6.8.3.2   Response <RecXferCanRs>

Tag	Type	Usage	Description
<Status>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, <StatusCode> defaults to 0 (zero).
<RqUID>	UUID	Required Echoed	Request Identifier.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header aggregate.
<MsgRsHdr>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header aggregate.
<AsyncRqUID>	UUID	Optional Echoed	Asynchronous Request Identifier.

Tag	Type	Usage	Description
<CustId>	Aggregate *see section* *3.2.3.1*	Optional Echoed	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<RecXferId>	Identifier	Required Echoed	Recurring Transfer Model Identifier.
<CascadeDel>	Boolean	Optional Echoed	Cascade Delete.
<RecXferRec>	Aggregate *see section* *6.3.19.1*	Optional XOR	Recurring Transfer Model Record Aggregate. This aggregate is provided in cases where the server keeps the recurring transfer model's record on the server even after receiving a deletion request. This may occur if the deletion is not immediate, or if the server maintains the record with a status of Cancelled.
<DependentType>	Open Enum	Optional XOR Repeating	Open Enum containing a list of depending object types that exist for the recurring model.
<CSPRefId>	Identifier	Optional	Customer Service Provider Reference Identifier.
<SPRefId>	Identifier	Optional	Service Provider Reference Identifier.

6.8.4   Recurring Transfer Model Inquiry

6.8.4.1   Request <RecXferInqRq>

Tag	Type	Usage	Description
<RqUID>	UUID	Required	Request Identifier. Sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.
<MsgRqHdr>	Aggregate *see section* *3.2.1.1*	Optional	Message Request Header aggregate.
<AsyncRqUID>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously. For more information, see section 3.2.12.1.
<CustId>	Aggregate *see section* *3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<RecCtrlIn>	Aggregate *see section* *3.2.12.2.1*	Optional	Records Control Input Aggregate.
<RecXferId>	Identifier	Optional Repeating	Recurring Transfer Model Identifier. Assigned by the server at the time the Recurring Transfer Model is first added. Cannot be changed by the client. This field is used as a selection criterion.

Tag	Type	Usage	Description
<DepAcctIdFrom>	Aggregate *see section 3.2.7.1.2*	Optional XOR	Deposit Account Identification Aggregate. This field is used as a selection criterion.
<CardAcctIdFrom>	Aggregate *see section 3.2.7.1.3*	Optional XOR	Card Account Identification Aggregate. This field is used as a selection criterion.
<LoanAcctIdFrom>	Aggregate *see section 3.2.7.1.4*	Optional XOR	Loan Account Identification Aggregate. This field is used as a selection criterion.
<SelRangeCurAmt>	Aggregate *see section 3.2.10.2*	Optional	Selection Range Amount Aggregate.
<CSPRefId>	Identifier	Optional Repeating	Customer Service Provider Reference Identifier. This field is used as a selection criterion.
<SPRefId>	Identifier	Optional Repeating	Service Provider Reference Identifier. This field is used as a selection criterion.
<IncToken>	Boolean	Optional	Include Token. If *True*, a <Token> should be included in the response to set a base for future Sync messages. If *False* or omitted, no <Token> is returned.
<DeliveryMethod>	Open Enum	Optional Profiled values	Delivery Method. Default is Channel. Value must be supported in Service Profile.

6.8.4.2   Response <RecXferInqRs>

Tag	Type	Usage	Description
<Status>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, <StatusCode> defaults to 0 (zero).
<RqUID>	UUID	Required Echoed	Request Identifier.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header aggregate.
<MsgRsHdr>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header aggregate.
<AsyncRqUID>	UUID	Optional Echoed	Asynchronous Request Identifier.
<CustId>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<RecCtrlOut>	Aggregate *see section 3.2.12.2.2*	Optional *but see Description*	Records Control Output Aggregate. Required if <RecCtrlIn> was provided in the request and the server supports Records Control.
<RecXferId>	Identifier	Optional Repeating Echoed	Recurring Transfer Model Identifier.

Tag	Type	Usage	Description
\<DepAcctIdFrom\>	Aggregate *see section 3.2.7.1.2*	Optional XOR Echoed	Deposit Account Detail Aggregate.
\<CardAcctIdFrom\>	Aggregate *see section 3.2.7.1.3*	Optional XOR Echoed	Card Account Detail Aggregate.
\<LoanAcctIdFrom\>	Aggregate *see section 3.2.7.1.4*	Optional XOR Echoed	Loan Account Detail Aggregate.
\<SelRangeCurAmt\>	Aggregate *see section 3.2.10.2*	Optional Echoed	Selection Range Amount Aggregate.
\<CSPRefId\>	Identifier	Optional Repeating Echoed	Customer Service Provider Reference Identifier.
\<SPRefId\>	Identifier	Optional Repeating Echoed	Service Provider Reference Identifier.
\<IncToken\>	Boolean	Optional Echoed	Include Token.
\<DeliveryMethod\>	Open Enum	Optional Echoed	Delivery Method.
\<RecXferRec\>	Aggregate *see section 6.3.19.1*	Optional Repeating	Recurring Transfer Model Record Aggregate. One record is returned for each of the customer's Recurring Transfer Models that meet the selection criteria specified in the request message.
\<Token\>	Identifier	Optional *but see Description*	Token. Server Assigned. Client should make no assumptions about the value of this token relative to others that it may have received. The token is significant only to the server that originally assigned it. The token is only included if \<IncToken\>=*True* in the request. \<Token\>=0 is returned if no records are returned within the response.

6.8.5   *Recurring Transfer Model Audit*

6.8.5.1   Request \<RecXferAudRq\>

Tag	Type	Usage	Description
\<RqUID\>	UUID	Required	Request Identifier. Sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.
\<MsgRqHdr\>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header aggregate.
\<AsyncRqUID\>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously. For more information, see section 3.2.12.1.

Tag	Type	Usage	Description
\<CustId\>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in \<SignonRq\>. For example, if a CSR or SP issues the request on behalf of the user, then \<CustId\> is required, and must contain the value of the user whose request is being issued.
\<RecCtrlIn\>	Aggregate *see section 3.2.12.2.1*	Optional	Message Records Control Input Aggregate.
\<SelRangeDt\>	Aggregate *see section 3.2.10.1*	Optional	Selection Range Date Aggregate.
\<Method\>	Closed Enum	Optional Repeating	Action. Used to identify actions associated with the object that is being audited (e.g. payment, transfer, etc.). Valid values: Add, Mod, Can This field is used as a selection criterion.
\<RecXferId\>	Identifier	Optional Repeating	Recurring Transfer Model Identifier. Assigned by the server at the time the Recurring Transfer Model is first added. Cannot be changed by the client. This field is used as a selection criterion.

6.8.5.2   Response \<RecXferAudRs\>

Tag	Type	Usage	Description
\<Status\>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, \<StatusCode\> defaults to 0 (zero).
\<RqUID\>	UUID	Required Echoed	Request Identifier.
\<MsgRqHdr\>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header aggregate.
\<MsgRsHdr\>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header aggregate.
\<AsyncRqUID\>	UUID	Optional Echoed	Asynchronous Request Identifier.
\<CustId\>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in \<SignonRq\>. For example, if a CSR or SP issues the request on behalf of the user, then \<CustId\> is required, and must contain the value of the user whose request is being issued.
\<RecCtrlOut\>	Aggregate *see section 3.2.12.2.2*	Optional *but see Description*	Records Control Output Aggregate. Required if \<RecCtrlIn\> was provided in the request and the server supports Records Control.
\<SelRangeDt\>	Aggregate *see section 3.2.10.1*	Optional Echoed	Selection Range Date Aggregate.

Tag	Type	Usage	Description
<Method>	Closed Enum	Optional Repeating Echoed	Action. Used to identify actions associated with the object that is being audited (e.g. payment, transfer, etc.).
			This field is used as a selection criterion.
<RecXferId>	Identifier	Optional Repeating Echoed	Recurring Transfer Model Identifier. Assigned by the server at the time the Recurring Transfer Model is first added. Cannot be changed by the client.
			This field is used as a selection criterion.
<RecXferMsgRec>	Aggregate	Optional Repeating	Recurring Transfer Message Record Aggregate.
<CustId>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the entity who initiated the request (e.g., the one whose identifier appears in the <SignonRq> of the original issued request). For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the CSR or SP.
<MsgRecDt>	Date/Time	Optional	Message Record Creation Date. The date/time at which the message record was stored/created by the service provider.
<RecXferAddRs>	Aggregate *see section 6.8.1.2*	Required XOR	Recurring Transfer Add Response Message Aggregate.
<RecXferModRs>	Aggregate *see section 6.8.2.2*	Required XOR	Recurring Transfer Modify Response Message Aggregate.
<RecXferCanRs>	Aggregate *see section 6.8.3.2*	Required XOR	Recurring Transfer Cancel Response Message Aggregate.
<RecXferRevRs>	Aggregate *see section 6.8.7.2*	Required XOR	Recurring Transfer Reversal Response Message Aggregate.
</RecXferMsgRec>			

6.8.6   Recurring Transfer Model Sync

6.8.6.1   Request <RecXferSyncRq>

Tag	Type	Usage	Description
<RqUID>	UUID	Required	Request Identifier. Sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header aggregate.
<AsyncRqUID>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously.
			For more information, see section 3.2.12.1.

Tag	Type	Usage	Description
\<CustId\>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in \<SignonRq\>. For example, if a CSR or SP issues the request on behalf of the user, then \<CustId\> is required, and must contain the value of the user whose request is being issued.
\<RecCtrlIn\>	Aggregate *see section 3.2.12.2.1*	Optional	Message Records Control Input Aggregate.
\<Token\>	Identifier	Required	Token. Server assigned. Client should make no assumptions about the value of this token relative to others that it may have received. The token is significant only to the server that originally assigned it. This is a token that has been previously sent by the server, or zero for first time requests.
\<DepAcctIdFrom\>	Aggregate *see section 3.2.7.1.2*	Required XOR	Deposit Account Identification Aggregate.
\<CardAcctIdFrom\>	Aggregate *see section 3.2.7.1.3*	Required XOR	Card Account Identification Aggregate.
\<LoanAcctIdFrom\>	Aggregate *see section 3.2.7.1.4*	Required XOR	Loan Account Identification Aggregate.

6.8.6.2   Response \<RecXferSyncRs\>

Tag	Type	Usage	Description
\<Status\>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, \<StatusCode\> defaults to 0 (zero).
\<RqUID\>	UUID	Required Echoed	Request Identifier.
\<MsgRqHdr\>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header aggregate.
\<MsgRsHdr\>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header aggregate.
\<AsyncRqUID\>	UUID	Optional Echoed	Asynchronous Request Identifier.
\<CustId\>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in \<SignonRq\>. For example, if a CSR or SP issues the request on behalf of the user, then \<CustId\> is required, and must contain the value of the user whose request is being issued.
\<RecCtrlOut\>	Aggregate *see section 3.2.12.2.2*	Optional *but see Description*	Records Control Output Aggregate. Required if \<RecCtrlIn\> was provided in the request and the server supports Records Control.

Tag	Type	Usage	Description
\<DepAcctIdFrom>	Aggregate *see section 3.2.7.1.2*	Required XOR	Deposit Account Identification Aggregate.
\<CardAcctIdFrom>	Aggregate *see section 3.2.7.1.3*	Required XOR	Card Account Identification Aggregate.
\<LoanAcctIdFrom>	Aggregate *see section 3.2.7.1.4*	Required XOR	Loan Account Identification Aggregate.
\<NewToken>	Identifier	Required	New Token. Server assigned. Client should make no assumptions about the value of this token relative to others that it may have received. The token is significant only to the server that originally assigned it.
\<RecXferMsgRec>	Aggregate	Optional Repeating	Recurring Transfer Message Record Aggregate.
\<CustId>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the entity who initiated the request (e.g., the one whose identifier appears in the \<SignonRq> of the original issued request). For example, if a CSR or SP issues the request on behalf of the user, then \<CustId> is required, and must contain the value of the CSR or SP.
\<MsgRecDt>	Date/Time	Optional	Message Record Creation Date. The date/time at which the message record was stored/created by the service provider.
\<RecXferAddRs>	Aggregate *see section 6.8.1.2*	Required XOR	Recurring Transfer Add Response Message Aggregate.
\<RecXferModRs>	Aggregate *see section 6.8.2.2*	Required XOR	Recurring Transfer Modify Response Message Aggregate.
\<RecXferCanRs>	Aggregate *see section 6.8.3.2*	Required XOR	Recurring Transfer Cancel Response Message Aggregate.
\</RecXferMsgRec>			

6.8.7   Recurring Transfer Reversal

6.8.7.1   Request \<RecXferRevRq>

Tag	Type	Usage	Description
\<RqUID>	UUID	Required	Request Identifier.
\<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header aggregate.
\<AsyncRqUID>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously. For more information, see section **3.2.12.1**.

Tag	Type	Usage	Description
<CustId>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<RevReasonCode>	Open Enum	Required	Reversal reason code. The reason why the client is requesting a reversal.
<Desc>	C-80	Optional	Description, text describing the reason for the reversal.
<RecXferMsgRqInfo>	Aggregate	Required	Recurring Transfer Message Request Information Aggregate. This aggregate contains the message to reverse. See RevOptSupt in the service profile for specific reversal messages supported.
<RecXferAddRq>	Aggregate *see section 6.8.1.1*	Required XOR	Recurring Transfer Add Request Message Aggregate.
<RecXferModRq>	Aggregate *see section 6.8.2.1*	Required XOR	Recurring Transfer Modify Request Message Aggregate.
<RecXferCanRq>	Aggregate *see section 6.8.3.1*	Required XOR	Recurring Transfer Cancel Request Message Aggregate.
<RecXferInqRq>	Aggregate *see section 6.8.4.1*	Required XOR	Recurring Transfer Inquiry Request Message Aggregate.
<RecXferAudRq>	Aggregate *see section 6.8.5.1*	Required XOR	Recurring Transfer Audit Request Message Aggregate.
<RecXferSyncRq>	Aggregate *see section 6.8.6.1*	Required XOR	Recurring Transfer Synchronization Request Message Aggregate.
</RecXferMsgRqInfo>			

6.8.7.2 Response <RecXferRevRs>

Tag	Type	Usage	Description
<Status>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, <StatusCode> defaults to 0 (zero).
<RqUID>	UUID	Required Echoed	Request Identifier.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header aggregate.
<MsgRsHdr>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header aggregate.
<AsyncRqUID>	UUID	Optional Echoed	Asynchronous Request Identifier.
<CustId>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Identification Aggregate.

Tag	Type	Usage	Description
<RevReasonCode>	Open Enum	Required Echoed	Reversal reason code. The reason why the client is requesting a reversal.
<Desc>	C-80	Optional Echoed	Description, text describing the reason for the reversal.
<RecXferMsgRqInfo>	Aggregate	Required Echoed	Recurring Transfer Message Request Information Aggregate. This aggregate contains the message to reverse. See RevOptSupt in the service profile for specific reversal messages supported.
<RecXferAddRq>	Aggregate *see section 6.8.1.1*	Required Echoed XOR	Recurring Transfer Add Request Message Aggregate.
<RecXferModRq>	Aggregate *see section 6.8.2.1*	Required Echoed XOR	Recurring Transfer Modify Request Message Aggregate.
<RecXferCanRq>	Aggregate *see section 6.8.3.1*	Required Echoed XOR	Recurring Transfer Cancel Request Message Aggregate.
<RecXferInqRq>	Aggregate *see section 6.8.4.1*	Required Echoed XOR	Recurring Transfer Inquiry Request Message Aggregate.
<RecXferAudRq>	Aggregate *see section 6.8.5.1*	Required Echoed XOR	Recurring Transfer Audit Request Message Aggregate.
<RecXferSyncRq>	Aggregate *see section 6.8.6.1*	Required Echoed XOR	Recurring Transfer Synchronization Request Message Aggregate.
</RecXferMsgRqInfo>			
<CSPRefId>	Identifier	Optional	Customer Service Provider Reference Identifier.
<SPRefId>	Identifier	Optional	Service Provider Reference Identifier.

6.9   Customer Communications

6.9.1   Check Order

6.9.1.1   Check Order Add

6.9.1.1.1   Request <ChkOrdAddRq>

The client sends a <ChkOrdAddRq> to place a new check order..

> *Note: The customer must determine the supported values for <Count> and <ChkBkStyleId> through an out-of-band process.*

Tag	Type	Usage	Description
<RqUID>	UUID	Required	Request Identifier. Sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header aggregate.

Tag	Type	Usage	Description
<AsyncRqUID>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously.
			For more information, see section 3.2.12.1.
<CustId>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<ChkOrdInfo>	Aggregate *see section 6.3.20.1*	Required	Check Order Information Aggregate.

6.9.1.1.2   Response <ChkOrdAddRs>

Tag	Type	Usage	Description
<Status>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, <StatusCode> defaults to 0 (zero).
<RqUID>	UUID	Required Echoed	Request Identifier.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header aggregate.
<MsgRsHdr>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header aggregate.
<AsyncRqUID>	UUID	Optional Echoed	Asynchronous Request Identifier.
<CustId>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<ChkOrdInfo>	Aggregate *see section 6.3.20.1*	Required Echoed	Check Order Information Aggregate.
<ChkOrdRec>	Aggregate *see section 6.3.20*	Required	Check Order Record Aggregate.
<CSPRefId>	Identifier	Optional	Customer Service Provider Reference Identifier.
<SPRefId>	Identifier	Optional	Service Provider Reference Identifier.

6.9.1.2   Check Order Modification

6.9.1.2.1   Request <ChkOrdModRq>

Check Order Modification Request message modifies a previously added check order record.

Tag	Type	Usage	Description
<RqUID>	UUID	Required	Request Identifier. Sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header aggregate.
<AsyncRqUID>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously. For more information, see section 3.2.12.1.
<CustId>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<ChkOrdId>	Identifier	Required	Check Order Model Identifier. Assigned by the server at the time the Check Order is first added. Cannot be changed by the client.
<ChkOrdInfo>	Aggregate *see section 6.3.20.1*	Required	Check Order Information Aggregate.

6.9.1.2.2   Response <ChkOrdModRs>

Tag	Type	Usage	Description
<Status>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, <StatusCode> defaults to 0 (zero).
<RqUID>	UUID	Required Echoed	Request Identifier.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header aggregate.
<MsgRsHdr>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header aggregate.
<AsyncRqUID>	UUID	Optional Echoed	Asynchronous Request Identifier.
<CustId>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<ChkOrdId>	Identifier	Required Echoed	Check Order Model Identifier. Assigned by the server at the time the Check Order is first added. Cannot be changed by the client.
<ChkOrdInfo>	Aggregate *see section 6.3.20.1*	Required Echoed	Check Order Information Aggregate.

Tag	Type	Usage	Description
\<ChkOrdRec\>	Aggregate *see section 6.3.20*	Required	Check Order Model Record Aggregate.
\<CSPRefId\>	Identifier	Optional	Customer Service Provider Reference Identifier.
\<SPRefId\>	Identifier	Optional	Service Provider Reference Identifier.

6.9.1.3   Check Order Cancellation

6.9.1.3.1   Request \<ChkOrdCanRq\>

Check Order Cancellation Request message cancels a previously added check order record.

Tag	Type	Usage	Description
\<RqUID\>	UUID	Required	Request Identifier. Sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.
\<MsgRqHdr\>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header aggregate.
\<AsyncRqUID\>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously.
			For more information, see section 3.2.12.1.
\<CustId\>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in \<SignonRq\>. For example, if a CSR or SP issues the request on behalf of the user, then \<CustId\> is required, and must contain the value of the user whose request is being issued.
\<ChkOrdId\>	Identifier	Required	Check Order Model Identifier. Assigned by the server at the time the Check Order is first added. Cannot be changed by the client.

6.9.1.3.2   Response \<ChkOrdCanRs\>

Tag	Type	Usage	Description
\<Status\>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, \<StatusCode\> defaults to 0 (zero).
\<RqUID\>	UUID	Required Echoed	Request Identifier.
\<MsgRqHdr\>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header aggregate.
\<MsgRsHdr\>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header aggregate.
\<AsyncRqUID\>	UUID	Optional Echoed	Asynchronous Request Identifier.

Tag	Type	Usage	Description
\<CustId\>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in \<SignonRq\>. For example, if a CSR or SP issues the request on behalf of the user, then \<CustId\> is required, and must contain the value of the user whose request is being issued.
\<ChkOrdId\>	Identifier	Required Echoed	Check Order Model Identifier. Assigned by the server at the time the Check Order is first added. Cannot be changed by the client.
\<ChkOrdRec\>	Aggregate *see section 6.3.20*	Optional	Check Order Model Record Aggregate.
\<CSPRefId\>	Identifier	Optional	Customer Service Provider Reference Identifier.
\<SPRefId\>	Identifier	Optional	Service Provider Reference Identifier.

6.9.1.4   Check Order Inquiry

6.9.1.4.1   Request \<ChkOrdInqRq\>

Check Order Inquiry Request message retrieves check order records based on selection criteria provided.

Tag	Type	Usage	Description
\<RqUID\>	UUID	Required	Request Identifier. Sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.
\<MsgRqHdr\>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header aggregate.
\<AsyncRqUID\>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously. For more information, see section 3.2.12.1.
\<CustId\>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in \<SignonRq\>. For example, if a CSR or SP issues the request on behalf of the user, then \<CustId\> is required, and must contain the value of the user whose request is being issued.
\<RecCtrlIn\>	Aggregate *see section 3.2.12.2.1*	Optional	Record Control In.
\<DepAcctId\>	Aggregate *see section 3.2.7.1.2*	Required XOR	Deposit Account Identifier Aggregate.
\<CardAcctId\>	Aggregate *see section 3.2.7.1.3*	Required XOR	Credt Card Account Identifier Aggregate.
\<LoanAcctId\>	Aggregate *see section 3.2.7.1.4*	Required XOR	Loan Account Identifier Aggregate.

Tag	Type	Usage	Description
\<ChkOrdId\>	Identifier	Optional Repeating	Check Order Identifier.
\<RecChkOrdId\>	Identifier	Optional Repeating	Recurring Check Order Identifier. This field is used as a selection criterion to select only check order instances generated by Recurring Check Order Model. Inclusion of \<RecChkOrdId\> must not cause this message to include the recurring check order model itself. The model may be retrieved using the Recurring Check Order Inquiry.
\<SelRangeDt\>	Aggregate *see section 3.2.10.1*	Optional	Selection Date Range.

6.9.1.4.2   Response \<ChkOrdInqRs\>

Tag	Type	Usage	Description
\<Status\>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, \<StatusCode\> defaults to 0 (zero).
\<RqUID\>	UUID	Required Echoed	Request Identifier.
\<MsgRqHdr\>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header aggregate.
\<MsgRsHdr\>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header aggregate.
\<AsyncRqUID\>	UUID	Optional Echoed	Asynchronous Request Identifier.
\<CustId\>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in \<SignonRq\>. For example, if a CSR or SP issues the request on behalf of the user, then \<CustId\> is required, and must contain the value of the user whose request is being issued.
\<RecCtrlOut\>	Aggregate *see section 3.2.12.2.2*	Optional *but see Description*	Record Control Output Aggregate. Required if \<RecCtrlIn\> was provided in the request and the server supports Record Control.
\<DepAcctId\>	Aggregate *see section 3.2.7.1.2*	Required XOR Echoed	Deposit Account Identifier Aggregate.
\<CardAcctId\>	Aggregate *see section 3.2.7.1.3*	Required XOR Echoed	Credit Card Account Identifier Aggregate.
\<LoanAcctId\>	Aggregate *see section 3.2.7.1.4*	Required XOR Echoed	Loan Account Identifier Aggregate.
\<ChkOrdId\>	Identifier	Optional Repeating Echoed	Check Order Identifier.

Tag	Type	Usage	Description
<RecChkOrdId>	Identifier	Optional Repeating Echoed	Recurring Check Order Identifier. This field is used as a selection criterion to select only check order instances generated by a Recurring Check Order Model. Inclusion of <RecChkOrdId> must not cause this message to include the recurring check order model itself. The model may be retrieved using the Recurring Check Order Inquiry.
<SelRangeDt>	Aggregate *see section 3.2.10.1*	Optional	Selection Date Range.
<ChkOrdRec>	Aggregate *see section 6.3.20*	Optional Repeating	Check Order Record Aggregate.
<CSPRefId>	Identifier	Optional	Customer Service Provider Reference Identifier.
<SPRefId>	Identifier	Optional	Service Provider Reference Identifier.

6.9.1.5   Check Order Audit

6.9.1.5.1   Request <ChkOrdAudRq>

Check Order Audit Request message audits a previously added check order record.

Tag	Type	Usage	Description
<RqUID>	UUID	Required	Request Identifier. Sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header aggregate.
<AsyncRqUID>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously.

For more information, see section 3.2.12.1. |
<CustId>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<RecCtrlIn>	Aggregate *see section 3.2.12.2.1*	Optional	Message Record Control Input Aggregate.
<SelRangeDt>	Aggregate *see section 3.2.10.1*	Optional	
<Method>	Closed Enum	Optional Repeating	Action. Used to identify actions associated with the object that is being audited (e.g. payment, ChkOrd, etc.).

Valid values: Add, Mod, Can

This field is used as selection criterion. |

Tag	Type	Usage	Description
<ChkOrdId>	Identifier	Optional Repeating	Check Order Model Identifier. Assigned by the server at the time the Check Order is first added. Cannot be changed by the client.
			This field is used as selection criterion.
<RecChkOrdId>	Identifier	Optional Repeating	Recurring Check Order Identifier. Assigned by the server at the time the Recurring Check Order Model is first added. Cannot be changed by the client.
			This field is used as selection criterion.

6.9.1.5.2   Response <ChkOrdAudRs>

Tag	Type	Usage	Description
<Status>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, <StatusCode> defaults to 0 (zero).
<RqUID>	UUID	Required Echoed	Request Identifier.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header aggregate.
<MsgRsHdr>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header aggregate.
<AsyncRqUID>	UUID	Optional Echoed	Asynchronous Request Identifier.
<CustId>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<RecCtrlOut>	Aggregate *see section 3.2.12.2.2*	Optional *but see Description*	Record Control Output Aggregate.
			Required if <RecCtrlIn> was provided in the request and the server supports Record Control.
<SelRangeDt>	Aggregate *see section 3.2.10.1*	Optional Echoed	Selection Range Date Aggregate.
<Method>	Closed Enum	Optional Repeating Echoed	Action. Used to identify actions associated with the object that is being audited (e.g. payment, ChkOrd, etc.).
			This field is used as selection criterion.
<ChkOrdId>	Identifier	Required Echoed	Check Order Model Identifier. Assigned by the server at the time the Check Order is first added. Cannot be changed by the client.
			This field is used as a selection criterion.
<RecChkOrdId>	Identifier	Optional Repeating	Recurring Check Order Identifier. Assigned by the server at the time the Recurring Check Order Model is first added. Cannot be changed by the client.
			This field is used as selection criterion.
<ChkOrdMsgRec>	Aggregate	Optional Repeating	Check Order Message Record Aggregate.

Tag	Type	Usage	Description
<CustId>	Aggregate *see section 3.2.3.1*	Optional Repeating	Customer Identification Aggregate.
<MsgRecDt>	DateTime	Optional	Message Record Creating Date. The date/time at which the message record was stored/created by the service provider.
<ChkOrdAddRs>	Aggregate *see section 6.9.1.1.2*	Required XOR	Check Order Add Response Message Aggregate.
<ChkOrdModRs>	Aggregate *see section 6.9.1.2.2*	Required XOR	Check Order Modify Response Message Aggregate.
<ChkOrdCanRs>	Aggregate *see section 6.9.1.3.2*	Required XOR	Check Order Cancel Response Message Aggregate.
<ChkOrdRevRs>	Aggregate *see section 6.9.1.7.2*	Required XOR	Check Order Reversal Response Message Aggregate.
</ChkOrdMsgRec>			

6.9.1.6   Check Order Synchronization

6.9.1.6.1   Request <ChkOrdSyncRq>

Check Order Synchronization Request message synchronizes the client with the server.

Tag	Type	Usage	Description
<RqUID>	UUID	Required	Request Identifier. Sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header aggregate.
<AsyncRqUID>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously. For more information, see section 3.2.12.1.
<CustId>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<RecCtrlIn>	Aggregate *see section 3.2.12.2.1*	Optional	Message Record Control Input Aggregate.
<Token>	Aggregate *see section 3.1.7*	Required	Token. Server assigned. Client should make no assumption about the value of this token relative to others that it may have received. The token is significant only to the server that originally assigned it. This is a token that has been previously sent by the server.

Tag	Type	Usage	Description
<ChkOrdId>	Identifier	Required	Check Order Model Identifier. Assigned by the server at the time the Check Order is first added. Cannot be changed by the client.

6.9.1.6.2   Response <ChkOrdSyncRs>

Tag	Type	Usage	Description
<Status>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, <StatusCode> defaults to 0 (zero).
<RqUID>	UUID	Required Echoed	Request Identifier.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header aggregate.
<MsgRsHdr>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header aggregate.
<AsyncRqUID>	UUID	Optional Echoed	Asynchronous Request Identifier.
<CustId>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<RecCtrlOut>	Aggregate *see section 3.2.12.2.2*	Optional *but see Description*	Record Control Output Aggregate. Required if <RecCtrlIn> was provided in the request and the server supports Record Control.
<Token>	Aggregate *see section 3.1.7*	Required Echoed	Token. Server assigned. Client should make no assumption about the value of this token relative to others that it may have received. The token is significant only to the server that originally assigned it. This is a token that has been previously sent by the server.
<ChkOrdId>	Identifier	Required Echoed	Check Order Model Identifier. Assigned by the server at the time the Check Order is first added. Cannot be changed by the client.
<NewToken>	Identifier	Required	New Token. Server assigned. Client should make no assumptions about the value of this token relative to others that it may have received. The token is significant only to the server that originally assigned it.
<ChkOrdMsgRec>	Aggregate	Optional Repeating	Check Order Message Record Aggregate.
<CustId>	Aggregate *see section 3.2.3.1*	Optional Repeating	Customer Identification Aggregate.
<MsgRecDt>	DateTime	Optional	Message Record Creating Date. The date/time at which the message record was stored/created by the service provider.
<ChkOrdAddRs>	Aggregate *see section 6.9.1.1.2*	Required XOR	Check Order Add Response Message Aggregate.

Tag	Type	Usage	Description
<ChkOrdModRs>	Aggregate *see section 6.9.1.2.2*	Required XOR	Check Order Modify Response Message Aggregate.
<ChkOrdCanRs>	Aggregate *see section 6.9.1.3.2*	Required XOR	Check Order Cancel Response Message Aggregate.
</ChkOrdMsgRec>			

6.9.1.7 Check Order Reversal

6.9.1.7.1 Request <ChkOrdRevRq>

Tag	Type	Usage	Description
<RqUID>	UUID	Required	Request Identifier.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header aggregate.
<AsyncRqUID>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously.
			For more information, see section **3.2.12.1**.
<CustId>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<RevReasonCode>	Open Enum	Required	Reversal reason code. The reason why the client is requesting a reversal.
<Desc>	C-80	Optional	Description, text describing the reason for the reversal.
<ChkOrdMsgRqInfo>	Aggregate	Required	Check Order Message Request Information Aggregate. This aggregate contains the message to reverse. See RevOptSupt in the service profile for specific reversal messages supported.
<ChkOrdAddRq>	Aggregate *see section 6.9.1.1.1*	Required XOR	Check Order Add Request Message Aggregate
<ChkOrdModRq>	Aggregate *see section 6.9.1.2.1*	Required XOR	Check Order Modify Request Message Aggregate
<ChkOrdCanRq>	Aggregate *see section 6.9.1.3.1*	Required XOR	Check Order Cancel Request Message Aggregate
<ChkOrdInqRq>	Aggregate *see section 6.9.1.4.1*	Required XOR	Check Order Inquiry Request Message Aggregate.
<ChkOrdAudRq>	Aggregate *see section 6.9.1.5.1*	Required XOR	Check Order Audit Request Message Aggregate
<ChkOrdSynchRq>	Aggregate	Required XOR	Check Order Synchronization Request Message Aggregate

Tag	Type	Usage	Description
</ChkOrdMsgRqInfo>			

6.9.1.7.2   Response <ChkOrdRevRs>

Tag	Type	Usage	Description
<Status>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, <StatusCode> defaults to 0 (zero).
<RqUID>	UUID	Required Echoed	Request Identifier.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header aggregate.
<MsgRsHdr>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header aggregate.
<AsyncRqUID>	UUID	Optional Echoed	Asynchronous Request Identifier.
<CustId>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Identification Aggregate.
<RevReasonCode>	Open Enum	Required Echoed	Reversal reason code. The reason why the client is requesting a reversal.
<Desc>	C-80	Optional Echoed	Description, text describing the reason for the reversal.
<ChkOrdMsgRqInfo>	Aggregate	Required Echoed	Check Order Message Request Information Aggregate. This aggregate contains the message to reverse. See RevOptSupt in the service profile for specific reversal messages supported.
<ChkOrdAddRq>	Aggregate *see section 6.9.1.1.1*	Required Echoed XOR	Check Order Add Request Message Aggregate
<ChkOrdModRq>	Aggregate *see section 6.9.1.2.1*	Required Echoed XOR	Check Order Modify Request Message Aggregate
<ChkOrdCanRq>	Aggregate *see section 6.9.1.3.1*	Required Echoed XOR	Check Order Cancel Request Message Aggregate
<ChkOrdInqRq>	Aggregate *see section 6.9.1.4.1*	Required Echoed XOR	Check Order Inquiry Request Message Aggregate.
<ChkOrdAudRq>	Aggregate *see section 6.9.1.5.1*	Required Echoed XOR	Check Order Audit Request Message Aggregate
<ChkOrdSynchRq>	Aggregate	Required Echoed XOR	Check Order Synchronization Request Message Aggregate
</ChkOrdMsgRqInfo>			
<CSPRefId>	Identifier	Optional	Customer Service Provider Reference Identifier.
<SPRefId>	Identifier	Optional	Service Provider Reference Identifier.

6.9.2 *Recurring Check Order*

6.9.2.1 **Recurring Check Order Add**

6.9.2.1.1 *Request <RecChkOrdAddRq>*

The Recurring Check Order Model Add message allows a client to set-up a recurring or repeating check order.

Tag	Type	Usage	Description
<RqUID>	UUID	Required	Request Identifier. Sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header aggregate.
<AsyncRqUID>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously. For more information, see section 3.2.12.1.
<CustId>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<ChkOrdInfo>	Aggregate *see section 6.3.20.1*	Required	Check Order Information Aggregate.
<RecModelInfo>	Aggregate *see section 3.2.11.1*	Required	Recurring Model Information Aggregate.
<DupChkOverride>	Boolean	Optional	Duplicat Check Override Flag. When set to *True*, rrequest that the server not perform duplicat checking if any is normally performed. The client is affirming that this is a new Check Order being added.

6.9.2.1.2 *Response <RecChkOrdAddRs>*

Tag	Type	Usage	Description
<Status>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, <StatusCode> defaults to 0 (zero).
<RqUID>	UUID	Required Echoed	Request Identifier.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header aggregate.
<MsgRsHdr>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header aggregate.
<AsyncRqUID>	UUID	Optional Echoed	Asynchronous Request Identifier.

Tag	Type	Usage	Description
\<CustId\>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in \<SignonRq\>. For example, if a CSR or SP issues the request on behalf of the user, then \<CustId\> is required, and must contain the value of the user whose request is being issued.
\<ChkOrdInfo\>	Aggregate *see section 6.3.20.1*	Required Echoed	Check Order Information Aggregate.
\<RecModelInfo\>	Aggregate *see section 3.2.11.1*	Required Echoed	Recurring Model Information Aggregate.
\<DupChkOverride\>	Boolean	Optional Echoed	Duplicat Check Override Flag. When set to *True*, reqquest that the server not perform duplicat checking if any is normally performed. The client is affirming that this is a new Check Order being added.
\<RecChkOrdRec\>	Aggregate *see section 6.3.21*	Required	Recurring Check Order Record Aggregate.
\<CSPRefId\>	Identifier	Optional	Customer Service Provider Reference Identifier.
\<SPRefId\>	Identifier	Optional	Service Provider Reference Identifier.

6.9.2.2   Recurring Check Order Modification

6.9.2.2.1   *Request \<RecChkOrdModRq\>*

The Recurring Check Order Model Modification message allows a client to modify a previously added recurring or repeating check order.

Tag	Type	Usage	Description
\<RqUID\>	UUID	Required	Request Identifier. Sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.
\<MsgRqHdr\>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header aggregate.
\<AsyncRqUID\>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously. For more information, see section 3.2.12.1.
\<CustId\>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in \<SignonRq\>. For example, if a CSR or SP issues the request on behalf of the user, then \<CustId\> is required, and must contain the value of the user whose request is being issued.
\<RecChkOrdId\>	Identifier	Required	Recurring Check Order Model Identifier. Assigned by the server at the tgime the Recurring Check Order Model is first added. Cannot be changed by the client.
\<ChkOrdInfo\>	Aggregate *see section 6.3.20.1*	Required	Check Order Information Aggregate.

Tag	Type	Usage	Description
<RecModelInfo>	Aggregate *see section 3.2.11.1*	Required	Recurring Model Information Aggregate.
<ModPending>	Boolean	Optional Profiled support	Modify Pending Flag. If allowed by profile and set by the client, any changes to the recurring model must be propagated to pending check orders previoiusly spawned from the model. Regardless of this field, any instances spawned in the future must be based on the changed model.

6.9.2.2.2   Response <RecChkOrdModRs>

Tag	Type	Usage	Description
<Status>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, <StatusCode> defaults to 0 (zero).
<RqUID>	UUID	Required Echoed	Request Identifier.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header aggregate.
<MsgRsHdr>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header aggregate.
<AsyncRqUID>	UUID	Optional Echoed	Asynchronous Request Identifier.
<CustId>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<RecChkOrdId>	Identifier	Required Echoed	Recurring Check Order Model Identifier.
<ChkOrdInfo>	Aggregate *see section 6.3.20.1*	Required Echoed	Check Order Information Aggregate.
<RecModelInfo>	Aggregate *see section 3.2.11.1*	Required Echoed	Recurring Model Information Aggregate.
<ModPending>	Boolean	Optional Profiled support Echoed	Modify Pending Flag.
<RecChkOrdRec>	Aggregate *see section 6.3.21*	Required	Recurring Check Order Record Aggregate.
<CSPRefId>	Identifier	Optional	Customer Service Provider Reference Identifier.
<SPRefId>	Identifier	Optional	Service Provider Reference Identifier.

6.9.2.3 Recurring Check Order Cancellation

6.9.2.3.1 Request <RecChkOrdCanRq>

Recurring Check Order Cancellation Request message cancels a previously added recurring check order record.

Tag	Type	Usage	Description
<RqUID>	UUID	Required	Request Identifier. Sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header aggregate.
<AsyncRqUID>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously.
			For more information, see section 3.2.12.1.
<CustId>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<RecChkOrdId>	Identifier	Required	Recurring Check Order Model Identifier. Assigned by the server at the thime the Recurring Check Order Model is first added. Cannot be changed by the client.l
<CascadeDel>	Boolean	Optional	Cascade Delete. If *True*, the server must delete all dependent objects when this object is deleted. If *False* or omitted, the recurring model must not be deleted if dependent Check Orders exist.

6.9.2.3.2 Response <RecChkOrdCanRs>

Tag	Type	Usage	Description
<Status>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, <StatusCode> defaults to 0 (zero).
<RqUID>	UUID	Required Echoed	Request Identifier.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header aggregate.
<MsgRsHdr>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header aggregate.
<AsyncRqUID>	UUID	Optional Echoed	Asynchronous Request Identifier.
<CustId>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.

Tag	Type	Usage	Description
<RecChkOrdId>	Identifier	Required Echoed	Recurring Check Order Model Identifier.
<CascadeDel>	Boolean	Optional Echoed	Cascade Delete Flag.
<RecChkOrdRec>	Aggregate *see section 6.3.21*	Optional XOR	Check Order Model Record Aggregate.
<DependentType>	Open Enum	Optional XOR Repeating	Dependent object type that exists for the customer/service link. One element must be returned for each dependent. Defined values: ChkOrd
<CSPRefId>	Identifier	Optional	Customer Service Provider Reference Identifier.
<SPRefId>	Identifier	Optional	Service Provider Reference Identifier.

6.9.2.4   Recurring Check Order Inquiry

6.9.2.4.1   *Request <RecChkOrdInqRq>*

Recurring Check Order Inquiry Request message retrieves recurring check order records based on selection criteria provided.

Tag	Type	Usage	Description
<RqUID>	UUID	Required	Request Identifier. Sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header aggregate.
<AsyncRqUID>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously. For more information, see section 3.2.12.1.
<CustId>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<RecCtrlIn>	Aggregate *see section 3.2.12.2.1*	Optional	Record Control In.
<DepAcctId>	Aggregate *see section 3.2.7.1.2*	Required XOR	Deposit Account Identifier Aggregate.
<CardAcctId>	Aggregate *see section 3.2.7.1.3*	Required XOR	Credt Card Account Identifier Aggregate.
<LoanAcctId>	Aggregate *see section 3.2.7.1.4*	Required XOR	Loan Account Identifier Aggregate.

Tag	Type	Usage	Description
<RecChkOrdId>	Identifier	Optional Repeating	Recurring Check Order Identifier. This field is used as a selection criterion to select only check order instances generated by Recurring Check Order Model. Inclusion of <RecChkOrdId> must not cause this message to include the recurring check order model itself. The model may be retrieved using the Recurring Check Order Inquiry.
<CSPRefId>	Identifier	Optional Repeating	Customer Service Provider Reference Identifier. This field is used as a selection criterion.
<SPRefId>	Identifier	Optional Repeating	Service Provider Reference Identifier. This field is used as a selection criterion.
<SelRangeDt>	Aggregate *see section 3.2.10.1*	Optional	Selection Date Range.
<IncToken>	Boolean	Optional	Include Token. If *True*, a <Token> should be included in the response to set a base for future Audit messages. If *False* or omitted, no <Token> is returned.

6.9.2.4.2   Response <RecChkOrdInqRs>

Tag	Type	Usage	Description
<Status>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, <StatusCode> defaults to 0 (zero).
<RqUID>	UUID	Required Echoed	Request Identifier.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header aggregate.
<MsgRsHdr>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header aggregate.
<AsyncRqUID>	UUID	Optional Echoed	Asynchronous Request Identifier.
<CustId>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<RecCtrlOut>	Aggregate *see section 3.2.12.2.2*	Optional *but see Description*	Record Control Output Aggregate. Required if <RecCtrlIn> was provided in the request and the server supports Record Control.
<DepAcctId>	Aggregate *see section 3.2.7.1.2*	Required XOR Echoed	Deposit Account Identifier Aggregate.
<CardAcctId>	Aggregate *see section 3.2.7.1.3*	Required XOR Echoed	Credit Card Account Identifier Aggregate.
<LoanAcctId>	Aggregate *see section 3.2.7.1.4*	Required XOR Echoed	Loan Account Identifier Aggregate.

Tag	Type	Usage	Description
<RecChkOrdId>	Identifier	Optional Repeating Echoed	Recurring Check Order Identifier. This field is used as a selection criterion to select only check order instances generated by a Recurring Check Order Model. Inclusion of <RecChkOrdId> must not cause this message to include the recurring check order model itself. The model may be retrieved using the Recurring Check Order Inquiry.
<CSPRefId>	Identifier	Optional Repeating Echoed	Customer Service Provider Reference Identifier. This field is used as a selection criterion.
<SPRefId>	Identifier	Optional Repeating Echoed	Service Provider Reference Identifier. This field is used as a selection criterion.
<SelRangeDt>	Aggregate *see section 3.2.10.1*	Optional	Selection Date Range.
<RecChkOrdRec>	Aggregate *see section 6.3.21*	Optional Repeating	Check Order Record Aggregate.
<Token>	Identifier	Optional *but see Description*	Token. Server Assigned. Client should make no assumptions about the value of this token relative to others that it may have received. The token is significant only to the server that originally assigned it. The token is only included if <IncToken>=*True* in the request. <Token>=0 is returned if no records are returned within the response.

6.9.2.5   Recurring Check Order Audit

6.9.2.5.1   Request <RecChkOrdAudRq>

Recurring Check Order Audit Request message audits a previously added recurring check order record.

Tag	Type	Usage	Description
<RqUID>	UUID	Required	Request Identifier. Sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header aggregate.
<AsyncRqUID>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously. For more information, see section 3.2.12.1.
<CustId>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<RecCtrlIn>	Aggregate *see section 3.2.12.2.1*	Optional	Message Record Control Input Aggregate.

Tag	Type	Usage	Description
\<SelRangeDt>	Aggregate *see section 3.2.10.1*	Optional	
\<Method>	Closed Enum	Optional Repeating	Action. Used to identify actions associated with the object that is being audited (e.g. payment, ChkOrd, etc.).
			Valid values: Add, Mod, Can
			This field is used as selection criterion.
\<RecChkOrdId>	Identifier	Optional Repeating	Recurring Check Order Identifier. Assigned by the server at the time the Recurring Check Order Model is first added. Cannot be changed by the client.
			This field is used as selection criterion.

6.9.2.5.2 Response \<RecChkOrdAudRs>

Tag	Type	Usage	Description
\<Status>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, \<StatusCode> defaults to 0 (zero).
\<RqUID>	UUID	Required Echoed	Request Identifier.
\<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header aggregate.
\<MsgRsHdr>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header aggregate.
\<AsyncRqUID>	UUID	Optional Echoed	Asynchronous Request Identifier.
\<CustId>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in \<SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then \<CustId> is required, and must contain the value of the user whose request is being issued.
\<RecCtrlOut>	Aggregate *see section 3.2.12.2.2*	Optional *but see Description*	Record Control Output Aggregate.
			Required if \<RecCtrlIn> was provided in the request and the server supports Record Control.
\<SelRangeDt>	Aggregate *see section 3.2.10.1*	Optional Echoed	Selection Range Date Aggregate.
\<Method>	Closed Enum	Optional Repeating Echoed	Action. Used to identify actions associated with the object that is being audited (e.g. payment, ChkOrd, etc.).
			This field is used as selection criterion.
\<RecChkOrdId>	Identifier	Optional Repeating Echoed	Recurring Check Order Identifier. Assigned by the server at the time the Recurring Check Order Model is first added. Cannot be changed by the client.
			This field is used as selection criterion.
\<RecChkOrdMsgRec>	Aggregate	Optional Repeating	Recurring Check Order Message Record Aggregate.

Tag	Type	Usage	Description
<CustId>	Aggregate *see section 3.2.3.1*	Optional Repeating	Customer Identification Aggregate.
<MsgRecDt>	DateTime	Optional	Message Record Creating Date. The date/time at which the message record was stored/created by the service provider.
<RecChkOrdAddRs>	Aggregate *see section 6.9.2.1.2*	Required XOR	Recurring Check Order Add Response Message Aggregate.
<RecChkOrdModRs>	Aggregate *see section 6.9.2.2.2*	Required XOR	Recurring Check Order Modify Response Message Aggregate.
<RecChkOrdCanRs>	Aggregate *see section 6.9.2.3.2*	Required XOR	Recurring Check Order Cancel Response Message Aggregate.
</RecChkOrdMsgRec>			

6.9.2.6   Recurring Check Order Synchronization

6.9.2.6.1   Request <RecChkOrdSyncRq>

Recurring Check Order Synchronization Request message synchronizes the client with the server.

Tag	Type	Usage	Description
<RqUID>	UUID	Required	Request Identifier. Sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header aggregate.
<AsyncRqUID>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously. For more information, see section 3.2.12.1.
<CustId>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<RecCtrlIn>	Aggregate *see section 3.2.12.2.1*	Optional	Message Record Control Input Aggregate.
<Token>	Aggregate *see section 3.1.7*	Required	Token. Server assigned. Client should make no assumption about the value of this token relative to others that it may have received. The token is significant only to the server that originally assigned it. This is a token that has been previously sent by the server.
<DepAcctId>	Aggregate *see section 3.2.7.1.2*	Required XOR	Deposit Account Identification Aggregate.

Tag	Type	Usage	Description
<CardAcctId>	Aggregate *see section 3.2.7.1.3*	Required XOR	Card Account Identification Aggregate.
<LoanAcctId>	Aggregate *see section 3.2.7.1.4*	Required XOR	Loan Account Identification Aggregate.

6.9.2.6.2   Response <RecChkOrdSyncRs>

Tag	Type	Usage	Description
<Status>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, <StatusCode> defaults to 0 (zero).
<RqUID>	UUID	Required Echoed	Request Identifier.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header aggregate.
<MsgRsHdr>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header aggregate.
<AsyncRqUID>	UUID	Optional Echoed	Asynchronous Request Identifier.
<CustId>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<RecCtrlOut>	Aggregate *see section 3.2.12.2.2*	Optional *but see Description*	Record Control Output Aggregate. Required if <RecCtrlIn> was provided in the request and the server supports Record Control.
<DepAcctId>	Aggregate *see section 3.2.7.1.2*	Required XOR Echoed	Deposit Account Identification Aggregate.
<CardAcctId>	Aggregate *see section 3.2.7.1.3*	Required XOR Echoed	Card Account Identification Aggregate.
<LoanAcctId>	Aggregate *see section 3.2.7.1.4*	Required XOR Echoecd	Loan Account Identification Aggregate.
<NewToken>	Identifier	Required	New Token. Server assigned. Client should make no assumptions about the value of this token relative to others that it may have received. The token is significant only to the server that originally assigned it.
<RecChkOrdMsgRec>	Aggregate	Optional Repeating	Recurring Check Order Message Record Aggregate.
<CustId>	Aggregate *see section 3.2.3.1*	Optional Repeating	Customer Identification Aggregate.
<MsgRecDt>	DateTime	Optional	Message Record Creating Date. The date/time at which the message record was stored/created by the service provider.

Tag	Type	Usage	Description
<RecChkOrdAddRs>	Aggregate *see section 6.9.2.1.2*	Required XOR	Recurring Check Order Add Response Message Aggregate.
<RecChkOrdModRs>	Aggregate *see section 6.9.2.2.2*	Required XOR	Recurring Check Order Modify Response Message Aggregate.
<RecChkOrdCanRs>	Aggregate *see section 6.9.2.3.2*	Required XOR	Recurring Check Order Cancel Response Message Aggregate.
</RecChkOrdMsgRec>			

6.9.2.7   Recurring Check Order Instance Add

The Recurring Check Order Instance Add messages allow a client to manually trigger the spawning of a check order instance from a Recurring Check Order Model that has its frequency value defined as "Manually". This message is particularly useful because check orders occur on an irregular frequency basis. Values provided within <ChkOrdInfo> override the values already specified within the recurring check order model specified by <RecChkOrdId>

6.9.2.7.1   Request <RecChkOrdInstAddRq>

Tag	Type	Usage	Description
<RqUID>	UUID	Required	Request Identifier. Sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header aggregate.
<AsyncRqUID>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously.
			For more information, see section 3.2.12.1.
<CustId>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<RecChkOrdId>	Identifier	Required	Recurring Check Order Model Identifier. The identifier for the recurring check order model.
<ChkOrdDetail>	Aggregate *see section 6.3.22*	Optional	Check Order Detail.

6.9.2.7.2   Response <RecChkOrdInstAddRs>

Tag	Type	Usage	Description
<Status>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, <StatusCode> defaults to 0 (zero).
<RqUID>	UUID	Required Echoed	Request Identifier.

Tag	Type	Usage	Description
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header aggregate.
<MsgRsHdr>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header aggregate.
<AsyncRqUID>	UUID	Optional Echoed	Asynchronous Request Identifier.
<CustId>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<RecChkOrdId>	Identifier	Required Echoed	Recurring Check Order Model Identifier. The identifier for the recurring check order model.
<ChkOrdDetail>	Aggregate *see section 6.3.22*	Optional Echoed	Check Order Detail.

6.9.3 Deposit Book Order

The Deposit Book Order message allows a client to order a Deposit Book from a Financial Institution.

6.9.3.1 Request <DepBkOrdAddRq>

The client must specify a client identifier, account identifier, number of deposit slips, deposit book style, and delivery method when ordering a Deposit Book.

Tag	Type	Usage	Description
<RqUID>	UUID	Required	Request Identifier. Sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header aggregate.
<AsyncRqUID>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously. For more information, see section 3.2.12.1.
<CustId>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<DepBkOrdInfo>	Aggregate *see section 6.3.23.1*	Required	Deposit Book Order Info Aggregate.

6.9.3.2 Response <DepBkOrdAddRs>

Tag	Type	Usage	Description
<Status>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, <StatusCode> defaults to 0 (zero).
<RqUID>	UUID	Required Echoed	Request Identifier.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header aggregate.
<MsgRsHdr>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header aggregate.
<AsyncRqUID>	UUID	Optional Echoed	Asynchronous Request Identifier.
<CustId>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<DepBkOrdInfo>	Aggregate *see section 6.3.23.1*	Required Echo	Deposit Book Order Info Aggregate.
<DepBkOrdRec>	Aggregate *see section 6.3.23*	Required	Deposit Book Order Record Aggregate.
<CSPRefId>	Identifier	Optional	Customer Service Provider Reference Identifier.
<SPRefId>	Identifier	Optional	Service Provider Reference Identifier.

6.9.4 Deposit Book Order Reversal

6.9.4.1 Request <DepBkOrdRevRq>

Tag	Type	Usage	Description
<RqUID>	UUID	Required	Request Identifier.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header aggregate.
<AsyncRqUID>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously. For more information, see section **3.2.12.1**.
<CustId>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.

Tag	Type	Usage	Description
<RevReasonCode>	Open Enum	Required	Reversal reason code. The reason why the client is requesting a reversal.
<Desc>	C-80	Optional	Description, text describing the reason for the reversal.
<DepBkOrdMsgRqInfo>	Aggregate	Required	Deposit Book Order Message Request Information Aggregate. This aggregate contains the message to reverse. See RevOptSupt in the service profile for specific reversal messages supported.
<DepBkOrdAddRq>	Aggregate *see section 6.9.3.1*	Optional	Deposit Book Order Add Request Message Aggregate.
</DepBkOrdMsgRqInfo>			

6.9.4.2   Response <DepBkOrdRevRs>

Tag	Type	Usage	Description
<Status>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, <StatusCode> defaults to 0 (zero).
<RqUID>	UUID	Required Echoed	Request Identifier.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header aggregate.
<MsgRsHdr>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header aggregate.
<AsyncRqUID>	UUID	Optional Echoed	Asynchronous Request Identifier.
<CustId>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Identification Aggregate.
<RevReasonCode>	Open Enum	Required Echoed	Reversal reason code. The reason why the client is requesting a reversal.
<Desc>	C-80	Optional Echoed	Description, text describing the reason for the reversal.
<DepBkOrdMsgRqInfo>	Aggregate	Required Echoed	Deposit Book Order Message Request Information Aggregate. This aggregate contains the message to reverse. See RevOptSupt in the service profile for specific reversal messages supported.
<DepBkOrdAddRq>	Aggregate *see section 6.9.3.1*	Required	Deposit Book Order Add Request Message Aggregate.
</DepBkOrdMsgRqInfo>			
<CSPRefId>	Identifier	Optional	Customer Service Provider Reference Identifier.
<SPRefId>	Identifier	Optional	Service Provider Reference Identifier.

6.10   Debit and Credit

6.10.1   Debit Messages

The debit messages are used to obtain authorization and perform a debit of money from an account for a cash (or other valuable media) withdrawal or the purchase of merchandise.

In the POS environment, for some merchandise purchases it is common that an authorization is required before the purchase is completed. To perform this the DebitAuthAddRq message is used to get authorization for the amount to be debited from the specified account. At this stage the funds may be held, but not yet debited from the account. On receipt of a good response, the DebitAddRq message is used to fulfill the merchandise purchase. The actual amount dispensed may be less than that previously authorized, but in any case the DebitAdd message should initiate the actual debit and stop the hold on the funds. The link between the Debit Authorization and the actual Debit is by means of the DebitAuthId, which is supplied by the server in the DebitAuthAdd response.

In an ATM or Teller environment, to perform a cash (or other valuable media) withdrawal, the DebitAddRq message is used to debit the specified amount from an account. On receipt of a good response, the cash (or other valuable media) is dispensed.

To modify a previous debit authorization or debit the DebitAuthModRq or DebitModRq messages can be used. The modify messages are typically used in situations where only a partial amount of money requested could be dispensed. If the customer cancels the Debit authorization or Debit, or the terminal could not dispense any of the debit, the client terminal can use the DebitAuthRevRq or DebitRevRq messages to reverse the previous debit authorization or debit.

6.10.1.1   Debit Authorization Add

The Debit Authorization Add message is used to request authorization for a debit.

6.10.1.1.1 Request <DebitAuthAddRq>

Tag	Type	Usage	Description
<RqUID>	UUID	Required	Request Identifier.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header aggregate.
<AsyncRqUID>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously.

For more information, see section 3.2.12.1. |
| **<CustId>** | Aggregate *see section 3.2.3.1* | Optional | Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued. |
| **<DebitAuthInfo>** | Aggregate *see section 6.3.26.1* | Required | Debit Authorization Information Aggregate |

6.10.1.1.2 Response <DebitAuthAddRs>

Tag	Type	Usage	Description

Tag	Type	Usage	Description
\<Status\>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, \<StatusCode\> defaults to 0 (zero).
\<RqUID\>	UUID	Required Echoed	Request Identifier.
\<MsgRqHdr\>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header aggregate.
\<MsgRsHdr\>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header aggregate.
\<AsyncRqUID\>	UUID	Optional Echoed	Asynchronous Request Identifier.
\<CustId\>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Identification Aggregate.
\<DebitAuthInfo\>	Aggregate *see section 6.3.26.1*	Required Echoed	Debit Authorization Info Aggregate
\<DebitAuthRec\>	Aggregate *see section 6.3.26*	Required	Debit Authorization Record Aggregate.
\<CSPRefId\>	Identifier	Optional	Customer Service Provider Reference Identifier.
\<SPRefId\>	Identifier	Optional	Service Provider Reference Identifier.

6.10.1.2 Debit Authorization Modify

The Debit Authorization Modify message is used to modify the Information previously supplied in a Debit Authorization Add request.

6.10.1.2.1 Request \<DebitAuthModRq\>

Tag	Type	Usage	Description
\<RqUID\>	UUID	Required	Request Identifier.
\<MsgRqHdr\>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header aggregate.
\<AsyncRqUID\>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously. For more information, see section 3.2.12.1.
\<CustId\>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in \<SignonRq\>. For example, if a CSR or SP issues the request on behalf of the user, then \<CustId\> is required, and must contain the value of the user whose request is being issued.
\<DebitAuthId\>	Identifier	Required	Debit Authorization Identifier. Assigned by the server at the time the Debit Authorization is first added. Cannot be changed by the client.

Tag	Type	Usage	Description
<DebitAuthInfo>	Aggregate *see section 6.3.26.1*	Required	Debit Authorization Info Aggregate

6.10.1.2.2 Response <DebitAuthModRs>

Tag	Type	Usage	Description
<Status>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, <StatusCode> defaults to 0 (zero).
<RqUID>	UUID	Required Echoed	Request Identifier.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header aggregate.
<MsgRsHdr>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header aggregate.
<AsyncRqUID>	UUID	Optional Echoed	Asynchronous Request Identifier.
<CustId>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Identification Aggregate.
<DebitAuthId>	Identifier	Required Echoed	Debit Authorization Identifier
<DebitAuthInfo>	Aggregate *see section 6.3.26.1*	Required Echoed	Debit Authorization Info Aggregate
<DebitAuthRec>	Aggregate *see section 6.3.26*	Required	Debit Authorization Record Aggregate
<CSPRefId>	Identifier	Optional	Customer Service Provider Reference Identifier.
<SPRefId>	Identifier	Optional	Service Provider Reference Identifier.

6.10.1.3 Debit Authorization Cancel
The client uses this message to cancel or reverse a previously added debit authorization.

6.10.1.3.1 Request <DebitAuthCanRq>

Tag	Type	Usage	Description
<RqUID>	UUID	Required	Request Identifier.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header aggregate.
<AsyncRqUID>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously. For more information, see section 3.2.12.1.

Tag	Type	Usage	Description
<CustId>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<DebitAuthId>	Identifier	Required	Debit Authorization Identifier. Assigned by the server at the time the Debit Authorization is first added. Cannot be changed by the client.
<ClientChgCode>	Open Enum	Required	Client Change Code. This indicates the reason why an Authorization was cancelled or modified by the client, or why the fulfillment amount differed from the authorized amount. Defined Values: None, ConsumerCancelled, TerminalExceptionAmountKnown, TerminalExceptionAmountUnknown, ConsumerExceptionAmountKnown, ConsumerExceptionAmountUnknown This element will be removed from the DebitAuthCanRq message in IFX 2.0.

6.10.1.3.2 Response <DebitAuthCanRs>

Tag	Type	Usage	Description
<Status>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, <StatusCode> defaults to 0 (zero).
<RqUID>	UUID	Required Echoed	Request Identifier.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header aggregate.
<MsgRsHdr>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header aggregate.
<AsyncRqUID>	UUID	Optional Echoed	Asynchronous Request Identifier.
<CustId>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Identification Aggregate.
<DebitAuthId>	Identifier	Required Echoed	Debit Authorization Identifier
<DebitAuthRec>	Aggregate *see section 6.3.26*	Optional	Debit Authorization Record Aggregate.
<ClientChgCode>	Open Enum	Required Echoed	Client Change Code. This element will be removed from the DebitAuthCanRs message in IFX 2.0.
<CSPRefId>	Identifier	Optional	Customer Service Provider Reference Identifier.
<SPRefId>	Identifier	Optional	Service Provider Reference Identifier.

6.10.1.4 Debit Authorization Inquiry

6.10.1.4.1 Request <DebitAuthInqRq>

Tag	Type	Usage	Description
<RqUID>	UUID	Required	Request Identifier.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header aggregate.
<AsyncRqUID>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously.
			For more information, see section 3.2.12.1.
<CustId>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<RecCtrlIn>	Aggregate *see section 3.2.12.2.1*	Optional	Records Control Input Aggregate.
<DebitAuthId>	Identifier	Optional Repeating	Debit Authorization Identifier.
<CSPRefId>	Identifier	Optional Repeating	Customer Service Provider Reference Identifier. This field is used as a selection criterion.
<SPRefId>	Identifier	Optional Repeating	Service Provider Reference Identifier. This field is used as a selection criterion.
<DepAcctId>	Aggregate *see section 3.2.7.1.2*	Optional XOR	Deposit Account Identification. This field is used as a selection criterion.
<CardAcctId>	Aggregate *see section 3.2.7.1.3*	Optional XOR	Card Account Identification. This field is used as a selection criterion.
<ClientChgCode>	Open Enum	Optional Repeating	Client Change Code. This field is used as a selection criterion. This element will be removed from the DebitAuthInqRq message in IFX 2.0.
<SelRangeDt>	Aggregate *see section 3.2.10.1*	Optional	Selection Range Date Aggregate.
<SelRangeCurAmt>	Aggregate *see section 3.2.10.2*	Optional	Selection Range Amount Aggregate.
<IncToken>	Boolean	Optional	Include Token. If *True*, a <Token> should be included in the response to set a base for future Sync messages. If *False* or omitted, no <Token> is returned.

6.10.1.4.2 Response <DebitAuthInqRs>

Tag	Type	Usage	Description
\<Status>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, \<StatusCode> defaults to 0 (zero).
\<RqUID>	UUID	Required Echoed	Request Identifier.
\<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header aggregate.
\<MsgRsHdr>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header aggregate.
\<AsyncRqUID>	UUID	Optional Echoed	Asynchronous Request Identifier.
\<CustId>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Identification Aggregate.
\<RecCtrlOut>	Aggregate *see section 3.2.12.2.2*	Optional *but see Description*	Records Control Output Aggregate. Required if \<RecCtrlIn> was provided in the request and the server supports Records Control.
\<DebitAuthId>	Identifier	Optional Repeating Echoed	Debit Authorization Identifier.
\<CSPRefId>	Identifier	Optional Repeating Echoed	Customer Service Provider Reference Identifier. This field is used as a selection criterion.
\<SPRefId>	Identifier	Optional Repeating Echoed	Service Provider Reference Identifier. This field is used as a selection criterion.
\<DepAcctId>	Aggregate *see section 3.2.7.1.2*	Optional XOR Echoed	Deposit Account Identification. This field is used as a selection criterion.
\<CardAcctId>	Aggregate *see section 3.2.7.1.3*	Optional XOR Echoed	Card Account Identification. This field is used as a selection criterion.
\<ClientChgCode>	Closed Enum	Optional Repeating Echoed	Client Change Code. This field is used as a selection criterion. This element will be removed from the DebitAuthInqRs message in IFX 2.0.
\<SelRangeDt>	Aggregate *see section 3.2.10.1*	Optional Echoed	Selection Range Date Aggregate.
\<SelRangeCurAmt>	Aggregate *see section 3.2.10.2*	Optional Echoed	Selection Range Amount Aggregate.
\<DebitAuthRec>	Aggregate *see section 6.3.26*	Optional Repeating	Debit Authorization Record Aggregate. These records are generated by the server and reflect the current state of the customer's Debit Authorizations. The records are filtered by the selection criteria specified in the request message.

Tag	Type	Usage	Description
<Token>	Identifier	Optional *but see Description*	Token. Server Assigned. Client should make no assumptions about the value of this token relative to others that it may have received. The token is significant only to the server that originally assigned it. The token is only included if <IncToken>=*True* in the request. <Token>=0 is returned if no records are returned within the response.

6.10.1.5   Debit Authorization Audit

6.10.1.5.1 Request <DebitAuthAudRq>

Tag	Type	Usage	Description
<RqUID>	UUID	Required	Request Identifier.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header aggregate.
<AsyncRqUID>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously. For more information, see section 3.2.12.1.
<CustId>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<RecCtrlIn>	Aggregate *see section 3.2.12.2.1*	Optional	Records Control Input Aggregate.
<SelRangeDt>	Aggregate *see section 3.2.10.1*	Optional	Selection Range Date Aggregate.
<Method>	Closed Enum	Optional Repeating	Audit Selection Action. Used to identify actions associated with the object that is being audited. Valid values: Add, Mod, Can. This field is used as a selection criterion.
<DebitAuthId>	Identifier	Optional Repeating	Debit Authorization Identifier. Assigned by the server at the time the Debit Authorization is first added. Cannot be changed by the client. This field is used as a selection criterion.

6.10.1.5.2 Response <DebitAuthAudRs>

Tag	Type	Usage	Description
<Status>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, <StatusCode> defaults to 0 (zero).
<RqUID>	UUID	Required Echoed	Request Identifier.

Tag	Type	Usage	Description
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header aggregate.
<MsgRsHdr>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header aggregate.
<AsyncRqUID>	UUID	Optional Echoed	Asynchronous Request Identifier.
<CustId>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Identification Aggregate.
<RecCtrlOut>	Aggregate *see section 3.2.12.2.2*	Optional *but see Description*	Message Records Control Output Aggregate. Required if <RecCtrlIn> was provided in the request and the server supports Records Control.
<SelRangeDt>	Aggregate *see section 3.2.10.1*	Optional Echoed	Selection Range Date Aggregate.
<Method>	Closed Enum	Optional Repeating Echoed	Audit Selection Action. This field is used as a selection criterion.
<DebitAuthId>	Identifier	Optional Repeating Echoed	Debit Authorization Identifier. Assigned by the server at the time the Debit Authorization is first added. Cannot be changed by the client. This field is used as a selection criterion.
<DebitAuthMsgRec>	Aggregate	Optional Repeating	Debit Authorization Message Record Aggregate.
<CustId>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate.
<MsgRecDt>	Date/Time	Optional	Message Record Creation Date. The date/time at which the message record was stored/created by the service provider.
<DebitAuthAddRs>	Aggregate *see section 6.10.1.1.2*	Required XOR	Debit Authorization Add Response Message Aggregate.
<DebitAuthModRs>	Aggregate *see section 6.10.1.2.2*	Required XOR	Debit Authorization Modify Response Message Aggregate.
<DebitAuthRevRs>	Aggregate *see section 6.10.1.8.2*	Required XOR	Debit Authorization Reversal Response Message Aggregate.
<DebitAuthStatusModRs>	Aggregate *see section 6.10.1.7.2*	Required XOR	Debit Authorization Status Modify Response Message Aggregate.
<DebitAuthCanRs>	Aggregate *see section 6.10.1.3.2*	Required XOR	Debit Authorization Cancel Response Message Aggregate.
</DebitAuthMsgRec>			

6.10.1.6 Debit Authorization Synchronization

6.10.1.6.1 Request <DebitAuthSyncRq>

Tag	Type	Usage	Description
<RqUID>	UUID	Required	Request Identifier.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header aggregate.
<AsyncRqUID>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously.
			For more information, see section 3.2.12.1.
<CustId>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<RecCtrlIn>	Aggregate *see section 3.2.12.2.1*	Optional	Message Records Control Input Aggregate.
<Token>	Identifier	Required	Token. Server assigned. Client should make no assumptions about the value of this token relative to others that it may have received. The token is significant only to the server that originally assigned it.
			This is a token that has been previously sent by the server, or zero for first time requests.
<DepAcctId>	Aggregate *see section 3.2.7.1.2*	Required XOR	Deposit Account Identification Aggregate.
<CardAcctId>	Aggregate *see section 3.2.7.1.3*	Required XOR	Card Account Identification Aggregate.

6.10.1.6.2 Response <DebitAuthSyncRs>

Tag	Type	Usage	Description
<Status>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, <StatusCode> defaults to 0 (zero).
<RqUID>	UUID	Required Echoed	Request Identifier.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header aggregate.
<MsgRsHdr>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header aggregate.
<AsyncRqUID>	UUID	Optional Echoed	Asynchronous Request Identifier.

Tag	Type	Usage	Description
<CustId>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Identification Aggregate.
<RecCtrlOut>	Aggregate *see section 3.2.12.2.2*	Optional *but see Description*	Records Control Output Aggregate. Required if <RecCtrlIn> was provided in the request and the server supports Records Control.
<DepAcctId>	Aggregate *see section 3.2.7.1.2*	Required XOR	Deposit Account Identification Aggregate.
<CardAcctId>	Aggregate *see section 3.2.7.1.3*	Required XOR	Card Account Identification Aggregate.
<Token>	Identifier	Required Echoed	Token. Server assigned. Client should make no assumptions about the value of this token relative to others that it may have received. The token is significant only to the server that originally assigned it. This is a token that has been previously sent by the server, or zero for first time requests.
<NewToken>	Identifier	Required	New Token. Server assigned. Client should make no assumptions about the value of this token relative to others that it may have received. The token is significant only to the server that originally assigned it.
<DebitAuthMsgRec>	Aggregate	Optional Repeating	Debit Message Record Aggregate.
<CustId>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate.
<MsgRecDt>	Date/Time	Optional	Message Record Creation Date. The date/time at which the message record was stored/created by the service provider.
<DebitAuthAddRs>	Aggregate *see section 6.10.1.1.2*	Required XOR	Debit Authorization Add Response Message Aggregate.
<DebitAuthModRs>	Aggregate *see section 6.10.1.2.2*	Required XOR	Debit Authorization Modify Response Message Aggregate.
<DebitAuthCanRs>	Aggregate *see section 6.10.1.3.2*	Required XOR	Debit Authorization Cancel Response Message Aggregate.
</DebitAuthMsgRec>			

6.10.1.7 Debit Authorization Status Modify

6.10.1.7.1 Request <DebitAuthStatusModRq>

Tag	Type	Usage	Description
<RqUID>	UUID	Required	Request Identifier.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header Aggregate.

Tag	Type	Usage	Description
\<AsyncRqUID\>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously.
			For more information, see section **3.2.12.1**.
\<CustId\>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in \<SignonRq\>. For example, if a CSR or SP issues the request on behalf of the user, then \<CustId\> is required, and must contain the value of the user whose request is being issued.
\<DebitAuthId\>	Identifier	Required	Debit Authorization Identifier.
\<DebitAuthStatus\>	Aggregate *see section 6.3.26.2*	Required	Debit Authorization Status Aggregate.

6.10.1.7.2 Response \<DebitAuthStatusModRs\>

Tag	Type	Usage	Description
\<Status\>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, \<StatusCode\> defaults to 0 (zero).
\<RqUID\>	UUID	Required Echoed	Request Identifier.
\<MsgRqHdr\>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header Aggregate.
\<MsgRsHdr\>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header Aggregate.
\<AsyncRqUID\>	Identifier	Optional Echoed	Asynchronous Request Identifier.
\<CustId\>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in \<SignonRq\>. For example, if a CSR or SP issues the request on behalf of the user, then \<CustId\> is required, and must contain the value of the user whose request is being issued.
\<DebitAuthId\>	UUID	Required Echoed	Debit Authorization Identifier.
\<DebitAuthStatus\>	Aggregate *see section 6.3.26.2*	Required Echoed	Debit Authorization Status Aggregate.
\<CSPRefId\>	Identifier	Optional	Customer Service Reference Identifier.
\<SPRefId\>	Identifier	Optional	Service Provider Reference Identifier.

6.10.1.8 Debit Authorization Reversal

6.10.1.8.1 Request \<DebitAuthRevRq\>

Tag	Type	Usage	Description

Tag	Type	Usage	Description
\<RqUID\>	UUID	Required	Request Identifier.
\<MsgRqHdr\>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header aggregate.
\<AsyncRqUID\>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously.
			For more information, see section **3.2.12.1**.
\<CustId\>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in \<SignonRq\>. For example, if a CSR or SP issues the request on behalf of the user, then \<CustId\> is required, and must contain the value of the user whose request is being issued.
\<RevReasonCode\>	Open Enum	Required	Reversal reason code. The reason why the client is requesting a reversal.
\<Desc\>	C-80	Optional	Description, text describing the reason for the reversal.
\<DebitAuthMsgRqInfo\>	Aggregate	Required	Debit Authorization Message Request Information Aggregate. This aggregate contains the message to reverse. See RevOptSupt in the service profile for specific reversal messages supported.
\<DebitAuthAddRq\>	Aggregate *see section 6.10.1.1.1*	Required XOR	Debit Authorization Add Request Message Aggregate.
\<DebitAuthModRq\>	Aggregate *see section 6.10.1.2.1*	Required XOR	Debit Authorization Modify Request Message Aggregate
\<DebitAuthCanRq\>	Aggregate *see section 6.10.1.3.1*	Required XOR	Debit Authorization Cancel Request Message Aggregate
\<DebitAuthInqRq\>	Aggregate *see section 6.10.1.4.1*	Required XOR	Debit Authorization Inquiry Request Message Aggregate
\<DebitAuthAudRq\>	Aggregate *see section 6.10.1.5.1*	Required XOR	Debit Authorization Audit Request Message Aggregate
\<DebitAuthSyncRq\>	Aggregate *see section 6.10.1.6.1*	Required XOR	Debit Authorization Synchronization Message Aggregate
\</DebitAuthMsgRqInfo\>			

6.10.1.8.2 Response \<DebitAuthRevRs\>

Tag	Type	Usage	Description
\<Status\>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, \<StatusCode\> defaults to 0 (zero).
\<RqUID\>	UUID	Required Echoed	Request Identifier.

Tag	Type	Usage	Description
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header aggregate.
<MsgRsHdr>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header aggregate.
<AsyncRqUID>	UUID	Optional Echoed	Asynchronous Request Identifier.
<CustId>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Identification Aggregate.
<RevReasonCode>	Open Enum	Required Echoed	Reversal reason code. The reason why the client is requesting a reversal.
<Desc>	C-80	Optional Echoed	Description, text describing the reason for the reversal.
<DebitAuthMsgRqInfo>	Aggregate	Required Echoed	Debit Authorization Message Request Information Aggregate. This aggregate contains the message to reverse. See RevOptSupt in the service profile for specific reversal messages supported.
<DebitAuthAddRq>	Aggregate *see section 6.10.1.1.1*	Required Echoed XOR	Debit Authorization Add Request Message Aggregate.
<DebitAuthModRq>	Aggregate *see section 6.10.1.2.1*	Required Echoed XOR	Debit Authorization Modify Request Message Aggregate
<DebitAuthCanRq>	Aggregate *see section 6.10.1.3.1*	Required Echoed XOR	Debit Authorization Cancel Request Message Aggregate
<DebitAuthInqRq>	Aggregate *see section 6.10.1.4.1*	Required Echoed XOR	Debit Authorization Inquiry Request Message Aggregate
<DebitAuthAudRq>	Aggregate *see section 6.10.1.5.1*	Required Echoed XOR	Debit Authorization Audit Request Message Aggregate
<DebitAuthSyncRq>	Aggregate *see section 6.10.1.6.1*	Required Echoed XOR	Debit Authorization Synchronization Message Aggregate
</DebitAuthMsgRqInfo>			
<CSPRefId>	Identifier	Optional	Customer Service Provider Reference Identifier.
<SPRefId>	Identifier	Optional	Service Provider Reference Identifier.

6.10.1.9 Debit Add

6.10.1.9.1 Request <DebitAddRq>

The Debit Add message is used for withdrawals at ATM and bank tellers. It is also used for purchases from a POS device.

Tag	Type	Usage	Description
<RqUID>	UUID	Required	Request Identifier.

Tag	Type	Usage	Description
\<MsgRqHdr\>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header aggregate.
\<AsyncRqUID\>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously. For more information, see section 3.2.12.1.
\<CustId\>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in \<SignonRq\>. For example, if a CSR or SP issues the request on behalf of the user, then \<CustId\> is required, and must contain the value of the user whose request is being issued.
\<DebitInfo\>	Aggregate *see section 6.3.27*	Required	Debit Information Aggregate

6.10.1.9.2 Response \<DebitAddRs\>

Tag	Type	Usage	Description
\<Status\>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, \<StatusCode\> defaults to 0 (zero).
\<RqUID\>	UUID	Required Echoed	Request Identifier.
\<MsgRqHdr\>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header aggregate.
\<MsgRsHdr\>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header aggregate.
\<AsyncRqUID\>	UUID	Optional Echoed	Asynchronous Request Identifier.
\<CustId\>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Identification Aggregate.
\<DebitInfo\>	Aggregate *see section 6.3.27*	Required Echoed	Debit Information Aggregate
\<DebitRec\>	Aggregate *see section 6.3.25*	Optional *but see Description*	Debit Record Aggregate. This aggregate will be required in IFX 2.0.
\<CSPRefId\>	Identifier	Optional	Customer Service Provider Reference Identifier.
\<SPRefId\>	Identifier	Optional	Service Provider Reference Identifier.

6.10.1.10 Debit Modify

6.10.1.10.1 Request \<DebitModRq\>

Tag	Type	Usage	Description

Tag	Type	Usage	Description
<RqUID>	UUID	Required	Request Identifier.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header Aggregate.
<AsyncRqUID>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously.
			For more information, see section **3.2.12.1**.
<CustId>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<DebitId>	Identifier	Required	Debit Identifier. Assigned by the server at the time the Debit is first added. Cannot be changed by the client.
<DebitInfo>	Aggregate *see section 6.3.27*	Required	Debit Info Aggregate

6.10.1.10.2 Response <DebitModRs>

Tag	Type	Usage	Description
<Status>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, <StatusCode> defaults to 0 (zero).
<RqUID>	UUID	Required Echoed	Request Identifier.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header Aggregate.
<MsgRsHdr>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header Aggregate.
<AsyncRqUID>	UUID	Optional Echoed	Asynchronous Request Identifier.
<CustId>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Identification Aggregate.
<DebitId>	Identifier	Required Echoed	Debit Identifier
<DebitInfo>	Aggregate *see section 6.3.27*	Required Echoed	Debit Info Aggregate
<DebitRec>	Aggregate *see section 6.3.25*	Required	Debit Record Aggregate
<CSPRefId>	Identifier	Optional	Customer Service Provider Reference Identifier.
<SPRefId>	Identifier	Optional	Service Provider Reference Identifier.

6.10.1.11 Debit Cancel

6.10.1.11.1 Request <DebitCanRq>

Tag	Type	Usage	Description
<RqUID>	UUID	Required	Request Identifier.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header Aggregate.
<AsyncRqUID>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously.
			For more information, see section **3.2.12.1**.
<CustId>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<DebitId>	Identifier	Required	Debit Identifier. Assigned by the server at the time the Debit is first added. Cannot be changed by the client.
<ClientChgCode>	Open Enum	Required	Client Change Code. This indicates the reason why an Authorization was cancelled or modified by the client, or why the fulfillment amount differed from the authorized amount.
			Defined Values:
			None, ConsumerCancelled, TerminalExceptionAmountKnown, TerminalExceptionAmountUnknown, ConsumerExceptionAmountKnown, ConsumerExceptionAmountUnknown

6.10.1.11.2 Response <DebitCanRs>

Tag	Type	Usage	Description
<Status>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, <StatusCode> defaults to 0 (zero).
<RqUID>	UUID	Required Echoed	Request Identifier.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header Aggregate.
<MsgRsHdr>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header Aggregate.
<AsyncRqUID>	UUID	Optional Echoed	Asynchronous Request Identifier.
<CustId>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Identification Aggregate.
<DebitId>	Identifier	Required Echoed	Debit Identifier

Tag	Type	Usage	Description
\<DebitRec\>	Aggregate *see section 6.3.25*	Optional	Debit Record Aggregate.
\<ClientChgCode\>	Open Enum	Required Echoed	Client Change Code.
\<CSPRefId\>	Identifier	Optional	Customer Service Provider Reference Identifier.
\<SPRefId\>	Identifier	Optional	Service Provider Reference Identifier.

6.10.1.12 Debit Inquiry

6.10.1.12.1 Request \<DebitInqRq\>

Tag	Type	Usage	Description
\<RqUID\>	UUID	Required	Request Identifier.
\<MsgRqHdr\>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header Aggregate.
\<AsyncRqUID\>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously. For more information, see section **3.2.12.1**.
\<CustId\>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in \<SignonRq\>. For example, if a CSR or SP issues the request on behalf of the user, then \<CustId\> is required, and must contain the value of the user whose request is being issued.
\<RecCtrlIn\>	Aggregate *see section 3.2.12.2.1*	Optional	Records Control Input Aggregate.
\<DebitId\>	Identifier	Optional Repeating	Debit Identifier.
\<CSPRefId\>	Identifier	Optional Repeating	Customer Service Provider Reference Identifier. This field is used as a selection criterion.
\<SPRefId\>	Identifier	Optional Repeating	Service Provider Reference Identifier. This field is used as a selection criterion.
\<DepAcctId\>	Aggregate *see section 3.2.7.1.2*	Optional XOR	Deposit Account Identification. This field is used as a selection criterion.
\<CardAcctId\>	Aggregate *see section 3.2.7.1.3*	Optional XOR	Card Account Identification. This field is used as a selection criterion.
\<ClientChgCode\>	Open Enum	Optional Repeating	Client Change Code. This field is used as a selection criterion.
\<SelRangeDt\>	Aggregate *see section 3.2.10.1*	Optional	Selection Range Date Aggregate.

Tag	Type	Usage	Description
<SelRangeCurAmt>	Aggregate *see section 3.2.10.2*	Optional	Selection Range Amount Aggregate.
<IncToken>	Boolean	Optional	Include Token. If *True*, a <Token> should be included in the response to set a base for future Sync messages. If *False* or omitted, no <Token> is returned.

6.10.1.12.2 Response <DebitInqRs>

Tag	Type	Usage	Description
<Status>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, <StatusCode> defaults to 0 (zero).
<RqUID>	UUID	Required Echoed	Request Identifier.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header Aggregate.
<MsgRsHdr>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header Aggregate.
<AsyncRqUID>	UUID	Optional Echoed	Asynchronous Request Identifier.
<CustId>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Identification Aggregate.
<RecCtrlOut>	Aggregate *see section 3.2.12.2.2*	Optional *but see Description*	Records Control Output Aggregate. Required if <RecCtrlIn> was provided in the request and the server supports Records Control.
<DebitId>	Identifier	Optional Repeating Echoed	Debit Identifier.
<CSPRefId>	Identifier	Optional Repeating Echoed	Customer Service Provider Reference Identifier. This field is used as a selection criterion.
<SPRefId>	Identifier	Optional Repeating Echoed	Service Provider Reference Identifier. This field is used as a selection criterion.
<DepAcctId>	Aggregate *see section 3.2.7.1.2*	Optional XOR Echoed	Deposit Account Identification. This field is used as a selection criterion.
<CardAcctId>	Aggregate *see section 3.2.7.1.3*	Optional XOR Echoed	Card Account Identification. This field is used as a selection criterion.
<ClientChgCode>	Closed Enum	Optional Repeating Echoed	Client Change Code. This field is used as a selection criterion.
<SelRangeDt>	Aggregate *see section 3.2.10.1*	Optional Echoed	Selection Range Date Aggregate.

Tag	Type	Usage	Description
<SelRangeCurAmt>	Aggregate *see section* *3.2.10.2*	Optional Echoed	Selection Range Amount Aggregate.
<DebitRec>	Aggregate *see section* *6.3.25*	Optional Repeating	Debit Record Aggregate. These records are generated by the server and reflect the current state of the customer's Debit Authorizations. The records are filtered by the selection criteria specified in the request message.
<Token>	Identifier	Optional *but see* *Description*	Token. Server Assigned. Client should make no assumptions about the value of this token relative to others that it may have received. The token is significant only to the server that originally assigned it. The token is only included if <IncToken>=*True* in the request. <Token>=0 is returned if no records are returned within the response.

6.10.1.13 Debit Audit

6.10.1.13.1 Request <DebitAudRq>

Tag	Type	Usage	Description
<RqUID>	UUID	Required	Request Identifier.
<MsgRqHdr>	Aggregate *see section* *3.2.1.1*	Optional	Message Request Header Aggregate.
<AsyncRqUID>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously. For more information, see section **3.2.12.1**.
<CustId>	Aggregate *see section* *3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<RecCtrlIn>	Aggregate *see section* *3.2.12.2.1*	Optional	Records Control Input Aggregate.
<SelRangeDt>	Aggregate *see section* *3.2.10.1*	Optional	Selection Range Date Aggregate.
<Method>	Closed Enum	Optional Repeating	Audit Selection Action. Used to identify actions associated with the object that is being audited. Valid values: Add, Mod, Can. This field is used as a selection criterion.
<DebitId>	Identifier	Optional Repeating	Debit Identifier. Assigned by the server at the time the Debit is first added. Cannot be changed by the client. This field is used as a selection criterion.

6.10.1.13.2 Response <DebitAudRs>

Tag	Type	Usage	Description
\<Status\>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, \<StatusCode\> defaults to 0 (zero).
\<RqUID\>	UUID	Required Echoed	Request Identifier.
\<MsgRqHdr\>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header Aggregate.
\<MsgRsHdr\>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header Aggregate.
\<AsyncRqUID\>	UUID	Optional Echoed	Asynchronous Request Identifier.
\<CustId\>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Identification Aggregate.
\<RecCtrlOut\>	Aggregate *see section 3.2.12.2.2*	Optional *but see Description*	Message Records Control Output Aggregate. Required if \<RecCtrlIn\> was provided in the request and the server supports Records Control.
\<SelRangeDt\>	Aggregate *see section 3.2.10.1*	Optional Echoed	Selection Range Date Aggregate.
\<Method\>	Closed Enum	Optional Repeating Echoed	Audit Selection Action. This field is used as a selection criterion.
\<DebitId\>	Identifier	Optional Repeating Echoed	Debit Identifier. Assigned by the server at the time the Debit is first added. Cannot be changed by the client. This field is used as a selection criterion.
\<DebitMsgRec\>	Aggregate	Optional Repeating	Debit Message Record Aggregate.
\<CustId\>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate.
\<MsgRecDt\>	Date/Time	Optional	Message Record Creation Date. The date/time at which the message record was stored/created by the service provider.
\<DebitAddRs\>	Aggregate *see section 6.10.1.9.2*	Required XOR	Debit Add Response Message Aggregate.
\<DebitModRs\>	Aggregate *see section 6.10.1.10.2*	Required XOR	Debit Modify Response Message Aggregate.
\<DebitRevRs\>	Aggregate *see section 6.10.1.16.2*	Required XOR	Debit Reversal Response Message Aggregate.
\<DebitStatusModRs\>	Aggregate *see section 6.10.1.15.2*	Required XOR	Debit Status Modify Response Message Aggregate.
\<DebitCanRs\>	Aggregate *see section 6.10.1.11.2*	Required XOR	Debit Cancel Response Message Aggregate.
\</DebitMsgRec\>			

6.10.1.14 Debit Synchronization

6.10.1.14.1 Request <DebitSyncRq>

Tag	Type	Usage	Description
<RqUID>	UUID	Required	Request Identifier.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header Aggregate.
<AsyncRqUID>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously. For more information, see section **3.2.12.1**.
<CustId>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<RecCtrlIn>	Aggregate *see section 3.2.12.2.1*	Optional	Message Records Control Input Aggregate.
<Token>	Identifier	Required	Token. Server assigned. Client should make no assumptions about the value of this token relative to others that it may have received. The token is significant only to the server that originally assigned it. This is a token that has been previously sent by the server, or zero for first time requests.
<DepAcctId>	Aggregate *see section 3.2.7.1.2*	Required XOR	Deposit Account Identification Aggregate.
<CardAcctId>	Aggregate *see section 3.2.7.1.3*	Required XOR	Card Account Identification Aggregate.

6.10.1.14.2 Response <DebitSyncRs>

Tag	Type	Usage	Description
<Status>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, <StatusCode> defaults to 0 (zero).
<RqUID>	UUID	Required Echoed	Request Identifier.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header Aggregate.
<MsgRsHdr>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header Aggregate.
<AsyncRqUID>	UUID	Optional Echoed	Asynchronous Request Identifier.

Tag	Type	Usage	Description
<CustId>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Identification Aggregate.
<RecCtrlOut>	Aggregate *see section 3.2.12.2.2*	Optional *but see Description*	Records Control Output Aggregate. Required if <RecCtrlIn> was provided in the request and the server supports Records Control.
<DepAcctId>	Aggregate *see section 3.2.7.1.2*	Required XOR	Deposit Account Identification Aggregate.
<CardAcctId>	Aggregate *see section 3.2.7.1.3*	Required XOR	Card Account Identification Aggregate.
<Token>	Identifier	Required Echoed	Token. Server assigned. Client should make no assumptions about the value of this token relative to others that it may have received. The token is significant only to the server that originally assigned it. This is a token that has been previously sent by the server, or zero for first time requests.
<NewToken>	Identifier	Required	New Token. Server assigned. Client should make no assumptions about the value of this token relative to others that it may have received. The token is significant only to the server that originally assigned it.
<DebitMsgRec>	Aggregate	Optional Repeating	Debit Message Record Aggregate.
<CustId>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate.
<MsgRecDt>	Date/Time	Optional	Message Record Creation Date. The date/time at which the message record was stored/created by the service provider.
<DebitAddRs>	Aggregate *see section 6.10.1.9.2*	Required XOR	Debit Add Response Message Aggregate.
<DebitModRs>	Aggregate *see section 6.10.1.10.2*	Required XOR	Debit Modify Response Message Aggregate.
<DebitStatusModRs>	Aggregate *see section 6.10.1.15.2*	Required XOR	Debit Status Modify Response Message Aggregate.
<DebitCanRs>	Aggregate *see section 6.10.1.11.2*	Required XOR	Debit Cancel Response Message Aggregate.
</DebitMsgRec>			

6.10.1.15 Debit Status Modify

6.10.1.15.1 Request <DebitStatusModRq>

Tag	Type	Usage	Description
<RqUID>	UUID	Required	Request Identifier.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header Aggregate.

Tag	Type	Usage	Description
\<AsyncRqUID\>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously.
			For more information, see section **3.2.12.1**.
\<CustId\>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in \<SignonRq\>. For example, if a CSR or SP issues the request on behalf of the user, then \<CustId\> is required, and must contain the value of the user whose request is being issued.
\<DebitId\>	Identifier	Required	Debit Identifier.
\<DebitStatus\>	Aggregate *see section 6.3.25.1*	Required	Debit Status Aggregate.

6.10.1.15.2 Response \<DebitStatusModRs\>

Tag	Type	Usage	Description
\<Status\>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, \<StatusCode\> defaults to 0 (zero).
\<RqUID\>	UUID	Required Echoed	Request Identifier.
\<MsgRqHdr\>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header Aggregate.
\<MsgRsHdr\>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header Aggregate.
\<AsyncRqUID\>	UUID	Optional Echoed	Asynchronous Request Identifier.
\<CustId\>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in \<SignonRq\>. For example, if a CSR or SP issues the request on behalf of the user, then \<CustId\> is required, and must contain the value of the user whose request is being issued.
\<DebitId\>	Identifier	Required Echoed	Debit Identifier.
\<DebitStatus\>	Aggregate *see section 6.3.25.1*	Required Echoed	Debit Status Aggregate.
\<CSPRefId\>	Identifier	Optional	Customer Service Reference Identifier.
\<SPRefId\>	Identifier	Optional	Service Provider Reference Identifier.

6.10.1.16 Debit Reversal

6.10.1.16.1 Debit Reversal Request \<DebitRevRq\>

Tag	Type	Usage	Description

Tag	Type	Usage	Description
<RqUID>	UUID	Required	Request Identifier. Sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header Aggregate.
<AsyncRqUID>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously.
			For more information, see section **3.2.12.1**.
<CustId>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<RevReasonCode>	Open Enum	Required	Reversal reason code. The reason why the client is requesting a reversal.
<Desc>	C-80	Optional	Description, text describing the reason for the reversal.
<DebitMsgRqInfo>	Aggregate	Required	Debit Message Request Information Aggregate. This aggregate contains the message to reverse. See RevOptSupt in the service profile for specific reversal messages supported.
<DebitAddRq>	Aggregate *see section 6.10.1.9.1*	Required XOR	Debit Add Request Message Aggregate.
<DebitModRq>	Aggregate *see section 6.10.1.10.1*	Required XOR	Debit Modify Request Message Aggregate.
<DebitStatusModRq>	Aggregate *see section 6.10.1.15.1*	Required XOR	Debit Status Modify Request Message Aggregate.
<DebitCanRq>	Aggregate *see section 6.10.1.11.1*	Required XOR	Debit Cancel Request Message Aggregate.
<DebitInqRq>	Aggregate *see section 6.10.1.12.1*	Required XOR	Debit Inquiry Request Message Aggregate.
<DebitAudRq>	Aggregate *see section 6.10.1.13.1*	Required XOR	Debit Audit Request Message Aggregate.
<DebitSyncRq>	Aggregate *see section 6.10.1.14.1*	Required XOR	Debit Sync Request Message Aggregate.
</DebitMsgRqInfo>			

6.10.1.16.2 Debit Reversal Response <DebitRevRs>

Tag	Type	Usage	Description

Tag	Type	Usage	Description
<Status>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, <StatusCode> defaults to 0 (zero).
<RqUID>	UUID	Required Echoed	Request Identifier.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header Aggregate.
<MsgRsHdr>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header Aggregate.
<AsyncRqUID>	UUID	Optional Echoed	Asynchronous Request Identifier.
<CustId>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Identification Aggregate.
<RevReasonCode>	Open Enum	Required Echoed	Reversal reason code.
<Desc>	C-80	Optional Echoed	Description, text describing the reason for the reversal.
<DebitMsgRqInfo>	Aggregate	Required Echoed	Debit Message Request Information Aggregate. This aggregate contains the message to reverse.
<DebitAddRq>	Aggregate *see section 6.10.1.9.1*	Required XOR Echoed	Debit Add Request Message Aggregate.
<DebitModRq>	Aggregate *see section 6.10.1.10.1*	Required XOR Echoed	Debit Modify Request Message Aggregate.
<DebitStatusModRq>	Aggregate *see section 6.10.1.15.1*	Required XOR Echoed	Debit Status Modify Request Message Aggregate.
<DebitCanRq>	Aggregate *see section 6.10.1.11.1*	Required XOR Echoed	Debit Cancel Request Message Aggregate.
<DebitInqRq>	Aggregate *see section 6.10.1.12.1*	Required XOR Echoed	Debit Inquiry Request Message Aggregate.
<DebitAudRq>	Aggregate *see section 6.10.1.13.1*	Required XOR Echoed	Debit Audit Request Message Aggregate.
<DebitSyncRq>	Aggregate *see section 6.10.1.14.1*	Required XOR Echoed	Debit Sync Request Message Aggregate.
</DebitMsgRqInfo>			
<CSPRefId>	Identifier	Optional	Customer Service Provider Reference Identifier.
<SPRefId>	Identifier	Optional	Service Provider Reference Identifier.

6.10.2 Credit Messages

The credit messages are used to obtain authorization to perform a credit of money to an account for cash, check or envelope deposits, or merchandise returns.

A credit can either be performed by itself or require a prior authorization. Typically, in an ATM environment a credit is performed with the use of a CreditAdd message. If a problem occurred during the credit fulfillment and only a partial amount of the credit was received then the CreditModRq message can be used to modify a previously added Credit. If the entire Credit needs to be reversed, the CreditRevRq message can be used.

Credits in the POS or Teller environments sometimes require a prior authorization. For these cases the CreditAuthAddRq message is used to get authorization for the amount to be credited to the specified account. At this stage the funds may be posted, but not yet credited to the account. On receipt of a good response, the credit is fulfilled and the client will send a CreditAddRq message. If a link is required between the Credit Authorization and the actual Credit the CreditAuthId, that is supplied by the server in the CreditAuthAdd response can be used. The CreditAuthModRq message can be used to modify a previously added Credit Authorization. If the Credit Authorization needs to be reversed, the CreditAuthRevRq message can be used.

6.10.2.1   Credit Authorization Add

6.10.2.1.1 Request <CreditAuthAddRq>

The Credit Authorization Add message is used to request authorization for a credit.

Tag	Type	Usage	Description
<RqUID>	UUID	Required	Request Identifier.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header aggregate.
<AsyncRqUID>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously. For more information, see section 3.2.12.1.
<CustId>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<CreditAuthInfo>	Aggregate *see section 6.3.28.1*	Required	Credit Authorization Information Aggregate

6.10.2.1.2 Response <CreditAuthAddRs>

Tag	Type	Usage	Description
<Status>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, <StatusCode> defaults to 0 (zero).
<RqUID>	UUID	Required Echoed	Request Identifier.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header aggregate.
<MsgRsHdr>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header aggregate.
<AsyncRqUID>	UUID	Optional Echoed	Asynchronous Request Identifier.

Tag	Type	Usage	Description
<CustId>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Identification Aggregate.
<CreditAuthInfo>	Aggregate *see section 6.3.28.1*	Required Echoed	Credit Authorization Information Aggregate
<CreditAuthRec>	Aggregate *see section 6.3.28*	Required	Credit Authorization Record Aggregate.
<CSPRefId>	Identifier	Optional	Customer Service Provider Reference Identifier.
<SPRefId>	Identifier	Optional	Service Provider Reference Identifier.

6.10.2.2 Credit Authorization Modify

The Credit Authorization Modify message is used to modify the Information previously supplied in a Credit Authorization Add request.

6.10.2.2.1 Request <CreditAuthModRq>

Tag	Type	Usage	Description
<RqUID>	UUID	Required	Request Identifier.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header aggregate.
<AsyncRqUID>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously. For more information, see section 3.2.12.1.
<CustId>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<CreditAuthId>	Identifier	Required	Credit Authorization Identifier. Assigned by the server at the time the Credit Authorization is first added. Cannot be changed by the client.
<CreditAuthInfo>	Aggregate *see section 6.3.28.1*	Required	Credit Authorization Information Aggregate

6.10.2.2.2 Response <CreditAuthModRs>

Tag	Type	Usage	Description
<Status>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, <StatusCode> defaults to 0 (zero).
<RqUID>	UUID	Required Echoed	Request Identifier.

Tag	Type	Usage	Description
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header aggregate.
<MsgRsHdr>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header aggregate.
<AsyncRqUID>	UUID	Optional Echoed	Asynchronous Request Identifier.
<CustId>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Identification Aggregate.
<CreditAuthId>	Identifier	Required Echoed	Credit Authorization Identifier
<CreditAuthInfo>	Aggregate *see section 6.3.28.1*	Required Echoed	Credit Authorization Information Aggregate
<CreditAuthRec>	Aggregate *see section 6.3.28*	Required	Credit Authorization Record Aggregate
<CSPRefId>	Identifier	Optional	Customer Service Provider Reference Identifier.
<SPRefId>	Identifier	Optional	Service Provider Reference Identifier.

6.10.2.3   Credit Authorization Cancel

The client uses this message to cancel or reverse a previously added credit authorization.

6.10.2.3.1 Request <CreditAuthCanRq>

Tag	Type	Usage	Description
<RqUID>	UUID	Required	Request Identifier.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header aggregate.
<AsyncRqUID>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously. For more information, see section 3.2.12.1.
<CustId>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<CreditAuthId>	Identifier	Required	Credit Authorization Identifier. Assigned by the server at the time the Credit Authorization is first added. Cannot be changed by the client.

Tag	Type	Usage	Description
<ClientChgCode>	Open Enum	Required	Client Change Code. This indicates the reason why an Authorization was cancelled or modified by the client, or why the fulfillment amount differed from the authorized amount.
			Defined Values:
			None, ConsumerCancelled, TerminalExceptionAmountKnown, TerminalExceptionAmountUnknown, ConsumerExceptionAmountKnown, ConsumerExceptionAmountUnknown
			This element will be removed from the CreditAuthCanRq message in IFX 2.0.

6.10.2.3.2 Response <CreditAuthCanRs>

Tag	Type	Usage	Description
<Status>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, <StatusCode> defaults to 0 (zero).
<RqUID>	UUID	Required Echoed	Request Identifier.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header aggregate.
<MsgRsHdr>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header aggregate.
<AsyncRqUID>	UUID	Optional Echoed	Asynchronous Request Identifier.
<CustId>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Identification Aggregate.
<CreditAuthId>	Identifier	Required Echoed	Credit Authorization Identifier
<ClientChgCode>	Open Enum	Required Echoed	Debit / Credit Client Change Code.
			This element will be removed from the CreditAuthCanRs message in IFX 2.0.
<CreditAuthRec>	Aggregate *see section 6.3.28*	Optional	Credit Authorization Record Aggregate.
<CSPRefId>	Identifier	Optional	Customer Service Provider Reference Identifier.
<SPRefId>	Identifier	Optional	Service Provider Reference Identifier.

6.10.2.4   Credit Authorization Inquiry

6.10.2.4.1 Request <CreditAuthInqRq>

Tag	Type	Usage	Description
<RqUID>	UUID	Required	Request Identifier.

Tag	Type	Usage	Description
\<MsgRqHdr\>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header aggregate.
\<AsyncRqUID\>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously. For more information, see section 3.2.12.1.
\<CustId\>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in \<SignonRq\>. For example, if a CSR or SP issues the request on behalf of the user, then \<CustId\> is required, and must contain the value of the user whose request is being issued.
\<RecCtrlIn\>	Aggregate *see section 3.2.12.2.1*	Optional	Records Control Input Aggregate.
\<CreditAuthId\>	Identifier	Optional Repeating	Credit Authorization Identifier.
\<CSPRefId\>	Identifier	Optional Repeating	Customer Service Provider Reference Identifier. This field is used as a selection criterion.
\<SPRefId\>	Identifier	Optional Repeating	Service Provider Reference Identifier. This field is used as a selection criterion.
\<DepAcctId\>	Aggregate *see section 3.2.7.1.2*	Optional XOR	Deposit Account Identification. This field is used as a selection criterion.
\<CardAcctId\>	Aggregate *see section 3.2.7.1.3*	Optional XOR	Card Account Identification. This field is used as a selection criterion.
\<ClientChgCode\>	Open Enum	Optional Repeating	Client Change Code. This field is used as a selection criterion. This element will be removed from the CreditAuthInqRq message in IFX 2.0.
\<SelRangeDt\>	Aggregate *see section 3.2.10.1*	Optional	Selection Range Date Aggregate.
\<SelRangeCurAmt\>	Aggregate *see section 3.2.10.2*	Optional	Selection Range Amount Aggregate.
\<IncToken\>	Boolean	Optional	Include Token. If *True*, a \<Token\> should be included in the response to set a base for future Sync messages. If *False* or omitted, no \<Token\> is returned.

6.10.2.4.2 Response \<CreditAuthInqRs\>

Tag	Type	Usage	Description
\<Status\>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, \<StatusCode\> defaults to 0 (zero).

Tag	Type	Usage	Description
\<RqUID>	UUID	Required Echoed	Request Identifier.
\<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header aggregate.
\<MsgRsHdr>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header aggregate.
\<AsyncRqUID>	UUID	Optional Echoed	Asynchronous Request Identifier.
\<CustId>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Identification Aggregate.
\<RecCtrlOut>	Aggregate *see section 3.2.12.2.2*	Optional *but see Description*	Records Control Output Aggregate. Required if \<RecCtrlIn> was provided in the request and the server supports Records Control.
\<CreditAuthId>	Identifier	Optional Repeating Echoed	Credit Authorization Identifier.
\<CSPRefId>	Identifier	Optional Repeating Echoed	Customer Service Provider Reference Identifier. This field is used as a selection criterion.
\<SPRefId>	Identifier	Optional Repeating Echoed	Service Provider Reference Identifier. This field is used as a selection criterion.
\<DepAcctId>	Aggregate *see section 3.2.7.1.2*	Optional XOR Echoed	Deposit Account Identification. This field is used as a selection criterion.
\<CardAcctId>	Aggregate *see section 3.2.7.1.3*	Optional XOR Echoed	Card Account Identification. This field is used as a selection criterion.
\<ClientChgCode>	Open Enum	Optional Repeating Echoed	Client Change Code. This field is used as a selection criterion. This element will be removed from the CreditAuthInqRs message in IFX 2.0.
\<SelRangeDt>	Aggregate *see section 3.2.10.1*	Optional Echoed	Selection Range Date Aggregate.
\<SelRangeCurAmt>	Aggregate *see section 3.2.10.2*	Optional Echoed	Selection Range Amount Aggregate.
\<CreditAuthRec>	Aggregate *see section 6.3.28*	Optional Repeating	Credit Authorization Record Aggregate. These records are generated by the server and reflect the current state of the customer's Credit Authorizations. The records are filtered by the selection criteria specified in the request message.
\<Token>	Identifier	Optional *but see Description*	Token. Server Assigned. Client should make no assumptions about the value of this token relative to others that it may have received. The token is significant only to the server that originally assigned it. The token is only included if \<IncToken>=*True* in the request. \<Token>=0 is returned if no records are returned within the response.

6.10.2.5   Credit Authorization Audit

6.10.2.5.1 Request <CreditAuthAudRq>

Tag	Type	Usage	Description
<RqUID>	UUID	Required	Request Identifier.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header aggregate.
<AsyncRqUID>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously. For more information, see section 3.2.12.1.
<CustId>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<RecCtrlIn>	Aggregate *see section 3.2.12.2.1*	Optional	Records Control Input Aggregate.
<SelRangeDt>	Aggregate *see section 3.2.10.1*	Optional	Selection Range Date Aggregate.
<Method>	Closed Enum	Optional Repeating	Audit Selection Action. Used to identify actions associated with the object that is being audited. Valid values: Add, Mod, Can. This field is used as a selection criterion.
<CreditAuthId>	Identifier	Optional Repeating	Credit Authorization Identifier. Assigned by the server at the time the Credit Authorization is first added. Cannot be changed by the client. This field is used as a selection criterion.

6.10.2.5.2 Response <CreditAuthAudRs>

Tag	Type	Usage	Description
<Status>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, <StatusCode> defaults to 0 (zero).
<RqUID>	UUID	Required Echoed	Request Identifier.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header aggregate.
<MsgRsHdr>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header aggregate.
<AsyncRqUID>	UUID	Optional Echoed	Asynchronous Request Identifier.

Tag	Type	Usage	Description
\<CustId\>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Identification Aggregate.
\<RecCtrlOut\>	Aggregate *see section 3.2.12.2.2*	Optional *but see Description*	Message Records Control Output Aggregate. Required if \<RecCtrlIn\> was provided in the request and the server supports Records Control.
\<SelRangeDt\>	Aggregate *see section 3.2.10.1*	Optional Echoed	Selection Range Date Aggregate.
\<Method\>	Closed Enum	Optional Repeating Echoed	Audit Selection Action. This field is used as a selection criterion.
\<CreditAuthId\>	Identifier	Optional Repeating Echoed	Credit Authorization Identifier. Assigned by the server at the time the Credit Authorization is first added. Cannot be changed by the client. This field is used as a selection criterion.
\<CreditAuthMsgRec\>	Aggregate	Optional Repeating	Credit Authorization Message Record Aggregate.
\<CustId\>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate.
\<MsgRecDt\>	Date/Time	Optional	Message Record Creation Date. The date/time at which the message record was stored/created by the service provider.
\<CreditAuthAddRs\>	Aggregate *see section 6.10.2.1.2*	Required XOR	Credit Authorization Add Response Message Aggregate.
\<CreditAuthModRs\>	Aggregate *see section 6.10.2.2.2*	Required XOR	Credit Authorization Modify Response Message Aggregate.
\<CreditAuthRevRs\>	Aggregate *see section 6.10.2.7.2*	Required XOR	Credit Authorization Reversal Response Message Aggregate.
\<CreditAuthStatusModRs\>	Aggregate *see section 6.10.2.8.2*	Required XOR	Credit Authorization Status Modify Response Message Aggregate.
\<CreditAuthCanRs\>	Aggregate *see section 6.10.2.3.2*	Required XOR	Credit Authorization Cancel Response Message Aggregate.
\</CreditAuthMsgRec\>			

6.10.2.6   Credit Authorization Sync

6.10.2.6.1 Request \<CreditAuthSyncRq\>

Tag	Type	Usage	Description
\<RqUID\>	UUID	Required	Request Identifier.
\<MsgRqHdr\>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header aggregate.

Tag	Type	Usage	Description
<AsyncRqUID>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously.
			For more information, see section 3.2.12.1.
<CustId>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<RecCtrlIn>	Aggregate *see section 3.2.12.2.1*	Optional	Message Records Control Input Aggregate.
<Token>	Identifier	Required	Token. Server assigned. Client should make no assumptions about the value of this token relative to others that it may have received. The token is significant only to the server that originally assigned it.
			This is a token that has been previously sent by the server, or zero for first time requests.
<DepAcctId>	Aggregate *see section 3.2.7.1.2*	Required XOR	Deposit Account Identification Aggregate.
<CardAcctId>	Aggregate *see section 3.2.7.1.3*	Required XOR	Card Account Identification Aggregate.

6.10.2.6.2 Response <CreditAuthSyncRs>

Tag	Type	Usage	Description
<Status>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, <StatusCode> defaults to 0 (zero).
<RqUID>	UUID	Required Echoed	Request Identifier.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header aggregate.
<MsgRsHdr>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header aggregate.
<AsyncRqUID>	UUID	Optional Echoed	Asynchronous Request Identifier.
<CustId>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Identification Aggregate.
<RecCtrlOut>	Aggregate *see section 3.2.12.2.2*	Optional *but see Description*	Records Control Output Aggregate. Required if <RecCtrlIn> was provided in the request and the server supports Records Control.
<DepAcctId>	Aggregate *see section 3.2.7.1.2*	Required XOR	Deposit Account Identification Aggregate.

Tag	Type	Usage	Description
<CardAcctId>	Aggregate *see section 3.2.7.1.3*	Required XOR	Card Account Identification Aggregate.
<Token>	Identifier	Required Echoed	Token. Server assigned. Client should make no assumptions about the value of this token relative to others that it may have received. The token is significant only to the server that originally assigned it. This is a token that has been previously sent by the server, or zero for first time requests.
<NewToken>	Identifier	Required	New Token. Server assigned. Client should make no assumptions about the value of this token relative to others that it may have received. The token is significant only to the server that originally assigned it.
<CreditAuthMsgRec>	Aggregate	Optional Repeating	Credit Authorization Message Record Aggregate.
<CustId>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate.
<MsgRecDt>	Date/Time	Optional	Message Record Creation Date. The date/time at which the message record was stored/created by the service provider.
<CreditAuthAddRs>	Aggregate *see section 6.10.2.1.2*	Required XOR	Credit Authorization Add Response Message Aggregate.
<CreditAuthModRs>	Aggregate *see section 6.10.2.2.2*	Required XOR	Credit Authorization Modify Response Message Aggregate.
<CreditAuthStatusModRs>	Aggregate *see section 6.10.2.8.2*	Required XOR	Credit Authorization Status Modify Response Message Aggregate.
<CreditAuthCanRs>	Aggregate *see section 6.10.2.3.2*	Required XOR	Credit Authorization Cancel Response Message Aggregate.
</CreditAuthMsgRec>			

6.10.2.7   Credit Authorization Reversal

6.10.2.7.1 Request *<CreditAuthRevRq>*

Tag	Type	Usage	Description
<RqUID>	UUID	Required	Request Identifier.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header aggregate.
<AsyncRqUID>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously. For more information, see section **3.2.12.1**.

Tag	Type	Usage	Description
<CustId>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<RevReasonCode>	Open Enum	Required	Reversal reason code. The reason why the client is requesting a reversal.
<Desc>	C-80	Optional	Description, text describing the reason for the reversal.
<CreditAuthMsgRqInfo>	Aggregate	Required	Credit Authorization Message Request Information Aggregate. This aggregate contains the message to reverse. See RevOptSupt in the service profile for specific reversal messages supported.
<CreditAuthAddRq>	Aggregate *see section 6.10.2.1.1*	Required XOR	Credit Authorization Add Request Aggregate
<CreditAuthModRq>	Aggregate *see section 6.10.2.2.1*	Required XOR	Credit Authorization Modify Request Aggregate
<CreditAuthCanRq>	Aggregate *see section 6.10.2.3.1*	Required XOR	Credit Authorization Cancel Request Aggregate
<CreditAuthInqRq>	Aggregate *see section 6.10.2.4.1*	Required XOR	Credit Authorization Inquiry Request Message Aggregate.
<CreditAuthAudRq>	Aggregate *see section 6.10.2.5.1*	Required XOR	Credit Authorization Audit Request Aggregate
<CreditAuthSyncRq>	Aggregate *see section 6.10.2.6.1*	Required XOR	Credit Authorization Synchronization Request Message Aggregate.
</CreditAuthMsgRqInfo>			

6.10.2.7.2 Response <CreditAuthRevRs>

Tag	Type	Usage	Description
<Status>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, <StatusCode> defaults to 0 (zero).
<RqUID>	UUID	Required Echoed	Request Identifier.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header aggregate.
<MsgRsHdr>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header aggregate.
<AsyncRqUID>	UUID	Optional Echoed	Asynchronous Request Identifier.
<CustId>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Identification Aggregate.

Tag	Type	Usage	Description
<RevReasonCode>	Open Enum	Required Echoed	Reversal reason code. The reason why the client is requesting a reversal.
<Desc>	C-80	Optional Echoed	Description, text describing the reason for the reversal.
<CreditAuthMsgRqInfo>	Aggregate	Required Echoed	Credit Authorization Message Request Information Aggregate. This aggregate contains the message to reverse. See RevOptSupt in the service profile for specific reversal messages supported.
<CreditAuthAddRq>	Aggregate *see section 6.10.2.1.1*	Required Echoed XOR	Credit Authorization Add Request Aggregate
<CreditAuthModRq>	Aggregate *see section 6.10.2.2.1*	Required Echoed XOR	Credit Authorization Modify Request Aggregate
<CreditAuthCanRq>	Aggregate *see section 6.10.2.3.1*	Required Echoed XOR	Credit Authorization Cancel Request Aggregate
<CreditAuthInqRq>	Aggregate *see section 6.10.2.4.1*	Required Echoed XOR	Credit Authorization Inquiry Request Message Aggregate.
<CreditAuthAudRq>	Aggregate *see section 6.10.2.5.1*	Required Echoed XOR	Credit Authorization Audit Request Aggregate
<CreditAuthSyncRq>	Aggregate *see section 6.10.2.6.1*	Required Echoed XOR	Credit Authorization Synchronization Request Message Aggregate.
</CreditAuthMsgRqInfo>			
<CSPRefId>	Identifier	Optional	Customer Service Provider Reference Identifier.
<SPRefId>	Identifier	Optional	Service Provider Reference Identifier.

6.10.2.8   Credit Authorization Status Modify

6.10.2.8.1 Request <CreditAuthStatusModRq>

Tag	Type	Usage	Description
<RqUID>	UUID	Required	Request Identifier.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header Aggregate.
<AsyncRqUID>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously. For more information, see section **3.2.12.1**.
<CustId>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<CreditAuthId>	Identifier	Required	Credit Authorization Identifier.

Tag	Type	Usage	Description
<CreditAuthStatus>	Aggregate *see section 6.3.28.2*	Required	Credit Authorization Status Aggregate.

6.10.2.8.2 Response <CreditAuthStatusModRs>

Tag	Type	Usage	Description
<Status>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, <StatusCode> defaults to 0 (zero).
<RqUID>	UUID	Required Echoed	Request Identifier.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header Aggregate.
<MsgRsHdr>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header Aggregate.
<AsyncRqUID>	UUID	Optional Echoed	Asynchronous Request Identifier.
<CustId>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<CreditAuthId>	Identifier	Required Echoed	Credit Authorization Identifier.
<CreditAuthStatus>	Aggregate *see section 6.3.28.2*	Required Echoed	Credit Authorization Status Aggregate.
<CSPRefId>	Identifier	Optional	Customer Service Reference Identifier.
<SPRefId>	Identifier	Optional	Service Provider Reference Identifier.

6.10.2.9   Credit Add

6.10.2.9.1 Request <CreditAddRq>

The Credit Add message is used to add a credit for immediate action or to confirm a previously added credit authorization.

Tag	Type	Usage	Description
<RqUID>	UUID	Required	Request Identifier.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header aggregate.
<AsyncRqUID>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously. For more information, see section 3.2.12.1.

Tag	Type	Usage	Description
<CustId>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<CreditInfo>	Aggregate *see section 6.3.29*	Required	Credit Information Aggregate

6.10.2.9.2 Response <CreditAddRs>

Tag	Type	Usage	Description
<Status>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, <StatusCode> defaults to 0 (zero).
<RqUID>	UUID	Required Echoed	Request Identifier.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header aggregate.
<MsgRsHdr>	Aggregate *see section 3.2.1.2*	Optional	Message Respone Header aggregate.
<AsyncRqUID>	UUID	Optional Echoed	Asynchronous Request Identifier.
<CustId>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Identification Aggregate.
<CreditInfo>	Aggregate *see section 6.3.29*	Required Echoed	Credit Information Aggregate
<CreditRec>	Aggregate *see section 6.3.2*	Optional *but see Description*	Credit Record Aggregate. This aggregate will be required in IFX 2.0.
<CSPRefId>	Identifier	Optional	Customer Service Provider Reference Identifier.
<SPRefId>	Identifier	Optional	Service Provider Reference Identifier.

6.10.2.10 Credit Modify

The Credit Modify message is used to modify the Information previously supplied in a Credit Add request.

6.10.2.10.1 Request <CreditModRq>

Tag	Type	Usage	Description
<RqUID>	UUID	Required	Request Identifier.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header Aggregate.

Tag	Type	Usage	Description
\<AsyncRqUID\>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously.
			For more information, see section **3.2.12.1**.
\<CustId\>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in \<SignonRq\>. For example, if a CSR or SP issues the request on behalf of the user, then \<CustId\> is required, and must contain the value of the user whose request is being issued.
\<CreditId\>	Identifier	Required	Credit Identifier. Assigned by the server at the time the Credit is first added. Cannot be changed by the client.
\<CreditInfo\>	Aggregate *see section 6.3.29*	Required	Credit Information Aggregate

6.10.2.10.2 Response <CreditModRs>

Tag	Type	Usage	Description
\<Status\>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, \<StatusCode\> defaults to 0 (zero).
\<RqUID\>	UUID	Required Echoed	Request Identifier.
\<MsgRqHdr\>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header Aggregate.
\<MsgRsHdr\>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header Aggregate.
\<AsyncRqUID\>	UUID	Optional Echoed	Asynchronous Request Identifier.
\<CustId\>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Identification Aggregate.
\<CreditId\>	Identifier	Required Echoed	Credit Identifier
\<CreditInfo\>	Aggregate *see section 6.3.29*	Required Echoed	Credit Information Aggregate
\<CreditRec\>	Aggregate *see section 6.3.2*	Required	Credit Record Aggregate
\<CSPRefId\>	Identifier	Optional	Customer Service Provider Reference Identifier.
\<SPRefId\>	Identifier	Optional	Service Provider Reference Identifier.

6.10.2.11 Credit Cancel

The client uses this message to cancel a previously added credit.

6.10.2.11.1 Request <CreditCanRq>

Tag	Type	Usage	Description
\<RqUID\>	UUID	Required	Request Identifier.
\<MsgRqHdr\>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header Aggregate.
\<AsyncRqUID\>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously.
			For more information, see section **3.2.12.1**.
\<CustId\>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in \<SignonRq\>. For example, if a CSR or SP issues the request on behalf of the user, then \<CustId\> is required, and must contain the value of the user whose request is being issued.
\<CreditId\>	Identifier	Required	Credit Identifier. Assigned by the server at the time the Credit is first added. Cannot be changed by the client.
\<ClientChgCode\>	Open Enum	Required	Client Change Code. This indicates the reason why a credit was cancelled or modified by the client.
			Defined Values:
			None, ConsumerCancelled, TerminalExceptionAmountKnown, TerminalExceptionAmountUnknown, ConsumerExceptionAmountKnown, ConsumerExceptionAmountUnknown

6.10.2.11.2 Response \<CreditCanRs\>

Tag	Type	Usage	Description
\<Status\>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, \<StatusCode\> defaults to 0 (zero).
\<RqUID\>	UUID	Required Echoed	Request Identifier.
\<MsgRqHdr\>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header Aggregate.
\<MsgRsHdr\>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header Aggregate.
\<AsyncRqUID\>	UUID	Optional Echoed	Asynchronous Request Identifier.
\<CustId\>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Identification Aggregate.
\<CreditId\>	Identifier	Required Echoed	Credit Identifier
\<ClientChgCode\>	Open Enum	Required Echoed	Client Change Code.

Tag	Type	Usage	Description
<CreditRec>	Aggregate *see section 6.3.2*	Optional	Credit Record Aggregate
<CSPRefId>	Identifier	Optional	Customer Service Provider Reference Identifier.
<SPRefId>	Identifier	Optional	Service Provider Reference Identifier.

6.10.2.12 Credit Inquiry

6.10.2.12.1   Request <CreditInqRq>

Tag	Type	Usage	Description
<RqUID>	UUID	Required	Request Identifier.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header Aggregate.
<AsyncRqUID>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously. For more information, see section **3.2.12.1**.
<CustId>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<RecCtrlIn>	Aggregate *see section 3.2.12.2.1*	Optional	Records Control Input Aggregate.
<CreditId>	Identifier	Optional Repeating	Credit Identifier.
<CSPRefId>	Identifier	Optional Repeating	Customer Service Provider Reference Identifier. This field is used as a selection criterion.
<SPRefId>	Identifier	Optional Repeating	Service Provider Reference Identifier. This field is used as a selection criterion.
<DepAcctId>	Aggregate *see section 3.2.7.1.2*	Optional XOR	Deposit Account Identification. This field is used as a selection criterion.
<CardAcctId>	Aggregate *see section 3.2.7.1.3*	Optional XOR	Card Account Identification. This field is used as a selection criterion.
<ClientChgCode>	Open Enum	Optional Repeating	Client Change Code. This field is used as a selection criterion.
<SelRangeDt>	Aggregate *see section 3.2.10.1*	Optional	Selection Range Date Aggregate.
<SelRangeCurAmt>	Aggregate *see section 3.2.10.2*	Optional	Selection Range Amount Aggregate.

Tag	Type	Usage	Description
<IncToken>	Boolean	Optional	Include Token. If *True*, a <Token> should be included in the response to set a base for future Sync messages. If *False* or omitted, no <Token> is returned.

6.10.2.12.2 Response <CreditInqRs>

Tag	Type	Usage	Description
<Status>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, <StatusCode> defaults to 0 (zero).
<RqUID>	UUID	Required Echoed	Request Identifier.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header Aggregate.
<MsgRsHdr>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header Aggregate.
<AsyncRqUID>	UUID	Optional Echoed	Asynchronous Request Identifier.
<CustId>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Identification Aggregate.
<RecCtrlOut>	Aggregate *see section 3.2.12.2.2*	Optional *but see Description*	Records Control Output Aggregate. Required if <RecCtrlIn> was provided in the request and the server supports Records Control.
<CreditId>	Identifier	Optional Repeating Echoed	Credit Identifier.
<CSPRefId>	Identifier	Optional Repeating Echoed	Customer Service Provider Reference Identifier.
<SPRefId>	Identifier	Optional Repeating Echoed	Service Provider Reference Identifier.
<DepAcctId>	Aggregate *see section 3.2.7.1.2*	Optional XOR Echoed	Deposit Account Identification.
<CardAcctId>	Aggregate *see section 3.2.7.1.3*	Optional XOR Echoed	Card Account Identification.
<ClientChgCode>	Open Enum	Optional Repeating Echoed	Client Change Code. This field is used as a selection criterion.
<SelRangeDt>	Aggregate *see section 3.2.10.1*	Optional Echoed	Selection Range Date Aggregate.
<SelRangeCurAmt>	Aggregate *see section 3.2.10.2*	Optional Echoed	Selection Range Amount Aggregate.

Tag	Type	Usage	Description
<CreditRec>	Aggregate *see section 6.3.2*	Optional Repeating	Credit Record Aggregate. These records are generated by the server and reflect the current state of the customer's Credits. The records are filtered by the selection criteria specified in the request message.
<Token>	Identifier	Optional *but see Description*	Token. Server Assigned. Client should make no assumptions about the value of this token relative to others that it may have received. The token is significant only to the server that originally assigned it.
			The token is only included if <IncToken>=*True* in the request. <Token>=0 is returned if no records are returned within the response.

6.10.2.13 Credit Audit

6.10.2.13.1 Request <CreditAudRq>

Tag	Type	Usage	Description
<RqUID>	UUID	Required	Request Identifier.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header Aggregate.
<AsyncRqUID>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously.
			For more information, see section **3.2.12.1**.
<CustId>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<RecCtrlIn>	Aggregate *see section 3.2.12.2.1*	Optional	Records Control Input Aggregate.
<SelRangeDt>	Aggregate *see section 3.2.10.1*	Optional	Selection Range Date Aggregate.
<Method>	Closed Enum	Optional Repeating	Audit Selection Action. Used to identify actions associated with the object that is being audited.
			Valid values: Add, Mod, Can.
			This field is used as a selection criterion.
<CreditId>	Identifier	Optional Repeating	Credit Identifier. Assigned by the server at the time the Credit is first added. Cannot be changed by the client.
			This field is used as a selection criterion.

6.10.2.13.2 Response <CreditAudRs>

Tag	Type	Usage	Description

Tag	Type	Usage	Description
<Status>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, <StatusCode> defaults to 0 (zero).
<RqUID>	UUID	Required Echoed	Request Identifier.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header Aggregate.
<MsgRsHdr>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header Aggregate.
<AsyncRqUID>	UUID	Optional Echoed	Asynchronous Request Identifier.
<CustId>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Identification Aggregate.
<RecCtrlOut>	Aggregate *see section 3.2.12.2.2*	Optional *but see Description*	Message Records Control Output Aggregate. Required if <RecCtrlIn> was provided in the request and the server supports Records Control.
<SelRangeDt>	Aggregate *see section 3.2.10.1*	Optional Echoed	Selection Range Date Aggregate.
<Method>	Closed Enum	Optional Repeating Echoed	Audit Selection Action. This field is used as a selection criterion.
<CreditId>	Identifier	Optional Repeating Echoed	Credit Identifier. Assigned by the server at the time the Credit is first added. Cannot be changed by the client. This field is used as a selection criterion.
<CreditMsgRec>	Aggregate	Optional Repeating	Credit Message Record Aggregate.
<CustId>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate.
<MsgRecDt>	Date/Time	Optional	Message Record Creation Date. The date/time at which the message record was stored/created by the service provider.
<CreditAddRs>	Aggregate *see section 6.10.2.9.2*	Required XOR	Credit Add Response Message Aggregate.
<CreditModRs>	Aggregate *see section 6.10.2.10.2*	Required XOR	Credit Modify Response Message Aggregate.
<CreditRevRs>	Aggregate *see section 6.10.2.16.2*	Required XOR	Credit Reversal Response Message Aggregate.
<CreditStatusModRs>	Aggregate *see section 6.10.2.15.2*	Required XOR	Credit Status Modify Response Message Aggregate.
<CreditCanRs>	Aggregate *see section 6.10.2.11.2*	Required XOR	Credit Cancel Response Message Aggregate.

Tag	Type	Usage	Description
</CreditMsgRec>			

6.10.2.14 Credit Sync

6.10.2.14.1 Request <CreditSyncRq>

Tag	Type	Usage	Description
<RqUID>	UUID	Required	Request Identifier.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header Aggregate.
<AsyncRqUID>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously. For more information, see section **3.2.12.1**.
<CustId>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<RecCtrlIn>	Aggregate *see section 3.2.12.2.1*	Optional	Message Records Control Input Aggregate.
<Token>	Identifier	Required	Token. Server assigned. Client should make no assumptions about the value of this token relative to others that it may have received. The token is significant only to the server that originally assigned it. This is a token that has been previously sent by the server, or zero for first time requests.
<DepAcctId>	Aggregate *see section 3.2.7.1.2*	Required XOR	Deposit Account Identification Aggregate.
<CardAcctId>	Aggregate *see section 3.2.7.1.3*	Required XOR	Card Account Identification Aggregate.

6.10.2.14.2 Response <CreditSyncRs>

Tag	Type	Usage	Description
<Status>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, <StatusCode> defaults to 0 (zero).
<RqUID>	UUID	Required Echoed	Request Identifier.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header Aggregate.
<MsgRsHdr>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header Aggregate.

Tag	Type	Usage	Description
<AsyncRqUID>	UUID	Optional Echoed	Asynchronous Request Identifier.
<CustId>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Identification Aggregate.
<RecCtrlOut>	Aggregate *see section 3.2.12.2.2*	Optional *but see Description*	Records Control Output Aggregate. Required if <RecCtrlIn> was provided in the request and the server supports Records Control.
<DepAcctId>	Aggregate *see section 3.2.7.1.2*	Required XOR	Deposit Account Identification Aggregate.
<CardAcctId>	Aggregate *see section 3.2.7.1.3*	Required XOR	Card Account Identification Aggregate.
<Token>	Identifier	Required Echoed	Token. Server assigned. Client should make no assumptions about the value of this token relative to others that it may have received. The token is significant only to the server that originally assigned it. This is a token that has been previously sent by the server, or zero for first time requests.
<NewToken>	Identifier	Required	New Token. Server assigned. Client should make no assumptions about the value of this token relative to others that it may have received. The token is significant only to the server that originally assigned it.
<CreditMsgRec>	Aggregate	Optional Repeating	Credit Message Record Aggregate.
<CustId>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate.
<MsgRecDt>	Date/Time	Optional	Message Record Creation Date. The date/time at which the message record was stored/created by the service provider.
<CreditAddRs>	Aggregate *see section 6.10.2.9.2*	Required XOR	Credit Add Response Message Aggregate.
<CreditModRs>	Aggregate *see section 6.10.2.10.2*	Required XOR	Credit Modify Response Message Aggregate.
<CreditStatusModRs>	Aggregate *see section 6.10.2.15.2*	Required XOR	Credit Status Modify Response Message Aggregate.
<CreditCanRs>	Aggregate *see section 6.10.2.11.2*	Required XOR	Credit Cancel Response Message Aggregate.
</CreditMsgRec>			

6.10.2.15 Credit Status Modify

6.10.2.15.1 Request <CreditStatusModRq>

Tag	Type	Usage	Description
<RqUID>	UUID	Required	Request Identifier.

Tag	Type	Usage	Description
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header Aggregate.
<AsyncRqUID>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously. For more information, see section **3.2.12.1**.
<CustId>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<CreditId>	Identifier	Required	Credit Identifier.
<CreditStatus>	Aggregate *see section 6.3.3*	Required	Credit Status Aggregate.

6.10.2.15.2 Response <CreditStatusModRs>

Tag	Type	Usage	Description
<Status>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, <StatusCode> defaults to 0 (zero).
<RqUID>	UUID	Required Echoed	Request Identifier.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header Aggregate.
<MsgRsHdr>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header Aggregate.
<AsyncRqUID>	UUID	Optional Echoed	Asynchronous Request Identifier.
<CustId>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<CreditId>	Identifier	Required Echoed	Credit Identifier.
<CreditStatus>	Aggregate *see section 6.3.3*	Required Echoed	Credit Status Aggregate.
<CSPRefId>	Identifier	Optional	Customer Service Reference Identifier.
<SPRefId>	Identifier	Optional	Service Provider Reference Identifier.

6.10.2.16 Credit Reversal

6.10.2.16.1 Request <CreditRevRq>

Tag	Type	Usage	Description
<RqUID>	UUID	Required	Request Identifier. Sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header Aggregate.
<AsyncRqUID>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously.
			For more information, see section **3.2.12.1**.
<CustId>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<RevReasonCode>	Open Enum	Required	Reversal reason code. The reason why the client is requesting a reversal.
<Desc>	C-80	Optional	Description, text describing the reason for the reversal.
<CreditMsgRqInfo>	Aggregate	Required	Credit Message Request Information Aggregate. This aggregate contains the message to reverse. See RevOptSupt in the service profile for specific reversal messages supported.
<CreditAddRq>	Aggregate *see section 6.10.2.9.1*	Required XOR	Credit Add Request Message Aggregate.
<CreditModRq>	Aggregate *see section 6.10.2.10.1*	Required XOR	Credit Modify Request Message Aggregate.
<CreditStatusModRq>	Aggregate *see section 6.10.2.15.1*	Required XOR	Credit Status Modify Request Message Aggregate.
<CreditCanRq>	Aggregate *see section 6.10.2.11.1*	Required XOR	Credit Cancel Request Message Aggregate.
<CreditInqRq>	Aggregate *see section 6.10.2.12.1*	Required XOR	Credit Inquiry Request Message Aggregate.
<CreditAudRq>	Aggregate *see section 6.10.2.13.1*	Required XOR	Credit Audit Request Message Aggregate.
<CreditSyncRq>	Aggregate *see section 6.10.2.14.1*	Required XOR	Credit Sync Request Message Aggregate.
</CreditMsgRqInfo>			

6.10.2.16.2 Response <CreditRevRs>

Tag	Type	Usage	Description
\<Status\>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, \<StatusCode\> defaults to 0 (zero).
\<RqUID\>	UUID	Required Echoed	Request Identifier.
\<MsgRqHdr\>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header Aggregate.
\<MsgRsHdr\>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header Aggregate.
\<AsyncRqUID\>	UUID	Optional Echoed	Asynchronous Request Identifier.
\<CustId\>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Identification Aggregate.
\<RevReasonCode\>	Open Enum	Required Echoed	Reversal reason code.
\<Desc\>	C-80	Optional Echoed	Description, text describing the reason for the reversal.
\<CreditMsgRqInfo\>	Aggregate	Required Echoed	Credit Message Request Information Aggregate. This aggregate contains the message to reverse.
\<CreditAddRq\>	Aggregate *see section 6.10.2.9.1*	Required XOR Echoed	Credit Add Request Message Aggregate.
\<CreditModRq\>	Aggregate *see section 6.10.2.10.1*	Required XOR Echoed	Credit Modify Request Message Aggregate.
\<CreditStatusModRq\>	Aggregate *see section 6.10.2.15.1*	Required XOR Echoed	Credit Status Modify Request Message Aggregate.
\<CreditCanRq\>	Aggregate *see section 6.10.2.11.1*	Required XOR Echoed	Credit Cancel Request Message Aggregate.
\<CreditInqRq\>	Aggregate *see section 6.10.2.12.1*	Required XOR Echoed	Credit Inquiry Request Message Aggregate.
\<CreditAudRq\>	Aggregate *see section 6.10.2.13.1*	Required XOR Echoed	Credit Audit Request Message Aggregate.
\<CreditSyncRq\>	Aggregate *see section 6.10.2.14.1*	Required XOR Echoed	Credit Sync Request Message Aggregate.
\</CreditMsgRqInfo\>			
\<CSPRefId\>	Identifier	Optional	Customer Service Provider Reference Identifier.
\<SPRefId\>	Identifier	Optional	Service Provider Reference Identifier.

6.11   Deposit Application

This transaction set allows a customer to submit a deposit account application to a financial institution. The decision to create an account on the financial institutions system of permanent record is left to the financial institution. The financial institution may or may not post the account before the account is fundend. The message set does not force a financial institution to create a customer record for the applicant. Typically upon receipt of the application, a financial instituion suspends the application until the funds for the account arrive. The message set replaces the paper form typically filled in by the customer at a branch.

6.11.1   Deposit Application Add

Allows a client to apply for a deposit account. The client sends a <DepAppAddRq> message to request an account be opened in the applicant's name. The financial institution replies with a <DepAppAddRs> signifying that the application was received and is scheduled for processing.

6.11.1.1   Request <DepAppAddRq>

Tag	Type	Usage	Description
<RqUID>	UUID	Required	Request Identifier. Sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header aggregate.
<AsyncRqUID>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously. For more information, see section 3.2.12.1.
<CustId>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<DepAppInfo>	Aggregate *see section 6.3.30*	Required	Deposit Application Information Aggregate.

6.11.1.2   Response <DepAppAddRs>

Tag	Type	Usage	Description
<Status>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, <StatusCode> defaults to 0 (zero).
<RqUID>	UUID	Required Echoed	Request Identifier. Sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header aggregate.
<MsgRsHdr>	Aggregate *see section 3.2.1.2*	Optional	Message Respone Header aggregate.

Tag	Type	Usage	Description
<AsyncRqUID>	UUID	Optional Echoed	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously. For more information, see section 3.2.12.1.
<CustId>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<DepAppInfo>	Aggregate *see section 6.3.30*	Required Echoed	Deposit Application Information Aggregate.
<DepAppRec>	Aggregate *see section 6.3.31*	Required	Deposit Application Record Aggregate.
<CSPRefId>	Identifier	Optional	Customer Service Provider Reference Identifier.
<SPRefId>	Identifier	Optional	Service Provider Reference Identifier.

6.11.2   Deposit Application Modification

Allows a client to modify a previously added deposit application.

6.11.2.1   Request <DepAppModRq>

Tag	Type	Usage	Description
<RqUID>	UUID	Required	Request Identifier. Sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header aggregate.
<AsyncRqUID>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously. For more information, see section 3.2.12.1.
<CustId>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<DepAppId>	Identifier	Required	Deposit Application Identifer. Assigned by the server at the time the application is first added.
<DepAppInfo>	Aggregate *see section 6.3.30*	Required	Deposit Application Information Aggregate.

6.11.2.2 Response <DepAppModRs>

Tag	Type	Usage	Description
<Status>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, <StatusCode> defaults to 0 (zero).
<RqUID>	UUID	Required Echoed	Request Identifier. Sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header aggregate.
<MsgRsHdr>	Aggregate *see section 3.2.1.2*	Optional	Message Respones Header aggregate.
<AsyncRqUID>	UUID	Optional Echoed	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously. For more information, see section 3.2.12.1.
<CustId>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<DepAppId>	Identifier	Required Echoed	Deposit Application Identifier.
<DepAppInfo>	Aggregate *see section 6.3.30*	Required Echoed	Deposit Application Information Aggregate.
<DepAppRec>	Aggregate *see section 6.3.31*	Required	Deposit Application Record Aggregate.
<CSPRefId>	Identifier	Optional	Customer Service Provider Reference Identifier.
<SPRefId>	Identifier	Optional	Service Provider Reference Identifier.

6.11.3 *Deposit Application Cancel*

Allows a client to cancel a previously added deposit account application.

6.11.3.1 Request <DepAppCanRq>

Tag	Type	Usage	Description
<RqUID>	UUID	Required	Request Identifier. Sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header aggregate.
<AsyncRqUID>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously. For more information, see section 3.2.12.1.

Tag	Type	Usage	Description
\<CustId\>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in \<SignonRq\>. For example, if a CSR or SP issues the request on behalf of the user, then \<CustId\> is required, and must contain the value of the user whose request is being issued.
\<DepAppId\>	Identifer	Required	Deposit Application Identifer. Assigned by the server at the time the application is first added.

6.11.3.2   Response \<DepAppCanRs\>

Tag	Type	Usage	Description
\<Status\>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, \<StatusCode\> defaults to 0 (zero).
\<RqUID\>	UUID	Required Echoed	Request Identifier. Sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.
\<MsgRqHdr\>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header aggregate.
\<MsgRsHdr\>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header aggregate.
\<AsyncRqUID\>	UUID	Optional Echoed	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously. For more information, see section 3.2.12.1.
\<CustId\>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in \<SignonRq\>. For example, if a CSR or SP issues the request on behalf of the user, then \<CustId\> is required, and must contain the value of the user whose request is being issued.
\<DepAppId\>	Identifier	Required Echoed	Deposit Application Identifer. Assigned by the server at the time the application is first added.
\<DepAppRec\>	Aggregate *see section 6.3.31*	Optional	Deposit Application Record Aggregate.
\<CSPRefId\>	Identifer	Optional	Customer Service Provider Reference Identifier.
\<SPRefId\>	Identifer	Optional	Service Provider Reference Identifier.

6.11.4   *Deposit Application Inquiry*

Allows a client to inquiry on a previously added deposit account application.

6.11.4.1   Request \<DepAppInqRq\>

Tag	Type	Usage	Description
\<RqUID\>	UUID	Required	Request Identifier. Sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.

Tag	Type	Usage	Description
\<MsgRqHdr\>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header aggregate.
\<AsyncRqUID\>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously.
			For more information, see section 3.2.12.1.
\<CustId\>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in \<SignonRq\>. For example, if a CSR or SP issues the request on behalf of the user, then \<CustId\> is required, and must contain the value of the user whose request is being issued.
\<DepAppId\>	Identifer	Required Repeating	Deposit Application Identifer. Assigned by the server at the time the application is first added.
\<IncToken\>	Boolean	Optional	Include Token. If *True*, a \<Token\> should be included in the response to set a base for future Sync messages. If *False* or omitted, no \<Token\> is returned.

6.11.4.2   Response \<DepAppInqRs\>

Tag	Type	Usage	Description
\<Status\>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, \<StatusCode\> defaults to 0 (zero).
\<RqUID\>	UUID	Required Echoed	Request Identifier. Sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.
\<MsgRqHdr\>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header aggregate.
\<MsgRsHdr\>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header aggregate.
\<AsyncRqUID\>	UUID	Optional Echoed	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously.
			For more information, see section 3.2.12.1.
\<CustId\>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in \<SignonRq\>. For example, if a CSR or SP issues the request on behalf of the user, then \<CustId\> is required, and must contain the value of the user whose request is being issued.
\<DepAppId\>	Identifier	Required Repeating Echoed	Deposit Application Identifer. Assigned by the server at the time the application is first added.
\<DepAppRec\>	Aggregate *see section 6.3.31*	Optional Repeating	Deposit Application Record Aggregate.

Tag	Type	Usage	Description
<CSPRefId>	Identifer	Optional	Customer Service Provider Reference Identifier.
<SPRefId>	Identifier	Optional	Service Provider Reference Identifier.
<Token>	Identifier	Optional *but see Description*	Token. Server Assigned. Client should make no assumptions about the value of this token relative to others that it may have received. The token is significant only to the server that originally assigned it. The token is only included if <IncToken>=*True* in the request. <Token>=0 is returned if no records are returned within the response.

6.11.5   Deposit Application Audit

Allows a client to audit a previously added deposit account application.

6.11.5.1   Request <DepAppAudRq>

Tag	Type	Usage	Description
<RqUID>	UUID	Required	Request Identifier. Sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header aggregate.
<AsyncRqUID>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously. For more information, see section 3.2.12.1.
<CustId>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<RecCtrlIn>	Aggregate *see section 3.2.12.2.1*	Optional	Records Control Input Aggregate.
<SelRangeDt>	Aggregate *see section 3.2.10.1*	Optional	Selection Range Date Aggregate.
<Method>	Closed Enum	Optional Repeating	Audit Selection Action. Used to identify actions associated with the object that is being audited. Valid values: Aud, Mod, Can. This field is used as a selection criterion.
<DepAppId>	Identifer	Optional Repeating	Deposit Application Identifer. Assigned by the server at the time the application is first added. This field is used as a selection criterion.

6.11.5.2   Response <DepAppAudRs>

Tag	Type	Usage	Description

Tag	Type	Usage	Description
<Status>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, <StatusCode> defaults to 0 (zero).
<RqUID>	UUID	Required Echoed	Request Identifier. Sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header aggregate.
<MsgRsHdr>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header aggregate.
<AsyncRqUID>	UUID	Optional Echoed	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously.
			For more information, see section 3.2.12.1.
<CustId>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<RecCtrlOut>	Aggregate *see section 3.2.12.2.2*	Optional *but see Description*	Message Records Control Output Aggregate.
			Required if <RecCtrlIn> was provided in the request and the server supports Records Control.
<SelRangeDt>	Aggregate *see section 3.2.10.1*	Optional Echoed	Selection Range Date Aggregate.
<Method>	Closed Enum	Optional Repeating Echoed	Audit Selection Action. Used to identify actions associated with the object that is being audited.
			Valid values: Aud, Mod, Can.
			This field is used as a selection criterion.
<DepAppId>	Identifier	Required Echoed	Deposit Application Identifer. Assigned by the server at the time the application is first added.
<DepAppMsgRec>	Aggregate	Optional Repeating	Deposit Application Message Record Aggregagte.
<CustId>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate.
<MsgRecDt>	DateTime	Optional	Message Record Creation Date. The date/time at which the message record was stored/created by the service provider.
<DepAppAddRs>	Aggregagte	Required XOR	Deposit Application Add Response Message Aggregate.
<DepAppModRs>	Aggregate *see section 6.11.2.2*	Required XOR	Deposit Application Modify Response Message Aggregate.
<DepAppCanRs>	Aggregate *see section 6.11.3.2*	Required XOR	Deposit Application Cancel Response Message Aggregate.
</DepAppMsgRec>			

6.11.6   Deposit Application Sync

Allows a client to synchronize with a previously added deposit account application.

6.11.6.1   Request <DepAppSyncRq>

Tag	Type	Usage	Description
<RqUID>	UUID	Required	Request Identifier. Sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header aggregate.
<AsyncRqUID>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously. For more information, see section 3.2.12.1.
<CustId>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<RecCtrlIn>	Aggregate *see section 3.2.12.2.1*	Optional	Records Control Input Aggregate.
<Token>	Identifier	Required	Token. Server assigned. Client should make no assumptions about the value of this token relative to others that it may have received. The token is significant only to the server that originally assigned it. This is a token that has been previously sent by the server, or zero for first time requests.

6.11.6.2   Response <DepAppSyncRs>

Tag	Type	Usage	Description
<Status>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, <StatusCode> defaults to 0 (zero).
<RqUID>	UUID	Required Echoed	Request Identifier. Sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header aggregate.
<MsgRsHdr>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header aggregate.
<AsyncRqUID>	UUID	Optional Echoed	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously. For more information, see section 3.2.12.1.

Tag	Type	Usage	Description
<CustId>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<RecCtrlOut>	Aggregate *see section 3.2.12.2.2*	Optional *but see Description*	Message Records Control Output Aggregate. Required if <RecCtrlIn> was provided in the request and the server supports Records Control.
<NewToken>	Identifier	Required	New Token. Server assigned. Client should make no assumptions about the value of this token relative to others that it may have received. The token is significant only to the server that originally assigned it. If the client sent a token in the request, the server returns a new token based on this audit message.
<DepAppMsgRec>	Aggregate	Optional Repeating	Deposit Application Message Record Aggregagte.
<CustId>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate.
<MsgRecDt>	DateTime	Optional	Message Record Creation Date. The date/time at which the message record was stored/created by the service provider.
<DepAppAddRs>	Aggregagte	Required XOR	Deposit Application Add Response Message Aggregate.
<DepAppModRs>	Aggregate *see section 6.11.2.2*	Required XOR	Deposit Application Modify Response Message Aggregate.
<DepAppCanRs>	Aggregate *see section 6.11.3.2*	Required XOR	Deposit Application Cancel Response Message Aggregate.
</DepAppMsgRec>			

6.12   Bank Account Statement Image Inquiry

A client may use the <BankAcctStmtImgInqRq> message to retrieve an image of a bank account statement for a single statement period or multiple statement periods. This message should be used when a client must receive pre-rendered statement images and can be useful for supporting legacy statement printing host systems. A client may optionally provide a date range to limit the number of bank account statement image records that are returned in the response. Note that the statement is based on the normal cutoff cycles and the date range must return statements that *ended* (cutoff or cycled) during the specified range. The client may specify a date range that results in no statement, because there were no statement end cutoffs within the range.

> *Note:*   *The client may specify a delivery method picked from those supported in the Service Profile.*

6.12.1.1   Request <BankAcctStmtImgInqRq>

Tag	Type	Usage	Description
<RqUID>	UUID	Required	Request Identifier.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header Aggregate.

Tag	Type	Usage	Description
\<AsyncRqUID\>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously.
			For more information, see section **3.2.12.1**.
\<CustId\>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in \<SignonRq\>. For example, if a CSR or SP issues the request on behalf of the user, then \<CustId\> is required, and must contain the value of the user whose request is being issued.
\<RecCtrlIn\>	Aggregate *see section 3.2.12.2.1*	Optional	Records Control Input Aggregate.
\<DepAcctId\>	Aggregate *see section 3.2.7.1.2*	Required XOR	Deposit Account Identification Aggregate.
\<CardAcctId\>	Aggregate *see section 3.2.7.1.3*	Required XOR	Card Account Identification Aggregate.
\<LoanAcctId\>	Aggregate *see section 3.2.7.1.4*	Required XOR	Loan Account Identification Aggregate.
\<SelRangeDt\>	Aggregate *see section 3.2.10.1*	Optional	Selection Range Date Aggregate.
\<IncDetail\>	Boolean	Optional	Include Detail Indicator. If *True*, the response should include the detail statement information. If *False* or omitted, the detail information should not be included.
\<StmtType\>	Open Enum	Optional	Statement type
			Defined values: Partial, Legal
\<DeliveryMethod\>	Open Enum	Optional Profiled values	Delivery Method. Default is Channel.
			Value must be supported in Service Profile.
\<DeliveryMedia\>	Open Enum	Optional	Delivery Media
			Defined values: CD, Diskette, DVD, Paper. Default value is Paper.
			Note: *\<DeliveryMedia\> is only valid if \<DeliveryMethod\> is a manual process. (i.e. Post, UPS)*
\<ContactInfo\>	Aggregate *see section 3.2.3.7*	Optional	Contact Information Aggregate.

6.12.1.2 Response \<BankAcctStmtImgInqRs\>

Tag	Type	Usage	Description
\<Status\>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, \<StatusCode\> defaults to 0 (zero).
\<RqUID\>	UUID	Required Echoed	Request Identifier.

Tag	Type	Usage	Description
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header Aggregate.
<MsgRsHdr>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header Aggregate.
<AsyncRqUID>	UUID	Optional Echoed	Asynchronous Request Identifier.
<CustId>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<RecCtrlOut>	Aggregate *see section 3.2.12.2.2*	Optional *but see Description*	Records Control Output Aggregate. Required if <RecCtrlIn> was provided in the request and the server supports Records Control.
<DepAcctId>	Aggregate *see section 3.2.7.1.2*	Optional XOR Echoed	Deposit Account Identification Aggregate.
<CardAcctId>	Aggregate *see section 3.2.7.1.3*	Optional XOR Echoed	Card Account Identification Aggregate.
<LoanAcctId>	Aggregate *see section 3.2.7.1.4*	Optional XOR Echoed	Loan Account Identification Aggregate.
<SelRangeDt>	Aggregate *see section 3.2.10.1*	Optional Echoed	Selection Range Date Aggregate.
<IncDetail>	Boolean	Optional Echoed	Include Detail Indicator.
<StmtType>	Open Enum	Optional Echoed	Statement type
<DeliveryMethod>	Open Enum	Optional Echoed	Delivery Method.
<DeliveryMedia>	Open Enum	Optional Echoed	Delivery Media
<ContactInfo>	Aggregate *see section 3.2.3.7*	Optional Echoed	Contact Information Aggregate.
<BankAcctStmtImgInqRec>	Aggregate	Optional Repeating	Bank Account Statement Image Record Aggregate.
<StartDt>	DateTime	Required	Selection Start Date for this statement.
<EndDt>	DateTime	Required	End for this statement.
<StmtImg>	Binary	Optional OR	Statement Image.
<ImageURL>	URL	Optional OR	URL for retrieving a bank-rendered statement copy.
</BankAcctStmtImgInqRec>			

Tag	Type	Usage	Description
<CSPRefId>	Identifier	Optional	Customer Service Provider Reference Identifier.
<SPRefId>	Identifier	Optional	Service Provider Reference Identifier.

6.12.2 Bank Account Statement Image Reversal

The Bank Account Statement Image Reversal Message allows a client to reverse a previous Bank Account Statement Image Inquiry request.

> Note *In case subsequent messages have to be sent, see 3.2.11.2 Records Control, the first Bank Account Statement Image Inquiry Request Message Aggregate have to be included in the* <BankAcctStmtImgInqRq> *tag. All subsequent messages will be reversed automatically.*

6.12.2.1 Bank Account Statement Image Reversal Request <BankAcctStmtImgRevRq>

Tag	Type	Usage	Description
<RqUID>	UUID	Required	Request Identifier. Sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header Aggregate.
<AsyncRqUID>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously. For more information, see section **3.2.12.1**.
<CustId>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<RevReasonCode>	Open Enum	Required	Reversal reason code. The reason why the client is requesting a reversal.
<Desc>	C-80	Optional	Description, text describing the reason for the reversal.
<BankAcctStmtImgMsgRqInfo>	Aggregate	Required	Object Message Request Information Aggregate. This aggregate contains the message to reverse. See RevOptSupt in the service profile for specific reversal messages supported.
<BankAcctStmtImgInqRq>	Aggregate *see section 6.12.1.1*	Optional	Bank Account Statement Image Inquiry Request Message Aggregate.
</BankAcctStmtImgMsgRqInfo>			

6.12.2.2 Bank Account Statement Image Reversal Response <BankAcctStmtImgRevRs>

Tag	Type	Usage	Description
<Status>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, <StatusCode> defaults to 0 (zero).
<RqUID>	UUID	Required Echoed	Request Identifier.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header Aggregate.
<MsgRsHdr>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header Aggregate.
<AsyncRqUID>	UUID	Optional Echoed	Asynchronous Request Identifier.
<CustId>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Identification Aggregate.
<RevReasonCode>	Open Enum	Required Echoed	Reversal reason code.
<Desc>	C-80	Optional Echoed	Description, text describing the reason for the reversal.
<BankAcctStmtImgMsgRqInfo>	Aggregate	Required Echoed	Object Message Request Information Aggregate. This aggregate contains the message to reverse.
<BankAcctStmtImgInqRq>	Aggregate *see section 6.12.1.1*	Required Echoed	Bank Account Statement Image Inquiry Request Message Aggregate.
</BankAcctStmtImgMsgRqInfo>			
<CSPRefId>	Identifier	Optional	Customer Service Provider Reference Identifier.
<SPRefId>	Identifier	Optional	Service Provider Reference Identifier.

6.13 Payment Enclosed

Allows a client to make a payment towards outstanding debts, such as loans, made by deposit enclosed in an envelope (along with the payment coupon). The <PmtEnclAddRq> creates a new Payment Enclosed transaction. In case of a Payment Enclosed transaction, customers' deposit or credit card accounts are NOT involved in the payment process (it's an 'offline' payment). The payment is simply made by enclosing the check/money order/cash with the payment invoice/statement in an envelope deposited in the ATM's deposit slot. However, the customer needs to enter the amount of the payment when asked by the ATM, which is logged at the ATM (journal) and with the transaction at the acquiring/issuing server/host. This payment is subject to verification by the bank, later (and in case of a dispute, a dispute resolution is performed).

The client sends a <PmtEnclAddRq> message to request a Payment Enclosed. The server replies with a <PmtEnclAddRs> signifying that the Payment Enclosed request was received and is subject to verification by the financial institution.

6.13.1 Payment Enclosed Add

The Payment Enclosed Add message is used by a client to request a payment enclosed transaction.

6.13.1.1   Request <PmtEnclAddRq>

Tag	Type	Usage	Description
<RqUID>	UUID	Required	Request Identifier. Sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header Aggregate.
<AsyncRqUID>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously.
<CustId>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<PmtEnclInfo>	Aggregate *see section 6.3.32*	Required	Payment Enclosed Information aggregate.

6.13.1.2   Response <PmtEnclAddRs>

Tag	Type	Usage	Description
<Status>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, <StatusCode> defaults to 0 (zero).
<RqUID>	UUID	Required Echoed	Request Identifier. Sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header Aggregate.
<MsgRsHdr>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header Aggregate.
<AsyncRqUID>	UUID	Optional Echoed	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously.
<CustId>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<PmtEnclInfo>	Aggregate *see section 6.3.32*	Required Echoed	Payment Enclosed Information aggregate.
<PmtEnclRec>	Aggregate *see section 6.3.32.2*	Required	Payment Enclosed Record Aggregate.

Tag	Type	Usage	Description
<CSPRefId>	Identifier	Optional	Customer Service Provider Reference Identifier.
<SPRefId>	Identifier	Optional	Service Provider Reference Identifier.

6.13.2   Payment Enclosed Modify

The Payment Enclosed Modify message is used to modify the information previously supplied in a Payment Enclosed Add request.

6.13.2.1   Request <PmtEnclModRq>

<RqUID>	UUID	Required	Request Identifier. Sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header Aggregate.
<AsyncRqUID>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously.
<CustId>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<PmtEnclId>	Identifier	Required	Payment Enclosed Identifier. Assigned by the server at the time the Payment Enclosed is first added. Cannot be changed by the client.
<PmtEnclInfo>	Aggregate *see section 6.3.32*	Required	Payment Enclosed Information aggregate.

6.13.2.2   Response <PmtEnclModRs>

<Status>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, <StatusCode> defaults to 0 (zero).
<RqUID>	UUID	Required Echoed	Request Identifier. Sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header Aggregate.
<MsgRsHdr>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header Aggregate.

\<AsyncRqUID\>	UUID	Optional Echoed	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously.
\<CustId\>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in \<SignonRq\>. For example, if a CSR or SP issues the request on behalf of the user, then \<CustId\> is required, and must contain the value of the user whose request is being issued.
\<PmtEnclId\>	Identifier	Required Echoed	Payment Enclosed Identifier.
\<PmtEnclInfo\>	Aggregate *see section 6.3.32*	Required Echoed	Payment Enclosed Information aggregate.
\<PmtEnclRec\>	Aggregate *see section 6.3.32.2*	Required	Payment Enclosed Record Aggregate.
\<CSPRefId\>	Identifier	Optional	Customer Service Provider Reference Identifier.
\<SPRefId\>	Identifier	Optional	Service Provider Reference Identifier.

6.13.3 Payment Enclosed Inquiry

The Payment Enclosed Inquiry message allows a client to get a list of previous Payment Enclosed objects, optionally, based on specified selection criteria.

6.13.3.1 Request \<PmtEnclInqRq\>

\<RqUID\>	UUID	Required	Request Identifier. Sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.
\<MsgRqHdr\>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header Aggregate.
\<AsyncRqUID\>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously.
\<CustId\>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in \<SignonRq\>. For example, if a CSR or SP issues the request on behalf of the user, then \<CustId\> is required, and must contain the value of the user whose request is being issued.
\<RecCtrlIn\>	Aggregate *see section 3.2.12.2.1*	Optional	Records Control Input Aggregate.
\<PmtEnclId\>	Identifier	Optional Repeating	Payment Enclosed Identifier(s).
\<CSPRefId\>	Identifier	Optional Repeating	Customer Service Provider Reference Identifier. This field is used as a selection criterion.

<SPRefId>	Identifier	Optional Repeating	Service Provider Reference Identifier. This field is used as a selection criterion.
<SelRangePrcDt>	Aggregate *see section 3.2.10.1*	Optional XOR	Selection Range Payment Processing Date Aggregate.
<SelRangeDueDt>	Aggregate *see section 3.2.10.1*	Optional XOR	Selection Range Payment Due Date Aggregate.
<SelRangeCurAmt>	Aggregate *see section 3.2.10.2*	Optional	Selection Range Amount Aggregate.

6.13.3.2   Response <PmtEnclInqRs>

<Status>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, <StatusCode> defaults to 0 (zero).
<RqUID>	UUID	Required Echoed	Request Identifier. Sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header Aggregate.
<MsgRsHdr>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header Aggregate.
<AsyncRqUID>	UUID	Optional Echoed	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously.
<CustId>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<RecCtrlOut>	Aggregate *see section 3.2.12.2.2*	Optional *but see Description*	Records Control Output Aggregate. Required if <RecCtrlIn> was provided in the request and the server supports Records Control.
<PmtEnclId>	Identifier	Optional Repeating Echoed	Payment Enclosed Identifier(s).
<CSPRefId>	Identifier	Optional Repeating Echoed	Customer Service Provider Reference Identifier. This field is used as a selection criterion.
<SPRefId>	Identifier	Optional Repeating Echoed	Service Provider Reference Identifier. This field is used as a selection criterion.
<SelRangePrcDt>	Aggregate *see section 3.2.10.1*	Optional XOR Echoed	Selection Range Payment Processing Date Aggregate.

<SelRangeDueDt>	Aggregate *see section* *3.2.10.1*	Optional XOR Echoed	Selection Range Payment Due Date Aggregate.
<SelRangeCurAmt>	Aggregate *see section* *3.2.10.2*	Optional Echoed	Selection Range Amount Aggregate.
<PmtEnclRec>	Aggregate *see section* *6.3.32.2*	Optional Repeating	Payment Enclosed Record Aggregate. These records are generated by the server and reflect the current state of the customer's Payment Enclosed transactions. The records are filtered by the selection criteria specified in the request message.

6.13.4   Payment Enclosed Cancel

The Payment Enclosed Cancel message is used to cancel a previously added Payment Enclosed Add request.

6.13.4.1   Request <PmtEnclCanRq>

<RqUID>	UUID	Required	Request Identifier. Sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.
<MsgRqHdr>	Aggregate *see section* *3.2.1.1*	Optional	Message Request Header Aggregate.
<AsyncRqUID>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously.
<CustId>	Aggregate *see section* *3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<PmtEnclId>	Identifier	Required	Payment Enclosed Identifier. Assigned by the server at the time the Payment Enclosed is first added. Cannot be changed by the client.

6.13.4.2   Response <PmtEnclCanRs>

<Status>	Aggregate *see section* *3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, <StatusCode> defaults to 0 (zero).
<RqUID>	UUID	Required Echoed	Request Identifier. Sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.
<MsgRqHdr>	Aggregate *see section* *3.2.1.1*	Optional Echoed	Message Request Header Aggregate.
<MsgRsHdr>	Aggregate *see section* *3.2.1.2*	Optional	Message Response Header Aggregate.

<AsyncRqUID>	UUID	Optional Echoed	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously.
<CustId>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<PmtEnclId>	Identifier	Required Echoed	Payment Enclosed Identifier.
<PmtEnclRec>	Aggregate *see section 6.3.32.2*	Optional	Payment Enclosed Record Aggregate. This aggregate is provided in cases where the server keeps the Payment Enclosed records on the server even after receiving a cancellation request. This may occur if the cancellation is not immediate, or if the server maintains the record with a status of Cancelled.
<CSPRefId>	Identifier	Optional	Customer Service Provider Reference Identifier.
<SPRefId>	Identifier	Optional	Service Provider Reference Identifier.

6.13.5 Payment Enclosed Audit

The Payment Enclosed Audit message allows to play back the Payment Enclosed messages associated with the current customer since some past point in time.

6.13.5.1 Request <PmtEnclAudRq>

<RqUID>	UUID	Required	Request Identifier. Sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header Aggregate.
<AsyncRqUID>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously.
<CustId>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<RecCtrlIn>	Aggregate *see section 3.2.12.2.1*	Optional	Records Control Input Aggregate.
<SelRangeDt>	Aggregate *see section 3.2.10.1*	Optional	Selection Range Date Aggregate.

<Method>	Closed Enum	Optional Repeating	Audit Selection Action. Used to identify actions associated with the object that is being audited.
			Valid values: Add, Mod, Can, Rev.
			This field is used as a selection criterion.
<PmtEnclId>	Identifier	Optional Repeating	Payment Enclosed Identifier(s).
			This field is used as a selection criterion.

6.13.5.2   Response <PmtEnclAudRs>

<Status>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, <StatusCode> defaults to 0 (zero).
<RqUID>	UUID	Required Echoed	Request Identifier. Sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header Aggregate.
<MsgRsHdr>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header Aggregate.
<AsyncRqUID>	UUID	Optional Echoed	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously.
<CustId>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<RecCtrlOut>	Aggregate *see section 3.2.12.2.2*	Optional *but see Description*	Records Control Output Aggregate.
			Required if <RecCtrlIn> was provided in the request and the server supports Records Control.
<SelRangeDt>	Aggregate *see section 3.2.10.1*	Optional Echoed	Selection Range Date Aggregate.
<Method>	Closed Enum	Optional Repeating Echoed	Audit Selection Action. Used to identify actions associated with the object that is being audited.
			Valid values: Add, Mod, Can, Rev.
			This field is used as a selection criterion.
<PmtEnclId>	Identifier	Optional Repeating Echoed	Payment Enclosed Identifier(s).
			This field is used as a selection criterion.
<PmtEnclMsgRec>	Aggregate	Optional Repeating	Payment Enclosed Message Record Aggregate.
<CustId>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate.

<MsgRecDt>	Date/Time	Optional	Message Record Creation Date. The date/time at which the message record was stored/created by the service provider.
<PmtEnclAddRs>	Aggregate *see section 6.13.1.2*	Required XOR	Payment Enclosed Add Response Message Aggregate.
<PmtEnclModRs>	Aggregate *see section 6.13.2.2*	Required XOR	Payment Enclosed Modify Response Message Aggregate.
<PmtEnclCanRs>	Aggregate *see section 6.13.4.2*	Required XOR	Payment Enclosed Cancel Response Message Aggregate.
<PmtEnclRevRs>	Aggregate *see section 6.13.6.2*	Required XOR	Payment Enclosed Reversal Response Message Aggregate.
</PmtEnclMsgRec>			

6.13.6   Payment Enclosed Reversal

The Payment Enclosed Reversal message allows a client to reverse a previously authorized Payment Enclosed Add transaction request. The Payment Enclosed message is introduced to address issues related to delays in receiving Payment Enclosed Add transaction responses and also to help address situations where the Consumer either cancels or walks-away from the ATM/POS terminal before their transactions are complete. In general, the Payment Enclosed Reversal message provides the capability to "undo" a previous Payment Enclosed Add request in an 'online' environment.

6.13.6.1   Request <PmtEnclRevRq>

<RqUID>	UUID	Required	Request Identifier. Sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header Aggregate.
<AsyncRqUID>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously.
<CustId>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<RevReasonCode>	Open Enum	Required	Reversal reason code. The reason why the client is requesting a reversal.
<Desc>	C-80	Optional	Description, text describing the reason for the reversal.
<PmtEnclMsgRqInfo>	Aggregate	Required	Payment Enclosed Message Request Information Aggregate. This aggregate contains the message to reverse. See RevOptSupt in the service profile for specific reversal messages supported.

\<PmtEnclAddRq\>	Aggregate *see section 6.13.1.1*	Required XOR	Payment Enclosed Add Request Message Aggregate.
\<PmtEnclModRq\>	Aggregate *see section 6.13.2.1*	Required XOR	Payment Enclosed Modify Request Message Aggregate.
\<PmtEnclInqRq\>	Aggregate *see section 6.13.3.1*	Required XOR	Payment Enclosed Inquiry Request Message Aggregate.
\<PmtEnclCanRq\>	Aggregate *see section 6.13.4.1*	Required XOR	Payment Enclosed Cancel Request Message Aggregate.
\<PmtEnclAudRq\>	Aggregate *see section 6.13.5.1*	Required XOR	Payment Enclosed Audit Request Message Aggregate.
\</PmtEnclMsgRqInfo\>			

6.13.6.2   Response \<PmtEnclRevRs\>

\<RqUID\>	UUID	Required Echoed	Request Identifier. Sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.
\<MsgRqHdr\>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header Aggregate.
\<MsgRsHdr\>	Aggregate *see section 3.2.1.2*	Optional	Message Request Header Aggregate.
\<AsyncRqUID\>	UUID	Optional Echoed	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously.
\<CustId\>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in \<SignonRq\>. For example, if a CSR or SP issues the request on behalf of the user, then \<CustId\> is required, and must contain the value of the user whose request is being issued.
\<RevReasonCode\>	Open Enum	Required Echoed	Reversal reason code. The reason why the client is requesting a reversal.
\<Desc\>	C-80	Optional Echoed	Description, text describing the reason for the reversal.
\<PmtEnclMsgRqInfo\>	Aggregate	Required Echoed	Payment Enclosed Message Request Information Aggregate. This aggregate contains the message to reverse. See RevOptSupt in the service profile for specific reversal messages supported.
\<PmtEnclAddRq\>	Aggregate *see section 6.13.1.1*	Required Echoed XOR	Payment Enclosed Add Request Message Aggregate.

<PmtEnclModRq>	Aggregate *see section 6.13.2.1*	Required Echoed XOR	Payment Enclosed Modify Request Message Aggregate.
<PmtEnclInqRq>	Aggregate *see section 6.13.3.1*	Required Echoed XOR	Payment Enclosed Inquiry Request Message Aggregate.
<PmtEnclCanRq>	Aggregate *see section 6.13.4.1*	Required Echoed XOR	Payment Enclosed Cancel Request Message Aggregate.
<PmtEnclAudRq>	Aggregate *see section 6.13.5.1*	Required Echoed XOR	Payment Enclosed Audit Request Message Aggregate.
</PmtEnclMsgRqInfo>			
<CSPRefId>	Identifier	Optional	Customer Service Provider Reference Identifier.
<SPRefId>	Identifier	Optional	Service Provider Reference Identifier.

6.14   Check Issue

6.14.1   Check Issue Add

A commercial customer originates a check issue add message to provide Positive Pay information to a financial institution. This message is generated when a check is printed. The message instructs the financial institution to honor the check if the information on the presented check matches the information in the <ChkIssueInfo> aggregate. If the information does not match, the financial institution should generate a <ChkIssueModRq> to the commercial customer notifying them that the check was not honored. If the commercial customer wishes the financial institution to pay the check anyway, the commercial customer sends a <ChkIssueStatusModRq> with a <ChkIssueStatusCode> of 'Pay'.

6.14.1.1   Request <ChkIssueAddRq>

Tag	Type	Usage	Description
<RqUID>	UUID	Required	Request Identifier.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header Aggregate.
<AsyncRqUID>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously. For more information, see section **3.2.12.1**.
<CustId>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<ChkIssueInfo>	Aggregate *see section 6.3.34*	Required	Check Issue Information Aggregate

6.14.1.2 Response <ChkIssueAddRs>

Tag	Type	Usage	Description
<Status>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, <StatusCode> defaults to 0 (zero).
<RqUID>	UUID	Required Echoed	Request Identifier.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header Aggregate.
<MsgRsHdr>	Aggregate *see section 3.2.1.2*	Optional	Message Request Header Aggregate.
<AsyncRqUID>	UUID	Optional Echoed	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously. For more information, see section **3.2.12.1**.
<CustId>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<ChkIssueInfo>	Aggregate *see section 6.3.34*	Required Echoed	Check Issue Information Aggregate
<ChkIssueRec>	Aggregate *see section 6.3.34.2*	Required	Check Issue Record Aggregate
<CSPRefId>	Identifier	Optional	Customer Service Provider Reference Identifier.
<SPRefId>	Identifier	Optional	Service Provider Reference Identifier.

6.14.2 Check Issue Modification

A financial institution originates a check issue modification message to provide to a commercial customer information as to a check that was presented and the information as provided in the <ChkIssueAddRq> did not match. This would be considered an 'exception item'. This message, <ChkIssueModRq> is generated when a check is not honored. The commercial client responds with a <ChkIssueModRs> noting in the <ChkIssueStatusCode> of the <ChkIssueStatus> as to what action the FI should take on this exception item, i.e., Pay or Rejected.

The <ChkIssueModRq> may also be sent by the commercial client to modify any information originally submitted in the <ChkIssueAddRq>. This would allow the FI to update the system of record.

6.14.2.1 Request <ChkIssueModRq>

Tag	Type	Usage	Description
<RqUID>	UUID	Required	Request Identifier.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header Aggregate.

<AsyncRqUID>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously. For more information, see section **3.2.12.1**.
<CustId>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<ChkIssueId>	Identifier	Required *See Description*	Check Issue Identifier. If a Check Issue Id was not assigned due to a check not included in the <ChkIssueAddRq>, i.e., fraudulant check being presented, a Check Issue Id should be created upon the issuance of the <ChkIssueModRq> by the FI.
<ChkIssueInfo>	Aggregate *see section 6.3.34*	Required	Check Issue Information Aggregate

6.14.2.2   Response <ChkIssueModRs>

<Status>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, <StatusCode> defaults to 0 (zero).
<RqUID>	UUID	Required Echoed	Request Identifier.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header Aggregate.
<MsgRsHdr>	Aggregate *see section 3.2.1.2*	Optional	Message Request Header Aggregate.
<AsyncRqUID>	UUID	Optional Echoed	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously. For more information, see section **3.2.12.1**.
<CustId>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<ChkIssueId>	Identifier	Optional Echoed	Check Issue Identifier.
<ChkIssueInfo>	Aggregate *see section 6.3.34*	Required Echoed	Check Issue Information Aggregate

<ChkIssueRec>	Aggregate *see section 6.3.34.2*	Required	Check Issue Record Aggregate
<CSPRefId>	Identifier	Optional	Customer Service Provider Reference Identifier.
<SPRefId>	Identifier	Optional	Service Provider Reference Identifier.

6.14.3   Check Issue Status Modification

The check issue modification status message modifies the status of an issued check. A financial institution sets the status code to pending when it receives a check issue add message, and the check has not been presented. The financial institution can change the status to Paid when honoring a check, or Rejected if the presented check information does not match the check information on file. A commercial customer can override a rejected check by sending a status modification message with the status code set to Pay.

6.14.3.1   Request <ChkIssueStatusModRq>

<RqUID>	UUID	Required	Request Identifier.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header Aggregate.
<AsyncRqUID>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously.
			For more information, see section **3.2.12.1**.
<CustId>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<ChkIssueId>	Identifier	Required	Check Issue Identifier
<ChkIssueStatus>	Aggregate *see section 6.3.34.1*	Required	Check Issue Status Aggregate

6.14.3.2   Response <ChkIssueStatusModRs>

<Status>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, <StatusCode> defaults to 0 (zero).
<RqUID>	UUID	Required Echoed	Request Identifier.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header Aggregate.

Tag	Type	Usage	Description
<MsgRsHdr>	Aggregate *see section 3.2.1.2*	Optional	Message Request Header Aggregate.
<AsyncRqUID>	UUID	Optional Echoed	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously. For more information, see section **3.2.12.1**.
<CustId>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<ChkIssueId>	Identifier	Required Echoed	Check Issue Identifier
<ChkIssueStatus>	Aggregate *see section 6.3.34.1*	Required Echoed	Check Issue Status Aggregate
<ChkIssueRec>	Aggregate *see section 6.3.34.2*	Required	Check Issue Record Aggregate

6.14.4   Check Issue Delete

A commercial customer can request that a previously sent issued check be removed from the financial institutions system of record by sending a check issue deletion request.

6.14.4.1   Request <ChkIssueDelRq>

Tag	Type	Usage	Description
<RqUID>	UUID	Required	Request Identifier.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header Aggregate.
<AsyncRqUID>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously. For more information, see section **3.2.12.1**.
<CustId>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<ChkIssueId>	Identifier	Required Repeating	Check Issue Identifier

6.14.4.2   Response <ChkIssueDelRs>

Tag	Type	Usage	Description

Tag	Type	Usage	Description
\<Status\>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, \<StatusCode\> defaults to 0 (zero).
\<RqUID\>	UUID	Required Echoed	Request Identifier.
\<MsgRqHdr\>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header Aggregate.
\<MsgRsHdr\>	Aggregate *see section 3.2.1.2*	Optional	Message Request Header Aggregate.
\<AsyncRqUID\>	UUID	Optional Echoed	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously. For more information, see section **3.2.12.1**.
\<CustId\>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in \<SignonRq\>. For example, if a CSR or SP issues the request on behalf of the user, then \<CustId\> is required, and must contain the value of the user whose request is being issued.
\<ChkIssueId\>	Identifier	Required Repeating Echoed	Check Issue Identifier
\<ChkIssueRec\>	Aggregate *see section 6.3.34.2*	Required	Check Issue Record Aggregate
\<CSPRefId\>	Identifier	Optional	Customer Service Provider Reference Identifier.
\<SPRefId\>	Identifier	Optional	Service Provider Reference Identifier.

6.14.5   Check Issue Inquiry

A commercial customer initiates the Check Issue Inquiry message by sending a \<ChkIssueInqRq\> message.

6.14.5.1   Request \<ChkIssueInqRq\>

Tag	Type	Usage	Description
\<RqUID\>	UUID	Required	Request Identifier. Sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.
\<MsgRqHdr\>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header Aggregate.
\<AsyncRqUID\>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously. For more information, see section **3.2.12.1**.

<CustId>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<RecCtrlIn>	Aggregate *see section 3.2.12.2.1*	Optional	Records Control Input Aggregate.
<ChkIssueId>	Identifier	Optional Repeating	Check Issue Identifier. Assigned by the server at the time the Check Issue is first added. Cannot be modified by the client. This field is used as a selection criterion.
<DepAcctId>	Aggregate *see section 3.2.7.1.2*	Optional	Deposit Account Aggregate. Source account for check issue. This field is used as a selection criterion.
<ChkNum>	NC-12	Optional Repeating	Check Number. Assigned by the Pay provider or the CPP if payment is by paper check.
<ChkIssueStatusCode>	Closed Enum	Optional Repeating	Check Issue Status Code. This identifies the check issue processing status. Valid values: Pending, Paid, Rejected, Pay, NoPay This field is used as a selection criterion.
<SelRangeDueDt>	Aggregate *see section 3.2.10.1*	Optional	Selection Range Due Date Aggregate.
<SelRangePaidDt>	Aggregate *see section 3.2.10.1*	Optional	Selection Range PaidDate Aggregate.
<SelRangeCurAmt>	Aggregate *see section 3.2.10.2*	Optional	Selection Range Amount Aggregate.
<CSPRefId>	Identifier	Optional Repeating	Customer Service Provider Reference Identifier. <CSPRefId> is used to inquire about a transaction corresponding to a confirmation number that was returned to the client when the transaction was added or modified. When a transaction has been modified, only the <CSPRefId> received in the most recent PMPMODRS is valid. The use of an <CSPRefId> from an earlier response is likely to result in a "transaction not found" response.
<SPRefId>	Identifier	Optional Repeating	Service Provider Reference Identifier. Same usage as <CSPRefId>.
<IncToken>	Boolean	Optional	Include Token. If *True*, a <Token> must be included in the response, if the Service Profile indicates support for transaction synchronization, to set a base for future synchronization messages. If *False* or omitted, the <Token> may be omitted in the response.
<IncHistory>	Boolean	Optional	Include History. If *True*, the response should include issuances that have already occurred, as well as those scheduled to occur.

6.14.5.2   Response <ChkIssueInqRs>

<Status>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, <StatusCode> defaults to 0 (zero).
<RqUID>	UUID	Required Echoed	Request Identifier.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header Aggregate.
<MsgRsHdr>	Aggregate *see section 3.2.1.2*	Optional	Message Request Header Aggregate.
<AsyncRqUID>	UUID	Optional Echoed	Asynchronous Request Identifier.
<CustId>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<RecCtrlOut>	Aggregate *see section 3.2.12.2.2*	Optional *but see Description*	Records Control Output Aggregate. Required if <RecCtrlIn> was provided in the request and the server supports Records Control.
<ChkIssueId>	Identifier	Optional Repeating Echoed	Check Issue Identifier. Assigned by the server at the time the Check Issue is first added. Cannot be modified by the client. This field is used as a selection criterion.
<DepAcctId>	Aggregate *see section 3.2.7.1.2*	Optional Echoed	Deposit Account Aggregate. Source account for check issue. This field is used as a selection criterion.
<ChkNum>	NC-12	Optional Repeating Echoed	Check Number. Assigned by the Pay provider or the CPP if payment is by paper check.
<ChkIssueStatusCode>	Closed Enum	Optional Repeating Echoed	Check Issue Status Code. This identifies the check issue processing status. Valid values: Pending, Paid, Rejected, Pay, NoPay This field is used as a selection criterion.
<SelRangeDueDt>	Aggregate *see section 3.2.10.1*	Optional Echoed	Selection Range Due Date Aggregate.
<SelRangePaidDt>	Aggregate *see section 3.2.10.1*	Optional Echoed	Selection Range PaidDate Aggregate.
<SelRangeCurAmt>	Aggregate *see section 3.2.10.2*	Optional Echoed	Selection Range Amount Aggregate.

<CSPRefId>	Identifier	Optional Repeating Echoed	Customer Service Provider Reference Identifier. <CSPRefId> is used to inquire about a transaction corresponding to a confirmation number that was returned to the client when the transaction was added or modified. When a transaction has been modified, only the <CSPRefId> received in the most recent PMPMODRS is valid. The use of an <CSPRefId> from an earlier response is likely to result in a "transaction not found" response.
<SPRefId>	Identifier	Optional Repeating Echoed	Service Provider Reference Identifier. Same usage as <CSPRefId>.
<IncToken>	Boolean	Optional Echoed	Include Token. If *True*, a <Token> must be included in the response, if the Service Profile indicates support for transaction synchronization, to set a base for future synchronization messages. If *False* or omitted, the <Token> may be omitted in the response.
<IncHistory>	Boolean	Optional Echoed	Include History. If *True*, the response should include issuances that have already occurred, as well as those scheduled to occur.
<ChkIssueRec>	Aggregate *see section 6.3.34.2*	Optional Repeating	Check Issue Record Aggregate. One record is returned for each of the customer's checks that meets the selection criteria specified in the request message.
<Token>	Identifier	Optional *but see Description*	Token. Server Assigned. Client should make no assumptions about the value of this token relative to others that it may have received. The token is significant only to the server that originally assigned it. The token is only included if <IncToken>=*True* in the request. <Token>=0 is returned if no records are returned within the response.

6.14.6   Check Issue Audit

A commercial customer originates a check issue audit request to obtain all changes related to a particular issued check.

6.14.6.1   Request <ChkIssueAudRq>

<RqUID>	UUID	Required	Request Identifier.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header Aggregate.
<AsyncRqUID>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously. For more information, see section **3.2.12.1**.
<CustId>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.

<RecCtrlIn>	Aggregate *see section 3.2.12.2.1*	Optional	Records Control Input Aggregate.
<DepAcctId>	Aggregate *see section 3.2.7.1.2*	Required	Deposit Account Identification Aggregate.
<SelRangeDt>	Aggregate *see section 3.2.10.1*	Optional	Selection Range Date Aggregate. Date of the Check Issue request.
<ChkRange>	Aggregate *see section 6.3.16*	Optional	Selection Range—Check Number Aggregate.

6.14.6.2   Response <ChkIssueAudRs>

<Status>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, <StatusCode> defaults to 0 (zero).
<RqUID>	UUID	Required Echoed	Request Identifier.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header Aggregate.
<MsgRsHdr>	Aggregate *see section 3.2.1.2*	Optional	Message Request Header Aggregate.
<AsyncRqUID>	UUID	Optional Echoed	Asynchronous Request Identifier.
<CustId>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<RecCtrlOut>	Aggregate *see section 3.2.12.2.2*	Optional *but see Description*	Records Control Output Aggregate. Required if <RecCtrlIn> was provided in the request and the server supports Records Control.
<DepAcctId>	Aggregate *see section 3.2.7.1.2*	Required Echoed	Deposit Account Identification Aggregate.
<SelRangeDt>	Aggregate *see section 3.2.10.1*	Optional Echoed	Check Issue Audit Selection Criteria Aggregate.
<ChkRange>	Aggregate *see section 6.3.16*	Optional Echoed	Selection Range Date Aggregate.
<ChkIssueMsgRec>	Aggregate	Optional Repeating	Check Issue Message Record Aggregate.

<CustId>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the entity who initiated the request (e.g., the one whose identifier appears in the <SignonRq> of the original issued request). For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the CSR or SP.
<MsgRecDt>	Date/Time	Optional	Audit Record Creation Date. Date/time at which the audit record was stored/created by the service provider.
<ChkIssueAddRs>	Aggregate *see section 6.14.1.2*	Required XOR	Check Issue Add Response Message Aggregate.
<ChkIssueModRs>	Aggregate *see section 6.14.2.2*	Required XOR	Check Issue Modify Response Message Aggregate.
<ChkIssueStatusModRs>	Aggregate *see section 6.14.3.2*	Required XOR	Check Issue Status Modify Response Message Aggregate.
<ChkIssueDelRs>	Aggregate *see section 6.14.4.2*	Required XOR	Check Issue Delete Response Message Aggregate.
</ChkIssueMsgRec>			

6.14.7   Check Issue Sync

6.14.7.1   Request <ChkIssueSyncRq>

<RqUID>	UUID	Required	Request Identifier.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header Aggregate.
<AsyncRqUID>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously. For more information, see section **3.2.12.1**.
<CustId>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<RecCtrlIn>	Aggregate *see section 3.2.12.2.1*	Optional	Records Control Input Aggregate.
<DepAcctId>	Aggregate *see section 3.2.7.1.2*	Required	Deposit Account Identification Aggregate.

<Token>	Identifier	Required	Token. Server assigned. Client should make no assumptions about the value of this token relative to others that it may have received. The token is significant only to the server that originally assigned it.
			This is a token that has been previously sent by the server, or zero for first time requests.

6.14.7.2   Response <ChkIssueSyncRs>

<Status>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, <StatusCode> defaults to 0 (zero).
<RqUID>	UUID	Required Echoed	Request Identifier.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header Aggregate.
<MsgRsHdr>	Aggregate *see section 3.2.1.2*	Optional	Message Request Header Aggregate.
<AsyncRqUID>	UUID	Optional Echoed	Asynchronous Request Identifier.
<CustId>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<RecCtrlOut>	Aggregate *see section 3.2.12.2.2*	Optional *but see Description*	Records Control Output Aggregate. Required if <RecCtrlIn> was provided in the request and the server supports Records Control.
<DepAcctId>	Aggregate *see section 3.2.7.1.2*	Required Echoed	Deposit Account Identification Aggregate.
<Token>	Identifier	Required Echoed	Check Issue Synchronization Aggregate.
<NewToken>	Identifier	Required	New Token. Server assigned. Client should make no assumptions about the value of this token relative to others that it may have received. The token is significant only to the server that originally assigned it.
<ChkIssueMsgRec>	Aggregate	Optional Repeating	Check Issue Message Record Aggregate.
<CustId>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the entity who initiated the request (e.g., the one whose identifier appears in the <SignonRq> of the original issued request). For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the CSR or SP.

<MsgRecDt>	Date/Time	Optional	Audit Record Creation Date. Date/time at which the audit record was stored/created by the service provider.
<ChkIssueAddRs>	Aggregate *see section 6.14.1.2*	Required XOR	Check Issue Add Response Message Aggregate.
<ChkIssueModRs>	Aggregate *see section 6.14.2.2*	Required XOR	Check Issue Modify Response Message Aggregate.
<ChkIssueStatusModRs>	Aggregate *see section 6.14.3.2*	Required XOR	Check Issue Status Modify Response Message Aggregate.
<ChkIssueDelRs>	Aggregate *see section 6.14.4.2*	Required XOR	Check Issue Delete Response Message Aggregate.
</ChkIssueMsgRec>			

6.15 Bank Service Checksum

6.15.1 Bank Service Checksum Add

The Bank Service Checksum Add message allows a client to provide a Bank Service Checksum record against which the server can verify receipt of a batch of issued transactions, where the control data such as sum amount and count are provided to the server. The Bank Service Checksum Add message can be sent together with or separate from the associated messages within a bank service to facilitate both cases where the control record is sent via the same or different routes due to authorization and segregation reasons.

6.15.1.1 Request <BankSvcChkSumAddRq>

Tag	Type	Usage	Description
<RqUID>	UUID	Required	Request Identifier. Sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header Aggregate.
<AsyncRqUID>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously. For more information, see section **3.2.12.1**.
<CustId>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<SvcRqUID>	UUID	Optional	Service Request Identifier. Identifier for the file generated by the client. This must be the same as the <RqUID> of <BankSvcRq>, so that the server can match it to the <BankSvcRq> containing the associated issuances.

Tag	Type	Usage	Description
\<BankSvcChkSumInfo\>	Aggregate *see section 6.3.1.1*	Required	Bank Service Checksum Information Aggregate. Provides the BankSvcChkSum information used by the server to validate a bank service batch.

6.15.1.2   Response \<BankSvcChkSumAddRs\>

Tag	Type	Usage	Description
\<Status\>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, \<StatusCode\> defaults to 0 (zero).
\<RqUID\>	UUID	Required Echoed	Request Identifier.
\<MsgRqHdr\>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header Aggregate.
\<MsgRsHdr\>	Aggregate *see section 3.2.1.2*	Optional	Message Request Header Aggregate.
\<AsyncRqUID\>	UUID	Optional Echoed	Asynchronous Request Identifier.
\<CustId\>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Identifier.
\<SvcRqUID\>	UUID	Optional Echoed	Service Request Identifier.
\<BankSvcChkSumInfo\>	Aggregate *see section 6.3.1.1*	Required Echoed	Bank Service Checksum Information
\<BankSvcChkSumRec\>	Aggregate *see section 6.3.1*	Required	Bank Service Checksum Record Aggregate.
\<CSPRefId\>	Identifier	Optional	Customer Service Provider Reference Identifier.
\<SPRefId\>	Identifier	Optional	Service Provider Reference Identifier.

6.15.2   Bank Service Checksum Modify

6.15.2.1   Request \<BankSvcChkSumModRq\>

Tag	Type	Usage	Description
\<RqUID\>	UUID	Required	Request Identifier. Sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.
\<MsgRqHdr\>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header Aggregate.

Tag	Type	Usage	Description
<AsyncRqUID>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously. For more information, see section **3.2.12.1**.
<CustId>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<SvcRqUID>	UUID	Optional	Service Request Identifier. Identifier for the file generated by the client. This must be the same as the <RqUID> of <BankSvcRq>, so that the server can match it to the <BankSvcRq> containing the associated issuances.
<BankSvcChkSumId>	Identifier	Required	Bank Service Checksum Control Identifier. Assigned by the server at the time the Bank Service Checksum Control is first added. Cannot be modified by the client.
<BankSvcChkSumInfo>	Aggregate *see section 6.3.1.1*	Required	Bank Service Checksum Information Aggregate. Provides the checksums used by the server to validate a bank service batch.

6.15.2.2 Response <BankSvcChkSumModRs>

Tag	Type	Usage	Description
<Status>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, <StatusCode> defaults to 0 (zero).
<RqUID>	UUID	Required Echoed	Request Identifier.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header Aggregate.
<MsgRsHdr>	Aggregate *see section 3.2.1.2*	Optional	Message Request Header Aggregate.
<AsyncRqUID>	UUID	Optional Echoed	Asynchronous Request Identifier.
<CustId>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Identifier Aggregate.
<SvcRqUID>	UUID	Optional Echoed	Service Request Identifier.
<BankSvcChkSumId>	Identifier	Required Echoed	Bank Service Checksum Control Identifier.
<BankSvcChkSumInfo>	Aggregate *see section 6.3.1.1*	Required Echoed	Bank Service Checksum Information.
<BankSvcChkSumRec>	Aggregate *see section 6.3.1*	Required	Bank Service Checksum Record Aggregate.

Tag	Type	Usage	Description
\<CSPRefId\>	Identifier	Optional	Customer Service Provider Reference Identifier.
\<SPRefId\>	Identifier	Optional	Service Provider Reference Identifier.

6.15.3   Bank Service Checksum Status Modify

6.15.3.1   Request \<BankSvcChkSumStatusModRq\>

Tag	Type	Usage	Description
\<RqUID\>	UUID	Required	Request Identifier. Sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.
\<MsgRqHdr\>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header Aggregate.
\<AsyncRqUID\>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously. For more information, see section **3.2.12.1**.
\<CustId\>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in \<SignonRq\>. For example, if a CSR or SP issues the request on behalf of the user, then \<CustId\> is required, and must contain the value of the user whose request is being issued.
\<SvcRqUID\>	UUID	Optional	Service Request Identifier. Identifier for the file generated by the client. This must be the same as the \<RqUID\> of \<BankSvcRq\>, so that the server can match it to the \<BankSvcRq\> containing the associated issuances.
\<BankSvcChkSumId\>	Identifier	Required	Bank Service Checksum Control Identifier. Assigned by the server at the time the BankSvcChkSum Control is first added. Cannot be modified by the client.
\<BankSvcChkSumStatus\>	Aggregate *see section 6.3.1.2*	Required	Bank Service Checksum Status Aggregate.

6.15.3.2   Response \<BankSvcChkSumStatusModRs\>

Tag	Type	Usage	Description
\<Status\>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, \<StatusCode\> defaults to 0 (zero).
\<RqUID\>	UUID	Required Echoed	Request Identifier.
\<MsgRqHdr\>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header Aggregate.

Tag	Type	Usage	Description
<MsgRsHdr>	Aggregate *see section 3.2.1.2*	Optional	Message Request Header Aggregate.
<AsyncRqUID>	UUID	Optional Echoed	Asynchronous Request Identifier.
<CustId>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Identifier Aggregate.
<SvcRqUID>	UUID	Optional Echoed	Service Request Identifier.
<BankSvcChkSumId>	Identifier	Required Echoed	Bank Service Checksum Control Identifier.
<BankSvcChkSumStatus>	Aggregate *see section 6.3.1.2*	Required Echoed	Bank Service Checksum Status Aggregate.
<BankSvcChkSumRec>	Aggregate *see section 6.3.1*	Required	Bank Service Checksum Record Aggregate.
<CSPRefId>	Identifier	Optional	Customer Service Provider Reference Identifier.
<SPRefId>	Identifier	Optional	Service Provider Reference Identifier.

6.15.4 Bank Service Checksum Delete

6.15.4.1 Request <BankSvcChkSumDelRq>

Tag	Type	Usage	Description
<RqUID>	UUID	Required	Request Identifier. Optionally sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header Aggregate.
<AsyncRqUID>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously. For more information, see section **3.2.12.1**.
<CustId>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<SvcRqUID>	UUID	Optional	Bank Service Request Identifier. Identifier for the file generated by the client. This must be the same as the <RqUID> of <BankSvcRq>, so that the server can match it to the <BankSvcRq> containing the associated issuances.
<BankSvcChkSumId>	Identifier	Required	Bank Service Checksum Control Identifier. Assigned by the server at the time the Bank Service Checksum Control is first added. Cannot be modified by the client.

6.15.4.2   Response <BankSvcChkSumDelRs>

Tag	Type	Usage	Description
<Status>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, <StatusCode> defaults to 0 (zero).
<RqUID>	UUID	Required Echoed	Request Identifier.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header Aggregate.
<MsgRsHdr>	Aggregate *see section 3.2.1.2*	Optional	Message Request Header Aggregate.
<AsyncRqUID>	UUID	Optional Echoed	Asynchronous Request Identifier.
<CustId>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Identification Aggregate.
<SvcRqUID>	UUID	Optional Echoed	Bank Service Request Identifier.
<BankSvcChkSumId>	Identifier	Required Echoed	Bank Service Checksum Control Identifier.
<BankSvcChkSumRec>	Aggregate *see section 6.3.1*	Required	Bank Service Checksum Record Aggregate.
<CSPRefId>	Identifier	Optional	Customer Service Provider Reference Identifier.
<SPRefId>	Identifier	Optional	Service Provider Reference Identifier.

6.15.5   Bank Service Checksum Inquiry

6.15.5.1   Request <BankSvcChkSumInqRq>

Tag	Type	Usage	Description
<RqUID>	UUID	Required	Request Identifier. Sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header Aggregate.
<AsyncRqUID>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously. For more information, see section **3.2.12.1**.

Tag	Type	Usage	Description
\<CustId\>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in \<SignonRq\>. For example, if a CSR or SP issues the request on behalf of the user, then \<CustId\> is required, and must contain the value of the user whose request is being issued.
\<SvcRqUID\>	UUID	Optional	Bank Service Request Identifier. Identifier for the file generated by the client. This must be the same as the \<RqUID\> of \<BankSvcRq\>, so that the server can match it to the \<BankSvcRq\> containing the associated issuances. This is used as a selection criterion.
\<BankSvcChkSumId\>	Identifier	Required	Bank Service Checksum Identifier. Assigned by the server at the time the bank service check issue is first added. Cannot be modified by the client. This is used as a selection criterion.
\<BankSvcChkSumStatusCode\>	Open Enum	Required	Bank Service Checksum Status Code. Defined values: Pending, Posted, Refused, Rejected, Returned, DelPend, Deleted This is used as a selection criterion.

6.15.5.2   Response \<BankSvcChkSumInqRs\>

Tag	Type	Usage	Description
\<Status\>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, \<StatusCode\> defaults to 0 (zero).
\<RqUID\>	UUID	Required Echoed	Request Identifier.
\<MsgRqHdr\>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header Aggregate.
\<MsgRsHdr\>	Aggregate *see section 3.2.1.2*	Optional	Message Request Header Aggregate.
\<AsyncRqUID\>	UUID	Optional Echoed	Asynchronous Request Identifier.
\<CustId\>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Identification Aggregate.
\<SvcRqUID\>	UUID	Optional Echoed	Bank Service Request Identifier.
\<BankSvcChkSumId\>	Identifier	Required Echoed	Bank Service Checksum Identifier.
\<BankSvcChkSumStatusCode\>	Open Enum	Required Echoed	Bank Service Checksum Status Code.

Tag	Type	Usage	Description
<BankSvcChkSumRec>	Aggregate *see section 6.3.1*	Required	Bank Service Checksum Record Aggregate.
<CSPRefId>	Identifier	Optional	Customer Service Provider Reference Identifier.
<SPRefId>	Identifier	Optional	Service Provider Reference Identifier.

6.15.6   Bank Service Checksum Audit

6.15.6.1   Request <BankSvcChkSumAudRq>

Tag	Type	Usage	Description
<RqUID>	UUID	Required	Request Identifier. Sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header Aggregate.
<AsyncRqUID>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously. For more information, see section **3.2.12.1**.
<CustId>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<RecCtrlIn>	Aggregate *see section 3.2.12.2.1*	Optional	Records Control Input Aggregate.
<SelRangeDt>	Aggregate *see section 3.2.10.1*	Optional	Selection Range Date Aggregate. This is used as a selection criterion.
<Method>	Closed Enum	Optional Repeating	Method. Valid Values: Add, Mod, Can This is used as a selection criterion.
<BankSvcChkSumId>	Identifier	Optional Repeating	Bank Service Checksum Control Identifier. This is used as a selection criterion.

6.15.6.2   Response <BankSvcChkSumAudRs>

Tag	Type	Usage	Description
<Status>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, <StatusCode> defaults to 0 (zero).
<RqUID>	UUID	Required Echoed	Request Identifier.

Tag	Type	Usage	Description
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header Aggregate.
<MsgRsHdr>	Aggregate *see section 3.2.1.2*	Optional	Message Request Header Aggregate.
<AsyncRqUID>	UUID	Optional Echoed	Asynchronous Request Identifier.
<CustId>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Identification Aggregate.
<RecCtrlOut>	Aggregate *see section 3.2.12.2.2*	Optional *but see Description*	Records Control Output Aggregate. Required if <RecCtrlIn> was provided in the request and the server supports Records Control.
<SelRangeDt>	Aggregate *see section 3.2.10.1*	Optional Echoed	Selection Range Date Aggregate.
<Method>	Closed Enum	Optional Repeating Echoed	Method.
<BankSvcChkSumId>	Identifier	Optional Echoed	Bank Service Checksum Identifier.
<BankSvcChkSumMsgRec>	Aggregate	Optional Repeating	Bank Service Checksum Message Record Aggregate.
<CustId>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the entity who initiated the request (e.g., the one whose identifier appears in the <SignonRq> of the original issued request). For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the CSR or SP.
<MsgRecDt>	Timestamp	Optional	Audit Record Creation Date. Date when this audit record is created.
<BankSvcChkSumAddRs>	Aggregate *see section 6.15.1.2*	Required XOR	Bank Service Checksum Add Response Message Aggregate.
<BankSvcChkSumModRs>	Aggregate *see section 6.15.2.2*	Required XOR	Bank Service Checksum Modify Response Message Aggregate.
<BankSvcChkSumStatusModRs>	Aggregate *see section 6.15.3.2*	Required XOR	Bank Service Checksum Status Modify Response Message Aggregate.
<BankSvcChkSumDelRs>	Aggregate *see section 6.15.4.2*	Required XOR	Bank Service Checksum Delete Response Message Aggregate.
</BankSvcChkSumMsgRec>			
<CSPRefId>	Identifier	Optional	Customer Service Provider Reference Identifier.
<SPRefId>	Identifier	Optional	Service Provider Reference Identifier.

6.15.7   Bank Service Checksum Synchronization

6.15.7.1   Request <BankSvcChkSumSyncRq>

Tag	Type	Usage	Description
<RqUID>	UUID	Required	Request Identifier. Sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.
<MsgRqHdr>	Aggregate *see section* 3.2.1.1	Optional Echoed	Message Request Header Aggregate.
<AsyncRqUID>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously. For more information, see section **3.2.12.1**.
<CustId>	Aggregate *see section* 3.2.3.1	Optional	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<RecCtrlIn>	Aggregate *see section* 3.2.12.2.1	Optional	Records Control Input Aggregate.
<Token>	Identifier	Required	Token. Server assigned. Client should make no assumptions about the value of this token relative to others that it may have received. The token is only significant to the server that originally assigned it. This is a token that has been previously sent by the server, or zero for first time requests.

6.15.7.2   Response <BankSvcChkSumSyncRs>

Tag	Type	Usage	Description
<Status>	Aggregate *see section* 3.2.12.1	Optional	Response Status Aggregate. If this aggregate is absent, <StatusCode> defaults to 0 (zero).
<RqUID>	UUID	Required Echoed	Request Identifier.
<MsgRqHdr>	Aggregate *see section* 3.2.1.1	Optional Echoed	Message Request Header Aggregate.
<MsgRsHdr>	Aggregate *see section* 3.2.1.2	Optional	Message Request Header Aggregate.
<AsyncRqUID>	UUID	Optional Echoed	Asynchronous Request Identifier.
<CustId>	Aggregate *see section* 3.2.3.1	Optional Echoed	Customer Identification Aggregate.

Tag	Type	Usage	Description
\<RecCtrlOut>	Aggregate *see section 3.2.12.2.2*	Optional	Records Control Output Aggregate.
\<Token>	Identifier	Required Echoed	Token.
\<NewToken>	Identifier	Required	New Token. Server assigned. Client should make no assumptions about the value of this token relative to others that it may have received. The token is only significant to the server that originally assigned it.
\<BankSvcChkSumMsgRec>	Aggregate	Optional Repeating	Bank Service Checksum Message Record Aggregate.
\<CustId>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the entity who initiated the request (e.g., the one whose identifier appears in the \<SignonRq> of the original issued request). For example, if a CSR or SP issues the request on behalf of the user, then \<CustId> is required, and must contain the value of the CSR or SP.
\<MsgRecDt>	Timestamp	Optional	Audit Record Creation Date. Date when this audit record is created.
\<BankSvcChkSumAddRs>	Aggregate *see section 6.15.1.2*	Required XOR	Bank Service Checksum Add Response Message Aggregate.
\<BankSvcChkSumModRs>	Aggregate *see section 6.15.2.2*	Required XOR	Bank Service Checksum Modify Response Message Aggregate.
\<BankSvcChkSumStatusModRs>	Aggregate *see section 6.15.3.2*	Required XOR	Bank Service Checksum Status Modify Response Message Aggregate.
\<BankSvcChkSumDelRs>	Aggregate *see section 6.15.4.2*	Required XOR	Bank Service Checksum Delete Response Message Aggregate.
\</BankSvcChkSumMsgRec>			
\<CSPRefId>	Identifier	Optional	Customer Service Provider Reference Identifier.
\<SPRefId>	Identifier	Optional	Service Provider Reference Identifier.

6.16   Banking Service Profile \<BankSvcProfInfo>

The Profile for the Banking Service \<BankSvcProfInfo> is defined below. This profile is returned to the client in \<SvcProfInqRq> and provides information on how the client should use the Banking Service.

Tag	Type	Usage	Description
\<SvcCore>	Aggregate *see section 5.2.2.1*	Required	Service Core Aggregate. Information specified for every service.

Tag	Type	Usage	Description
<MsgSupt>	Open Enum	Optional Repeating	Supported Messages. This is a list of messages that are supported for Banking. The convention is to use the name of the message without the Rq or Rs so that each message is only listed once.
			Defined values: BalInq, BalRev, AcctInq, AcctRev, DepAcctStmtInq, DepAcctStmtAdvise, DepAcctStmtRev, CCAcctStmtInq, CCAcctStmtRev, BankAcctStmtImgInq, BankAcctStmtImgRev, DepAcctTrnInq, DepAcctTrnRev, CCAcctTrnInq, CCAcctTrnRev, BankAcctTrnImgInq, BankAcctTrnImgRev, IntRateInq, IntRateRev, BankAcctTaxInq, BankAcctTaxRev, ForExRateInq, ForExRateRev, StpChkAdd, StpChkCan, StpChkInq, StpChkAud, StpChkSync, StpChkRev, XferAdd, XferMod, XferCan, XferStatusMod, XferInq, XferAud, XferSync, XferRev, RecXferAdd, RecXferMod, RecXferCan, RecXferInq, RecXferAud, RecXferSync, RecXferRev, ChkOrdAdd, ChkOrdRev, DepBkOrdAdd, DepBkOrdRev, DebitAuthAdd, DebitAuthMod, DebitAuthStatusMod, DebitAuthCan, DebitAuthInq, DebitAuthAud, DebitAuthSync, DebitAuthRev, DebitAdd, DebitMod, DebitStatusMod, DebitCan, DebitInq, DebitAud, DebitSync, DebitRev, CreditAuthAdd, CreditAuthMod, CreditAuthStatusMod, CreditAuthCan, CreditAuthInq, CreditAuthAud, CreditAuthSync, CreditAuthRev, CreditAdd, CreditMod, CreditStatusMod, CreditCan, CreditInq, CreditAud, CreditSync, CreditRev, DepAppAdd, DepAppInq, DepAppCan, DepAppMod, DepAppAud, DepAppSync, PmtEnclAdd, PmtEnclMod, PmtEnclCan, PmtEnclInq, PmtEnclAud, PmtEnclRev, BankSvcChkSumAdd, BankSvcChkSumMod, BankSvcChkSumStatusMod, BankSvcChkSumDel, BankSvcChkSumInq, BankSvcChkSumAud, BankSvcChkSumSync
<RevOptSupt>	Open Enum	Optional Repeating	Reversal Options Supported. This is a list of reversal messages supported for Banking. The convention is to use the name of the message without the Rq or Rs so that each message is only listed one.
			Defined values: BalInq, AcctInq, DepAcctStmtInq, CCAcctStmtInq, BankAcctStmtImgInq, DepAcctTrnInq, CCAcctTrnInq, BankAcctTrnImgInq, IntRateInq, ForExRateInq, StpChkAdd, StpChkCan, StpChkInq, StpChkAud, StpChkSync, XferAdd, XferMod, XferCan, XferStatusMod, XferInq, XferAud, XferSync, RecXferAdd, RecXferMod, RecXferCan, RecXferInq, RecXferAud, RecXferSync, ChkOrdAdd, DepBkOrdAdd, DebitAuthAdd, DebitAuthMod, DebitAuthCan, DebitAuthInq, DebitAuthAud, DebitAuthSync, DebitAdd, CreditAuthAdd, CreditAuthMod, CreditAuthCan, CreditAuthInq, CreditAuthAud, CreditAuthSync, CreditAdd, DebitAuthStatusMod, DebitMod, DebitCan, DebitInq, DebitAud, DebitSync, DebitStatusMod, CreditAuthStatusMod, CreditMod, CreditCan, CreditInq, CreditAud, CreditSync, CreditStatusMod, PmtEnclAdd, PmtEnclMod, PmtEnclCan, PmtEnclInq, PmtEnclAud
<OptSupt>	Open Enum	Optional Repeating	Options Supported.
			Defined values: RecCtrl, SuppressEcho.
<CustBankPrefSupt>	Open Enum	Optional Repeating	Customer Bank Service Preferences Supported.
			Defined values: Language, MktgInfo, CustWithdrawalPref

Tag	Type	Usage	Description
\<DeliveryMethod\>	Open Enum	Optional Repeating	Delivery Method.
			Defined values: Channel, Courier, Email, Fax, HomeBank, Overnight, Post, LocalPrinter, TwoDay, UPS.
			Note that although these are valid values for this element, they may not be appropriate for a particular message and may result in rejection. In this case, a response must be sent to the customer with an appropriate Status Code.
\<StopChkProf\>	Aggregate	Optional	Stop Check Profile Aggregate. Included if supported.
\<OptSupt\>	Open Enum	Optional Repeating	Options Supported.
			Valid values: StopChkRangeSC.
\<PrcSched\>	Aggregate *see section 5.2.2.3*	Optional	Processing Schedule Aggregate. If omitted, the default processing schedule is assumed.
\<Fee\>	Currency Amount	Optional Repeating	Default Check Stop Fee(s).
\</StopChkProf\>			
\<XferProf\>	Aggregate *see section 6.3.24*	Optional *but see Description*	Transfer Profile Aggregate. Must be included if funds transfers are supported.
\<InterXferProf\>	Aggregate	Optional *but see Description*	Interbank Transfer Profile Aggregate. Must be included if interbank transfers are supported
\<XferProf\>	Aggregate *see section 6.3.24*	Optional	Transfer Profile Aggregate.
\<DomXferFeeCurAmt\>	Currency Amount	Optional	Standard fee for a domestic interbank transfer.
\<IntlXferFeeCurAmt\>	Currency Amount	Optional	Standard fee for an international interbank transfer.
\<USA.ACHProf\>	Aggregate	Optional	ACH Profile Aggregate. This is for use in the United States.
\<DaysWith\>	Long	Required	Number of Days Before Withdrawal. Number of days before processing date that funds are withdrawn.
\<DfltDaysToPay\>	Long	Required	Default Days to Pay.
\<CanWind\>	Long	Required	Cancellation Window—Refers to number of days after a transfer is initiated using ACH during which the transfer may be cancelled.
\</USA.ACHProf\>			
\</InterXferProf\>			

7　　The Pay Service <PaySvc>

The IFX Specification defines all functionality related to consumer and business payment in the Pay Service <PaySvc>.

Clients use the Pay Service to manage a list of payees, schedule individual payments, and define recurring payment models that automatically generate fixed-amount payments at regular intervals to a given payee. Payments may be made to organizations (Billers) with which the CPP (also described in this chapter as the customer's Pay provider, and may be a Financial Institution or Service Provider) has contracts for payment consolidation (Standard Payee), or obtains equivalent information from the BSP (Biller Payee). Payments may also be made to individuals or organizations that have no such relationship with the customer's Pay provider (Fully-Specified Payees) or to accounts at other Financial Institutions (Interbank Transfer Payees).

7.1　　Description

7.1.1　　Payees

7.1.1.1　　Types of Payees

As mentioned previously, a payee may be one of four types: a Standard Payee, a Fully-Specified Payee, a Biller Payee, or a Transfer Payee.

7.1.1.1.1　Standard Payee

A *Standard Payee* is a merchant or payment recipient whose remittance attributes are well known to the CPP. Typically remittance attributes include remittance method (ACH-CIE, ACH-CTX, RPS, ePay, etc.,) concentrator, lockbox, posting exceptions contact, etc. A standard payee is uniquely identified within a CPP with a payee identifier <StdPayeeId>. The customer need only reference the Standard Payee. A customer should not be able to modify the <StdPayeeId> of a standard payee. Standard Payees are typically national, regional, or large local companies or organizations (such as billers) that have contracted with the CPP to consolidate payments from multiple consumers and small businesses. Because the CPP consolidates payments, it already has sufficient remittance information about the biller in its systems, and the customer does not need to enter all the information required for a Fully-Specified Payee. In some countries, Standard Payees are assigned unique identifiers on a national basis, while in others a biller identifier may only be significant to a single CPP. IFX allows for both cases by qualifying a biller number with the name of the organization that assigned it (e.g., the <StdPayeeId> is the Biller as known by the CPP, and the <BillerId> is the Biller as known by the Biller or BSP). Note that anytime a <StdPayeeId> exists outside of a payment message, it must include the CPP <SPName> to scope its value.

7.1.1.1.2　Fully-Specified Payee

A *Fully-Specified Payee* is a payee for which a customer must enter all information needed for his or her Pay provider to identify the payee and payment destination, such as the payee's full name and address. Some Pay providers may also require that customers enter a telephone number for the payee. Most individuals and small businesses are likely to be Fully-Specified Payees.

7.1.1.1.3　Biller Payee

A *Biller Payee* is one where the details about that payee (name, address, acctid, remittance information, payment instruments) have been obtained from a BSP as a result of a Biller Inquiry <BillerInqRq> or Presentment Service/Account Link Add <SvcAcctAddRq>.

7.1.1.1.4　Transfer Payee

A *Transfer Payee* is a payee for which the customer knows the Financial Institution and Account information where a payment may be made using a funds transfer.

7.1.1.1.5　Changing Payee Information

Generally, the customer is permitted to change name/address or destination account information about the *Fully-Specified Payee* and *Transfer Payee,* but the customer cannot change *Biller* name and address information. Only Financial Institutions or Service Providers may change *Biller* information.

7.1.1.2   Common Payee Information

Although a customer must specify some different information for each type of payee, there are a number of attributes common to all types of payees.

A *Payee Name* and, optionally, a *Payee Nickname* may be assigned to each payee. The name is typically the payee's legal name and may be used by the Pay provider as well as the client. A payee nickname may be assigned by the customer and is only used as a user-friendly name for ease of recall for later use.

All Pay providers should allow a *Customer Payee Account* number to be stored as part of each payee record. This number is then sent to the Payee with a payment to allow the payee to correlate the payment with one of their customer accounts. While this element is not required for all payments, it should be included if it is known. The *Customer Payee Account* within the Payee definition is used as a default for all payments to that payee. If a *Customer Payee Account* is specified as part of the payment, it must be used for that payment, but *must not* modify the C*ustomer Payee Account* within the payee definition.

While not explicitly supported within the IFX specification, a customer may have multiple accounts with the payee. For example, a customer who purchases auto and homeowners insurance from the same insurance company may want to set up payments for both policies. The customer adds the insurance company to his payee list as two independent payees, each with it's own customer account number, based on instructions from the insurance company ("use account 11023732 for auto insurance payments, and account 97584324 for homeowners insurance payments.") The customer may use the <Nickname> field to differentiate between payees, e.g., ABC Insurance—Home and ABC Insurance—Auto.

Default Payment Information may also be assigned to a payee so that when a customer is adding a payment or a recurring payment model for that payee, some fields may be pre-populated with default values for the customer's convenience. The customer may always elect to change the values for a particular payment, but the defaults are useful to help a customer quickly enter payment information for common payments. Payment defaults, which may be assigned to a payee, are the funding account for payments, the category to which payments are assigned, and the memo that is associated with the payment.

7.1.1.3   Customer Payee Management—Customer Payee List and Customer Payee Identifier

Some Pay Service Providers require an individual list of Payees for each customer, while others do not. Pay Service Providers within the United States typically support this feature, while Canadian and European systems generally do not support it. A Service Profile option is provided to specify whether this feature is supported or not.

Each customer maintains a list of payees on the Pay provider's server. This list contains payees that the customer has added, either explicitly using the Customer Payee Add message <CustPayeeAddRq> or implicitly using the Payment Add <PmtAddRq> or Recurring Payment Model Add <RecPmtAddRq> messages (which allow a payee to be added along with a payment). In any case, the customer's Payee list may contain any or all of the four types of payees and only contains payees that the customer has added. Each payee on the customer's payee list is assigned a unique identifier <CustPayeeId> that is used for unambiguous communications between the client and server. The <CustPayeeId> is unique only for that particular customer; another customer may use the same <CustPayeeId> value to refer to a different payee.

Once a payee appears on the customer's payee list, the client may use the Customer Payee Modify <CustPayeeModRq> and Customer Payee Delete <CustPayeeDelRq> messages to keep the payee information up-to-date or to remove the payee from the customer's payee list.

A client may request a current view of the customer's payee list at any time by using the Customer Payee Inquiry <CustPayeeInqRq> message. Clients that keep a local copy of a customer's payee list may use this message to refresh their copy of the list. They may also request a "playback" of the messages that affected the customer's payee list since a given time by using the Customer Payee Audit <CustPayeeAudRq> and/or Customer Payee Synchronization <CustPayeeSyncRq> messages.

7.1.1.4 Pay Provider Payee Management—Duplicate Checking and Payee Type Conversion

Pay providers also manage customer Payee Lists. Two common practices among Pay providers are duplicate checking and payee type conversion.

Some Pay providers perform *Duplicate checking* when a Payee record is added to or modified on a customer's Payee List. In this case, the Pay provider verifies that the Payee is not already on the customer's payee list.

Payee type conversion to Biller allows payments to Billers to be paid effectively. A customer may be unaware that a payee is actually considered a biller by his or her Pay provider and may enter a payment to a Fully Specified Payee by entering the payee's complete name and address. The Pay provider may check additions or modifications to a customer's Payee List against its biller database and convert the Fully Specified Payee to a biller payee if a match is detected. This checking may be performed when the Customer Payee is added or modified, or done in a background process, which converts the Fully-Specified payee to a Biller payee at some time after it is added to or modified on the payee list. The customer may obtain information about the payee conversion the next time a Customer Payee Inquiry message is sent, or when the client performs a Customer Payee Synchronization message to update its records.

Payee type conversion to Fully Specified Payee may occur if a Biller terminates a relationship with the customer's Pay provider. In this case, a Fully-Specified Payee aggregate may be substituted for the Biller ID in the Payee record on the customer's Payee List.

7.1.1.5 Standard Payee Inquiry

Although the list of Standard Payees with contractual arrangements with a Pay provider may be very long, it is sometimes desirable for a client to search for a particular Standard Payee. A client may use the Standard Payee Inquiry <StdPayeeInqRq> message for this purpose. If the client finds a biller that the customer is interested in adding to the payee list, the client may add it simply by using the Customer Payee Add <CustPayeeAddRq> message with the Standard Payee's Identifier <StdPayeeId>.

7.1.2 Payments

7.1.2.1 Funding Accounts

The Pay Service uses the <DepAcctIdFrom> and <CardAcctIdFrom> aggregates to identify accounts as funding accounts for payment. When a customer schedules a Payment or creates a Recurring Payment Model, he or she specifies which account should be used as the funding account for that Payment, whether or not it is the default funding account.

A user may activate multiple funding accounts for paying bills and making other payments. If a single funding account is specified during Service Activation, then the User Interface may use that funding account as a default account. The funding account must be specified in <PmtInfo> for each individual payment message .

7.1.2.2 Payment Processing Flow

When a customer decides to make a payment, several dates are important. The date he or she transmits the request to the server is the payment set-up date. The customer may, depending on the Pay Service Profile, either supply the payment processing date <PrcDt> or payment due date <DueDt> in the <PmtInfo> aggregate in the request.

The processing date is the date by which the service must begin processing it for payment. In the case where a customer's CPP actually provides payment support directly, the processing date is the date by which the CPP must retrieve the payment request from its warehouse, generate remittance data, and possibly initiate funds transfers. The processing date is not necessarily the same as the date the funds are withdrawn from the customer's funding account. The actual funds withdrawal date varies depending on who actually processes the funds transfer requests.

The due date is the date the customer's payment needs to be in the hands of the payee. The customer establishes the date when he or she scheduled the payment. If supported by the CSP model, the customer may enter the due date for a payment and the service must calculate the processing date.

A customer processing date may be adjusted by a service for FI holidays and other non-processing days. A processing date may be adjusted either forward or backward depending on the Service Profile. Once the original processing date is adjusted, it is known as the adjusted processing date and must be the actual processing date.

If a user interface or client wishes to allow a customer to enter a due date even though the CSP model is processing date, the client must calculate a processing date for the customer. To aid in that calculation, for each biller payee, <DaysToPay> is provided in <StdPayeeRec>. <DaysToPay> is the number of business days it takes for the CPP to deliver the payment to the payee. This data is provided by the CPP and is returned when the client uses a Standard Payee Inquiry to obtain the Standard Payees. For fully specified payees and transfer payees, the <DfltDaysToPay> (for Fully-Specified payees) and <DfltXferDaysToPay> (for Transfer Payees) are provided in the Service Profile.

To be strictly accurate, a client would also need to know the SPs' holidays, using <HolInqRs>, and non-processing day schedule, via Service Profile <PrcDaysOff>, as well. But in general, the client may subtract the number of business days it would take for payment to arrive (<DaysToPay>, <DfltDaysToPay> or <DfltXferDaysToPay> depending on payee type) from the entered due date, and use that as the processing date. Business day is another term for processing day (the inverse of the <PrcDaysOff> for an FI or its service provider). If the FI holidays were available, the client may also check for and adjust for holidays.

Once a payment is processed, two deliveries take place: remittance data delivery and the funds delivery. Remittance data is information provided to payees about the payment, such as the customer's name, address, the payment amount, and account number with the payee <BillingAcct>. The delivery mechanism for this data may be electronic, such as using the Remittance Add <RemitAddRq> message or another electronic delivery channel, or paper, and may be separate from the actual delivery of the funds. The funds delivery consists of a withdrawal from the customer's funding account and a credit or deposit to the payee's account or a designated account at a BPP for that payee.

A payee may post a payment to the customer's account <BillingAcct> either when they receive the remittance data or when they receive the funds, at the discretion of the payee. The date that posting occurs is called the Biller Posting Date and is reflected in IFX as the Effective Date <EffDt> associated with the Payment Status of *Posted*.

The Biller Posting Date should ideally be earlier than or equal to the Due Date, although this is rarely guaranteed by a CPP due to variations in postal service and billers' accounts receivables systems.

Billers may consider the payment to be "on-time" if it is received by the Due Date, even if they require additional days to process and actually post the payment. If the billing cycle occurs prior to posting, such that the next bill is received without reflecting the prior payment, a customer care issue may arise. In order to facilitate customer care at the CSP/CPP, the biller may provide information to their BSP about the number of days they require to post an electronic payment after it is received <DaysToEPost>. The CSP may then retrieve this data from the BSP, using a Biller Inquiry. <DaysToEPost> is returned within <BillerRec>.

If the payee was paid with a draft drawn on the customer's actual funding account, then the funds withdrawal date mentioned above would be the date the debit cleared the customer's account. However, if the drafts are drawn on an FI consolidation account or if the payee is being paid electronically, the funds may be withdrawn (to go into the consolidation account) at the bank's discretion. This typically takes place once the CPP receives the funds transfer requests generated during payment processing. How a CPP's customers know when their funds are actually withdrawn is typically up to the CPP to disclose when the customer is signing up for the service.

Payment Processing Flow

Payment Setup Date

Adjusted/Actual Processing Date   **Original Processing Date**

Money Delivery (Funds Withdrawal Date)

Remittance Delivery Date

Money Delivery (Deposit Date)

Biller Posting Date

Due Date   *Check Cleared Date*
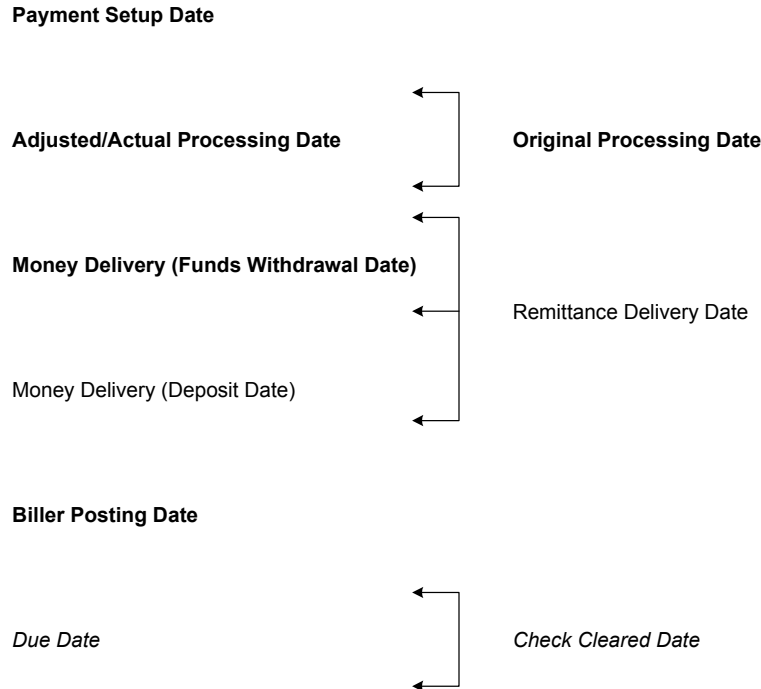
The diagram illustrates the flow of a payment using its significant event dates as briefly introduced in the earlier text. The dates on the right hand side represent floating events. For example, the Remittance Delivery may occur before Funds Withdrawal, before Deposit to the payee, or after the deposit to the payee.

In the above diagram, the dates in bold are dates the customer usually cares about. In the service provider scenario, the customer cares about the Processing Date or Adjusted Processing Date because it may be the only date the service provider may supply about a processed/completed payment. Typically, service providers firmly know the Payment Setup Date, Processing Date, and Adjusted Processing Date.

The other dates in the chart are usually only known by a service provider if the actual money or remittance delivery organization (such as the customer's FI) notified the service provider of them. The same is also true for Biller Posting Date; i.e., a service provider would only know this date if it was supplied by the payees receiving payments and posting them to customer accounts <BillingAcct>.

Due Date and Check Cleared Date are in italics to show they may or may not be of importance to the customer. Due Date may be of importance if the customer enters or is more comfortable with the Due Date model. Check Cleared Date is only of importance if the payment was made via paper check and if the customer is in dispute with their payee about proper credit being given them for a payment.

7.1.2.3 Payment Invoices for Businesses

The IFX Service allows a customer to indicate which invoices are included in a payment, including line item detail. A customer may create a payment that pays one or more invoices or may choose to pay specific line items from one or more invoices.

7.1.2.4 Recurring Payments

By specifying the normal parameters associated with a Payment (funding account, payee, amount, date, etc.) and adding a Recurring Message Detail aggregate, a client may define a Recurring Payment Model. The Recurring Payment Model may be either closed-ended (with a specified total number of payments or last payment date) or open-ended (continue until canceled).

Once a customer has defined Recurring Payment Models, the Pay provider must automatically generate payments based on those models at whatever frequency is specified in the Recurring Message Detail aggregate.

7.1.2.5   Payment Life Cycle and Payment Status

The IFX Specification contains a payment status reporting aggregate <PmtStatus> that is based on a typical payment life cycle. This life cycle may be described as follows:

A payment is *Scheduled*, either by a customer using the Payment Add Request message or by the Pay provider using one of the customer's Recurring Payment Models. Once scheduled, a customer may cancel a payment using the Payment Cancel Request message up until the time that payment is submitted for processing, resulting in a <PmtStatusCode> of *Cancelled*. A customer may request that a payment that has been generated from a recurring payment model be skipped (<PmtStatusCode> of *Skip*).

Payment processing has a number of possible outcomes. A payment may be *processed* with a successful result. A *Processed* payment may be *Returned*. The payment may be posted, rejected or refused by the BPP or Biller, indicated by <PmtStatusCode> values of *Posted, RemitRejected*, or *RemitRefused*, respectively. Those Pay providers that request a Payment Authorization from the customer's Financial Institution before payment may indicate additional payment states, such as *PmtAuthRejected* or *PmtAuthHeld*.

Since a Pay provider may not have access to the customer's bank account data, its records may show *Processed* as the final state for a successful payment. In the case where the Pay provider issues a paper check to a payee on behalf of the customer, the Pay provider may indicate to the customer the check has cleared (<PmtStatusCode> *CheckCleared*).

7.1.2.6   Additional Payment Information

Several elements are provided within the payment instruction to communicate other payment-related information back to the payee. These include the <BillRefInfo> and <Memo> elements, the <InvoiceInfo> aggregate, and the <PmtSummAmt> aggregate.

The <BillRefInfo> element is biller-defined text from the bill summary that may be sent back with the payment to assist in the biller's accounts receivable reconciliation. This may be equivalent to scanline information included with a biller's paper remittance stub.

> *Note:*   *Some payment processing systems currently handle a maximum of 22 characters for <BillRefInfo>, so an implementation must consider data truncation of larger inputs.*

The <Memo> element may be used by CPPs that convert electronic payment instructions into paper checks.

The <InvoiceInfo> aggregate, like the <Memo> element, is information that may be delivered to the payee, either with the payment or through some other means, including email or as a separate paper item. This element may be used to allow a small business to identify, along with a payment, which invoices this payment "pays" and may include other characteristics such as discounts, credit adjustments, and/or free-form text.

The <PmtSummAmt> aggregate is used to indicate additional information about the total payment amount, as specified by the <CurAmt> in <RemitInfo>. It is comprised of the <BillSummAmtId> and the associated currency amount <CurAmt>. The <BillSummAmtId> is obtained from the <BillInfo> on a presented bill and is a tag that identifies the payment category to the biller. This element is used to indicate which payment amount is being paid when multiple payable amounts were presented, or to allocate a supplemental payment to one or more categories, or to designate a breakdown of the total amount into sub-amount categories. Examples of how this data may be presented and used may be found in the text above the <BillInfo> aggregate (see Section 8.3.3.1).

7.2   Pay Service Message Summary

Message / Message Name	Req.	Comments
Standard Payee Inquiry **<StdPayeeInqRq>** **<StdPayeeInqRs>**		Allows a client to inquire about payees that are well known to the CPP, which may be used to assist customers in selecting payees to be added to the customer's personal payee list <CustPayeeId>.

Message / Message Name	Req.	Comments
Standard Payee Reversal **<StdPayeeRevRq>** **<StdPayeeRevRs>**		Allows a client to reverse an action about payees.
Customer Payee Add **<CustPayeeAddRq>** **<CustPayeeAddRs>**		Allows a client to create a new Payee record on the customer's Payee List. This Payee may be one of four types: a Biller Payee specified using <BillerPayee> aggregate, a standard payee specified using <StdPayeeId> aggregate, a Fully-Specified Payee using the <FSPayee> aggregate, or an Interbank Transfer Payee using the <XferPayee> aggregate.
Customer Payee Modify **<CustPayeeModRq>** **<CustPayeeModRs>**		Allows a client to modify an existing Payee record on a customer's Payee List.
Customer Payee Type Modify **<CustPayeeTypeModRq>** **<CustPayeeTypeModRs>**		Allows a CSP to request a change in a Customer's Payee Type.
Customer Payee Delete **<CustPayeeDelRq>** **<CustPayeeDelRs>**		Allows a client to delete an existing Payee record on a customer's Payee List.
Customer Payee Inquiry **<CustPayeeInqRq>** **<CustPayeeInqRs>**		Allows a client to view a summary of current Payee records associated with the current customer. (Payee List)
Customer Payee Audit **<CustPayeeAudRq>** **<CustPayeeAudRs>**		Allows client to play back the Customer Payee messages associated with the current customer since some past point in time.
Customer Payee Synchronization **<CustPayeeSyncRq>** **<CustPayeeSyncRs>**		Allows client to play back the Customer Payee messages associated with the current customer since some past point in time.
Payment Add **<PmtAddRq>** **<PmtAddRs>**	Yes	Allows a client to create a new Payment associated with the current customer. The Payee is automatically added to the customer's Payee List if it is not already there.
Payment Modify **<PmtModRq>** **<PmtModRs>**		Allows a client to modify pending Payment records.
Payment Status Modify **<PmtStatusModRq>** **<PmtStatusModRs>**		Allows an SP to request a modification in the status of a Payment.
Payment Cancel **<PmtCanRq>** **<PmtCanRs>**	Yes	Allows a client to cancel pending Payment records.
Payment Status Advise **<PmtStatusAdviseRq>** **<PmtStatusAdviseRs>**		Allows a client to advise an entity of a change in Payment Status
Payment Inquiry **<PmtInqRq>** **<PmtInqRs>**	Yes	Allows a client to view pending Payment records.

Message / Message Name	Req.	Comments
Payment Reversal **<PmtRevRq>** **<PmtRevRs>**		Allows a client to reverse the Payment messages associated with a current customer.
Payment Audit **<PmtAudRq>** **<PmtAudRs>**		Allows a client to play back the Payment messages associated with the current customer since some past point in time. Note that an audit must return payment messages for both pending and completed payments.
Payment Synchronization **<PmtSyncRq>** **<PmtSyncRs>**		Allows a client to play back the Payment messages associated with the current customer since some past point in time. Note that a sync must return payment messages for both pending and completed payments.
Payment Authorization Add **<PmtAuthAddRq>** **<PmtAuthAddRs>**		Allows a client to request a new Payment Authorization.
Payment Authorization Modify **<PmtAuthModRq>** **<PmtAuthModRs>**		Allows a client to modify existing Payment Authorization records.
Payment Authorization Cancel **<PmtAuthCanRq>** **<PmtAuthCanRs>**	Yes	Allows a client to cancel existing Payment Authorization records.
Payment Authorization Inquiry **<PmtAuthInqRq>** **<PmtAuthInqRs>**	Yes	Allows a client to view existing Payment Authorization records.
Payment Authorization Audit **<PmtAuthAudRq>** **<PmtAuthAudRs>**		Allows a client to play back the Payment Authorization messages since some past point in time. Note that an audit must return payment messages for both pending and completed authorizations.
Payment Authorization Synchronization **<PmtAuthSyncRq>** **<PmtAuthSyncRs>**		Allows a client to play back the Payment Authorization messages since some past point in time. Note that a synchronization must return payment messages for both pending and completed authorizations.
Payment Authorization Reversal **<PmtAuthRevRq>** **<PmtAuthRevRs>**		Allows a client to reverse the action on existing Payment Authorization records.
Remittance Add **<RemitAddRq>** **<RemitAddRs>**		Allows a client to create a new Remittance object.
Remittance Modify **<RemitModRq>** **<RemitModRs>**		Allows a client to modify existing Remittance records.
Remittance Status Modify **<RemitStatusModRq>** **<RemitStatusModRs>**		Allows an SP to request a modification in the status of a Remittance Advice.
Remittance Delete **<RemitDelRq>** **<RemitDelRs>**		Allows a client to delete existing Remittance records.

Message / Message Name	Req.	Comments
Remittance Inquiry **<RemitInqRq>** **<RemitInqRs>**		Allows a client to view existing Remittance records.
Remittance Audit **<RemitAudRq>** **<RemitAudRs>**		Allows a client to play back the Remittance messages since some past point in time. Note that an audit must return payment messages for both pending and completed remittances.
Remittance Synchronization **<RemitSyncRq>** **<RemitSyncRs>**		Allows a client to play back the Payment messages since some past point in time. Note that a synchronization must return payment messages for both pending and completed remittances.
Remittance Reversal **<RemitRevRq>** **<RemitRevRs>**		Allows a client to reverse the action on Remittance records.
Recurring Payment Model Add **<RecPmtAddRq>** **<RecPmtAddRs>**		Allows a client to create a new Recurring Payment Model associated with the current customer. The first Payment may be generated when the Recurring Payment Model is added. The Payee is automatically added to the customer's Payee List if it is not already there.
Recurring Payment Instance Add **<RecPmtInstAddRq>** **<RecPmtInstAddRs>**		Allows a client to generate a payment instance from a Manual frequency Recurring Payment Model.
Recurring Payment Model Modify **<RecPmtModRq>** **<RecPmtModRs>**		Allows a client to modify a Recurring Payment Model associated with the current customer. Pending payments generated from the modified Recurring Payment Model may also be modified if supported by the Pay Provider.
Recurring Payment Model Cancel **<RecPmtCanRq>** **<RecPmtCanRs>**		Allows a client to delete a Recurring Payment Model associated with the current customer. Pending payments generated from the deleted Recurring Payment Model may also be deleted.
Recurring Payment Model Inquiry **<RecPmtInqRq>** **<RecPmtInqRs>**		Allows a client to view a summary of Recurring Payment Models associated with the current customer.
Recurring Payment Model Audit **<RecPmtAudRq>** **<RecPmtAudRs>**		Allows a client to play back the Recurring Payment Model messages associated with the current customer since some past point in time.
Recurring Payment Model Synchronization **<RecPmtSyncRq>** **<RecPmtSyncRs>**		Allows a client to play back the Recurring Payment Model messages associated with the current customer since some past point in time.
Payment Acknowledgement Inquiry **<PmtAckInqRq>** **<PmtAckInqRs>**		Allows a client to retrieve acknowledgement of payments.
Payment Acknowledgement Reversal **<PmtAckRevRq>** **<PmtAckRevRs>**		Allows a client to reverse the action on acknowledgement of payments.

Message / Message Name	Req.	Comments
Payment Acknowledgement Advise **\<PmtAckAdviseRq>** **\<PmtAckAdviseRs>**		Allows a client to advise on the acknowledgement of payments.
Batch Checksum Add **\<ChksumAddRq>** **\<ChksumAddRs>**		Used to provide information associated with the service and its summary control totals.
Batch Checksum Modify **\<ChksumModRq>** **\<ChksumModRs>**		Used to modify information associated with the service and its summary control totals.
Batch Checksum Status Modify **\<ChksumStatusModRq>** **\<ChksumStatusModRs>**		Used to modify the status related to information associated with the service and its summary control totals.
Batch Checksum Delete **\<ChksumDelRq>** **\<ChksumDelRs>**		Used to remove information associated with the service and its summary control totals.
Batch Checksum Inquiry **\<ChksumInqRq>** **\<ChksumInqRs>**		Used to retrieve information associated with the service and its summary control totals.
Batch Checksum Audit **\<ChksumAudRq>** **\<ChksumAudRs>**		Used to track the history of information associated with the service and its summary control totals.
Batch Checksum Synchronization **\<ChksumSyncRq>** **\<ChksumSyncRs>**		Used to play back checksum messages associated with the service and its summary control totals.
Comprehensible Remittance Statement Add **\<CompRemitStmtAddRq>** **\<CompRemitStmtAddRs>**		Allows a financial institution to transmit lockbox remittance information to a client for posting.
Comprehensible Remittance Statement Inquiry **\<CompRemitStmtInqRq>** **\<CompRemitStmtInqRs>**		Allows a customer to inquiry about remittance information on a specific lockbox.
Comprehensible Remittance Statement Audit **\<CompRemitStmtAudRq>** **\<CompRemitStmtAudRs>**		Request to obtain all changes related to a particular comprehensible remittance.
Comprehensible Remittance Statement Synchronization **\<CompRemitStmtSyncRq>** **\<CompRemitStmtSyncRs>**		Request to synchronize to a particular comprehensible remittance.

7.3   Pay Service Common Aggregates

7.3.1   Standard Payee Record \<StdPayeeRec>

Tag	Type	Usage	Description

Tag	Type	Usage	Description
<StdPayeeId>	Aggregate *see section 7.3.1.1*	Required	Customer's Payee Identifier. The Standard Payee as known to the CPP.
<StdPayeeInfo>	Aggregate *see section 7.3.1.2*	Required	Standard Payee Information Aggregate

7.3.1.1   Standard Payee ID <StdPayeeId>

The <StdPayeeId> aggregate is used to describe a *Predefined Payee*, such as a Biller. The <StdPayeeId> is the Biller as known by the CPP. <StdPayeeId> is a synonym of <BillerId>, which is the Biller identification as known by the Biller or BSP.

Tag	Type	Usage	Description
<SPName>	Identifier	Required	Service Provider Name. Used to qualify <BillerNum>. This is the name of the organization that assigned <BillerNum>, i.e., the CPP (or agent of the CPP).
<BillerNum>	Identifier	Required	Biller Number. Assigned by the CPP.

7.3.1.2   Standard Payee Information <StdPayeeInfo>

Tag	Type	Usage	Description
<Name>	C-40	Optional	Payee Name.
<PostAddr>	Aggregate *see section 3.2.3.7.1*	Optional	Standard Payee Address Aggregate.
<AcctMask>	C-32	Optional Repeating	Account Mask. Zero or more edit masks to facilitate <BillingAcct> entry and editing.
<IndustId>	Aggregate *see section 3.2.13*	Optional	Industry Identifier
<DaysToPay>	Long	Required	Days to Pay. Minimum number of business days needed to process. Assigned by the Pay provider. Cannot be changed by the client.

7.3.2   Customer Payee Record Aggregate <CustPayeeRec>

Tag	Type	Usage	Description
<CustPayeeId>	Identifier	Required	Customer's Payee Identifier. If the server supports Payee Lists, the server assigns <CustPayeeId> at the time the Payee is first added.
<CustPayeeInfo>	Aggregate *see section 7.3.2.1*	Required	New Payee Information. This is an instance of <CustPayeeInfo> that provides information about the payee after any adjustment has been made
<BillerContact>	Aggregate *see section 3.2.6.1*	Optional	Biller Contact Information Aggregate.
<DaysToPay>	Long	Required	Days to pay. Number of days required to complete the payment.

7.3.2.1   Customer Payee Information Aggregate <CustPayeeInfo>

The <CustPayeeInfo> aggregate is used in most messages related to Payees and is optionally used in <PmtAddRq> and <RecPmtAddRq> if a new Payment or Recurring Payment Model is being added for a new Customer Payee.

> *Note: <Name> is optional in this aggregate. The following table provides guidelines for including <Name> within implementations.*

Payee Type	Usage	Explanation
Fully Specified Payee	Required	Required in order to issue the check.
Transfer Payee	Recommended	Recommended for problem resolution.
Standard Payee	Recommended	Recommended for problem resolution.
Biller Payee	Required	Required because it is always returned as part of <BillerRec>.

Tag	Type	Usage	Description
<Name>	C-40	Optional *but see Description*	Payee Name. Initially assigned by the customer when the Payee is added. May be modified by the client. Should contain Biller Name <BillerName> in the case of a Biller. For more information, see the above table. Does not affect the behavior of interbank transfer payments, but recommended usage is destination account name for interbank transfer payees.
<Nickname>	C-40	Optional	Payee Nickname. Optionally assigned by the customer. Server always echoes it in immediate response if received in a request. Pay provider may indicate support for longer-term storage of nicknames in the Service Profile. May be modified by the client.
<StdPayeeId>	Aggregate *see section 7.3.1.1*	Required XOR	Standard Payee Identification Aggregate.
<FSPayee>	Aggregate *see section 7.3.2.1.1*	Required XOR	Fully-Specified Payee Aggregate.
<XferPayee>	Aggregate *see section 7.3.2.1.3*	Required XOR	Transfer Payee Aggregate.
<BillerPayee>	Aggregate *see section 7.3.2.1.2*	Required XOR	Biller Payee Aggregate.
<BillingAcct>	C-32	Optional	Customer Account Number with Payee. This is the customer's identification with this Payee. For Transfer Payees, the account number for the target account is carried in the <XferPayee> aggregate, rather than in this element.
<AcctPayAcctId>	Identifier	Optional	Accounts Payable Account Identifier. Vendor Account Number with Customer (Payer). This is the payer's Accounts Payable account number for reconciliation.
<DfltPmtInfo>	Aggregate	Optional	Default Payment Information Aggregate. Used to add default payment information when adding a new payee. This would not be included when paying a bill unless the user was also adding a new payee at the same time.

Tag	Type	Usage	Description
<DepAcctIdFrom>	Aggregate *see section 3.2.7.1.2*	Optional XOR Profiled support	Deposit Account Aggregate. The default funding account must be used when the Payment definition does not specify the funding account. The client may modify the default funding account. Either <DepAcctIdFrom> or <CardAcctIdFrom> may be specified. A default funding account for bill presentment may be specified during service activation when used within <PmtInfo>.
<CardAcctIdFrom>	Aggregate *see section 3.2.7.1.3*	Optional XOR Profiled support	Card Account Identifier Aggregate. Used when the default funding account is a Card account.
<Category>	C-40	Optional	Default Category for Payments. May be modified by the client.
<Memo>	C-255	Optional	Default Memo for Payments. May be modified by the client.
</DfltPmtInfo>			

7.3.2.1.1 Fully-Specified Payee Aggregate <FSPayee>

The <FSPayee> aggregate is used to provide complete information about a Fully-Specified Payee.

Tag	Type	Usage	Description
<PostAddr>	Aggregate *see section 3.2.3.7.1*	Required	Payee Address Aggregate.
<OrgContact>	Aggregate *see section 3.2.5.2*	Required XOR	Payee contact information.
<ContactInfo>	Aggregate *see section 3.2.3.7*	Required XOR	Contact Information Aggregate.
<LegalName>	C-96	Optional	Legal Name. The legal name of the Payee.
<OrgId>	Aggregate *see section 3.2.5.1.1*	Optional Repeating	Organization Identifier.

7.3.2.1.2 Biller Payee Aggregate <BillerPayee>

The <BillerPayee> aggregate is used to provide information for payment to a Biller.

Tag	Type	Usage	Description
<BillerId>	Aggregate *see section 8.3.1.1*	Required	Biller Identification Aggregate.
<BillerContact>	Aggregate *see section 3.2.6.1*	Optional	Biller Contact Information.
<HistRetentionDays>	Long	Optional	Number of days that Bill Summary and Bill Detail information is available for inquiries. The Bill Detail information may be available for a longer period of time.
<BillerPayInfo>	Aggregate *see section 3.2.6.2*	Optional	Biller Pay Information Aggregate.

7.3.2.1.3 Transfer Payee Aggregate <XferPayee>

The <XferPayee> aggregate is used to provide complete information about a Transfer Payee.

Tag	Type	Usage	Description
<DepAcctIdTo>	Aggregate	Required	Deposit Account Aggregate.
			In this context, <DepAcctIdTo> specifies the target account for the transfer.

7.3.3 Payment Record Aggregate <PmtRec>

Tag	Type	Usage	Description
<PmtId>	Identifier	Required	Payment Identifier. Assigned by the server at the time the Payment is first added.
<RecPmtId>	Identifier	Optional *but see Description*	Recurring Payment Model Identifier. Assigned by the server at the time the Recurring Payment Model is first added.
			Returned if payment was generated from a Recurring Payment Model.
<SvcRqUID>	UUID	Optional	Service Request Unique Identifier. The UUID of the service wrapper containing the payment. This is used to identify the batch in which the payment was sent.
<PmtInfo>	Aggregate *see section 7.3.3.1*	Required	Payment Information Aggregate.
<PmtStatus>	Aggregate *see section 7.3.3.2*	Required	Payment Status Aggregate.
<PmtRemitAck>	Aggregate *see section 7.3.3.3*	Optional	Payment Remittance (Credit) Acknowlegement Aggregate. Status record for the "credit side" of the payment transaction.
<PmtRemitInfo>	Aggregate	Optional	Payment Remittance Information. The name and address to which a payment was actually remitted, if different than the name and address for the payee (typically, fully-specified payee).
<Name>	C-40	Optional	Payment Remittance Name. The name of the organization to which the payment was remitted.
<PostAddr>	Aggregate *see section 3.2.3.7.1*	Optional	Postal Address Aggregate.
</PmtRemitInfo>			
<CreatedDt>	Date	Optional	Message Date. Date payment was actually processed and entered history.
<OrigPmtPrcDt>	Date	Optional XOR	Original Payment Processing Date.
<OrigPmtDueDt>	Date	Optional XOR	Original Payment Due Date.
<FIDebitTrcNum>	NC-7	Optional	FI Debit Trace Number.
<FICreditTrcNum>	NC-7	Optional	FI Credit Trace Number.
<CSPRefId>	Identifier	Optional	Customer Service Provider Reference Identifier. This is the <CSPRefId> of the message that created or last modified the payment.

Tag	Type	Usage	Description
\<SPRefId\>	Identifier	Optional	Service Provider Reference Identifier. This is the \<SPRefId\> of the message that created or last modified the payment.

7.3.3.1   Payment Information Aggregate \<PmtInfo\>

The \<PmtInfo\> aggregate is used in most messages related to Payments and Recurring Payments.

Tag	Type	Usage	Description
\<CurAmt\>	Aggregate	Optional	Total Payment Amount. Total debit from funding account.
\<LegalName\>	C-96	Optional XOR	Legal Name. The legal name of the account holder of the funding account.
\<PayerInfo\>	Aggregate *see section 7.3.3.1.1*	Optional XOR	Payer Information. The information about the party that is the account holder of the funding account. Note: In IFX 1.3 or later, this aggregate takes precedence over \<CustId\> if both are present.
\<RemitInfo\>	Aggregate *see section 7.3.5.1*	Required Repeating *but see Description* Profiled support	Remittance Information Aggregate. **Note:** *This aggregate may repeat only in IFX 1.1 or later when the server indicates support for it to repeat, via \<OptSupt\> = PmtMultiRemit.*
\<DepAcctIdFrom\>	Aggregate *see section 3.2.7.1.2*	Required XOR	Deposit Account Aggregate. The funding account for this payment.
\<CardAcctIdFrom\>	Aggregate *see section 3.2.7.1.3*	Required XOR	Card Account Identifier Aggregate. Used for card payments and transfers.
\<Category\>	C-40	Optional	Payment Category. The customer assigns categories.
\<PrcDt\>	Date	Required XOR Profiled support	Payment Processing Date. The customer enters this date as the original requested processing date, not corrected for holidays and non-processing days. May be modified by the customer.
\<DueDt\>	Date	Required XOR Profiled support	Payment Due Date. The customer enters this date to represent the date the payment is due to arrive at the payee. May be modified by the customer.
\<ImmediatePmt\>	Boolean	Optional Profiled support	Immediate Payment Indicator. If set to *True*, the payment must be executed immediately, and not at end of day. Subject to support in Service Profile.
\<ForExDealId\>	Identifier	Optional	Foreign Exchange Deal Identifier (Committed deals).
\<PmtRefId\>	Identifier	Optional	Payment reference identifier. Identifier for the payment generated by the Payer, generally in the Accounts Payable system.

7.3.3.1.1 Payer Information Aggregate \<PayerInfo\>

The \<PayerInfo\> aggregate provides complete information about a Payer.

Tag	Type	Usage	Description
\<PersonInfo\>	Aggregate *see section 3.2.4.1*	Required XOR	Person Information. Personal funding-account holder information.

Tag	Type	Usage	Description
<OrgInfo>	Aggregate *see section 3.2.5.1.2*	Requried XOR	Organization Information. Organization funding-account holder information.
<OrgId>	Aggregate *see section 3.2.5.1.1*	Optional	Organization Identifier. Organizatization funding-account holder ID.

7.3.3.2   Payment Status Aggregate <PmtStatus>

The <PmtStatus> is returned in responses to Add or Modify Payment and Add or Modify Recurring Payment Model.

Tag	Type	Usage	Description
<PmtStatusCode>	Closed Enum	Required	Payment Status Code. This identifies the payment processing status.
			Valid values: Scheduled, Processed, FundAcctDebited, Skipped, Cancelled, Failed, PmtAuthHeld, PmtAuthNoFunds, PmtAuthInactive, PmtAuthClosed, Posted, CheckCleared, Returned, RemitPending, RemitRefused, RemitRejected
<StatusDesc>	C-255	Optional	Status Description. Explanatory text associated with this status.
<EffDt>	DateTime	Required	Payment Status Date. The date associated with the state change to the current state.
<StatusModBy>	Open Enum	Optional	Status Modified By. If present, indicates who modified the Payment Status Code.
			Defined values: Customer, CSP, CSPSR, BSP, BSPSR, CPP, CPPSR, BPP, and BPPSR.
<PmtAuthId>	Identifier	Optional	Payment Authorization Identifier. If present, payment authorization was obtained for the payment and the authorizing SP assigned this identifier. If omitted, no authorization was obtained.
<RemitId>	Identifier	Optional	Remittance Idenitifier. If present, indicates the identifier of the remittance advice associated with this payment. May apply for <PmtStatusCode> of Posted, RemitPending, RemitRejected, or RemitRefused
<RecPmtMod>	Boolean	Optional	Recurring Payment Modified Indicator. Required to be *True* if a <PmtModRq> has subsequently modified a payment generated from a Recurring Payment Model so that it no longer matches the Recurring Payment Model. This may only be supplied for recurring payment instances.
<PmtAuthCount>	Long	Optional	Payment Authorization Count. For Pay providers that require an authorization from the customer's CPP. This is the number of times that the payment has been submitted for authorization.
<PmtMethod>	Open Enum	Optional	Payment Method. Indicates how the payment was transmitted to the payee; e.g., via check or electronically.
			Defined values: Check, Electronic, ACH, FedNet, SWIFT, CHIPS, CHAPS, BookEntry, Draft
<ChkNum>	NC-12	Optional	Check Number.
			Assigned by the Pay provider or the CPP if payment is by paper check.
<Memo>	C-255	Optional	Memo. Intended to be optionally used by a CPP to store any necessary text associated with the payment.

7.3.3.3 Payment Remittance Acknowldegement Aggregate <PmtRemitAck>

Contains information about the credit side of the payment transaction status.   Includes information such as the total transaction fee charged to the payer, Checksum control total of successful or failed credit transactions, and individual credit transaction status.

Tag	Type	Usage	Description
<TotalFeeCharge>	Aggregate	Optional	Total Fee Charge. Total fees charged to the Payer for successful credit transactions remitted to (multiple) payees.
<Count>	Long	Required	Transaction Count. Indicates the number of successfully transacted credit transactions for which the Payee was charged.
<CompositeCurAmt>	Aggregate *see section 3.2.16.1*	Required	Currency Value Sum. Indicates the total fee charged for successfully transacted payment credit transactions.
</TotalFeeCharge>			
<CompositePmtRemitInfo>	Aggregate	Required Repeating	Payment Remittance Acknowledgement Information. Information about the status of payment credit transactions.
<PmtRemitChksum>	Aggregate	Optional	Payment Remittance (Credit) Checksum Control Record.
<AckType>	Open Enum	Required	Acknowledgement Type. Sent by the CPP indicating the state of the transaction processing.
			Defined values: Accepted, Rejected, Pending
<Count>	Long	Required	Transaction Count. Indicates the number of transactions associated with the acknowledgement type.
<TotalCurAmt>	Currency Amount	Required	Currency Value Sum. References the accumulated amount associated with the acknowledgement type.
<Memo>	C-255	Optional	Memo. Text associated with the acknowledgement type.
<CompositeContactInfo>	Aggregate *see section 1.1.1.1*	Optional	CPP Contact Information. Information on how the customer can contact the CPP regarding the trancation set associated with the acknowledgement type.
</PmtRemitChksum>			
<PmtRemitDetail>	Aggregate	Optional Repeating	Payment Remittance (Credit) Detail.
<PmtRemitRefId>	Identifier	Required	Payment Remittance (Credit) Reference Transaction Identifier. Assigned by the client application.
<CurAmt>	Currency Amount	Required	Currency Amount. References the amount of the Payment Remit (credit) fo rthe acknowledgement type.
<PmtRemitStatus>	Aggregate *see section 7.3.3.4*	Optional	Payment Remittance Status.
</PmtRemitDetail>			
</CompositePmtRemitInfo>			

7.3.3.4 Payment Remittance Status Aggregate <PmtRemitStatus>

Tag	Type	Usage	Description
<PmtRemitStatusCode>	Closed Enum	Required	Payment Remittance Status Code. Identifies the Payment Remittance (Credit) status of a payment.
			Valid Values: Accepted, Rejected, Pending
<StatusDesc>	C-255	Optional	Status Description. Explanatory text associated with the status.
<EffDt>	DateTime	Required	Processing Status Date. Date associated with the status change.
<StatusModBy>	Open Enum	Optional	Status Modified By. Indicates who modified the Payment Remittance Status Code.
			Defined values: Customer, CSP, CSPSR, BSP, BSPSR, CPP, CPPSR, BPP, and BPPSR.
<Memo>	C-255	Optional	Memo. Text associated with the Payment Remittance Status.

7.3.3.5 Recurring Payment Record <RecPmtRec>

Tag	Type	Usage	Description
<RecPmtId>	Identifier	Required	Recurring Payment Model Identifier. Assigned by the server at the time the Recurring Payment Model is first added.
<PmtInfo>	Aggregate *see section 7.3.3.1*	Required	Payment Information Aggregate.
<RecModelInfo>	Aggregate *see section 3.2.11.1*	Required	Recurring Model Information Aggregate.
<RemainingInsts>	Long	Required	Remaining Instance Count. The server must calculate this number as the number of actual payments to be made plus the number of instances to skip.
			Server must calculate on <RecPmtAddRq> and return in response. Server must recalculate in case of an <RecPmtModRq> that changes <RecSeriesEnd> or when a payment is spawned. The server is not required to generate a <RecPmtModRs> on each change to <RemainingInsts>.

7.3.4 Payment Authorization Record Aggregate <PmtAuthRec>

Tag	Type	Usage	Description
<PmtAuthId>	Identifier	Required	Payment Authorization Identifier
<PmtAuthInfo>	Aggregate *see section 7.3.4.1*	Required	Payment Authorization Information Aggregate.
<PmtAuthStatus>	Aggregate *see section 7.3.4.2*	Required	Payment Authorization Status Aggregate

7.3.4.1 Payment Authorization Information Aggregate <PmtAuthInfo>

Tag	Type	Usage	Description

Tag	Type	Usage	Description
<DepAcctIdFrom>	Aggregate *see section 3.2.7.1.2*	Required XOR	Deposit Account Aggregate. The funding account for this payment.
<CardAcctIdFrom>	Aggregate *see section 3.2.7.1.3*	Required XOR	Card Account Identifier Aggregate. Used for card payments and transfers. A funding account must be specified.
<CurAmt>	Aggregate	Required	Currency Amount. The amount of funds being requested for a payment authorization.
<Name>	C-80	Optional	Payee name.

7.3.4.2   Payment Authorization Status Aggregate <PmtAuthStatus>

Tag	Type	Usage	Description
<PmtAuthStatusCode>	Open Enum	Required	Payment Authorization Status Code Valid Values are: Authorized, Cleared, Expired
<StatusDesc>	C-255	Optional	Status Description. Explanatory text associated with this status.
<EffDt>	DateTime	Required	Effective Date. Effective date of this status.
<ExpDt>	DateTime	Optional	Expiration Date. Expiration date of this status. For example, the value of Authorized for <PmtAuthStatusCode> may only be valid for a limited amount of time.
<StatusModBy>	Open Enum	Optional	Status Modified By. If present, indicates who modified the Status Code. Defined values: FI

7.3.5   *Remittance Record Aggregate <RemitRec>*

Tag	Type	Usage	Description
<RemitId>	Identifier	Required	Remittance Identifier. Assigned by the server at the time the Remittance is first added.
<RemitInfo>	Aggregate *see section 7.3.5.1*	Required	Remittance Record Aggregate.
<RemitStatus>	Aggregate *see section 7.3.5.2*	Required	Remittance Status Aggregate.

7.3.5.1   Remittance Information Aggregate <RemitInfo>

The <RemitInfo> aggregate is used in most messages related to Remittance of Payments and Recurring Payments. <RemitInfo> contains the elements that pertain to remittance.

Tag	Type	Usage	Description
<RemitInstruction>	Aggregate *see section 7.3.5.1.1*	Optional	Remit instruction. Specifies the remittance delivery instruction, delivery method and provide remit advice codes for reconciliation process to link the payments and remittance advice.
<SettlementInfo>	Aggregate *see section 3.2.15.1*	Optional	Payment/Settlement information.

Tag	Type	Usage	Description
\<CounterpartyInfo\>	Aggregate *see section 7.3.6.1*	Optional	Counterparty Information Aggregate.
\<ChkInfo\>	Aggregate	Optional	Check information.
\<ChkNum\>	C-40	Optional	Check number. Assigned by the Payer or CSP or CPP if payment is by paper check.
\<OrigDt\>	Date	Optional	Check origination date. The date on which the check is originated.
\<Name\>	C-40	Optional	Name of payer to appear on check (as originator). This can be Paying organization (Payer or Invoice Receiver) or check originator (CSP or CPP).
\<PostAddr\>	Aggregate *see section 3.2.3.7.1*	Optional	Postal address of the company to appear on check (as originator). This can be Paying company (Payer or Invoice Receiver) or check originator (CSP or CPP) postal address.
\<OrgPhone\>	Phone Number	Optional	Phone number of the organization to appear on check (as originator). This can be Paying organization (Payer or Invoice Receiver) or check originator (CSP or CPP) phone number.
\<Memo\>	C-255	Optional	Text description on the check.
\</ChkInfo\>			
\<ChkURL\>	URL	Optional	URL location to obtain an image of the check.
\<CustPayeeId\>	Identifier	Required XOR	Customer's Payee Identifier. If the server supports Payee Lists, the server assigns \<CustPayeeId\> at the time the Payee is first added.
\<CustPayeeInfo\>	Aggregate *see section 7.3.2.1*	Required XOR	Payee Information Aggregate. Must be used if \<CustPayeeId\> is unassigned.
The following structure is for use in IFX 1.0.1 or later. Either this structure, or the one following it, may be used, but not both. This structure will be deprecated in IFX 2.0.			
\<BillRefInfo\>	NC-80	Optional	Biller Reference Information. Biller-defined text from \<BillRec\>, for the biller's Accounts Receivable reconciliation. Depending on the Payment Provider and the payment network and message used to make the payment, \<BillRefInfo\> may be truncated to 22 characters.
\<BillId\>	UUID	Optional	Bill Identifier. Bill Service Provider defined universally-unique identifier, to be provided if the payment is related to an IFX-presented bill.
\<PmtId\>	Identifier	Optional	Payment Identifier. The identifier of the payment associated with this remittance. This is included to enable the BPP to inform the CPP of payment posting against this remittance.
\<Memo\>	C-255	Optional	Memo for Payment. From Customer to Payee. Depending on the Payment Provider and the payment network and message used to make the payment, \<Memo\> may be truncated or dropped.
\<BillingAcct\>	C-32	Optional *but see Description*	Customer Account Number with Payee. Required if the Service Provider does not support \<CustPayeeId\>. Inclusion here must not modify the \<BillingAcct\> at the CPP.
\<CurAmt\>	Currency Amount	Required	Currency Amount. The total amount being paid.

Tag	Type	Usage	Description
<PmtSummAmt>	Aggregate	Optional Repeating	Payment Summary Amount. Used to designate the identifier for the amount being paid, or to specify the allocation of the total amount being paid or a portion of the total amount.
<BillSummAmtId>	Identifier	Required	Bill Summary Amount Identifier. Biller's identifier for this currency amount, which is obtained in the <BillInfo> for a presented bill. The consumer can return this in the payment messages to identify to the biller the type of amount being paid or allocated by the consumer.
<CurAmt>	Currency Amount	Required	Currency Amount. Used to specify the amount being associated with the identifier specified by <BillSummAmtId>. This may be the same as the total amount being paid or a portion of the total amount (such as a supplemental payment amount or sub-amount allocation).
</PmtSummAmt>			
<InvoiceInfo>	Aggregate *see section 7.3.6*	Optional Repeating	Invoice Information Aggregate. Provides information about which Invoices and optionally Line Items are paid by this payment. The amount of the payment is always <CurAmt> above, which may or may not match the total of invoices or line items listed here.
The following structure is for use in IFX 1.1.0 or later. Either this structure, or the one preceding it, may be used, but not both. This structure will replace the above structure, which will be deprecated in IFX 2.0.			
<RemitDetail>	Aggregate	Required Repeating	Remittance detail information aggregate.
<InvoiceReceiver>	Aggregate *see section 7.3.5.1.3*	Optional	Invoice/credit memo receiver
<InvoiceSender>	Aggregate *see section 7.3.5.1.2*	Optional	Invoice/credit memo sender
<CurAmt>	Aggregate	Optional *but see Description*	Payment Currency Amount. Total amount being paid on behalf of subsidiary account. This element must be present if <RemitDetail> is repeating.
<PmtSummAmt>	Aggregate	Optional Repeating	Payment Summary Amount. Used to designate the identifier for the amount being paid, or to specify the allocation of the total amount being paid or a portion of the total amount.
<RefInfo>	Aggregate *see section 3.2.17*	Optional Repeating	Reference Information Aggregate. Identification information, typically supplied by the customer, used to correlate the bank account transaction to an associated <XferInfo> or <RemitInfo> sent by the client.
<BillRefInfo>	NC-80	Optional	Biller Reference Information. Biller-defined text from <BillRec>, for the biller's Accounts Receivable reconciliation. Depending on the Payment Provider and the payment network and message used to make the payment, <BillRefInfo> may be truncated to 22 characters. Invoice Sender Reference Information.
<BillId>	UUID	Optional	Bill Identifier. Bill Service Provider defined universally-unique identifier, to be provided if the payment is related to an IFX-presented bill.
<Memo>	C-255	Optional	Memo for Payment. From Customer to Payee. Depending on the Payment Provider and the payment network and message used to make the payment, <Memo> may be truncated or dropped.

Tag	Type	Usage	Description
`<BillingAcct>`	C-32	Optional *but see Description*	Customer Account Number with Payee. Account Reference information supplied by the Payee to the Payer. Required if the Service Provider does not support `<CustPayeeId>`. Inclusion here must not modify the `<BillingAcct>` at the CPP.
`<InvoiceInfo>`	Aggregate *see section 7.3.6*	Optional Repeating	Invoice Information Aggregate. Provides information about which Invoices and optionally Line Items are paid by this payment. The amount of the payment is always `<CurAmt>` above, which may or may not match the total of invoices or line items listed here.
`</RemitDetail>`			
`<PmtId>`	Identifier	Optional	Payment Identifier. The identifier of the payment associated with this remittance. This is included to enable the BPP to inform the CPP of payment posting against this remittance.
`<RefInfo>`	Aggregate *see section 3.2.17*	Optional	Reference information associated with a payment information generated by the payment application.
`<CurAmt>`	Currency Amount	Required	Currency Amount. The total amount being credited to the payee's account.
`<PmtRemitRefId>`	Identifier	Optional	Payment Remittance (Credit) Reference Transaction Identifier.
`<PmtLegalRpt>`	Aggregate *see section 7.3.5.1.4*	Optional	Payment Legal Reporting. For cross border payments in some foreign countries, it is sometimes mandatory to provide legal reporting information to the foreign central bank. This aggregate contains the information to satisfy the legal reporting requirement for payments to foreign payees. It is the responsibility of the reporting party (payer or payee) to pass the payment legal reporting information to CPP for CPP to forward to Central Bank. This information may be pass from the payer to CPP. CPP may then pass it to the country's central bank. CPPs should not modify the information in this aggregate. Following are some countries in which this aggregate is used: Austria, Belgium, Germany, Denmark, Finland, Japan, Netherlands, Norway, and Sweden.

7.3.5.1.1   Remittance Instruction `<RemitInstruction>`

The Remittance Instruction aggregate specifies the remittance delivery method, and remit advice codes for reconciliation process to link the payments and remittance advice.

Tag	Type	Usage	Description
`<RemitRefId>`	Identifier	Optional	Remittance reference identifier. Remittance reference identifier generated for the payment, generally in the Accounts Payable system. Remit reference identifier links a payment and a remittance, and is used to re-join the two information streams when remittance and payment travel separately.
`<PmtRefId>`	Identifier	Optional	Payment reference identifier. The Payee payment reference number generated by the payer associated with the payment, generally in the Accounts Payable system.
`<DeliveryInstruction>`	Open Enum	Optional	Remittance Delivery Instruction. Instruction specifies the responsible party to generate the remittance if the payment and the remittance are to be separated. Define Value: CPPDeliver, CSPDeliver, PayerDeliver.

Tag	Type	Usage	Description
<DeliveryMethod>	Open Enum	Optional	Remittance Delivery Method.
			Defined Values: Fax, Email, EDI, Post, URL
<ContactInfo>	Aggregate *see section 3.2.3.7*	Optional	Remittance contact information. In the case that contact info is needed, may be used in the case the remittance is sent separately from the payment.

7.3.5.1.2 Invoice Sender Aggregate <InvoiceSender>

The Invoice Sender Aggregate describes the Invoice or credit memo sender, which is the organization or person who receives the credit for the payment on the invoice.

Tag	Type	Usage	Description
<OrgId>	Aggregate *see section 3.2.5.1.1*	Required	Organization Identifier.
<OrgInfo>	Aggregate *see section 3.2.5.1.2*	Optional	Organization Identifier Information Aggregate.
<PostAddr>	Aggregate *see section 3.2.3.7.1*	Optional	Invoice Sender's Postal Address.
<AcctPayAcct>	Identifier	Optional	Accounts Payable Account identifier. Vendor Account Number assigned by Customer (Payer). This is the payer's Accounts Payable account number for reconciliation.

7.3.5.1.3 Invoice Receiver Aggregate <InvoiceReceiver>

The Invoice Receiver Aggregate describes the invoice or credit memo receiver, which is the organization or person who owns the liability of the invoice.

Tag	Type	Usage	Description
<OrgId>	Aggregate *see section 3.2.5.1.1*	Required	Organization Identifier.
<OrgInfo>	Aggregate *see section 3.2.5.1.2*	Optional	Organization Identifier Information Aggregate.
<PostAddr>	Aggregate *see section 3.2.3.7.1*	Optional	Invoice Receiver's Postal Address.

7.3.5.1.4 Payment Legal Reporting Aggregate <PmtLegalRpt>

For cross border payments in some foreign countries, it is sometimes mandatory to provide legal reporting information to the foreign central bank. This aggregate contains the information to satisfy the legal reporting requirement for payments to foreign payees. It is the responsibility of the reporting party (payer or payee) to pass the payment legal reporting information to CPP for CPP to forward to Central Bank. This information may be pass from the payer to CPP. CPP may then pass it to the country's central bank. CPPs should not modify the information in this aggregate.

Following are some countries in which this aggregate is used: Austria, Belgium, Germany, Denmark, Finland, Japan, Netherlands, Norway, and Sweden.

Tag	Type	Usage	Description

Tag	Type	Usage	Description
<OrgInfo>	Aggregate *see section 3.2.5.1.2*	Required	Legal reporting organization information. Information about the entity that is obligated to send the notification to central bank about a payment to a foreign payee. This entity could be the invoice receiving organization or the paying organization, depending on different foreign country regulation.
<OrgId>	Aggregate *see section 3.2.5.1.1*	Optional	Organization Identifier. This is the tax identifier as required in legal reporting.
<PayeeCountry>	NC-3	Required	ISO code of the residual country of the payee.
<CurAmt>	Currency Amount	Optional	Payment Currency Amount. Total amount being paid to Payee.
<PrcDt>	Date	Optional	Payment Processing Date. The customer enters this date as the original requested processing date.
<PmtLegalRptData>	Aggregate	Required Repeating	Detail information for legal reporting.
<LegalRptCode>	C-36	Required	Country specific code given by the central bank. Example: T015L-LVAWU
<SupplRptCode>	C-36	Optional	Supplemental country dependent code. Could be more detail or could be another code.
<Desc>	C-80	Optional	Description for additional information to the legal reporting code.
<Memo>	C-255	Optional	Memo for Legal Reporting. From Customer to Central Bank.
<SupplyingCountry>	NC-3	Optional	ISO code of the country from which goods or services come.
<CurAmt>	Currency Amount	Required	Currency Amount. Amount Paid to supplying country on invoice. The Payer provides this amount to the CPP. CPP will pass this to central bank. CPP do nothing with this amount.
<ImportDt>	Date	Optional	Date the goods or services are imported.
</PmtLegalRptData>			

7.3.5.2   Remittance Status Aggregate <RemitStatus>

Tag	Type	Usage	Description
<RemitStatusCode>	Closed Enum	Required	Remit Status Code. This identifies the remittance status. Valid values: Pending, Posted, Refused, Rejected, Returned, DelPend, Deleted
<StatusDesc>	C-255	Optional	Status Description. Explanatory text associated with this status.
<EffDt>	DateTime	Required	Processing Status Date. The date associated with the state change to the current state.
<StatusModBy>	Open Enum	Optional	Status Modified By. If present, indicates who modified the Remit Status Code. Defined values: Customer, CSP, CSPSR, BSP, BSPSR, CPP, CPPSR, BPP, and BPPSR.
<Memo>	C-255	Optional	Memo. Intended to be optionally used by a BPP to store any necessary text associated with the remittance status.

7.3.6 Invoice Information Aggregate <InvoiceInfo>

Tag	Type	Usage	Description
<InvoiceType>	Open Enum	Optional	Invoice Type.
			Defined values:Invoice, CreditMemo
<InvoiceNum>	C-32	Required	Invoice Number. The client enters this identifier since it does not serve as a database key on the server.
<RefInfo>	Aggregate *see section 3.2.17*	Optional Repeating	Reference Information. Additional Invoice reference. Used to unique identify an invoice for either country-specific or company-specific system tracking and reconciliation.
<TotalCurAmt>	Currency Amount	Required	Invoice Due Currency Amount.
<PaidCurAmt>	Currency Amount	Optional	Invoice Paid Currency Amount.
<EffDt>	DateTime	Required	Invoice Date.
<Desc>	C-80	Optional	Invoice Description.
<InvoiceVouchNum>	C-80	Optional	Invoice Voucher Number. The invoice recipient's internal voucher number confirming receipt of the goods listed on the invoice.
<Fee>	Aggregate *see section 3.2.16.2*	Optional Repeating	Invoice Fee(s), such as late fees, etc.
<InvoicePremium>	Currency Amount	Optional	Invoice Premium.
<Discount>	Aggregate	Optional	Discount Information Aggregate.
<Rate>	Decimal	Required XOR	Discount Percentage.
<CurAmt>	Currency Amount	Required XOR	Discount Currency Amount.
<EffDt>	DateTime	Optional	Discount Date.
<Desc>	C-80	Optional	Discount Description.
</Discount>			
<TaxInfo>	Aggregate *see section 3.2.16.3*	Optional	Tax Aggregate.
<InvoiceAdj>	Aggregate	Optional	Adjustment Aggregate.
<InvoiceAdjNum>	C-32	Optional	Adjustment Number.
<AdjType>	Open Enum	Optional	Adjustment Type. This element describes the reason for the adjustment.
			Defined values: Return, Damage
<CurAmt>	Currency Amount	Required	Adjustment Currency Amount.
<EffDt>	DateTime	Optional	Adjustment Date.
<Desc>	C-80	Optional	Adjustment Description. This is often a text representation of <AdjType>.
</InvoiceAdj>			
<InvoiceLineItem>	Aggregate	Optional Repeating	Invoice Line Item Aggregate.

Tag	Type	Usage	Description
<InvoiceLineItemNum>	C-12	Required	Line Item Number
<AdjType>	Open Enum	Optional	Adjustment Type. This element describes the reason for the adjustment.
			Defined values: Return, Damage
<CurAmt>	Currency Amount	Required	Line Item Amount.
<Desc>	C-80	Optional	Line Item Description.
</InvoiceLineItem>			

7.3.6.1 Counterparty Information Aggregate <CounterpartyInfo>

The Counterparty Information aggregate <CounterpartyInfo> contains information regarding the source (originator) party of the transaction posted on the account.

Tag	Type	Usage	Description
<LegalName>	C-96	Optional	Legal Name. Used by entities like trusts, or businesses in the case where an individual is "doing business as" a company. Name of payor of funds.
<RefInfo>	Aggregate *see section 3.2.17*	Optional Repeating	Reference Information Aggregate. References unique to the counterparty of the transaction.
<DepAcctId>	Aggregate *see section 3.2.7.1.2*	Optional XOR	Deposit Account Identification Aggregate
<CardAcctId>	Aggregate *see section 3.2.7.1.3*	Optional XOR	Card Account Identification Aggregate
<LoanAcctId>	Aggregate *see section 3.2.7.1.4*	Optional XOR	Loan Account Identification Aggregate

7.3.7 Checksum Record Aggregate <ChksumRec>

Tag	Type	Usage	Description
<ChksumId>	Identifier	Required	Checksum Identifier. Assigned by the server at the time the Payment is first added.
<ChksumInfo>	Aggregate *see section 7.3.7.1*	Required	Checksum Information Aggregate.
<ChksumStatus>	Aggregate *see section 7.3.7.2*	Required	Checksum Status Aggregate.

7.3.7.1 Checksum Information Aggregate <ChksumInfo>

The Checksum Information <ChksumInfo> aggregate provides information about the payment file and its summary control totals. These control totals provide information necessary for the server to verify the data contained in a Pay Service Request <PaySvcRq>.

Tag	Type	Usage	Description

Tag	Type	Usage	Description
<RefInfo>	Aggregate *see section 3.2.17*	Optional	Payment transaction reference Identifier number. Identifier for the payment file generated by the client.
<MsgType>	Open Enum	Required	Checksum Message Type. Message type of transaction being calculated for the checksum.
			Defined values: All IFX Add and Mod Messages.
			Example: PmtAddRq, RemitAddRq, etc.
<Count>	Long	Required	Transaction count. This count indicates how many request messages should be contained in the file or <PaySvcRq>.
<Chksum>	Aggregate	Optional Repeating	Checksum data. Define different types of Checksum and values for verification of control Checksum totals with file transactions. Example: transaction amount sum, account Id sum or bank Id sum (foreign payment transactions).
<ChksumType>	Open Enum	Required	Checksum type identifier.
			Defined values: Sum, Hash, Mod, etc.
<ChksumValue>	Decimal	Required	Value of checksum type <ChksumType>. Example, if used for the sum, <ChksumValue> contains the total amount of all payment transactions of the file, without currency implication. Thus it contains the total of all <CurAmt>s in all <RemitInfo>s of the associated <PaySvcRq>, reflecting all payment items in the file.
</Chksum>			

7.3.7.2 Checksum Status Aggregate <ChksumStatus>

The Checksum Status <ChksumStatus> is returned in responses to Add or Modify Checksum.

Tag	Type	Usage	Description
<ChksumStatusCode>	Open Enum	Required	Chksum Status Code. This identifies the checksum status.
			Defined values: Pending, Posted, Refused, Rejected, Returned, DelPend, Deleted
<StatusDesc>	C-255	Optional	Status Description. Explanatory text associated with this status.
<EffDt>	DateTime	Required	Status Date. The date associated with the state change to the current state.
<StatusModBy>	Open Enum	Optional	Status Modified By. If present, indicates who modified the Remit Status Code.
			Defined values: Customer, CSP, CSPSR, BSP, BSPSR, CPP, CPPSR, BPP, and BPPSR.
<Memo>	C-255	Optional	Memo. Intended to be optionally used by a BPP to store any necessary text associated with the Checksum Control status.

7.3.8 Comprehensible Remittance Statement Record Aggregate <CompRemitStmtRec>

Tag	Type	Usage	Description
<CompRemitStmtId>	Identifier	Required	Comprehensible Remittance Statement Identifier Aggregate. Assigned by the server at the time the Comprehensible Remittance Statement is first added.

Tag	Type	Usage	Description
`<CompRemitStmtInfo>`	Aggregate *see section 7.3.8.1*	Required	Comprehensible Remittance Statement Information Aggregate.
`<CompRemitStmtStatus>`	Aggregate *see section 7.3.8.2*	Required	Comprehensible Remittance Statement Status.

7.3.8.1 Comprehensible Remittance Statement Information Aggregate `<CompRemitStmtInfo>`

Tag	Type	Usage	Description
`<LockboxDepId>`	Aggregate *see section 7.3.8.1.1*	Optional XOR	Lockbox Deposit Identifier. A unique identifier assigned to a specific lockbox where deposits are collected.
`<DepAcctId>`	Aggregate *see section 3.2.7.1.2*	Optional XOR	Deposit Account Id Aggregate. Deposit account for remittanc etransactions.
`<CurAmt>`	Currency Amount	Required	Currency Amount. Total amunt of the remittance deposit cycle.
`<Count>`	Long	Optional	Count. Total number of batches processed in the remittance cycle.
`<RefInfo>`	Aggregate *see section 3.2.17*	Optional	Reference information associated with remittance deposit information.
`<PrcDt>`	DateTime	Required	Processing Date. The date the remittance payments were processed by the financial institution.
`<BatchInfo>`	Aggregate *see section 7.3.8.1.2*	Optional Repeating	Batch Information Aggregate. One or more batches may be processed within a remittance deposit.
`<LockboxURL>`	URL	Optional	Lockbox URL. URL containing lockbox remittance information.

7.3.8.1.1 Lockbox Deposit Identifier Aggregate `<LockboxDepId>`

Tag	Type	Usage	Description
`<LockboxAcctId>`	Identifier	Required	Lockbox Account Identifier. Assigned by FI to identify lockbox to customer, i.e., PO Box number at post office
`<DepAcctId>`	Aggregate *see section 3.2.7.1.2*	Optional	Deposit Account Id Aggregate. Deposit account for lockbox transactions.

7.3.8.1.2 Batch Information Aggregate `<BatchInfo>`

Tag	Type	Usage	Description
`<RefInfo>`	Aggregate *see section 3.2.17*	Optional	Reference Information Aggregate. Reference information associated with remittance information.
`<CurAmt>`	Currency Amount	Required	Currency Amount. Total amount of the remittance deposit cycle.

Tag	Type	Usage	Description
<Count>	Long	Required	Transaction count. Indicates the number of request messages contained in the file or <PaySvcRq>.
<RemitInfo>	Aggregate *see section 7.3.5.1*	Required Repeating	Remittance Record Aggregate.

7.3.8.2   Comprehensible Remittance Statement Status Aggregate <CompRemitStmtStatus>

Tag	Type	Usage	Description
<CompRemitStmtStatusCode>	Closed Enum	Required	Comprehensible Remittance Status Code. Values: Pending, Deposited, Rejected
<StatusDesc>	C-255	Optional	Status Description. Explanatory text associated with the comprehensible remittance status.
<EffDt>	DateTime	Optional	Effective Date/Time. The date/time the status code took effect.
<StatusModBy>	Open Enum	Optional	Status Mod By. Indicates who modified the status. Defined Values: CSP, FI

7.3.9   Payment Status Record Aggregate <PmtStatusRec>

The <PmtStatusRec> contains information regarding the status of the payment at the CPP or the Payment Network.

Tag	Type	Usage	Description
<PmtId>	Identifier	Required	Payment Identifier. The identifier of the payment matching the selection criteria of the request.
<PmtStatus>	Aggregate *see section 7.3.3.2*	Required	Payment Status Aggregate.
<PmtRefId>	Identifier	Optional	Payment reference identifier. Identifier for the payment generated by the client. It can be the document generated by the Payer's Accounts Payable system, or it can be the identifier generated in the payment request (same as the <RqUId> of the <PmtAddRq>). **Note:** *If CPP uses the AddRq or InqRq message to push acknowledgement to client, there is no place to echo back the <RqUID> from the <PmtAddRq>, so this <PmtRefId> is required for client to link the status back to the original payment request.*
<SvcRqUID>	UUID	Optional	Service Request Identifier. The payment service request identifier that was created and sent by the client to the CPP.
<NetworkTrnInfo>	Aggregate *see section 3.2.18*	Optional	Network Transaction Information Aggregate. Contains transaction identifiers for the network that processed the payment.

Tag	Type	Usage	Description
`<CompositeContactInfo>`	Aggregate *see section 1.1.1.1*	Optional	CPP Remittance contact information. In the case that contact info is needed, may be used in the case the URL is needed to link the customer back to the CPP site for additional information about the service or transaction. Remittance is sent separately from the payment.
`<PmtRemitAck>`	Aggregate *see section 7.3.3.3*	Optional Repeating	Payment Remittance (Credit) Status Record Aggregate.

7.3.10 Payment Acknowledgement Record Aggregate <PmtAckRec>

The <PmtAckRec> contains acknowledgment records at a consolidated or summarized level applicable to a <SvcRqUID>, or batch of payments submitted by the customer.

Tag	Type	Usage	Description
`<SvcRqUID>`	UUID	Required	Service Request Identifier. The payment file identifier that was created by the payer in the Accounts Payable system when the file is created and sent.
`<PmtAckInfo>`	Aggregate	Required Repeating	Payment Acknowledgement Information Aggregate.
`<PmtStatusCode>`	Closed Enum	Required	Payment Status Code. This identifies the payment processing status. Valid values: Scheduled, Processed, Failed
`<Count>`	Long	Required	Transaction counts. This count would indicate how many requested transactions were acknowledged for the type indicated.
`<CurAmt>`	Currency Amount	Required	Amount. This amount references the accumulated amount for the <AckType> identified above.
`<Memo>`	C-255	Optional	Memo for appropriate text concerning the acknowledgment.
`<CompositeContactInfo>`	Aggregate *see section 1.1.1.1*	Optional	CPP contact information. In the case that contact info is needed, may be used in the case the URL is needed to link the customer back to the CPP site for additional information about the service or transaction.
`</PmtAckInfo>`			

7.4 Standard Payee

7.4.1 Standard Payee Inquiry

The Standard Payee Inquiry message may be used to assist customers in selecting payees to be added to the customer's personal payee list <CustPayeeId>. The response returns one or more well-known payees that match the selection criteria of the request.

7.4.1.1 Request <StdPayeeInqRq>

If the client omits all search elements in the <StdPayeeInqRq>, the client is requesting a complete directory of payees. Otherwise, the client wants to filter results based on the included elements.

For each payee that matches the elements in the request, the CPP returns the complete name and address of the payee, the payee ID, and the CPP name.

This inquiry request is similar to the <BillerInqRq>, which returns billers serviced by a BSP.

Note: billers known by a <BillerId> to a BSP may be well-known merchants to a CPP and therefore will have a corresponding <StdPayeeId>. In fact, an entity playing both BSP and CSP roles may use the same actual value, but implementers must never count on this.

Tag	Type	Usage	Description
<RqUID>	UUID	Required	Request Identifier. Sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header aggregate.
<AsyncRqUID>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously.
			For more information, see section 3.2.12.1.
<CustId>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<RecCtrlIn>	Aggregate *see section 3.2.12.2.1*	Optional	Records Control Input Aggregate.
<StdPayeeId>	Aggregate *see section 7.3.1.1*	Optional Repeating	Payee Identification. This field is used as a selection criterion.
<Name>	C-40	Optional Repeating	Payee Name. This field is used as a selection criterion.
<BillingAcct>	C-32	Optional Repeating	Customer (Billing) Account Number with the Payee. Useful for determining appropriate merchant instance via account scheme.
<PostalCode>	C-11	Optional Repeating	Postal Code. This is the postal code of the billing account. This field is used as a selection criterion. (1) It may used to limit the search to billers doing business within a limited geography, or (2) It may be used to identify the correct legal/entity or remittance address such as the cable company for a specific city, e.g. TCI—Sunnyvale CA.
<IndustId>	Aggregate *see section 3.2.13*	Optional Repeating	Industry Identifier
<DeliveryMethod>	Open Enum	Optional Profiled values	Delivery Method. Default is Channel. Value must be supported in Service Profile.

7.4.1.2   Response <StdPayeeInqRs>

Tag	Type	Usage	Description
<Status>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, <StatusCode> defaults to 0 (zero).

Tag	Type	Usage	Description
<RqUID>	UUID	Required Echoed	Request Identifier.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header aggregate.
<MsgRsHdr>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header aggregate.
<AsyncRqUID>	UUID	Optional Echoed	Asynchronous Request Identifier.
<CustId>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<RecCtrlOut>	Aggregate *see section 3.2.12.2.2*	Optional *but see Description*	Records Control Output Aggregate. Required if <RecCtrlIn> was provided in the request and the server supports Records Control.
<StdPayeeId>	Aggregate *see section 7.3.1.1*	Optional Repeating Echoed	Payee Identification.
<Name>	C-40	Optional Repeating Echoed	Payee Name. This field is used as a selection criterion.
<BillingAcct>	C-32	Optional Repeating Echoed	Customer (Billing) Account Number with the Payee. Useful for determining appropriate merchant instance via account scheme.
<PostalCode>	C-11	Optional Repeating Echoed	Postal Code. This is the postal code of the billing account. This field is used as a selection criterion. (1) It may used to limit the search to billers doing business within a limited geography, or (2) It may be used to identify the correct legal/entity or remittance address such as the cable company for a specific city, e.g. TCI—Sunnyvale CA.
<IndustId>	Aggregate *see section 3.2.13*	Optional Repeating Echoed	Industry Identifier
<DeliveryMethod>	Open Enum	Optional Echoed	Delivery Method.
<StdPayeeRec>	Aggregate *see section 7.3.1*	Optional Repeating	Standard Payee Record Aggregate. One record is returned for each Payee defined for this customer who meets the selection criteria in the request.

7.4.2   *Standard Payee Reversal*

7.4.2.1   Request <StdPayeeRevRq>

Tag	Type	Usage	Description
<RqUID>	UUID	Required	Request Identifier.

Tag	Type	Usage	Description
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header aggregate.
<AsyncRqUID>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously.
			For more information, see section **3.2.12.1**.
<CustId>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<RevReasonCode>	Open Enum	Required	Reversal reason code. The reason why the client is requesting a reversal.
<Desc>	C-80	Optional	Description, text describing the reason for the reversal.
<StdPayeeMsgRqInfo>	Aggregate	Required	Standard Payee Message Request Information Aggregate. This aggregate contains the message to reverse. See RevOptSupt in the service profile for specific reversal messages supported.
<StdPayeeInqRq>	Aggregate *see section 7.4.1.1*	Optional	Standard Payee Inquiry Request Message Aggregate.
</StdPayeeMsgRqInfo>			

7.4.2.2   Response <StdPayeeRevRs>

Tag	Type	Usage	Description
<Status>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, <StatusCode> defaults to 0 (zero).
<RqUID>	UUID	Required Echoed	Request Identifier.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header aggregate.
<MsgRsHdr>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header aggregate.
<AsyncRqUID>	UUID	Optional Echoed	Asynchronous Request Identifier.
<CustId>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Identification Aggregate.
<RevReasonCode>	Open Enum	Required Echoed	Reversal reason code. The reason why the client is requesting a reversal.
<Desc>	C-80	Optional Echoed	Description, text describing the reason for the reversal.

Tag	Type	Usage	Description
\<StdPayeeMsgRqInfo\>	Aggregate	Required Echoed	Standard Payee Message Request Information Aggregate. This aggregate contains the message to reverse. See RevOptSupt in the service profile for specific reversal messages supported.
\<StdPayeeInqRq\>	Aggregate *see section 7.4.1.1*	Optional	Standard Payee Inquiry Request Message Aggregate.
\</StdPayeeMsgRqInfo\>			
\<CSPRefId\>	Identifier	Optional	Customer Service Provider Reference Identifier.
\<SPRefId\>	Identifier	Optional	Service Provider Reference Identifier.

7.5 Customer Payee

7.5.1 Customer Payee Add

The client uses the Customer Payee Add message to add a Payee to a Customer Payee list on the Pay provider's server.

7.5.1.1 Request \<CustPayeeAddRq\>

Tag	Type	Usage	Description
\<RqUID\>	UUID	Required	Request Identifier. Sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.
\<MsgRqHdr\>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header aggregate.
\<AsyncRqUID\>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously. For more information, see section 3.2.12.1.
\<CustId\>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in \<SignonRq\>. For example, if a CSR or SP issues the request on behalf of the user, then \<CustId\> is required, and must contain the value of the user whose request is being issued.
\<CustPayeeInfo\>	Aggregate *see section 7.3.2.1*	Required	Payee Record Aggregate.

7.5.1.2 Response \<CustPayeeAddRs\>

Tag	Type	Usage	Description
\<Status\>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, \<StatusCode\> defaults to 0 (zero).
\<RqUID\>	UUID	Required Echoed	Request Identifier.

Tag	Type	Usage	Description
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header aggregate.
<MsgRsHdr>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header aggregate.
<AsyncRqUID>	UUID	Optional Echoed	Asynchronous Request Identifier.
<CustId>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<CustPayeeInfo>	Aggregate *see section 7.3.2.1*	Required Echoed	Payee Record Aggregate.
<CustPayeeRec>	Aggregate *see section 7.3.2*	Required	Customer Payee Record.
<CSPRefId>	Identifier	Optional	Customer Service Provider Reference Identifier.
<SPRefId>	Identifier	Optional	Service Provider Reference Identifier.

7.5.2   *Customer Payee Modify*

The Modify Payee message allows a client to change information about a payee on a customer's Payee list on the Pay provider's server.

7.5.2.1   Request <CustPayeeModRq>

The <CustPayeeModRq> message is sent by a client to modify a Payee record on a customer's Payee list on the server.

Tag	Type	Usage	Description
<RqUID>	UUID	Required	Request Identifier. Sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header aggregate.
<AsyncRqUID>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously. For more information, see section 3.2.12.1.
<CustId>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.

Tag	Type	Usage	Description
<CustPayeeId>	Identifier	Required	Customer's Payee Identifier. If the server supports Payee Lists, the server assigns <CustPayeeId> at the time the Payee is first added.
<CustPayeeInfo>	Aggregate *see section 7.3.2.1*	Required	Payee Info Aggregate.
<ModPending>	Boolean	Optional Profiled support	Modify Pending Payments indicates whether to propagate the payee change to pending single payments. Changes to payees are always propagated to payment models. If *True*, changes to payees are propagated to pending payments If *False*, or omitted, changes are not propagated. This option must be ignored when <ModPendingType> ≠ IfRequested in the Pay Service Profile.

7.5.2.2   Response <CustPayeeModRs>

The server sends the <CustPayeeModRs> message in response to a <CustPayeeModRq> message from a client.

Tag	Type	Usage	Description
<Status>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, <StatusCode> defaults to 0 (zero).
<RqUID>	UUID	Required Echoed	Request Identifier.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header aggregate.
<MsgRsHdr>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header aggregate.
<AsyncRqUID>	UUID	Optional Echoed	Asynchronous Request Identifier.
<CustId>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<CustPayeeId>	Identifier	Required Echoed	Customer's Payee Identifier.
<CustPayeeInfo>	Aggregate *see section 7.3.2.1*	Required Echoed	Payee Record Aggregate.
<ModPending>	Boolean	Optional Profiled support Echoed	Modify Pending Payments Flag.
<CustPayeeRec>	Aggregate *see section 7.3.2*	Required	Customer Payee Record.
<CSPRefId>	Identifier	Optional	Customer Service Provider Reference Identifier.

Tag	Type	Usage	Description
<SPRefId>	Identifier	Optional	Service Provider Reference Identifier.

7.5.3 Customer Payee Type Modify

A user with special authorization, typically a CSR, may send a request to the Payment Service Provider to change the payee type.

7.5.3.1 Request <CustPayeeTypeModRq>

The user must specify security details, the payee identifier, the payee type, and the new payee type.

Tag	Type	Usage	Description
<RqUID>	UUID	Required	Request Identifier. Sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header aggregate.
<AsyncRqUID>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously.
			For more information, see section 3.2.12.1.
<CustId>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<CustPayeeId>	Identifier	Required	Customer Payee Identifier. Assigned by the server at the time the Payee is first added. Cannot be modified by the client.
<PayeeType>	Closed Enum	Required	Payee Type. May be Standard Payee, Biller, Fully-Specified Payee or Interbank Transfer.
			Valid values: Biller, FSPayee, Xfer, Std

7.5.3.2 Response <CustPayeeTypeModRs>

Tag	Type	Usage	Description
<Status>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, <StatusCode> defaults to 0 (zero).
<RqUID>	UUID	Required Echoed	Request Identifier.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header aggregate.
<MsgRsHdr>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header aggregate.
<AsyncRqUID>	UUID	Optional Echoed	Asynchronous Request Identifier.

Tag	Type	Usage	Description
\<CustId\>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in \<SignonRq\>. For example, if a CSR or SP issues the request on behalf of the user, then \<CustId\> is required, and must contain the value of the user whose request is being issued.
\<CustPayeeId\>	Aggregate *see section 7.1.1.3*	Required Echoed	Payee Identifier.
\<PayeeType\>	Closed Enum	Required Echoed	Payee Type. May be Standard Payee, Biller, Fully-Specified Payee or Interbank Transfer. Valid values: Biller, FSPayee, Xfer, Std
\<CSPRefId\>	Identifier	Optional	Customer Service Provider Reference Identifier.
\<SPRefId\>	Identifier	Optional	Service Provider Reference Identifier.

7.5.4   *Customer Payee Delete*

A client uses the Customer Payee Delete message to delete a Payee from a Customer Payee list on the server. If the \<CascadeDel\> element is not set to *True*, an error must be returned if a payee delete is attempted when payments are still pending. If this error is returned, the types of dependent objects must be communicated to the client in the response message.

7.5.4.1   Request \<CustPayeeDelRq\>

A client initiates the Delete Payee message by sending a \<CustPayeeDelRq\> message to the server.

Tag	Type	Usage	Description
\<RqUID\>	UUID	Required	Request Identifier. Sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.
\<MsgRqHdr\>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header aggregate.
\<AsyncRqUID\>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously. For more information, see section 3.2.12.1.
\<CustId\>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in \<SignonRq\>. For example, if a CSR or SP issues the request on behalf of the user, then \<CustId\> is required, and must contain the value of the user whose request is being issued.
\<CustPayeeId\>	Identifier	Required	Customer's Payee Identifier. If the server supports Payee Lists, the server assigns \<CustPayeeId\> at the time the Payee is first added.
\<CascadeDel\>	Boolean	Optional	Cascade Delete. If *True*, server must delete all dependent objects when this object is deleted. If *False* or omitted, the customer payee must not be deleted if dependent objects exist. For a customer payee, a dependant objects are recurring models or payments.

7.5.4.2 Response <CustPayeeDelRs>

The server responds to the client's <CustPayeeDelRq> message by sending a <CustPayeeDelRs> message.

Tag	Type	Usage	Description
<Status>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, <StatusCode> defaults to 0 (zero).
<RqUID>	UUID	Required Echoed	Request Identifier.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header aggregate.
<MsgRsHdr>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header aggregate.
<AsyncRqUID>	UUID	Optional Echoed	Asynchronous Request Identifier.
<CustId>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<CustPayeeId>	Identifier	Required Echoed	Customer's Payee Identifier.
<CascadeDel>	Boolean	Optional Echoed	Cascade Delete.
<CustPayeeRec>	Aggregate *see section 7.3.2*	Optional XOR	Customer Payee Record.
<DependentType>	Open Enum	Optional Repeating XOR	Open Enum containing a list of dependent object types that exist for the customer payee. Defined values: RecPmt, Pmt
<CSPRefId>	Identifier	Optional	Customer Service Provider Reference Identifier.
<SPRefId>	Identifier	Optional	Service Provider Reference Identifier.

7.5.5 Customer Payee Inquiry

A client uses the Payee Inquiry message to obtain records from the Customer Payee list on a server. Clients that store a local copy of a Customer Payee list may use this message to "refresh" the Payee list.

7.5.5.1 Request <CustPayeeInqRq>

Tag	Type	Usage	Description
<RqUID>	UUID	Required	Request Identifier. Sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header aggregate.

Tag	Type	Usage	Description
<AsyncRqUID>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously.
			For more information, see section 3.2.12.1.
<CustId>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<RecCtrlIn>	Aggregate *see section 3.2.12.2.1*	Optional	Records Control Input Aggregate.
<DeliveryMethod>	Open Enum	Optional Profiled values	Delivery Method. Used to request the delivery channel for requested information. Included here primarily to allow customer using an IVR (interactive voice response) unit to request a delivery option for their Payee list.
			Defined values: Channel, HomeBank, Post, UPS, Courier.
			Default value is *Channel*
			Value must be supported in Service Profile.
<CustPayeeId>	Identifier	Optional Repeating	Customer's Payee Identifier. This field is used as a selection criterion. If the server supports Payee Lists, the server assigns <CustPayeeId> at the time the Payee is first added.
<PayeeType>	Closed Enum	Optional Repeating	Payee Type. May be Standard Payee, Biller, Fully-Specified Payee or Interbank Transfer.
			Valid values: Biller, FSPayee, Xfer, Std
			This field is used as a selection criterion.
<Name>	C-40	Optional Repeating	Payee Name.
			This field is used as a selection criterion.
<Nickname>	C-40	Optional Repeating	Payee Nickname. Optionally assigned by the customer. Only stored by the Pay provider if indicated in the Service Profile.
			This field is used as a selection criterion.
<IncToken>	Boolean	Optional	Include Token. If *True*, a <Token> must be included in the response, if the Service Profile indicates support for customer payee synchronization, to set a base for future synchronization messages. If *False* or omitted, the <Token> may be omitted in the response.

7.5.5.2   Response <CustPayeeInqRs>

Tag	Type	Usage	Description
<Status>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, <StatusCode> defaults to 0 (zero).
<RqUID>	UUID	Required Echoed	Request Identifier.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header aggregate.

Tag	Type	Usage	Description
<MsgRsHdr>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header aggregate.
<AsyncRqUID>	UUID	Optional Echoed	Asynchronous Request Identifier.
<CustId>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<RecCtrlOut>	Aggregate *see section 3.2.12.2.2*	Optional *but see Description*	Records Control Output Aggregate. Required if <RecCtrlIn> was provided in the request and the server supports Records Control.
<DeliveryMethod>	Open Enum	Optional Profiled values Echoed	Delivery Method.
<CustPayeeId>	Identifier	Optional Repeating Echoed	Customer's Payee Identifier.
<PayeeType>	Closed Enum	Optional Repeating Echoed	Payee Type.
<Name>	C-40	Optional Repeating Echoed	Payee Name.
<Nickname>	C-40	Optional Repeating Echoed	Payee Nickname.
<IncToken>	Boolean	Optional Echoed	Include Token.
<CustPayeeRec>	Aggregate *see section 7.3.2*	Optional Repeating	Payee List Record Aggregate. One record is returned for each Payee defined for this customer who meets the selection criteria in the request.
<Token>	Identifier	Optional *but see Description*	Token. Server Assigned. Client should make no assumptions about the value of this token relative to others that it may have received. The token is significant only to the server that originally assigned it. The token must be provided if <IncToken> = *True* in the request and the server supports synchronization. <Token> = 0 is returned if no records are returned within the response.

7.5.6   Customer Payee Audit

7.5.6.1   Request <CustPayeeAudRq>

Tag	Type	Usage	Description
<RqUID>	UUID	Required	Request Identifier. Sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.

Tag	Type	Usage	Description
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header aggregate.
<AsyncRqUID>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously. For more information, see section 3.2.12.1.
<CustId>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<RecCtrlIn>	Aggregate *see section 3.2.12.2.1*	Optional	Records Control Input Aggregate.
<SelRangeDt>	Aggregate *see section 3.2.10.1*	Optional	Selection Range Date Aggregate.
<Method>	Closed Enum	Optional Repeating	Payee Method. Valid values are: Add, Mod, Del This field is used as a selection criterion.
<CustPayeeId>	Identifier	Optional Repeating	Customer's Payee Identifier. This field is used as a selection criterion. If the server supports Payee Lists, the server assigns <CustPayeeId> at the time the Payee is first added.

7.5.6.2   Response <CustPayeeAudRs>

Tag	Type	Usage	Description
<Status>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, <StatusCode> defaults to 0 (zero).
<RqUID>	UUID	Required Echoed	Request Identifier.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header aggregate.
<MsgRsHdr>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header aggregate.
<AsyncRqUID>	UUID	Optional Echoed	Asynchronous Request Identifier.
<CustId>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.

Tag	Type	Usage	Description
<RecCtrlOut>	Aggregate *see section 3.2.12.2.2*	Optional *but see Description*	Message Records Control Output Aggregate. Required if <RecCtrlIn> was provided in the request and the server supports Records Control.
<SelRangeDt>	Aggregate *see section 3.2.10.1*	Optional Echoed	Selection Range Date Aggregate.
<Method>	Closed Enum	Optional Repeating Echoed	Payee Method.
<CustPayeeId>	Identifier	Optional Repeating Echoed	Customer's Payee Identifier. This field is used as a selection criterion. If the server supports Payee Lists, the server assigns <CustPayeeId> at the time the Payee is first added.
<CustPayeeMsgRec>	Aggregate	Optional Repeating	Payee Message Record Aggregate.
<CustId>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the entity who initiated the request (e.g., the one whose identifier appears in the <SignonRq> of the original issued request). For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the CSR or SP.
<MsgRecDt>	Timestamp	Optional	Audit Record Creation Date. Date when this audit record is created.
<CustPayeeAddRs>	Aggregate *see section 7.5.1.2*	Required XOR	Customer Payee Add Response Message Aggregate.
<CustPayeeModRs>	Aggregate *see section 7.5.2.2*	Required XOR	Customer Payee Modify Response Message Aggregate.
<CustPayeeTypeModRs>	Aggregate *see section 7.5.3.2*	Required XOR	Customer Payee Type Modify Response Message Aggregate.
<CustPayeeDelRs>	Aggregate *see section 7.5.4.2*	Required XOR	Customer Payee Delete Response Message Aggregate.
</CustPayeeMsgRec>			

7.5.7   Customer Payee Synchronization

The Customer Payee Synchronization message is used by a client for retrieving a list of changes to a customer's Payee list. This message may be used by clients that keep local copies of a customer's Payee messages to synchronize their local databases of Payee messages against the databases of Payee messages kept by Pay providers. Typically results of this message tells a client what the customer has done using other clients since they last used this one, but it may also inform a client about changes the Pay provider made (e.g., converted a Payee from a Fully-Specified Payee to a Biller).

7.5.7.1   Request <CustPayeeSyncRq>

Tag	Type	Usage	Description
<RqUID>	UUID	Required	Request Identifier. Sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.

Tag	Type	Usage	Description
\<MsgRqHdr\>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header aggregate.
\<AsyncRqUID\>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously. For more information, see section 3.2.12.1.
\<CustId\>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in \<SignonRq\>. For example, if a CSR or SP issues the request on behalf of the user, then \<CustId\> is required, and must contain the value of the user whose request is being issued.
\<RecCtrlIn\>	Aggregate *see section 3.2.12.2.1*	Optional	Records Control Input Aggregate.
\<Token\>	Identifier	Required	Token. Server assigned. Client should make no assumptions about the value of this token relative to others that it may have received. The token is only significant to the server that originally assigned it. This is a token that has been previously sent by the server, or zero for first time requests.

7.5.7.2   Response \<CustPayeeSyncRs\>

Tag	Type	Usage	Description
\<Status\>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, \<StatusCode\> defaults to 0 (zero).
\<RqUID\>	UUID	Required Echoed	Request Identifier.
\<MsgRqHdr\>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header aggregate.
\<MsgRsHdr\>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header aggregate.
\<AsyncRqUID\>	UUID	Optional Echoed	Asynchronous Request Identifier.
\<CustId\>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in \<SignonRq\>. For example, if a CSR or SP issues the request on behalf of the user, then \<CustId\> is required, and must contain the value of the user whose request is being issued.
\<RecCtrlOut\>	Aggregate *see section 3.2.12.2.2*	Optional *but see Description*	Message Records Control Output Aggregate. Required if \<RecCtrlIn\> was provided in the request and the server supports Records Control.
\<Token\>	Identifier	Required Echoed	Token.

Tag	Type	Usage	Description
\<NewToken\>	Identifier	Required	New Token. Server assigned. Client should make no assumptions about the value of this token relative to others that it may have received. The token is only significant to the server that originally assigned it. If the client sent a token in the request, the server returns a new token based on this message.
\<CustPayeeMsgRec\>	Aggregate	Optional Repeating	Customer Payee Message Record Aggregate.
\<CustId\>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the entity who initiated the request (e.g., the one whose identifier appears in the \<SignonRq\> of the original issued request). For example, if a CSR or SP issues the request on behalf of the user, then \<CustId\> is required, and must contain the value of the CSR or SP.
\<MsgRecDt\>	Timestamp	Optional	Audit Record Creation Date. Date when this audit record is created.
\<CustPayeeAddRs\>	Aggregate *see section 7.5.1.2*	Required XOR	Customer Payee Add Response Message Aggregate.
\<CustPayeeModRs\>	Aggregate *see section 7.5.2.2*	Required XOR	Customer Payee Modify Response Message Aggregate.
\<CustPayeeTypeModRs\>	Aggregate *see section 7.5.3.2*	Required XOR	Customer Payee Type Modify Response Message Aggregate.
\<CustPayeeDelRs\>	Aggregate *see section 7.5.4.2*	Required XOR	Customer Payee Delete Response Message Aggregate.
\</CustPayeeMsgRec\>			

7.6   Single Payment

7.6.1   *Payment Add*

The Payment Add message allows a client to schedule a single payment, where the amount is input by the customer or from a presented bill. The Payment Add message may reference an existing payee or add a new one, by specifying the information within \<RemitInfo\> \<PayeeInfo\>. If the Payment service provider supports \<CustPayeeId\>, the client must specify an existing \<CustPayeeId\> or include the \<CustPayeeInfo\> aggregate, but not both. Whether or not the Payment service provider supports \<CustPayeeId\>, the \<CustPayeeInfo\> aggregate may specify an existing standard payee or create a new fully specified or transfer payee. It is not possible to modify an existing payee within an Add Payment message. The customer may modify a payee via \<CustPayeeModRq\>.

7.6.1.1   Request \<PmtAddRq\>

Tag	Type	Usage	Description
\<RqUID\>	UUID	Required	Request Identifier. Sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.
\<MsgRqHdr\>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header aggregate.

Tag	Type	Usage	Description
<AsyncRqUID>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously. For more information, see section 3.2.12.1.
<CustId>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<PmtInfo>	Aggregate *see section 7.3.3.1*	Required	Payment Information Aggregate.
<DupChkOverride>	Boolean	Optional	Duplicate Check Override Flag. When set to *True*, requests that the server not perform duplicate checking if any is normally performed. The client is affirming that this is a new payment being added.

7.6.1.2   Response <PmtAddRs>

The <PmtAddRs> message is used to provide an acknowledgement to a customer-initiated <PmtAddRq>. It is also used in the Payment Audit Response <PmtAudRs> and Payment Synchronization Response <PmtSyncRs> to communicate to the client that payments have been added by the customer using <PmtAddRq> and by the Pay provider using the customer's Recurring Payment Models.

Tag	Type	Usage	Description
<Status>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, <StatusCode> defaults to 0 (zero).
<RqUID>	UUID	Required Echoed	Request Identifier.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header aggregate.
<MsgRsHdr>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header aggregate.
<AsyncRqUID>	UUID	Optional Echoed	Asynchronous Request Identifier.
<CustId>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<PmtInfo>	Aggregate *see section 7.3.3.1*	Required Echoed	Payment Information Aggregate.
<DupChkOverride>	Boolean	Optional Echoed	Duplicate Check Override Flag.

Tag	Type	Usage	Description
<PmtRec>	Aggregate *see section 7.3.3*	Required	Payment Record Aggregate.
<CSPRefId>	Identifier	Optional	Customer Service Provider Reference Identifier.
<SPRefId>	Identifier	Optional	Service Provider Reference Identifier.

7.6.2 Payment Modify

The Payment Modify message allows a client to modify the current information about a Payment that was set up using the Payment Add message or a Payment that was generated from a Recurring Payment Model. For information on Conventions for Modification of Server-Based Data, see Chapter 2. The Payment Modify message may neither add a new payee nor modify the definition of an existing payee. The client may change the previously defined payee, to whom the payment is intended, by using either the <CustPayeeId>, if the Payment service provider supports its use, or the <StdPayeeId> within <CustPayeeInfo> within <PmtInfo>.

7.6.2.1 Request <PmtModRq>

Tag	Type	Usage	Description
<RqUID>	UUID	Required	Request Identifier. Sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header aggregate.
<AsyncRqUID>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously. For more information, see section 3.2.12.1.
<CustId>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<PmtId>	Identifier	Required	Payment Identifier. Assigned by the server at the time the Payment is first added. Cannot be modified by the client.
<PmtInfo>	Aggregate *see section 7.3.3.1*	Required	Payment Information Aggregate. If the server supports Customer Payee lists, the client must specify the Payee using <CustPayeeId> within <PmtInfo>. Adding new Payees using the <PmtModRq> message is not supported. **Note:** *Values of <InvoiceInfo>s included in this aggregate must be interpreted as replacing all existing <InvoiceInfo>s associated with this payment.*

Tag	Type	Usage	Description
<UpPayee>	Boolean	Optional	Update Payee—Update Payee to current level. Used if a previous CustPayeeMod was done without propagating the changes to all pending payments, and the client would like the payee changes propagated to this particular payment. If *True*, the Payee information for this payment should be updated to match the current level of payee information known to the server. If *False*, or omitted, do not update the Payee information to the current level. Note that the rest of the <CustPayeeInfo> aggregate may still be used to update the Payee information for this payment, either within a <PmtModRq> or <CustPayeeModRq> as appropriate.
<DupChkOverride>	Boolean	Optional	Duplicate Check Override Flag. When set to *True*, requests that the server not perform duplicate checking if any is normally performed. The client is affirming that this is a distinct payment, even if the modification creates a duplicate of another payment.

7.6.2.2   Response <PmtModRs>

Tag	Type	Usage	Description
<Status>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, <StatusCode> defaults to 0 (zero).
<RqUID>	UUID	Required Echoed	Request Identifier.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header aggregate.
<MsgRsHdr>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header aggregate.
<AsyncRqUID>	UUID	Optional Echoed	Asynchronous Request Identifier.
<CustId>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<PmtId>	Identifier	Required Echoed	Payment Identifier.
<PmtInfo>	Aggregate *see section 7.3.3.1*	Required Echoed	Payment Information Aggregate.
<UpPayee>	Boolean	Optional Echoed	Update Payee Flag.
<DupChkOverride>	Boolean	Optional Echoed	Duplicate Check Override Flag.
<PmtRec>	Aggregate *see section 7.3.3*	Required	Payment Record Aggregate.
<CSPRefId>	Identifier	Optional	Customer Service Provider Reference Identifier.

Tag	Type	Usage	Description
<SPRefId>	Identifier	Optional	Service Provider Reference Identifier.

7.6.3   Payment Status Modify

A user with special authorization, typically a CSR, may send a request to the SP Server to modify the payment status of a specific payment instance. Customers may also send a request to the SP Server to modify the payment status in order to undo a Skip applied to one or more payments. The only fields that may be changed are within the payment status aggregate <PmtStatus>.

7.6.3.1   Request <PmtStatusModRq>

The user must specify the payment identifier along with the payment details to be changed.

Tag	Type	Usage	Description
<RqUID>	UUID	Required	Request Identifier. Sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header aggregate.
<AsyncRqUID>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously. For more information, see section 3.2.12.1.
<CustId>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<PmtId>	Identifier	Required	Payment Identifier. Assigned by the server at the time the Payment is first added. Cannot be modified by the client. Used to reference a specific payment.
<PmtRefId>	Identifier	Optional	Payment reference identifier. Identifier for the payment generated by the client. It can be the document generated by the Payer's Accounts Payable system, or it can be the identifier generated in the payment request (same as the <RqUId> of the <PmtAddRq>). ***Note:*** *If the CPP uses StatusModRq or InqRs to push acknowledgement to client, there is no place to echo back the <RqUID> from the <PmtAddRq>, so this <PmtRefId> is required for the customer to link the status back to the original payment request.*
<SvcRqUID>	UUID	Optional	Service Request Identifier. The payment service request identifier that was created and sent by the client to the CPP.
<PmtStatus>	Aggregate *see section 7.3.3.2*	Required	Payment Status Aggregate.

7.6.3.2   Response <PmtStatusModRs>

Tag	Type	Usage	Description

Tag	Type	Usage	Description
<Status>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, <StatusCode> defaults to 0 (zero).
<RqUID>	UUID	Required Echoed	Request Identifier.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header aggregate.
<MsgRsHdr>	Aggregate *see section 3.2.1.2*	Optional	Message Respoonse Header aggregate.
<AsyncRqUID>	UUID	Optional Echoed	Asynchronous Request Identifier.
<CustId>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<PmtId>	Identifier	Required Echoed	Payment Identifier.
<PmtRefId>	Identifier	Optional Echoed	Payment reference identifier.
<SvcRqUID>	UUID	Optional Echoed	Service Request Identifier.
<PmtStatus>	Aggregate *see section 7.3.3.2*	Required Echoed	Payment Status Aggregate.
<NetworkTrnInfo>	Aggregate *see section 3.2.18*	Optional	Network Transaction Information Aggregate. Contains information regarding the network that processed the transaction.
<CompositeContactInfo>	Aggregate *see section 1.1.1.1*	Optional	CPP contact information. In the case that contact info is needed, may be used in the case the URL is needed to link the customer back to the CPP site for additional information about the service or transaction.
<CSPRefId>	Identifier	Optional	Financial Institution Reference Number.
<SPRefId>	Identifier	Optional	Service Provider Reference Number.

7.6.4   Payment Status Advise

PmtStatusAdvise message is used to advise an entity of a change in Payment Status. For every status change in the lifecycle of a payment, a PmtStatusAdvise messages may be sent.

7.6.4.1   Request <PmtStatusAdviseRq>

Tag	Type	Usage	Description
<RqUID>	UUID	Required	Request Identifier. Sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header Aggregate.

\<CustId\>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in \<SignonRq\>. For example, if a CSR or SP issues the request on behalf of the user, then \<CustId\> is required, and must contain the value of the user whose request is being issued.
\<PmtStatusRec\>	Aggregate *see section 7.3.9*	Required	Payment Status Record Aggregate. Contains information regarding the status of the payment at the CPP or the Payment Network.

7.6.4.2   Response \<PmtStatusAdviseRs\>

Tag	Type	Usage	Description
\<Status\>	Aggregate *see section 3.2.12.1*	Required	Response Status Aggregate. If this aggregate is absent, \<StatusCode\> defaults to 0 (zero).
\<MsgRsHdr\>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header Aggregate
\<RqUID\>	UUID	Required Echoed	Request Identifier. Sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.
\<MsgRqHdr\>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header Aggregate.
\<CustId\>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in \<SignonRq\>. For example, if a CSR or SP issues the request on behalf of the user, then \<CustId\> is required, and must contain the value of the user whose request is being issued.
\<PmtStatusRec\>	Aggregate *see section 7.3.9*	Required Echoed	Payment Status Record Aggregate. Contains information regarding the status of the payment at the CPP or the Payment Network.
\<CSPRefId\>	Identifier	Optional	Customer Service Provider Reference Identifier. \<CSPRefId\> is used to inquire about a payment corresponding to a confirmation number that was returned to the client when the payment was added or modified. When a payment has been modified, only the \<CSPRefId\> received in the most recent \<PmtStatusAdviveRs\> is valid.
\<SPRefId\>	Identifier	Optional	Service Provider Reference Identifier. Same usage as \<CSPRefId\>.

7.6.5   *Payment Cancel*

The Cancel Payment message allows a client to cancel a Payment that was set up using the Payment Add message or generated from a Recurring Payment Model.

7.6.5.1   Request \<PmtCanRq\>

Tag	Type	Usage	Description
\<RqUID\>	UUID	Required	Request Identifier. Sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.

Tag	Type	Usage	Description
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header aggregate.
<AsyncRqUID>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously.
			For more information, see section 3.2.12.1.
<CustId>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<PmtId>	Identifier	Required	Payment Identifier. Assigned by the server at the time the Payment is first added. Cannot be modified by the client.

7.6.5.2   Response <PmtCanRs>

Tag	Type	Usage	Description
<Status>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, <StatusCode> defaults to 0 (zero).
<RqUID>	UUID	Required Echoed	Request Identifier. Sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.
			Server must return <RqUID> if provided by client in a <PmtCanRq>.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header aggregate.
<MsgRsHdr>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header aggregate.
<AsyncRqUID>	UUID	Optional Echoed	Asynchronous Request Identifier.
<CustId>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<PmtId>	Identifier	Required Echoed	Payment Identifier.
<PmtRec>	Aggregate *see section 7.3.3*	Optional	Payment Record Aggregate. This aggregate is provided in cases where the server keeps the payment's record on the server even after receiving a deletion request. This may occur if the deletion is not immediate, or if the server maintains the record with a status of Cancelled.
<CSPRefId>	Identifier	Optional	Customer Service Provider Reference Identifier.
<SPRefId>	Identifier	Optional	Service Provider Reference Identifier.

7.6.6 Payment Inquiry

The Payment Inquiry message allows a client to get a list of the customer's pending, completed, or failed payments on the CPP's server. The client may select on a number of criteria. Clients that keep local copies of a customer's payments may use this message to "refresh" their payment list. This message must not return a customer's recurring payment models but must return any pending, completed, or failed payments that were generated from recurring payment models.

The <FIDebitTrcNum> and <FICreditTrcNum> elements exist to support the case where the CPP submits a payment request to the FI (or CSP) that actually sends the payment messages. For example, a CPP may send a file of ACH-style messages that the FI forwards as actual ACH messages. The FI/CSP may edit the file; e.g., process their in-house messages and forward the rest. As part of this process the FI/CSP may assign reference numbers that correlate the messages to their system of record. There are both credit and debit fields to handle cases like "US on US" where two messages are spawned by the same payment request. Note that the debit and credit trace numbers are received offline but are added to support message histories. Also note that in this scenario all 4 message IDs are used: <SPRefId> is the Pay engine (CPP) reference to the payment and <CSPRefId> is the FI/CSP reference to the payment. One or both were probably returned to the client as confirmation number(s) when the payment was added or modified. <FIDebitTrcNum> and <FICreditTrcNum> are references to the messages that actually move funds to and from accounts.

7.6.6.1 Request <PmtInqRq>

A client initiates the Payment Inquiry message by sending a <PmtInqRq> message.

Tag	Type	Usage	Description
<RqUID>	UUID	Required	Request Identifier. Sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header aggregate.
<AsyncRqUID>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously. For more information, see section 3.2.12.1.
<CustId>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<RecCtrlIn>	Aggregate *see section 3.2.12.2.1*	Optional	Records Control Input Aggregate.
<PmtType>	Closed Enum	Optional Repeating	Payment Type. Valid values: Pmt, RecPmt. This field is used as a selection criterion.
<PmtId>	Identifier	Optional Repeating	Payment Identifier. Assigned by the server at the time the Payment is first added. Cannot be modified by the client. This field is used as a selection criterion.

Tag	Type	Usage	Description
<RecPmtId>	Identifier	Optional Repeating	Recurring Payment Model Identifier. Assigned by the server at the time the Recurring Payment Model is first added. Cannot be modified by the client.
			This field is used as a selection criterion.
<DepAcctIdFrom>	Aggregate *see section 3.2.7.1.2*	Optional XOR	Deposit Account Aggregate.
			Source account for payment. This field is used as a selection criterion.
<CardAcctIdFrom>	Aggregate *see section 3.2.7.1.3*	Optional XOR	Card Account Identifier Aggregate.
			Source account for payment. This field is used as a selection criterion.
<Category>	C-40	Optional Repeating	Payment Category. The customer assigns categories.
			This field is used as a selection criterion.
<Memo>	C-255	Optional Repeating	Memo for Payment. From Customer to Payee.
			This field is used as a selection criterion.
<ChkNum>	NC-12	Optional Repeating	Check Number. Assigned by the Pay provider or the CPP if payment is by paper check.
<ChkClrDt>	Date	Optional Repeating	Check Cleared Date. The date when the check cleared the account on which it was drawn.
<FIDebitTrcNum>	NC-7	Optional Repeating	FI Debit Trace Number.
<FICreditTrcNum>	NC-7	Optional Repeating	FI Credit Trace Number.
<PmtMethod>	Open Enum	Optional Repeating	Payment Method. Intended to provide the customer with information regarding the method of payment used by the service provider to transfer the funds to the payee.
			Defined values: Check, Electronic, ACH, FedNet, SWIFT, CHIPS, CHAPS, BookEntry, Draft
<CustPayeeId>	Identifier	Optional Repeating	Customer's Payee Identifier. This field is used as a selection criterion.
<BillingAcct>	C-32	Optional Repeating	Customer Account Number with Payee.
			This field is used as a selection criterion.
<Name>	C-40	Optional Repeating	Payee Name.
			This field is used as a selection criterion.
<StdPayeeId>	Aggregate *see section 7.3.1.1*	Optional Repeating	Customer's Payee Identifier Aggregate. <StdPayeeId> is a synonym for <BillerId>.
			This field is used as a selection criterion.
<Nickname>	C-40	Optional Repeating	Payee Nickname.
			This field is used as a selection criterion.
<PmtStatusCode>	Closed Enum	Optional Repeating	Payment Status Code. This identifies the payment processing status.
			Valid values: Scheduled, Processed, FundAcctDebited, Skip, Cancelled, Failed, PmtAuthHeld, PmtAuthNoFunds, PmtAuthInactive, PmtAuthClosed, Posted, CheckCleared, Returned, RemitPending, RemitRefused, RemitRejected
			This field is used as a selection criterion.

Tag	Type	Usage	Description
<SelRangeDueDt>	Aggregate *see section 3.2.10.1*	Optional	Selection Range Due Date Aggregate.
<SelRangePrcDt>	Aggregate *see section 3.2.10.1*	Optional	Selection Range Processing Date Aggregate.
			Usage is range for actual processing date, possibly adjusted for holidays and non-processing days—as opposed to original customer-entered processing date if using the processing date model.
<SelRangeCurAmt>	Aggregate *see section 3.2.10.2*	Optional	Selection Range Amount Aggregate.
<DeliveryMethod>	Open Enum	Optional Profiled values	Delivery Method. Used to request the delivery channel for requested information. Included here primarily to allow customer using an IVR (interactive voice response) unit to request a delivery option for their payment list.
			Valid values: Channel, HomeBank, Post, UPS, Courier. Default value is Channel
			Value must be supported in Service Profile.
<CSPRefId>	Identifier	Optional Repeating	Customer Service Provider Reference Identifier. <CSPRefId> is used to inquire about a payment corresponding to a confirmation number that was returned to the client when the payment was added or modified. When a payment has been modified, only the <CSPRefId> received in the most recent PMPMODRS is valid. The use of an <CSPRefId> from an earlier response is likely to result in a "payment not found" response.
<SPRefId>	Identifier	Optional Repeating	Service Provider Reference Identifier. Same usage as <CSPRefId>.
<IncToken>	Boolean	Optional	Include Token. If *True*, a <Token> must be included in the response, if the Service Profile indicates support for payment synchronization, to set a base for future synchronization messages. If *False* or omitted, the <Token> may be omitted in the response.
<IncHistory>	Boolean	Optional	Include History. If *True*, the response should include payments that have already occurred, as well as those scheduled to occur.

7.6.6.2   Response <PmtInqRs>

The server responds to a <PmtInqRq> message by returning a <PmtInqRs> message to the client.

Tag	Type	Usage	Description
<Status>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, <StatusCode> defaults to 0 (zero).
<RqUID>	UUID	Required Echoed	Request Identifier.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header aggregate.
<MsgRsHdr>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header aggregate.

Tag	Type	Usage	Description
<AsyncRqUID>	UUID	Optional Echoed	Asynchronous Request Identifier.
<CustId>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<RecCtrlOut>	Aggregate *see section 3.2.12.2.2*	Optional *but see Description*	Records Control Output Aggregate. Required if <RecCtrlIn> was provided in the request and the server supports Records Control.
<PmtType>	Closed Enum	Optional Repeating Echoed	Payment Type. This field is used as a selection criterion.
<PmtId>	Identifier	Optional Repeating Echoed	Payment Identifier. Assigned by the server at the time the Payment is first added. Cannot be modified by the client. This field is used as a selection criterion.
<RecPmtId>	Identifier	Optional Repeating Echoed	Recurring Payment Model Identifier. Assigned by the server at the time the Recurring Payment Model is first added. Cannot be modified by the client. This field is used as a selection criterion.
<DepAcctIdFrom>	Aggregate *see section 3.2.7.1.2*	Optional XOR Echoed	Deposit Account Aggregate. Source account for payment. This field is used as a selection criterion.
<CardAcctIdFrom>	Aggregate *see section 3.2.7.1.3*	Optional XOR Echoed	Card Account Identifier Aggregate. Source account for payment. This field is used as a selection criterion.
<Category>	C-40	Optional Repeating Echoed	Payment Category. The customer assigns categories. This field is used as a selection criterion.
<Memo>	C-255	Optional Repeating Echoed	Memo for Payment. From Customer to Payee. This field is used as a selection criterion.
<ChkNum>	NC-12	Optional Repeating Echoed	Check Number. Assigned by the Pay provider or the CPP if payment is by paper check.
<ChkClrDt>	Date	Optional Repeating Echoed	Check Cleared Date. The date when the check cleared the account on which it was drawn.
<FIDebitTrcNum>	NC-7	Optional Repeating Echoed	FI Debit Trace Number.
<FICreditTrcNum>	NC-7	Optional Repeating Echoed	FI Credit Trace Number.
<PmtMethod>	Open Enum	Optional Repeating Echoed	Payment Method. Intended to provide the customer with information regarding the method of payment used by the service provider to transfer the funds to the payee.
<CustPayeeId>	Identifier	Optional Repeating Echoed	Customer's Payee Identifier. This field is used as a selection criterion.

Tag	Type	Usage	Description
\<BillingAcct\>	C-32	Optional Repeating Echoed	Customer Account Number with Payee. This field is used as a selection criterion.
\<Name\>	C-40	Optional Repeating Echoed	Payee Name.
\<StdPayeeId\>	Aggregate *see section 7.3.1.1*	Optional Repeating Echoed	Customer's Payee Identifier Aggregate. \<StdPayeeId\> is a synonym for \<BillerId\>.
\<Nickname\>	C-40	Optional Repeating Echoed	Payee Nickname.
\<PmtStatusCode\>	Closed Enum	Optional Repeating Echoed	Payment Status Code.
\<SelRangeDueDt\>	Aggregate *see section 3.2.10.1*	Optional Echoed	Selection Range Due Date Aggregate.
\<SelRangePrcDt\>	Aggregate *see section 3.2.10.1*	Optional Echoed	Selection Range Processing Date Aggregate.
\<SelRangeCurAmt\>	Aggregate *see section 3.2.10.2*	Optional Echoed	Selection Range Amount Aggregate.
\<DeliveryMethod\>	Open Enum	Optional Echoed	Delivery Method.
\<CSPRefId\>	Identifier	Optional Repeating Echoed	Customer Service Provider Reference Identifier.
\<SPRefId\>	Identifier	Optional Repeating Echoed	Service Provider Reference Identifier.
\<IncToken\>	Boolean	Optional Echoed	Include Token.
\<IncHistory\>	Boolean	Optional Echoed	Include History.
\<PmtRec\>	Aggregate *see section 7.3.3*	Optional Repeating	Payment Record Aggregate. One record is returned for each of the customer's payments that meets the selection criteria specified in the request message. Note that payments may have been generated by a client (using \<PmtAddRq\> or may have been generated by the Pay provider from one of the customer's Recurring Payment Models.
\<Token\>	Identifier	Optional *but see Description*	Token. Server Assigned. Client should make no assumptions about the value of this token relative to others that it may have received. The token is significant only to the server that originally assigned it. The token is only included if \<IncToken\>=*True* in the request. \<Token\>=0 is returned if no records are returned within the response.

7.6.7 *Payment Reversal*

7.6.7.1 Request <PmtRevRq>

Tag	Type	Usage	Description
<RqUID>	UUID	Required	Request Identifier.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header aggregate.
<AsyncRqUID>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously.
			For more information, see section **3.2.12.1**.
<CustId>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<RevReasonCode>	Open Enum	Required	Reversal reason code. The reason why the client is requesting a reversal.
<Desc>	C-80	Optional	Description, text describing the reason for the reversal.
<PmtMsgRqInfo>	Aggregate	Required	Payment Message Request Information Aggregate. This aggregate contains the message to reverse. See RevOptSupt in the service profile for specific reversal messages supported.
<PmtAddRq>	Aggregate *see section 7.6.1.1*	Required XOR	Payment Add Request Message Aggregate.
<PmtModRq>	Aggregate *see section 7.6.2.1*	Required XOR	Payment Modify Request Message Aggregate.
<PmtInqRq>	Aggregate *see section 7.6.6.1*	Required XOR	Payment Inquiry Request Message Aggregate.
</PmtMsgRqInfo>			

7.6.7.2 Response <PmtRevRs>

Tag	Type	Usage	Description
<Status>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, <StatusCode> defaults to 0 (zero).
<RqUID>	UUID	Required Echoed	Request Identifier.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header aggregate.
<MsgRsHdr>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header aggregate.

Tag	Type	Usage	Description
<AsyncRqUID>	UUID	Optional Echoed	Asynchronous Request Identifier.
<CustId>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Identification Aggregate.
<RevReasonCode>	Open Enum	Required Echoed	Reversal reason code. The reason why the client is requesting a reversal.
<Desc>	C-80	Optional Echoed	Description, text describing the reason for the reversal.
<PmtMsgRqInfo>	Aggregate	Required Echoed	Payment Message Request Information Aggregate. This aggregate contains the message to reverse. See RevOptSupt in the service profile for specific reversal messages supported.
<PmtAddRq>	Aggregate *see section 7.6.1.1*	Required Echoed XOR	Payment Add Request Message Aggregate.
<PmtModRq>	Aggregate *see section 7.6.2.1*	Required Echoed XOR	Payment Add Request Message Aggregate.
<PmtInqRq>	Aggregate *see section 7.6.6.1*	Required Echoed XOR	Payment Add Request Message Aggregate.
</PmtMsgRqInfo>			
<CSPRefId>	Identifier	Optional	Customer Service Provider Reference Identifier.
<SPRefId>	Identifier	Optional	Service Provider Reference Identifier.

7.6.8   Payment Status Inquiry

7.6.8.1   Request <PmtStatusInqRq>

A client initiates the Payment Status Inquiry message by sending a <PmtStatusInqRq> message.

Tag	Type	Usage	Description
<RqUID>	UUID	Required	Request Identifier. Sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header aggregate.
<AsyncRqUID>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously. For more information, see section 3.2.12.1.
<CustId>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.

Tag	Type	Usage	Description
\<RecCtrlIn\>	Aggregate *see section 3.2.12.2.1*	Optional	Records Control Input Aggregate.
\<SettlementMethod\>	Open Enum	Optional Repeating	Settlement Method. Either the method accepted by the Biller or BPP for settling payments, or the payment/settlement method, provided as the general method for settling payment.
			Defined values: RPS, EPay, ACH, Concentrator, FedNet, SWIFT, CHIPS, CHAPS, BookEntry, Draft, OutsourcedCheck
			This field is used as a selection criterion.
\<PmtId\>	Identifier	Optional Repeating	Payment Identifier. Assigned by the server at the time the Payment is first added. Cannot be modified by the client.
			This field is used as a selection criterion.
\<PmtRefId\>	Identifier	Optional Repeating	Payment reference identifier. Identifier for the payment generated by the client. It can be the document generated by the Payer's Accounts Payable system, or it can be the identifier generated in the payment request (same as the \<RqUId\> of the \<PmtAddRq\>).
			Note: *If CPP uses the AddRq or InqRq message to push acknowledgement to client, there is no place to echo back the \<RqUID\> from the \<PmtAddRq\>, so this \<PmtRefId\> is required for client to link the status back to the original payment request.*
\<SvcRqUID\>	UUID	Optional	Service Request Identifier. The payment service request identifier that was created and sent by the client to the CPP.
\<PmtRemitRefId\>	Identifier	Optional	Payment Remittance (Credit) Reference Transaction Identifer.
\<DepAcctIdFrom\>	Aggregate *see section 3.2.7.1.2*	Optional XOR	Deposit Account Aggregate.
			Source account for payment. This field is used as a selection criterion.
\<CardAcctIdFrom\>	Aggregate *see section 3.2.7.1.3*	Optional XOR	Card Account Identifier Aggregate.
			Source account for payment. This field is used as a selection criterion.
\<Category\>	C-40	Optional Repeating	Payment Category. The customer assigns categories.
			This field is used as a selection criterion.
\<ChkNum\>	NC-12	Optional Repeating	Check Number. Assigned by the Pay provider or the CPP if payment is by paper check.
\<CustPayeeId\>	Identifier	Optional Repeating XOR	Customer's Payee Identifier. This field is used as a selection criterion.
\<FSPayee\>	Aggregate *see section 7.3.2.1.1*	Optional Repeating XOR	Fully-Specified Payee Aggregate.
\<Name\>	C-40	Optional Repeating	Payee Name.
			This field is used as a selection criterion.
\<PayerInfo\>	Aggregate *see section 7.3.3.1.1*	Optional Repeating	Payer information. This field is used as a selection criterion.

Tag	Type	Usage	Description
\<PmtStatusCode\>	Closed Enum	Optional Repeating	Payment Status Code. This identifies the payment processing status.
			Valid values: Scheduled, Processed, FundAcctDebited, Skip, Cancelled, Failed, PmtAuthHeld, PmtAuthNoFunds, PmtAuthInactive, PmtAuthClosed, Posted, CheckCleared, Returned, RemitPending, RemitRefused, RemitRejected
			This field is used as a selection criterion.
\<SelRangeDueDt\>	Aggregate *see section 3.2.10.1*	Optional	Selection Range Due Date Aggregate.

7.6.8.2   Response \<PmtStatusInqRs\>

The server respondes to a Payment Status Inquiry Request \<PmtStatusInqRq\> message by sending a Payment Status Inquiry Response \<PmtStatusInqRs\> message with information aabout the \<PmtAddRq\>.

Tag	Type	Usage	Description
\<Status\>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, \<StatusCode\> defaults to 0 (zero).
\<RqUID\>	UUID	Required Echoed	Request Identifier. Sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.
\<MsgRqHdr\>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header aggregate.
\<MsgRsHdr\>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header aggregate.
\<AsyncRqUID\>	UUID	Optional Echoed	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously.
			For more information, see section 3.2.12.1.
\<CustId\>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in \<SignonRq\>. For example, if a CSR or SP issues the request on behalf of the user, then \<CustId\> is required, and must contain the value of the user whose request is being issued.
\<RecCtrlIn\>	Aggregate *see section 3.2.12.2.1*	Optional Echoed	Records Control Input Aggregate.
\<SettlementMethod\>	Open Enum	Optional Repeating Echoed	Settlement Method. Either the method accepted by the Biller or BPP for settling payments, or the payment/settlement method, provided as the general method for settling payment.
			Defined values: RPS, EPay, ACH, Concentrator, FedNet, SWIFT, CHIPS, CHAPS, BookEntry, Draft, OutsourcedCheck

Tag	Type	Usage	Description
<PmtId>	Identifier	Optional Repeating Echoed	Payment Identifier. Assigned by the server at the time the Payment is first added. Cannot be modified by the client.
			This field is used as a selection criterion.
<PmtRefId>	Identifier	Optional Repeating Echoed	Payment reference identifier. Identifier for the payment generated by the client. It can be the document generated by the Payer's Accounts Payable system, or it can be the identifier generated in the payment request (same as the <RqUId> of the <PmtAddRq>).
			Note: *If CPP uses the AddRq or InqRq message to push acknowledgement to client, there is no place to echo back the <RqUID> from the <PmtAddRq>, so this <PmtRefId> is required for client to link the status back to the original payment request.*
<SvcRqUID>	UUID	Optional Echoed	Service Request Identifier. The payment service request identifier that was created and sent by the client to the CPP.
<PmtRemitRefId>	Identifier	Optional Echoed	Payment Remittance (Credit) Reference Transaction Identifer.
<DepAcctIdFrom>	Aggregate *see section 3.2.7.1.2*	Optional XOR Echoed	Deposit Account Aggregate.
			Source account for payment. This field is used as a selection criterion.
<CardAcctIdFrom>	Aggregate *see section 3.2.7.1.3*	Optional XOR Echoed	Card Account Identifier Aggregate.
			Source account for payment. This field is used as a selection criterion.
<Category>	C-40	Optional Repeating Echoed	Payment Category. The customer assigns categories.
			This field is used as a selection criterion.
<ChkNum>	NC-12	Optional Repeating Echoed	Check Number. Assigned by the Pay provider or the CPP if payment is by paper check.
<CustPayeeId>	Identifier	Optional Repeating XOR Echoed	Customer's Payee Identifier. This field is used as a selection criterion.
<FSPayee>	Aggregate *see section 7.3.2.1.1*	Optional Repeating XOR Echoed	Fully-Specified Payee Aggregate.
<Name>	C-40	Optional Repeating Echoed	Payee Name.
			This field is used as a selection criterion.
<PayerInfo>	Aggregate *see section 7.3.3.1.1*	Optional Repeating Echoed	Payer Information. This field is used as a selection criterion.
<PmtStatusCode>	Closed Enum	Optional Repeating Echoed	Payment Status Code. This identifies the payment processing status.
			Valid values: Scheduled, Processed, FundAcctDebited, Skip, Cancelled, Failed, PmtAuthHeld, PmtAuthNoFunds, PmtAuthInactive, PmtAuthClosed, Posted, CheckCleared, Returned, RemitPending, RemitRefused, RemitRejected
			This field is used as a selection criterion.

Tag	Type	Usage	Description
<SelRangeDueDt>	Aggregate *see section 3.2.10.1*	Optional Echoed	Selection Range Due Date Aggregate.
<PmtStatusRec>	Aggregate *see section 7.3.9*	Optional Repeating	Payment Status Record Aggregate. Contains information regarding the status of the payment at the CPP or the Payment Network.
<CSPRefId>	Identifier	Optional Repeating	Customer Service Provider Reference Identifier. <CSPRefId> is used to inquire about a payment corresponding to a confirmation number that was returned to the client when the payment was added or modified. When a payment has been modified, only the <CSPRefId> received in the most recent <PmtModRs> is valid. The use of an <CSPRefId> from an earlier response is likely to result in a "payment not found" response.
<SPRefId>	Identifier	Optional Repeating	Service Provider Reference Identifier. Same usage as <CSPRefId>.

7.6.9   Payment Audit

7.6.9.1   Request <PmtAudRq>

Tag	Type	Usage	Description
<RqUID>	UUID	Required	Request Identifier. Sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header aggregate.
<AsyncRqUID>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously. For more information, see section 3.2.12.1.
<CustId>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<RecCtrlIn>	Aggregate *see section 3.2.12.2.1*	Optional	Records Control Input Aggregate.
<SelRangeDt>	Aggregate *see section 3.2.10.1*	Optional	Selection Range Date Aggregate.
<Method>	Closed Enum	Optional Repeating	Payment Method. Valid Values: Add, Mod, Can This field is used as a selection criterion.
<PmtId>	Identifier	Optional Repeating	Payment Identifier. Assigned by the server at the time the Payment is first added. Cannot be modified by the client. This field is used as a selection criterion.

7.6.9.2   Response <PmtAudRs>

Tag	Type	Usage	Description
<Status>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, <StatusCode> defaults to 0 (zero).
<RqUID>	UUID	Required Echoed	Request Identifier.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header aggregate.
<MsgRsHdr>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header aggregate.
<AsyncRqUID>	UUID	Optional Echoed	Asynchronous Request Identifier.
<CustId>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<RecCtrlOut>	Aggregate *see section 3.2.12.2.2*	Optional *but see Description*	Records Control Output Aggregate. Required if <RecCtrlIn> was provided in the request and the server supports Records Control.
<SelRangeDt>	Aggregate *see section 3.2.10.1*	Optional Echoed	Selection Range Date Aggregate.
<Method>	Closed Enum	Optional Repeating Echoed	Payment Method.
<PmtId>	Identifier	Optional Echoed	Payment Identifier. Assigned by the server at the time the Payment is first added. Cannot be modified by the client. This field is used as a selection criterion.
<PmtMsgRec>	Aggregate	Optional Repeating	Payment Message Record Aggregate.
<CustId>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the entity who initiated the request (e.g., the one whose identifier appears in the <SignonRq> of the original issued request). For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the CSR or SP.
<MsgRecDt>	Timestamp	Optional	Audit Record Creation Date. Date when this audit record is created.
<PmtAddRs>	Aggregate *see section 7.6.1.2*	Required XOR	Payment Add Response Message Aggregate.
<PmtModRs>	Aggregate *see section 7.6.2.2*	Required XOR	Payment Modify Response Message Aggregate.
<PmtStatusModRs>	Aggregate *see section 7.6.3.2*	Required XOR	Payment Status Modify Response Message Aggregate.

Tag	Type	Usage	Description
<PmtCanRs>	Aggregate *see section 7.6.5.2*	Required XOR	Payment Cancel Response Message Aggregate.
<PmtRevRs>	Aggregate *see section 7.6.7.2*	Required XOR	Payment Reversal Response Message Aggregate.
</PmtMsgRec>			

7.6.10 Payment Synchronization

A client uses the Payment Sync message to retrieve a list of changes that have occurred to a customer's payments. This message may be used to enable a client that keeps local copies of a customer's payment messages to synchronize its local database with the Pay provider. The results of this message tell a client what the customer has done using other clients since they last used this one. It may also inform a client about changes the Pay provider made (i.e., generated another payment instance for a recurring payment model, completed a payment or rejected a payment).

Some examples of changes the Pay provider would communicate to a customer include:

- New pending payment generated from a Recurring Payment Model—returned with a <PmtAddRs>;
- Successful payment processing—returned with <PmtModRs> with new <PmtStatus> indicating success;
- Unsuccessful payment processing—returned with <PmtModRs> with new <PmtStatus> indicating status and reason for failure, if available;
- Payment assigned a check number—returned with <PmtModRs> with new <PmtStatus> including <ChkNum>; and
- Payment check cleared—returned with <PmtModRs> with new <PmtStatus> indicating status.

7.6.10.1 Request <PmtSyncRq>

Tag	Type	Usage	Description
<RqUID>	UUID	Required	Request Identifier. Sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header aggregate.
<AsyncRqUID>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously.
			For more information, see section 3.2.12.1.
<CustId>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<RecCtrlIn>	Aggregate *see section 3.2.12.2.1*	Optional	Records Control Input Aggregate.

Tag	Type	Usage	Description
<Token>	Identifier	Required	Token. Server assigned. Client should make no assumptions about the value of this token relative to others that it may have received. The token is only significant to the server that originally assigned it. This is a token that has been previously sent by the server, or zero for first time requests.

7.6.10.2   Response <PmtSyncRs>

Tag	Type	Usage	Description
<Status>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, <StatusCode> defaults to 0 (zero).
<RqUID>	UUID	Required Echoed	Request Identifier.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header aggregate.
<MsgRsHdr>	Aggregate *see section 3.2.1.2*	Optional	Message Rewsponse Header aggregate.
<AsyncRqUID>	UUID	Optional Echoed	Asynchronous Request Identifier.
<CustId>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<RecCtrlOut>	Aggregate *see section 3.2.12.2.2*	Optional	Records Control Output Aggregate.
<Token>	Identifier	Required Echoed	Token. Server assigned. Client should make no assumptions about the value of this token relative to others that it may have received. The token is only significant to the server that originally assigned it. This is a token that has been previously sent by the server, or zero for first time requests.
<NewToken>	Identifier	Required	New Token. Server assigned. Client should make no assumptions about the value of this token relative to others that it may have received. The token is only significant to the server that originally assigned it.
<PmtMsgRec>	Aggregate	Optional Repeating	Payment Message Record Aggregate.
<CustId>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the entity who initiated the request (e.g., the one whose identifier appears in the <SignonRq> of the original issued request). For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the CSR or SP.
<MsgRecDt>	Timestamp	Optional	Audit Record Creation Date. Date when this audit record is created.

Tag	Type	Usage	Description
<PmtAddRs>	Aggregate *see section 7.6.1.2*	Required XOR	Payment Add Response Message Aggregate.
<PmtModRs>	Aggregate *see section 7.6.2.2*	Required XOR	Payment Modify Response Message Aggregate.
<PmtStatusModRs>	Aggregate *see section 7.6.3.2*	Required XOR	Payment Status Modify Response Message Aggregate.
<PmtCanRs>	Aggregate *see section 7.6.5.2*	Required XOR	Payment Cancel Response Message Aggregate.
</PmtMsgRec>			

7.7 Payment Authorization

The Payment Authorization transactions are used by a CPP to request authorization for payment from the holder of the funding account, which could be a different organization than the CPP or CSP. The <PmtAuthAdd> would generally be done at the time of payment processing for current day payments. This message asks the Financial Institution for a decision on whether or not the CPP is authorized to make a particular payment. The <PmtAuth> messages facilitate what is commonly called the "Good Funds" model.

7.7.1 Payment Authorization Add

7.7.1.1 Request <PmtAuthAddRq>

Tag	Type	Usage	Description
<RqUID>	UUID	Required	Request Identifier. Sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header aggregate.
<AsyncRqUID>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously.
			For more information, see section 3.2.12.1.
<CustId>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<PmtAuthInfo>	Aggregate *see section 7.3.4.1*	Required	Payment Authorization Information Aggregate.

7.7.1.2 Response <PmtAuthAddRs>

Tag	Type	Usage	Description

Tag	Type	Usage	Description
<Status>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, <StatusCode> defaults to 0 (zero).
<RqUID>	UUID	Required Echoed	Request Identifier. Sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header aggregate.
<MsgRsHdr>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header aggregate.
<AsyncRqUID>	UUID	Optional Echoed	Asynchronous Request Identifier.
<CustId>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<PmtAuthInfo>	Aggregate *see section 7.3.4.1*	Required Echoed	Payment Authorization Information Aggregate.
<PmtAuthRec>	Aggregate *see section 7.3.4*	Required	Payment Authorization Record Aggregate

7.7.2   Payment Authorization Modify

7.7.2.1   Request <PmtAuthModRq>

Tag	Type	Usage	Description
<RqUID>	UUID	Required	Request Identifier. Sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header aggregate.
<AsyncRqUID>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously. For more information, see section 3.2.12.1.
<CustId>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<PmtAuthId>	Identifier	Required	Payment Authorization Identifier.

Tag	Type	Usage	Description
<PmtAuthInfo>	Aggregate *see section 7.3.4.1*	Required	Payment Authorization Information Aggregate.

7.7.2.2   Response <PmtAuthModRs>

Tag	Type	Usage	Description
<Status>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, <StatusCode> defaults to 0 (zero).
<RqUID>	UUID	Required Echoed	Request Identifier. Sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header aggregate.
<MsgRsHdr>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header aggregate.
<AsyncRqUID>	UUID	Optional Echoed	Asynchronous Request Identifier.
<CustId>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<PmtAuthId>	Identifier	Required Echoed	Payment Authorization Identifier
<PmtAuthInfo>	Aggregate *see section 7.3.4.1*	Required Echoed	Payment Authorization Information Aggregate.
<PmtAuthRec>	Aggregate *see section 7.3.4*	Required	Payment Authorization Record Aggregate

7.7.3   *Payment Authorization Cancel*

7.7.3.1   Request <PmtAuthCanRq>

Tag	Type	Usage	Description
<RqUID>	UUID	Required	Request Identifier. Sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header aggregate.
<AsyncRqUID>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously. For more information, see section 3.2.12.1.

Tag	Type	Usage	Description
<CustId>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<PmtAuthId>	Identifier	Required	Payment Authorization Identifier.

7.7.3.2   Response <PmtAuthCanRs>

Tag	Type	Usage	Description
<Status>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, <StatusCode> defaults to 0 (zero).
<RqUID>	UUID	Required Echoed	Request Identifier. Sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header aggregate.
<MsgRsHdr>	Aggregate *see section 3.2.1.2*	Optional	Message Respone Header aggregate.
<AsyncRqUID>	UUID	Optional Echoed	Asynchronous Request Identifier.
<CustId>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<PmtAuthId>	Identifier	Required Echoed	Payment Authorization Identifier.
<PmtAuthRec>	Aggregate *see section 7.3.4*	Optional	Payment Authorization Record. This aggregate is provided in cases where the server keeps the payment authorization's record on the server even after receiving a deletion request. This may occur if the deletion is not immediate, or if the server maintains the record with a status of Cancelled.

7.7.4   *Payment Authorization Inquiry*

7.7.4.1   Request <PmtAuthInqRq>

Tag	Type	Usage	Description
<RqUID>	UUID	Required	Request Identifier. Sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header aggregate.

Tag	Type	Usage	Description
<AsyncRqUID>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously. For more information, see section 3.2.12.1.
<CustId>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<RecCtrlIn>	Aggregate *see section 3.2.12.2.1*	Optional	Records Control Input Aggregate
<PmtAuthId>	Identifier	Optional Repeating	Payment Authorization Identifier. Used as a selection criterion.
<SelRangeCurAmt>	Aggregate *see section 3.2.10.2*	Optional	Selection Range Currency Amount. Used as a selection criterion.
<SelRangeDt>	Aggregate *see section 3.2.10.1*	Optional	Date Selection Range. Used as a selection criterion.
<DeliveryMethod>	Open Enum	Optional Profiled values	Delivery Method. Default is Channel. Value must be supported in Service Profile.

7.7.4.2   Response <PmtAuthInqRs>

Tag	Type	Usage	Description
<Status>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, <StatusCode> defaults to 0 (zero).
<RqUID>	UUID	Required Echoed	Request Identifier. Sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header aggregate.
<MsgRsHdr>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header aggregate.
<AsyncRqUID>	UUID	Optional Echoed	Asynchronous Request Identifier.
<CustId>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<RecCtrlOut>	Aggregate *see section 3.2.12.2.2*	Optional	Records Control Output Aggregate.

Tag	Type	Usage	Description
<PmtAuthId>	Identifier	Optional Repeating Echoed	Payment Authorization Identifier.
<SelRangeCurAmt>	Aggregate *see section 3.2.10.2*	Optional Echoed	Selection Range Currency Amount. Used as a selection criterion.
<SelRangeDt>	Aggregate *see section 3.2.10.1*	Optional Echoed	Date Selection Range. Used as a selection criterion.
<DeliveryMethod>	Open Enum	Optional Echoed	Delivery Method.
<PmtAuthRec>	Aggregate *see section 7.3.4*	Optional Repeating	Payment Authorization Record. One aggregate for each record matching the selection criteria in the request.

7.7.5   Payment Authorization Audit

7.7.5.1   Request <PmtAuthAudRq>

Tag	Type	Usage	Description
<RqUID>	UUID	Required	Request Identifier. Sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header aggregate.
<AsyncRqUID>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously.
			For more information, see section 3.2.12.1.
<CustId>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<RecCtrlIn>	Aggregate *see section 3.2.12.2.1*	Optional	Records Control Input Aggregate.
<SelRangeDt>	Aggregate *see section 3.2.10.1*	Optional	Selection Range Date Aggregate.
<Method>	Closed Enum	Optional Repeating	Payment Action.
			Valid values: Add, Mod, Can
			This field is used as a selection criterion.
<PmtAuthId>	Identifier	Optional Repeating	Payment Authorization Identifier. Assigned by the server at the time the Payment Authorization is first added. Cannot be modified by the client.
			This field is used as a selection criterion.

7.7.5.2 Response <PmtAuthAudRs>

Tag	Type	Usage	Description
<Status>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, <StatusCode> defaults to 0 (zero).
<RqUID>	UUID	Required Echoed	Request Identifier.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header aggregate.
<MsgRsHdr>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header aggregate.
<AsyncRqUID>	UUID	Optional Echoed	Asynchronous Request Identifier.
<CustId>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<RecCtrlOut>	Aggregate *see section 3.2.12.2.2*	Optional *but see Description*	Records Control Output Aggregate. Required if <RecCtrlIn> was provided in the request and the server supports Records Control.
<SelRangeDt>	Aggregate *see section 3.2.10.1*	Optional Echoed	Selection Range Date Aggregate.
<Method>	Closed Enum	Optional Repeating Echoed	Payment Method.
<PmtAuthId>	Identifier	Optional Repeating Echoed	Payment Authorization Identifier. Assigned by the server at the time the Payment Authorization is first added. Cannot be modified by the client. This field is used as a selection criterion.
<PmtAuthMsgRec>	Aggregate	Optional Repeating	Payment Authorization Message Record Aggregate.
<CustId>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the entity who initiated the request (e.g., the one whose identifier appears in the <SignonRq> of the original issued request). For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the CSR or SP.
<MsgRecDt>	Timestamp	Optional	Audit Record Creation Date. Date when this audit record is created.
<PmtAuthAddRs>	Aggregate *see section 7.7.1.2*	Required XOR	Add Payment Authorization Response Message Aggregate.
<PmtAuthModRs>	Aggregate *see section 7.7.2.2*	Required XOR	Modify Payment Authorization Response Message Aggregate.
<PmtAuthCanRs>	Aggregate *see section 7.7.3.2*	Required XOR	Cancel Payment Authorization Response Message Aggregate.

Tag	Type	Usage	Description
<PmtAuthRevRs>	Aggregate *see section 7.7.7.2*	Required XOR	Reversal Payment Authorization Response Message Aggregate.
</PmtAuthMsgRec>			

7.7.6   Payment Authorization Synchronization

7.7.6.1   Request <PmtAuthSyncRq>

Tag	Type	Usage	Description
<RqUID>	UUID	Required	Request Identifier. Sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header aggregate.
<AsyncRqUID>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously.
			For more information, see section 3.2.12.1.
<CustId>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<RecCtrlIn>	Aggregate *see section 3.2.12.2.1*	Optional	Records Control Input Aggregate.
<Token>	Identifier	Required	Token. Server assigned. Client should make no assumptions about the value of this token relative to others that it may have received. The token is only significant to the server that originally assigned it.
			This is a token that has been previously sent by the server, or zero for first time requests.

7.7.6.2   Response <PmtAuthSyncRs>

Tag	Type	Usage	Description
<Status>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, <StatusCode> defaults to 0 (zero).
<RqUID>	UUID	Required Echoed	Request Identifier.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header aggregate.
<MsgRsHdr>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header aggregate.

Tag	Type	Usage	Description
<AsyncRqUID>	UUID	Optional Echoed	Asynchronous Request Identifier.
<CustId>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<RecCtrlOut>	Aggregate *see section 3.2.12.2.2*	Optional	Records Control Output Aggregate.
<Token>	Identifier	Required Echoed	Token. Server assigned. Client should make no assumptions about the value of this token relative to others that it may have received. The token is only significant to the server that originally assigned it.
			This is a token that has been previously sent by the server, or zero for first time requests.
<NewToken>	Identifier	Required	New Token. Server assigned. Client should make no assumptions about the value of this token relative to others that it may have received. The token is only significant to the server that originally assigned it.
<PmtAuthMsgRec>	Aggregate	Optional Repeating	Payment Authorization Message Record Aggregate.
<CustId>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the entity who initiated the request (e.g., the one whose identifier appears in the <SignonRq> of the original issued request). For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the CSR or SP.
<MsgRecDt>	Timestamp	Optional	Audit Record Creation Date. Date when this audit record is created.
<PmtAuthAddRs>	Aggregate *see section 7.7.1.2*	Required XOR	Add Payment Authorization Response Message Aggregate.
<PmtAuthModRs>	Aggregate *see section 7.7.2.2*	Required XOR	Modify Payment Authorization Response Message Aggregate.
<PmtAuthCanRs>	Aggregate *see section 7.7.3.2*	Required XOR	Cancel Payment Authorization Response Message Aggregate.
</PmtAuthMsgRec>			

7.7.7 Payment Authorization Reversal

7.7.7.1 Request <PmtAuthRevRq>

Tag	Type	Usage	Description
<RqUID>	UUID	Required	Request Identifier.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header aggregate.

Tag	Type	Usage	Description
<AsyncRqUID>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously.
			For more information, see section **3.2.12.1**.
<CustId>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<RevReasonCode>	Open Enum	Required	Reversal reason code. The reason why the client is requesting a reversal.
<Desc>	C-80	Optional	Description, text describing the reason for the reversal.
<PmtAuthMsgRqInfo>	Aggregate	Required	Payment Authorization Message Request Information Aggregate. This aggregate contains the message to reverse. See RevOptSupt in the service profile for specific reversal messages supported.
<PmtAuthAddRq>	Aggregate *see section 7.7.1.1*	Required XOR	Payment Authorization Add Request Message Aggregate.
<PmtAuthModRq>	Aggregate *see section 7.7.2.1*	Required XOR	Payment Authorization Modify Request Message Aggregate.
<PmtAuthCanRq>	Aggregate *see section 7.7.3.1*	Required XOR	Payment Authorization Cancel Request Message Aggregate.
<AcctInqRq>	Aggregate *see section 6.4.3.1*	Required XOR	Payment Authorization Inquiry Request Message Aggregate.
<PmtAuthAudRq>	Aggregate *see section 7.7.5.1*	Required XOR	Payment Authorization Audit Request Message Aggregate.
<PmtAuthSynchRq>	Aggregate	Required XOR	Payment Authorization Synchronization Request Message Aggregate.
</PmtAuthMsgRqInfo>			

7.7.7.2   Response <PmtAuthRevRs>

Tag	Type	Usage	Description
<Status>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, <StatusCode> defaults to 0 (zero).
<RqUID>	UUID	Required Echoed	Request Identifier.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header aggregate.
<MsgRsHdr>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header aggregate.

Tag	Type	Usage	Description
<AsyncRqUID>	UUID	Optional Echoed	Asynchronous Request Identifier.
<CustId>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Identification Aggregate.
<RevReasonCode>	Open Enum	Required Echoed	Reversal reason code. The reason why the client is requesting a reversal.
<Desc>	C-80	Optional Echoed	Description, text describing the reason for the reversal.
<PmtAuthMsgRqInfo>	Aggregate	Required Echoed	Payment Authorization Message Request Information Aggregate. This aggregate contains the message to reverse. See RevOptSupt in the service profile for specific reversal messages supported.
<PmtAuthAddRq>	Aggregate *see section 7.7.1.1*	Required Echoed XOR	Payment Authorization Add Request Message Aggregate.
<PmtAuthModRq>	Aggregate *see section 7.7.2.1*	Required Echoed XOR	Payment Authorization Modify Request Message Aggregate.
<PmtAuthCanRq>	Aggregate *see section 7.7.3.1*	Required Echoed XOR	Payment Authorization Cancel Request Message Aggregate.
<AcctInqRq>	Aggregate *see section 6.4.3.1*	Required Echoed XOR	Payment Authorization Inquiry Request Message Aggregate.
<PmtAuthAudRq>	Aggregate *see section 7.7.5.1*	Required Echoed XOR	Payment Authorization Audit Request Message Aggregate.
<PmtAuthSynchRq>	Aggregate	Required Echoed XOR	Payment Authorization Synchronization Request Message Aggregate.
</PmtAuthMsgRqInfo>			
<CSPRefId>	Identifier	Optional	Customer Service Provider Reference Identifier.
<SPRefId>	Identifier	Optional	Service Provider Reference Identifier.

7.8   Remittance

7.8.1   Remittance Add

The Remittance Add message allows a client to transmit a remittance advice to a BPP or BSP for posting.

7.8.1.1   Request <RemitAddRq>

Tag	Type	Usage	Description
<RqUID>	UUID	Required	Request Identifier. Sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header aggregate.

Tag	Type	Usage	Description
<AsyncRqUID>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously.
			For more information, see section 3.2.12.1.
<CustId>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<EffDt>	DateTime	Required	Effective Date. The date the payment was processed by the CPP. It is used for remittance advice from CPP to BPP. The date can be used by the BPP to associate the date the payment was processed with the posting date. This date should match the date settlement to the BPP/Biller was initiated.
<RemitInfo>	Aggregate *see section 7.3.5.1*	Required	Remittance Record Aggregate.
<DupChkOverride>	Boolean	Optional	Duplicate Check Override Flag. When set to *True*, requests that the server not perform duplicate checking if any is normally performed. The client is affirming that this is a new remittance being added.

7.8.1.2   Response <RemitAddRs>

The <RemitAddRs> message is used to provide an acknowledgement to a CPP-initiated <RemitAddRq>. It is also used in the Remittance Audit Response <RemitAudRs> to communicate to the client that remittances have been added by the CPP using <RemitAddRq>.

Tag	Type	Usage	Description
<Status>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, <StatusCode> defaults to 0 (zero).
<RqUID>	UUID	Required Echoed	Request Identifier.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header aggregate.
<MsgRsHdr>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header aggregate.
<AsyncRqUID>	UUID	Optional Echoed	Asynchronous Request Identifier.
<CustId>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.

Tag	Type	Usage	Description
<EffDt>	DateTime	Required Echoed	Effective Date. The date the payment was processed by the CPP. It is used for remittance advice from CPP to BPP. The date can be used by the BPP to associate the date the payment was processed with the posting date. This date should match the date settlement to the BPP/Biller was initiated.
<RemitInfo>	Aggregate *see section 7.3.5.1*	Required Echoed	Remittance Record Aggregate.
<DupChkOverride>	Boolean	Optional Echoed	Duplicate Check Override Flag.
<RemitRec>	Aggregate *see section 7.3.5*	Required	Remittance Record Aggregate.
<SPRefId>	Identifier	Optional	Service Provider Reference Identifier.

7.8.2   Remittance Modify

The Remittance Modify message allows a client to modify the current information about a Remittance that was set up using the Remittance Add message.

7.8.2.1   Request <RemitModRq>

Tag	Type	Usage	Description
<RqUID>	UUID	Required	Request Identifier. Sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header aggregate.
<AsyncRqUID>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously. For more information, see section 3.2.12.1.
<CustId>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<RemitId>	Identifier	Required	Remittance Identifier. Assigned by the server at the time the Remittance is first added. Cannot be modified by the client.
<EffDt>	DateTime	Required	Effective Date. The date the payment was processed by the CPP. It is used for remittance advice from CPP to BPP. The date can be used by the BPP to associate the date the payment was processed with the posting date. This date should match the date settlement to the BPP/Biller was initiated.
<RemitInfo>	Aggregate *see section 7.3.5.1*	Required	Remittance Information Aggregate.

Tag	Type	Usage	Description
<DupChkOverride>	Boolean	Optional	Duplicate Check Override Flag. When set to *True*, requests that the server not perform duplicate checking if any is normally performed. The client is affirming that this is a new remittance being added.

7.8.2.2   Response <RemitModRs>

The <RemitModRs> message is used to provide an acknowledgement to a CSP-initiated <RemitModRq>. It is also used in the Remittance Audit Response <RemitAudRs> to communicate to the client that the customer has modified remittances by using <RemitModRq>.

Tag	Type	Usage	Description
<Status>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, <StatusCode> defaults to 0 (zero).
<RqUID>	UUID	Required Echoed	Request Identifier.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header aggregate.
<MsgRsHdr>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header aggregate.
<AsyncRqUID>	UUID	Optional Echoed	Asynchronous Request Identifier.
<CustId>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<EffDt>	DateTime	Required Echoed	Effective Date. The date the payment was processed by the CPP. It is used for remittance advice from CPP to BPP. The date can be used by the BPP to associate the date the payment was processed with the posting date. This date should match the date settlement to the BPP/Biller was initiated.
<RemitId>	Identifier	Required Echoed	Remittance Identifier.
<RemitInfo>	Aggregate *see section 7.3.5.1*	Required Echoed	Remittance Record Aggregate.
<DupChkOverride>	Boolean	Optional Echoed	Duplicate Check Override Flag.
<RemitRec>	Aggregate *see section 7.3.5*	Required	Remittance Record Aggregate.
<SPRefId>	Identifier	Optional	Service Provider Reference Identifier.

7.8.3   Remittance Status Modify

7.8.3.1   Request <RemitStatusModRq>

The client must specify the remittance identifier along with the status.

Tag	Type	Usage	Description
<RqUID>	UUID	Required	Request Identifier. Sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header aggregate.
<AsyncRqUID>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously. For more information, see section 3.2.12.1.
<CustId>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<RemitId>	Identifier	Required	Remittance Identifier Aggregate. Assigned by the server at the time the Remittance is first added. Cannot be modified by the client. Used to reference a specific remittance.
<RemitStatus>	Aggregate *see section 7.3.5.2*	Required	Remittance Status Aggregate.

7.8.3.2   Response <RemitStatusModRs>

Tag	Type	Usage	Description
<Status>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, <StatusCode> defaults to 0 (zero).
<RqUID>	UUID	Required Echoed	Request Identifier.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header aggregate.
<MsgRsHdr>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header aggregate.
<AsyncRqUID>	UUID	Optional Echoed	Asynchronous Request Identifier.
<CustId>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.

Tag	Type	Usage	Description
<RemitId>	Identifier	Required Echoed	Remittance Identifier.
<RemitStatus>	Aggregate *see section 7.3.5.2*	Required Echoed	Remittance Status Aggregate.
<CSPRefId>	Identifier	Optional	Financial Institution Reference Number.
<SPRefId>	Identifier	Optional	Service Provider Reference Number.

7.8.4   Remittance Delete

The Delete Remittance message allows a client to delete a Remittance that was set up using the Remittance Add message.

7.8.4.1   Request <RemitDelRq>

Tag	Type	Usage	Description
<RqUID>	UUID	Required	Request Identifier. Sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header aggregate.
<AsyncRqUID>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously. For more information, see section 3.2.12.1.
<CustId>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<RemitId>	Identifier	Required	Remittance Identifier. Assigned by the server at the time the Remittance is first added. Cannot be modified by the client.

7.8.4.2   Response <RemitDelRs>

Tag	Type	Usage	Description
<Status>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, <StatusCode> defaults to 0 (zero).
<RqUID>	UUID	Required Echoed	Request Identifier. Sent by a client as a universally unique identifier for the message. Used to correlate responses with requests. Server must return <RqUID> if provided by client in a <PmtCanRq>.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header aggregate.

Tag	Type	Usage	Description
<MsgRsHdr>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header aggregate.
<AsyncRqUID>	UUID	Optional Echoed	Asynchronous Request Identifier.
<CustId>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<RemitId>	Identifier	Required Echoed	Remittance Identifier
<RemitRec>	Aggregate *see section 7.3.5*	Optional	Remittance Record Aggregate. This aggregate is provided in cases where the server keeps the remittance record on the server even after receiving a deletion request. This may occur if the deletion is not immediate, or if the server maintains the record with a status of Deleted.
<SPRefId>	Identifier	Optional	Service Provider Reference Identifier.

7.8.5 Remittance Inquiry

7.8.5.1 Request <RemitInqRq>

Tag	Type	Usage	Description
<RqUID>	UUID	Required	Request Identifier. Sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header aggregate.
<AsyncRqUID>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously. For more information, see section 3.2.12.1.
<CustId>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<RemitId>	Identifier	Optional Repeating	Remittance Identifier. Used as a selection criterion.
<SelRangeCurAmt>	Aggregate *see section 3.2.10.2*	Optional	Selection Range Currency Amount. Used as a selection criterion.
<SelRangeDt>	Aggregate *see section 3.2.10.1*	Optional	Date Selection Range. Used as a selection criterion.

Tag	Type	Usage	Description
<DeliveryMethod>	Open Enum	Optional Profiled values	Delivery Method. Default is Channel. Value must be supported in Service Profile.

7.8.5.2 Response <RemitInqRs>

Tag	Type	Usage	Description
<Status>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, <StatusCode> defaults to 0 (zero).
<RqUID>	UUID	Required Echoed	Request Identifier. Sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header aggregate.
<MsgRsHdr>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header aggregate.
<AsyncRqUID>	UUID	Optional Echoed	Asynchronous Request Identifier.
<CustId>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<RemitId>	Identifier	Optional Repeating Echoed	Remittance Identifier.
<SelRangeCurAmt>	Aggregate *see section 3.2.10.2*	Optional Echoed	Selection Range Currency Amount. Used as a selection criterion.
<SelRangeDt>	Aggregate *see section 3.2.10.1*	Optional Echoed	Date Selection Range. Used as a selection criterion.
<DeliveryMethod>	Open Enum	Optional Echoed	Delivery Method.
<RemitRec>	Aggregate *see section 7.3.5*	Optional Repeating	Remittance Record Aggregate.

7.8.6 Remittance Audit

Remittance Audit allows a client to audit Remittance Add/Modify/Delete messages associated with the current customer. When the <RemitStatus> changes, the server must generate an Rs message to the Rq that created the pending state. The <Status> <Severity> must always be Info.

7.8.6.1 Request <RemitAudRq>

Tag	Type	Usage	Description

Tag	Type	Usage	Description
<RqUID>	UUID	Required	Request Identifier. Sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header aggregate.
<AsyncRqUID>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously. For more information, see section 3.2.12.1.
<CustId>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<RecCtrlIn>	Aggregate *see section 3.2.12.2.1*	Optional	Records Control Input Aggregate.
<SelRangeDt>	Aggregate *see section 3.2.10.1*	Optional	Selection Range Date Aggregate.
<Method>	Closed Enum	Optional Repeating	Remittance Method. Valid Values: Add, Mod, Del This field is used as a selection criterion.
<RemitId>	Identifier	Optional Repeating	Remittance Identifier. Assigned by the server at the time the remittance is first added. Cannot be modified by the client. This field is used as a selection criterion.

7.8.6.2 Response <RemitAudRs>

Tag	Type	Usage	Description
<Status>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, <StatusCode> defaults to 0 (zero).
<RqUID>	UUID	Required Echoed	Request Identifier.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header aggregate.
<MsgRsHdr>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header aggregate.
<AsyncRqUID>	UUID	Optional Echoed	Asynchronous Request Identifier.

Tag	Type	Usage	Description
<CustId>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<RecCtrlOut>	Aggregate *see section 3.2.12.2.2*	Optional	Records Control Output Aggregate.
<SelRangeDt>	Aggregate *see section 3.2.10.1*	Optional Echoed	Selection Range Date Aggregate.
<Method>	Closed Enum	Optional Repeating Echoed	Remittance Method.
<RemitId>	Identifier	Optional Repeating Echoed	Remittance Identifier. Assigned by the server at the time the remittance is first added. Cannot be modified by the client. This field is used as a selection criterion.
<RemitMsgRec>	Aggregate	Optional Repeating	Remittance Message Record Aggregate.
<CustId>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the entity who initiated the request (e.g., the one whose identifier appears in the <SignonRq> of the original issued request). For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the CSR or SP.
<MsgRecDt>	Timestamp	Optional	Audit Record Creation Date. Date when this audit record is created.
<RemitAddRs>	Aggregate *see section 7.8.1.2*	Required XOR	Remittance Add Response Message Aggregate.
<RemitModRs>	Aggregate *see section 7.8.2.2*	Required XOR	Remittance Modify Response Message Aggregate.
<RemitStatusModRs>	Aggregate *see section 7.8.3.2*	Required XOR	Remittance Status Modify Response Message Aggregate.
<RemitDelRs>	Aggregate *see section 7.8.4.2*	Required XOR	Remittance Delete Response Message Aggregate.
<RemitRevRs>	Aggregate *see section 7.8.8.2*	Required XOR	Remittance Reversal Response Message Aggregate.
</RemitMsgRec>			

7.8.7   *Remittance Synchronization*

A client uses the Remittance Synchronization message to retrieve a list of changes that have occurred to a client's remittance. This message will commonly be used to enable a client to synchronize with the server to get up-to-date status information about a remittance. The results of this message typically will status change of a remittance advice.

7.8.7.1 Request <RemitSyncRq>

Tag	Type	Usage	Description
<RqUID>	UUID	Required	Request Identifier. Sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header aggregate.
<AsyncRqUID>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously.
			For more information, see section 3.2.12.1.
<CustId>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<RecCtrlIn>	Aggregate *see section 3.2.12.2.1*	Optional	Records Control Input Aggregate.
<Token>	Identifier	Required	Token. Server assigned. Client should make no assumptions about the value of this token relative to others that it may have received. The token is only significant to the server that originally assigned it.
			This is a token that has been previously sent by the server, or "0" for first time requests.

7.8.7.2 Response <RemitSyncRs>

Tag	Type	Usage	Description
<Status>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, <StatusCode> defaults to 0 (zero).
<RqUID>	UUID	Required Echoed	Request Identifier.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header aggregate.
<MsgRsHdr>	Aggregate *see section 3.2.1.2*	Optional	Message Respoonse Header aggregate.
<AsyncRqUID>	UUID	Optional Echoed	Asynchronous Request Identifier.
<CustId>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.

Tag	Type	Usage	Description
<RecCtrlOut>	Aggregate *see section 3.2.12.2.2*	Optional	Records Control Output Aggregate.
<Token>	Identifier	Required Echoed	Token. Server assigned. Client should make no assumptions about the value of this token relative to others that it may have received. The token is only significant to the server that originally assigned it. This is a token that has been previously sent by the server, or "0" for first time requests.
<NewToken>	Identifier	Required	New Token. Server assigned. Client should make no assumptions about the value of this token relative to others that it may have received. The token is only significant to the server that originally assigned it.
<RemitMsgRec>	Aggregate	Optional Repeating	Remittance Message Record Aggregate.
<CustId>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the entity who initiated the request (e.g., the one whose identifier appears in the <SignonRq> of the original issued request). For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the CSR or SP.
<MsgRecDt>	Timestamp	Optional	Audit Record Creation Date. Date when this audit record is created.
<RemitAddRs>	Aggregate *see section 7.8.1.2*	Required XOR	Remittance Add Response Message Aggregate.
<RemitModRs>	Aggregate *see section 7.8.2.2*	Required XOR	Remittance Modify Response Message Aggregate.
<RemitStatusModRs>	Aggregate *see section 7.8.3.2*	Required XOR	Remittance Status Modify Response Message Aggregate.
<RemitDelRs>	Aggregate *see section 7.8.4.2*	Required XOR	Remittance Delete Response Message Aggregate.
</RemitMsgRec>			

7.8.8   Remittance Reversal

7.8.8.1   Request <RemitRevRq>

Tag	Type	Usage	Description
<RqUID>	UUID	Required	Request Identifier.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header aggregate.
<AsyncRqUID>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously. For more information, see section **3.2.12.1**.

Tag	Type	Usage	Description
<CustId>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<RevReasonCode>	Open Enum	Required	Reversal reason code. The reason why the client is requesting a reversal.
<Desc>	C-80	Optional	Description, text describing the reason for the reversal.
<RemitMsgRqInfo>	Aggregate	Required	Remit Message Request Information Aggregate. This aggregate contains the message to reverse. See RevOptSupt in the service profile for specific reversal messages supported.
<RemitAddRq>	Aggregate *see section 7.8.1.1*	Required XOR	Remit Add Request Message Aggregate.
<RemitModRq>	Aggregate *see section 7.8.2.1*	Required XOR	Remit Modify Request Message Aggregate.
<RemitStatusModRq>	Aggregate *see section 7.8.3.1*	Required XOR	Remit Status Modify Request Message Aggregate.
<RemitDelRq>	Aggregate *see section 7.8.4.1*	Required XOR	Remit Delete Request Message Aggregate.
<RemitInqRq>	Aggregate *see section 7.8.5.1*	Required XOR	Remit Inquiry Request Message Aggregate.
<RemitAudRq>	Aggregate *see section 7.8.6.1*	Required XOR	Remit Audit Request Message Aggregate.
<RemitSyncRq>	Aggregate *see section 7.8.7.1*	Required XOR	Remit Synchronization Request Message Aggregate.
</RemitMsgRqInfo>			

7.8.8.2　Response <RemitRevRs>

Tag	Type	Usage	Description
<Status>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, <StatusCode> defaults to 0 (zero).
<RqUID>	UUID	Required Echoed	Request Identifier.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header aggregate.
<MsgRsHdr>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header aggregate.
<AsyncRqUID>	UUID	Optional Echoed	Asynchronous Request Identifier.

Tag	Type	Usage	Description
\<CustId>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Identification Aggregate.
\<RevReasonCode>	Open Enum	Required Echoed	Reversal reason code. The reason why the client is requesting a reversal.
\<Desc>	C-80	Optional Echoed	Description, text describing the reason for the reversal.
\<RemitMsgRqInfo>	Aggregate	Required Echoed	Remit Message Request Information Aggregate. This aggregate contains the message to reverse. See RevOptSupt in the service profile for specific reversal messages supported.
\<RemitAddRq>	Aggregate *see section 7.8.1.1*	Required XOR	Remit Add Request Message Aggregate.
\<RemitModRq>	Aggregate *see section 7.8.2.1*	Required XOR	Remit Modify Request Message Aggregate.
\<RemitStatusModRq>	Aggregate *see section 7.8.3.1*	Required XOR	Remit Status Modify Request Message Aggregate.
\<RemitDelRq>	Aggregate *see section 7.8.4.1*	Required XOR	Remit Delete Request Message Aggregate.
\<RemitInqRq>	Aggregate *see section 7.8.5.1*	Required XOR	Remit Inquiry Request Message Aggregate.
\<RemitIAudRq>	Aggregate	Required XOR	Remit Audit Request Message Aggregate.
\<RemitSyncRq>	Aggregate *see section 7.8.7.1*	Required XOR	Remit Synchronization Request Message Aggregate.
\</RemitMsgRqInfo>			
\<CSPRefId>	Identifier	Optional	Customer Service Provider Reference Identifier.
\<SPRefId>	Identifier	Optional	Service Provider Reference Identifier.

7.9   Recurring Payment Model

7.9.1   Recurring Payment Model Add

The Recurring Payment Model Add message allows a client to set up a recurring or repeating payment where the payment amount is the same. Examples of these types of payments are mortgages, car loans, equity loans, etc. The initial and/or final payment amount may be different from the normal recurring payment amount if supported by the Pay provider.

7.9.1.1   Request \<RecPmtAddRq>

Tag	Type	Usage	Description
\<RqUID>	UUID	Required	Request Identifier. Sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.
\<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header aggregate.

Tag	Type	Usage	Description
<AsyncRqUID>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously. For more information, see section 3.2.12.1.
<CustId>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<PmtInfo>	Aggregate *see section 7.3.3.1*	Required	Payment Information Aggregate.
<RecModelInfo>	Aggregate *see section 3.2.11.1*	Required	Recurring Model Information Aggregate.
<DupChkOverride>	Boolean	Optional Profiled support	Duplicate Check Override Flag. When set to *True*, requests that the server not perform duplicate checking if any is normally performed. The client is affirming that this is a new recurring payment model being added.

7.9.1.2   Response <RecPmtAddRs>

Tag	Type	Usage	Description
<Status>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, <StatusCode> defaults to 0 (zero).
<RqUID>	UUID	Required Echoed	Request Identifier.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header aggregate.
<MsgRsHdr>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header aggregate.
<AsyncRqUID>	UUID	Optional Echoed	Asynchronous Request Identifier.
<CustId>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<PmtInfo>	Aggregate *see section 7.3.3.1*	Required Echoed	Payment Information Aggregate.
<RecModelInfo>	Aggregate *see section 3.2.11.1*	Required Echoed	Recurring Model Information Aggregate.

Tag	Type	Usage	Description
<DupChkOverride>	Boolean	Optional Profiled support Echoed	Duplicate Check Override Flag.
<RecPmtRec>	Aggregate *see section 7.3.3.5*	Required	Recurring Payment Model Record Aggregate.
<CSPRefId>	Identifier	Optional	Customer Service Provider Reference Identifier.
<SPRefId>	Identifier	Optional	Service Provider Reference Identifier.

7.9.2 Recurring Payment Instance Add

The Recurring Payment Instance Add message allows a client to manually trigger the spawning of a payment instance from a Recurring Payment Model that has its frequency value defined as "Manually". This message is particularly useful when payments to a specific payee need to occur on an irregular frequency basis or perhaps when a client desires direct control of the spawning to manually override certain elements of each payment instance. Values provided within <DueDt>, <CurAmt>, <RemitInstruction>, <RemitDetail>, and <PmtLegalRpt> override the values already specified within the recurring payment model specified by <RecPmtId>.

7.9.2.1 Request <RecPmtInstAddRq>

Tag	Type	Usage	Description
<RqUID>	UUID	Required	Request Identifier. Sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header aggregate.
<AsyncRqUID>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously. For more information, see section 3.2.12.1.
<CustId>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<RecPmtId>	Identifier	Required	Recurring Payment Model Identifier. The identifier for the recurring payment model.
<DueDt>	DateTime	Required	Due Date. The date that the payment is due.
<CurAmt>	Currency Amount	Optional	Currency Amount. The total amount being debited from the funding account. If this element is not present, the amount specified in the recurring model referenced by <RecPmtId> is the intended amount.
<RemitInstruction>	Aggregate *see section 7.3.5.1.1*	Optional	Remittance Instruction.
<RemitDetail>	Aggregate	Optional Repeating	Remittance Detail.

Tag	Type	Usage	Description
\<PmtLegalRpt\>	Aggregate *see section 7.3.5.1.4*	Optional	Payment Legal Reporting Aggregate.

7.9.2.2 Response \<RecPmtInstAddRs\>

Tag	Type	Usage	Description
\<Status\>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, \<StatusCode\> defaults to 0 (zero).
\<RqUID\>	UUID	Required Echoed	Request Identifier.
\<MsgRqHdr\>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header aggregate.
\<MsgRsHdr\>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header aggregate.
\<AsyncRqUID\>	UUID	Optional Echoed	Asynchronous Request Identifier.
\<CustId\>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in \<SignonRq\>. For example, if a CSR or SP issues the request on behalf of the user, then \<CustId\> is required, and must contain the value of the user whose request is being issued.
\<RecPmtId\>	Identifier	Required Echoed	Recurring Payment Model Identifier. The identifier for the recurring payment model.
\<DueDt\>	DateTime	Required Echoed	Due Date. The date that the payment is due.
\<CurAmt\>	Currency Amount	Optional Echoed	Currency Amount. The total amount being debited from the funding account. If this element is not present, the amount specified in the recurring model referenced by \<RecPmtId\> is the intended amount.
\<RemitDetail\>	Aggregate	Optional Repeating Echoed	Remittance Detail.
\<PmtLegalRpt\>	Aggregate *see section 7.3.5.1.4*	Optional Echoed	Payment Legal Reporting Aggregate.
\<PmtRec\>	Aggregate *see section 7.3.3*	Required	Payment Record.

7.9.3 Recurring Payment Model Modify

The Modify Recurring Payment Model message allows a client to modify the current definition of a recurring or repeating payment. Whether a modification to the model affects any current pending payment instance(s) from the model depends on profile settings and the message contents.

7.9.3.1 Request \<RecPmtModRq\>

Tag	Type	Usage	Description

Tag	Type	Usage	Description
\<RqUID\>	UUID	Required	Request Identifier. Sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.
\<MsgRqHdr\>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header aggregate.
\<AsyncRqUID\>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously. For more information, see section 3.2.12.1.
\<CustId\>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in \<SignonRq\>. For example, if a CSR or SP issues the request on behalf of the user, then \<CustId\> is required, and must contain the value of the user whose request is being issued.
\<RecPmtId\>	Identifier	Required	Recurring Payment Model Identifier. Assigned by the server at the time the Recurring Payment Model is first added. Cannot be modified by the client.
\<PmtInfo\>	Aggregate *see section 7.3.3.1*	Required	Payment Information Aggregate. **Note:** *If the server supports Customer Payee lists, the Payee must be specified using \<PmtInfo\> \<CustPayeeId\>.*
\<RecModelInfo\>	Aggregate *see section 3.2.11.1*	Required	Recurring Model Information Aggregate.
\<ModPending\>	Boolean	Optional Profiled support	Modify Pending Indicator. If set to *True*, all pending payment instances that were automatically generated from the recurring model that is being modified must also be modified. If absent or set to *False*, only the model is modified; any pending payments must remain unaffected by the modification of the model. This element must be ignored unless \<ModPendingType\> in the Pay Service Profile = IfRequested.

7.9.3.2 Response \<RecPmtModRs\>

Tag	Type	Usage	Description
\<Status\>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, \<StatusCode\> defaults to 0 (zero).
\<RqUID\>	UUID	Required Echoed	Request Identifier.
\<MsgRqHdr\>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header aggregate.
\<MsgRsHdr\>	Aggregate *see section 3.2.1.2*	Optional	Message Request Header aggregate.
\<AsyncRqUID\>	UUID	Optional Echoed	Asynchronous Request Identifier.

Tag	Type	Usage	Description
<CustId>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<PmtInfo>	Aggregate *see section 7.3.3.1*	Required Echoed	Payment Information Aggregate.
<RecModelInfo>	Aggregate *see section 3.2.11.1*	Required Echoed	Recurring Model Information Aggregate.
<ModPending>	Boolean	Optional Profiled support Echoed	Modify Pending Flag.
<RecPmtRec>	Aggregate *see section 7.3.3.5*	Required	Recurring Payment Model Record Aggregate.
<CSPRefId>	Identifier	Optional	Customer Service Provider Reference Identifier.
<SPRefId>	Identifier	Optional	Service Provider Reference Identifier.

7.9.4   Recurring Payment Model Cancel

The Recurring Payment Model Cancel message allows a client to cancel a customer's Recurring Payment Model. Cancellation of a Recurring Payment Model always also cancels any pending Payments that were generated from that model.

7.9.4.1   Request <RecPmtCanRq>

Tag	Type	Usage	Description
<RqUID>	UUID	Required	Request Identifier. Sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header aggregate.
<AsyncRqUID>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously. For more information, see section 3.2.12.1.
<CustId>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<RecPmtId>	Identifier	Required	Recurring Payment Model Identifier. Assigned by the server at the time the Recurring Payment Model is first added. Cannot be modified by the client.

Tag	Type	Usage	Description
<CascadeDel>	Boolean	Optional	Cascade Delete. If *True*, server must delete all dependent payments when this model is deleted. If *False* or omitted, the model must not be deleted if dependent payments exist.

7.9.4.2 Response <RecPmtCanRs>

Tag	Type	Usage	Description
<Status>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, <StatusCode> defaults to 0 (zero).
<RqUID>	UUID	Required Echoed	Request Identifier.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header aggregate.
<MsgRsHdr>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header aggregate.
<AsyncRqUID>	UUID	Optional Echoed	Asynchronous Request Identifier.
<CustId>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<RecPmtId>	Identifier	Required Echoed	Recurring Payment Model Identifier.
<CascadeDel>	Boolean	Optional Echoed	Cascade Delete.
<RecPmtRec>	Aggregate *see section 7.3.3.5*	Optional XOR	Recurring Payment Record. This aggregate is provided in cases where the server keeps the recurring payment model's record on the server even after receiving a deletion request. This may occur if the deletion is not immediate, or if the server maintains the record with a status of Cancelled.
<DependentType>	Open Enum	Optional XOR Repeating	Open Enum containing a list of dependent object types that exist for the model. Valid values: Pmt
<CSPRefId>	Identifier	Optional	Customer Service Provider Reference Identifier.
<SPRefId>	Identifier	Optional	Service Provider Reference Identifier.

7.9.5 Recurring Payment Model Inquiry

The Recurring Payment Inquiry message allows a client to get a list of the current Recurring Payment Models. Clients that keep local copies of a customer's Recurring Payment Models may use this message to "refresh" their copies of the customer's Recurring Payment Models.

7.9.5.1 Request <RecPmtInqRq>

Tag	Type	Usage	Description

Tag	Type	Usage	Description
<RqUID>	UUID	Required	Request Identifier. Sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header aggregate.
<AsyncRqUID>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously.
			For more information, see section 3.2.12.1.
<CustId>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<RecCtrlIn>	Aggregate *see section 3.2.12.2.1*	Optional	Records Control Input Aggregate.
<RecPmtId>	Identifier	Optional Repeating	Recurring Payment Model Identifier. Assigned by the server at the time the Recurring Payment Model is first added. Cannot be modified by the client.
			This field is used as a selection criterion.
<CustPayeeId>	Identifier	Optional Repeating	Customer's Payee Identifier. This field is used as a selection criterion. If the server supports Payee Lists, the server assigns <CustPayeeId> at the time the Payee is first added.
<DepAcctIdFrom>	Aggregate *see section 3.2.7.1.2*	Optional XOR	Deposit Account Aggregate.
			Source account for payment. This field is used as a selection criterion.
<CardAcctIdFrom>	Aggregate *see section 3.2.7.1.3*	Optional XOR	Card Account Identifier Aggregate.
			Source account for payment. This field is used as a selection criterion.
<Category>	C-40	Optional Repeating	Payment Category. The customer assigns categories.
			This field is used as a selection criterion.
<Memo>	C-255	Optional Repeating	Memo for Payment. From Customer to Payee.
			This field is used as a selection criterion.
<BillingAcct>	C-32	Optional Repeating	Customer Account Number with Payee.
			This field is used as a selection criterion.
<SelRangeCurAmt>	Aggregate *see section 3.2.10.2*	Optional	Selection Range Amount Aggregate.
<CSPRefId>	Identifier	Optional Repeating	Customer Service Provider Reference Identifier.
			This field is used as a selection criterion.
<SPRefId>	Identifier	Optional Repeating	Service Provider Reference Identifier.
			This field is used as a selection criterion.

Tag	Type	Usage	Description
<IncToken>	Boolean	Optional	Include Token. If *True*, a <Token> should be included in the response to set a base for future Sync messages. If *False* or omitted, no <Token> is returned.
<DeliveryMethod>	Open Enum	Optional Profiled values	Delivery Method. Default is Channel. Value must be supported in Service Profile.

7.9.5.2   Response <RecPmtInqRs>

Tag	Type	Usage	Description
<Status>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, <StatusCode> defaults to 0 (zero).
<RqUID>	UUID	Required Echoed	Request Identifier.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header aggregate.
<MsgRsHdr>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header aggregate.
<AsyncRqUID>	UUID	Optional Echoed	Asynchronous Request Identifier.
<CustId>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<RecCtrlOut>	Aggregate *see section 3.2.12.2.2*	Optional *but see Description*	Records Control Output Aggregate. Required if <RecCtrlIn> was provided in the request and the server supports Records Control.
<RecPmtId>	Identifier	Optional Repeating Echoed	Recurring Payment Model Identifier.
<CustPayeeId>	Identifier	Optional Repeating Echoed	Customer's Payee Identifier.
<DepAcctIdFrom>	Aggregate *see section 3.2.7.1.2*	Optional XOR Echoed	Deposit Account Aggregate.
<CardAcctIdFrom>	Aggregate *see section 3.2.7.1.3*	Optional XOR Echoed	Card Account Identifier Aggregate.
<Category>	C-40	Optional Repeating Echoed	Payment Category.
<Memo>	C-255	Optional Repeating Echoed	Memo for Payment.

Tag	Type	Usage	Description
\<BillingAcct>	C-32	Optional Repeating Echoed	Customer Account Number with Payee.
\<SelRangeCurAmt>	Aggregate *see section 3.2.10.2*	Optional Echoed	Selection Range Amount Aggregate.
\<CSPRefId>	Identifier	Optional Repeating Echoed	Customer Service Provider Reference Identifier.
\<SPRefId>	Identifier	Optional Repeating Echoed	Service Provider Reference Identifier.
\<IncToken>	Boolean	Optional Echoed	Include Token.
\<DeliveryMethod>	Open Enum	Optional Echoed	Delivery Method.
\<RecPmtRec>	Aggregate *see section 7.3.3.5*	Optional Repeating	Recurring Payment Model Record Aggregate. One record is returned for each of the customer's Recurring Payment Models that meets the selection criteria specified in the request message
\<Token>	Identifier	Optional *but see Description*	Token. Server Assigned. Client should make no assumptions about the value of this token relative to others that it may have received. The token is significant only to the server that originally assigned it. The token is only included if \<IncToken>=*True* in the request. \<Token>=0 is returned if no records are returned within the response.

7.9.6   *Recurring Payment Model Audit*

7.9.6.1   Request \<RecPmtAudRq>

Tag	Type	Usage	Description
\<RqUID>	UUID	Required	Request Identifier. Sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.
\<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header aggregate.
\<AsyncRqUID>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously. For more information, see section 3.2.12.1.
\<CustId>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in \<SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then \<CustId> is required, and must contain the value of the user whose request is being issued.
\<RecCtrlIn>	Aggregate *see section 3.2.12.2.1*	Optional	Message Records Control Input Aggregate.

Tag	Type	Usage	Description
<SelRangeDt>	Aggregate *see section 3.2.10.1*	Optional	Selection Range Date Aggregate.
<Method>	Closed Enum	Optional Repeating	Recurring Payment Model Method. Valid values: Add, Mod, Can This field is used as a selection criterion.
<RecPmtId>	Identifier	Optional Repeating	Recurring Payment Model Identifier. Assigned by the server at the time the Recurring Payment Model is first added. This field is used as a selection criterion.

7.9.6.2   Response <RecPmtAudRs>

Tag	Type	Usage	Description
<Status>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, <StatusCode> defaults to 0 (zero).
<RqUID>	UUID	Required Echoed	Request Identifier.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header aggregate.
<MsgRsHdr>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header aggregate.
<AsyncRqUID>	UUID	Optional Echoed	Asynchronous Request Identifier.
<CustId>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<RecCtrlOut>	Aggregate *see section 3.2.12.2.2*	Optional *but see Description*	Records Control Output Aggregate. Required if <RecCtrlIn> was provided in the request and the server supports Records Control.
<SelRangeDt>	Aggregate *see section 3.2.10.1*	Optional Echoed	Selection Range Date Aggregate.
<Method>	Closed Enum	Optional Repeating Echoed	Recurring Payment Model Method.
<RecPmtId>	Identifier	Optional Repeating Echoed	Recurring Payment Model Identifier.
<RecPmtMsgRec>	Aggregate	Optional Repeating	Recurring Payment Message Record Aggregate.

Tag	Type	Usage	Description
<CustId>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the entity who initiated the request (e.g., the one whose identifier appears in the <SignonRq> of the original issued request). For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the CSR or SP.
<MsgRecDt>	Timestamp	Optional	Audit Record Creation Date. Date when this audit record is created.
<RecPmtAddRs>	Aggregate *see section 7.9.1.2*	Required XOR	Add Recurring Payment Model Response Message Aggregate.
<RecPmtModRs>	Aggregate *see section 7.9.3.2*	Required XOR	Modify Recurring Payment Model Response Message Aggregate.
<RecPmtCanRs>	Aggregate *see section 7.9.4.2*	Required XOR	Cancel Recurring Payment Model Response Message Aggregate.
<RecPmtRevRs>	Aggregate *see section 7.9.4.2*	Required XOR	Reversal Recurring Payment Model Response Message Aggregate.
</RecPmtMsgRec>			

7.9.7   Recurring Payment Model Sync

The Recurring Payment Model Sync message allows clients to retrieve a list of changes that have occurred to a customer's Recurring Payment Models. This message may be used to enable a client that keeps local copies of a customer's Recurring Payment messages to synchronize its database against the one kept by the Pay provider. The results of this message tell a client what the customer has done using other clients since they last used this one.

7.9.7.1   Request <RecPmtSyncRq>

Tag	Type	Usage	Description
<RqUID>	UUID	Required	Request Identifier. Sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header aggregate.
<AsyncRqUID>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously. For more information, see section 3.2.12.1.
<CustId>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<RecCtrlIn>	Aggregate *see section 3.2.12.2.1*	Optional	Message Records Control Input Aggregate.

Tag	Type	Usage	Description
\<Token\>	Identifier	Required	Token. Server assigned. Client should make no assumptions about the value of this token relative to others that it may have received. The token is only significant to the server that originally assigned it. This is a token that has been previously sent by the server, or zero for first-time requests.

7.9.7.2   Response \<RecPmtSyncRs\>

Tag	Type	Usage	Description
\<Status\>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, \<StatusCode\> defaults to 0 (zero).
\<RqUID\>	UUID	Required Echoed	Request Identifier.
\<MsgRqHdr\>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header aggregate.
\<MsgRsHdr\>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header aggregate.
\<AsyncRqUID\>	UUID	Optional Echoed	Asynchronous Request Identifier.
\<CustId\>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in \<SignonRq\>. For example, if a CSR or SP issues the request on behalf of the user, then \<CustId\> is required, and must contain the value of the user whose request is being issued.
\<RecCtrlOut\>	Aggregate *see section 3.2.12.2.2*	Optional *but see Description*	Records Control Output Aggregate. Required if \<RecCtrlIn\> was provided in the request and the server supports Records Control.
\<Token\>	Identifier	Required Echoed	Token. Server assigned. Client should make no assumptions about the value of this token relative to others that it may have received. The token is only significant to the server that originally assigned it. This is a token that has been previously sent by the server, or zero for first-time requests.
\<NewToken\>	Identifier	Required	Token. Server assigned. Client should make no assumptions about the value of this token relative to others that it may have received. The token is only significant to the server that originally assigned it. If the client sent a token in the request, the server returns a new token based on this audit message.
\<RecPmtMsgRec\>	Aggregate	Optional Repeating	Recurring Payment Message Record Aggregate.
\<CustId\>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the entity who initiated the request (e.g., the one whose identifier appears in the \<SignonRq\> of the original issued request). For example, if a CSR or SP issues the request on behalf of the user, then \<CustId\> is required, and must contain the value of the CSR or SP.

Tag	Type	Usage	Description
<MsgRecDt>	Timestamp	Optional	Audit Record Creation Date. Date when this audit record is created.
<RecPmtAddRs>	Aggregate *see section 7.9.1.2*	Required XOR	Add Recurring Payment Model Response Message Aggregate.
<RecPmtModRs>	Aggregate *see section 7.9.3.2*	Required XOR	Modify Recurring Payment Model Response Message Aggregate.
<RecPmtCanRs>	Aggregate *see section 7.9.4.2*	Required XOR	Cancel Recurring Payment Model Response Message Aggregate.
</RecPmtMsgRec>			

7.10   Payment Acknowledgement

Upon the receipt of a payment request <PmtAddRq>, it is the fiduciary responsibility of the banking partner (CPP) to supply to the client an acknowledgment of the various processing cycles in an event driven environment. The acknowledgments are both positive and negative, indicating successful and unsuccessful application processing, respectively, of the payment transaction sent. This is an event driven process.

The client is obligated to build a business process to review these messages within a defined timeline for applicable handling of failed messages by the client. All successful notifications assume that the payment has been forwarded to the appropriate payment platform for settlement processing. The client has the responsibility to notify the banking partner of non-receipt of the acknowledgment.

The acknowledgments contain both summary information (total number of payments and dollars accepted, rejected, and changed) along with detail information for each payment file or group of payments transmitted.

There are payment systems which also generate acknowledgments when the file is received by that specific payment system along with another acknowledgment which carries a reference number assigned by that payment system. Example: A wire payment transaction has been acknowledged as successful by the banking application system and has been passed to the wire platform system. This platform acknowledges receipt of the transaction by creating an acknowledgment message. During the processing cycle, a unique fed reference number is assigned to the transaction and is communicated to the client via another acknowledgment. This fed reference number is then tied back to the original transaction for tracking and archiving purposes.

Level 1, Acknowledgment Summary and Detail: Validation of application processing for all transactions requested; supplies counts and values at a summary level; positive or negative acceptance supplied at detail level.

Level 2, Acknowledgment Detail: Validation of receipt of transaction at Payment Network; single transaction based process

Level 3, Acknowledgment Detail: Validation of network processing and return of trace reference number; single transaction based process

The payment document number and AP batch number is recorded in the customer's general ledger for account reconciliation processing after the payment has been settled. These are two keys that are required to be carried throughout the payment cycle and acknowledgment process for final reconciliation by the customer.

Depicted below is a sample of this flow:
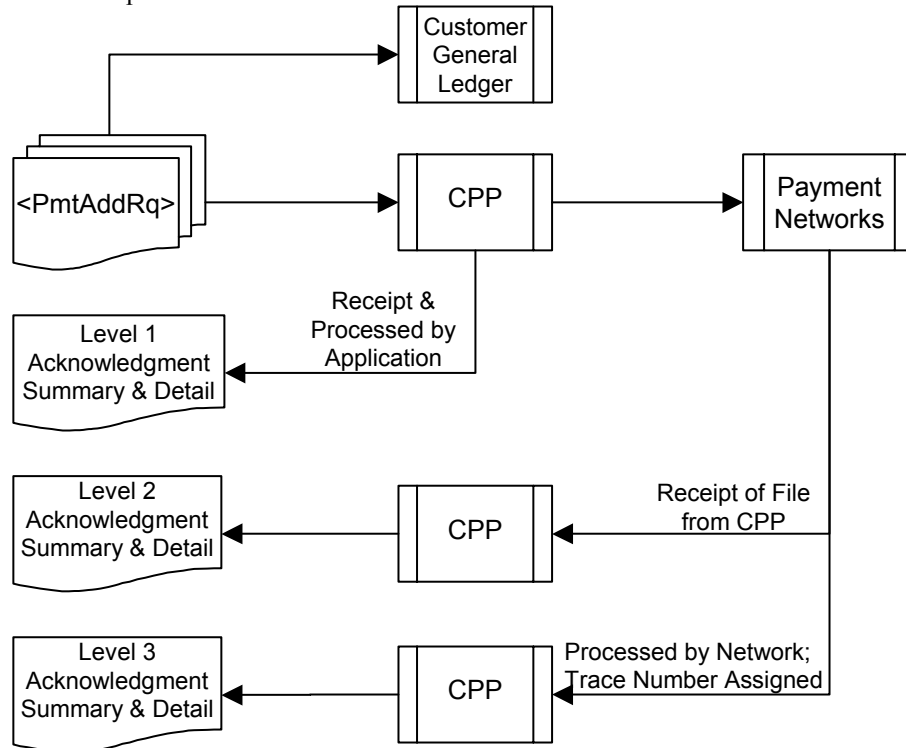
7.10.1   Payment Acknowledgment Inquiry

A customer may send a request to the CPP inquiring about the consolidated summary of a single or group of payments submitted for processing.

7.10.1.1   Request <PmtAckInqRq>

The user must specify the payment reference identifier.

Tag	Type	Usage	Description
<RqUID>	UUID	Required	Request Identifier. Sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header aggregate.
<AsyncRqUID>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously. For more information, see section 3.2.12.1.
<CustId>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<SelRangeDueDt>	Aggregate *see section 3.2.10.1*	Optional	Selection Range Due Date Aggregate. This is used as a selection criterion.

Tag	Type	Usage	Description
\<SvcRqUID\>	UUID	Optional	Service Request Identifier. The payment file identifier that was created by the payer in the Accounts Payable system when the file is created and sent. This is used as a selection criterion.

7.10.1.2   Response \<PmtAckInqRs\>

The CPP may respond to the \<PmtAckInqRq\> with the following \<PmtAckInqRs\>, acknowledging the receipt and application processing of the single or group of payments processed within the Pay Service.

Tag	Type	Usage	Description
\<Status\>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, \<StatusCode\> defaults to 0 (zero).
\<RqUID\>	UUID	Required Echoed	Request Identifier. Sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.
\<MsgRqHdr\>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header aggregate.
\<MsgRsHdr\>	Aggregate *see section 3.2.1.2*	Optional	Message Rewsponse Header aggregate.
\<AsyncRqUID\>	UUID	Optional Echoed	Asynchronous Request Identifier.
\<CustId\>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in \<SignonRq\>. For example, if a CSR or SP issues the request on behalf of the user, then \<CustId\> is required, and must contain the value of the user whose request is being issued.
\<SelRangeDueDt\>	Aggregate *see section 3.2.10.1*	Optional Echoed	Selection Range Due Date Aggregate.
\<SvcRqUID\>	UUID	Optional Echoed	Service Request Identifier.
\<PmtAckRec\>	Aggregate *see section 7.3.10*	Optional Repeating	Payment Acknowledgment Record Aggregate. Contains acknowledgment records at a consolidated or summarized level applicable to a \<SvcRqUID\>, or batch of payments submitted by the customer.
\<CSPRefId\>	Identifier	Optional	Financial Institution Reference Number.
\<SPRefId\>	Identifier	Optional	Service Provider Reference Number.

7.10.2   *Payment Acknowledgment Reversal*

7.10.2.1   Request \<PmtAckRevRq\>

Tag	Type	Usage	Description
\<RqUID\>	UUID	Required	Request Identifier.

Tag	Type	Usage	Description
\<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header aggregate.
\<AsyncRqUID>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously.
			For more information, see section **3.2.12.1**.
\<CustId>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in \<SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then \<CustId> is required, and must contain the value of the user whose request is being issued.
\<RevReasonCode>	Open Enum	Required	Reversal reason code. The reason why the client is requesting a reversal.
\<Desc>	C-80	Optional	Description, text describing the reason for the reversal.
\<PmtAckMsgRqInfo>	Aggregate	Required	Payment Acknowledgement Message Request Information Aggregate. This aggregate contains the message to reverse. See RevOptSupt in the service profile for specific reversal messages supported.
\<PmtAckInqRq>	Aggregate *see section 7.10.1.1*	Optional	Payment Acknowledgement Inquiry Request Message Aggregate.
</PmtAckMsgRqInfo>			

7.10.2.2   Response \<PmtAckRevRs>

Tag	Type	Usage	Description
\<Status>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, \<StatusCode> defaults to 0 (zero).
\<RqUID>	UUID	Required Echoed	Request Identifier.
\<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header aggregate.
\<MsgRsHdr>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header aggregate.
\<AsyncRqUID>	UUID	Optional Echoed	Asynchronous Request Identifier.
\<CustId>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Identification Aggregate.
\<RevReasonCode>	Open Enum	Required Echoed	Reversal reason code. The reason why the client is requesting a reversal.
\<Desc>	C-80	Optional Echoed	Description, text describing the reason for the reversal.

Tag	Type	Usage	Description
<PmtAckMsgRqInfo>	Aggregate	Required Echoed	Payment Acknowledgement Message Request Information Aggregate. This aggregate contains the message to reverse. See RevOptSupt in the service profile for specific reversal messages supported.
<PmtAckInqRq>	Aggregate *see section 7.10.1.1*	Optional	Payment Acknowledgement Inquiry Request Message Aggregate.
</PmtAckMsgRqInfo>			
<CSPRefId>	Identifier	Optional	Customer Service Provider Reference Identifier.
<SPRefId>	Identifier	Optional	Service Provider Reference Identifier.

7.10.3   Payment Acknowledgment Advise

PmtAckAdvise message is used to acknowledge the receipt and application processing of a single or group of payments processed within the Pay Service.

7.10.3.1   Request <PmtAckAdviseRq>

Tag	Type	Usage	Description
<RqUID>	UUID	Required	Request Identifier. Sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header Aggregate.
<CustId>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<PmtAckRec>	Aggregate *see section 7.3.10*	Required	Payment Acknowledgment Record Aggregate. Contains acknowledgment records at a consolidated or summarized level applicable to a <SvcRqUID>, or batch of payments submitted by the customer.

7.10.3.2   Response <PmtAckAdviseRs>

Tag	Type	Usage	Description
<Status>	Aggregate *see section 3.2.12.1*	Required	Response Status Aggregate. If this aggregate is absent, <StatusCode> defaults to 0 (zero).
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header Aggregate.
<MsgRsHdr>	Aggregate *see section 3.2.1.2*	Optional Echoed	Message Response Header Aggregate
<RqUID>	UUID	Required Echoed	Request Identifier. Sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.

<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header Aggregate.
<CustId>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<PmtAckRec>	Aggregate *see section 7.3.10*	Required Echoed	Payment Acknowledgment Record Aggregate. Contains acknowledgment records at a consolidated or summarized level applicable to a <SvcRqUID>, or batch of payments submitted by the customer.
<CSPRefId>	Identifier	Optional Repeating	Customer Service Provider Reference Identifier. <CSPRefId> is used to inquire about a payment corresponding to a confirmation number that was returned to the client when the payment was added or modified. When a payment has been modified, only the <CSPRefId> received in the most recent <ChksumAdviveRs> is valid.
<SPRefId>	Identifier	Optional Repeating	Service Provider Reference Identifier. Same usage as <CSPRefId>.

7.11   Batch Checksum

7.11.1   Checksum Add

The Checksum Add message allows a client to provide a chechsum record against which the server can verify scheduled payments, where the control data such as sum amount and count are provided to the server. The Checksum Add message can be sent together with or separate from the payment messages within a pay service to facilitate both cases where the control record is sent via the same or different routes due to authorization and segregation reasons.

7.11.1.1   Request <ChksumAddRq>

Tag	Type	Usage	Description
<RqUID>	UUID	Required	Request Identifier. Sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header aggregate.
<AsyncRqUID>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously. For more information, see section 3.2.12.1.
<CustId>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.

Tag	Type	Usage	Description
<SvcRqUID>	UUID	Optional	Pay Service Request Identifier. Identifier for the payment file generated by the client. This must be the same as the <RqUID> of <PaySvcRq>, so that the server can match it to the <PaySvcRq> containing the associated payments.
<ChksumInfo>	Aggregate *see section 7.3.7.1*	Required	Payment Control Information Aggregate. Provides the checksums used by the server to validate a payment batch.

7.11.1.2   Response <ChksumAddRs>

Tag	Type	Usage	Description
<Status>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, <StatusCode> defaults to 0 (zero).
<RqUID>	UUID	Required Echoed	Request Identifier.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header aggregate.
<MsgRsHdr>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header aggregate.
<AsyncRqUID>	UUID	Optional Echoed	Asynchronous Request Identifier.
<CustId>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Identifier.
<SvcRqUID>	UUID	Optional Echoed	Pay Service Request Identifier.
<ChksumInfo>	Message	Required Echoed	Payment Control Info.
<ChksumRec>	Aggregate *see section 7.3.7*	Required	Checksum Control Record Aggregate.
<CSPRefId>	Identifier	Optional	Customer Service Reference Provider Identifier.
<SPRefId>	Identifier	Optional	Service Provider Reference Identifier.

7.11.2   Checksum Modify

7.11.2.1   Request <ChksumModRq>

Tag	Type	Usage	Description
<RqUID>	UUID	Required	Request Identifier. Sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header aggregate.

Tag	Type	Usage	Description
<AsyncRqUID>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously. For more information, see section 3.2.12.1.
<CustId>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<SvcRqUID>	UUID	Optional	Pay Service Request Identifier. Identifier for the payment file generated by the client. This must be the same as the <RqUID> of <PaySvcRq>, so that the server can match it to the <PaySvcRq> containing the associated payments.
<ChksumId>	Identifier	Required	Checksum Control Identifier. Assigned by the server at the time the Checksum Control is first added. Cannot be modified by the client.
<ChksumInfo>	Message	Required	Payment Control Information Aggregate. Provides the checksums used by the server to validate a payment batch.

7.11.2.2   Response <ChksumModRs>

Tag	Type	Usage	Description
<Status>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, <StatusCode> defaults to 0 (zero).
<RqUID>	UUID	Required Echoed	Request Identifier.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header aggregate.
<MsgRsHdr>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header aggregate.
<AsyncRqUID>	UUID	Optional Echoed	Asynchronous Request Identifier.
<CustId>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Identifier Aggregate.
<SvcRqUID>	UUID	Optional Echoed	Pay Service Request Identifier.
<ChksumId>	Identifier	Required Echoed	Checksum Control Identifier.
<ChksumInfo>	Message	Required Echoed	Payment Control Info.
<ChksumRec>	Aggregate *see section 7.3.7*	Required	Checksum Control Record Aggregate.

Tag	Type	Usage	Description
<CSPRefId	Identifier	Optional	Customer Service Provider Reference Identifier.
<SPRefId>	Identifier	Optional	Service Provider Reference Identifier.

7.11.3   Checksum Status Modify

7.11.3.1   Request <ChksumStatusModRq>

Tag	Type	Usage	Description
<RqUID>	UUID	Required	Request Identifier. Sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header aggregate.
<AsyncRqUID>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously.
			For more information, see section 3.2.12.1.
<CustId>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<SvcRqUID>	UUID	Optional	Pay Service Request Identifier. Identifier for the payment file generated by the client. This must be the same as the <RqUID> of <PaySvcRq>, so that the server can match it to the <PaySvcRq> containing the associated payments.
<ChksumId>	Identifier	Required	Checksum Control Identifier. Assigned by the server at the time the Checksum Control is first added. Cannot be modified by the client.
<ChksumStatus>	Aggregate *see section 7.3.7.2*	Required	Checksum Control Status Aggregate.

7.11.3.2   Response <ChksumStatusModRs>

Tag	Type	Usage	Description
<Status>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, <StatusCode> defaults to 0 (zero).
<RqUID>	UUID	Required Echoed	Request Identifier.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header aggregate.
<MsgRsHdr>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header aggregate.

Tag	Type	Usage	Description
<AsyncRqUID>	UUID	Optional Echoed	Asynchronous Request Identifier.
<CustId>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Identifier Aggregate.
<SvcRqUID>	UUID	Optional Echoed	Pay Service Request Identifier.
<ChksumId>	Identifier	Required Echoed	Checksum Control Identifier.
<ChksumStatus>	Aggregate *see section 7.3.7.2*	Required Echoed	Chechkum Control Status Aggregate.
<CSPRefId>	Identifier	Optional	Customer Service Provider Reference Identifier.
<SPRefId>	Identifier	Optional	Service Provider Reference Identifier.

7.11.4   Checksum Delete

7.11.4.1   Request <ChksumDelRq>

Tag	Type	Usage	Description
<RqUID>	UUID	Required	Request Identifier. Optionally sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header aggregate.
<AsyncRqUID>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously. For more information, see section 3.2.12.1.
<CustId>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<SvcRqUID>	UUID	Optional	Pay Service Request Identifier. Identifier for the payment file generated by the client. This must be the same as the <RqUID> of <PaySvcRq>, so that the server can match it to the <PaySvcRq> containing the associated payments.
<ChksumId>	Identifier	Required	Checksum Control Identifier. Assigned by the server at the time the Checksum Control is first added. Cannot be modified by the client.

7.11.4.2   Response <ChksumDelRs>

Tag	Type	Usage	Description
<Status>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, <StatusCode> defaults to 0 (zero).

Tag	Type	Usage	Description
<RqUID>	UUID	Required Echoed	Request Identifier.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header aggregate.
<MsgRsHdr>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header aggregate.
<AsyncRqUID>	UUID	Optional Echoed	Asynchronous Request Identifier.
<CustId>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Identification Aggregate.
<SvcRqUID>	UUID	Optional Echoed	Pay Service Request Identifier.
<ChksumId>	Identifier	Required Echoed	Checksum Control Identifier.
<ChksumRec>	Aggregate *see section 7.3.7*	Optional	Checksum Control Record Aggregate.
<CSPRefId>	Identifier	Optional	Customer Service Provider Reference Identifier.
<SPRefId>	Identifier	Optional	Service Provider Reference Identifier.

7.11.5   Checksum Inquiry

7.11.5.1   Request <ChksumInqRq>

Tag	Type	Usage	Description
<RqUID>	UUID	Required	Request Identifier. Sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header aggregate.
<AsyncRqUID>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously. For more information, see section 3.2.12.1.
<CustId>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<SvcRqUID>	UUID	Optional	Pay Service Request Identifier. Identifier for the payment file generated by the client. This must be the same as the <RqUID> of <PaySvcRq>, so that the server can match it to the <PaySvcRq> containing the associated payments. This is used as a selection criterion.

Tag	Type	Usage	Description
<ChksumId>	Identifier	Required	Checksum Identifier. Assigned by the server at the time the Payment is first added. Cannot be modified by the client.
			This is used as a selection criterion.
<ChksumStatusCode>	Open Enum	Required	Checksum Status Code. This identifies the remittance status.
			Defined values: Pending, Posted, Refused, Rejected, Returned, DelPend, Deleted
			This is used as a selection criterion.

7.11.5.2   Response <ChksumInqRs>

Tag	Type	Usage	Description
<Status>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, <StatusCode> defaults to 0 (zero).
<RqUID>	UUID	Required Echoed	Request Identifier.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header aggregate.
<MsgRsHdr>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header aggregate.
<AsyncRqUID>	UUID	Optional Echoed	Asynchronous Request Identifier.
<CustId>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Identification Aggregate.
<SvcRqUID>	UUID	Optional Echoed	Pay Service Request Identifier.
<ChksumId>	Identifier	Required Echoed	Checksum Identifier.
<ChksumStatusCode>	Open Enum	Required Echoed	Checksum Status Code.
<ChksumRec>	Aggregate *see section 7.3.7*	Optional	Checksum Record Aggregate.

7.11.6   Checksum Audit

7.11.6.1   Request <ChksumAudRq>

Tag	Type	Usage	Description
<RqUID>	UUID	Required	Request Identifier. Sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header aggregate.

Tag	Type	Usage	Description
<AsyncRqUID>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously.
			For more information, see section 3.2.12.1.
<CustId>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<RecCtrlIn>	Aggregate *see section 3.2.12.2.1*	Optional	Records Control Input Aggregate.
<SelRangeDt>	Aggregate *see section 3.2.10.1*	Optional	Selection Range Date Aggregate. This is used as a selection criterion.
<Method>	Closed Enum	Optional Repeating	Method. Valid Values: Add, Mod, Can This is used as a selection criterion.
<ChksumId>	Identifier	Optional Repeating	Checksum Control Identifier. This is used as a selection criterion.

7.11.6.2   Response <ChksumAudRs>

Tag	Type	Usage	Description
<Status>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, <StatusCode> defaults to 0 (zero).
<RqUID>	UUID	Required Echoed	Request Identifier.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header aggregate.
<MsgRsHdr>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header aggregate.
<AsyncRqUID>	UUID	Optional Echoed	Asynchronous Request Identifier.
<CustId>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Identification Aggregate.
<RecCtrlOut>	Aggregate *see section 3.2.12.2.2*	Optional *but see Description*	Records Control Output Aggregate. Required if <RecCtrlIn> was provided in the request and the server supports Records Control.
<SelRangeDt>	Aggregate *see section 3.2.10.1*	Optional Echoed	Selection Range Date Aggregate.

Tag	Type	Usage	Description
<Method>	Closed Enum	Optional Repeating Echoed	Payment Method.
<ChksumId>	Identifier	Optional Echoed	Checksum Identifier.
<ChksumMsgRec>	Aggregate	Optional Repeating	Checksum Message Record Aggregate.
<CustId>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the entity who initiated the request (e.g., the one whose identifier appears in the <SignonRq> of the original issued request). For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the CSR or SP.
<MsgRecDt>	Timestamp	Optional	Audit Record Creation Date. Date when this audit record is created.
<ChksumAddRs>	Aggregate *see section 7.11.1.2*	Required XOR	Checksum Add Response Message Aggregate.
<ChksumModRs>	Aggregate *see section 7.11.2.2*	Required XOR	Checksum Modify Response Message Aggregate.
<ChksumStatusModRs>	Aggregate *see section 7.11.3.2*	Required XOR	Checksum Status Modify Response Message Aggregate.
<ChksumDelRs>	Aggregate *see section 7.11.4.2*	Required XOR	Checksum Delete Response Message Aggregate.
</ChksumMsgRec>			

7.11.7   Checksum Synchronization

7.11.7.1   Request <ChksumSyncRq>

Tag	Type	Usage	Description
<RqUID>	UUID	Required	Request Identifier. Sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header aggregate.
<AsyncRqUID>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously. For more information, see section 3.2.12.1.
<CustId>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.

Tag	Type	Usage	Description
<RecCtrlIn>	Aggregate *see section 3.2.12.2.1*	Optional	Records Control Input Aggregate.
<Token>	Identifier	Required	Token. Server assigned. Client should make no assumptions about the value of this token relative to others that it may have received. The token is only significant to the server that originally assigned it. This is a token that has been previously sent by the server, or zero for first time requests.

7.11.7.2   Response <ChksumSyncRs>

Tag	Type	Usage	Description
<Status>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, <StatusCode> defaults to 0 (zero).
<RqUID>	UUID	Required Echoed	Request Identifier.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header aggregate.
<MsgRsHdr>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header aggregate.
<AsyncRqUID>	UUID	Optional Echoed	Asynchronous Request Identifier.
<CustId>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Identification Aggregate.
<RecCtrlOut>	Aggregate *see section 3.2.12.2.2*	Optional	Records Control Output Aggregate.
<Token>	Identifier	Required Echoed	Token.
<NewToken>	Identifier	Required	New Token. Server assigned. Client should make no assumptions about the value of this token relative to others that it may have received. The token is only significant to the server that originally assigned it.
<ChksumMsgRec>	Aggregate	Optional Repeating	Checksum Message Record Aggregate.
<CustId>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the entity who initiated the request (e.g., the one whose identifier appears in the <SignonRq> of the original issued request). For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the CSR or SP.
<MsgRecDt>	Timestamp	Optional	Audit Record Creation Date. Date when this audit record is created.
<ChksumAddRs>	Aggregate *see section 7.11.1.2*	Required XOR	Checksum Add Response Message Aggregate.

Tag	Type	Usage	Description
<ChksumModRs>	Aggregate *see section 7.11.2.2*	Required XOR	Checksum Modify Response Message Aggregate.
<ChksumStatusModRs>	Aggregate *see section 7.11.3.2*	Required XOR	Checksum Status Modify Response Message Aggregate.
<ChksumDelRs>	Aggregate *see section 7.11.4.2*	Required XOR	Checksum Delete Response Message Aggregate.
</ChksumMsgRec>			

7.11.8 Checksum Advise

ChkSumAdvise message is used to advise an entity of the processing status of a batch of transactions. This message would subsequently be sent in response to the ChksumAdd message set.

7.11.8.1 Request <ChksumAdviseRq>

Tag	Type	Usage	Description
<RqUID>	UUID	Required	Request Identifier. Sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header Aggregate.
<CustId>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<SvcRqUID>	UUID	Optional	Pay Service Request Identifier. Identifier for the payment file generated by the client. This must be the same as the <RqUID> of <PaySvcRq>, so that the server can match it to the <PaySvcRq> containing the associated payments.
<ChksumRec>	Aggregate *see section 7.3.7*	Required XOR	Checksum Control Record Aggregate.
<ChksumId>	Identifier	Required XOR	Checksum Identifier. Assigned by the server at the time the Payment is first added.
<ChksumStatus>	Aggregate *see section 7.3.7.2*	Required	Checksum Status Aggregate.

7.11.8.2 Response <ChksumAdviseRs>

Tag	Type	Usage	Description
<Status>	Aggregate *see section 3.2.12.1*	Required	Response Status Aggregate. If this aggregate is absent, <StatusCode> defaults to 0 (zero).
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header Aggregate.

<MsgRsHdr>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header Aggregate
<RqUID>	UUID	Required Echoed	Request Identifier. Sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header Aggregate.
<CustId>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<SvcRqUID>	UUID	Optional Echoed	Pay Service Request Identifier. Identifier for the payment file generated by the client. This must be the same as the <RqUID> of <PaySvcRq>, so that the server can match it to the <PaySvcRq> containing the associated payments.
<ChksumRec>	Aggregate *see section 7.3.7*	Required XOR Echoed	Checksum Control Record Aggregate.
<ChksumId>	Identifier	Required XOR Echoed	Checksum Identifier. Assigned by the server at the time the Payment is first added.
<ChksumStatus>	Aggregate *see section 7.3.7.2*	Required Echoed	Checksum Status Aggregate.
<CSPRefId>	Identifier	Optional Repeating	Customer Service Provider Reference Identifier. <CSPRefId> is used to inquire about a payment corresponding to a confirmation number that was returned to the client when the payment was added or modified. When a payment has been modified, only the <CSPRefId> received in the most recent <ChksumAdviveRs> is valid.
<SPRefId>	Identifier	Optional Repeating	Service Provider Reference Identifier. Same usage as <CSPRefId>.

7.12   Comprehensible Remittance

In order to facilitate collection of check payments made to the biller, lockbox relationships are established with a bank(s) throughout a regional location. The lockbox addresses are generally located in major cities in the region, i.e., San Francisco, Chicago, Dallas, Atlanta, etc. Boxes are at the post office site with direct delivery of items to the applicable bank lockbox processing site. Multiple deliveries can be made throughout the day.

The lockbox sites take a delivery from the post office and process the data, identifying each delivery as a batch. There are multiple cutoff times during the day for this processing which can result in multiple deposits being posted to an account. A batch or multiple batches are processed and posted to the biller's account as a lockbox deposit. The batch information includes a batch number, item count of checks, batch total amount, processing date and lockbox number.

The lockbox site process includes the following:

- Opening envelopes containing check and one or more documents (invoice, correspondence, etc.)
- Data input or scanning of check information with verification of specific data either by data input or voice activation

- Data input or scanning of remittance detail associated with a check (based on instructions agreed out of band between biller and bank)

- Imaging of check and documents associated with check

- Mailing of copy of image of check with documents to biller (ranges from overnight to a infinite number of days based on agreement)

- Providing biller with CD Rom with images of check and documents or URL to viewing and/or obtain images on-line

The biller imports electronic data into their Accounts Receivable data base to perform cash posting processes. Utilizing identified data elements to successfully match payment with outstanding bills on the account, cash posting can result in a successful match and removal of the receivables item(s). Failures would result in exception processing requiring a copy of the check and/or documents sent with the check or obtaining the image of the check and associated documents. This process can also be facilitated via manual input by the Cash Application department.

Processes also exist to post the payment on the AR account if the appropriate MICR information is matched. This includes the Payor Bank Id and the Payor Account Number. Individual items are still required to be reconciled, but the payment is posted on the account for credit facilitation and statistical calculations.

The Treasury and/or Cash Application departments at the biller require the batch information to reconcile the number of items processed at the lockbox site vs. the number of items processed during the cash application (whether automated or manual). Dependent upon the posting process, the total amount of the batches are also reconciled to the posted deposit amount on the account.

Services are provided by various FI's to send a message to a client containing all of the receipts information received from various sources: ACH, lockbox, and wires. Therefore, a comprehensible receipts message could be facilitated from a lockbox remittance message.

Depicted below is the flow for the process:
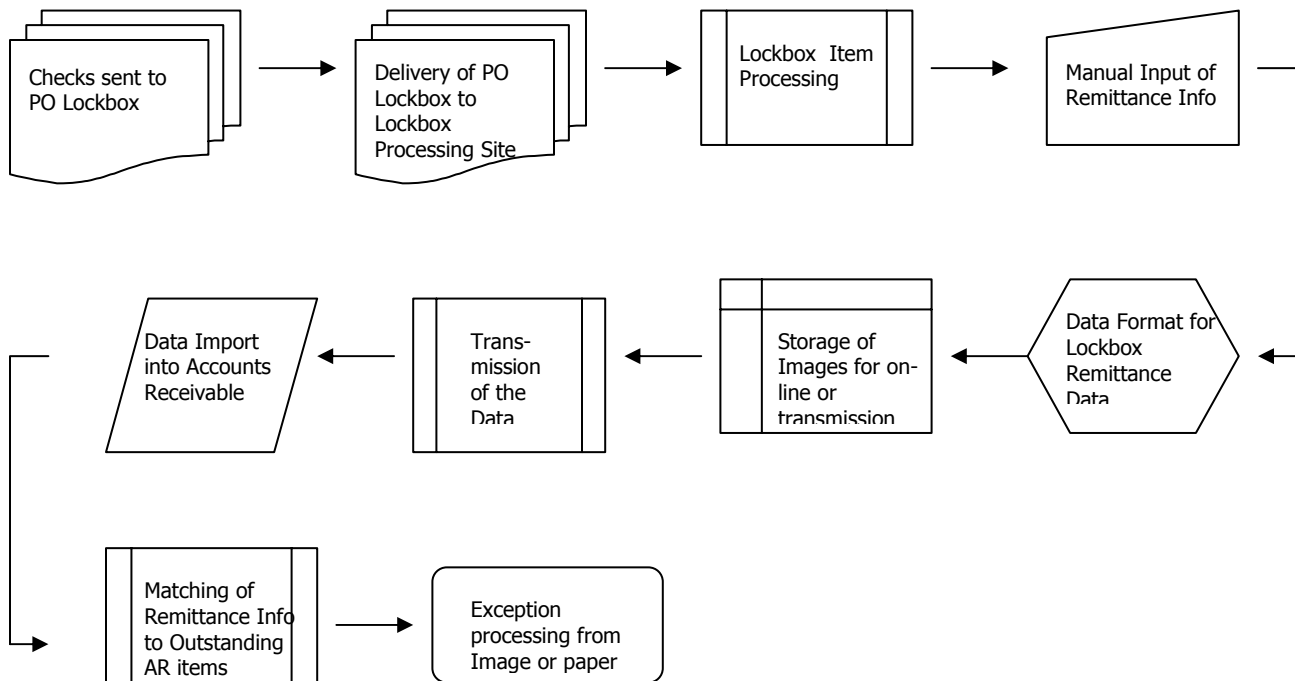
7.12.1 Comprehensible Remittance Statement Add

The Comprehensible Remittance Statement Add message allows a financial institution to transmit lockbox remittance information along with electronic payment remittance information (ACH, low value, wires, or high value) to a client for posting.

7.12.1.1 Request <CompRemitStmtAddRq>

Tag	Type	Usage	Description
<RqUID>	UUID	Required	Request Identifier. Sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header aggregate.
<AsyncRqUID>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously. For more information, see section 3.2.12.1.
<CustId>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<EffDt>	DateTime	Required	Effective Date. The date the remittance information was processed by the FI or CPP.
<CompRemitStmtInfo>	Aggregate *see section 7.3.8.1*	Required	Comprehensible Remittance Statement Information Aggregate.
<DupChkOverride>	Boolean	Optional	Duplicate Check Override Flag. When set to True, requests that the server not perform duplicate checking if any is normally performed. The client is affirming that this is a new remittance being added.

7.12.1.2 Response<CompRemitStmtAddRs>

Customers provide an acknowledgement to an FI initiated <CompRemitStmtAddRq> by sending a <CompRemitStmtAddRs> message.

Tag	Type	Usage	Description
<Status>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, <StatusCode> defaults to 0 (zero).
<RqUID>	UUID	Required Echoed	Request Identifier. Sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header aggregate.
<MsgRsHdr>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header aggregate.

Tag	Type	Usage	Description
\<AsyncRqUID\>	UUID	Optional Echoed	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously.
			For more information, see section 3.2.12.1.
\<CustId\>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Identification Aggregate.
\<EffDt\>	DateTime	Required Echoed	Effective Date. The date the remittance information was processed by the FI or CPP.
\<CompRemitStmtInfo\>	Aggregate *see section 7.3.8.1*	Required Echoed	Comprehensible Remittance Statement Information Aggregate.
\<CompRemitStmtRec\>	Aggregate *see section 7.3.8*	Required	Comprehensible Remittance Statement Record Aggregate.
\<CSPRefId\>	Identifier	Optional	Customer Service Provider Reference Identifier.
\<SPRefId\>	Identifier	Optional	Service Provider Reference Identifier.

7.12.2   Comprehensible Remittance Statement Inquiry

The Comprehensible Remittance Statement Inquiry message allows a customer to inquiry about remittance information on a specific lockbox or deposit account.

7.12.2.1   Request \<CompRemitStmtInqRq\>

Tag	Type	Usage	Description
\<RqUID\>	UUID	Required	Request Identifier. Sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.
\<MsgRqHdr\>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header aggregate.
\<AsyncRqUID\>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously.
			For more information, see section 3.2.12.1.
\<CustId\>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in \<SignonRq\>. For example, if a CSR or SP issues the request on behalf of the user, then \<CustId\> is required, and must contain the value of the user whose request is being issued.
\<RecCtrlIn\>	Aggregate *see section 3.2.12.2.1*	Optional	Records Control Input Aggregate.
\<LockboxDepId\>	Aggregate *see section 7.3.8.1.1*	Optional Repeating	Lockbox Deposit Identifier.
\<DepAcctId\>	Aggregate *see section 3.2.7.1.2*	Optional Repeating	Deposit Account Id Aggregate.

Tag	Type	Usage	Description
\<SelRangeCurAmt\>	Aggregate *see section 3.2.10.2*	Optional	Selection Range Currency Amount. Used as a selection criterion.
\<SelRangeDt\>	Aggregate *see section 3.2.10.1*	Optional	Date Selection Range. Used as a selection criterion.
\<IncToken\>	Boolean	Optional	Include Token. If *True*, a \<Token\> should be included in the response to set a base for future Sync messages. If *False* or omitted, no \<Token\> is returned.
\<DeliveryMethod\>	Open Enum	Optional Profiled Values	Delivery Method. Default is Channel. Value must be supported in Service Profile.

7.12.2.2   Response\<CompRemitStmtInqRs\>

Tag	Type	Usage	Description
\<Status\>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, \<StatusCode\> defaults to 0 (zero).
\<RqUID\>	UUID	Required Echoed	Request Identifier. Sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.
\<MsgRqHdr\>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header aggregate.
\<MsgRsHdr\>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header aggregate.
\<AsyncRqUID\>	UUID	Optional Echoed	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously. For more information, see section 3.2.12.1.
\<CustId\>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Identification Aggregate.
\<RecCtrlOut\>	Aggregate *see section 3.2.12.2.2*	Optional *but see Description*	Records Control Output Aggregate. Required if \<RecCtrlIn\> was provided in the request and the server supports Records Control.
\<LockboxDepId\>	Aggregate *see section 7.3.8.1.1*	Optional Repeating Echoed	Lockbox Deposit Identifier.
\<DepAcctId\>	Aggregate *see section 3.2.7.1.2*	Optional Repeating Echoed	Deposit Account Id Aggregate.
\<SelRangeCurAmt\>	Aggregate *see section 3.2.10.2*	Optional Echoed	Selection Range Currency Amount.
\<SelRangeDt\>	Aggregate *see section 3.2.10.1*	Optional Echoed	Date Selection Range.

Tag	Type	Usage	Description
<DeliveryMethod>	Open Enum	Optional Echoed	Delivery Method.
<CompRemitStmtRec>	Aggregate *see section 7.3.8*	Required	Comprehensible Remittance Statement Record Aggregate.
<Token>	Identifier	Optioinal	Token. Server Assigned. Client should make no assumptions about the value of this token relative to others that it may have received. The token is significant onlyl to the server that originally assigned it.

The token is only included if <IncToken>=True in the request. <Token>=0 is returned if no records are returned within the response. |

7.12.3   Comprehensible Remittance Statement Audit

A customer originates a Comprehensible Remittance Statement audit request to obtain all changes related to a particular comprehensible remittance addition by the FI.

7.12.3.1   Request <CompRemitStmtAudRq>

Tag	Type	Usage	Description
<RqUID>	UUID	Required	Request Identifier. Sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header aggregate.
<AsyncRqUID>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously.

For more information, see section 3.2.12.1. |
<CustId>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<RecCtrlIn>	Aggregate *see section 3.2.12.2.1*	Optional	Records Control Input Aggregate.
<SelRangeDt>	Aggregate *see section 3.2.10.1*	Optional	Date Selection Range. Used as a selection criterion.
<Method>	Closed Enum	Optional Repeating	Audit Selection Action. Used to identify actions associated with the object that is being audited

Valid values: Add, Mod, Delete.

This field is used as a selection criterion. |
| <LockboxDepId> | Aggregate *see section 7.3.8.1.1* | Optional Repeating | Lockbox Deposit Identifier.

This field is used as a selection criterion. |

Tag	Type	Usage	Description
<DepAcctId>	Aggregate *see section 3.2.7.1.2*	Optional Repeating	Deposit Account Id Aggregate. This field is used as a selection criterion.

7.12.3.2 Response<CompRemitStmtAudRs>

Tag	Type	Usage	Description
<Status>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, <StatusCode> defaults to 0 (zero).
<RqUID>	UUID	Required Echoed	Request Identifier. Sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header aggregate.
<MsgRsHdr>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header aggregate.
<AsyncRqUID>	UUID	Optional Echoed	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously. For more information, see section 3.2.12.1.
<CustId>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Identification Aggregate.
<RecCtrlOut>	Aggregate *see section 3.2.12.2.2*	Optional *but see Description*	Records Control Output Aggregate. Required if <RecCtrlIn> was provided in the request and the server supports Records Control.
<SelRangeDt>	Aggregate *see section 3.2.10.1*	Optional Echoed	Date Selection Range.
<Method>	Closed Enum	Optional Repeating Echoed	Audit Selection Action. Used to identify actions associated with the object that is being audited.
<LockboxDepId>	Aggregate *see section 7.3.8.1.1*	Optional Repeating Echoed	Lockbox Deposit Identifier.
<DepAcctId>	Aggregate *see section 3.2.7.1.2*	Optional Repeating Echoed	Deposit Account Id Aggregate.
<CompRemitStmtMsgRec>	Aggregate	Optional Repeating	Comprehensive Remittance Statement Message Record Aggregate
<CustId>	Aggregate *see section 3.2.3.1*	Optioinal	Customer Identification Aggregate. This is the identifier of the entity who initiated the request (e.g., the one whose identifier appears in the <SignonRq> of the original issued request). For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the CSR or SP.

Tag	Type	Usage	Description
<MsgRecDt>	DateTime	Optional	Audit Record Creation Date. Date/time at which the audit record was stored/created by the service provider.
<CompRemitStmtAddRs>	Aggregate *see section 7.12.1.2*	Required	Comprehensible Remittance Statement Add Response Message Aggregate. One record is returned for each Comprehensible Remittance Statement message, which meets the selection criteria specified in the request.
</CompRemitStmtMsgRec >			

7.12.4   Comprehensible Remittance Statement Synchronization

7.12.4.1   Request <CompRemitStmtSyncRq>

Tag	Type	Usage	Description
<RqUID>	UUID	Required	Request Identifier. Sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header aggregate.
<AsyncRqUID>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously. For more information, see section 3.2.12.1.
<CustId>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<RecCtrlIn>	Aggregate *see section 3.2.12.2.1*	Optional	Records Control Input Aggregate.
<LockboxDepId>	Aggregate *see section 7.3.8.1.1*	Required XOR	Lockbox Deposit Identifier. This field is used as a selection criterion.
<DepAcctId>	Aggregate *see section 3.2.7.1.2*	Required XOR	Deposit Account Id Aggregate. This field is used as a selection criterion.
<Token>	Identifier	Required	Token. Server assigned. Client should make no assumptions about the value of this token relative to others that it may have received. The token is significant only to the server that originally assigned it. This is a token that has been previously sent by the server, or zero for first time requests.

7.12.4.2   Response<CompRemitStmtSyncRs>

Tag	Type	Usage	Description

Tag	Type	Usage	Description
<Status>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, <StatusCode> defaults to 0 (zero).
<RqUID>	UUID	Required Echoed	Request Identifier. Sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header aggregate.
<MsgRsHdr>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header aggregate.
<AsyncRqUID>	UUID	Optional Echoed	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously. For more information, see section 3.2.12.1.
<CustId>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Identification Aggregate.
<RecCtrlOut>	Aggregate *see section 3.2.12.2.2*	Optional *but see Description*	Records Control Output Aggregate. Required if <RecCtrlIn> was provided in the request and the server supports Records Control.
<LockboxDepId>	Aggregate *see section 7.3.8.1.1*	Required XOR Echoed	Lockbox Deposit Identifier.
<DepAcctId>	Aggregate *see section 3.2.7.1.2*	Required XOR Echoed	Deposit Account Id Aggregate.
<Token>	Identifier	Required Echoed	Token. Server assigned. Client should make no assumptions about the value of this token relative to others that it may have received. The token is significant only to the server that originally assigned it. This is a token that has been previously sent by the server, or zero for first time requests.
<NewToken>	Identifier	Required	New Token. Server assigned. Client should make no assumptions about the value of this token relative to others that it may have received. The token is significant only to the server that originally assigned it.
<CompRemitStmtMsgRec>	Aggregate	Optional Repeating	Comprehensive Remittance Statement Message Record Aggregate
<CustId>	Aggregate *see section 3.2.3.1*	Optioinal	Customer Identification Aggregate. This is the identifier of the entity who initiated the request (e.g., the one whose identifier appears in the <SignonRq> of the original issued request). For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the CSR or SP.
<MsgRecDt>	DateTime	Optional	Audit Record Creation Date. Date/time at which the audit record was stored/created by the service provider.

Tag	Type	Usage	Description
\<CompRemitStmtAddRs\> **\</CompRemitStmtMsgRec\>**	Aggregate *see section 7.12.1.2*	Required	Comprehensible Remittance Statement Add Response Message Aggregate. One record is returned for each Comprehensible Remittance Statement message, which meets the selection criteria specified in the request.

7.13 Pay Service Profile \<PaySvcProfInfo\>

The profile for the Pay Service is defined below. This profile is returned to the client in the Service Profile Inquiry \<SvcProfInqRs\> response and provides information on how the client should use the Pay Service.

Tag	Type	Usage	Description
\<SvcCore\>	Aggregate *see section 5.2.2.1*	Required	Service Core Aggregate. Information specified for every service and version.
\<MsgSupt\>	Open Enum	Required Repeating	Supported Messages. This is a list of messages that are supported for Pay. The convention is to use the name of the message without the Rq or Rs so that each message is only listed once.
			Defined values: StdPayeeInq, CustPayeeAdd, CustPayeeMod, CustPayeeTypeMod, CustPayeeDel, CustPayeeInq, CustPayeeAud, CustPayeeSync, PmtAdd, PmtMod, PmtStatusMod, PmtStatusAdvise, PmtCan, PmtInq, PmtAud, PmtSync, PmtAuthAdd, PmtAuthMod, PmtAuthCan, PmtAuthInq, PmtAuthAud, PmtAuthSync, RemitAdd, RemitMod, RemitStatusMod, RemitDel, RemitInq, RemitAud, RemitSync, RecPmtAdd, RecPmtInstAdd, RecPmtMod, RecPmtCan, RecPmtInq, RecPmtAud, RecPmtSync, PmtAckInq, PmtAckAdvise, ChksumAdd, ChksumMod, ChksumStatusMod, ChksumDel, ChksumInq, ChksumAud, ChksumSync, ChksumAdvise, CompRemitStmtAdd, CompRemitStmtInq, CompRemitStmtAud, CompRemitStmtSync
\<OptSupt\>	Open Enum	Optional Repeating	Options Supported.
			Defined values:
			AcctNickname, BillerPayee, CustPayee, CustPayeeNickName, ForEx, FSPayee, ImmediatePmt, LineItem, PmtMultiRemit, RecCtrl, RecFinalCurAmt, RecInitialCurAmt, Skip, SuppressEcho, XferPayee
\<PrcSched\>	Aggregate *see section 5.2.2.3*	Optional	Processing Schedule Aggregate. If omitted, the default processing schedule is assumed.
\<PmtModel\>	Closed Enum	Required	Payment Model. Indicates whether the client must enter the date that the server should initiate processing of the payment \<PrcDt\> or the date that the payment is due \<DueDt\> within the Payment Record Aggregate \<PmtInfo\>.
			Valid values: DueDt, PrcDt.
\<DfltDaysToPay\>	Long	Required	Default Days to Pay. The default number of days required to complete the payment by check. Does not include transfers.
\<DfltXferDaysToPay\>	Long	Required	Default Transfer Days to Pay. The default number of days required to complete the payment by transfer.

Tag	Type	Usage	Description
\<DaysWith>	Long	Required	Withdrawal Date Offset. Used in determination of date to withdraw funds from customer account. Usage is \<DueDt>—\<DaysToPay> + \<DaysWith> provides withdrawal date.
			NOTE—If the value of \<DaysWith> is −1 then the withdrawal date is the same as \<DueDt>.
\<HistRetentionDays>	Long	Required	History Retention Days. Number of days after a message is processed that it is available for inquiries.
\<DeliveryMethod>	Open Enum	Optional Repeating	Delivery Method. Used to request the delivery channel for requested information.
			Defined values: Channel, HomeBank, Post, UPS, Courier.
			Note: although these are valid values for this element, they may not be appropriate for a particular message and may result in rejection. In that case, a response must be sent to the customer with an appropriate Status Code.
\<ModPendingType>	Closed Enum	Required	Payee Modify Pending specifies the rules used by the server to propagate changes to payees to pending single payments. Changes to payees are always propagated to payment models.
			Valid values: Always, Never, IfRequested.
\<RecPmtProf>	Aggregate	Optional	Recurring Payment Profile Aggregate.
\<Freq>	Open Enum	Required Repeating	Recurring Model Frequency. Usage is a list of supported frequencies.
			Defined values: Daily, Weekly, BiWeekly, TwiceMonthly, Monthly, EndOfMonth, FourWeeks, BiMonthly, Quarterly, SemiAnnually, Anually
\<ModPendingType>	Closed Enum	Required	Client Modify Pending Type. Specifies the rules used by the server to propagate changes to recurring models to pending payments.
			Valid values: Always, Never, IfRequested.
\</RecPmtProf>			

8 The Bill Presentment Service <PresSvc>

Bill Presentment is the electronic delivery of a bill from a biller to a customer. While this chapter focuses on the presentation of bills, this service may be used for the presentation of other documents such as notices, statements, and invoices. A company that distributes bills often has a variety of information that is to be presented to a customer, including payment information, announcements of new services, and changes in the terms and conditions of service. A biller also needs to know that they and their service providers have been able to get the bills to the customers. They often need to know that the customer has seen the bill, statement, or notice, and having information about the customer's actions returned to the biller and their service provider should facilitate business.

To support the widest intersection of customers who want bills, and billers that want bill delivery, the concept of the biller directory has been developed. Basically, the flow is that a customer's agent (the CSP) locates billers through the Biller Directory, uses the customer, customer/service link, and service/account link messages to enable presentment and payment; obtains the current list of bills available for the consumer; and allows the consumer to initiate payment. This chapter covers the methods for finding billers, obtaining bills, and updating bill statuses.

Although some billers may provide Bill Presentment service themselves, many may choose to work with a BSP that provides Bill Presentment service on behalf of many billers. For this reason, Bill Presentment focuses on connecting customers to BSPs.

8.1 Description

This section summarizes the process of receiving bills electronically, starting with the steps required to find a BSP and set up Bill Presentment service.

To receive bills electronically, a CSP determines which BSPs provide Bill Presentment service for the billers. Subsequently, the customer:

- Finds one or more billers by searching a biller directory server. The location and access to such servers is not currently defined within IFX.
- Determines which BSPs provide Bill Presentment service for the billers.
- Enables the bill presentment service with a CSP, who forwards the message on to the BSP or Biller.
- Enrolls with a CPP for a Payment service (if not already enrolled).
- Activates bill presentment with the BSP or Biller for one or more accounts with one or more billers.
- Requests electronic bills from the BSP via the CSP.
- Optionally pays the bills.

8.1.1 Biller Inquiry

To find billers, the client sends a <BillerInqRq> request. A <BillerInqRs> response is returned.

Contained within the response is the <BillerInfo> aggregate. This aggregate provides information about a biller, including preferences, such as Disclosures (see Section 5.4.1.2 for additional information). In some situations, the biller or BSP may prefer to never receive the account activation request without prior acceptance of the Biller's Bill Presentment disclosure by the customer. The <DiscReqd> tag allows the Biller to specify whether or not the customer must have accepted the Biller's disclosure prior to delivery of the account activation to the BSP. In the case where the value of this tag is True, the biller should specify the <DiscId> and <DiscDt> information. In the event a Customer Service Provider (CSP) stores Biller Disclosures locally, the <DiscDt> will allow the CSP to determine if an update to a disclosure has occurred, when processing a <BillerInqRs>. If the CSP found that an update to the disclosure had occurred, the new disclosure information can be obtained by

sending a <DiscInqRq> to the BSP, using the <DiscId> found within the <BillerInfo>. If the CSP does not store Biller disclosures locally, the <DiscInqRq> could be done at the time the CSP receives a request from their customer to activate a particular Biller. If the value of the <DiscReqd> tag is False or the tag is not present, the Biller/BSP either does not require acceptance of a disclsoure or will specify the disclosure in response to an account activation <SvcAcctAddRq>.

> *Note:* *Customer acceptance of a disclosure is associated with the Disclosure Identifier <DiscId> for that disclosure. If it is necessary to track the version of a disclosure that a customer has accepted, it is recommended that a Biller or BSP use a new <DiscId> any time that they modify their Bill Presentment disclosure. The <DiscId> itself is the versioning mechanism for disclosures within IFX.*

8.1.1.1   Search Arguments

If the client omits all elements in the <BillerInqRq>, the client is requesting a complete directory of billers. Otherwise, the client wants to filter results based on the included elements.

8.1.2   Activate Bill Delivery

Once the customer has located a biller, the customer enables the bill presentment service using the <CustSvcAddRq>. After enabling the service, the customer may activate presentment accounts for one or more billers at that BSP using the <SvcAcctAddRq> in order to receive bills. Account Activation may be an off-line process for many billers, and in such situations the <SvcAcctStatusCode> must return *xxx*Pend with an optional estimated decision date in the message. <SvcAcctInqRq> is used to request the current account status.

8.1.3   Account Inquiry

The <SvcAcctInqRs> response must return a <PresAcctId> aggregate and <SvcAcctStatus> <StatusDesc> for each of the customer's accounts with the billers at that BSP. The response may list only those accounts that have been activated for Bill Presentment service; it is not required to list all available accounts.

Unlike a financial institution, BSPs generally will not have information about all the accounts of its supported billers. Billers that also serve as their own BSPs may be able return available accounts as well as activated accounts.

8.1.4   Bill Presentment

The customer obtains bills that are ready for presentment using the <BillInqRq>. The responding <BillInqRs> includes bill summaries. Bill Summary is information from a biller that is essential for a customer to understand what is owed, which may include, but is not limited to, Amount Owed, Date Due, Biller, and Customer's Account Number with the Biller.

A customer may also elect to view an electronic version of the detail of the bill. Bill Detail is information from a biller that provides invoice line level information to a customer, such as credit card charges, telephone calls, or kilowatts used.

IFX provides an <ImageURL>, included in <StmtImage>, to indicate the location where bill detail information may be found and retrieved. The definition of structured data for communicating Bill Detail in email or other message formats is planned for future releases of the specification.

The bill record aggregate <BillRec> provides the ability to deliver bill-related information, including bill summary information about a single bill, a billing statement, a notification (textual information sent from the biller to communicate information about the bill presentment service), or an invoice, depending on the value in the <BillType> element. The Bill Summary may include such information as amount due, date due, and pointers to more information; e.g., a set of URLs that may be used to access bill detail and other information. The actual elements used are likely to vary depending on the Bill Type. For example, with <BillType> Notice, the biller may send <Memo> data but none of the other optional elements.

A Biller may use the <Memo> element within the Bill Summary to send human-readable text to the customer, regardless of bill type. This may include, for example, special instructions for accessing information within the bill detail when sent with a bill, or regulatory or other announcements when sent as part of a bill or statement or alone as part of a notice.

The bill date, <BillDt>, is the bill's "as of" date. It is sometimes referred to as "invoice date" or "statement date" on a paper bill. It is often the date that the biller generated the bill. It is not the date on which the BSP received the bill for publication or the date that the payment is due. <DueDt> is the date used by the biller to indicate when payment is to be received according to the terms and conditions of the account.

8.1.4.1   Bill Summary Amount and Sub-Amount

8.1.4.1.1   Bill Summary Amount

The Bill Summary Amount <BillSummAmt> aggregate is used to specify any currency values for the bill, including "amount due" when applicable. If <BillType> = Bill, at least one occurrence of this aggregate must be present where the amount is a payable amount (i.e., <BillAmtType> is <Payable>).

8.1.4.1.1.1 Amount Type

The <BillSummAmt> aggregate is designed to allow maximum flexibility in specifying amounts on a bill summary. The requirements vary greatly by industry, to the extent that a single "amount due" is not sufficient. Some currency values on a bill represent options on an amount to pay (specified by the <BillSummAmtType> = Payable). Some values on the bill summary may be for information only and are not an amount to be paid (specified by the <BillSummAmtType> = InfoOnly). Other values may be presented to denote a category for a supplemental or overpayment by the consumer (specified by the <BillSummAmtType> = Supplemental). Examples for each type are:

- Payable
    a) Amount Due and Minimum Amount Due on a credit card
    b) One-month, Three-month, and Six-month premium on an insurance policy
    c) Amount Due and Late Amount Due on a utility bill.

- InfoOnly
    a) Discount amount for an early payment of a utility bill
    b) Current Charges, Current Credits, Statement Balance, and Finance Charges on a credit card.

- Supplemental
    a) Extra Principal and Extra Escrow on a mortgage
    b) Tip and Charitable Contribution on a newspaper bill.

> *Note:*   *When multiple Payable amounts are specified within a bill summary, they must be treated as mutually exclusive; i.e., only one Payable amount may be selected for payment. Supplemental amounts are not mutually exclusive; i.e., the consumer may select more than one Supplemental amount for allocating an overpayment.*

8.1.4.1.2   Bill Summary Sub-Amount

Another type of amount that may be presented is a breakdown of a total amount, typically a payable amount. The <BillSummSubAmt> aggregate allows for this breakdown, and associates the breakdown with a specific amount within a <BillSummAmt> aggregate. The amounts within the sub-amount aggregate may be designated as a category that may be selected by the consumer for allocation of their total payment (<AllocateAllowed> = *True*), or the amounts within this breakdown may be information only (<AllocateAllowed> = *False*).

> **Sub-Amount Example:** A consolidated insurance bill may include Payable amounts of 3-month premium and 6-month premium, at the <BillSummAmt> level. Within the <BillSummAmt> aggregate for the 3-month premium, the biller may include two sub-amount aggregates, one for the life insurance 3-month premium and one for the health insurance 3-month premium. A similar breakdown may be included for the 6-month premium. If the biller wants the consumer to be able to specify what portion of their total payment (perhaps if they were paying something other than the amount billed, for example) was for life and what portion was for health, he would designate these sub-amounts as <AllocateAllowed> = *True*. If the sub-amounts were included for information only, the biller would designate these amounts as <AllocateAllowed> = *False*.

The table below demonstrates some possible industry examples for using the amount types and sub-amount aggregate.

8.1.4.1.3   Amount ID

The <BillSummAmt> aggregate also allows for the BSP or biller to specify an identifier, or tag, for each amount specified. This identifier may be included in a payment message to return not only the amount paid, but also the specific amount category being paid. Using this identifier allows the consumer to specify the total amount paid, as well as how the allocation of the payment is intended.

8.1.4.1.4   Description Data

Each amount within the <BillSummAmt> aggregate must carry with it a short description <ShortDesc>. This is the description that may be displayed to the consumer to correspond with the currency amount specified, where a shortened description is needed, such as in a grid box. A more detailed description <Desc> may be specified for clarity, when the <ShortDesc> is insufficient to properly define the amount. The <Memo> tag may be used to provide Help text or terms that pertain to an amount. In addition, the aggregate allows the Biller to categorize the amount into a type, specified by an open Enum list, when possible. This would allow the CSP to have a machine-readable understanding of the amount being specified.

8.1.4.1.5   Important Notes about Bill Summary Amounts

- All values for <CurAmt> are valid, including 0 and negative amounts.

- A CSP is not obligated to display all amounts specified within the bill summary. The CSP should display the bill summary amounts, beginning with the first occurrence in the message, proceeding in order and giving priority to Payable amounts, versus InfoOnly or Supplemental amounts or amounts specified within the <BillSummSubAmt> aggregate. Therefore, each occurrence of the <BillSummAmt> aggregate should be included in the message in priority order, with the first being the most important amount to display to the consumer, the second being the next highest amount in importance, and so forth, with the last occurrence being the least important. Although it may vary by client, billers should consider that amounts specified as <BillSummAmtType> = Supplemental or specified within <BillSummSubAmt> might only be displayed when the bill is selected for payment.

The CSP or CPP may choose not to perform any edit checks on the amount actually paid by the consumer.

8.1.4.1.6   Bill Summary Amount Examples

Industry Example	Short Desc	Payable	InfoOnly	Supplemental	SubAmt	AllocateAllowed
Utilities	Amount Due	×				
	Gas				×	False
	Electric				×	False
	Charity Donation			×		
	Late Charge		×			
	Late Amount Due	×				
	Early Discount		×			
	Discount Amt Due	×				
	Early Gas				×	False
	Early Electric				×	False
Newspaper	Monthly Amount	×				
	3 Month Saver	×				
	6 Month Super-Saver	×				

	Tip			×		
	Charity Donation			×		
Mortgage	Principal		×			
	Interest		×			
	RE Taxes		×			
	Insurance		×			
	Escrow Total		×			
	PMI		×			
	Total Amnt Due	×				
	Extra Principal			×		
	Extra Escrow			×		
Credit Card	Current Balance	×				
	Minimum Amt Due	×				
	Revolving Minimum				×	True
	Long Term Minimum				×	True
	New Charges		×			
	New Credits		×			
	Finance Charges		×			
	Late Charges		×			
Insurance	3-month Premium	×				
	3-month Life				×	True
	3-month Health				×	True
	6-month Premium	×				
	6-month Life				×	True
	6-month Health				×	True
	Non-Smoker Disc%		×			

8.1.5 Bill Payment

The customer may pay the bill using the payment messages specified in Chapter 7. Note that the <BillerId> used to identify a biller may be different for presentment and payment, even if the CPP and BSP are the same organization. Note also that the <BillRec> aggregate contains both the biller presentment account number and a <BillRefInfo> data element, which are returned with the payment to facilitate correct posting of the payment by the biller's accounts receivable.

8.2 Bill Presentment Service Message Summary

Function / Message Name	Req.	Comments
Biller Inquiry **\<BillerInqRq\>** **\<BillerInqRs\>**	Yes	Allows client to view a summary of current Biller records maintained by the Pay or Presentment service provider. Payment billers may or may not have been previously added to the customer's Payee List.
Biller Reversal **\<BillerRevRq\>** **\<BillerRevRs\>**		Allows a client to reverse an action of a Biller record.
Bill Inquiry **\<BillInqRq\>** **\<BillInqRs\>**	Yes	Allows client to retrieve bills from the biller.
Bill Reversal **\<BillRevRq\>** **\<BillRevRs\>**		Allows a client to reverse an action to bills from the biller.
Bill Status Modification **\<BillStatusModRq\>** **\<BillStatusModRs\>**		Allows a client (Customer/CSP/CPP) to notify the BSP/Biller that the status of a bill or payment for the bill has changed.

8.3 Presentment Service Common Aggregates

8.3.1 Biller Record Aggregate \<BillerRec\>

The \<BillerRec\> aggregate is widely used to provide summary-level information about a Biller.

The elements, \<BillerStatus\>, \<EffDt\>, \<BSPReferTo\>, \<SPName\>, and \<OrgContact\> support the case where a Biller discontinues usage of a BSP and includes "forwarding information" if available.

Tag	Type	Usage	Description
\<BillerId\>	Aggregate *see section 8.3.1.1*	Required Repeating	Biller Identification Aggregate.
\<BillerInfo\>	Aggregate *see section 8.3.1.2*	Required	Biller Information Aggregate
\<BillerStatus\>	Aggregate *see section 8.3.1.3*	Optional	Biller Status Aggregate. If absent, the meaning of \<BillerStatus\> \<BillerStatusCode\> is Active.
\<BSPReferTo\>	Aggregate	Optional	BSP Refer to Aggregate. The new BSP for this Biller, if known.
\<SPName\>	Identifier	Optional	Service Provider Name. Name of replacement BSP.
\<OrgContact\>	Aggregate *see section 3.2.5.2*	Optional	Contact information for replacement BSP.
\</BSPReferTo\>			

8.3.1.1 Biller Identification <BillerId>

The <BillerId> aggregate is widely used to uniquely identify a Biller within a BSP (i.e., the <SPName> would be the BSP).

The <StdPayeeId> is the Biller's ID as known to the CPP; i.e., the <SPName> would be the CPP. Billers may be known differently by different organizations, so when both <BillerId> and <StdPayeeId> are used in a message (such as in those messages that use the <PresAcctId> aggregate) the <BillerId> is the Biller's ID as known by the BSP and the <StdPayeeId> is the Biller's ID as known by the CPP. The presence of these two IDs may help in resolving the identification of Billers known differently by other organizations.

Tag	Type	Usage	Description
<SPName>	Identifier	Required	Service Provider Name. Used to qualify <BillerNum>. This is the name of the BSP that assigned <BillerNum>.
<BillerNum>	Identifier	Required	Biller Number. Assigned by the Pay/Presentment provider. Cannot be changed by the client.

8.3.1.2 Biller Information Aggregate <BillerInfo>

Tag	Type	Usage	Description
<Name>	C-40	Required	Biller Name. Assigned by the service provider. Cannot be changed by the client.
<BillerContact>	Aggregate *see section 3.2.6.1*	Optional	Biller Contact Information.
<IndustId>	Aggregate *see section 3.2.13*	Optional	Industry Identifier.
<SecretPrompt>	Aggregate *see section 3.2.14.1*	Optional Repeating AND	Secret Prompt Aggregate. If omitted, the biller does not require the customer to enter any secrets for client enrollment.
<CryptType>	Open Enum	Optional Repeating AND	Encryption type to indicate encryption used for transmitting authentication information. The Biller may specify one or more encryption types that it accepts. Defined values: None, PKCS#1, PKCS#7, PKCS#10, ECB, CBC, CFB, OFB, TECB, TCBC, TCBC-I, TCFB, TCFB-P, TOFB, TOFB-I, AECB, ACBC, ACFB, AOFB, ACM
<HistRetentionDays>	Long	Optional	Number of days that Bill Summary and Bill Detail information is available for inquiries. The Bill Detail information may be available for a longer period of time.
<CSPCustInfoReq>	Boolean	Optional	CSP Customer name and address information Required. If *True*, the Biller requires that the Customer name and address be sent with an account activation. IF *False* or omitted, the customer name and address are not required in the account activation.
<BillerPayInfo>	Aggregate *see section 3.2.6.2*	Optional	Biller Pay Information Aggregate.
<LogoURL>	URL	Optional	Logo URL. URL of the biller's logo.
<Logo>	Binary	Optional	Biller Logo. If the client requested images, the logo should be included here in this response.

Tag	Type	Usage	Description
\<BillerEnrollURL\>	URL	Optional	Biller Information URL.
			URL of human-readable description of additional information the biller would like the customer to have with regard to signing up.
			The resource may also include an interactive session to verify a customer's identity and eligiblity to receive bills for an account. If successful, the session results in a security token or password that the customer may use in response to a \<SecretPrompt\> when activating the account for bill presentment.
\<BillerAcctIdInfo\>	Aggregate *see section 8.3.2*	Optional	Biller Account Information Aggregate. Provides additional information to the customer to assist in entry of the customer's account numbers with the biller.
\<DiscReqd\>	Boolean	Optional	Disclosure Required. If *True*, the biller requires that their disclosure be presented to the customer prior to account activation. If *False*, the presentment of the disclosure is not required prior to the account activation request. This may be *False* if no disclosure is required or when the biller wishes to return the disclosure in the account activation response.
\<DiscId\>	UUID	Optional	Disclosure Identifier. The identifier of the disclosure associated with this biller. The \<DiscInfo\> associated with this identifier can be obtained by performing a Disclosure Inquiry.
\<DiscDt\>	Date	Optional	Date Last Disclosure Change.

8.3.1.3   Biller Status Aggregate \<BillerStatus\>

Tag	Type	Usage	Description
\<BillerStatusCode\>	Closed Enum	Required	Biller Status Code.
			Valid Values: Available, AvailPend, Deleted, DelPend, Suspended.
\<StatusDesc\>	C-255	Optional	Status Description. Explanatory text associated with this status.
\<EffDt\>	DateTime	Optional *but see Description*	Date Time Effective. The Date the \<BillerStatus\> was changed or, in the case of DeletePending, the date the delete should take effect.
			Required if \<BillerStatusCode\> = *xxx*Pend.
\<StatusModBy\>	Open Enum	Optional	Status Modified By. If present, indicates who modified the Biller Status Code.
			Defined values: Customer, CSP, CSPSR, BSP, BSPSR, CPP, CPPSR, BPP, and BPPSR.
\<Memo\>	C-255	Optional	Memo Information. To provide additional information about the status.

8.3.2   Biller Account Identification Information Aggregate \<BillerAcctIdInfo\>

The \<BillerAcctIdInfo\> aggregate specifies the format of valid account numbers acceptable to the biller.

Tag	Type	Usage	Description

Tag	Type	Usage	Description
<AcctFormat>	NC-1024	Optional	Account Format.
			Regular expression describing the account number format. The definition and behavior of "Regular Expression" is per IEEE Std 1003.2-1992 (POSIX.2). General definition may be found at http://www.ciser.cornell.edu/info/regex.html
<AcctMask>	NC-32	Optional Repeating	Account Mask.
			String describing the edit mask to identify a valid billing account number <BillingAcct> for that biller. The client uses the account edit mask to assist the user in entering the account number.
<AcctHelpMsg>	C-∞	Optional	Account Help Message.
			Human-readable message that the client may display to assist the customer in entering his or her account number.
<AcctRestrictMsg>	C-1024	Optional	Account Restriction Message.
			Human-readable description of any restrictions on who may sign up with this biller. Enforcement of any restrictions is by the biller or the biller's agent during the <SvcAcctAddRq>. Other service providers are only responsible for communicating this information.
<AcctValidateURL>	URL	Optional	URL for Account Validation.
			URL for validation. The client application may use this to validate the customer's account number.

8.3.2.1   Account Number Validation

Servers may implement a lightweight CGI (or equivalent) to validate account numbers. The URL provided in the <AcctValidateURL> may be accessed with an HTTP GET with three arguments: <BillerId>, <AcctId> and <PostalCode>. The URL should respond with a text file that includes the following values:

Status: (Required)

- Error: An error condition (wrong number of parameters, Database error, etc.). Clarifying text may accompany the error status.
- Pass: The account number is in an acceptable form for this biller (this does not guarantee that the account will be accepted for the service).
- Fail: The account number does not correspond to an acceptable account number for this biller. Clarifying text may accompany the failed status.

Account: (Optional) The preferred format or version of the account number presented in the request.

Heading: (Optional) Additional text to help explain problems to end-users.

8.3.2.1.1   Example

```
<AcctValidateURL> = http://testit.com/validate.cgi
Client application uses HTTP GET with
"http://testit.com/validate.cgi?billerid=5454&accountnumber=123-456-
7890&customerpostalcode=12345"
```

The server would respond with one of these:

- Error:
```
Content-type: text/plain
<Status>Error</Status>
<Heading>The server is unable to process your request at this time. Please
resubmit.</Heading>
```

- Failure:
  ```
  Content-type: text/plain
  <Status>FAIL</Status>
  <Heading>123-456-7890 does not appear to be a valid account number</Heading>
  ```
- Passed:
  ```
  Content-type: text/plain
  <Status>passed</Status>
  <AcctId>1234567890</AcctId>
  ```

8.3.3 Bill Record <BillRec>

The bill record aggregate <BillRec> provides the ability to deliver bill-related information, including bill summary information about a single bill, a billing statement, a notification (textual information sent from the biller to communicate information about the bill presentment service), or an invoice, depending on the value in the <BillType> element. The Bill Summary may include such information as amount due, date due, and pointers to more information; e.g., a set of URLs that may used to access bill detail and other information. The actual elements used are likely to vary depending on the Bill Type. For example, with <BillType> Notice, the biller may send <Memo> data but none of the other optional elements.

Tag	Type	Usage	Description
<BillId>	UUID	Required	Identifier for this bill within the BSP.
<BillInfo>	Aggregate *see section 8.3.3.1*	Required	Bill Information Aggregate.
<BillStatus>	Aggregate *see section 8.3.3.2*	Optional	Bill Status Aggregate.
<BillPmtStatus>	Aggregate *see section 8.3.3.3*	Optional	Bill Payment Status Aggregate.

8.3.3.1 Bill Information Aggregate <BillInfo>

Tag	Type	Usage	Description
<BillType>	Open Enum	Required	Bill Type.
			Defined values: Bill, Statement, Notice, Invoice.
<CustId>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the customer to whom the bill is issued. If absent, the bill is issued to the customer specified in the message response.
<PresAcctId>	Aggregate *see section 3.2.7.2.1*	Required	Bill Presentment Account Identification Aggregate.
<Memo>	C-255	Optional	Biller-information displayed to user by the biller. This may be used for notice information when <BillType> = Notice.
<BillSummAmt>	Aggregate	Optional Repeating	Bill Summary Amount Aggregate. Used for any currency amounts that are being presented in the Bill Summary.
			Note: *If <BillType> = Bill, at least one occurrence of this aggregate must be present where the amount is a payable amount (<BillAmtType> = Payable).*

Tag	Type	Usage	Description
<BillSummAmtId>	Identifier	Optional	Bill Summary Amount Identifier. Biller's identifier for this currency amount. This may be returned in the payment message to identify the type of amount being paid by the consumer. This value needs to be provided by the Biller if the intent is to allow the consumer to designate the particular amount being paid (using <PaySummAmt> in <PmtInfo> aggregate).
<BillSummAmtCode>	Open Enum	Optional	Bill Summary Amount Code. Indicates the type of amount being specified in <CurAmt>, using a standard list of billing amounts that are machine-readable by the client and may be used for internal processing. The short description <ShortDesc> should be used to describe the amount in a display to the consumer.
			Defined values: TotalAmtDue, MinAmtDue, MaxAmtDue, LateAmtDue, DiscAmtDue, MonthlyAmt, QuarterlyAmt, SemiAnnualAmt, AnnualAmt, ExtdAmt, PrevBal, Charges, Credits, StmtBal, LateChg, FinanceChg, Tip, Principal, Interest, Escrow, PMI, Donation
<ShortDesc>	C-15	Required	Short Description. A short description of the amount specified in <CurAmt> within this aggregate. To be used for display to the consumer.
<Desc>	C-80	Optional	Description. A longer description of the amount specified in <CurAmt>. To be used when the short description <ShortDesc> may be insufficient to clearly describe the amount.
<Memo>	C-255	Optional	Memo. Additional information about the amount specified in <CurAmt>. This may be used to further describe terms or instructions that may apply to the amount specified.
<CurAmt>	Currency Amount	Required	Currency Amount. The amount being specified as described by the short description. This amount may be a zero or negative value.
<BillSummAmtType>	Closed Enum	Required	Bill Summary Amount Type. Indicates the type of amount specified in <CurAmt>. Note that the client must treat amounts that are Payable as mutually exclusive (i.e., the consumer may select only one). Amounts that are Supplemental are not mutually exclusive (i.e., the consumer may select more than one for indicating breakdown of an additional payment).
			Valid values: Payable, Supplemental, InfoOnly.
<BillSummSubAmt>	Aggregate	Optional Repeating	Bill Summary Sub-Amount. Allows for the specification of a breakdown of an amount specified in <CurAmt> within the higher level aggregate <BillSummAmt>.
<BillSummAmtId>	Identifier	Required	Bill Summary Amount Identifier. Biller's identifier for this currency sub-amount. Used in a payment message to allocate a portion of a total payment amount to this sub-amount.
<ShortDesc>	C-15	Required	Short Description. A short description of the amount specified in <CurAmt> within this aggregate. To be used for display to the consumer.
<Desc>	C-80	Optional	Description. A longer description of the amount specified in <CurAmt>. To be used when the short description <ShortDesc> may be insufficient to clearly describe the amount.
<Memo>	C-255	Optional	Memo. Additional information about the amount specified in <CurAmt>. This may be used to further describe terms or instructions that may apply to the amount specified.

Tag	Type	Usage	Description
<CurAmt>	Currency Amount	Required	Currency Amount. The amount being specified as described by the short description. This amount may be a zero or negative value.
<AllocateAllowed>	Boolean	Optional	Allocation Allowed Indicator. If *True*, the <BillSummAmtId> for the amount specified may be used to indicate the allocation breakdown of the total payment amount within the <PayInfo> aggregate in a payment message. If *False* or omitted, the amount specified is for information only.
</BillSummSubAmt>			
</BillSummAmt>			
<DueDt>	Date	Optional	Payment due date. If absent and <BillType> is Bill, the meaning is that the bill is due upon receipt.
<BillDt>	Date	Required	Bill date.
<OpenDt>	Date	Optional	Opening statement date.
<CloseDt>	Date	Optional	Closing statement date.
<PmtInst>	Aggregate *see section 3.2.15*	Optional Repeating	Payment Instrument Aggregate. Types of payment that the biller accepts via the Pay provider. This aggregate may be used to override types of payments options indicated previously in the <BillerRec>. Note that a restriction here may result in no valid payment type being available to the customer, which may result in the CSP sending a <BillStatusModRq> to the BSP indicating that the bill is Unpayable.
<NotifyReqd>	Boolean	Optional	If *True*, the server requires that the client send a <BillStatusModRq> for each change of status of either the bill or its payment. The server may not send *True* if the client did not set <NotifyWilling> to *True* in <BillInqRq>.
<ViewDtlPref>	Open Enum	Optional	Indicates biller preference that customer view the detail of the bill. Defined values: None, Preferred. Bill detail is available through the <URL> specified within the <StmtImage> aggregate.
<StmtImage>	Aggregate *see section 8.3.3.1.1*	Optional	Statement image aggregate.
<BillRefInfo>	NC-80	Optional	Biller-defined text to include with the payment, for the biller's Accounts Receivable reconciliation. It is sent with electronic payment requests.

8.3.3.1.1 Statement Image <StmtImage>

The <StmtImage> aggregate provides one or more URLs that point to a fully rendered image of the bill, in HTML. The <URL> should include enough information for the HTTP server to authenticate the client requesting the statement image. For security reasons, the authentication information embedded within the URLs should expire after a period of time deemed prudent by the service provider. Information about the authentication information expiration date and time is included in <ExpDt>.

<ImageURL> accesses the complete bill image. This URL may contain navigation to other sites or to other pages of bill images at the same site.

To support off-line viewing of the bill, the server may provide one or more additional URLs. Each <PrefetchURL> points to a local Web page.

Tag	Type	Usage	Description

Tag	Type	Usage	Description
<ImageURL>	URL	Required	URL address for retrieving an image of the complete bill encoded as HTML. This may be cached by the client for later display, or it may be viewed live directly from the Web.
<PrefetchURL>	URL	Optional Repeating	List of URLs required in order to display an HTML image of the complete bill, to support off-line viewing.
<ExpDt>	DateTime	Optional	Date/Time after which embedded authentication token expires. If absent, the embedded token never expires.

8.3.3.2   Bill Status Aggregate <BillStatus>

Tag	Type	Usage	Description
<BillStatusCode>	Open Enum	Required	Bill Status Code. Defined values are: New, Delivered, Viewed, Retired, Withdrawn, and Undeliverable.
<StatusDesc>	C-255	Optional	Status Description. Explanatory text associated with this status.
<EffDt>	DateTime	Optional	Effective Date Time. The date and time the Bill Status became effective.
<StatusModBy>	Open Enum	Optional	Status Modified By. If present, indicates who modified the Bill Status Code.
			Defined values: Customer, CSP, CSPSR, BSP, BSPSR, CPP, CPPSR, BPP, and BPPSR.
<Memo>	C-255	Optional	Memo Information. To provide additional information about the status.

8.3.3.3   Bill Payment Status Aggregate <BillPmtStatus>

Tag	Type	Usage	Description
<PmtId>	Identifier	Optional	Payment Identifier. The ID of the payment associated with the bill.
<BillPmtStatusCode>	Open Enum	Required	Bill Payment Status Code. Valid values are: None, AutoPay, Scheduled, Processed, PaidOutOfBand, Cancelled, Unpayable, and Posted.
<EffDt>	DateTime	Optional	Effective Date Time. The date and time the Bill Payment Status became effective.
<StatusModBy>	Open Enum	Optional	Status Modified By. If present, indicates who modified the Bill Payment Status Code.
			Defined values: Customer, CSP, CSPSR, BSP, BSPSR, CPP, CPPSR, BPP, BPPSR.
<Memo>	C-255	Optional	Memo Information. To provide additional information, for example, if the payment was rejected, it may contain the reason for the rejection.

8.4   Presentment Service Message Detail

The Presentment Service contains messages to search for billers and to obtain bills.

Typically, the client periodically requests a list of bills from the BSP. The BSP responds with a list of bills; each bill contains summary data such as the due date and amount due. For each bill, the BSP might also return a <URL> to a Web site that contains an HTML-rendered version of the bill.

The server must include the BSP's <SPName> in the <BillerRec> aggregate for each biller sent in the <BillerInqRs>. The client may then use this <SPName> to activate accounts <SvcAcctAddRq>, request bills <BillInqRq>, and change status of bills <BillStatusModRq>. Since the bills for a specific customer may originate from multiple BSPs, it is the client software's responsibility to ensure that the correct <SPName> is used in each message that it originates.

8.4.1 Biller Inquiry

The Biller Inquiry message enables a client to retrieve a list of all Billers known to the BSP that meets certain selection criteria.

Note that the Biller directory timestamp <UpDt> selection criterion allows the CSP to request all directory entries that have been added or changed since a point in time.

8.4.1.1 Request <BillerInqRq>

Tag	Type	Usage	Description
<RqUID>	UUID	Required	Request Identifier. Sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header aggregate.
<AsyncRqUID>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously. For more information, see section 3.2.12.1.
<CustId>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<RecCtrlIn>	Aggregate *see section 3.2.12.2.1*	Optional	Records Control Input Aggregate.
<BillerId>	Aggregate *see section 8.3.1.1*	Optional Repeating	Biller Identification. This field is used as a selection criterion.
<Name>	C-40	Optional Repeating	Biller Name. Assigned by the server at the time the Biller is first added. Cannot be changed by the client. This field is used as a selection criterion.
<PostAddr>	Aggregate *see section 3.2.3.7.1*	Optional Repeating	Biller Postal Address Aggregate.
<Phone>	Phone Number	Optional Repeating	Customer Service Telephone Number. This field is used as a selection criterion against all customer service phone numbers associated with billers.
<PostalCode>	C-11	Optional Repeating	Postal Code. This is the postal code of the billing account. This field is used as a selection criterion: (1) It may be used to limit the search to billers doing business within a limited geography, or (2) It may be used to identify the correct legal/entity or remittance address such as the cable company for a specific city, e.g. TCI- Sunnyvale CA.

Tag	Type	Usage	Description
<UpDt>	Timestamp	Optional	Biller Directory update timestamp. This is the time supplied by the server. If present, <BillerInqRs> must include at least those Billers whose information has changed or been added since <UpDt>.
<IndustId>	Aggregate *see section 3.2.13*	Optional Repeating	Industry Identifier This field is used as a selection criterion.
<IncBillerContact>	Boolean	Optional	Include Biller Contact Information. If *True*, the <BillerContact> aggregate is returned for each biller; otherwise, it is not returned.
<IncImages>	Boolean	Optional	Include Images. If *True*, the client requests that images <Logo>s be returned.
<DeliveryMethod>	Open Enum	Optional Profiled values	Delivery Method. Default is Channel. Value must be supported in Service Profile.

8.4.1.2   Response <BillerInqRs>

Tag	Type	Usage	Description
<Status>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, <StatusCode> defaults to 0 (zero).
<RqUID>	UUID	Required Echoed	Request Identifier.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header aggregate.
<MsgRsHdr>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header aggregate.
<AsyncRqUID>	UUID	Optional Echoed	Asynchronous Request Identifier.
<CustId>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<RecCtrlOut>	Aggregate *see section 3.2.12.2.2*	Optional *but see Description*	Records Control Output Aggregate. Required if <RecCtrlIn> was provided in the request and the server supports Records Control.
<BillerId>	Identifier	Optional Repeating Echoed	Biller Identification Aggregate.
<Name>	C-40	Optional Repeating Echoed	Biller Name.
<PostAddr>	Aggregate *see section 3.2.3.7.1*	Optional Repeating Echoed	Biller Postal Address Aggregate.

Tag	Type	Usage	Description
<Phone>	Phone Number	Optional Repeating Echoed	Customer Service Telephone Number.
<PostalCode>	C-11	Optional Repeating Echoed	Postal Code. This is the postal code of the billing account.
<UpDt>	Timestamp	Optional Echoed	Biller Directory timestamp.
<IndustId>	Aggregate *see section 3.2.13*	Optional Repeating Echoed	Industry Identifier
<IncBillerContact>	Boolean	Optional Echoed	Include Biller Contact Information.
<IncImages>	Boolean	Optional Echoed	Include Images.
<DeliveryMethod>	Open Enum	Optional Echoed	Delivery Method.
<NewUpDt>	Timestamp	Optional	New Biller Directory Timestamp. This is the response timestamp generated by the server.
<BillerRec>	Aggregate *see section 8.3.1*	Optional Repeating	Biller Information Aggregate.

8.4.2   Biller Inquiry Reversal

8.4.2.1   Request <BillerRevRq>

Tag	Type	Usage	Description
<RqUID>	UUID	Required	Request Identifier.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header aggregate.
<AsyncRqUID>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously.
			For more information, see section 3.2.12.1.
<CustId>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<RevReasonCode>	Open Enum	Required	Reversal reason code. The reason why the client is requesting a reversal.
<Desc>	C-80	Optional	Description, text describing the reason for the reversal.
<BillerMsgRqInfo>	Aggregate	Required	Biller Message Request Information Aggregate. This aggregate contains the message to reverse. See RevOptSupt in the service profile for specific reversal messages supported.

Tag	Type	Usage	Description
<BillerInqRq>	Aggregate *see section 8.4.1.1*	Optional	Biller Inquiry Request Message Aggregate.
</BillerMsgRqInfo>			

8.4.2.2   Response <BillerRevRs>

Tag	Type	Usage	Description
<Status>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, <StatusCode> defaults to 0 (zero).
<RqUID>	UUID	Required Echoed	Request Identifier.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header aggregate.
<MsgRsHdr>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header aggregate.
<AsyncRqUID>	UUID	Optional Echoed	Asynchronous Request Identifier.
<CustId>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Identification Aggregate.
<RevReasonCode>	Open Enum	Required Echoed	Reversal reason code. The reason why the client is requesting a reversal.
<Desc>	C-80	Optional Echoed	Description, text describing the reason for the reversal.
<BillerMsgRqInfo>	Aggregate	Required Echoed	Biller Message Request Information Aggregate. This aggregate contains the message to reverse. See RevOptSupt in the service profile for specific reversal messages supported.
<BillerInqRq>	Aggregate *see section 8.4.1.1*	Optional	Biller Inquiry Request Message Aggregate.
</BillerMsgRqInfo>			
<CSPRefId>	Identifier	Optional	Customer Service Provider Reference Identifier.
<SPRefId>	Identifier	Optional	Service Provider Reference Identifier.

8.4.3   Bill Inquiry

<BillInqRq> retrieves bills or counts of bills from the BSP. The BSP returns a <BillInqRs> that contains a list of zero or more bills, or counts of bills that match specific selection criteria.

The client requests bills from a BSP by using one or more selection criteria, including bill creation date range. To specify the date range, clients use <StartDt> and <EndDt>, which the server compares to <BillDt> within the <BillRec> aggregate.

The BSP returns information sufficient to identify the biller and provide the amount due, due date, and remittance information so that a payment may be made to the biller. The BSP does not provide a viewable form of the bill but may return a URL to an HTML rendering of the bill.

8.4.3.1   Request <BillInqRq>

Tag	Type	Usage	Description
<RqUID>	UUID	Required	Request Identifier. Sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header aggregate.
<AsyncRqUID>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously.
			For more information, see section 3.2.12.1.
<CustId>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<RecCtrlIn>	Aggregate *see section 3.2.12.2.1*	Optional	Records Control Input Aggregate. It is used in inquiry request messages to allow the client to specify a maximum number of records that the server may return.
<SPName>	Identifier	Required	Service Provider Name.
<SelRangeDt>	Aggregate *see section 3.2.10.1*	Optional	Selection Range Date Aggregate.
<BillId>	UUID	Optional Repeating	Bill Identifier.
			This field is used as a selection criterion.
<BillerId>	Aggregate *see section 8.3.1.1*	Optional Repeating	Biller Identifier Aggregate.
			This field is used as a selection criterion.
<BillType>	Open Enum	Optional Repeating	Bill Type.
			Defined values: Bill, Statement, Notice, Invoice
			This field is used as a selection criterion.
<BillStatusCode>	Open Enum	Optional Repeating	Bill Status Code.
			Defined values: New, Delivered, Viewed, Retired, Withdrawn, Undeliverable.
			This field is used as a selection criterion.
<BillPmtStatusCode>	Open Enum	Optional Repeating	Bill Payment Status Code.
			Defined values: None, Scheduled, Processed, Posted, PaidOutOfBand, AutoPay, Cancelled, Unpayable.
			This field is used as a selection criterion.
<NotifyWilling>	Boolean	Optional	Client Willing to Notify. If *True*, the client is prepared to send notifications of changes to bill status, if desired. If *False* or absent, the client cannot send notifications of bill status changes.
<IncCounts>	Boolean	Optional	Include Counts of Bills. If *True*, the client is requesting that the number of bills in each status included in the selection criteria <BillStatusCode>, <BillPmtStatusCode> is returned in the <BillCounts> aggregate in the response.

Tag	Type	Usage	Description
<IncSummary>	Boolean	Optional	Include Bill Summaries. If *True*, the client is requesting that the bill summaries for each bill as specified in the request be returned in the response, utilizing the <BillRec> aggregate.
<DeliveryMethod>	Open Enum	Optional Profiled values	Delivery Method. Default is Channel. Value must be supported in Service Profile.

8.4.3.2 Response <BillInqRs>

The <BillInqRs> may contain zero or more bill summaries <BillRec>. Each bill summary corresponds to a (usually monthly) bill. The response may contain either counts of bills or the collection of the bill summaries.

Tag	Type	Usage	Description
<Status>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, <StatusCode> defaults to 0 (zero).
<RqUID>	UUID	Required Echoed	Request Identifier.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header aggregate.
<MsgRsHdr>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header aggregate.
<AsyncRqUID>	UUID	Optional Echoed	Asynchronous Request Identifier.
<CustId>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<RecCtrlOut>	Aggregate *see section 3.2.12.2.2*	Optional *but see Description*	Records Control Output Aggregate. Required if <RecCtrlIn> was provided in the request and the server supports Records Control.
<SelRangeDt>	Aggregate *see section 3.2.10.1*	Optional Echoed	Selection Range Date Aggregate.
<BillId>	UUID	Optional Repeating Echoed	Bill Identifier.
<BillerId>	Aggregate *see section 8.3.1.1*	Optional Repeating Echoed	Biller Identifier Aggregate.
<BillStatusCode>	Open Enum	Optional Repeating Echoed	Bill Status Code. Defined values: New, Delivered, Viewed, Retired, Withdrawn, Undeliverable
<BillPmtStatusCode>	Open Enum	Optional Repeating Echoed	Bill Payment Status Code. Defined values: None, Scheduled, Processed, Posted, PaidOutOfBand, AutoPay, Cancelled, Unpayable

Tag	Type	Usage	Description
<NotifyWilling>	Boolean	Optional Echoed	Client Willing to Notify.
<IncCounts>	Boolean	Optional Echoed	Include Counts of Bills.
<IncSummary>	Boolean	Optional Echoed	Include Bill Summaries.
<DeliveryMethod>	Open Enum	Optional Echoed	Delivery Method.
<BillRec>	Aggregate *see section 8.3.3*	Optional Repeating	Bill Record Aggregate. This is a list of bills that match the selection criteria.
<BillCounts>	Aggregate	Optional Repeating	Bill Counts Aggregate.
<BillStatusCounts>	Aggregate	Optional Repeating	Bill Status Counts. The count(s) of all the bills for that customer, which may be provided by a given status(s). Note, if no selection criteria <BillStatusCode>, <BillPmtStatusCode> are specified when <IncCounts>=*True*, counts are returned for every status with a non-zero count.
<BillStatusCode>	Open Enum	Required	Bill Status Code.
			Defined values: New, Delivered, Viewed, Retired, Withdrawn, Undeliverable
<Count>	Long	Required	Count of Bills with the given Bill Status Code.
</BillStatusCounts>			
<BillPmtStatusCounts>	Aggregate	Optional Repeating	Bill Payment Status Counts. The count(s) of all the bill payment statuses for that customer, which may be provided by a given status(s). Note selection criteria is ignored when <IncCounts>=*True*.
<BillPmtStatusCode>	Open Enum	Required	Bill Payment Status Code.
			Defined values None, Scheduled, Processed, Posted, PaidOutOfBand, AutoPay, Cancelled, Unpayable
<Count>	Long	Required	Count of Bills with the given Bill Payment Status Code.
</BillPmtStatusCounts>			
</BillCounts>			

8.4.4   Bill Reversal

8.4.4.1   Request <BillRevRq>

Tag	Type	Usage	Description
<RqUID>	UUID	Required	Request Identifier.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header aggregate.
<AsyncRqUID>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously.
			For more information, see section **3.2.12.1**.

Tag	Type	Usage	Description
<CustId>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<RevReasonCode>	Open Enum	Required	Reversal reason code. The reason why the client is requesting a reversal.
<Desc>	C-80	Optional	Description, text describing the reason for the reversal.
<BillMsgRqInfo>	Aggregate	Required	Bill Message Request Information Aggregate. This aggregate contains the message to reverse. See RevOptSupt in the service profile for specific reversal messages supported.
<BillInqRq>	Aggregate *see section 8.4.3.1*	Optional	Bill Inquiry Request Message Aggregate.
</BillMsgRqInfo>			

8.4.4.2   Response <BillRevRs>

Tag	Type	Usage	Description
<Status>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, <StatusCode> defaults to 0 (zero).
<RqUID>	UUID	Required Echoed	Request Identifier.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header aggregate.
<MsgRsHdr>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header aggregate.
<AsyncRqUID>	UUID	Optional Echoed	Asynchronous Request Identifier.
<CustId>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Identification Aggregate.
<RevReasonCode>	Open Enum	Required Echoed	Reversal reason code. The reason why the client is requesting a reversal.
<Desc>	C-80	Optional Echoed	Description, text describing the reason for the reversal.
<BillMsgRqInfo>	Aggregate	Required	Bill Message Request Information Aggregate. This aggregate contains the message to reverse. See RevOptSupt in the service profile for specific reversal messages supported.
<BillInqRq>	Aggregate *see section 8.4.3.1*	Optional	Bill Inquiry Request Message Aggregate.
</BillMsgRqInfo>			
<CSPRefId>	Identifier	Optional	Customer Service Provider Reference Identifier.
<SPRefId>	Identifier	Optional	Service Provider Reference Identifier.

8.4.5   *Bill Status Modify*

The BSP may request the client to send notifications of various state changes for the bill of the associated payment by setting <NotifyReqd> = *True* in the <BillRec> aggregate (see Section 15.3.2), if the client has indicated that it is capable of sending notifications; i.e., <NotifyWilling> = *True* within <BillInqRq>. The following table indicates which entity(s) sets each bill and bill payment status.

Status Code	Recognized/Set by
<BillStatusCode> (Status of bill):	
New	BSP/Biller (initially)
Delivered	BSP/Biller
Viewed	Customer or CSP by inference
Retired	Customer
Withdrawn	BSP/Biller
Undeliverable	CSP
<BillPmtStatusCode> (Status of a bill payment)	
None	BSP/Biller
AutoPay	BSP/Biller
Scheduled	CSP/CPP
Processed	CSP/CPP
Posted	BSP/Biller
PaidOutOfBand	Customer or CSP/CPP
Cancelled	Customer or CSP/CPP
Unpayable	CSP/CPP

Setting <BillStatusCode>=Viewed tells the BSP that the CSP has presented the specified bills to the customer. This is a stronger statement than acknowledging that the client has received the bills <BillStatusCode>=Delivered, specifically when the client software implements the pre-fetching (i.e. pull) model.

However, IFX does not define the meaning of "presenting to the customer." In particular, receipt of a <BillStatusCode>=Viewed by the BSP is not intended to have any legal significance. The specification also does not define the maximum elapsed time between the presentation of the bill and sending a notification.

8.4.5.1   Request <BillStatusModRq>

Tag	Type	Usage	Description
<RqUID>	UUID	Required	Request Identifier.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header aggregate.
<AsyncRqUID>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously. For more information, see section 3.2.12.1.

Tag	Type	Usage	Description
<CustId>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<BillId>	UUID	Required	Bill Identifier. Identifies the bill from the given biller.
<BillStatus>	Aggregate *see section 8.3.3.2*	Required OR	Bill Status Aggregate.
<BillPmtStatus>	Aggregate *see section 8.3.3.3*	Required OR	Bill Payment Status Aggregate.

8.4.5.2   Response <BillStatusModRs>

Tag	Type	Usage	Description
<Status>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, <StatusCode> defaults to 0 (zero).
<RqUID>	UUID	Required Echoed	Request Identifier.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header aggregate.
<MsgRsHdr>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header aggregate.
<AsyncRqUID>	UUID	Optional Echoed	Asynchronous Request Identifier.
<CustId>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<BillId>	UUID	Required Echoed	Bill Identifier.
<BillStatus>	Aggregate *see section 8.3.3.2*	Required OR Echoed	Bill Status Aggregate.
<BillPmtStatus>	Aggregate *see section 8.3.3.3*	Required OR Echoed	Bill Payment Status Aggregate.
<CSPRefId>	Identifier	Optional	Financial Institute Reference Identifier.
<SPRefId>	Identifier	Optional	Service Provider Reference Identifier.

8.5 Bill Presentment Service Profile <PresSvcProfInfo>

This section defines the profile aggregate for the Bill Presentment Service. This profile aggregate should be included in the <SvcProfInqRs> response for those servers that support the Bill Presentment Service.

Tag	Type	Usage	Description
<SvcCore>	Aggregate *see section 5.2.2.1*	Required	Service Core Aggregate. Information specified for every service.
<MsgSupt>	Open Enum	Required Repeating	Supported Messages. This is a list of messages that may be supported for Presentment. The convention is to use the name of the message without the Rq or Rs so that each message is only listed once.
			Defined values: BillerInq, BillerRev, BillInq, BillRev, BillStatusMod.
<OptSupt>	Open Enum	Optional Repeating	Options Supported.
			Defined values: AcctNickname, RecCtrl, SuppressEcho.
<PrcSched>	Aggregate *see section 5.2.2.3*	Optional	Processing Schedule Aggregate. If omitted, the default processing schedule is assumed.

9 The Valuable Media Service <MediaSvc>

The IFX Specification provides support for the tracking of valuable media at IFX client devices (e.g. ATMs, teller cash drawers) through the Valuable Media Service. The Valuable Media Service includes functions such as Media Account Adjustments (i.e. by a withdrawal, deposit or replenishment activity) and Media Account Inventory Inquiries for retrieving the balance of inventory in Media Accounts. These can be used for tracking a wide variety of valuable media such as cash, coins, postage stamps, coupons, checks, and envelopes located at IFX client devices.

The message set provided by the Valuable Media Service can be used in conjunction with the Debit and Credit messages from the Banking Service.

9.1 Description

The Valuable Media Service message set is used to support the tracking of valuable media at IFX client devices. To achieve this, the notion of a Media Account is used. A ***Media Account*** is an entity that allows a financial institution to manage the value of the media stored within a single IFX client device. The ***Media Account*** may refer to a physical container, a logical grouping of containers, or the complete set of valuable media in the IFX client device.

Whenever the valuable media in a ***Media Account*** is adjusted (i.e. by a withdrawal, deposit or replenishment activity), this adjustment should be reported to the server. Depending on the environment, the adjustments may be collected by the client and sent to the server in batches at certain times or when the batch has reached a certain size, or each adjustment may be sent as it occurs.

The Valuable Media Service allows clients to perform the following functions:

- Debit and credit media accounts with the use of Add Media Account Adjust messages.
- Inquiry of media account results (i.e. balances) with the use of Media Account Inventory Inquiry messages.

A Media Account may refer to a physical container, a logical grouping of containers, or the complete set of valuable media in the IFX client device. An example from the Self-Service (i.e. ATM) environment may be useful. Assume an ATM with identifier "123" has five cassettes and the Financial Institution has decided that $5 notes are always stored in cassettes one and two and $20 notes are always stored in cassettes three, four, and five.

If the Financial Institution is *not* interested in keeping track of which cassette is used for dispensing a note, the <MediaAcctAdjInfo> aggregate may be used by setting the <MediaAcctId> to 123 (the ATM identifier), and using one <MediaItem> aggregate for all $5 notes dispensed and another <MediaItem> aggregate for all $20 notes dispensed.

If, however, the Financial Institution *is* interested in keeping track of which cassette is used for dispensing a note, the <MediaAcctAdjInfo> aggregate may be used by setting the <MediaAcctId> to 123 (the ATM identifier), and using a separate <MediaItem> aggregate for each cassette. The Financial Institution establishes the values used for the identifiers. One <MediaItem> aggregate would be used for the $5 notes in cassette one (with <MediaContainerId> set to cassette1), a second <MediaItem> aggregate for the $5 notes in cassette two (with <MediaContainerId> set to cassette2), a third <MediaItem> aggregate for the $20 notes in cassette three (with <MediaContainerId> set to cassette3), a fourth <MediaItem> aggregate for the $20 notes in cassette four (with <MediaContainerId> set to cassette4), and the fifth <MediaItem> would be used for the $20 notes in cassette five (with <MediaContainerId> set to cassette5).

These are just two examples how a Media Account and the <MediaAcctAdjInfo> aggregate can be used.

9.2 Valuable Media Service Message Summary

Function / Message Name	Required	Comments
Media Account Adjust Add **\<MediaAcctAdjAddRq\>** **\<MediaAcctAdjAddRs\>**	Yes	Allows a client to request a debit or credit of valuable media from a specified media account.
Media Account Adjust Inquiry **\<MediaAcctAdjInqRq\>** **\<MediaAcctAdjInqRs\>**		Allows a client to view media account adjust records.
Media Account Adjust Audit **\<MediaAcctAdjAudRq\>** **\<MediaAcctAdjAudRs\>**		Allows a client to play back the media account adjust messages associated with the client device since some past point in time.
Media Account Adjust Synchronization **\<MediaAcctAdjSyncRq\>** **\<MediaAcctAdjSyncRs\>**		Allows a client to synchronize the media account adjust messages associated with the client device since some past point in time.
Media Account Adjust Reversal **\<MediaAcctAdjRevRq\>** **\<MediaAcctAdjRevRs\>**		Allows a client to reverse a previous media account adjust request message.
Media Account Inventory Inquiry **\<MediaAcctInventoryInqRq\>** **\<MediaAcctInventoryInqRs\>**	Yes	Allows a client to retrieve the inventory of valuable media in a media account at the time the message is executed.
Media Account InventoryReversal **\<MediaAcctInventoryRevRq\>** **\<MediaAcctInventoryRevRs\>**		Allows a client to reverse a previous media account inventory request message.

9.3 Valuable Media Service Common Aggregates

9.3.1 Media Item \<MediaItem\>

The \<MediaItem\> aggregate is used to describe the media in a media account transaction. This aggregate is used with the Media Account Adjust Add and Media Account Inventory Inquiry messages.

Tag	Type	Usage	Description
\<MediaContainerId\>	Identifier	Optional	Media Container identifier. This identifier can be used to identify the container of the media item.

Tag	Type	Usage	Description
<MediaType>	Open Enum	Required	Media Type. Defines the type of media being described by the <MediaItem> aggregate. Defined values: • Cash • UnverifiedCashDeposit • VerifiedCashDeposit • Coin • Stamp • Coupon • Check • UnverifiedCheckDeposit • VerifiedCheckDeposit • EnvelopeDeposit • MultiDeposit • Card
<MediaSubType>	Open Enum	Optional	Media Subtype. Used to further clarify a media type. When media type is Check, defined values are: Travelers, Managers, Bank, Cashiers
<MediaCashInfo>	Aggregate *see section 9.3.2.2*	Optional	Contains details about a cash media item being deposited.
<Count>	Long	Required OR	A count of the number of media items in this aggregate.
<CurAmt>	Currency Amount	Required OR	The currency value of an individual media item.
<TotalCurAmt>	Currency Amount	Required OR	The sum of the currency value of all media items of this type.

9.3.2   Media Account Record Aggregate <MediaAcctAdjRec>

The <MediaAcctAdjRec> aggregate is used in responses to both the Media Account Adjust Add Response <MediaAcctAdjAddRs> and Media Account Adjust Inquiry Response <MediaAcctAdjInqRs> messages.

Tag	Type	Usage	Description
<MediaAcctAdjId>	Identifier	Required	Media Account Adjust Identifier. Assigned by the server at the time the Media Account Adjust transaction is first added. Cannot be changed by the client.
<MediaAcctAdjInfo>	Aggregate *see section 9.3.2.1*	Required	Media Account Adjust Information Aggregate.

9.3.2.1   Media Account Adjust Information Aggregate <MediaAcctAdjInfo>

The <MediaAcctAdjInfo> aggregate is used in adjusting the value of a media account with the use of a Media Account Adjust Add message.

Tag	Type	Usage	Description
<MediaAcctId>	Identifier	Required	Media Account Identifier. This identifier can refer to a physical container, a logical grouping of containers, or the complete set of valuable media at a client device.

Tag	Type	Usage	Description
<MediaTrnType>	Open Enum	Optional	Media Transaction Type. The type of media account transaction associated with this request.
			Defined values are:
			Dispense – Valuable media was dispensed.
			Deposit – Valuable media was deposited.
			ReplenishSet – Replenishment, the amounts reported are set absolutely.
			ReplenishAdd – Replenishment, the amounts reported are added.
			ReplenishRemove – Replenishment, the amounts reported are subtracted.
			BalanceSet – Balancing, the amounts reported are set absolutely.
<CurAmt>	Currency Amount	Optional	Currency Amount. The total value of the media in this media account transaction.
<TrnAuthId>	Identifier	Optional	Transaction Authorization Identifier. The client may use this identifier in a Banking Service Debit/Credit message to associate a Media Account Adjust message with the original authorization identifier generated. For a Media Account Adjust message to debit, the <TrnAuthId> would be set to the <DebitId> value that was returned from a Banking Service Debit Authorization message. For a Media Account Adjust message to credit, the <TrnAuthId> would be set to the <CreditId> value that was returned from a Banking Service Credit Authorization message.
			Note: May be renamed to <TrnId> in IFX 2.0
<TrnRqUID>	Identifier	Optional	RqUID of the financial transaction request associated with this media adjustment.
<MediaItem>	Aggregate *see section 9.3.1*	Required Repeating	Media Item aggregate. This aggregate is used to describe the details of the media in a media account transaction.

9.3.2.2   Media Cash Information Aggregate <MediaCashInfo>

The <MediaCashInfo> aggregate contains details about a cash media item being deposited.

Tag	Type	Usage	Description
<CashValidity>	Open Enum	Optional	Cash Validity. The validity of the cash being deposited.
			Defined Values:
			Genuine, Counterfeit, Suspect
<CashQuality>	Open Enum	Optional	Cash Quality. The quality of the cash being deposited.
			Defined Values:
			Fit, Unfit
<CashSignature>	Binary	Optional	Cash Signature. The signature of the cash being deposited.
<CashSerialNum>	C-30	Optional	Cash Serial Number. The serial number of the cash being deposited.

9.4 Media Account Adjust

A client uses the Media Account Adjust messages when media has been dispensed from the client device during a customer withdrawal transaction or when media has been deposited to the client device during a customer deposit transaction. For a dispense action, the Media Account Adjust message can be associated with the Banking Service Debit Message by setting the TrnAuthId to the DebitAuthId value that was received in the withdrawal. For a deposit action, the Media Account Adjust message can be associated with the Banking Service Credit Message by setting the TrnAuthId to the CreditAuthId used for the deposit.

A client also uses the Media Account Adjust messages when media is added or removed from the client device during a replenishment function.

9.4.1 Media Account Adjust Add

A client uses the Media Account Adjust Add message to inform the server of a media account adjustment, due to either a dispense, deposit, or a replenishment action.

9.4.1.1 Request <MediaAcctAdjAddRq>

Tag	Type	Usage	Description
<RqUID>	UUID	Required	Request Identifier. Sent by the client as a universally unique identifier for the message. Used to correlate responses with requests.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header aggregate.
<AsyncRqUID>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously.

For more information, see section 3.2.11.1. |
| **<CustId>** | Aggregate *see section 3.2.3.1* | Optional *but see description* | Customer Identification aggregate. This is the identification of the client device for which the request is being issued. This element is required if the owner of the object(s) specified in this request is not the user specified in <SignonRq>. |
| **<MediaAcctAdjInfo>** | Aggregate *see section 9.3.2.1* | Required | Media Account Adjust Information Aggregate. This aggregate is used to uniquely identify the media account used in this media account transaction. |

9.4.1.2 Response <MediaAcctAdjAddRs>

Tag	Type	Usage	Description
<Status>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, <StatusCode> defaults to 0 (zero).
<RqUID>	UUID	Required Echoed	Request Identifier.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header aggregate.
<MsgRsHdr>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header aggregate.
<AsyncRqUID>	UUID	Optional Echoed	Asynchronous Request Identifier.

Tag	Type	Usage	Description
<CustId>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Identifier aggregate.
<MediaAcctAdjInfo>	Aggregate *see section 9.3.2.1*	Required Echoed	Media Account Adjust Information Aggregate.
<MediaAcctAdjRec>	Aggregate *see section 9.3.2*	Required	Media Account Adjust Record Aggregate.
<CSPRefId>	Identifier	Optional	Customer Service Provider Reference Identifier.
<SPRefId>	Identifier	Optional	Service Provider Reference Identifier.

9.4.2   Media Account Adjust Inquiry

A client can use the Media Account Adjust Inquiry message to get a list of transactions that have occurred on Media Accounts.

9.4.2.1   Request <MediaAcctAdjInqRq>

Tag	Type	Usage	Description
<RqUID>	UUID	Required	Request Identifier.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header aggregate.
<AsyncRqUID>	UUID	Optional	Asynchronous Request Identifier.
<CustId>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification aggregate. This is the identification of the client device for which the request is being issued. This element is required if the owner of the object(s) specified in this request is not the user specified in <SignonRq>.
			This field is used as a selection criterion.
<RecCtrlIn>	Aggregate *see section 3.2.12.2.1*	Optional	Records Control Input Aggregate.
<MediaAcctAdjId>	Identifier	Optional Repeating	Media Account Adjust Identifier. Assigned by the server at the time the Media Account Adjust transaction is first added. Cannot be changed by the client.
			This field is used as a selection criterion.
<CSPRefId>	Identifier	Optional Repeating	Customer Service Provider Reference Identifier.
			This field is used as a selection criterion.
<SPRefId>	Identifier	Optional Repeating	Service Provider Reference Identifier.
			This field is used as a selection criterion.
<MediaAcctId>	Identifier	Optional	Media Account Identifier. This is used to uniquely identify the media account used in a media account transaction.
			This field is used as a selection criterion.
<SelRangeDt>	Aggregate *see section 3.2.10.1*	Optional	Selection Range Date Aggregate.
			This field is used as a selection criterion.

Tag	Type	Usage	Description
<SelRangeCurAmt>	Aggregate *see section 3.2.10.2*	Optional	Selection Range Currency Amount Aggregate. The currency amount used in this selection criterion is the total value of the media in a media account transaction (the MediaTrnAmt). This field is used as a selection criterion.
<IncToken>	Boolean	Optional	Include Token. If *True*, a <Token> should be included in the response to set a base for future Sync messages. If *False* or omitted, no <Token> is returned.

9.4.2.2   Response <MediaAcctAdjInqRs>

Tag	Type	Usage	Description
<Status>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, <StatusCode> defaults to 0 (zero).
<RqUID>	UUID	Required Echoed	Request Identifier.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header aggregate.
<MsgRsHdr>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header aggregate.
<AsyncRqUID>	UUID	Optional Echoed	Asynchronous Request Identifier.
<CustId>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Identifier aggregate.
<RecCtrlOut>	Aggregate *see section 3.2.12.2.2*	Optional but see description	Records Control Output Aggregate. Required if <RecCtrlIn> was provided in the request and the server supports Records Control.
<MediaAcctAdjId>	Identifier	Optional Echoed	Media Account Adjust Identifier.
<CSPRefId>	Identifier	Optional Repeating Echoed	Customer Service Provider Reference Identifier.
<SPRefId>	Identifier	Optional Repeating Echoed	Service Provider Reference Identifier.
<MediaAcctId>	Aggregate	Optional Echoed	Media Account Identifier.
<SelRangeDt>	Aggregate *see section 3.2.10.1*	Optional Echoed	Selection Range Date Aggregate.
<SelRangeCurAmt>	Aggregate *see section 3.2.10.2*	Optional Echoed	Selection Range Currency Amount Aggregate.
<MediaAcctAdjRec>	Aggregate *see section 9.3.2*	Optional Repeating	Media Account Adjust Record Aggregate. These records are generated by the server and reflect the current state of the Media Account Id's Media Account Adjust transactions. The records are filtered by the selection criteria specified in the request message.

Tag	Type	Usage	Description
\<Token\>	Identifier	Optional but see description	Token. Server Assigned. Client should make no assumptions about the value of this token relative to others that it may have received. The token is significant only to the server that originally assigned it.
			The token is only included if \<IncToken\> = *True* in the request. \<Token\> = 0 is returned if no records are returned within the response.

9.4.3   Media Account Adjust Audit

A client can use the Media Account Adjust Audit message to audit Media Account Adjust Add messages. When the Media Account object changes, the server must generate an Rs message corresponding to the Rq that created the pending state.

9.4.3.1   Request \<MediaAcctAdjAudRq\>

Tag	Type	Usage	Description
\<RqUID\>	UUID	Required	Request Identifier.
\<MsgRqHdr\>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header aggregate.
\<AsyncRqUID\>	UUID	Optional	Asynchronous Request Identifier.
\<CustId\>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification aggregate. This is the identification of the client device for which the request is being issued. This element is required if the owner of the object(s) specified in this request is not the user specified in \<SignonRq\>.
			This field is used as a selection criterion.
\<RecCtrlIn\>	Aggregate *see section 3.2.12.2.1*	Optional	Records Control Input Aggregate.
\<SelRangeDt\>	Aggregate *see section 3.2.10.1*	Optional	Selection Range Date Aggregate.
			This field is used as a selection criterion.
\<Method\>	Closed Enum	Optional Repeating	Audit Selection Action. Used to identify actions associated with the object that is being audited.
			Valid values: Add.
			This field is used as a selection criterion.
\<MediaAcctAdjId\>	Identifier	Optional Repeating	Media Account Adjust Identifier. Assigned by the server at the time the Media Account Adjust transaction is first added. Cannot be changed by the client.
			This field is used as a selection criterion.

9.4.3.2   Response \<MediaAcctAdjAudRs\>

Tag	Type	Usage	Description
\<Status\>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, \<StatusCode\> defaults to 0 (zero).
\<RqUID\>	UUID	Required Echoed	Request Identifier.

Tag	Type	Usage	Description
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header aggregate.
<MsgRsHdr>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header aggregate.
<AsyncRqUID>	UUID	Optional Echoed	Asynchronous Request Identifier.
<CustId>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Identifier aggregate.
<RecCtrlOut>	Aggregate *see section 3.2.12.2.2*	Optional but see description	Records Control Output Aggregate. Required if <RecCtrlIn> was provided in the request and the server supports Records Control.
<SelRangeDt>	Aggregate *see section 3.2.10.1*	Optional Echoed	Selection Range Date Aggregate.
<Method>	Closed Enum	Optional Repeating Echoed	Audit Selection Criterion.
<MediaAcctAdjId>	Identifier	Optional Echoed	Media Account Adjust Identifier.
<MediaAcctAdjMsgRec>	Aggregate	Optional Repeating	Media Account Adjust Message Record Aggregate.
<CustId>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the entity who initiated the request (e.g., the one whose identifier appears in the <SignonRq> of the original issued request). For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the CSR or SP.
<MsgRecDt>	Date/Time	Optional	Message Record Creation Date. The date/time at which the message record was stored/created by the service provider.
<MediaAcctAdjAddRs>	Aggregate *see section 9.4.1.2*	Required	Media Account Adjust Add Response Message Aggregate.
<MediaAcctAdjRevRs>	Aggregate *see section 9.4.5.2*	Required	Media Account Adjust Reversal Response Message Aggregate.
</MediaAcctAdjMsgRec>			

9.4.4   Media Account Adjust Synchronization

A client can use the Media Account Adjust Synchronization message to synchronize on Media Account Adjust Add messages. When the Media Account object changes, the server must generate a response message to the request that created the pending state.

9.4.4.1   Request <MediaAcctAdjSyncRq>

Tag	Type	Usage	Description
<RqUID>	UUID	Required	Request Identifier.

Tag	Type	Usage	Description
\<MsgRqHdr\>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header aggregate.
\<AsyncRqUID\>	UUID	Optional	Asynchronous Request Identifier.
\<CustId\>	Aggregate *see section 3.2.3.1*	Required	Customer Identifier aggregate. This is the identification of the client device for which the request is being issued.
\<RecCtrlIn\>	Aggregate *see section 3.2.12.2.1*	Optional	Records Control Input Aggregate.
\<Token\>	Identifier	Required	Token. Server assigned. Client should make no assumptions about the value of this token relative to others that it may have received. The token is significant only to the server that originally assigned it. This is a token that has been previously sent by the server, or zero for first time requests.
\<MediaAcctId\>	Identifier	Required	Media Account Identifier.

9.4.4.2 Response \<MediaAcctAdjSyncRs\>

Tag	Type	Usage	Description
\<Status\>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, \<StatusCode\> defaults to 0 (zero).
\<RqUID\>	UUID	Required Echoed	Request Identifier.
\<MsgRqHdr\>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header aggregate.
\<MsgRsHdr\>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header aggregate.
\<AsyncRqUID\>	UUID	Optional Echoed	Asynchronous Request Identifier.
\<CustId\>	Aggregate *see section 3.2.3.1*	Required	Customer Identification aggregate. This is the identification of the client device for which the request is being issued.
\<RecCtrlOut\>	Aggregate *see section 3.2.12.2.2*	Optional but see description	Records Control Output Aggregate. Required if \<RecCtrlIn\> was provided in the request and the server supports Records Control.
\<Token\>	Identifier	Required	Token. Server assigned. Client should make no assumptions about the value of this token relative to others that it may have received. The token is significant only to the server that originally assigned it. This is a token that has been previously sent by the server, or zero for first time requests.
\<MediaAcctId\>	Identifier	Required	Media Account Information Aggregate.
\<NewToken\>	Identifier	Required	New Token. Server assigned. Client should make no assumptions about the value of this token relative to others that it may have received. The token is significant only to the server that originally assigned it.

Tag	Type	Usage	Description
\<MediaAcctAdjMsgRec>	Aggregate	Optional Repeating	Media Account Adjust Message Record Aggregate.
\<CustId>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the entity who initiated the request (e.g., the one whose identifier appears in the \<SignonRq> of the original issued request). For example, if a CSR or SP issues the request on behalf of the user, then \<CustId> is required, and must contain the value of the CSR or SP.
\<MsgRecDt>	Date/Time	Optional	Message Record Creation Date. The date/time at which the message record was stored/created by the service provider.
\<MediaAcctAdjAddRs>	Aggregate *see section 9.4.1.2*	Required	Media Account Adjust Add Response Message Aggregate.
\</MediaAcctAdjMsgRec>			

9.4.5   Media Account Adjust Reversal

9.4.5.1   Request \<MediaAcctAdjRevRq>

Tag	Type	Usage	Description
\<RqUID>	UUID	Required	Request Identifier.
\<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header aggregate.
\<AsyncRqUID>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously.
\<CustId>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in \<SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then \<CustId> is required, and must contain the value of the user whose request is being issued.
\<RevReasonCode>	Open Enum	Required	Reversal reason code. The reason why the client is requesting a reversal.
\<Desc>	C-80	Optional	Description, text describing the reason for the reversal.
\<MediaAcctAdjMsgRqInfo>	Aggregate	Required	Media Account Adjustment Message Request Information Aggregate. This aggregate contains the message to reverse. See RevOptSupt in the service profile for specific reversal messages supported.
\<MediaAcctAdjAddRq>	Aggregate *see section 9.4.1.1*	Required XOR	Media Account Adjustment Add Request Message Aggregate.
\<MediaAcctAdjInqRq>	Aggregate *see section 9.4.2.1*	Required XOR	Media Account Adjustment Inquiry Request Message Aggregate.
\<MediaAcctAdjAudRq>	Aggregate *see section 9.4.3.1*	Required XOR	Media Account Adjustment Audit Request Message Aggregate.

Tag	Type	Usage	Description
<MediaAcctAdjSyncRq>	Aggregate *see section 9.4.4.1*	Required XOR	Media Account Adjustment Synchronization Request Message Aggregate.
</MediaAcctAdjMsgRqInfo>			

9.4.5.2   Response <MediaAcctAdjRevRs>

Tag	Type	Usage	Description
<Status>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, <StatusCode> defaults to 0 (zero).
<RqUID>	UUID	Required Echoed	Request Identifier.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header aggregate.
<MsgRsHdr>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header aggregate.
<AsyncRqUID>	UUID	Optional Echoed	Asynchronous Request Identifier.
<CustId>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Identification Aggregate.
<RevReasonCode>	Open Enum	Required Echoed	Reversal reason code. The reason why the client is requesting a reversal.
<Desc>	C-80	Optional Echoed	Description, text describing the reason for the reversal.
<MediaAcctAdjMsgRqInfo>	Aggregate	Required Echoed	Media Account Adjustment Message Request Information Aggregate. This aggregate contains the message to reverse. See RevOptSupt in the service profile for specific reversal messages supported.
<MediaAcctAdjAddRq>	Aggregate *see section 9.4.1.1*	Required Echoed XOR	Media Account Adjustment Add Request Message Aggregate.
<MediaAcctAdjInqRq>	Aggregate *see section 9.4.2.1*	Required Echoed XOR	Media Account Adjustment Inquiry Request Message Aggregate.
<MediaAcctAdjAudRq>	Aggregate *see section 9.4.3.1*	Required Echoed XOR	Media Account Adjustment Audit Request Message Aggregate.
<MediaAcctAdjSyncRq>	Aggregate *see section 9.4.4.1*	Required Echoed XOR	Media Account Adjustment Synchronization Request Message Aggregate.
</MediaAcctAdjMsgRqInfo>			
<CSPRefId>	Identifier	Optional	Customer Service Provider Reference Identifier.
<SPRefId>	Identifier	Optional	Service Provider Reference Identifier.

9.5 Media Account Inventory

A client may use the following messages to retrieve the inventory balance of media accounts. These may be used for consolidation and replenishment purposes.

9.5.1 Media Account Inventory Inquiry

A client may use the Media Account Inventory Inquiry message to retrieve the inventory balance of a media account.

9.5.1.1 Request <MediaAcctInventoryInqRq>

Tag	Type	Usage	Description
<RqUID>	UUID	Required	Request Identifier. Sent by the client as a universally unique identifier for the message. Used to correlate responses with requests.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header aggregate.
<AsyncRqUID>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously. For more information, see section 3.2.11.1.
<CustId>	Aggregate *see section 3.2.3.1*	Optional *but see description*	Customer Identification aggregate. This is the identification of the client device for which the request is being issued. This element is required if the owner of the object(s) specified in this request is not the user specified in <SignonRq>.
<MediaAcctId>	Identifier	Required	Media Account Identifier. This is used to uniquely identify the media account used in this media account transaction.

9.5.1.2 Response <MediaAcctInventoryInqRs>

Tag	Type	Usage	Description
<Status>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, <StatusCode> defaults to 0 (zero).
<RqUID>	UUID	Required Echoed	Request Identifier. Sent by the client as a universally unique identifier for the message. Used to correlate responses with requests.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header aggregate.
<MsgRsHdr>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header aggregate.
<AsyncRqUID>	UUID	Optional Echoed	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously. For more information, see section 3.2.11.1.
<CustId>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Identification aggregate.

Tag	Type	Usage	Description
<MediaAcctId>	Identifier	Required Echoed	Media Account Identifier.
<CurAmt>	Currency Amount	Optional	Currency Amount. The total value of the media in this media account.
<MediaItem>	Aggregate *see section 9.3.1*	Required Repeating	Media Item aggregate. This aggregate is used to describe the details of the media in inventory.

9.5.2   Media Account Inventory Reversal

9.5.2.1   Request <MediaAcctInventoryRevRq>

Tag	Type	Usage	Description
<RqUID>	UUID	Required	Request Identifier.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header aggregate.
<AsyncRqUID>	UUID	Optional	Asynchronous Request Identifier. Sent by a client to retrieve a response that was asynchronously generated by a server, generally in the case where the response would have taken too long to build and be able to be sent synchronously.
			For more information, see section **3.2.12.1**.
<CustId>	Aggregate *see section 3.2.3.1*	Optional	Customer Identification Aggregate. This is the identifier of the user for whom the request is being issued. This element is required if the owner of the object(s) specified in the request is not the user specified in <SignonRq>. For example, if a CSR or SP issues the request on behalf of the user, then <CustId> is required, and must contain the value of the user whose request is being issued.
<RevReasonCode>	Open Enum	Required	Reversal reason code. The reason why the client is requesting a reversal.
<Desc>	C-80	Optional	Description, text describing the reason for the reversal.
<MediaAcctInventoryMsgRqInfo>	Aggregate	Required	Media Account Inventory Message Request Information Aggregate. This aggregate contains the message to reverse. See RevOptSupt in the service profile for specific reversal messages supported.
<MediaAcctInventoryInqRq>	Aggregate *see section 9.5.1.1*	Optional	Media Account Inventory Inquiry Request Message Aggregate.
</MediaAcctInventoryMsgRqInfo>			

9.5.2.2   Response <MediaAcctInventoryRevRs>

Tag	Type	Usage	Description
<Status>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, <StatusCode> defaults to 0 (zero).

Tag	Type	Usage	Description
<RqUID>	UUID	Required Echoed	Request Identifier.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header aggregate.
<MsgRsHdr>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header aggregate.
<AsyncRqUID>	UUID	Optional Echoed	Asynchronous Request Identifier.
<CustId>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Identification Aggregate.
<RevReasonCode>	Open Enum	Required Echoed	Reversal reason code. The reason why the client is requesting a reversal.
<Desc>	C-80	Optional Echoed	Description, text describing the reason for the reversal.
<MediaAcctInventoryMsgRqInfo>	Aggregate	Required Echoed	Media Account Inventory Message Request Information Aggregate. This aggregate contains the message to reverse. See RevOptSupt in the service profile for specific reversal messages supported.
<MediaAcctInventoryInqRq>	Aggregate *see section 9.5.1.1*	Optional	Media Account Invenory Inquiry Request Message Aggregate.
</MediaAcctInventoryMsgRqInfo>			
<CSPRefId>	Identifier	Optional	Customer Service Provider Reference Identifier.
<SPRefId>	Identifier	Optional	Service Provider Reference Identifier.

9.6 Valuable Media Service Profile <MediaSvcProfInfo>

The Profile for the Valuable Media Service <MediaSvcProfInfo> is defined below. This profile may be returned to the client in <SvcProfInqRq>, providing information on how the client should use the Valuable Media Service.

Tag	Type	Usage	Description
<SvcCore>	Aggregate *see section 5.2.2.1*	Required	Service Core Aggregate. Information specified for every service.
<MsgSupt>	Open Enum	Required Repeating	Messages Supported. Defined values: MediaAcctInventoryInq, MediaAcctAdjAdd, MediaAcctAdjInq, MediaAcctAdjAud, MediaAcctAdjSync, MediaAcctInventoryRev, MediaAcctAdjRev
<OptSupt>	Open Enum	Optional Repeating	Options Supported. Defined values: RecCtrl, SuppressEcho

10 Root Service

The IFX Specification provides device management and terminal management capabilities through the Root Service. The Root Service includes device management functions to allow an IFX client (e.g. ATM) to provide

information about its devices and the status of those devices. The terminal management functions within the Root Service allows information (e.g. Reg E information, terminal identifier) pertaining to an IFX client (e.g. ATM) to be managed.

10.1 Description

The Device object and associated messages provide the following functionality:

- Allow an IFX client terminal (e.g. ATM or POS) to send device information and device status information to another IFX entity (e.g. Host, SP, CSP). The <DevAdvise> message provides this functionality.

- Allow an IFX client (e.g. Host, SP, CSP) to retrieve device information and device status information from an IFX terminal (e.g. ATM or POS). The <DevInq> message provides this functionality.

The Terminal Object is used to logically represent an IFX client terminal (e.g. ATM or POS). The Terminal Object messages provide for the following functionality:

- Allow the postal address (e.g. street address) of the terminal to be set.

- Allow the postal address (e.g. street address) of the terminal to be retrieved.

- Allow the common location name of the terminal to be set.

- Allow the common location name of the terminal to be retrieved.

- Allow the owner name of the terminal to be set.

- Allow the owner name of the terminal to be retrieved.

- Allow the time zone (and implicitly the time) of the terminal to be set.

- Allow the time zone information of the terminal to be retrieved.

- To request the state/status of the terminal to be set to "Open" or "Closed"

The Terminal Service Provider Object is used to logically represent the relationship an IFX client (e.g. ATM or POS terminal) has with a Service Provider. The Terminal Service Provider Object messages provide for the following functionality:

- Allow for one IFX client Terminal to be known as more than one TerminalId. This requirement is needed in some multiple Service Provider environments. The TerminalId used in this object is an identifier that uniquely identifies the Terminal within the context of the acquiring service provider.

- Allow a Service Provider to inform an IFX client Terminal that services are available/unavailable.

10.2   Root Service Message Summary

Function / Message Name	Required	Comments
Terminal Object Add **<TerminalObjAddRq>** **<TerminalObjAddRs>**		Allows a client to add a Terminal object. The client that uses this message is a host (e.g. CSP, SP).
Terminal Object Modify **<TerminalObjModRq>** **<TerminalObjModRs>**		Allows a client to modify the information in a Terminal object record. The client that uses this message is a host (e.g. CSP, SP).
Terminal Object Inquiry **<TerminalObjInqRq>** **<TerminalObjInqRs>**		Allows a client to retrieve Terminal object records. The client that uses this message is a host (e.g. CSP, SP).
Terminal Object Status Modify **<TerminalObjStatusModRq>** **<TerminalObjStatusModRs>**		Allows a client to request a change in the status of a Terminal object. The client that uses this message is a host (e.g. CSP, SP).
Terminal Object Audit **<TerminalObjAudRq>** **<TerminalObjAudRs>**		Allows a client to play back Terminal object messages associated with a specific Terminal object. The client that uses this message is a host (e.g. CSP, SP).
Terminal Object Advise **<TerminalObjAdviseRq>** **<TerminalObjAdviseRs>**		Allows a client to inform others of a change in a Terminal object record. The client that uses this message is a Terminal (e.g. ATM, POS).
Terminal Object Delete **<TerminalObjDelRq>** **<TerminalObjDelRs>**		Allows a client to delete a Terminal object. The client that uses this message is a host (e.g. CSP, SP).
Terminal Service Provider Object Add **<TerminalSPObjAddRq>** **<TerminalSPObjAddRs>**		Allows a client to add a Terminal Service Provider object. The client that uses this message is a Terminal (e.g. ATM, POS).
Terminal Service Provider Object Modify **<TerminalSPObjModRq>** **<TerminalSPObjModRs>**		Allows a client to modify the information in a Terminal Service Provider object record. The client that uses this message is a Terminal (e.g. ATM, POS).
Terminal Service Provider Object Inquiry **<TerminalSPObjInqRq>** **<TerminalSPObjInqRs>**		Allows a client to retrieve Terminal Service Provider object records. The client that uses this message is a Terminal (e.g. ATM, POS).
Terminal Service Provider Object Audit **<TerminalSPObjAudRq>** **<TerminalSPObjAudRs>**		Allows a client to play back Terminal Service Provider object messages associated with a specific Terminal Service Provider object. The client that uses this message is a Terminal (e.g. ATM, POS).
Terminal Service Provider Object Advise **<TerminalSPObjAdviseRq>** **<TerminalSPObjAdviseRs>**		Allows a client to inform others of a change in a Terminal Service Provider object record. The client that uses this message is a host (e.g. CSP, SP).

Function / Message Name	Required	Comments
Terminal Service Provider Object Delete		Allows a client to delete a Terminal Service Provider object. The client that uses this message is a Terminal (e.g. ATM, POS).
\<TerminalSPObjDelRq> **\<TerminalSPObjDelRs>**		
Device Inquiry		Allows a client to retrieve device information and device status information for devices located at a Terminal. The client that uses this message is a host (e.g. CSP, SP).
\<DevInqRq> **\<DevInqRs>**		
Device Advise		Allows a client to inform others that a change in a device status has occurred. The client that uses this message is a Terminal (e.g. ATM, POS).
\<DevAdviseRq> **\<DevAdviseRs>**		

10.3   Root Service Common Aggregates

10.3.1   Terminal Object Record Aggregate \<TerminalObjRec>

Tag	Type	Usage	Description
\<TerminalObjId>	Identifier	Required	Terminal Object Identifier. The identifier of the terminal object.
\<TerminalObjInfo>	Aggregate *see section 10.3.1.1*	Required	Terminal Object Information Aggregate.
\<TerminalObjStatus>	Aggregate *see section 10.3.1.2*	Required	Terminal Object Status Aggregate.

10.3.1.1   Terminal Object Information Aggregate \<TerminalObjInfo>

Tag	Type	Usage	Description
\<PostAddr>	Aggregate *see section 3.2.3.7.1*	Optional	Postal Address Aggregate. The physical address of the ATM/POS terminal.
\<Desc>	C-80	Optional	Description. The generally accepted name for the location of the ATM/POS terminal.
\<Name>	C-40	Optional	Name. Name of the owner or operator of the ATM/POS terminal.
\<UTCOffset>	Long	Optional	Offset from UTC in minutes for the ATM/POS terminal.

10.3.1.2   Terminal Object Status Aggregate \<TerminalObjStatus>

Tag	Type	Usage	Description
\<TerminalObjStatusCode>	Closed Enum	Required	Terminal Object Status Code. Current operation mode of the terminal. Valid Values: Open – The terminal accepts business transactions and administrative maintenance. Closed – The terminal does not accept business transactions, but does accept administrative maintenance.

Tag	Type	Usage	Description
<RequestedOperationMode>	Closed Enum	Optional	Requested Operation Mode. The requested operation mode of the terminal. Valid Values: Open – The terminal has been requested to accept business transactions and administrative maintenance. Closed – The terminal has been requested to not accept business transactions, but will continue to accept administrative maintenance.
<ExpectedOpenDt>	DateTime	Optional	Expected Open DateTime. The terminal date/time that the terminal operation mode is expected to return to "Open" if the operation mode is being changed to something other than "Open". There is no obligation inferred. A Terminal application could use this information to inform customers when the Terminal may again be available.
<StatusDesc>	C-255	Optional	Status Description. Explanatory text associated with this terminal object status.
<EffDt>	DateTime	Optional	Effective Date. The date/time the <TerminalObjStatusCode> took effect.
<StatusModBy>	Open Enum	Optional	Status Modified By. If present, indicates who modified the <TerminalObjStatusCode>. Valid Values: FI, CSP, CSPSR, Terminal.

10.3.2   Terminal Service Provider Object Record Aggregate <TerminalSPRec>

Tag	Type	Usage	Description
<TerminalSPObjId>	Identifier	Required	Terminal Service Provider Object Identifier. The identifier of the terminal service provider object.
<TerminalSPObjInfo>	Aggregate *see section 10.3.2.1*	Required	Terminal Service Provider Object Information Aggregate.
<TerminalSPObjStatus>	Aggregate *see section 10.3.2.2*	Required	Terminal Service Provider Object Status Aggregate.

10.3.2.1   Terminal Service Provider Object Information Aggregate <TerminalSPObjInfo>

Tag	Type	Usage	Description
<SPName>	Identifier	Required	Service Provider Name. The name of the Service Provider associated with this Terminal Information aggregate.
<TerminalId>	Identifier	Required	Terminal Identifier. Identification of terminal, such as ATM/POS terminal code or terminal number. This terminal identifier is used to uniquely identify the terminal within the context of the acquiring service provider.

10.3.2.2 Terminal Service Provider Object Status Aggregate <TerminalSPObjStatus>

Tag	Type	Usage	Description
<TerminalSPObjStatusCode>	Closed Enum	Required	Terminal Service Provider Object Status Code. Current state of the availability of services from this Service Provider to this terminal. Valid Values: ServiceAvailable – One or more services are available from the Service Provider. A service is a collection of one or more messages. ServiceUnavailable – No services are available from the Service Provider. A service is a collection of one or more messages.
<ExpectedAvailDt>	DateTime	Optional	Expected Availability DateTime. The date/time that service is expected to be available from this Service Provider. A service is a collection of one or more messages. There is no obligation inferred.
<StatusDesc>	C-255	Optional	Status Description. Explanatory text associated with this terminal service provider object status.
<EffDt>	DateTime	Optional	Effective Date. The date/time the <TerminalSPObjStatusCode> took effect.
<StatusModBy>	Open Enum	Optional	Status Modified By. If present, indicates who modified the <TerminalSPObjStatusCode>. Valid Values: FI, CSP, CSPSR, Terminal.

10.3.3 Device Record Aggregate <DevRec>

The <DevRec> aggregate is used to provide information about a device and the status of device.

Tag	Type	Usage	Description
<DevId>	Identifier	Required	Device Identifier.
<DevlInfo>	Aggregate	Required	Device Information Aggregate.
<DevStatus>	Aggregate *see section 10.3.3.2*	Required	Device Status Aggregate.

10.3.3.1 Device Information Aggregate <DevInfo>

Tag	Type	Usage	Description
<DevName>	C-255	Optional	Device Name. Human readable name of the device.

10.3.3.2 Device Status Aggregate <DevStatus>

Tag	Type	Usage	Description
<DevStatusCode>	Closed Enum	Required	Device Status Code. This is the State-of-Health (SOH) of the device. Valid Values: Healthy, Warning, Fatal, Suspend.

<DevDetail>	Aggregate *see section 10.3.3.3*	Required	Device Detail Aggregate.
<DevPlatformData>	Aggregate *see section 10.3.3.4*	Optional Repeating	Device Platform Data Aggregate. Native platform specific information pertaining to the device.
<StatusDesc>	C-255	Optional	Status Description. Explanatory text associated with this device status.
<EffDt>	DateTime	Optional	Effective Date. The date/time the <DevStatusCode> took effect.
<StatusModBy>	Open Enum	Optional	Status Modified By. If present, indicates who modified the <DevStatusCode>.

Valid Values:

Terminal, Maintenance |

10.3.3.3 Device Detail Aggregate <DevDetail>

Note that the categorization of the following Device objects have been modeled after the XFS service class definitions. Although this will use the same classification as the XFS standard, this in no way ties an IFX client to use the XFS standard when using devices.

Tag	Type	Usage	Description
<DevDetailIdc>	Aggregate *see section 10.3.3.6*	Required XOR	Device details specific to identification card devices (covered by the XFS IDC service class).
<DevDetailPtr>	Aggregate *see section 10.3.3.15*	Required XOR	Device details specific to printer devices (covered in the XFS PTR service class).
<DevDetailCdm>	Aggregate *see section 10.3.3.29*	Required XOR	Device details specific to cash dispenser modules (covered in the XFS IDC service class).
<DevDetailPIN>	Aggregate *see section 10.3.3.9*	Required XOR	Device details specific to the Personal Identification Number Keypads and other keyboard entry devices (covered in the XFS PIN service class).
<DevDetailChk>	Aggregate *see section 10.3.3.10*	Required XOR	Device details specific to Check Readers and Scanners (covered in the XFS CHK service class).
<DevDetailDep>	Aggregate *see section 10.3.3.11*	Required XOR	Device details specific to Depositories (covered in the XFS DEP service class).
<DevDetailTtu>	Aggregate *see section 10.3.3.8*	Required XOR	Device details specific to Text Terminal Units (covered in the XFS TTU service class).
<DevDetailSiu>	Aggregate *see section 10.3.3.18*	Required XOR	Device details specific to Sensors and Indicators Units (covered in the XFS SIU service class).
<DevDetailVdm>	Aggregate *see section 10.3.3.17*	Required XOR	Device details specific to Vendor Dependent Mode (covered in the XFS VDM service class).
<DevDetailCam>	Aggregate *see section 10.3.3.32*	Required XOR	Device details specific to cameras (covered in the XFS CAM service class).

<DevDetailAlm>	Aggregate *see section 10.3.3.7*	Required XOR	Device details specific to alarms (covered in the XFS ALM service class).
<DevDetailCim>	Aggregate *see section 10.3.3.26*	Required XOR	Device details specific to Cash In Modules (covered in the XFS CIM service class).
<DevDetailApp>	Aggregate *see section 10.3.3.13*	Required XOR	Device details specific to the Application.

10.3.3.4 Device Platform Data Aggregate <DevPlatformData>

The Device Platform Data Aggregate returns information specific to certain device platforms. This is necessary for logging and troubleshooting purposes.

The DevPlatformXfs Sub-Aggregates may be returned even if the client is not on that platform. For instance, a non-XFS client may return the DevPlatformXfs Aggregate.

Tag	Type	Usage	Description
<DevPlatformSource>	Closed Enum	Optional	Device Platform Source. Indicates whether the Device message is coming directly from the device platform layer, or the client has interpreted this data from another format.
			Valid Values: Native, Interpreted
<DevPlatformXfs>	Aggregate *see section 10.3.3.5*	Required	Device Platform XFS Aggregate. Device details specific to the XFS Device message standards.

Note: In future versions of IFX additional non-XFS device platform aggregates may be added to the <DevPlatformData> aggregate. When a new aggregate is added it is anticipated to be a required XOR aggregate with the <DevPlatformXfs> aggregate.

10.3.3.5 Device Platform XFS Aggregate <DevPlatformXfs>

This aggregate is used to return Platform Specific information, as defined by the XFS Device Platform standard. Note that non-XFS platforms may return this information as well.

Tag	Type	Usage	Description
<PlatformXfsRequestId>	Decimal	Optional	Platform XFS Request Id. The sequence number (in decimal form) assigned by the XFS Manager for the request that raised the alert. This value is expected as an integer value in the range 0- 4294967295.
<PlatformXfsTimestamp>	DateTime	Optional	Platform XFS Timestamp. The time that the alert was raised, based on the XFS Platform date/time. The client should include as much detail as is available (down to the fraction of a second).
<PlatformXfsEventId>	Decimal	Optional	Platform XFS Event Id. The Id of the event that raised the alert (for unsolicited events). This value is expected as an integer value in the range 0- 4294967295.
<PlatformXfsLogicalName>	C-255	Optional	Platform XFS Logical Name. The high-level, pre-defined logical name of a service used by the XFS Manager as a key to obtain the specific configuration information needed. (e.g. "SYSJOURNAL1", "PASSBOOKPTR3", "CASHDISP02") Here, it is used to identify the service that raised the alert.
<PlatformXfsPhysicalName>	C-255	Optional	Platform XFS Physical Name. The physical service name of the service that raised the alert.

Tag	Type	Usage	Description
\<PlatformXfsWkstName\>	C-255	Optional	Platform XFS Workstation Name. The name of the workstation in which the logical service name is defined (if any)
\<PlatformXfsState\>	Decimal	Optional	Platform XFS State. The new state of the physical device managed by the service.
			Example states from the XFS Specifications include:
			WFS_STAT_DEVONLINE = 0, WFS_STAT_DEVFFLINE = 1, WFS_STAT_POWEROFF = 2, WFS_STAT_DEVNODEVICE = 3, WFS_STAT_DEVHWERROR = 4, WFS_STAT_DEVUSERERROR = 5, WFS_STAT_DEVBUSY = 6
			This value is expected as an integer value, not outside the range 0- 4294967295, that is defined as a state by the XFS standard.
\<PlatformXfsAppId\>	C-255	Optional	Platform XFS Application Id. The application ID associated with the session that raised the alert (if any)
\<PlatformXfsAction\>	Decimal	Optional	Platform XFS Action. The actions required to manage the error as defined by the XFS specification.
			Example actions from the XFS specifications include: WFS_ERR_ACT_NOACTION = 0, WFS_ERR_ACT_RESET = 1, WFS_ERR_ACT_SWERROR = 2, WFS_ERR_ACT_CONFIG = 4, WFS_ERR_ACT_HWCLEAR = 8, WFS_ERR_ACT_HWMAINT = 16, WFS_ERR_ACT_SUSPEND = 32
			This value is expected as an integer value, not outside the range 0- 4294967295, that is defined as an action by the XFS standard.
\<PlatformXfsDesc\>	C-∞	Optional	Platform XFS Description. A vendor-specific description of the error.
\<PlatformXfsCmdCode\>	Decimal	Optional	Platform XFS Command Code. The WFSExecute "command" code or the WFSGetInfo "category" code; not used for other command completions. This value is expected as an integer value in the range 0- 4294967295.
\<PlatformXfsCmdHResult\>	Long	Optional	Platform XFS Command HResult. The Device Command Result code as specified by the XFS device standards.

10.3.3.6 Device Detail Identification Card Aggregate \<DevDetailIdc\>

Device details specific to identification card devices (devices covered by the XFS IDC service class).

Tag	Type	Usage	Description
\<DevIdcType\>	Open Enum	Required	Type of Card Reader/IDC.
			Valid Values: Motor, Swipe, Contactless, Dip
\<DevIdcOperation\>	Open Enum	Optional	The operation being performed at the time the alert is raised. It is <u>strongly</u> recommended that the following defined values are used rather than custom values in order to ensure maximum portability.
			Valid Values: Read, Write, EjectCard, RetainCard, ChipIo, Reset, ChipPower

Tag	Type	Usage	Description
\<DevIdcMedia\>	Closed Enum	Required	The state of the media in the IDC device.
			Valid Values: Present, NotPresent, Jammed, NotSupported, Unknown, Entering
\<DevIdcRetainBin\>	Closed Enum	Required	The state of the retain bin.
			Valid Values: OK, High, Full, NotSupported
\<DevIdcCards\>	Long	Optional	The number of cards retained since the count was last reset on the client.
\<DevIdcRetainedCard\>	Aggregate *see section 10.3.3.12*	Optional	Details of the cards that have been retained as a part of this alert.
\<DevIdcSecurity\>	Closed Enum	Optional	The state of the security unit for this device.
			Valid Values: NotSupported, NotReady, Open
\<DevIdcChipPower\>	Closed Enum	Optional	The state of the chip on the currently inserted card in the device:
			Valid Values: Online, PoweredOff, Busy, NoDevice, HwError, NoCard, NotSupported, Unknown

10.3.3.7   Device Detail Alarm Aggregate \<DevDetailAlm\>

Device details specific to Alarm devices (devices covered by the XFS ALM service class).

Tag	Type	Usage	Description
\<DevAlmType\>	Open Enum	Required	Type of Alarm/Alm.
			Valid Values: Alarm
\<DevAlmOperation\>	Open Enum	Optional	The operation being performed at the time the alert is raised. It is <u>strongly</u> recommended that the following defined values are used rather than custom values in order to ensure maximum portability.
			Valid Values: SetAlarm, Reset, NotSupported
\<DevAlmAlarm\>	Closed Enum	Required	The state of the Alarm at time of alert.
			Valid Values: Reset, Set, NotSupported

10.3.3.8   Device Detail Text Terminal Unit Aggregate \<DevDetailTtu\>

Device details specific to Text Terminal Unit devices (devices covered by the XFS TTU service class).

Tag	Type	Usage	Description
\<DevTtuType\>	Open Enum	Required	Type of Text Terminal Unit/Ttu.
			Valid Values: Fixed, Removable
\<DevTtuOperation\>	Open Enum	Optional	The operation being performed at the time the alert is raised. It is <u>strongly</u> recommended that the following defined values are used rather than custom values in order to ensure maximum portability.
			Valid Values: Write, Read, Reset, NotSupported
\<DevTtuKbd\>	Closed Enum	Required	The state of the keyboard at time of alert.
			Valid Values: On, Off, NotAvailable

Tag	Type	Usage	Description
<DevTtuKbdLock>	Closed Enum	Required	The state of the keyboard lock at time of alert. Valid Values: On, Off, NotAvailable
<DevTtuLed>	Closed Enum	Optional Repeating *but see Description*	The state of the leds at time of alert. If an LED is present this element should be returned. The order of the LED's listed is significant. Valid Values: Off, BlinkingSlowly, MediumFlash, QuickFlash, Continuous, NotAvailable

10.3.3.9   Device Detail PIN Keypad Aggregate <DevDetailPIN>

Device details specific to PIN Keypad and other encryption/decryption devices (devices covered by the XFS PIN service class).

Tag	Type	Usage	Description
<DevPINType>	Open Enum	Required	Type of PIN Keypad device. Valid Values: Keyboard, EncryptDecrypt, HwSecurityModule
<DevPINOperation>	Open Enum	Optional	The operation being performed at the time the alert is raised. It is <u>strongly</u> recommended that the following defined values be used rather than custom values in order to ensure maximum portability. Valid Values: Encrypt, Decrypt, RandomNum, GetData, GetPIN, ImportKey, DeriveKey, PresentIdc, GetPINBlock, PINInitialization, Reset
<DevPINEncStat>	Closed Enum	Optional	The state of the Encryption Module. Valid Values: Ready, NotReady, NotInitialized, Busy, Undefined, Initialized

10.3.3.10 Device Detail Check Reader/Scanner Aggregate <DevDetailChk>

Device details specific to check reader and scanner devices (devices covered by the XFS CHK service class).

Tag	Type	Usage	Description
<DevChkType>	Open Enum	Required	Type of check reader/scanner device. Valid Values: Check
<DevChkOperation>	Open Enum	Optional	The operation being performed at the time the alert is raised. It is <u>strongly</u> recommended that the following defined values be used rather than custom values in order to ensure maximum portability. Valid Values: Eject, Retain, Reset, Write, Read
<DevChkImageEnabled>	Boolean	Optional	Indicates a special case of check reader, i.e., image-enabled (or check scanner). If present, and the value is *True*, then the device is a check scanner. If omitted, or the value is *False,* then the device is a check reader.
<DevChkMedia>	Closed Enum	Required	The state of the media in the check reader/scanner device. Valid Values: NotSupported, NotPresent, Required, Present, Jammed

Tag	Type	Usage	Description
<DevChkInk>	Closed Enum	Optional	The state of the ink in the check reader/scanner device.
			Valid Values: NotSupported, Full, Low, Out

10.3.3.11 Device Detail Depositories Aggregate <DevDetailDep>

Device details specific to Depositories (devices covered by the XFS DEP service class).

Tag	Type	Usage	Description
<DevDepType>	Open Enum	Required	Device Depository Type. Type of Depository.
			Valid Values:
			EnvelopeMotorized, EnvelopeManual, EnvelopeNone, BagDrop
<DevDepOperation>	Open Enum	Optional	Device Depository Operation. The operation being performed at the time the alert is raised. It is <u>strongly</u> recommended that the following defined values are used rather than custom values in order to ensure maximum portability.
			Valid Values:
			Entry, Dispense, Retract, Reset, ResetCount
<DevDepTransport>	Closed Enum	Required	Device Depository Transport. The state of the deposit transport at time of alert.
			Valid Values:
			OK, Inoperable, Unknown, NotSupported
<DevDepContainer>	Closed Enum	Required	Device Depository Container. The state of the deposit container at time of the alert.
			Valid Values:
			OK, High, Full, Inoperable, Missing, Unknown, NotSupported
<DevDepEnvelopeSupply>	Closed Enum	Required	Device Depository Envelope Supply. The state of the deposit envelope supply at time of the alert.
			Valid Values:
			OK, Low, Empty, Inoperable, Unlocked, Missing, Unknown, NotSupported
<DevDepEnvelopeDispenser>	Closed Enum	Required	Device Depository Envelope Dispenser. The state of the deposit envelope dispenser at time of the alert.
			Valid Values:
			OK, Inoperable, Unknown, NotSupported
<DevDepPrinter>	Closed Enum	Required	Device Depository Printer. The state of the deposit printer at time of the alert.
			Valid Values:
			OK, Inoperable, Unknown, NotSupported
<DevDepToner>	Closed Enum	Required	Device Depository Toner. The state of the deposit printers toner at time of the alert.
			Valid Values:
			Full, Low, Out, Unknown, NotSupported

\<DevDepShutter\>	Closed Enum	Required	Device Depository Shutter. The state of the shutter at time of alert.
			Valid Values:
			Jam, Unknown, NotSupported, Open, Close
\<DevDepNumOfDeposits\>	Long	Optional	Total number of bags or envelopes in the deposit container

10.3.3.12 Device Identification Card Retained Card Aggregate \<DevIdcRetainedCard\>

Tag	Type	Usage	Description
\<CardMagData\>	Aggregate *see section 3.2.7.1.3.1*	Required	Card Magnetic Stripe Data. The data from the card that was retained.

10.3.3.13 Device Detail Application Aggregate \<DevDetailApp\>

Device details specific to the Application.

Tag	Type	Usage	Description
\<DevAppOperation\>	Open Enum	Optional Repeating	Device Application Operation. The operation being performed at the time the alert is raised. It is strongly recommended that the following defined values be used rather than custom values in order to ensure maximum portability.
			Valid Values:
			BusinessTransaction, AdminMaintenance, AppMaintenance, SoftwareInstallation.
			NOTE: If the application is "idle" then no \<DevAppOperation\> element would be included.
\<DevAppNormalOperation\>	Boolean	Required	Device Application Normal Operation. An indication that the application is available for consumer transactions.
			Valid Values:
			True, False.
\<DevAppComponent\>	Aggregate *see section 10.3.3.14*	Optional Repeating	Device Application Component Aggregate. The state of individual components in the application.

10.3.3.14 Device Detail Application Component Aggregate \<DevAppComponent\>

Device details specific an Application component.

Tag	Type	Usage	Description
\<Name\>	C-40	Required	Name. Application/Component Name
\<Version\>	NC-12	Optional	Version. Application/Component Version
\<DevHealthLevel\>	Open Enum	Optional	Device Health Level. The state of the application/component at time of alert.
			Valid Values: Healthy, Warning, Fatal, Suspend

10.3.3.15 Device Detail Printer Aggregate <DevDetailPtr>

Device details specific to printer devices (devices covered by the XFS PTR service class).

Tag	Type	Usage	Description
<DevPtrType>	Open Enum	Required	Device Printer Type. Type of the printer device.
			Valid Values: Receipt, Journal, Document, Passbook, Scanner
<DevPtrOperation>	Open Enum	Optional	Device Printer Operation. The operation being performed at the time the alert is raised. It is <u>strongly</u> recommended that the following defined values be used rather than custom values in order to ensure maximum portability.
			Valid Values: ControlMedia, Print, Reset, RetractMedia, DispensePaper, Read
<DevPtrMedia>	Closed Enum	Required	Device Printer Media. The state of the print media at the time of the alert.
			Valid Values: Present, NotPresent, Jammed, NotSupported, Unknown, Entering, Retracted
<DevPtrRetractBin>	Closed Enum	Optional	Device Printer Retract Bin. The state of the printer retract bin.
			Valid Values: OK, High, Full, Unknown
<DevPtrSupply>	Aggregate *see section 10.3.3.16*	Optional Repeating	Device Printer Supply Aggregate. The state of the print supply at time of alert. The <DevPtrSupply> aggregate repeats for different possible locations of the supply.
<DevPtrInk>	Closed Enum	Optional	Device Printer Ink. The state of the ink at the time of the alert.
			Valid Values: Full, Low, Out, NotSupported, Unknown
<DevPtrToner>	Closed Enum	Optional	Device Printer Toner. The state of the printer toner or ink supply.
			Valid Values: Full, Low, Out, Unknown, NotSupported
<DevPtrLamp>	Closed Enum	Optional	Device Printer Lamp. The state of the lamp at time of alert.
			Valid Values: OK, Fading, Inoperative, NotSupported, Unknown

10.3.3.16 Device Printer Supply Aggregate <DevPtrSupply>

Printer supply details specific to printer devices.

Tag	Type	Usage	Description
<DevPtrSupplyLocation>	Open Enum	Optional	Device Printer Supply Location. The location of the printer supply.
			Valid Values: Upper, Lower, External, Auxiliary, Auxiliary2, Park
<DevPtrSupplyLevel>	Closed Enum	Required	Device Printer Supply Level. The actual supply level for the printer.
			Valid Values: Full, Low, Out, NotSupported, Unknown, Jammed

10.3.3.17 Device Detail Vendor Dependent Mode Aggregate <DevDetailVdm>

Device details specific to Vendor Dependent Mode (covered by the XFS VDM service class).

Tag	Type	Usage	Description
<DevVdmOperation>	Open Enum	Optional	Device VDM Operation. The operation being performed at the time the alert is raised. It is <u>strongly</u> recommended that the following defined values be used rather than custom values in order to ensure maximum portability.
			Valid Values:
			Enter, Exit
<DevVdmServiceState>	Closed Enum	Optional	Device VDM Service State. The state of the VDM Service at time of alert.
			Valid Values:
			EnterPending, ExitPending, Active, Inactive

10.3.3.18 Device Detail Sensors and Indicators Aggregate <DevDetailSiu>

Device details specific to sensors and indicator devices (devices covered by the XFS and J/XFS SIU service class).

Tag	Type	Usage	Description
<DevSiuType>	Aggregate *see section 10.3.3.19*	Required	Device SIU Type Aggregate. Types of sensors and indictors available.
<DevSiuOperation>	Open Enum	Optional	Device SIU Operation. The operation being performed at the time the alert is raised. . It is <u>strongly</u> recommended that the following defined values be used rather than custom values in order to ensure maximum portability.
			Valid Values: SetPorts, SetDoor, SetIndicator, SetGuidelight, SetAuxiliary, Reset, NotSupported
<DevSiuSensorStatus>	Aggregate *see section 10.3.3.20*	Optional	Device SIU Sensor Status Aggregate. The current state of any general sensors that are available.
<DevSiuDoorStatus>	Aggregate *see section 10.3.3.21*	Optional	Device SIU Door Status Aggregate. The current state of any door sensors that are available.
<DevSiuIndicatorStatus>	Aggregate	Optional	Device SIU Indicator Status Aggregate. The current state of any indicators that are available.
<DevSiuAuxiliaryStatus>	Aggregate	Optional	Device SIU Auxiliary Status Aggregate. The current state of any auxiliary sensors that are available.
<DevSiuGuidelightStatus>	Aggregate *see section 10.3.3.25*	Optional	Device SIU Guidelight Status Aggregate. The current state of any guidelights that are available.

10.3.3.19 Device Sensors and Indicators Type Aggregate <DevSiuType>

Tag	Type	Usage	Description

Tag	Type	Usage	Description
<DevSiuSensors>	Closed Enum	Required	Device SIU Sensors. Specifies whether general sensors are available.
			Valid Values: Yes, No, NotSupported
<DevSiuDoors>	Closed Enum	Required	Device SIU Doors. Specifies whether door sensors are available.
			Valid Values: Yes, No, NotSupported
<DevSiuIndicators>	Closed Enum	Required	Device SIU Indicators. Specifies whether indicators are available.
			Valid Values: Yes, No, NotSupported
<DevSiuGuidelights>	Closed Enum	Required	Device SIU Guidelights. Specifies whether guidelights are available.
			Valid Values: Yes, No, NotSupported
<DevSiuAuxiliary>	Closed Enum	Required	Device SIU Auxiliary. Specifies whether auxiliary sensors are available.
			Valid Values: Yes, No, NotSupported

10.3.3.20 Device Sensors and Indicators Sensor Status Aggregate <DevSiuSensorStatus>

Tag	Type	Usage	Description
<DevSiuOperatorSwitch>	Open Enum	Optional	Device SIU Operator Switch. The state of the operator switch sensor
			Valid Values: Run, Maintenance, Supervisor, NotAvailable
<DevSiuTamper>	Open Enum	Required	Device SIU Tamper. The state of the tamper sensor
			Valid Values: Off, On, NotAvailable
<DevSiuSeismic>	Open Enum	Required	Device SIU Seismic. The state of the seismic sensor
			Valid Values: Off, On, NotAvailable
<DevSiuHeat>	Open Enum	Required	Device SIU Heat. The state of the external heat (fire) sensor
			Valid Values: Off, On, NotAvailable
<DevSiuProximity>	Open Enum	Required	Device SIU Proximity. The state of the proximity detector
			Valid Values: NotPresent, Present, NotAvailable
<DevSiuAmbientLight>	Open Enum	Required	Device SIU Ambient Light. The state of the ambient light sensor
			Valid Values: VeryDark, Dark, MediumLight, Light, VeryLight, NotAvailable

10.3.3.21 Device Sensors and Indicators Door Status Aggregate <DevSiuDoorStatus>

Tag	Type	Usage	Description
<DevSiuCabinet>	Open Enum	Required	Device SIU Cabinet. The state of the cabinet door(s).
			Valid Values: Closed, Open, Locked, Bolted, NotAvailable

Tag	Type	Usage	Description
\<DevSiuSafe\>	Open Enum	Required	Device SIU Safe. The state of the safe door(s).
			Valid Values: Closed, Open, Locked, Bolted, NotAvailable
\<DevSiuVandalShield\>	Open Enum	Required	Device SIU Vandal Shield. The state of the vandal shield.
			Valid Values: Closed, Open, Locked, Service, Keyboard, Ajar, Jammed, NotAvailable

10.3.3.22 Device Sensors and Indicators Indicator Status Aggregate \<DevSiuIndicatorStatus\>

Tag	Type	Usage	Description
\<DevSiuOpenClose\>	Open Enum	Required	Device SIU Open Close. The state of the open / close indicator.
			Valid Values: Closed, Open, NotAvailable
\<DevSiuFasciaLight\>	Open Enum	Required	Device SIU Fascia Light. The state of the fascia light.
			Valid Values: On, Off, NotAvailable
\<DevSiuAudio\>	Open Enum	Required	Device SIU Audio. The state of the audio indicator.
			Valid Values: Off, KeyClick, Exclamation, Warning, Error, Critical, Continuous, NotAvailable
\<DevSiuHeating\>	Open Enum	Required	Device SIU Heating. The state of the internal heating.
			Valid Values: Off, On, NotAvailable

10.3.3.23 Device Sensors and Indicators Auxiliary Status Aggregate \<DevSiuAuxiliaryStatus\>

Tag	Type	Usage	Description
\<DevSiuVolume\>	Long	Required	Device SIU Volume. An integer (0-1000) indicating the audio volume level.
			0 = Off, 1=Minimum volume, 1000=Maximum volume.
\<DevSiuUps\>	Open Enum	Required	Device SIU Uninterruptable Power Supply. The state of the uninterruptable power supply.
			Valid Values: Available, Low, Engaged, Powering, Recovered, NotAvailable
\<DevSiuRemoteStatus\>	Aggregate *see section 10.3.3.24*	Required	Device SIU Remote Status. The state of the remote status monitor indicators.
\<DevSiuAudibleAlarm\>	Open Enum	Required	Device SIU Audible Alarm. The state of the audible alarm.
			Valid Values: Off, On, NotAvailable

10.3.3.24 Device Sensors and Indicators Remote Status Aggregate \<DevSiuRemoteStatus\>

Tag	Type	Usage	Description

Tag	Type	Usage	Description
<DevSiuGreenLed>	Closed Enum	Required	Device SIU Green LED. The state of the green LED. Valid Values: Off, On, NotAvailable
<DevSiuRedLed>	Closed Enum	Required	Device SIU Red LED. The state of the red LED. Valid Values: Off, On, NotAvailable
<DevSiuAmberLed>	Closed Enum	Required	Device SIU Amber LED. The state of the amber LED. Valid Values: Off, On, NotAvailable

10.3.3.25 Device SIU Guidelight Status Aggregate <DevSiuGuidelightStatus>

Tag	Type	Usage	Description
<DevSiuIdcGuidelight>	Open Enum	Required	Device SIU IDC Guidelight. The state of the card reader guidelight. Valid Values: Off, SlowFlash, MediumFlash, QuickFlash, Continuous, NotAvailable
<DevSiuCdmGuidelight>	Open Enum	Required	Device SIU CDM Guidelight. The state of the note dispenser guidelight. Valid Values: Off, SlowFlash, MediumFlash, QuickFlash, Continuous, NotAvailable
<DevSiuCoinOutGuidelight>	Open Enum	Required	Device SIU Coin Out Guidelight. The state of the coin dispenser guidelight. Valid Values: Off, SlowFlash, MediumFlash, QuickFlash, Continuous, NotAvailable
<DevSiuReceiptGuidelight>	Open Enum	Required	Device SIU Receipt Guidelight. The state of the receipt printer guidelight. Valid Values: Off, SlowFlash, MediumFlash, QuickFlash, Continuous, NotAvailable
<DevSiuPassbookGuidelight>	Open Enum	Required	Device SIU Passbook Guidelight. The state of the passbook printer guidelight. Valid Values: Off, SlowFlash, MediumFlash, QuickFlash, Continuous, NotAvailable
<DevSiuDepGuidelight>	Open Enum	Required	Device SIU DEP Guidelight. The state of the envelope depository guidelight. Valid Values: Off, SlowFlash, MediumFlash, QuickFlash, Continuous, NotAvailable
<DevSiuChkGuidelight>	Open Enum	Required	Device SIU CHK Guidelight. The state of the check reader guidelight. Valid Values: Off, SlowFlash, MediumFlash, QuickFlash, Continuous, NotAvailable
<DevSiuCimGuidelight>	Open Enum	Required	Device SIU CIM Guidelight. The state of the note accept guidelight. Valid Values: Off, SlowFlash, MediumFlash, QuickFlash, Continuous, NotAvailable
<DevSiuDocumentGuidelight>	Open Enum	Required	Device SIU Document Guidelight. The state of the document printer guidelight. Valid Values: Off, SlowFlash, MediumFlash, QuickFlash, Continuous, NotAvailable

Tag	Type	Usage	Description
<DevSiuCoinInGuidelight>	Open Enum	Required	Device SIU Coin In Guidelight. The state of the coin acceptor guidelight.
			Valid Values: Off, SlowFlash, MediumFlash, QuickFlash, Continuous, NotAvailable
<DevSiuScannerGuidelight>	Open Enum	Required	Device SIU Scanner Guidelight. The state of the scanner guidelight.
			Valid Values: Off, SlowFlash, MediumFlash, QuickFlash, Continuous, NotAvailable

10.3.3.26 Device Detail Cash Acceptor Aggregate <DevDetailCim>

Device details specific to cash acceptance devices (devices covered by the XFS CIM service class).

Tag	Type	Usage	Description
<DevCimType>	Open Enum	Required	Device CIM Type. Type of cash accept device.
			Valid Values: TellerBill, SelfServiceBill, TellerCoin, SelfServiceCoin
<DevCimOperation>	Open Enum	Optional	Device CIM Operation. The operation being performed at the time the alert is raised.
			Valid Values: CashInStart, CashIn, CashInEnd, Rollback, Retract, OpenShutter, CloseShutter, StartExchange, EndExchange, OpenSafeDoor, Reset, NotSupported
<DevCimSafeDoor>	Open Enum	Required	Device CIM Safe Door. The status of the safe door.
			Valid Values: Open, Closed, Unknown, NotSupported
<DevCimCashUnits>	Open Enum	Required	Device CIM Cash Units. The status of the acceptor cash units.
			Valid Values: OK, Warning, Fatal, Unknown
<DevCimIntermediateStacker>	Open Enum	Required	Device CIM Intermediate Stacker. The status of the intermediate stacker.
			Valid Values: Empty, NotEmpty, Full, Unknown, NotSupported
<DevCimStackerItems>	Open Enum	Required	Device CIM Stacker Items. Describes whether the items on the intermediate stacker have been in customer access.
			Valid Values: CustomerAccess, NoCustomerAccess, AccessUnknown, NoItems
<DevCimNoteReader>	Open Enum	Required	Device CIM Note Reader. The status of the banknote reader.
			Valid Values: OK, Inoperable, Unknown, NotSupported
<DevCimDropBox>	Open Enum	Required	Device CIM Drop Box. The status of the drop box. The drop box is an area within the CIM where items which have caused a problem during an operation are stored.
			Valid Values: Empty, NotEmpty, NotSupported

Tag	Type	Usage	Description
<DevCimPosition>	Aggregate *see section 10.3.3.27*	Required	Device CIM Position Aggregate.
<DevCimBin>	Aggregate *see section 10.3.3.28*	Required Repeating	Device CIM Bin Aggregate. Aggregate for each supported acceptor bin.

10.3.3.27 Device Cash Acceptance Position Aggregate <DevCimPosition>

Tag	Type	Usage	Description
<DevCimPositionLocation>	Open Enum	Required	Device CIM Position Location. Specifies the location of this position for cash acceptance.
			Valid values: InLeft, InRight, InCenter, InTop, InBottom, InFront, InRear, OutLeft, OutRight, OutCenter, OutTop, OutBottom, OutFront, OutRear
<DevCimShutter>	Open Enum	Required	Device CIM Shutter. The state of the shutter for this position.
			Valid values: Closed, Open, Jammed, Unknown, NotSupported
<DevCimPositionStatus>	Open Enum	Required	Device CIM Position Status. The state of this position.
			Valid values: Empty, NotEmpty, Unknown, NotSupported
<DevCimTransport>	Open Enum	Required	Device CIM Transport. The state of the transport for this position.
			Valid values: OK, Inoperable, Unknown, NotSupported
<DevCimTransportItems>	Open Enum	Required	Device CIM Transport Items. The state of the transport for this position.
			Valid values: Empty, NotEmpty, CustomerAccess, Unknown, NotSupported

10.3.3.28 Device Cash Acceptance Module Bin Aggregate <DevCimBin>

Accepter bin details specific to CIM devices.

Tag	Type	Usage	Description
<DevCimBinNum>	Long	Required	Device CIM Bin Number. Number of the cash unit bin.
<DevCimBinType>	Open Enum	Required	Device CIM Bin Type. Type of media accepted by this device.
			Valid Values: Recycling, CashIn, RepContainer, RetractCassette
<DevCimBinStatus>	Closed Enum	Required	Device CIM Bin Status. The operational status of the currently referenced CIM cash bin.
			Valid Values: OK, Full, High, Empty, Inoperable, Missing, NotAvailable, NoReference, Manipulated
<DevCimBinCurCode>	NC-3	Optional	Device CIM Bin Currency Code. Three-letter currency code as defined by ISO-4217
<DevCimBinDenomination>	Decimal	Optional	Device CIM Bin Denomination. Denomination of the acceptor bin.

Tag	Type	Usage	Description
<DevCimBinCount>	Long	Optional	Device CIM Bin Count. The total number of notes of all types in the cash unit.
<DevCimBinMaxCount>	Long	Optional	Device CIM Bin Maximum Count. The maximum number of notes the cash unit can hold.

10.3.3.29 Device Detail Cash Dispenser Module <DevDetailCdm>

Device details specific to the cash dispenser device (devices covered by the XFS CDM service class).

Tag	Type	Usage	Description
<DevCdmType>	Open Enum	Required	Device CDM Type. Type of dispenser device.
			Valid Values: TellerBill, SelfServiceBill, TellerCoin, SelfServiceCoin
<DevCdmOperation>	Open Enum	Optional	Device CDM Operation. The operation being performed at the time the alert is raised. It is <u>strongly</u> recommended that the following defined values be used rather than custom values in order to ensure maximum portability.
			Valid Values: Dispense, Present, Retract, Reject, Count, CloseShutter, OpenShutter, StartExchange, EndExchange, OpenSafeDoor, Calibrate, Reset, Test
<DevCdmSafeDoor>	Closed Enum	Required	Device CDM Safe Door. The status of the safe door.
			Valid Values: Open, Closed, NotSupported, Unknown
<DevCdmCashUnits>	Open Enum	Required	Device CDM Cash Units. The status of the dispenser cash units.
			Valid Values: OK, Warning, Fatal, Unknown
<DevCdmIntermediateStacker>	Open Enum	Required	Device CDM Intermediate Stacker. The status of the intermediate stacker.
			Valid Values: Empty, NotEmpty, CustomerAccess, NotEmpty, Unknown, NotSupported
<DevCdmPosition>	Aggregate *see section 10.3.3.30*	Required	Device CDM Position Aggregate.
<DevCdmBin>	Aggregate *see section 10.3.3.31*	Required Repeating	Device CDM Bin Aggregate. Aggregate for each supported dispenser bin.
			Note: At least 1 bin is required by the dispenser device.

10.3.3.30 Device Cash Dispenser Position Aggregate <DevCdmPosition>

Tag	Type	Usage	Description
<DevCdmPositionLocation>	Open Enum	Required	Device CDM Position Location. Specifies the location of this position for cash dispense.
			Valid values: OutLeft, OutRight, OutCenter, OutTop, OutBottom, OutFront, OutRear

Tag	Type	Usage	Description
<DevCdmShutter>	Open Enum	Required	Device CDM Shutter. The state of the shutter for this position. Valid values: Closed, Open, Jammed, Unknown, NotSupported
<DevCdmPositionStatus>	Open Enum	Required	Device CDM Position Status. The state of this position. Valid values: Empty, NotEmpty, Unknown, NotSupported
<DevCdmTransport>	Open Enum	Required	Device CDM Transport. The state of the transport for this position. Valid values: OK, Inoperable, Unknown, NotSupported
<DevCdmTransportItems>	Open Enum	Required	Device CDM Transport Items. The state of the transport for this position. Valid values: Empty, NotEmpty, CustomerAccess, Unknown, NotSupported

10.3.3.31 Device Cash Dispenser Module Bin Aggregate <DevCdmBin>

Dispenser bin details specific to CDM devices.

Tag	Type	Usage	Description
<DevCdmBinNum>	Long	Required	Device CDM Bin Number. Number of the dispenser bin.
<DevCdmBinType>	Open Enum	Required	Device CDM Bin Type. Type of dispenser media. Valid Values: RejectCassette, BillCassette, CoinCylinder, CoinDispenser, RetractCassette, Coupon, Document, RepContainer, Recycling, NotApplicable
<DevCdmBinStatus>	Closed Enum	Required	Device CDM Bin Status. The operational status of the currently referenced CDM cash bin. Valid Values: OK, Jammed, NotSupported, NotAvailable, Unknown
<DevCdmBinMedia>	Closed Enum	Required	Device CDM Bin Media. The state of the media in the currently referenced CDM cash bin. Valid Values: OK, High, Full, Low, Empty, NotSupported, Unknown
<DevCdmBinCurCode>	NC-3	Optional	Device CDM Bin Currency Code. Three-letter currency code as defined by ISO-4217
<DevCdmBinDenomination>	Decimal	Optional	Device CDM Bin Denomination. Denomination of the dispenser bin.
<DevCdmBinCount>	Long	Optional	Device CDM Bin Count. The total number of notes of all types in the cash unit.
<DevCdmBinMaxCount>	Long	Optional	Device CDM Bin Maximum Count. The maximum number of notes the cash unit can hold.

10.3.3.32 Device Detail Camera Aggregate <DevDetailCam>

Device details specific to camera devices (devices covered by the XFS CAM service class).

Tag	Type	Usage	Description

Tag	Type	Usage	Description
<DevCamType>	Open Enum	Required	Device CAM Type. Type of Camera device. Only one type is currently defined.
			Valid Values:
			Camera
<DevCamOperation>	Open Enum	Optional	Device CAM Operation. The operation being performed at the time the alert is raised.
			Valid Values:
			TakePicture, Reset, NotSupported
<DevCamStatusArea>	Open Enum	Required	Device CAM Status Area. Area monitored by the camera.
			Valid Values:
			Room, Person, ExitSlot
<DevCamStatusMedia>	Closed Enum	Optional	Device CAM Status Media. The state of the recording media of the camera.
			Valid Values:
			OK, High, Full, NotSupported, Unknown
<DevCamStatusState>	Closed Enum	Optional	Device CAM Status State. State of the camera.
			Valid Values:
			NotSupported, OK, Inoperative, Unknown
<DevCamStatusPictures>	Long	Optional	Device CAM Status Pictures. Number of pictures stored on the recording media of the camera.

10.4   Terminal Object

10.4.1   Terminal Object Add Message

An IFX client uses the <TerminalObjAddRq> message to add a Terminal object. Typically the term "client" is associated with a Terminal and "server" is associated with a Host (e.g. CSP, SP). However, since the Terminal object resides on the Terminal, the IFX client that will use this message will be a Host (e.g. CSP, SP).

10.4.1.1   Request <TerminalObjAddRq>

Tag	Type	Usage	Description
<RqUID>	UUID	Required	Request Identifier. Sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header Aggregate.
<TerminalObjInfo>	Aggregate *see section 10.3.1.1*	Required	Terminal Object Information aggregate.

10.4.1.2   Response <TerminalObjAddRs>

Tag	Type	Usage	Description

Tag	Type	Usage	Description
<Status>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, <StatusCode> defaults to 0 (zero).
<RqUID>	UUID	Required Echoed	Request Identifier. Sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header Aggregate.
<MsgRsHdr>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header Aggregate.
<TerminalObjInfo>	Aggregate *see section 10.3.1.1*	Required Echoed	Terminal Object Information Aggregate.
<TerminalObjRec>	Aggregate *see section 10.3.1*	Required	Terminal Object Record Aggregate.
<CSPRefId>	Identifier	Optional	Customer Service Provider Reference Identifier.
<SPRefId>	Identifier	Optional	Service Provider Reference Identifier.

10.4.2   Terminal Object Modify

An IFX client uses the <TerminalObjModRq> message to modify a Terminal object. Typically the term "client" is associated with a Terminal and "server" is associated with a Host (e.g. CSP, SP). However, since the Terminal object resides on the Terminal, the IFX client that will use this message will be a Host (e.g. CSP, SP).

10.4.2.1   Request <TerminalObjModRq>

Tag	Type	Usage	Description
<RqUID>	UUID	Required	Request Identifier. Sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header Aggregate.
<TerminalObjId>	Identifier	Required	Terminal Object Identifier.
<TerminalObjInfo>	Aggregate *see section 10.3.1.1*	Required	Terminal Object Information Aggregate.

10.4.2.2   Response <TerminalObjModRs>

Tag	Type	Usage	Description
<Status>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, <StatusCode> defaults to 0 (zero).
<RqUID>	UUID	Required Echoed	Request Identifier. Sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header Aggregate.

<MsgRsHdr>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header Aggregate.
<TerminalObjId>	Identifier	Required Echoed	Terminal Object Identifier.
<TerminalObjInfo>	Aggregate *see section 10.3.1.1*	Required Echoed	Terminal Object Information Aggregate.
<TerminalObjRec>	Aggregate *see section 10.3.1*	Required	Terminal Object Record Aggregate.
<CSPRefId>	Identifier	Optional	Customer Service Provider Reference Identifier.
<SPRefId>	Identifier	Optional	Service Provider Reference Identifier.

10.4.3   Terminal Object Inquiry

An IFX client uses the <TerminalObjInqRq> message to retrieve Terminal objects. Typically the term "client" is associated with a Terminal and "server" is associated with a Host (e.g. CSP, SP). However, since the Terminal object resides on the Terminal, the IFX client that will use this message will be a Host (e.g. CSP, SP).

10.4.3.1   Request <TerminalObjInqRq>

Tag	*Type*	*Usage*	*Description*
<RqUID>	UUID	Required	Request Identifier. Sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header Aggregate.
<TerminalObjId>	Identifier	Optional	Terminal Object Identifier.
			This field is used as a selection criterion.

10.4.3.2   Response <TerminalObjInqRs>

Tag	*Type*	*Usage*	*Description*
<Status>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, <StatusCode> defaults to 0 (zero).
<RqUID>	UUID	Required Echoed	Request Identifier. Sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header Aggregate.
<MsgRsHdr>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header Aggregate.
<TerminalObjId>	Identifier	Optional Echoed	Terminal Object Identifier.
<TerminalObjRec>	Aggregate *see section 10.3.1*	Optional Repeating	Terminal Object Record Aggregate.

10.4.4   Terminal Object Delete

An IFX client uses the <TerminalObjDelRq> message to delete a Terminal object. Typically the term "client" is associated with a Terminal and "server" is associated with a Host (e.g. CSP, SP). However, since the Terminal object resides on the Terminal, the IFX client that will use this message will be a Host (e.g. CSP, SP).

10.4.4.1   Request <TerminalObjDelRq>

Tag	Type	Usage	Description
<RqUID>	UUID	Required	Request Identifier. Sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header Aggregate.
<TerminalObjId>	Identifier	Required	Terminal Object Identifier.

10.4.4.2   Response <TerminalObjDelRs>

Tag	Type	Usage	Description
<Status>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, <StatusCode> defaults to 0 (zero).
<RqUID>	UUID	Required Echoed	Request Identifier. Sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header Aggregate.
<MsgRsHdr>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header Aggregate.
<TerminalObjId>	Identifier	Required Echoed	Terminal Object Identifier.
<TerminalObjRec>	Aggregate *see section 10.3.1*	Optional	Terminal Object Record Aggregate.
<CSPRefId>	Identifier	Optional	Customer Service Provider Reference Identifier.
<SPRefId>	Identifier	Optional	Service Provider Reference Identifier.

10.4.5   Terminal Object Audit

An IFX client uses the <TerminalObjAudRq> message to audit messages associated with a particular Terminal object. Typically the term "client" is associated with a Terminal and "server" is associated with a Host (e.g. CSP, SP). However, since the Terminal object resides on the Terminal, the IFX client that will use this message will be a Host (e.g. CSP, SP).

10.4.5.1   Request <TerminalObjAudRq>

Tag	Type	Usage	Description
<RqUID>	UUID	Required	Request Identifier. Sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.

Tag	Type	Usage	Description
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header Aggregate.
<RecCtrlIn>	Aggregate *see section 3.2.12.2.1*	Optional	Records Control Input Aggregate.
<SelRangeDt>	Aggregate *see section 3.2.10.1*	Optional	Selection Range Date Aggregate.
<Method>	Closed Enum	Optional Repeating	Audit Selection Action. Used to identify actions associated with the object that is being audited (e.g. transfer). Valid values: Add, Mod, Del. This field is used as a selection criterion.
<TerminalObjId>	Identifier	Required Repeating	Terminal Object Identifier.

10.4.5.2   Response <TerminalObjAudRs>

Tag	Type	Usage	Description
<Status>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, <StatusCode> defaults to 0 (zero).
<RqUID>	UUID	Required Echoed	Request Identifier. Sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header Aggregate.
<MsgRsHdr>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header Aggregate.
<RecCtrlOut>	Aggregate *see section 3.2.12.2.2*	Optional but see Description	Message Records Control Output Aggregate. Required if <RecCtrlIn> was provided in the request and the server supports Records Control.
<SelRangeDt>	Aggregate *see section 3.2.10.1*	Optional Echoed	Selection Range Date Aggregate.
<Method>	Closed Enum	Optional Repeating Echoed	Audit Selection Action. Used to identify actions associated with the object that is being audited (e.g. transfer). This field is used as a selection criterion.
<TerminalObjId>	Identifier	Required Repeating Echoed	Terminal Object Identifier.
<TerminalObjMsgRec>	Aggregate	Optional Repeating	Terminal Object Message Record Aggregate.
<MsgRecDt>	Date/Time	Optional	Message Record Creation Date. The date/time at which the message record was stored/created by the service provider.
<TerminalObjAddRs>	Aggregate *see section 10.4.1.2*	Required XOR	Terminal Object Add Response Message Aggregate.

Tag	Type	Usage	Description
<Status>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, <StatusCode> defaults to 0 (zero).
<RqUID>	UUID	Required Echoed	Request Identifier. Sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header Aggregate.
<MsgRsHdr>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header Aggregate.
<TerminalObjModRs>	Aggregate *see section 10.4.2.2*	Required XOR	Terminal Object Modify Response Message Aggregate.
<TerminalObjDelRs>	Aggregate *see section 10.4.4.2*	Required XOR	Terminal Object Delete Response Message Aggregate.
</TerminalObjMsgRec>			
<CSPRefId>	Identifier	Optional	Customer Service Provider Reference Identifier.
<SPRefId>	Identifier	Optional	Service Provider Reference Identifier.

10.4.6   Terminal Object Status Modify

An IFX client uses the <TerminalObjStatusModRq> message to modify the status of a particular Terminal object. Typically the term "client" is associated with a Terminal and "server" is associated with a Host (e.g. CSP, SP). However, since the Terminal object resides on the Terminal, the IFX client that will use this message will be a Host (e.g. CSP, SP).

10.4.6.1   Request <TerminalObjStatusModRq>

Tag	Type	Usage	Description
<RqUID>	UUID	Required	Request Identifier. Sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header Aggregate.
<TerminalObjId>	Identifier	Required	Terminal Object Identifier.
<RestartType>	Open Enum	Optional	Restart Type. Requests that the terminal perform a restart prior to changing the operation mode. Valid Values: Restart
<EffUrgency>	Closed Enum	Optional	Effective Urgency. This element indicates the urgency that a terminal operation mode should take effect. Valid Values: Immediate, CurrentTransactionEnd, SessionEnd

Tag	Type	Usage	Description
<ExpectedOpenDt>	DateTime	Optional	Expected Open DateTime. The terminal date/time that the terminal operation mode is expected to return to "Open" if the operation mode is being changed to something other than "Open". There is no obligation inferred. A Terminal application could use this information to inform customers when the Terminal may again be available.
<TerminalObjStatus>	Aggregate *see section 10.3.1.2*	Required	Terminal Object Status Aggregate.

10.4.6.2   Response <TerminalObjStatusModRs>

Tag	Type	Usage	Description
<Status>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, <StatusCode> defaults to 0 (zero).
<RqUID>	UUID	Required Echoed	Request Identifier. Sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header Aggregate.
<MsgRsHdr>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header Aggregate.
<TerminalObjId>	Identifier	Required Echoed	Terminal Object Identifier.
<RestartType>	Open Enum	Optional Echoed	Restart Type.
<EffUrgency>	Closed Enum	Optional Echoed	Effective Urgency.
<ExpectedOpenDt>	DateTime	Optional Echoed	Expected Open DateTime.
<TerminalObjStatus>	Aggregate *see section 10.3.1.2*	Required Echoed	Terminal Object Status Aggregate.
<TerminalObjRec>	Aggregate *see section 10.3.1*	Required	Terminal Object Record Aggregate.
<CSPRefId>	Identifier	Optional	Customer Service Provider Reference Identifier.
<SPRefId>	Identifier	Optional	Service Provider Reference Identifier.

10.4.7   Terminal Object Advise

Since the Terminal object resides on the Terminal, the Terminal will use the <TerminalObjAdviseRq> message to inform other IFX entities (e.g. Hosts, CSP, SP) of changes to the Terminal object.

If it is not necessary to Advise of changes to the data elements in the <TerminalObjInfo> aggregate then the <TerminalObjId> and <TerminalObjStatus> would be sent in the request. Otherwise, the entire <TerminalObjRec> aggregate should be sent.

10.4.7.1   Request <TerminalObjAdviseRq>

Tag	Type	Usage	Description
<RqUID>	UUID	Required	Request Identifier. Sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header Aggregate.
<TerminalObjRec>	Aggregate *see section 10.3.1*	Required XOR	Terminal Object Record Aggregate.
<TerminalObjId>	Identifier	Required XOR	Terminal Object Identifier.
<TerminalObjStatus>	Aggregate *see section 10.3.1.2*	Required	Terminal Object Status Aggregate.

10.4.7.2   Response <TerminalObjAdviseRs>

Tag	Type	Usage	Description
<Status>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, <StatusCode> defaults to 0 (zero).
<RqUID>	UUID	Required Echoed	Request Identifier. Sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header Aggregate.
<MsgRsHdr>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header Aggregate.
<TerminalObjRec>	Aggregate *see section 10.3.1*	Required XOR Echoed	Terminal Object Record Aggregate.
<TerminalObjId>	Identifier	Required XOR Echoed	Terminal Object Identifier.
<TerminalObjStatus>	Aggregate *see section 10.3.1.2*	Required Echoed	Terminal Object Status Aggregate.

10.5   Terminal Service Provider Object

10.5.1   Terminal Service Provider Object Add Message

An IFX client uses the <TerminalSPObjAddRq> message to add a Terminal Service Provider object.

10.5.1.1   Request <TerminalSPObjAddRq>

Tag	Type	Usage	Description

Tag	Type	Usage	Description
<RqUID>	UUID	Required	Request Identifier. Sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header Aggregate.
<TerminalSPObjInfo>	Aggregate *see section 10.3.2.1*	Required	Terminal Object Information aggregate.

10.5.1.2   Response <TerminalSPObjAddRs>

Tag	Type	Usage	Description
<Status>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, <StatusCode> defaults to 0 (zero).
<RqUID>	UUID	Required Echoed	Request Identifier. Sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header Aggregate.
<MsgRsHdr>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header Aggregate.
<TerminalSPObjInfo>	Aggregate *see section 10.3.2.1*	Required Echoed	Terminal Service Provider Object Information Aggregate.
<TerminalSPObjRec>	Aggregate	Required	Terminal Service Provider Object Record Aggregate.
<CSPRefId>	Identifier	Optional	Customer Service Provider Reference Identifier.
<SPRefId>	Identifier	Optional	Service Provider Reference Identifier.

10.5.2   *Terminal Service Provider Object Modify*

An IFX client uses the <TerminalSPObjModRq> message to modify a Terminal Service Provider object.

10.5.2.1   Request <TerminalSPObjModRq>

Tag	Type	Usage	Description
<RqUID>	UUID	Required	Request Identifier. Sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header Aggregate.
<TerminalSPObjId>	Identifier	Required	Terminal Service Provider Object Identifier.
<TerminalSPObjInfo>	Aggregate *see section 10.3.2.1*	Required	Terminal Service Provider Object Information Aggregate.

10.5.2.2 Response <TerminalSPObjModRs>

Tag	Type	Usage	Description
<Status>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, <StatusCode> defaults to 0 (zero).
<RqUID>	UUID	Required Echoed	Request Identifier. Sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header Aggregate.
<MsgRsHdr>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header Aggregate.
<TerminalSPObjId>	Identifier	Required Echoed	Terminal Service Provider Object Identifier.
<TerminalSPObjInfo>	Aggregate *see section 10.3.2.1*	Required Echoed	Terminal Service Provider Object Information Aggregate.
<TerminalSPObjRec>	Aggregate	Required	Terminal Service Provider Object Record Aggregate.
<CSPRefId>	Identifier	Optional	Customer Service Provider Reference Identifier.
<SPRefId>	Identifier	Optional	Service Provider Reference Identifier.

10.5.3 Terminal Service Provider Object Inquiry

An IFX client uses the <TerminalSPObjInqRq> message to retrieve Terminal Service Provider objects.

10.5.3.1 Request <TerminalSPObjInqRq>

Tag	Type	Usage	Description
<RqUID>	UUID	Required	Request Identifier. Sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header Aggregate.
<RecCtrlIn>	Aggregate *see section 3.2.12.2.1*	Optional	Records Control Input Aggregate.
<TerminalSPObjId>	Identifier	Optional Repeating	Terminal Service Provider Object Identifier. This field is used as a selection criterion.
<SPName>	Identifier	Optional Repeating	Service Provider Name. This field is used as a selection criterion.
<TerminalId>	Identifier	Optional Repeating	Terminal Identifier. This field is used as a selection criterion.
<TerminalSPObjStatusCode>	Closed Enum	Optional Repeating	Terminal Service Provider Object Status Code. This field is used as a selection criterion.
<CSPRefId>	Identifier	Optional Repeating	Customer Service Provider Reference Identifier. This field is used as a selection criterion.

Tag	Type	Usage	Description
<SPRefId>	Identifier	Optional Repeating	Service Provider Reference Identifier. This field is used as a selection criterion.

10.5.3.2   Response <TerminalSPObjInqRs>

Tag	Type	Usage	Description
<Status>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, <StatusCode> defaults to 0 (zero).
<RqUID>	UUID	Required Echoed	Request Identifier. Sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header Aggregate.
<MsgRsHdr>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header Aggregate.
<RecCtrlOut>	Aggregate *see section 3.2.12.2.2*	Optional *but see Description*	Records Control Output Aggregate. Required if <RecCtrlIn> was provided in the request and the server supports Records Control.
<TerminalSPObjId>	Identifier	Optional Repeating Echoed	Terminal Service Provider Object Identifier.
<SPName>	Identifier	Optional Repeating Echoed	Service Provider Name.
<TerminalId>	Identifier	Optional Repeating Echoed	Terminal Identifier.
<TerminalSPObjStatusCode>	Closed Enum	Optional Repeating Echoed	Terminal Service Provider Object Status Code.
<CSPRefId>	Identifier	Optional Repeating Echoed	Customer Service Provider Reference Identifier.
<SPRefId>	Identifier	Optional Repeating Echoed	Service Provider Reference Identifier.
<TerminalSPObjRec>	Aggregate	Optional Repeating	Terminal Service Provider Object Record Aggregate.

10.5.4   Terminal Service Provider Object Delete

An IFX client uses the <TerminalSPObjDelRq> message to delete a Terminal Service Provider object.

10.5.4.1   Request <TerminalSPObjDelRq>

Tag	Type	Usage	Description
<RqUID>	UUID	Required	Request Identifier. Sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.

Tag	Type	Usage	Description
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header Aggregate.
<TerminalSPObjId>	Identifier	Required	Terminal Service Provider Object Identifier.

10.5.4.2   Response <TerminalSPObjDelRs>

Tag	Type	Usage	Description
<Status>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, <StatusCode> defaults to 0 (zero).
<RqUID>	UUID	Required Echoed	Request Identifier. Sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header Aggregate.
<MsgRsHdr>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header Aggregate.
<TerminalSPObjId>	Identifier	Required Echoed	Terminal Service Provider Object Identifier.
<TerminalSPObjRec>	Aggregate	Optional	Terminal Service Provider Object Record Aggregate.
<CSPRefId>	Identifier	Optional	Customer Service Provider Reference Identifier.
<SPRefId>	Identifier	Optional	Service Provider Reference Identifier.

10.5.5   Terminal Service Provider Object Audit

An IFX client uses the <TerminalSPObjAudRq> message to audit messages associated with a particular Terminal Service Provider object.

10.5.5.1   Request <TerminalSPObjAudRq>

Tag	Type	Usage	Description
<RqUID>	UUID	Required	Request Identifier. Sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header Aggregate.
<RecCtrlIn>	Aggregate *see section 3.2.12.2.1*	Optional	Records Control Input Aggregate.
<SelRangeDt>	Aggregate *see section 3.2.10.1*	Optional	Selection Range Date Aggregate.
<Method>	Closed Enum	Optional Repeating	Audit Selection Action. Used to identify actions associated with the object that is being audited (e.g. transfer). Valid values: Add, Mod, Del This field is used as a selection criterion.

Tag	Type	Usage	Description
<RqUID>	UUID	Required	Request Identifier. Sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header Aggregate.
<TerminalSPObjId>	Identifier	Required Repeating	Terminal Service Provider Object Identifier.

10.5.5.2 Response <TerminalSPObjAudRs>

Tag	Type	Usage	Description
<Status>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, <StatusCode> defaults to 0 (zero).
<RqUID>	UUID	Required Echoed	Request Identifier. Sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header Aggregate.
<MsgRsHdr>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header Aggregate.
<RecCtrlOut>	Aggregate *see section 3.2.12.2.2*	Optional but see Description	Message Records Control Output Aggregate. Required if <RecCtrlIn> was provided in the request and the server supports Records Control.
<SelRangeDt>	Aggregate *see section 3.2.10.1*	Optional Echoed	Selection Range Date Aggregate.
<Method>	Closed Enum	Optional Repeating Echoed	Audit Selection Action. Used to identify actions associated with the object that is being audited (e.g. transfer). This field is used as a selection criterion.
<TerminalSPObjId>	Identifier	Required Repeating Echoed	Terminal Service Provider Object Identifier.
<TerminalSPObjMsgRec>	Aggregate	Optional Repeating	Terminal Service Provider Object Message Record Aggregate.
<MsgRecDt>	Date/Time	Optional	Message Record Creation Date. The date/time at which the message record was stored/created by the service provider.
<TerminalSPObjAddRs>	Aggregate *see section 10.5.1.2*	Required XOR	Terminal Service Provider Object Add Response Message Aggregate.
<TerminalSPObjModRs>	Aggregate *see section 10.5.2.2*	Required XOR	Terminal Service Provider Object Modify Response Message Aggregate.
<TerminalSPObjDelRs>	Aggregate *see section 10.5.4.2*	Required XOR	Terminal Service Provider Object Delete Response Message Aggregate.
</TerminalSPObjMsgRec>			

Tag	Type	Usage	Description
<Status>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, <StatusCode> defaults to 0 (zero).
<RqUID>	UUID	Required Echoed	Request Identifier. Sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header Aggregate.
<MsgRsHdr>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header Aggregate.
<CSPRefId>	Identifier	Optional	Customer Service Provider Reference Identifier.
<SPRefId>	Identifier	Optional	Service Provider Reference Identifier.

10.5.6   Terminal Service Provider Object Advise

The <TerminalSPObjAdvise> message is used by a Service Provider to inform a Terminal that a change has occurred in the Terminal Service Provider Object. The primary use of the message is to inform a Terminal that service is available/unavailable from the Service Provider. Service is defined as a collection of one or more messages.

If it is not necessary to Advise of changes to the data elements in the <TerminalSPObjInfo> aggregate then the <TerminalSPObjId> and <TerminalSPObjStatus> would be sent in the request. Otherwise, the entire <TerminalSPObjRec> aggregate should be sent.

10.5.6.1   Request <TerminalSPObjAdviseRq>

Tag	Type	Usage	Description
<RqUID>	UUID	Required	Request Identifier. Sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header Aggregate.
<TerminalSPObjRec>	Aggregate	Required XOR	Terminal Service Provider Object Record Aggregate.
<TerminalSPObjId>	Identifier	Required XOR	Terminal Service Provider Object Identifier.
<TerminalSPObjStatus>	Aggregate *see section 10.3.2.2*	Required	Terminal Service Provider Object Status Aggregate.

10.5.6.2   Response <TerminalSPObjAdviseRs>

Tag	Type	Usage	Description
<Status>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, <StatusCode> defaults to 0 (zero).
<RqUID>	UUID	Required Echoed	Request Identifier. Sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.

<MsgRqHdr>	Aggregate *see section* 3.2.1.1	Optional Echoed	Message Request Header Aggregate.
<MsgRsHdr>	Aggregate *see section* 3.2.1.2	Optional	Message Response Header Aggregate.
<TerminalSPObjRec>	Aggregate	Required XOR Echoed	Terminal Service Provider Object Record Aggregate.
<TerminalSPObjId>	Identifier	Required XOR Echoed	Terminal Service Provider Object Identifier.
<TerminalSPObjStatus>	Aggregate *see section* 10.3.2.2	Required Echoed	Terminal Service Provider Object Status Aggregate.

10.6 Device Messages

An IFX client uses the <DevInqRq> message to retrieve Device records from an IFX Terminal that can have one or more Device records. Each Device record contains the information and status of a particular device at an IFX Terminal. Typically the term "client" is associated with a Terminal and "server" is associated with a Host (e.g. CSP, SP). However, since the Device records reside on the Terminal, the IFX client that will use this message will be a Host (e.g. CSP, SP).

10.6.1 Device Inquiry

10.6.1.1 Request <DevInqRq>

Tag	Type	Usage	Description
<RqUID>	UUID	Required	Request Identifier. Sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.
<MsgRqHdr>	Aggregate *see section* 3.2.1.1	Optional	Message Request Header Aggregate.
<RecCtrlIn>	Aggregate *see section* 3.2.12.2.1	Optional	Records Control Input Aggregate.
<DevId>	Identifier	Optional Repeating	Device Identifier. This field is used as a selection criterion.
<DevName>	C-255	Optional Repeating	Device Name. Human readable name of the device. This field is used as a selection criterion.
<DevType>	Closed Enum	Optional Repeating	Device Type. Device Type to return (i.e. as defined by <DevDetailXxx> Aggregates). For example if two DevType values are included in this inquiry with values of Idc and Ptr, the devices are filtered to only return devices that contain <DevDetailIdc> and <DevDetailPtr> aggregates. Valid Values: Idc, Ptr, Cdm, PIN, Chk, Dep, Ttu, Siu, Vdm, Cam, Alm, Ceu, Cim, App. This field is used as a selection criterion.

<DevStatusCode>	Closed Enum	Optional Repeating	Device Status Code. State-of-Health (SOH) of the devices to return. Devices returned should contain one of DevStatusCode values included in this request. For example if two DevStatusCode values are included in this inquiry with values of Warning and Fatal, the devices are filtered to only return devices with a status of Warning and Fatal.
			Valid Values:
			Healthy, Warning, Fatal, Suspend
			This field is used as a selection criterion.
<CSPRefId>	Identifier	Optional Repeating	Customer Service Provider Reference Identifier.
			This field is used as a selection criterion.
<SPRefId>	Identifier	Optional Repeating	Service Provider Reference Identifier.
			This field is used as a selection criterion.

10.6.1.2 Response <DevInqRs>

Tag	Type	Usage	Description
<Status>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, <StatusCode> defaults to 0 (zero).
<RqUID>	UUID	Required Echoed	Request Identifier. Sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.
<MsgRqHdr>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header Aggregate.
<MsgRsHdr>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header Aggregate.
<RecCtrlOut>	Aggregate *see section 3.2.12.2.2*	Optional *but see Description*	Records Control Output Aggregate. Required if <RecCtrlIn> was provided in the request and the server supports Records Control.
<DevId>	Identifier	Optional Repeating Echoed	Device Identifier.
<DevName>	C-255	Optional Repeating Echoed	Device Name.
<DevType>	Closed Enum	Optional Repeating Echoed	Device Type.
<DevStatusCode>	Closed Enum	Optional Repeating Echoed	Device Status Code.
<CSPRefId>	Identifier	Optional Repeating Echoed	Customer Service Provider Reference Identifier.
<SPRefId>	Identifier	Optional Repeating Echoed	Service Provider Reference Identifier.

Tag	Type	Usage	Description
\<DevRec\>	Aggregate *see section 10.3.3*	Optional Repeating	Device Record Aggregate.

10.6.2   Device Advise

An IFX Terminal uses the \<DevAdviseRq\> message to inform other IFX entities (e.g. Hosts, CSP, SP) of a change in the status of a device at the Terminal. It is expected the \<DevAdviseRq\> message will be sent to indicate both device faults, and when device faults have been resolved. If one transaction causes multiple device faults, then one \<DevAdviseRq\> message will be transmitted for each device fault.

10.6.2.1   Request \<DevAdviseRq\>

Tag	Type	Usage	Description
\<RqUID\>	UUID	Required	Request Identifier. Sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.
\<MsgRqHdr\>	Aggregate *see section 3.2.1.1*	Optional	Message Request Header Aggregate.
\<CustId\>	Aggregate *see section 3.2.3.1*	Optional	Customer Identifier. The customer at the terminal at the time of the failure.
\<TrnRqUID\>	UUID	Optional Repeating	Transaction RqUID. The \<RqUID\>(s) of requests attempted to complete when failure occurred.
\<ReconRqUID\>	UUID	Optional Repeating	Reconciliation RqUID. RqUID(s) of requests issued to attempt to reconcile the \<TrnRqUID\>(s) to what actually happened at the device.
\<DevRec\>	Aggregate *see section 10.3.3*	Required XOR	Device Record Aggregate.
\<DevId\>	Identifier	Required XOR	Device Identifier.
\<DevStatus\>	Aggregate *see section 10.3.3.2*	Required	Device Status Aggregate.

10.6.2.2   Response \<DevAdviseRs\>

Tag	Type	Usage	Description
\<Status\>	Aggregate *see section 3.2.12.1*	Optional	Response Status Aggregate. If this aggregate is absent, \<StatusCode\> defaults to 0 (zero).
\<RqUID\>	UUID	Required Echoed	Request Identifier. Sent by a client as a universally unique identifier for the message. Used to correlate responses with requests.
\<MsgRqHdr\>	Aggregate *see section 3.2.1.1*	Optional Echoed	Message Request Header Aggregate.
\<MsgRsHdr\>	Aggregate *see section 3.2.1.2*	Optional	Message Response Header Aggregate.

<CustId>	Aggregate *see section 3.2.3.1*	Optional Echoed	Customer Identifier.
<TrnRqUID>	UUID	Optional Repeating Echoed	Transaction RqUID.
<ReconRqUID>	UUID	Optional Repeating Echoed	Reconciliation RqUID.
<DevRec>	Aggregate *see section 10.3.3*	Required XOR Echoed	Device Record Aggregate.
<DevId>	Identifier	Required XOR Echoed	Device Identifier.
<DevStatus>	Aggregate *see section 10.3.3.2*	Required Echoed	Device Status Aggregate.

10.7 Root Service Profile Information <RootSvcProfInfo>

Tag	Type	Usage	Description
<SvcCore>	Aggregate *see section 5.2.2.1*	Required	Service Core Aggregate. Information specified for every service.
<MsgSupt>	Open Enum	Optional Repeating	Messages Supported for the Root Service. Defined values: TerminalObjAdd, TerminalObjMod, TerminalObjInq, TerminalObjDel, TerminalObjAud, TerminalObjStatusMod, TerminalObjAdvise, TerminalSPObjAdd, TerminalSPObjMod, TerminalSPObjInq, TerminalSPObjDel, TerminalSPObjAud, TerminalSPObjAdvise, DevInq, DevAdvise
<OptSupt>	Open Enum	Optional Repeating	Options Supported. Defined values: RecCtrl, SuppressEcho

A   Response Status Codes

The following table provides the complete set of valid Response Status codes defined for the Response Status Aggregate <Status> in Section 3.2.12.1, along with conditions that may cause the Response Status to be sent.

Default text is provided in the <Desc> field, which is intended to be human-readable. The service provider may return this default text to the client, or may replace it with custom text that is more specific or translated into the appropriate language. The client may display the text it receives from the service provider, or may replace it with custom text that is more specific or translated into the appropriate language. The <StatusCode> and <Severity> elements are used by the system and must be included as they appear in the table.

Negative <StatusCode> values (values less than zero) may be used for custom conditions. Clients receiving negative values for <StatusCode> with <Severity> *Warn* or *Info* must treat the response as if it were <StatusCode> 0 (Success). Clients receiving negative values for <StatusCode> with <Severity> *Error* must treat the response as if it were <StatusCode> 100 (General Error).

<StatusCode>	<Severity>	<Desc> (Default Text)	Condition
0	Info	Success	The service provider successfully processed the request.
1	Info	Client Up to Date	The client's indicator (i.e., date-time or token) of last update in the request is equal to the server's indicator of last update processed.
100	Error	General Error	There was an error that prevented the service provider from processing the transaction. No additional information is provided.
200	Error	General Data Error	One or more of the elements in the request is either invalid or is inconsistent with other elements. No additional information is available.
300	Error	System Not Available	The service provider for this transaction is not available due to a technical problem. Try again later
400	Error	Function Not Available	The function selected is not available. Other functions may be available.
500	Error	Unsupported Service	The service provider does not support the specified service offering.
600	Error	Unsupported Message	The server does not support the message.
700	Error	Unsupported Function	The server does not support one or more functions within the request.
800	Error	Object Already Committed	The object cannot be modified or canceled because it has already been committed for processing.
810	Error	Message Cannot Be Reversed	The server is unable to reverse the requested message.
900	Warn	Message Accepted for Asynchronous Processing	The message was accepted for future processing.
910	Error	Asynchronous Request Does Not Match Original Request	The message does not match the message referenced by the <AsyncRqUID>.
1000	Error	Duplicate <RqUID>	A request with this client message identifier <RqUID> has already been received and processed.
1010	Error	The <Cursor> returned within <RecCtrlIn> is invalid or has expired	The suggested action is to resend the original message, omitting the cursor, to restart the sequence of messages required to retrieve all of the records.
1020	Error	Required Element Not Included	The request message does not contain one or more required elements.
1040	Warn	Request Declined	The Service Provider has declined your request (e.g., a no pay was received for a payment or a credit limit change request was declined).

1050	Error	Invalid Enum Value	Customer input of an Enum value is not valid. The condition may echo the element name in error or the value input by the Customer.
1060	Error	Cannot Modify Element	The service provider does not allow modifications to one or more elements in a modification request.
1080	Warn	No Changes Made	No changes have been made. The requested change in a <xxxModRq> matched the existing data on the server.
1100	Warn	Too Many Records Requested	The request has asked for more records than the service provider can satisfy. The quantity that the service provider can satisfy has been provided.
1110	Info	Records Control Active	Records control is being used in the response, and RecCtrlOut is present in the response.
1120	Info	No Records Match Selection Criteria	No records match the selection criteria of the request.
1140	Warn	Some Selection Criteria Not Supported	The server does not support one or more of the selection criterion fields in the request. Records returned based on only those supported.
1160	Error	None of Selection Criteria Supported	The server supports none of the selection criterion fields specified in the request. No records returned.
1180	Error	Token Value Prior to Available History	The client has sent a token that precedes the history currently available. The client must do an inquiry to resync.
1200	Error	Invalid Token	The token is not of the valid format. Used when the client may edit the token format.
1220	Error	Invalid Identifier	The reference identifier used is invalid.
1240	Warn	Detail Not Available	Detail for this item is not available at this time.
1260	Error	Unknown Object ID	The specified object ID does not exist.
1280	Warn	Object already exists.	The object that the client requested to add already exists. (e.g., the specified service is already activated for the customer or customer account, or customer is already enrolled.)
1300	Warn	Object not deleted; dependent objects exist	The object was not deleted because dependent objects exist, and the client did not request a cascading delete.
1310	Error	Cascade Delete Failed	The object was not deleted because the dependent objects could not be deleted at this time.
1320	Error	Object exists, but does not match <CustId>.	The object exists but does not match the <CustId>. Customer is not authorized to act.
1360	Error	<SPName> Invalid	The Service Provider name specified in a request was not found or invalid.
1380	Error	<SPName> within <CustId> is invalid	The Service Provider Name specified within a customer identification aggregate is invalid.
1400	Error	<SPName> within <BillerId> is invalid.	The Service Provider Name specified within a biller identification aggregate is invalid.
1420	Error	<CustId> invalid	The customer identifier <CustId> specified is invalid or not found.
1480	Error	Edit Mask Error	The value specified does not match the edit mask.
1500	Error	Invalid Address	The specified address is incorrect.
1520	Error	Invalid City	The city specified was incorrect.
1540	Error	Invalid State Or Province	The state or province specified was incorrect.
1560	Error	Invalid Postal Code	The postal code specified was incorrect.
1580	Error	Invalid Country Code	The specified country code is not valid in ISO 3166.

1600	Error	<CryptType> not valid	<CryptType> not valid or not supported.
1620	Error	No <SPName>	Ambiguous request, <SPName> required
1640	Error	No <SvcName>	Ambiguous request, <SvcName> required
1700	Error	Security Violation	A security violation has occurred.
1720	Info	Customer Login ID And Password Will Be Sent Out-Of-Band	The service provider will send the customer Login ID and password through postal mail, e-mail, or some other means.
1740	Error	Authentication Failed	The customer could not be authenticated due to an incorrect login ID or password.
1760	Error	Authorization Failure	Customer is not eligible to perform this function
1790	Error	OBO is not active for this <CustId>	The currently logged-in customer does not have the authority to act on behalf of another user.
1800	Error	Service not Enabled	The customer has not enabled the service with the service provider.
1820	Error	Customer Session Already In Progress	The service provider supports only one session at a time per customer and there is already an active session for this <CustId>. Please try again later.
1840	Error	No Customer Session In Progress	The specified customer does not have a session in progress.
1860	Error	Immediate Customer Lock Out	The customer identity for the current session is suspect. End any active sessions.
1880	Error	Customer Locked Out	The service provider has received too many failed authentication attempts for this customer or has detected other suspicious activity. Please call the technical support telephone number.
1900	Warn	Password Change Required	The Logon is successful but the customer must enter a new password before any other messages will be allowed.
1910	Error	Must Change Password	The customer must change his or her password before any other messages will be allowed.
1920	Error	Invalid New Customer Identifier	The new customer ID entered is not valid because it does not meet edit criteria.
1940	Error	New Customer Identifier in use	The customer chose a customer identifier that is already in use.
1970	Error	Invalid New password	The new password entered as part of a <CustPswdModRq> is invalid.
1980	Error	Unsupported Application ID	The ID of the client application is unsupported.
2000	Error	Invalid Data in Single or Low Amount Field	Non-numeric or missing data was encountered in a single or low amount field.
2010	Error	Invalid Data in High Amount Field	Non-numeric or missing data was encountered in a high amount field.
2020	Error	Amount Too Small	The amount entered is too small (e.g., less than the minimum required). Please enter a larger amount.
2030	Error	Amount Too Large	The amount entered is too large (e.g., exceeds the limit for this transaction). Please enter a smaller amount.
2060	Error	Invalid High/Low Amount	Low amount is greater than high amount in an amount range.
2080	Error	Final Amount Exceeds Limit	The service provider has established a maximum value for the amount of the final transfer or payment, which has been exceeded.

2100	Error	Invalid Datetime - Single date or Low of Range	The specified single date/time or the low date/time in a range is not in a valid date/time format.
2110	Error	Invalid Datetime - High of Range	The specified high date/time in a range is not in a valid date/time format.
2120	Error	Invalid Datetime Range	The date range specified is invalid because the low date exceeds the high date.
2130	Error	Date Passed	The date entered is in the past. Please enter a valid date.
2140	Error	Datetime Too Soon	The specified date and time is too near; enter a later date and time.
2150	Error	Datetime Too Far In Future	The specified date and time is too far in the future. Reenter an earlier date.
2160	Error	Processing Date Precedes Today	The processing date required to meet the due date <DueDt> entered would be earlier than today.
2170	Warn	Requested Date Adjusted	The Service Provider successfully processed the request, but the date requested has been adjusted.
2180	Error	Request Is Too Late For Today's Work	The customer has asked to schedule a transfer or payment for today but the cutoff time for today's work has passed.
2190	Error	Weekend Or Holiday	The date entered is a weekend or holiday. The date cannot be adjusted.
2200	Warn	History Not Available for Full Date Range	History is not available for the full date range specified. All transactions that were available were returned.
2210	Warn	Account Balance Information Not Available	One or more account balances for the selected account is currently unavailable.
2220	Warn	Extended Account Balance Information Not Available	One or more extended account balances for the selected account is currently unavailable.
2300	Error	Single or Source Account Invalid	The Specified single account <xxxAcctId> or source account <xxxAccIdtFrom> identification aggregate contains invalid data or was not found.
2310	Error	Destination Account Invalid	The specified destination account identification aggregate <xxxAcctIdTo> contains invalid data or was not found.
2320	Error	Single Account or Source Account Not For Customer	The specified single account <xxxAcctId> or source account <xxxAcctIdFrom> does not correspond to the customer.
2340	Error	Destination Account Not For Customer	The specified destination account <xxxAcctIdTo> does not correspond to the customer.
2350	Error	Single or Source Account Closed	The specified single account <xxxAcctId> or source account <xxxAcctIdFrom> has been closed.
2360	Error	Destination Account Closed	The specified destination account <xxxAcctIdTo> has been closed.
2370	Error	Source And Destination Accounts Are Identical	A transfer that indicates that the source and destination accounts are the same is invalid.
2380	Error	Single or Source Account Not Authorized	The customer is not authorized to perform the requested action on the specified single or source account.
2390	Error	Destination Account Not Authorized	The customer is not authorized to perform the requested action on the specified destination account.
2400	Error	Single or Source Account Not Available	The single or source account requested is not available for transaction processing at this time. The customer may continue to access other accounts.
2410	Error	Destination Account Not Available	The destination account requested is not available for transaction processing at this time. The customer may continue to access other accounts.

2420	Error	Single or Source Account Not Eligible For Transaction	This single or source account is not authorized for this particular transaction.
2430	Error	Destination Account Not Eligible For Transaction	This destination account is not authorized for this particular transaction.
2440	Error	Invalid Single or Source Account Type	The single or source account type specified is not permitted by the service provider.
2450	Error	Invalid Destination Account Type	The service provider does not permit the destination account type specified.
2460	Error	Single or Source Account <BankId> Error	The Bank Identifier <BankId> in the source or single account identification aggregate is incorrect.
2470	Error	Destination Account <BankId> Error	The Bank Identifier <BankId> in the destination account is incorrect.
2480	Error	Branch ID Missing	A branch ID value must be provided in the single or source bank account information aggregate, but is missing.
2490	Error	Invalid <BankInfo>	The information contained within <BankInfo> is either invalid or insufficient to unambiguously identify the single or source bank.
2500	Error	Invalid Industry Id	The specified industry number is not valid for the Organization identifier.
2510	Error	Invalid Card Magnetic Strip Data	The card magnetic strip data provided is either incorrect or incomplete.
2520	Error	Unrecognized / Invalid Card Issuer	The card track data has been analyzed and found to be valid. The primary account number (PAN) has been extracted and the Issuer ID obtained from the PAN. The switch/server does not know how to route the transaction since it does not recognize the issuer ID (e.g. trying to use a Bank of Scotland debit card in the US). The most likely result would be for the ATM to return the card to the user after displaying a message such as "This ATM does not accept cards from your financial institution".
2530	Error	Unrecognized Card Number	The card track data has been analyzed and found to be valid. The PAN has been extracted from the card, the issuer is known and the transaction routed to the issuer. However the issuer cannot find this PAN on its database. This card is from an institution known to the ATM owner, but the card number is invalid. The most likely result would be for the ATM to capture this card (especially if the card was issued by the ATM owner).
2540	Error	Stolen Card	The card has been registered as stolen, and is therefore invalid as either an authentication mechanism for transactions or an identifier for one or more accounts.
2600	Error	Invalid Check Number Range	The specified check number range is invalid.
2610	Error	Request Is Too Late - Check has been paid	The customer has requested a Stop Payment on a check that has already cleared the bank and settled.
2620	Error	Check Number Not Found	The specified check number could not be found.
2640	Warn	Stop Check In Process	Stop Check is already in process.
2660	Error	Too Many Checks To Process	The check range specified in <StopChkAddRq> is too large for the service provider to handle.
2680	Error	Check Book Style Not Available	The specified checkbook style is not currently available. Please choose another style
2700	Error	Invalid Number Of Checks	The number of checks specified is not valid for the specified checkbook style.

2720	Error	Foreign Exchange not supported	The currency code on both accounts involved in a transfer must be the same as the currency code of the input amount, unless ForEx is specified in <OptSupt>.
2740	Error	Invalid Currency Code	The specified currency code is not valid in ISO 4217.
2760	Error	Invalid Language Code	The specified language code is not valid in RFC-1766.
2780	Error	Unsupported Frequency	The specified frequency is not supported for this message.
2900	Error	Duplicate Payment/Transfer Exists	The customer has attempted to schedule a payment/transfer that matches a previously entered payment/transfer (I.e., the day, amount and merchant or to account are the same). The client may resubmit using <DupChkOverride> = *True*, if the payment or transfer is not a duplicate.
2910	Error	Duplicate Recurring Model	The customer has attempted to schedule a recurring model that matches a previously entered model. The client may resubmit using <DupChkOverride> = *True*, if the model is not a duplicate.
2920	Error	Invalid skip count	The number of transfers or payments specified to skip exceeds the number of transfers or payments remaining for this recurring model.
2940	Error	Insufficient Funds	The service provider cannot process the transaction because the specified account does not have sufficient funds.
3000	Error	Number of Transfers Exceeds Limit	The number of transfers allowed for a monthly or statement cycle has been exceeded.
3020	Error	Daily Transfer Limit Exceeded	Total transfers for today exceed maximum allowable for one day.
3040	Error	Transfer Payment Greater Than Loan Balance	A transfer to a loan account (a payment) is declined because the loan payoff is less than the transfer amount.
3050	Warn	Authorized Amount Changed	The amount authorized for the transaction is different from that originally requested
3060	Error	Usage Limit Exceeded	The number of transactions of this type exceeds a predefined limit over a given time period.
3070	Warn	Changed Fee	The Fees required are different from those send in the request
3080	Error	Withdrawal Limit Exceeded	The amount the customer wishes to withdraw exceeds the withdrawal limit
3090	Error	No Fee Override Authorization	Client is not authorized to override the fee.
3100	Error	Invalid <BillerId>	The biller identifier specified could not be found or is invalid.
3120	Error	Invalid <StdPayeeId>	The standard payee identifier could not be found or was invalid.
3140	Error	Invalid Customer Payee ID	The specified Customer Payee ID <CustPayeeId> does not exist or no longer exists.
3180	Error	Invalid Payee Name	The name specified for the payee was incorrect or incomplete.
3200	Error	Payee List Full	The payee list is currently full. To add a payee, first delete an existing payee.
3240	Error	Invalid Customer Account With Payee	The pay account <PayAcct> specified by the user is invalid.
3260	Error	Payee Type Not Supported	This service provider does not support the payee type requested.

3300	Error	Number Of Payments Invalid	The number of payments entered is invalid. Please re-enter the number of payments.
3320	Error	Total Payment Amounts Scheduled For Today Exceeded Daily Limit	The sum of all payment amounts scheduled for today exceeds the daily limit (either at the SP or customer level).
3380	Error	Expired Card	The card specified has expired.
3400	Warn	Recurring Model Open-Ended	No final date or number of instances has been specified for the recurring model; it has been added as open-ended.
3500	Warn	Data Changed or Truncated	The server is responding with a warning that the <xxxRec> in the response is not identical to the data provided in the request. Rather, the data has been changed en route to storage.
3520	Error	Currencies don't match	The currencies of the amounts (e.g. in <SelRangeCurAmt>) are not the same.
3560	Error	Card Account ID Matches Multiple Accounts	The account cannot be uniquely identified based on the information provided by the Card Account Identifier.
3580	Error	Authorization is rejected	The Pay Provider is indicating that the payment has been rejected by the Customer Payment Provider
3600	Info	Authorization held over	The Pay provider is indicating that a payment has held over (typically for next day payment) by the CPP
3610	Error	Authorization is declined for insufficient funds.	The financial institution has declined the payment authorization because the specified account does not have sufficient funds.
3620	Error	Authorization is declined for inactive account.	The financial institution has declined the payment authorization because the specified account is inactive.
3630	Error	Authorization is declined for closed account.	The financial institution has declined the payment authorization because the specified account is closed.
3640	Error	Authorization is declined for other reason.	The financial institution has declined the payment authorization for another reason. Customer should contact the FI for specific reason.
3650	Error	Cash deposit limit exceeded.	The amount the customer wishes to deposit exceeds the cash deposit limit.
3700	Error	Object Type Not Supported	Client does not support indicated Security Object Type
3710	Error	Purpose Not Supported	Client does not support indicated purpose of Security Object
3720	Error	Format Not Supported	Client does not support indicated security Object Format
3730	Error	Key Storage Space not available	Storage space not available for key
3740	Error	Invalid Certificate	Certificate failed validation
3750	Error	Revoked Certificate	Certificate on Certificate Authorities revocation
3760	Error	Customer not related to account	Invalid Customer to Account Relationship combination.
3770	Warn	New Tax Responsible Party not Certified, Action Successful	Non-Certified Warn Indicator is set for Product, therefore, "New Tax Responsible Party not Certified, Action Successful" warning message is returned.
3780	Error	Delete would result in an invalid customer to account relationship	Delete would result in an invalid customer to account relationship
3800	Error	Message Authentication Error	The message authentication code (MAC) value is not correct.
3810	Error	<SPRefId> not recognized	The <SPRefID> is not recognized so cannot be used to process the request (<ForExDealAddRq>) or interpret the response (<ForExDealAddRs>).

3820	Error	<ForExDealId> not recognized	The <ForExDealId> provided for action (Inquire, Modify, Cancel, Audit, etc.) is not recognized.
3830	Error	<CurAmt> does not equal prior <CurAmt>	The <CurAmt> reported in the <ForExDealAddRq> does not match the <CurAmt> of the prior rate inquiry response (<ForExRateInqRs>).
3840	Error	<CurCode> does not equal prior <CurCode>	The <CurCode> reported in the deal request (<ForExDealAddRq>) does not match the CurCode of the prior rate inquiry (<ForExRateInqRs>).
3850	Error	Combination of CurAmt and CurCode do not match the <SPRefId>	The combination of <CurAmt> and <CurCode> reported in the deal request (<ForExDealAddRq>) does not match the combination of <CurAmt> and <CurCode> of the prior rate inquiry response (<ForExRateInqRs>).
3860	Error	Deal expired	The offer for the deal add requested has expired.
3870	Warn	No <ForExRateDealType> provided.	The <ForExRateDealType> has not been provided.
3880	Information	Remote Key Load Pending	Device is currently in a state that does not allow it to initiate the Remote Key Load process. Remote Key Load Process will be initiated at a future point in time.
3890	Error	Remote Key Load not supported	Device does not support Remote Key Load

A.1   All Messages

The following <StatusCode>s may be provided in any response and are not repeated in the following message tables:

<StatusCode>	<Desc>
0	Success
100	General Error
200	General Data Error
300	System Not Available
400	Function Not Available
500	Unsupported Service
600	Unsupported Message
700	Unsupported Function
900	Message Accepted for Asynchronous Processing
910	Asynchronous Request Does Not Match Original Request
1000	Duplicate <MsgUID>
1360	<SPName> Invalid
1910	Must Change Password
3800	Message Authentication Error

In addition the following <StatusCode>s may be received in *most* responses and are not included in the message tables:

<StatusCode>	<Desc>
1010	The <Cursor> submitted within <RecCtrlIn> is invalid or has expired
1020	Required Element Not Included

<StatusCode>	<Desc>
1110	Records Control Active
1380	<SPName> within <CustId> is invalid
1420	<CustId> invalid
1790	OBO is not active for this <CustId>
1800	Service Not Enabled

A.2 Security Messages

A.2.1 Signon (Signon)

<StatusCode>	<Desc>
1580	Invalid Country Code
1700	Security Violation
1740	Authentication Failed
1820	Customer Session Already In Progress
1880	Customer Locked Out
1900	Password Change Required
1980	Unsupported Application ID
2510	Invalid Card Magnetic Strip Data
2520	Unrecognized / Invalid Card Issuer
2530	Unrecognized Card Number
2540	Stolen Card
2760	Invalid Language Code
3380	Expired Card

A.2.2 Signoff (Signoff)

<StatusCode>	<Desc>
1840	No Customer Session In Progress

A.3 Base Service

A.3.1 Service Profile Inquiry (SvcProfInq)

<StatusCode>	<Desc>
1	Client Up To Date

A.3.2 Holiday Inquiry (HolInq)

<StatusCode>	<Desc>
1620	No <SPName>
1640	No <SvcName>

A.3.3   Customer Password Modify (CustPswdMod)

<StatusCode>	<Desc>
1700	Security Violation
1720	Customer Login ID and Password Will Be Sent Out-Of-Band
1740	Authentication Failed
1760	Authorization Failure
1970	Invalid New Password

A.3.4   Customer Add (CustAdd)

<StatusCode>	<Desc>
1280	Object already exists.
1500	Invalid Address
1520	Invalid City
1540	Invalid State Or Province
1560	Invalid Postal Code
1580	Invalid Country Code
1600	<CryptType> not valid
1720	Customer Login ID and/or Password will be Sent Out-Of-Band
1760	Authorization Failure
1920	Invalid New Customer Identifier
1940	New Customer Identifier in use
1970	Invalid New Password

A.3.5   Customer Modify (CustMod)

<StatusCode>	<Desc>
1060	Cannot Modify Element
1080	No Changes Made
1260	Unknown Object ID
1320	Object exists, but does not match <CustId>
1500	Invalid Address
1520	Invalid City
1540	Invalid State Or Province
1560	Invalid Postal Code
1580	Invalid Country Code

A.3.6   Customer Status Modify (CustStatusMod)

<StatusCode>	<Desc>
1080	No Changes Made
1260	Unknown Object ID

<StatusCode>	<Desc>
1320	Object exists, but does not match <CustId>

A.3.7   Customer Inquiry (CustInq)

<StatusCode>	<Desc>
1	Client Up To Date

A.3.8   Customer Synchronization (CustSync)

<StatusCode>	<Desc>
1	Client Up to Date
1100	Too Many Records Requested
1180	Token Value Prior to Available History
1200	Invalid Token

A.3.9   Customer/Service Link Add (CustSvcAdd)

<StatusCode>	<Desc>
1280	Object Already Exists
2300	Single or Source Account Invalid
2320	Single Account or Source Account Not For Customer
2350	Single or Source Account Closed
2380	Single or Source Account Not Authorized
2400	Single or Source Account Not Available
2420	Single or Source Account Not Eligible For Transaction
2460	Single or Source Account <BankId> Error
2480	Branch ID Missing
3380	Expired Card

A.3.10   Customer/Service Link Modify (CustSvcMod)

<StatusCode>	<Desc>
1050	Invalid Enum Value
1060	Cannot Modify Element
1080	No Changes Made
1260	Unknown Object ID
1320	Object exists, but does not match <CustId>
1980	Invalid Application ID
2300	Single or Source Account Invalid
2320	Single Account or Source Account Not For Customer
2350	Single or Source Account Closed
2380	Single or Source Account Not Authorized

<StatusCode>	<Desc>
2400	Single or Source Account Not Available
2420	Single or Source Account Not Eligible For Transaction
2440	Invalid Single or Source Account Type
2460	Single or Source Account <BankId> Error
2480	Branch ID Missing

A.3.11   Customer/Service Link Status Modify (CustSvcStatusMod)

<StatusCode>	<Desc>
1060	Cannot Modify Element
1080	No Changes Made

A.3.12   Customer/Service Link Delete (CustSvcDel)

<StatusCode>	<Desc>
1050	Invalid Enum Value
1300	Object Not deleted; dependent objects exist
1310	Cascade Delete Failed

A.3.13   Customer/Service Link Audit (CustSvcAud)

<StatusCode>	<Desc>
1050	Invalid Enum Value
1100	Too Many Records Requested
1360	<SPName> Invalid
2120	Invalid Datetime Range
2140	Datetime Too Soon

A.3.14   Customer/Service Link Synchronization (CustSvcSync)

<StatusCode>	<Desc>
1	Client Up To Date
1100	Too Many Records Requested
1180	Token Value Prior to Available History
1200	Invalid Token

A.3.15   Service/Account Link Add (SvcAcctAdd)

<StatusCode>	<Desc>
1600	<CryptType> not valid
2300	Single or Source Account Invalid

<StatusCode>	<Desc>
2320	Single Account or Source Account Not For Customer
2350	Single or Source Account Closed
2380	Single or Source Account Not Authorized
2400	Single or Source Account Not Available
2420	Single or Source Account Not Eligible For Transaction
2460	Single or Source Account <Bankid> Error
2480	Branch ID Missing

A.3.16   Service/Account Link Modify (SvcAcctMod)

<StatusCode>	<Desc>
1060	Cannot Modify Element
1600	<CryptType> not valid
2300	Single or Source Account Invalid
2320	Single Account or Source Account Not For Customer
2350	Single or Source Account Closed
2380	Single or Source Account Not Authorized
2400	Single or Source Account Not Available
2420	Single or Source Account Not Eligible For Transaction
2460	Single or Source Account <BankId> Error
2480	Branch ID Missing

A.3.17   Service/Account Link Status Modify (SvcAcctStatusMod)

<StatusCode>	<Desc>
1080	No Changes Made
1260	Unknown Object ID
1320	Object exists, but does not match <CustId>

A.3.18   Service/Account Link Identifier Modify (SvcAcctIdMod)

<StatusCode>	<Desc>
1080	No Changes Made
1260	Unknown Object ID
1320	Object exists, but does not match <CustId>

A.3.19   Service/Account Link Delete (SvcAcctDel)

<StatusCode>	<Desc>
1300	Object not deleted; dependent objects exist

<StatusCode>	<Desc>
1310	Cascade Delete Failed
2300	Single or Source Account Invalid
2320	Single Account or Source Account Not For Customer
2460	Single or Source Account <BankId> Error
2480	Branch ID Missing

A.3.20   Service/Account Link Inquiry (SvcAcctInq)

<StatusCode>	<Desc>
1050	Invalid Enum Value
1120	No Records Match Selection Criteria
1140	Some Selection Criteria Not Supported
1160	None of Selection Criteria Supported
2100	Invalid Datetime – Single date or Low of Range
2150	Datetime Too Far in Future

A.3.21   Service/Account Link Audit (SvcAcctAud)

<StatusCode>	<Desc>
1050	Invalid Enum Value
1100	Too Many Records Requested
1120	No Records Match Selection Criteria
1140	Some Selection Criteria Not Supported
1160	None of Selection Criteria Supported
2100	Invalid Datetime – Single date or Low of Range
2110	Invalid Datetime - High of Range
2120	Invalid Datetime Range
2150	Datetime Too Far in Future

A.3.22   Service/Account Link Synchronization (SvcAcctSync)

<StatusCode>	<Desc>
1	Client Up To Date
1100	Too Many Records Requested
1180	Token Value Prior to Available History
1200	Invalid Token

A.3.23   Service/Account Link Reversal (SvcAcctRev)

<StatusCode>	<Desc>
810	Message can not be reversed.

A.3.24   Disclosure Inquiry (DiscInq)

<StatusCode>	<Desc>
1050	Invalid Enum Value
1120	No Records Match Selection Criteria
1140	Some Selection Criteria Not Supported
1160	None of Selection Criteria Supported
1260	Unknown Object ID
1320	Object exists, but does not match <CustId>

A.3.25   Customer/Disclosure Link Status Modify (CustDiscStatusMod)

<StatusCode>	<Desc>
1050	Invalid Enum Value
1260	Unknown Object ID
1320	Object exists, but does not match <CustId>

A.3.26   Customer/Disclosure Link Inquiry (CustDiscInq)

<StatusCode>	<Desc>
1050	Invalid Enum Value
1120	No Records Match Selection Criteria
1140	Some Selection Criteria Not Supported
1160	None of Selection Criteria Supported
1260	Unknown Object ID
1320	Object exists, but does not match <CustId>

A.3.27   Customer Identifier Inquiry (CustIdInq)

<StatusCode>	<Desc>
1120	No Data Matches Selection Criteria

A.3.28   Customer Status Modify (CustStatusMod)

<StatusCode>	<Desc>
1720	Customer Login ID And Password Will Be Sent Out-Of-Band
1910	Must Change Password

A.3.29   Security Object Add (SecObjAdd)

<StatusCode>	<Desc>
1260	Unknown Object ID
3700	Object Type Not Supported

3710	Purpose Not Supported
3720	Format Not Supported
3730	Key Storage Space not available
3740	Invalid Certificate
3750	Revoked Certificate

A.3.30   Security Object Modify (SecObjMod)

<StatusCode>	*<Desc>*
1260	Unknown Object ID
3740	Invalid Certificate
3750	Revoked Certificate

A.3.31   Security Object Inquiry (SecObjInq)

<StatusCode>	*<Desc>*
1120	No Records Match Selection Criteria
1260	Unknown Object ID

A.3.32   Security Object Delete (SecObjDel)

<StatusCode>	*<Desc>*
1260	Unknown Object ID

A.3.33   Security Object Advise (SecObjAdvise)

<StatusCode>	*<Desc>*
3760	Remote Key Load Pending
3770	Remote Key Load not supported
3880	Remote Key Load Pending
3890	Remote Key Load not supported

A.3.34   Security Object Sync (SecObjSync)

<StatusCode>	*<Desc>*
1	Client Up To Date
200	General Data Error
1100	Too Many Records Requested
1180	Token Value Prior to Available History
1200	Invalid Token

A.3.35   Party Account Relationship Add (PartyAcctRelAdd)

<StatusCode>	*<Desc>*

200	General Data Error
300	System Not Available
1020	Required Element Not Included
2300	Single or Source Account Invalid
1280	Object already exists.
3700	Invalid Relationship Type for Product
3760	Customer not related to account
3770	New Tax Responsible Party not Certified, Action Successful

A.3.36   Party Account Relationship Modify (PartyAcctRelMod)

<StatusCode>	<Desc>
700	Unsupported Function
1020	Required Element Not Included
2300	Single or Source Account Invalid
3780	New Tax Responsible Party not Certified, Action Successful

A.3.37   Party Account Relationship Delete (PartyAcctRelDel)

<StatusCode>	<Desc>
1020	Required Element Not Included
2300	Single or Source Account Invalid
3780	Delete would result in an invalid customer to account relationship

A.3.38   Party Account Relationship Inquiry (PartyAcctRelInq)

<StatusCode>	<Desc>
1020	Required Element Not Included
1120	No record match selection criteria

A.4   Banking Service

A.4.1   Balance Inquiry (BalInq)

<StatusCode>	<Desc>
1520	Invalid City
1540	Invalid State or Province
1560	Invalid Postal Code
2210	Account Balance Information Not Available
2220	Extended Account Balance Information Not Available
2300	Single or Source Account Invalid
2320	Single Account or Source Account Not For Customer
2350	Single or Source Account Closed

<StatusCode>	<Desc>
2380	Single or Source Account Not Authorized
2400	Single or Source Account Not Available
2420	Single or Source Account Not Eligible For Transaction
2440	Invalid Single or Source Account Type
2460	Single or Source Account <BankId> error
2480	Branch ID Missing
2490	Bank Information Missing or Invalid
2510	Invalid Card Magnetic Strip Data
2520	Unrecognized / Invalid Card Issuer
2530	Unrecognized Card Number
2540	Stolen Card
3380	Expired Card
3560	Card Account ID Matches Multiple Accounts

A.4.2 Balance Reversal (BalRev)

<StatusCode>	<Desc>
810	Message can not be reversed.

A.4.3 Account Inquiry (AcctInq)

<StatusCode>	<Desc>
1520	Invalid City
1540	Invalid State or Province
1560	Invalid Postal Code
2300	Single or Source Account Invalid
2320	Single Account or Source Account Not For Customer
2350	Single or Source Account Closed
2380	Single or Source Account Not Authorized
2400	Single or Source Account Not Available
2420	Single or Source Account Not Eligible For Transaction
2440	Invalid Single or Source Account Type
2460	Single or Source Account <BankId> error
2480	Branch ID Missing
2490	Bank Information Missing or Invalid
2510	Invalid Card Magnetic Strip Data
2520	Unrecognized / Invalid Card Issuer
2530	Unrecognized Card Number
2540	Stolen Card
3380	Expired Card
3560	Card Account ID Matches Multiple Accounts

A.4.4   Account Reversal (AcctRev)

<StatusCode>	<Desc>
810	Message can not be reversed.

A.4.5   Deposit Account Statement Inquiry (DepAcctStmtInq)

<StatusCode>	<Desc>
1100	Too Many Records Requested
1120	No Data Matches Se lection Criteria
1160	None of Selection Criteria Supported
1240	Detail Not Available
1520	Invalid City
1540	Invalid State or Province
1560	Invalid Postal Code
2100	Invalid Datetime - Single date or Low of Range
2110	Invalid Datetime - High of Range
2120	Invalid Datetime Range
2200	History Not Available for Full Date Range
2300	Single or Source Account Invalid
2320	Single Account or Source Account Not For Customer
2350	Single or Source Account Closed
2380	Single or Source Account Not Authorized
2400	Single or Source Account Not Available
2420	Single or Source Account Not Eligible For Transaction
2440	Invalid Single or Source Account Type
2460	Single or Source Account <BankId> Error
2480	Branch ID Missing
2490	Bank Information Missing or Invalid
2510	Invalid Card Magnetic Strip Data
2520	Unrecognized / Invalid Card Issuer
2530	Unrecognized Card Number
2540	Stolen Card
3380	Expired Card
3560	Card Account ID Matches Multiple Accounts

A.4.6   Credit Card Statement Inquiry (CCAcctStmtInq)

<StatusCode>	<Desc>
1100	Too Many Records Requested
1120	No Data Matches Selection Criteria

<StatusCode>	<Desc>
1160	None of Selection Criteria Supported
1240	Detail Not Available
1520	Invalid City
1540	Invalid State or Province
1560	Invalid Postal Code
2100	Invalid Datetime - Single date or Low of Range
2110	Invalid Datetime - High of Range
2120	Invalid Datetime Range
2200	History Not Available for Full Date Range
2300	Single or Source Account Invalid
2320	Single Account or Source Account Not For Customer
2350	Single or Source Account Closed
2380	Single or Source Account Not Authorized
2400	Single or Source Account Not Available
2420	Single or Source Account Not Eligible For Transaction
2440	Invalid Single or Source Account Type
2460	Single or Source Account <BankId> Error
2480	Branch ID Missing
2520	Unrecognized / Invalid Card Issuer
2530	Unrecognized Card Number
2540	Stolen Card
3380	Expired Card

A.4.7 Credit Card Statement Reversal (CCAcctStmtRev)

<StatusCode>	<Desc>
810	Message can not be reversed.

A.4.8 Deposit Account Transaction Inquiry (DepAcctTrnInq)

<StatusCode>	<Desc>
1050	Invalid Enum Value
1120	No Data Matches Selection Criteria
1140	Some Selection Criteria Not Supported
1160	None of Selection Criteria Supported
1260	Unknown Object ID
1320	Object exists, but does not match <CustId>.
2000	Invalid Data in Single or Low Amount Field
2010	Invalid Data in High Amount Field
2020	Amount Too Small
2030	Amount Too Large

<StatusCode>	<Desc>
2060	Invalid High/Low Amount
2100	Invalid Datetime - Single date or Low of Range
2110	Invalid Datetime - High of Range
2120	Invalid Datetime Range
2200	History Not Available for Full Date Range
2300	Single or Source Account Invalid
2320	Single Account or Source Account Not For Customer
2350	Single or Source Account Closed
2380	Single or Source Account Not Authorized
2400	Single or Source Account Not Available
2420	Single or Source Account Not Eligible For Transaction
2440	Invalid Single or Source Account Type
2460	Single or Source Account <BankId> Error
2480	Branch ID Missing
2520	Unrecognized / Invalid Card Issuer
2530	Unrecognized Card Number
2540	Stolen Card
3380	Expired Card

A.4.9 Deposit Account Transaction Reversal (DepAcctTrnRev)

<StatusCode>	<Desc>
810	Message can not be reversed.

A.4.10 Credit Card Account Transaction Inquiry (CCAcctTrnInq)

<StatusCode>	<Desc>
1050	Invalid Enum Value
1120	No Data Matches Selection Criteria
1140	Some Selection Criteria Not Supported
1160	None of Selection Criteria Supported
1260	Unknown Object ID
1320	Object exists, but does not match <CustId>.
2000	Invalid Data in Single or Low Amount Field
2010	Invalid Data in High Amount Field
2020	Amount Too Small
2030	Amount Too Large
2060	Invalid High/Low Amount
2100	Invalid Datetime - Single date or Low of Range

Interactive Financial Exchange Business Message Specification

<StatusCode>	<Desc>
2110	Invalid Datetime - High of Range
2120	Invalid Datetime Range
2200	History Not Available for Full Date Range
2300	Single or Source Account Invalid
2320	Single Account or Source Account Not For Customer
2350	Single or Source Account Closed
2380	Single or Source Account Not Authorized
2400	Single or Source Account Not Available
2420	Single or Source Account Not Eligible For Transaction
2440	Invalid Single or Source Account Type
2460	Single or Source Account <BankId> Error
2480	Branch ID Missing
2520	Unrecognized / Invalid Card Issuer
2530	Unrecognized Card Number
2540	Stolen Card
3380	Expired Card

A.4.11   Credit Card Account Transaction Reversal (CCAcctTrnRev)

<StatusCode>	<Desc>
810	Message can not be reversed.

A.4.12   Interest Rate Inquiry (IntRateInq)

<StatusCode>	<Desc>
1050	Invalid Enum Value
1100	Too Many Records Requested
1120	No Data Matches Selection Criteria
1140	Some Selection Criteria Not Supported
1160	None of Selection Criteria Supported
2000	Invalid Data in Single or Low Amount Field
2120	Invalid Datetime
2130	Date Passed
2440	Invalid Single or Source Account Type

A.4.13   Interest Rate Rev (IntRateRev)

<StatusCode>	<Desc>
810	Message can not be reversed.

A.4.14   Bank Account Taxation Inquiry (BankAcctTaxInq)

\<StatusCode\>	*\<Desc\>*
1050	Invalid Enum Value
1120	No Data Matches Selection Criteria
1140	Some Selection Criteria Not Supported
1160	None of Selection Criteria Supported
1520	Invalid City
1540	Invalid State or Province
1560	Invalid Postal Code
2100	Invalid Datetime - Single date or Low of Range
2300	Single or Source Account Invalid
2320	Single Account or Source Account Not For Customer
2400	Single or Source Account Not Available
2460	Single or Source Account \<BankId\> Error
2480	Branch ID Missing
2490	Bank Information Missing or Invalid
2510	Invalid Card Magnetic Strip Data
2520	Unrecognized / Invalid Card Issuer
2530	Unrecognized Card Number
2540	Stolen Card
3380	Expired Card

A.4.15   Foreign Exchange Rate Inquiry (ForExRateInq)

\<StatusCode\>	*\<Desc\>*
1050	Invalid Enum Value
1120	No Data Matches Selection Criteria
2020	Amount Too Small
2030	Amount Too Large
2080	Final Amount Exceeds Limit
2300	Single or Source Account Invalid
2320	Single Account or Source Account Not For Customer
2350	Single or Source Account Closed
2380	Single or Source Account Not Authorized
2400	Single or Source Account Not Available
2420	Single or Source Account Not Eligible For Transaction
2460	Single or Source Account \<BankId\> Error
2480	Branch ID Missing
2510	Invalid Card Magnetic Strip Data
2520	Unrecognized / Invalid Card Issuer
2530	Unrecognized Card Number

<StatusCode>	<Desc>
2540	Stolen Card
2740	Invalid Currency Code
3380	Expired Card
3560	Card Account ID Matches Multiple Accounts

A.4.16　Foreign Exchange Rate Reversal (ForExRateRev)

<StatusCode>	<Desc>
810	Message can not be reversed.

A.4.17　ForExDealAdd

<StatusCode>	<Desc>
3810	<SPRefId> not recognized
3830	<CurAmt> does not equal <CurAmt> in prior <ForExRateInqRs>
3840	<CurCode> does not equal <CurCode> in prior< ForExRateInqRs>
3850	Combination of <CurAmt> and <CurCode> do not match the <SPRefId> from prior <ForExRateInqRs>
3860	Deal expired
3870	No <ForExRateDealType> provided.

A.4.18　ForExDealMod

<StatusCode>	<Desc>
3820	<ForExDealId> not recognized.

A.4.19　ForExDealInq

<StatusCode>	<Desc>
3820	<ForExDealId> not recognized.

A.4.20　ForExDealCan

<StatusCode>	<Desc>
3820	<ForExDealId> not recognized.

A.4.21　ForExDealAud

<StatusCode>	<Desc>
3820	<ForExDealId> not recognized.

A.4.22   ForExDealSync

<StatusCode>	<Desc>
3820	<ForExDealId> not recognized.

A.4.23   ForExDealRev

<StatusCode>	<Desc>
810	Message cannot be reversed.
3820	<ForExDealId> not recognized.

A.4.24   ForExDealStatusMod

<StatusCode>	<Desc>
3820	<ForExDealId> not recognized.

A.4.25   ForExDealStatusInq

<StatusCode>	<Desc>
3820	<ForExDealId> not recognized.

A.4.26   ForExDealAdvise

<StatusCode>	<Desc>
3820	<ForExDealId> not recognized.

A.4.27   Stop Check Add (StopChkAdd)

<StatusCode>	<Desc>
800	Object Already Committed
1520	Invalid City
1540	Invalid State or Province
1560	Invalid Postal Code
2300	Single or Source Account Invalid
2320	Single Account or Source Account Not For Customer
2350	Single or Source Account Closed
2380	Single or Source Account Not Authorized
2400	Single or Source Account Not Available
2420	Single or Source Account Not Eligible For Transaction
2460	Single or Source Account <BankId> Error
2480	Branch ID Missing
2490	Bank Information Missing or Invalid
2510	Invalid Card Magnetic Strip Data
2520	Unrecognized / Invalid Card Issuer
2530	Unrecognized Card Number

<StatusCode>	<Desc>
2540	Stolen Card
2600	Invalid Check Number Range
2610	Request Is Too Late - Check has been paid
2620	Check Number Not Found
2640	Stop Check In Process
3380	Expired Card

A.4.28   Stop Check Cancel (StopChkCan)

<StatusCode>	<Desc>
800	ObjectAlready Committed
1520	Invalid City
1540	Invalid State or Province
1560	Invalid Postal Code
2300	Single or Source Account Invalid
2320	Single Account or Source Account Not For Customer
2350	Single or Source Account Closed
2380	Single or Source Account Not Authorized
2400	Single or Source Account Not Available
2420	Single or Source Account Not Eligible For Transaction
2460	Single or Source Account <BankId> Error
2480	Branch ID Missing
2490	Bank Information Missing or Invalid
2510	Invalid Card Magnetic Strip Data
2520	Unrecognized / Invalid Card Issuer
2530	Unrecognized Card Number
2540	Stolen Card
2600	Invalid Check Number Range
2620	Check Number Not Found
3380	Expired Card

A.4.29   Stop Check Inquiry (StopChkInq)

<StatusCode>	<Desc>
1120	No Data Matches Selection Criteria
1140	Some Selection Criteria Not Supported
1160	None of Selection Criteria Supported
1520	Invalid City
1540	Invalid State or Province
1560	Invalid Postal Code
2300	Single or Source Account Invalid

\<StatusCode\>	\<Desc\>
2320	Single Account or Source Account Not For Customer
2350	Single or Source Account Closed
2380	Single or Source Account Not Authorized
2400	Single or Source Account Not Available
2420	Single or Source Account Not Eligible For Transaction
2460	Single or Source Account \<BankId\> Error
2480	Branch ID Missing
2490	Bank Information Missing or Invalid
2510	Invalid Card Magnetic Strip Data
2520	Unrecognized / Invalid Card Issuer
2530	Unrecognized Card Number
2540	Stolen Card
2600	Invalid Check Number Range
2620	Check Number Not Found
3380	Expired Card

A.4.30   Stop Check Audit (StopChkAud)

\<StatusCode\>	\<Desc\>
1	Client Up to Date
1120	No Data Matches Selection Criteria
1140	Some Selection Criteria Not Supported
1160	None of Selection Criteria Supported
1180	Token Value Prior to Available History
1520	Invalid City
1540	Invalid State or Province
1560	Invalid Postal Code
2120	Invalid Datetime Range
2300	Single or Source Account Invalid
2320	Single Account or Source Account Not For Customer
2350	Single or Source Account Closed
2380	Single or Source Account Not Authorized
2400	Single or Source Account Not Available
2420	Single or Source Account Not Eligible For Transaction
2460	Single or Source Account \<BankId\> Error
2480	Branch ID Missing
2490	Bank Information Missing or Invalid
2510	Invalid Card Magnetic Strip Data
2520	Unrecognized / Invalid Card Issuer
2530	Unrecognized Card Number

<StatusCode>	<Desc>
2540	Stolen Card
2600	Invalid Check Number Range
2620	Check Number Not Found
3380	Expired Card

A.4.31   Stop Check Synchronization (StopChkSync)

<StatusCode>	<Desc>
1	Client Up to Date
1100	Too Many Records Requested
1180	Token Value Prior to Available History
1200	Invalid Token
2300	Single or Source Account Invalid
2320	Single Account or Source Account Not For Customer
2350	Single or Source Account Closed
2380	Single or Source Account Not Authorized
2400	Single or Source Account Not Available
2420	Single or Source Account Not Eligible For Transaction
2460	Single or Source Account <BankId> Error
2480	Branch ID Missing
2490	Bank Information Missing or Invalid
2510	Invalid Card Magnetic Strip Data
2520	Unrecognized / Invalid Card Issuer
2530	Unrecognized Card Number
2540	Stolen Card
3380	Expired Card

A.4.32   Stop Check Reversal (StopChkRev)

<StatusCode>	<Desc>
810	Message can not be reversed.

A.4.33   Funds Transfer Add (XferAdd)

<StatusCode>	<Desc>
1040	Request Declined
2000	Invalid Data in Single or Low Amount Field
2020	Amount Too Small
2030	Amount Too Large
2130	Date Passed
2140	Datetime Too Soon

<StatusCode>	<Desc>
2150	Datetime Too Far In Future
2180	Request Is Too Late For Today's Work
2190	Weekend Or Holiday
2300	Single or Source Account Invalid
2310	Destination Account Invalid
2320	Single Account or Source Account Not For Customer
2340	Destination Account Not For Customer
2350	Single or Source Account Closed
2360	Destination Account Closed
2370	Source And Destination Accounts Are Identical
2380	Single or Source Account Not Authorized
2390	Destination Account Not Authorized
2400	Single or Source Account Not Available
2410	Destination Account Not Available
2420	Single or Source Account Not Eligible For Transaction
2430	Destination Account Not Eligible For Transaction
2440	Invalid Single or Source Account Type
2450	Invalid Destination Account Type
2460	Single or Source Account <BankId> Error
2470	Destination Account <BankId> Error
2480	Branch ID Missing
2740	Invalid Currency Code
2900	Duplicate Payment/Transfer Exists
2940	Insufficient Funds
3000	Number of Transfers Exceeds Limit
3020	Daily Transfer Limit Exceeded
3040	Transfer Payment Greater Than Loan Balance
3560	Card Account ID Matches Multiple Accounts

A.4.34 Funds Transfer Modify (XferMod)

<StatusCode>	<Desc>
1040	Request Declined
1060	Cannot Modify Element
1080	No Changes Made
1260	Unknown Object ID
1320	Object exists, but does not match <CustId>.
2000	Invalid Data in Single or Low Amount Field
2020	Amount Too Small
2030	Amount Too Large

<StatusCode>	<Desc>
2130	Date Passed
2140	Datetime Too Soon
2150	Datetime Too Far In Future
2180	Request Is Too Late For Today's Work
2190	Weekend Or Holiday
2300	Single or Source Account Invalid
2310	Destination Account Invalid
2320	Single Account or Source Account Not For Customer
2340	Destination Account Not For Customer
2350	Single or Source Account Closed
2360	Destination Account Closed
2370	Source And Destination Accounts Are Identical
2380	Single or Source Account Not Authorized
2390	Destination Account Not Authorized
2400	Single or Source Account Not Available
2410	Destination Account Not Available
2420	Single or Source Account Not Eligible For Transaction
2430	Destination Account Not Eligible For Transaction
2440	Invalid Single or Source Account Type
2450	Invalid Destination Account Type
2460	Single or Source Account <BankId> Error
2470	Destination Account <BankId> Error
2740	Invalid Currency Code
2900	Duplicate Payment/Transfer Exists
2940	Insufficient Funds
3000	Number of Transfers Exceeds Limit
3020	Daily Transfer Limit Exceeded
3040	Transfer Payment Greater Than Loan Balance
3560	Card Account ID Matches Multiple Accounts

A.4.35   Funds Transfer Status Modify (XferStatusMod)

<StatusCode>	<Desc>
1040	Request Declined
1060	Cannot Modify Element
1080	No Changes Made
1260	Unknown Object ID
1320	Object exists, but does not match <CustId>.
2100	Invalid Datetime in Single or Low of Range
2300	Single or Source Account Invalid

<StatusCode>	<Desc>
2320	Single Account or Source Account Not For Customer
2350	Single or Source Account Closed
2380	Single or Source Account Not Authorized
2400	Single or Source Account Not Available
2420	Single or Source Account Not Eligible For Transaction
2460	Single or Source Account <BankId> Error
2490	Bank Information Missing or Invalid

A.4.36   Funds Transfer Cancel (XferCan)

<StatusCode>	<Desc>
800	Object Already Committed
1260	Unknown Object ID
1320	Object exists, but does not match <CustId>.
2130	Date Passed
2180	Request Is Too Late For Today's Work

A.4.37   Funds Transfer Inquiry (XferInq)

<StatusCode>	<Desc>
1	Client Up To Date
1050	Invalid Enum Value
1100	Too Many Records Requested
1120	No Data Matches Selection Criteria
1140	Some Selection Criteria Not Supported
1160	None of Selection Criteria Supported
1180	Token Value Prior to Available History
1200	Invalid Token
1260	Unknown Object ID
1320	Object exists, but does not match <CustId>
2000	Invalid Data in Single or Low Amount Field
2010	Invalid Data in High Amount Field
2060	Invalid High/Low Amount
2100	Invalid Datetime - Single date or Low of Range
2120	Invalid Datetime Range
2150	Datetime Too Far In Future
2300	Single or Source Account Invalid
2310	Destination Account Invalid
2440	Invalid Single or Source Account Type
2470	Destination Account <BankId>Error
2510	Invalid Card Magnetic Strip Data

<StatusCode>	<Desc>
2520	Unrecognized / Invalid Card Issuer
2530	Unrecognized Card Number
2540	Stolen Card
3380	Expired Card

A.4.38   Funds Transfer Audit (XferAud)

<StatusCode>	<Desc>
1	Client Up To Date
1050	Invalid Enum Value
1100	Too Many Records Requested
1120	No Data Matches Selection Criteria
1140	Some Selection Criteria Not Supported
1160	None of Selection Criteria Supported
1180	Token Value Prior to Available History
1200	Invalid Token
1260	Unknown Object ID
1320	Object exists, but does not match <CustId>
2000	Invalid Data in Single or Low Amount Field
2010	Invalid Data in High Amount Field
2060	Invalid High/Low Amount
2100	Invalid Datetime - Single date or Low of Range
2120	Invalid Datetime Range
2150	Datetime Too Far In Future
2300	Single or Source Account Invalid
2310	Destination Account Invalid
2440	Invalid Single or Source Account Type

A.4.39   Funds Transfer Synchronization (XferSync)

<StatusCode>	<Desc>
1	Client Up to Date
1100	Too Many Records Requested
1180	Token Value Prior to Available History
1200	Invalid Token
2510	Invalid Card Magnetic Strip Data
2520	Unrecognized / Invalid Card Issuer
2530	Unrecognized Card Number
2540	Stolen Card
3380	Expired Card
3560	Card Account ID Matches Multiple Accounts

A.4.40 Funds Transfer Reversal (XferRev)

<StatusCode>	<Desc>
810	Message can not be reversed.

A.4.41 Recurring Funds Transfer Model Add (RecXferAdd)

<StatusCode>	<Desc>
1040	Request Declined
2000	Invalid Data in Single or Low Amount Field
2020	Amount Too Small
2030	Amount Too Large
2080	Final Amount Exceeds Limit
2100	Invalid Datetime - Single date or Low of Range
2110	Invalid Datetime - High of Range
2120	Invalid Datetime Range
2130	Date Passed
2140	Datetime Too Soon
2150	Datetime Too Far In Future
2180	Request Is Too Late For Today's Work
2190	Weekend Or Holiday
2300	Single or Source Account Invalid
2310	Destination Account Invalid
2320	Single Account or Source Account Not For Customer
2340	Destination Account Not For Customer
2350	Single or Source Account Closed
2360	Destination Account Closed
2370	Source And Destination Accounts Are Identical
2380	Single or Source Account Not Authorized
2390	Destination Account Not Authorized
2400	Single or Source Account Not Available
2410	Destination Account Not Available
2420	Single or Source Account Not Eligible For Transaction
2430	Destination Account Not Eligible For Transaction
2440	Invalid Single or Source Account Type
2450	Invalid Destination Account Type
2460	Single or Source Account <BankId> Error
2470	Destination Account <BankId> Error
2480	Branch ID Missing
2720	Foreign Exchange not supported.
2740	Invalid Currency Code

<StatusCode>	<Desc>
2780	Unsupported Frequency
2910	Duplicate Recurring Model
2920	Invalid skip count
2940	Insufficient Funds
3000	Number of Transfers Exceeds Limit
3020	Daily Transfer Limit Exceeded
3040	Transfer Payment Greater Than Loan Balance
3400	Recurring Model Open-Ended
3560	Card Account ID Matches Multiple Accounts

A.4.42 Recurring Funds Transfer Instance Add (RecXferInstAdd)

<StatusCode>	<Desc>
1040	Request Declined
1060	Cannot Modify Element
1080	No Changes Made
1260	Unknown Object ID
1320	Object exists, but does not match <CustId>.
2000	Invalid Data in Single or Low Amount Field
2020	Amount Too Small
2030	Amount Too Large
2080	Final Amount Exceeds Limit
2100	Invalid Datetime - Single date or Low of Range
2110	Invalid Datetime - High of Range
2120	Invalid Datetime Range
2130	Date Passed
2140	Datetime Too Soon
2150	Datetime Too Far In Future
2180	Request Is Too Late For Today's Work
2300	Single or Source Account Invalid
2310	Destination Account Invalid
2320	Single Account or Source Account Not For Customer
2340	Destination Account Not For Customer
2350	Single or Source Account Closed
2360	Destination Account Closed
2370	Source And Destination Accounts Are Identical
2380	Single or Source Account Not Authorized
2390	Destination Account Not Authorized
2400	Single or Source Account Not Available
2410	Destination Account Not Available

<StatusCode>	<Desc>
2420	Single or Source Account Not Eligible For Transaction
2430	Destination Account Not Eligible For Transaction
2440	Invalid Single or Source Account Type
2450	Invalid Destination Account Type
2460	Single or Source Account <BankId> Error
2470	Destination Account <BankId> Error
2720	Foreign Exchange not supported.
2740	Invalid Currency Code
2780	Unsupported Frequency
2910	Duplicate Recurring Model
2920	Invalid skip count
2940	Insufficient Funds
3000	Number of Transfers Exceeds Regulation 'D' Limit
3020	Daily Transfer Limit Exceeded
3040	Transfer Payment Greater Than Loan Balance
3560	Card Account ID Matches Multiple Accounts

A.4.43   Recurring Funds Transfer Model Modify (RecXferMod)

<StatusCode>	<Desc>
1040	Request Declined
1060	Cannot Modify Element
1080	No Changes Made
1260	Unknown Object ID
1320	Object exists, but does not match <CustId>.
2000	Invalid Data in Single or Low Amount Field
2020	Amount Too Small
2030	Amount Too Large
2080	Final Amount Exceeds Limit
2100	Invalid Datetime - Single date or Low of Range
2110	Invalid Datetime - High of Range
2120	Invalid Datetime Range
2130	Date Passed
2140	Datetime Too Soon
2150	Datetime Too Far In Future
2180	Request Is Too Late For Today's Work
2300	Single or Source Account Invalid
2310	Destination Account Invalid
2320	Single Account or Source Account Not For Customer
2340	Destination Account Not For Customer

<StatusCode>	<Desc>
2350	Single or Source Account Closed
2360	Destination Account Closed
2370	Source And Destination Accounts Are Identical
2380	Single or Source Account Not Authorized
2390	Destination Account Not Authorized
2400	Single or Source Account Not Available
2410	Destination Account Not Available
2420	Single or Source Account Not Eligible For Transaction
2430	Destination Account Not Eligible For Transaction
2440	Invalid Single or Source Account Type
2450	Invalid Destination Account Type
2460	Single or Source Account <BankId> Error
2470	Destination Account <BankId> Error
2720	Foreign Exchange not supported.
2740	Invalid Currency Code
2780	Unsupported Frequency
2910	Duplicate Recurring Model
2920	Invalid skip count
2940	Insufficient Funds
3000	Number of Transfers Exceeds Regulation 'D' Limit
3020	Daily Transfer Limit Exceeded
3040	Transfer Payment Greater Than Loan Balance
3560	Card Account ID Matches Multiple Accounts

A.4.44　Recurring Funds Transfer Model Cancel (RecXferCan)

<StatusCode>	<Desc>
800	Object Already Committed
1040	Request Declined
1260	Unknown Object ID
1320	Object exists, but does not match <CustId>.
2130	Date Passed
2180	Request Is Too Late For Today's Work

A.4.45　Recurring Funds Transfer Model Inquiry (RecXferInq)

<StatusCode>	<Desc>
1050	Invalid Enum Value
1100	Too Many Records Requested
1120	No Data Matches Selection Criteria
1140	Some Selection Criteria Not Supported

<StatusCode>	<Desc>
1160	None of Selection Criteria Supported
1180	Token Value Prior to Available History
1200	Invalid Token
1260	Unknown Object ID
1320	Object exists, but does not match <CustId>
2000	Invalid Data in Single or Low Amount Field
2010	Invalid Data in High Amount Field
2060	Invalid High/Low Amount
2100	Invalid Datetime - Single date or Low of Range
2120	Invalid Datetime Range
2150	Datetime Too Far In Future
2200	History Not Available for Full Date Range
2310	Destination Account Invalid
2440	Invalid Single or Source Account Type
2480	Branch ID Missing
2510	Invalid Card Magnetic Strip Data
2520	Unrecognized / Invalid Card Issuer
2530	Unrecognized Card Number
2540	Stolen Card
3380	Expired Card

A.4.46   Recurring Funds Transfer Model Audit (RecXferAud)

<StatusCode>	<Desc>
1	Client Up To Date
1050	Invalid Enum Value
1100	Too Many Records Requested
1120	No Data Matches Selection Criteria
1140	Some Selection Criteria Not Supported
1160	None of Selection Criteria Supported
1180	Token Value Prior to Available History
1200	Invalid Token
1260	Unknown Object ID
1320	Object exists, but does not match <CustId>
2000	Invalid Data in Single or Low Amount Field
2010	Invalid Data in High Amount Field
2060	Invalid High/Low Amount
2100	Invalid Datetime - Single date or Low of Range
2120	Invalid Datetime Range
2150	Datetime Too Far In Future

<StatusCode>	<Desc>
2200	History Not Available for Full Date Range
2300	Single or Source Account Invalid
2310	Destination Account Invalid
2440	Invalid Single or Source Account Type
2480	Branch ID Missing

A.4.47   Recurring Funds Transfer Synchronization (RecXferSync)

<StatusCode>	<Desc>
1	Client Up to Date
1100	Too Many Records Requested
1180	Token Value Prior to Available History
1200	Invalid Token
2510	Invalid Card Magnetic Strip Data
2520	Unrecognized / Invalid Card Issuer
2530	Unrecognized Card Number
2540	Stolen Card
3380	Expired Card
3560	Card Account ID Matches Multiple Accounts

A.4.48   Records Funds Transfer Model Reversal (RecXferRev)

<StatusCode>	<Desc>
810	Message can not be reversed.

A.4.49   Bank Account Transaction Image Inquiry (BankAcctTrnImgInq)

<StatusCode>	<Desc>
1120	No Data Matches Selection Criteria
1260	Unknown Object ID
1320	Object exists, but does not match <CustId>.
1500	Invalid Address
1520	Invalid City
1540	Invalid State Or Province
1560	Invalid Postal Code
1580	Invalid Country Code
2000	Invalid Data in Single or Low Amount Field
2100	Invalid Datetime - Single date or Low of Range
2300	Single or Source Account Invalid

<StatusCode>	<Desc>
2350	Single or Source Account Closed
2380	Single or Source Account Not Authorized
2400	Single or Source Account Not Available
2420	Single or Source Account Not Eligible For Transaction
2460	Single or Source Account <BankId> Error
2480	Branch ID Missing
2490	Bank Information Missing or Invalid
2510	Invalid Card Magnetic Strip Data
2520	Unrecognized / Invalid Card Issuer
2530	Unrecognized Card Number
2540	Stolen Card
2620	Check Number Not Found
3380	Expired Card

A.4.50   Bank Account Transaction Image Inquiry Reversal (BankAcctTrnImgRev)

<StatusCode>	<Desc>
810	Message can not be reversed.

A.4.51   Check Order (ChkOrdAdd)

<StatusCode>	<Desc>
1500	Invalid Address
1520	Invalid City
1540	Invalid State Or Province
1560	Invalid Postal Code
1580	Invalid Country Code
2300	Single or Source Account Invalid
2320	Single Account or Source Account Not For Customer
2350	Single or Source Account Closed
2400	Single or Source Account Not Available
2460	Single or Source Account <BankId> Error
2480	Branch ID Missing
2490	Bank Information Missing or Invalid
2510	Invalid Card Magnetic Strip Data
2600	Invalid Check Number Range
2680	Check Book Style Not Available

A.4.52   Check Order Reversal (ChkOrdRev)

<StatusCode>	<Desc>
810	Message can not be reversed.

A.4.53   Deposit Book Order (DepBookOrdAdd)

<StatusCode>	<Desc>
1050	Invalid Enum Value
1500	Invalid Address
1520	Invalid City
1540	Invalid State Or Province
1560	Invalid Postal Code
1580	Invalid Country Code
2300	Single or Source Account Invalid
2320	Single Account or Source Account Not For Customer
2350	Single or Source Account Closed
2380	Single or Source Account Not Authorized
2400	Single or Source Account Not Available
2420	Single or Source Account Not Eligible For Transaction
2460	Single or Source Account <BankId> Error
2480	Branch ID Missing
2490	Bank Information Missing or Invalid

A.4.54   Deposit Book Order Reversal (DepBkOrdRev)

<StatusCode>	<Desc>
810	Message can not be reversed.

A.4.55   Debit Authorization Add (DebitAuthAdd)

<StatusCode>	<Desc>
1050	Invalid Enum Value
2020	Amount Too Small
2030	Amount Too Large
2420	Single or Source Account Not Eligible For Transaction
2510	Invalid Card Magnetic Strip Data
2740	Invalid Currency Code
2940	Insufficient Funds
3050	Authorized Amount Changed
3060	Usage Limit Exceeded
3070	Changed Fee
3080	Withdrawal Limit Exceeded

<StatusCode>	<Desc>
3090	No Fee Override Authorization
3380	Expired Card
3560	Card Account ID Matches Multiple Accounts

A.4.56   Debit Authorization Modify (DebitAuthMod)

<StatusCode>	<Desc>
800	Object Already Committed
1050	Invalid Enum Value
1220	Invalid Identifier
2020	Amount Too Small
2030	Amount Too Large
2510	Invalid Card Magnetic Strip Data
2740	Invalid Currency Code
2940	Insufficient Funds
3050	Authorized Amount Changed
3070	Changed Fee
3080	Withdrawal Limit Exceeded
3090	No Fee Override Authorization
3560	Card Account ID Matches Multiple Accounts

A.4.57   Debit Authorization Cancel (DebitAuthCan)

<StatusCode>	<Desc>
800	Object Already Committed
1050	Invalid Enum Value
1220	Invalid Identifier
2510	Invalid Card Magnetic Strip Data

A.4.58   Debit Authorization Inquiry (DebitAuthInq)

<StatusCode>	<Desc>
1050	Invalid Enum Value
1100	Too Many Records Requested
1120	No Data Matches Selection Criteria
1140	Some Selection Criteria Not Supported
1160	None of Selection Criteria Supported
1180	Token Value Prior to Available History
1220	Invalid Identifier
1260	Unknown Object ID
1320	Object exists, but does not match <CustId>

<StatusCode>	<Desc>
2000	Invalid Data in Single or Low Amount Field
2010	Invalid Data in High Amount Field
2060	Invalid High/Low Amount
2100	Invalid Datetime - Single date or Low of Range
2120	Invalid Datetime Range
2150	Datetime Too Far In Future
2200	History Not Available for Full Date Range
2510	Invalid Card Magnetic Strip Data
2520	Unrecognized / Invalid Card Issuer
2530	Unrecognized Card Number
2540	Stolen Card
3380	Expired Card
3560	Card Account ID Matches Multiple Accounts

A.4.59 Debit Authorization Audit (DebitAuthAud)

<StatusCode>	<Desc>
1	Client Up To Date
1050	Invalid Enum Value
1100	Too Many Records Requested
1120	No Data Matches Selection Criteria
1140	Some Selection Criteria Not Supported
1160	None of Selection Criteria Supported
1220	Invalid Identifier
1260	Unknown Object ID
1320	Object exists, but does not match <CustId>
2100	Invalid Datetime - Single date or Low of Range
2120	Invalid Datetime Range
2150	Datetime Too Far In Future
2200	History Not Available for Full Date Range

A.4.60 Debit Authorization Sync (DebitAuthSync)

<StatusCode>	<Desc>
1	Client Up to Date
1100	Too Many Records Requested
1180	Token Value Prior to Available History
1200	Invalid Token
2510	Invalid Card Magnetic Strip Data
2520	Unrecognized / Invalid Card Issuer
2530	Unrecognized Card Number

<StatusCode>	<Desc>
2540	Stolen Card
3380	Expired Card
3560	Card Account ID Matches Multiple Accounts

A.4.61 Debit Authorization Status Modify (DebitAuthStatusMod)

<StatusCode>	<Desc>
1040	Request Declined
1060	Cannot Modify Element
1080	No Changes Made
1260	Unknown Object ID
1320	Object exists, but does not match <CustId>.
2100	Invalid Datetime in Single or Low of Range
2300	Single or Source Account Invalid
2320	Single Account or Source Account Not For Customer
2350	Single or Source Account Closed
2380	Single or Source Account Not Authorized
2400	Single or Source Account Not Available
2420	Single or Source Account Not Eligible For Transaction
2460	Single or Source Account <BankId> Error
2490	Bank Information Missing or Invalid

A.4.62 Debit Authorization Reversal (DebitAuthRev)

<StatusCode>	<Desc>
810	Message can not be reversed.

A.4.63 Debit Add (DebitAdd)

<StatusCode>	<Desc>
1050	Invalid Enum Value
1220	Invalid Identifier
2020	Amount Too Small
2030	Amount Too Large
2420	Single or Source Account Not Eligible For Transaction
2510	Invalid Card Magnetic Strip Data
2740	Invalid Currency Code
2940	Insufficient Funds
3060	Usage Limit Exceeded
3090	No Fee Override Authorization

A.4.64   Debit Modify (DebitMod)

<StatusCode>	<Desc>
800	Object Already Committed
1050	Invalid Enum Value
1220	Invalid Identifier
2020	Amount Too Small
2030	Amount Too Large
2510	Invalid Card Magnetic Strip Data
2740	Invalid Currency Code
2940	Insufficient Funds
3050	Authorized Amount Changed
3070	Changed Fee
3080	Withdrawal Limit Exceeded
3090	No Fee Override Authorization
3560	Card Account ID Matches Multiple Accounts

A.4.65   Debit Cancel (DebitCan)

<StatusCode>	<Desc>
800	Object Already Committed
1050	Invalid Enum Value
1220	Invalid Identifier
2510	Invalid Card Magnetic Strip Data

A.4.66   Debit Inquiry (DebitInq)

<StatusCode>	<Desc>
1050	Invalid Enum Value
1100	Too Many Records Requested
1120	No Data Matches Selection Criteria
1140	Some Selection Criteria Not Supported
1160	None of Selection Criteria Supported
1180	Token Value Prior to Available History
1220	Invalid Identifier
1260	Unknown Object ID
1320	Object exists, but does not match <CustId>
2000	Invalid Data in Single or Low Amount Field
2010	Invalid Data in High Amount Field
2060	Invalid High/Low Amount
2100	Invalid Datetime - Single date or Low of Range
2120	Invalid Datetime Range
2150	Datetime Too Far In Future

<StatusCode>	<Desc>
2200	History Not Available for Full Date Range
2510	Invalid Card Magnetic Strip Data
2520	Unrecognized / Invalid Card Issuer
2530	Unrecognized Card Number
2540	Stolen Card
3380	Expired Card
3560	Card Account ID Matches Multiple Accounts

A.4.67 Debit Audit (DebitAud)

<StatusCode>	<Desc>
1	Client Up To Date
1050	Invalid Enum Value
1100	Too Many Records Requested
1120	No Data Matches Selection Criteria
1140	Some Selection Criteria Not Supported
1160	None of Selection Criteria Supported
1220	Invalid Identifier
1260	Unknown Object ID
1320	Object exists, but does not match <CustId>
2100	Invalid Datetime - Single date or Low of Range
2120	Invalid Datetime Range
2150	Datetime Too Far In Future
2200	History Not Available for Full Date Range

A.4.68 Debit Sync (DebitSync)

<StatusCode>	<Desc>
1	Client Up to Date
1100	Too Many Records Requested
1180	Token Value Prior to Available History
1200	Invalid Token
2510	Invalid Card Magnetic Strip Data
2520	Unrecognized / Invalid Card Issuer
2530	Unrecognized Card Number
2540	Stolen Card
3380	Expired Card
3560	Card Account ID Matches Multiple Accounts

A.4.69 Debit Status Modify (DebitStatusMod)

<StatusCode>	<Desc>
1040	Request Declined
1060	Cannot Modify Element
1080	No Changes Made
1260	Unknown Object ID
1320	Object exists, but does not match <CustId>.
2100	Invalid Datetime in Single or Low of Range
2300	Single or Source Account Invalid
2320	Single Account or Source Account Not For Customer
2350	Single or Source Account Closed
2380	Single or Source Account Not Authorized
2400	Single or Source Account Not Available
2420	Single or Source Account Not Eligible For Transaction
2460	Single or Source Account <BankId> Error
2490	Bank Information Missing or Invalid

A.4.70 Debit Reversal (DebitRev)

<StatusCode>	<Desc>
810	Message can not be reversed.

A.4.71 Credit Authorization Add (CreditAuthAdd)

<StatusCode>	<Desc>
1050	Invalid Enum Value
2020	Amount Too Small
2030	Amount Too Large
2420	Single or Source Account Not Eligible For Transaction
2510	Invalid Card Magnetic Strip Data
2740	Invalid Currency Code
2940	Insufficient Funds
3050	Authorized Amount Changed
3060	Usage Limit Exceeded
3070	Changed Fee Advice
3080	Withdrawal Limit Exceeded
3090	No Fee Override Authorization
3380	Expired Card
3560	Card Account ID Matches Multiple Accounts

A.4.72   Credit Authorization Modify (CreditAuthMod)

<StatusCode>	<Desc>
800	Object Already Committed
1050	Invalid Enum Value
1220	Invalid Identifier
2020	Amount Too Small
2030	Amount Too Large
2510	Invalid Card Magnetic Strip Data
2740	Invalid Currency Code
2940	Insufficient Funds
3050	Authorized Amount Changed
3070	Changed Fee Advice
3080	Withdrawal Limit Exceeded
3090	No Fee Override Authorization
3560	Card Account ID Matches Multiple Accounts

A.4.73   Credit Authorization Cancel (CreditAuthCan)

<StatusCode>	<Desc>
800	Object Already Committed
1050	Invalid Enum Value
1220	Invalid Identifier
2510	Invalid Card Magnetic Strip Data

A.4.74   Credit Authorization Inquiry (CreditAuthInq)

<StatusCode>	<Desc>
1050	Invalid Enum Value
1100	Too Many Records Requested
1120	No Data Matches Selection Criteria
1140	Some Selection Criteria Not Supported
1160	None of Selection Criteria Supported
1180	Token Value Prior to Available History
1220	Invalid Identifier
1260	Unknown Object ID
1320	Object exists, but does not match <CustId>
2000	Invalid Data in Single or Low Amount Field
2010	Invalid Data in High Amount Field
2060	Invalid High/Low Amount
2100	Invalid Datetime - Single date or Low of Range
2120	Invalid Datetime Range
2150	Datetime Too Far In Future

<StatusCode>	<Desc>
2200	History Not Available for Full Date Range
2510	Invalid Card Magnetic Strip Data
2520	Unrecognized / Invalid Card Issuer
2530	Unrecognized Card Number
2540	Stolen Card
3380	Expired Card
3560	Card Account ID Matches Multiple Accounts

A.4.75   Credit Authorization Audit (CreditAuthAud)

<StatusCode>	<Desc>
1	Client Up To Date
1050	Invalid Enum Value
1100	Too Many Records Requested
1120	No Data Matches Selection Criteria
1140	Some Selection Criteria Not Supported
1160	None of Selection Criteria Supported
1220	Invalid Identifier
1260	Unknown Object ID
1320	Object exists, but does not match <CustId>
2100	Invalid Datetime - Single date or Low of Range
2120	Invalid Datetime Range
2150	Datetime Too Far In Future
2200	History Not Available for Full Date Range

A.4.76   Credit Authorization Sync (CreditAuthSync)

<StatusCode>	<Desc>
1	Client Up to Date
1100	Too Many Records Requested
1180	Token Value Prior to Available History
1200	Invalid Token
2510	Invalid Card Magnetic Strip Data
2520	Unrecognized / Invalid Card Issuer
2530	Unrecognized Card Number
2540	Stolen Card
3380	Expired Card
3560	Card Account ID Matches Multiple Accounts

A.4.77 Credit Authorization Status Modify (CreditAuthStatusMod)

<StatusCode>	<Desc>
1040	Request Declined
1060	Cannot Modify Element
1080	No Changes Made
1260	Unknown Object ID
1320	Object exists, but does not match <CustId>.
2100	Invalid Datetime in Single or Low of Range
2300	Single or Source Account Invalid
2320	Single Account or Source Account Not For Customer
2350	Single or Source Account Closed
2380	Single or Source Account Not Authorized
2400	Single or Source Account Not Available
2420	Single or Source Account Not Eligible For Transaction
2460	Single or Source Account <BankId> Error
2490	Bank Information Missing or Invalid

A.4.78 Credit Authorization Reversal (CreditAuthRev)

<StatusCode>	<Desc>
810	Message can not be reversed.

A.4.79 Credit Add (CreditAdd)

<StatusCode>	<Desc>
1050	Invalid Enum Value
1220	Invalid Identifier
2020	Amount Too Small
2030	Amount Too Large
2420	Single or Source Account Not Eligible For Transaction
2510	Invalid Card Magnetic Strip Data
2740	Invalid Currency Code
2940	Insufficient Funds
3060	Usage Limit Exceeded
3090	No Fee Override Authorization
3650	Cash deposit limit exceeded.

A.4.80 Credit Modify (CreditMod)

<StatusCode>	<Desc>
800	Object Already Committed

<StatusCode>	<Desc>
1050	Invalid Enum Value
1220	Invalid Identifier
2020	Amount Too Small
2030	Amount Too Large
2510	Invalid Card Magnetic Strip Data
2740	Invalid Currency Code
2940	Insufficient Funds
3050	Authorized Amount Changed
3070	Changed Fee Advice
3080	Withdrawal Limit Exceeded
3090	No Fee Override Authorization
3560	Card Account ID Matches Multiple Accounts

A.4.81   Credit Cancel (CreditCan)

<StatusCode>	<Desc>
800	Object Already Committed
1050	Invalid Enum Value
1220	Invalid Identifier
2510	Invalid Card Magnetic Strip Data

A.4.82   Credit Inquiry (CreditInq)

<StatusCode>	<Desc>
1050	Invalid Enum Value
1100	Too Many Records Requested
1120	No Data Matches Selection Criteria
1140	Some Selection Criteria Not Supported
1160	None of Selection Criteria Supported
1180	Token Value Prior to Available History
1220	Invalid Identifier
1260	Unknown Object ID
1320	Object exists, but does not match <CustId>
2000	Invalid Data in Single or Low Amount Field
2010	Invalid Data in High Amount Field
2060	Invalid High/Low Amount
2100	Invalid Datetime - Single date or Low of Range
2120	Invalid Datetime Range
2150	Datetime Too Far In Future
2200	History Not Available for Full Date Range
2510	Invalid Card Magnetic Strip Data

<StatusCode>	<Desc>
2520	Unrecognized / Invalid Card Issuer
2530	Unrecognized Card Number
2540	Stolen Card
3380	Expired Card
3560	Card Account ID Matches Multiple Accounts

A.4.83   Credit Audit (CreditAud)

<StatusCode>	<Desc>
1	Client Up To Date
1050	Invalid Enum Value
1100	Too Many Records Requested
1120	No Data Matches Selection Criteria
1140	Some Selection Criteria Not Supported
1160	None of Selection Criteria Supported
1220	Invalid Identifier
1260	Unknown Object ID
1320	Object exists, but does not match <CustId>
2100	Invalid Datetime - Single date or Low of Range
2120	Invalid Datetime Range
2150	Datetime Too Far In Future
2200	History Not Available for Full Date Range

A.4.84   Credit Sync (CreditSync)

<StatusCode>	<Desc>
1	Client Up to Date
1100	Too Many Records Requested
1180	Token Value Prior to Available History
1200	Invalid Token
2510	Invalid Card Magnetic Strip Data
2520	Unrecognized / Invalid Card Issuer
2530	Unrecognized Card Number
2540	Stolen Card
3380	Expired Card
3560	Card Account ID Matches Multiple Accounts

A.4.85   Credit Status Modify (CreditStatusMod)

<StatusCode>	<Desc>
1040	Request Declined

<StatusCode>	<Desc>
1060	Cannot Modify Element
1080	No Changes Made
1260	Unknown Object ID
1320	Object exists, but does not match <CustId>.
2100	Invalid Datetime in Single or Low of Range
2300	Single or Source Account Invalid
2320	Single Account or Source Account Not For Customer
2350	Single or Source Account Closed
2380	Single or Source Account Not Authorized
2400	Single or Source Account Not Available
2420	Single or Source Account Not Eligible For Transaction
2460	Single or Source Account <BankId> Error
2490	Bank Information Missing or Invalid

A.4.86   Credit Reversal (CreditRev)

<StatusCode>	<Desc>
810	Message can not be reversed.

A.4.87   Bank Account Statement Image Inquiry (BankAcctStmtImgInq)

<StatusCode>	<Desc>
1100	Too Many Records Requested
1110	Records Control Active
1120	No Data Matches Selection Criteria
1140	Some Selection Criteria Not Supported
1160	None of Selection Criteria Supported
1240	Detail Not Available
1500	Invalid Address
1520	Invalid City
1540	Invalid State or Province
1560	Invalid Postal Code
1580	Invalid Country Code
2100	Invalid Datetime - Single date or Low of Range
2110	Invalid Datetime - High of Range
2120	Invalid Datetime Range
2200	History Not Available for Full Date Range
2300	Single or Source Account Invalid
2320	Single Account or Source Account Not For Customer
2350	Single or Source Account Closed

<StatusCode>	<Desc>
2380	Single or Source Account Not Authorized
2400	Single or Source Account Not Available
2420	Single or Source Account Not Eligible For Transaction
2440	Invalid Single or Source Account Type
2460	Single or Source Account <BankId> Error
2480	Branch ID Missing
2490	Bank Information Missing or Invalid
2510	Invalid Card Magnetic Strip Data
2520	Unrecognized / Invalid Card Issuer
2530	Unrecognized Card Number
2540	Stolen Card
3380	Expired Card
3560	Card Account ID Matches Multiple Accounts

A.4.88   Bank Account Statement Image Reversal (BankAcctStmtImgRev)

<StatusCode>	<Desc>
810	Message can not be reversed.

A.4.89   Payment Enclosed Reversal (PmtEnclRev)

<StatusCode>	<Desc>
810	Message can not be reversed.

A.4.90   Check Issue Add (ChkIssueAdd)

<StatusCode>	<Desc>
1050	Invalid Enum Value
1280	Object already exists
1480	Edit Mask Error
1760	Authorization Failure
2300	Single or Source Account Invalid
2440	Invalid Single or Source Account Type
2900	Duplicate Payment/Transfer Exists
3580	Authorization is rejected
3600	Authorization held over

A.4.91   Check Issue Modify (ChkIssueMod)

<StatusCode>	<Desc>
800	Object Object Already Committed

<StatusCode>	<Desc>
1050	Invalid Enum Value
1060	Cannot Modify Element
1080	No changes made
1260	Unknown Object ID
1320	Object exists, but does not match <CustId>
1760	Authorization Failure
2300	Single or Source Account Invalid
2440	Invalid Single or Source Account Type
3580	Authorization is rejected
3600	Authorization held over

A.4.92   Check Issue Status Modify (ChkIssueStatusMod)

<StatusCode>	<Desc>
1050	Invalid Enum Value
1060	Cannot Modify Element
1080	No changes made
1260	Unknown Object ID
1320	Object Exists But Does Not Match <CustId>
1760	Authorization Failure
3580	Authorization is rejected
3600	Authorization held over

A.4.93   Check Issue Delete (ChkIssueDel)

<StatusCode>	<Desc>
800	Object Object Already CCommitted
1260	Unknown Object ID
1320	Object exists, but does not match <CustId>
1760	Authorization Failure
3580	Authorization is rejected
3600	Authorization held over

A.4.94   Check Issue Inquiry (ChkIssueInq)

<StatusCode>	<Desc>
1050	Invalid Enum Value
1100	Too Many Records Requested
1120	Selection No Data Matches Selection Criteria
1140	Some Selection Criteria Not Supported
1160	None of Selection Criteria Supported

<StatusCode>	<Desc>
1260	Unknown Object ID
1320	Object exists, but does not match <CustId>
2000	Invalid Data in Single or Low Amount Field
2010	Invalid Data in High Amount Field
2060	Invalid High/Low Amount
2100	Invalid Datetime - Single date or Low of Range
2110	Invalid Datetime – High of Range
2120	Invalid Datetime Range
2150	Datetime Too Far In Future
2200	History Not Available for Full Date Range
2300	Single or Source Account Invalid
2440	Invalid Single or Source Account Type

A.4.95   Check Issue Status Inquiry (ChkIssueStatusInq)

<StatusCode>	<Desc>
1050	Invalid Enum Value
1100	Too Many Records Requested
1120	Selection No Data Matches Selection Criteria
1140	Some Selection Criteria Not Supported
1160	None of Selection Criteria Supported
1260	Unknown Object ID
1320	Object exists, but does not match <CustId>
2100	Invalid Datetime – Single date or Low of Range
2110	Invalid Datetime – High of Range
2120	Invalid Datetime Range
2150	Datetime Too Far In Future
2200	History Not Available for Full Date Range
2300	Single or Source Account Invalid
2440	Invalid Single or Source Account Type

A.4.96   Check Issue Audit (ChkIssueAud)

<StatusCode>	<Desc>
1	Client Up To Date
1050	Invalid Enum Value
1100	Too Many Records Requested
1120	Selection No Data Matches Selection Criteria
1140	Some Selection Criteria Not Supported
1160	None of Selection Criteria Supported
1260	Unknown Object ID

<StatusCode>	<Desc>
1320	Object exists, but does not match <CustId>
2100	Invalid Datetime - Single date or Low of Range
2110	Invalid Datetime – High of Range
2120	Invalid Datetime Range
2150	Datetime Too Far In Future
2200	History Not Available for Full Date Range
2300	Single or Source Account Invalid
2440	Invalid Single or Source Account Type

A.4.97 Check Issue Synchronization (ChkIssueSync)

<StatusCode>	<Desc>
1	Client Up to Date
1100	Too Many Records Requested
1180	Token Value Prior to Available History
1200	Invalid Token
2300	Single or Source Account Invalid
2440	Invalid Single or Source Account Type

A.4.98 Bank Service Checksum Add (BankSvcChkSumAdd)

<StatusCode>	<Desc>
1280	Object already exists
1480	Edit Mask Error
2020	Amount Too Small
2030	Amount Too Large
2130	Date Passed
2160	Processing Date Precedes Today
2180	Request is Too Late for Today's Work
2190	Weekend or Holiday
3140	Invalid Customer Payee ID <CustPayeeId>
3580	Authorization is rejected
3600	Authorization held over

A.4.99 Bank Service Checksum Modify (BankSvcChkSumMod)

<StatusCode>	<Desc>
800	Object Already Committed
1080	No changes made
1260	Unknown Object ID
1320	Object exists, but does not match <CustId>

<StatusCode>	<Desc>
2020	Amount Too Small
2030	Amount Too Large
2130	Date Passed
2160	Processing Date Precedes Today
2180	Request is Too Late for Today's Work
2190	Weekend or Holiday
3140	Invalid Customer Payee ID<CustPayeeId>
3580	Authorization is rejected
3600	Authorization held over

A.4.100 Bank Service Checksum Status Modify (BankSvcChkSumStatusMod)

<StatusCode>	<Desc>
1050	Invalid Enum Value
1060	Cannot Modify Element
1080	No changes made
1260	Unknown Object ID
1320	Object Exists But Does Not Match <CustId>
1760	Authorization Failure
3580	Authorization is rejected
3600	Authorization held over

A.4.101 Bank Service Checksum Delete (BankSvcChkSumDel)

<StatusCode>	<Desc>
800	Object Object Already Committed
1260	Unknown Object ID
1320	Object exists, but does not match <CustId>

A.4.102 Bank Service Checksum Inquiry (BankSvcChkSumInq)

<StatusCode>	<Desc>
1050	Invalid Enum Value
1100	Too Many Records Requested
1120	Selection No Data Matches Selection Criteria
1140	Some Selection Criteria Not Supported
1160	None of Selection Criteria Supported
1260	Unknown Object ID
1320	Object exists, but does not match <CustId>

A.4.103 Bank Service Checksum Audit (BankSvcChkSumAud)

\<StatusCode\>	\<Desc\>
1050	Invalid Enum Value
1100	Too Many Records Requested
1120	Selection No Data Matches Selection Criteria
1140	Some Selection Criteria Not Supported
1160	None of Selection Criteria Supported
1260	Unknown Object ID
1320	Object exists, but does not match \<CustId\>
2100	Invalid Datetime - Single date or Low of Range
2110	Invalid Datetime – High of Range
2120	Invalid Datetime Range
2150	Datetime Too Far In Future
2200	History Not Available for Full Date Range

A.4.104 Bank Service Checksum Synchronization (BankSvcChkSumSync)

\<StatusCode\>	\<Desc\>
1100	Too Many Records Requested
1180	Token Value Prior to Available History
1200	Invalid Token

A.5   Pay Service

A.5.1   Standard Payee Inquiry (StdPayeeInq)

\<StatusCode\>	\<Desc\>
1050	Invalid Enum Value
1100	Too Many Records Requested
1120	No Records Match Selection Criteria
1140	Some Selection Criteria Not Supported
1160	None of Selection Criteria Supported
3120	Invalid \<StdPayeeId\>
3260	Payee Type Not Supported

A.5.2   Standard Payee Reversal (StdPayeeRev)

\<StatusCode\>	\<Desc\>
810	Message can not be reversed.

A.5.3   Customer Payee Add (CustPayeeAdd)

<StatusCode>	<Desc>
1280	Object already exists
1480	Edit Mask Error
1500	Invalid Address
1520	Invalid City
1540	Invalid State or Province
1560	Invalid Postal Code
1580	Invalid Country Code
3120	Invalid <StdPayeeId>
3200	Payee List Full
3240	Invalid Customer Account with Payee
3260	Payee Type Not Supported

A.5.4   Customer Payee Modify (CustPayeeMod)

<StatusCode>	<Desc>
1080	No Changes Made
1060	Cannot Modify Element
1260	Unknown Object ID
1320	Object exists, but does not match <CustId>
1480	Edit Mask Error
1500	Invalid Address
1520	Invalid City
1500	Invalid Address
1520	Invalid City
1540	Invalid State or Province
1560	Invalid Postal Code
1580	Invalid Country Code
3140	Invalid Customer Payee ID <CustPayeeId>
3180	Invalid Payee Name
3240	Invalid Customer Account with Payee
3260	Payee Type Not Supported

A.5.5   Customer Payee Delete (CustPayeeDel)

<StatusCode>	<Desc>
1260	Unknown Object ID
1300	Object not deleted; dependent objects exist
1310	Cascade Delete Failed
1320	Object exists, but does not match <CustId>
3140	Invalid Customer PayeeID<CustPayeeId>

A.5.6 Customer Payee Inquiry (CustPayeeInq)

<StatusCode>	<Desc>
1050	Invalid Enum Value
1100	Too Many Records Requested
1120	No Records Match Selection Criteria
1140	Some Selection Criteria Not Supported
1160	None of Selection Criteria Supported
3140	Invalid Customer Payee ID<CustPayeeId>
3260	Payee Type Not Supported

A.5.7 Customer Payee Audit (CustPayeeAud)

<StatusCode>	<Desc>
1	Client Up to Date
1050	Invalid Enum Value
1100	Too Many Records Requested
1140	Some Selection Criteria Not Supported
1160	None of Selection Criteria Supported
2100	Invalid Datetime - Single date or Low of Range
2110	Invalid Datetime - High of Range
2120	Invalid Datetime Range
2150	Datetime Too Far in Future
2200	History Not Available for Full Date Range
3140	Invalid Customer Payee ID <CustPayeeId>
3260	Payee Type Not Supported

A.5.8 Customer Payee Synchronization (CustPayeeSync)

<StatusCode>	<Desc>
1	Client Up to Date
1100	Too Many Records Requested
1180	Token Value Prior to Available History
1200	Invalid Token

A.5.9 Payment Add (PmtAdd)

<StatusCode>	<Desc>
1280	Object already exists
1480	Edit Mask Error
1500	Invalid Address
1520	Invalid City

<StatusCode>	<Desc>
1540	Invalid State or Province
1560	Invalid Postal Code
1580	Invalid Country Code
2020	Amount Too Small
2030	Amount Too Large
2130	Date Passed
2140	Datetime Too Soon
2150	Datetime Too Far in Future
2160	Processing Date Precedes Today
2180	Request is Too Late for Today's Work
2190	Weekend or Holiday
2310	Destination Account Invalid (for Transfer Payee Payments)
2900	Duplicate Payment/Transfer Exists
3100	Invalid <BillerId>
3120	Invalid <StdPayeeId>
3140	Invalid Customer Payee ID <CustPayeeId>
3180	Invalid Payee Name
3200	Payee List Full
3240	Invalid Customer Account with Payee
3260	Payee Type Not Supported
3320	Total Payment Amounts Scheduled for Today exceeded daily limit
3380	Expired Card
3580	Authorization is rejected
3600	Authorization held over

A.5.10 Payment Modify (PmtMod)

<StatusCode>	<Desc>
800	Object Object Already Committed
1060	Cannot Modify Element (some bill pay systems don't allow mod of Payee Name)
1080	No changes made
1260	Unknown Object ID
1320	Object exists, but does not match <CustId>
1500	Invalid Address
1520	Invalid City
1540	Invalid State or Province
1560	Invalid Postal Code
1580	Invalid Country Code
2020	Amount Too Small
2030	Amount Too Large

<StatusCode>	<Desc>
2130	Date Passed
2140	Datetime Too Soon
2150	Datetime Too Far in Future
2160	Processing Date Precedes Today
2180	Request is Too Late for Today's Work
2190	Weekend or Holiday
2310	Destination Account Invalid (for Transfer Payee Payments)
3100	Invalid <BillerId><BillerId>
3120	Invalid <StdPayeeId><StdPayeeId>
3140	Invalid Customer Payee ID<CustPayeeId>
3180	Invalid Payee Name
3200	Payee List Full
3240	Invalid Customer Account with Payee
3260	Payee Type Not Supported
3320	Total Payment Amounts Scheduled for Today exceeded daily limit
3380	Expired Card
3580	Authorization is rejected
3600	Authorization held over

A.5.11   Payment Status Modify (PmtStatusMod)

<StatusCode>	<Desc>
1050	Invalid Enum Value
1060	Cannot Modify Element
1080	No changes made
1260	Unknown Object ID
1320	Object Exists But Does Not Match <CustId>
1760	Authorization Failure
2100	Invalid Datetime – Single Date or Low of Range
2150	Datetime Too Far in Future
3580	Authorization is rejected
3600	Authorization held over

A.5.12   Payment Cancel (PmtCan)

<StatusCode>	<Desc>
800	Object Object Already CCommitted
1260	Unknown Object ID
1320	Object exists, but does not match <CustId>

A.5.13 Payment Inquiry (PmtInq)

<StatusCode>	<Desc>
1050	Invalid Enum Value
1100	Too Many Records Requested
1120	Selection No Data Matches Selection Criteria
1140	Some Selection Criteria Not Supported
1160	None of Selection Criteria Supported
1260	Unknown Object ID
1320	Object exists, but does not match <CustId>
2000	Invalid Data in Single or Low Amount Field
2010	Invalid Data in High Amount Field
2060	Invalid High/Low Amount
2100	Invalid Datetime - Single date or Low of Range
2110	Invalid Datetime – High of Range
2120	Invalid Datetime Range
2150	Datetime Too Far In Future
2200	History Not Available for Full Date Range
2300	Single or Source Account Invalid
2440	Invalid Single or Source Account Type
2510	Invalid Card Magnetic Strip Data
2520	Unrecognized / Invalid Card Issuer
2530	Unrecognized Card Number
2540	Stolen Card
3140	Invalid Customer Payee ID <CustPayeeId>
3380	Expired Card

A.5.14 Payment Status Inquiry (PmtStatusInq)

<StatusCode>	<Desc>
1050	Invalid Enum Value
1100	Too Many Records Requested
1120	Selection No Data Matches Selection Criteria
1140	Some Selection Criteria Not Supported
1160	None of Selection Criteria Supported
1260	Unknown Object ID
1320	Object exists, but does not match <CustId>
2100	Invalid Datetime – Single date or Low of Range
2110	Invalid Datetime – High of Range
2120	Invalid Datetime Range
2150	Datetime Too Far In Future
2200	History Not Available for Full Date Range

<StatusCode>	<Desc>
2300	Single or Source Account Invalid
2440	Invalid Single or Source Account Type
2510	Invalid Card Magnetic Strip Data
2520	Unrecognized / Invalid Card Issuer
2530	Unrecognized Card Number
2540	Stolen Card
3140	Invalid Customer Payee ID <CustPayeeId>
3380	Expired Card

A.5.15   Payment Audit (PmtAud)

<StatusCode>	<Desc>
1	Client Up To Date
1050	Invalid Enum Value
1100	Too Many Records Requested
1120	Selection No Data Matches Selection Criteria
1140	Some Selection Criteria Not Supported
1160	None of Selection Criteria Supported
1260	Unknown Object ID
1320	Object exists, but does not match <CustId>
2100	Invalid Datetime - Single date or Low of Range
2110	Invalid Datetime – High of Range
2120	Invalid Datetime Range
2150	Datetime Too Far In Future
2200	History Not Available for Full Date Range

A.5.16   Payment Synchronization (PmtSync)

<StatusCode>	<Desc>
1	Client Up to Date
1100	Too Many Records Requested
1180	Token Value Prior to Available History
1200	Invalid Token

A.5.17   Payment Reversal (PmtRev)

<StatusCode>	<Desc>
810	Message can not be reversed.

A.5.18   Payment Authorization Add (PmtAuthAdd)

<StatusCode>	<Desc>
1280	Object already exists

\<StatusCode\>	\<Desc\>
2000	Invalid Data in Single or Low Amount Field
2020	Amount Too Small
2030	Amount Too Large
2180	Request is Too Late for Today's Work
2190	Weekend or Holiday
3320	Total Payment Amounts Scheduled for Today exceeded daily limit
3380	Expired Card
3610	Authorization is declined for insufficient funds.
3620	Authorization is declined for inactive account.
3630	Authorization is declined for closed account.
3640	Authorization is declined for other reason.

A.5.19   Payment Authorization Modify (PmtAuthMod)

\<StatusCode\>	\<Desc\>
800	Transaction already committed
1060	Cannot Modify Element
1080	No changes made
1260	Unknown Object ID
1320	Object exists, but does not match \<CustId\>
2000	Invalid Data in Single or Low Amount Field
2020	Amount Too Small
2030	Amount Too Large
2180	Request is Too Late for Today's Work
2190	Weekend or Holiday
3320	Total Payment Amounts Scheduled for Today exceeded daily limit
3380	Expired Card
3610	Authorization is declined for insufficient funds.
3620	Authorization is declined for inactive account.
3630	Authorization is declined for closed account.
3640	Authorization is declined for other reason.

A.5.20   Payment Authorization Cancel (PmtAuthCan)

\<StatusCode\>	\<Desc\>
800	Transaction already committed
1260	Unknown Object ID
1320	Object exists, but does not match \<CustId\>

A.5.21 Payment Authorization Inquiry (PmtAuthInq)

<StatusCode>	*<Desc>*
1050	Invalid Enum Value
1100	Too Many Records Requested
1120	No Data Matches Search Criteria
1140	Some Selection Criteria Not Supported
1160	None of Selection Criteria Supported
1260	Unknown Object ID
1320	Object exists, but does not match <CustId>
2000	Invalid Data in Single or Low Amount Field
2100	Invalid Datetime - Single date or Low of Range
2110	Invalid Datetime – High of Range
2120	Invalid Datetime Range
2200	History Not Available for Full Date Range

A.5.22 Payment Authorization Audit (PmtAuthAud)

<StatusCode>	*<Desc>*
1	Client Up to Date
1050	Invalid Enum Value
1100	Too Many Records Requested
1120	No Data Matches Search Criteria
1140	Some Selection Criteria Not Supported
1160	None of Selection Criteria Supported
1260	Unknown Object ID
1320	Object exists, but does not match <CustId>
2100	Invalid Datetime - Single date or Low of Range
2110	Invalid Datetime – High of Range
2120	Invalid Datetime Range
2200	History Not Available for Full Date Range

A.5.23 Payment Authorization Synchronization (PmtAuthSync)

<StatusCode>	*<Desc>*
1	Client Up To Date
1100	Too Many Records Requested
1180	Token Value Prior to Available History
1200	Invalid Token

A.5.24   Payment Authorization Reversal (PmtAuthRev)

<StatusCode>	*<Desc>*
810	Message can not be reversed.

A.5.25   Remittance Add (RemitAdd)

<StatusCode>	*<Desc>*
1280	Object already exists
1500	Invalid Address
1520	Invalid City
1540	Invalid State or Province
1560	Invalid Postal Code
1580	Invalid Country Code
2000	Invalid Data in Single or Low Amount Field
2100	Invalid Datetime – Single date or Low of Range
2150	Datetime Too Far in Future
3100	Invalid <BillerId>
3180	Invalid Payee Name
3240	Invalid Customer Account with Payee

A.5.26   Remittance Modify (RemitMod)

<StatusCode>	*<Desc>*
800	Object Already Committed
1080	No changes made
1260	Unknown Object ID
1320	Object exists, but does not match <CustId>
1500	Invalid Address
1520	Invalid City
1540	Invalid State or Province
1560	Invalid Postal Code
1580	Invalid Country Code
2000	Invalid Data in Single or Low Amount Field
2100	Invalid Datetime – Single date or Low of Range
2150	Datetime Too Far in Future
3100	Invalid <BillerId>
3180	Invalid Payee Name
3240	Invalid Customer Account with Payee

A.5.27   Remittance Status Modify (RemitStatusMod)

<StatusCode>	*<Desc>*

<StatusCode>	<Desc>
1050	Invalid Enum Value
1060	Cannot Modify Element
1080	No Changes made
1260	Unknown Object ID
1320	Object exists, but does not match <CustId>
2100	Invalid Datetime – Single Date or Low of Range
2150	Datetime Too Far in Future

A.5.28   Remittance Delete (RemitDel)

<StatusCode>	<Desc>
800	Object Already Committed
1260	Unknown Object ID
1320	Object exists, but does not match <CustId>

A.5.29   Remittance Inquiry (RemitInq)

<StatusCode>	<Desc>
1050	Invalid Enum Value
1120	No Data Matches Selection Criteria
1140	Some Selection Criteria Not Supported
1160	None of Selection Criteria Supported
1260	Unknown Object ID
1320	Object exists, but does not match <CustId>
2000	Invalid Data in Single or Low Amount Field
2100	Invalid Datetime - Single date or Low of Range
2110	Invalid Datetime – High of Range
2120	Invalid Datetime Range
2150	Datetime Too Far In Future
2200	History Not Available for Full Date Range

A.5.30   Remittance Audit (RemitAud)

<StatusCode>	<Desc>
1	Client Up to Date
1050	Invalid Enum Value
1100	Too Many Records Requested
1120	No Data Matches Selection Criteria
1140	Some Selection Criteria Not Supported
1160	None of Selection Criteria Supported
1260	Unknown Object ID

<StatusCode>	<Desc>
1320	Object exists, but does not match <CustId>
2100	Invalid Datetime - Single date or Low of Range
2110	Invalid Datetime – High of Range
2120	Invalid Datetime Range
2150	Datetime Too Far In Future
2200	History Not Available for Full Date Range

A.5.31 Remittance Synchronization (RemitSync)

<StatusCode>	<Desc>
1	Client Up to Date
1100	Too Many Records Requested
1180	Token Value Prior to Available History
1200	Invalid Token

A.5.32 Remit Reversal (RemitRev)

<StatusCode>	<Desc>
810	Message can not be reversed.

A.5.33 Recurring Payment Model Add (RecPmtAdd)

<StatusCode>	<Desc>
1280	Object already exists
1480	Edit Mask Error
1500	Invalid Address
1520	Invalid City
1540	Invalid State or Province
1560	Invalid Postal Code
1580	Invalid Country Code
2020	Amount Too Small
2030	Amount Too Large
2130	Date Passed
2140	Datetime Too Soon
2150	Datetime Too Far in Future
2160	Processing Date Precedes Today
2180	Request is Too Late for Today's Work
2190	Weekend or Holiday
2310	Destination Account Invalid (for Transfer Payee Payments)
2780	UnsupportedUnsupported Frequency
2910	Duplicate Recurring Model

<StatusCode>	<Desc>
3140	Invalid Customer PayeeID<CustPayeeId>
3180	Invalid Payee Name
3200	Payee List Full
3240	Invalid Customer Account with Payee
3260	Payee Type Not Supported
3380	Expired Card
3400	Recurring Model Open-Ended

A.5.34   Recurring Payment Instance Add (RecPmtInstAdd)

<StatusCode>	<Desc>
1060	Cannot Modify Element
1080	No Changes Made
1260	Unknown Object ID
1300	Object not deleted; dependent objects exist
1310	Cascade Delete Failed
1320	Object exists, but does not match <CustId>
2020	Amount Too Small
2030	Amount Too Large
2130	Date Passed
2140	Datetime Too Soon
2150	Datetime Too Far in Future
2160	Processing Date Precedes Today
2180	Request is Too Late for Today's Work
2190	Weekend or Holiday
2310	Destination Account Invalid (for Transfer Payee Payments)
2780	Unsupported Frequency
2910	Duplicate Recurring Model
2920	Invalid Skip Count
3140	Invalid Customer Payee ID<CustPayeeId>
3180	Invalid Payee Name
3200	Payee List Full
3240	Invalid Customer Account with Payee
3260	Payee Type Not Supported
3380	Expired Card
3400	Recurring Model Open-Ended

A.5.35   Recurring Payment Model Modify (RecPmtMod)

<StatusCode>	<Desc>
1060	Cannot Modify Element

<StatusCode>	<Desc>
1080	No Changes Made
1260	Unknown Object ID
1300	Object not deleted; dependent objects exist
1310	Cascade Delete Failed
1320	Object exists, but does not match <CustId>
2020	Amount Too Small
2030	Amount Too Large
2130	Date Passed
2140	Datetime Too Soon
2150	Datetime Too Far in Future
2160	Processing Date Precedes Today
2180	Request is Too Late for Today's Work
2190	Weekend or Holiday
2310	Destination Account Invalid (for Transfer Payee Payments)
2780	Unsupported Frequency
2910	Duplicate Recurring Model
2920	Invalid Skip Count
3140	Invalid Customer Payee ID<CustPayeeId>
3180	Invalid Payee Name
3200	Payee List Full
3240	Invalid Customer Account with Payee
3260	Payee Type Not Supported
3380	Expired Card
3400	Recurring Model Open-Ended

A.5.36   Recurring Payment Model Cancel (RecPmtCan)

<StatusCode>	<Desc>
800	Object Object Already CommittedAC
1260	Unknown Object ID
1300	Object Not Deleted; dependent objects exist
1310	Cascade Delete Failed
1320	Object exists, but does not match <CustId>

A.5.37   Recurring Payment Model Inquiry (RecPmtInq)

<StatusCode>	<Desc>
1100	Too Many Records Requested
1120	Selection No Data Matches Selection Criteria
1140	Some Selection Criteria Not Supported
1160	None of Selection Criteria Supported

<StatusCode>	<Desc>
1260	Unknown Object ID
1320	Object exists, but does not match <CustId>
2000	Invalid Data in Single or Low Amount Field
2010	Invalid Data in High Amount Field
2060	Invalid High/Low Amount
2100	Invalid Datetime - Single date or Low of Range
2110	Invalid Datetime – High of Range
2120	Invalid Datetime Range
2150	Datetime Too Far In Future
2200	History Not Available for Full Date Range
2300	Single or Source Account Invalid
2440	Invalid Single or Source Account Type
2510	Invalid Card Magnetic Strip Data
2520	Unrecognized / Invalid Card Issuer
2530	Unrecognized Card Number
2540	Stolen Card
3140	Invalid Customer Payee ID<CustPayeeId>
3380	Expired Card

A.5.38   Recurring Payment Model Audit (RecPmtAud)

<StatusCode>	<Desc>
1	Client Up To Date
1050	Invalid Enum Value
1100	Too Many Records Requested
1120	Selection No Data Matches Selection Criteria
1140	Some Selection Criteria Not Supported
1160	None of Selection Criteria Supported
1260	Unknown Object ID
1320	Object exists, but does not match <CustId>
2100	Invalid Datetime - Single date or Low of Range
2110	Invalid Datetime – High of Range
2120	Invalid Datetime Range
2150	Datetime Too Far In Future
2200	History Not Available for Full Date Range

A.5.39   Recurring Payment Model Synchronization (RecPmtSync)

<StatusCode>	<Desc>
1	Client Up to Date

<StatusCode>	<Desc>
1100	Too Many Records Requested
1180	Token Value Prior to Available History
1200	Invalid Token

A.5.40 Payment Acknowledgment Inquiry (PmtAckInq)

<StatusCode>	<Desc>
1050	Invalid Enum Value
1100	Too Many Records Requested
1120	Selection No Data Matches Selection Criteria
1140	Some Selection Criteria Not Supported
1160	None of Selection Criteria Supported
1260	Unknown Object ID
1320	Object exists, but does not match <CustId>
2100	Invalid Datetime - Single date or Low of Range
2110	Invalid Datetime – High of Range
2120	Invalid Datetime Range
2150	Datetime Too Far In Future
2200	History Not Available for Full Date Range
2300	Single or Source Account Invalid
2440	Invalid Single or Source Account Type
2510	Invalid Card Magnetic Strip Data
2520	Unrecognized / Invalid Card Issuer
2530	Unrecognized Card Number
2540	Stolen Card
3140	Invalid Customer Payee ID <CustPayeeId>
3380	Expired Card

A.5.41 Payment Acknowledgement Reversal (PmtAckRev)

<StatusCode>	<Desc>
810	Message can not be reversed.

A.5.42 Checksum Add (ChksumAdd)

<StatusCode>	<Desc>
1280	Object already exists
1480	Edit Mask Error
2020	Amount Too Small
2030	Amount Too Large
2130	Date Passed

<StatusCode>	<Desc>
2160	Processing Date Precedes Today
2180	Request is Too Late for Today's Work
2190	Weekend or Holiday
3140	Invalid Customer Payee ID <CustPayeeId>
3580	Authorization is rejected
3600	Authorization held over

A.5.43   Checksum Modify (ChksumMod)

<StatusCode>	<Desc>
800	Object Already Committed
1080	No changes made
1260	Unknown Object ID
1320	Object exists, but does not match <CustId>
2020	Amount Too Small
2030	Amount Too Large
2130	Date Passed
2160	Processing Date Precedes Today
2180	Request is Too Late for Today's Work
2190	Weekend or Holiday
3140	Invalid Customer Payee ID<CustPayeeId>
3580	Authorization is rejected
3600	Authorization held over

A.5.44   Checksum Status Modify (ChksumStatusMod)

<StatusCode>	<Desc>
1050	Invalid Enum Value
1060	Cannot Modify Element
1080	No changes made
1260	Unknown Object ID
1320	Object Exists But Does Not Match <CustId>
1760	Authorization Failure
3580	Authorization is rejected
3600	Authorization held over

A.5.45   Checksum Cancel (ChksumCan)

<StatusCode>	<Desc>
800	Object Object Already Committed
1260	Unknown Object ID

<StatusCode>	<Desc>
1320	Object exists, but does not match <CustId>

A.5.46   Checksum Inquiry (ChksumInq)

<StatusCode>	<Desc>
1050	Invalid Enum Value
1100	Too Many Records Requested
1120	Selection No Data Matches Selection Criteria
1140	Some Selection Criteria Not Supported
1160	None of Selection Criteria Supported
1260	Unknown Object ID
1320	Object exists, but does not match <CustId>

A.5.47   Checksum Audit (ChksumAud)

<StatusCode>	<Desc>
1050	Invalid Enum Value
1100	Too Many Records Requested
1120	Selection No Data Matches Selection Criteria
1140	Some Selection Criteria Not Supported
1160	None of Selection Criteria Supported
1260	Unknown Object ID
1320	Object exists, but does not match <CustId>
2100	Invalid Datetime - Single date or Low of Range
2110	Invalid Datetime – High of Range
2120	Invalid Datetime Range
2150	Datetime Too Far In Future
2200	History Not Available for Full Date Range

A.5.48   Checksum Synchronization (ChksumSync)

<StatusCode>	<Desc>
1100	Too Many Records Requested
1180	Token Value Prior to Available History
1200	Invalid Token

A.5.49   Comprehensible Remittance Statement Add (CompRemitStmtAdd)

<StatusCode>	<Desc>
1280	Object already exists
1500	Invalid Address

<StatusCode>	<Desc>
1520	Invalid City
1560	Invalid State or Province
1580	Invalid Country Code
2000	Invalid Data in Single or Low Amount Field
2150	Datetime Too Far in Future
2300	Single or Source Account Invalid
2350	Single or Source Account Closed
3100	Invalid <BillerId>
3180	Invalid Payee Name

A.5.50 Comprehensible Remittance Statement Inq (CompRemitStmtInq)

<StatusCode>	<Desc>
1050	Invalid Enum Value
1120	No Data matches Selection Criteria
1140	Some Selection Criteria Not Supported
1160	None of Selection Criteria Supported
1260	Unknown Object ID
1320	Object exists, but does not match <CustId>
2000	Invalid Data in Single or Low Amount Field
2100	Invalid Datetime – Single date or Low of Range
2110	Invalid Datetime – High of Range
2120	Invalid Datetime range
2150	Datetime Too Far In Future
2200	History Not Available for Full Date Range
2300	Single or Source Account Invalid
2350	Single or Source Account Closed

A.5.51 Comprehensible Remittance Statement Audit (CompRemitStmtAud)

<StatusCode>	<Desc>
1	Client Up to Date
1050	Invalid Enum Value
1100	Too Many Records Requested
1120	No Data Matches Selection Criteria
1140	Some Selectino Criteria Not Supported
1160	None of Selection Criteria Supported
1260	Unknown Object ID
1320	Object Exists, but does not match <CustId>

\<StatusCode\>	\<Desc\>
2100	Invalid Datetime – Single date or Low of Range
2110	Invalid Datetime – High of Range
2120	Invalid Datetime Range
2150	Datetime Too Far in Future
2200	History Not Available for Full Date Range

A.5.52 *Comprehenesible Remittance Statement Synchronization (ComprRemitStmtSync)*

\<StatusCode\>	\<Desc\>
1	Client Up to Date
1100	Too many Records Requested
1180	Token Value Prior to Available History
1200	Invalid Token

A.6 Bill Presentment Service

A.6.1 *Biller Inquiry (BillerInq)*

\<StatusCode\>	\<Desc\>
1	Client Up to Date
1100	Too Many Records Requested
1120	Selection No Data Matches Selection Criteria
1140	Some Selection Criteria Not Supported
1160	None of Selection Criteria Supported
1400	\<SPName\> within \<BillerId\> is invalid
1500	Invalid Address
1520	Invalid City
1540	Invalid State Or Province
1560	Invalid Postal Code
1580	Invalid Country Code
2100	Invalid Datetime - Single date or Low of Range
2150	Datetime Too Far in Future
2500	Invalid Industry Id
3100	Invalid \<BillerId\>\<BillerId\>

A.6.2 *Biller Reversal (BillerRev)*

\<StatusCode\>	\<Desc\>
810	Message can not be reversed.

A.6.3   Bill Inquiry (BillInq)

<StatusCode>	<Desc>
1050	Invalid Enum Value
1100	Too Many Records Requested
1120	No Records Match Selection Criteria
1140	Some Selection Criteria Not Supported
1160	None of selection criteria supported
1260	Unknown Object ID
1320	Object exists, but does not match <CustId>
1400	<SPName> within <BillerId> is invalid
2100	Invalid Datetime - Single date or Low of Range
2110	Invalid Datetime - High of Range
2120	Invalid Datetime Range
2150	Datetime Too Far in Future
2200	History Not Available for Full Date Range
3100	Invalid <BillerId>

A.6.4   Bill Status Modification (BillStatusMod)

<StatusCode>	<Desc>
1050	Invalid Enum Value
1060	Cannot Modify Element
1080	No Changed made
1260	Unknown Object ID
1320	Object exists, but does not match <CustId>
1060	Cannot Modify Element
2100	Invalid Datetime - Single date or Low of Range
3140	Invalid Customer Payee ID
3180	Invalid Payee Name
3240	Invailid Customer Account With Payee

A.6.5   Bill Reversal (BillRev)

<StatusCode>	<Desc>
810	Message can not be reversed.

A.7   Root Services

A.7.1   Terminal Object Add (TerminalObjAdd)

<StatusCode>	<Desc>
1040	Request Declined
1280	Object already exists.

A.7.2 Terminal Object Modify (TerminalObjMod)

<StatusCode>	<Desc>
1050	Invalid Enum Value
1060	Cannot Modify Element
1080	No Changes Made
1260	Unknown Object ID

A.7.3 Terminal Object Inquiry (TerminalObjInq)

<StatusCode>	<Desc>
1050	Invalid Enum Value
1120	No Data Matches Selection Criteria
1140	Some Selection Criteria Not Supported
1160	None of Selection Criteria Supported
1260	Unknown Object Id
1520	Invalid City
1540	Invalid State or Province
1560	Invalid Postal Code

A.7.4 Terminal Object Delete (TerminalObjDel)

<StatusCode>	<Desc>
1260	Unknown Object ID

A.7.5 Terminal Object Audit (TerminalObjAud)

<StatusCode>	<Desc>
1	Client Up To Date
1050	Invalid Enum Value
1100	Too Many Records Requested
1120	No Data Matches Selection Criteria
1140	Some Selection Criteria Not Supported
1160	None of Selection Criteria Supported
1180	Token Value Prior to Available History
1200	Invalid Token
1260	Unknown Object ID

A.7.6 Terminal Object Status Modify (TerminalObjStatusMod)

<StatusCode>	<Desc>

<StatusCode>	<Desc>
1050	Invalid Enum Value
1080	No Changes Made
1260	Unknown Object ID
4010	Status Code did not change.

A.7.7 Terminal Service Provider Object Add (TerminalSPObjAdd)

<StatusCode>	<Desc>
1040	Request Declined
1280	Object already exists.

A.7.8 Terminal Service Provider Object Modify (TerminalSPObjMod)

<StatusCode>	<Desc>
1050	Invalid Enum Value
1060	Cannot Modify Element
1080	No Changes Made
1260	Unknown Object ID

A.7.9 Terminal Service Provider Object Inquiry (TerminalSPObjInq)

<StatusCode>	<Desc>
1050	Invalid Enum Value
1120	No Data Matches Selection Criteria
1140	Some Selection Criteria Not Supported
1160	None of Selection Criteria Supported
1260	Unknown Object Id
1520	Invalid City
1540	Invalid State or Province
1560	Invalid Postal Code

A.7.10 Terminal Service Provider Object Delete (TerminalSPObjDel)

<StatusCode>	<Desc>
1260	Unknown Object ID

A.7.11 Terminal Server Provider Object Audit (TerminalSPObjAud)

<StatusCode>	<Desc>
1	Client Up To Date
1050	Invalid Enum Value
1100	Too Many Records Requested
1120	No Data Matches Selection Criteria
1140	Some Selection Criteria Not Supported
1160	None of Selection Criteria Supported
1180	Token Value Prior to Available History
1200	Invalid Token
1260	Unknown Object ID

A.7.12 Device Inquiry (DevInq)

<StatusCode>	<Desc>
1050	Invalid Enum Value
1120	No Data Matches Selection Criteria
1140	Some Selection Criteria Not Supported
1160	None of Selection Criteria Supported
1260	Unknown Object Id
1520	Invalid City
1540	Invalid State or Province
1560	Invalid Postal Code

B   Compatibility Policy

B.1   Introduction

This appendix defines the compatibility policy for future editions of the IFX Specification. A number written as 'n1.n2.n3' identifies the specification's edition and represents three levels of change control where 'n1' is the version level, 'n2' is the release level and 'n3' is the maintenance level. The following sections provide the definitions and rules for changing editions to insure appropriate compatibility between the new and old levels.

B.2   Definitions

Before proceeding to the detailed change rules, it is important that the following key terms are defined.

B.2.1   Compatibility

Compatibility, as used in this policy, is dichotomized into two types. They are syntactic compatibility and semantic compatibility, and are defined as:

- *Syntactic compatibility*: is the consistent representation of message structure, i.e., formats, tag names, and element definitions, between old and new levels of the specification.
- *Semantic compatibility*: is the consistent representation of message function, i.e., capability, between old and new levels of the specification.

B.2.2   Change Levels

As mentioned above, the three control levels to identify change are version, release and maintenance. Starting with the lowest form of change, the definition of each change level is:

- *Maintenance Level Change*: used when correcting defects or providing clarification to the previous edition of the specification. Syntactic and semantic compatibility with previous edition must be maintained. Requires Steering Committee approval by majority vote.
- *Release Level Change*: used when adding new functionality to the specification. Same architecture must be followed. Syntactic and semantic compatibility with previous release level must be maintained from a standpoint that the new function does not impact usage of previous function. Requires Steering Committee approval by majority vote.
- *Version Level Change*: used when significantly modifying the architecture of the specification. Semantic compatibility with previous version must be maintained unless deviation is approved by the Steering Committee. However, syntactic compatibility with previous version is not guaranteed. As a result, version changes are not expected with any frequency. Architecture Committee must provide advanced warning of potential version level changes to Steering Committee. Requires Steering Committee approval by two-thirds majority vote.

B.3   Compatibility Rules

Following are the rules for maintaining syntactic and semantic compatibility between new and old levels of the IFX Specification:

- All specification changes that impact IFX messages must be fully compatible with the previous level of the specification. Specification changes must not require changes by an implementation, in order to continue processing at the same level of function.
- All new IFX fields and aggregates must be optional. However, a field (or aggregate) within a new aggregate may be required. An absent field (or aggregate) means "value not specified". Any processes using the new field (or aggregate) must be constructed such that, when the field (or aggregate) is absent, processing results are the same as before the field (or aggregate) existed. No warnings or other "non-zero" completion codes should be returned if the field (or aggregate) is absent.

- Optional IFX fields and aggregates must not later become mandatory. An implementation must not be required to populate an IFX field or aggregate to support a function it does not use.
- Repeating IFX fields or aggregates must not later become atomic (or vice versa).
- IFX fields and aggregates must not be removed from IFX messages.
- Data types of IFX fields must not change except for length increases.
- Where the domain of a field, e.g., a closed or open Enum, is a set of discrete valid values:
  - The semantic meaning of a valid value must not change.
  - A valid value must not later become invalid.
- The semantic meaning of an IFX field may only be clarified, not changed.
- The IFX tags of existing fields, aggregates or messages must not change.

B.3.1   Policy Guidelines

This section defines compatibility policy guidelines for other key factors that need to be safeguarded when creating a new edition of the specification.

B.3.1.1   Performance

A new edition of the IFX specification should aim for equivalent performance with the previous edition for the same feature set using required elements. Obviously, new features and additional optional data elements would typically imply an associated performance cost (although not necessarily).

B.3.1.2   Security

Application level security of a new edition of the IFX specification should provide compatible application level security support for the previous edition. Any deviation of security compatibility must be approved by the Steering Committee. However, cases may arise that would justify circumventing the security guideline as related to syntactic and/or semantic compatibility. One such example is where a security exposure is discovered with the approach used in the previous edition and an incompatible change is required. Another example is where the industry's security expectation level has increased such that it obsolesces the acceptability of the previous edition's security.

B.4   Effective Start Time

The IFX Forum has formally adopted this release-to-release compatibility policy starting with IFX Specification Version 1.0.1 as its baseline release. Therefore, the policy will be in effect for all subsequent releases of the specification until otherwise indicated by the IFX Forum.

B.5   Managing Change for Migrational Ease

As the IFX Specification continues to extend its features and services, it may be necessary to modify or add new building blocks (i.e., fields, aggregates, and/or messages) that obsolesces older constructs. For compliance with this Compatibility Policy, the obsolete constructs will only be removed from the specification during a version level change. Since version level changes are infrequent, this provides sufficient overlap time for migration from the old to new constructs.

B.5.1   Deprecation List

The following table outlines the IFX elements and aggregates that will be replaced in Version 2.0. Both the original and replacement elements and aggregates will be available until Version 2.0.

Original Tag	Replacement Tag	Replacement Introduced in Version
Customer Name Aggregate <CustName>	Person Name Aggregate <PersonName>	1.1.0
Customer Contact Aggregate <CustContact>	Contact Information Aggregate <ContactInfo>	1.1.0

Original Tag	Replacement Tag	Replacement Introduced in Version
Organization Contact Aggregate <OrgContact>	Contact Information Aggregate <ContactInfo>	1.1.0
Customer Name/Address Aggregate <CustNameAddr>	Person Information Aggregate <PersonInfo>	1.1.0
Transfer From Supported Aggregate <XferFromSupt> in Bank Account Record Aggregate <BankAcctRec>	Bank Account Feature Support Aggregate <BankAcctFeatSupt>	1.1.0
Transfer To Supported Aggregate <XferToSupt> in Bank Account Record Aggregate <BankAcctRec>	Bank Account Feature Support Aggregate <BankAcctFeatSupt>	1.1.0
Payment Supported Aggregate <PaySupt> in Bank Account Record Aggregate <BankAcctRec>	Bank Account Feature Support Aggregate <BankAcctFeatSupt>	1.1.0
ATM Message Information Aggregate <ATMTrnInfo> in Deposit Account Transaction Record Aggregate <DepAcctTrnRec>	Network Transaction Information Aggregate <NetworkTrnInfoRec>	1.1.0
ACH Message Information Aggregate <USA.ACHTrnInfo> in Deposit Account Transaction Record Aggregate <DepAcctTrnRec>	Network Transaction Information Aggregate <NetworkTrnInfoRec>	1.1.0
Settlement Information Aggregate <SettlementInfo> in Payment Instrument Aggregate <PmtInst>	Settlement Information Aggregate <SettlementInfo>	1.1.0
Stop Check Information Aggregate <StopChkInfo> **in Bank Account Transaction Image Inquiry (see section6.4.14) only**.	Check Number <ChkNum> and several other elements and aggregates	1.2.0
Customer Password Modfy Message <Cust PswdModRq>,<CustPswdModRs>	Customer Authentication Modify Message <CustAuthModRq>, <CustAuthModRs>	1.2.0
Credit Authorization Type element <CreditAuthType> in Credit Information Aggregate <CreditInfo>	Credit Type element <CreditType> in Credit Information Aggregate <CreditInfo>	1.4.0
Network Transaction Information Aggregate <NetworkTrnInfo> in Credit Information Aggregate <CreditInfo>	Network Transaction Information Aggregate <NetworkTrnInfo> in Message Request Header Aggregate <MsgRqHdr>	1.4.0
Client Change Code element <ClientChgCode> in Credit Authorization Cancel Message <CreditAuthCanRq> <CreditAuthCanRs>	None	1.4.0
Client Change Code element <ClientChgCode> in Credit Authorization Inquiry Message <CreditAuthInqRq> <CreditAuthInqRs>	None	1.4.0
Debit Authorization Type element <DebitAuthType> in Debit Information Aggregate <DebitInfo>	Debit Type element <DebitType> in Debit Information Aggregate <DebitInfo>	1.4.0
Network Transaction Information Aggregate <NetworkTrnInfo> in Debit Information Aggregate <DebitInfo>	Network Transaction Information Aggregate <NetworkTrnInfo> in Message Request Header Aggregate <MsgRqHdr>	1.4.0

Original Tag	Replacement Tag	Replacement Introduced in Version
Client Change Code element <ClientChgCode> in Debit Authorization Cancel Message <DebitAuthCanRq> <DebitAuthCanRs>	none	1.4.0
Client Change Code element <ClientChgCode> in Debit Authorization Inquiry Message <DebitAuthInqRq> <DebitAuthInqRs>	none	1.4.0

C   Index