­­



FIX Recommended Practices

Securities Settlement Status Management

FINANCIAL INFORMATION EXCHANGE (FIX)

RECOMMENDED PRACTICES

Global Post Trade Committee

Securities Settlement Status Management

July 15, 2023

Version [0.2] (Public Comment)

**Table of Contents**

[Preface 8](#_Toc140745004)

[1 Executive Summary 8](#_Toc140745005)

[2 Objectives 8](#_Toc140745006)

[3 Scope 8](#_Toc140745007)

[3.1 Out of Scope 9](#_Toc140745008)

[4 Target Audience 9](#_Toc140745009)

[5 Authors 9](#_Toc140745010)

[6 Security Settlement Status - Recommended Practices 9](#_Toc140745011)

[7 FIX Message Workflows 10](#_Toc140745012)

[7.1 Affirmation Drop-copy for Settlement Status 11](#_Toc140745013)

[7.2 Settlement Status Request and Report 12](#_Toc140745014)

[7.3 Usage of SettlStatusReason(2969) and SettlStatus(2968) fields 13](#_Toc140745015)

[8 Message and Component Tables 15](#_Toc140745016)

[8.1 Messages 15](#_Toc140745017)

[8.1.1 SettlementStatusRequest(35=EC) 15](#_Toc140745018)

[8.1.2 SettlementStatusRequestAck(35=ED) 16](#_Toc140745019)

[8.1.3 SettlementStatusReport(35=EE) 17](#_Toc140745020)

[8.1.4 SettlementStatusReportAck(35=EF) 19](#_Toc140745021)

[8.2 Components 20](#_Toc140745022)

[8.2.1 Parties 20](#_Toc140745023)

[8.2.2 RegulatoryTradeIDGrp 20](#_Toc140745024)

[8.2.3 SettlTradeDetails 21](#_Toc140745025)

[8.2.4 Instrument 23](#_Toc140745026)

[8.2.5 NestedParties 23](#_Toc140745027)

[8.2.6 SettlInstructionsData 24](#_Toc140745028)

[8.2.7 DlvnInstGrp 24](#_Toc140745029)

[8.2.8 SettlParties 25](#_Toc140745030)

[8.2.9 SettlPtysSubGrp 25](#_Toc140745031)

DISCLAIMER

THE INFORMATION CONTAINED HEREIN AND THE FINANCIAL INFORMATION EXCHANGE PROTOCOL (COLLECTIVELY, THE “FIX PROTOCOL”) ARE PROVIDED “AS IS” AND NO PERSON OR ENTITY ASSOCIATED WITH THE FIX PROTOCOL MAKES ANY REPRESENTATION OR WARRANTY, EXPRESS OR IMPLIED, AS TO THE FIX PROTOCOL (OR THE RESULTS TO BE OBTAINED BY THE USE THEREOF) OR ANY OTHER MATTER AND EACH SUCH PERSON AND ENTITY SPECIFICALLY DISCLAIMS ANY WARRANTY OF ORIGINALITY, ACCURACY, COMPLETENESS, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. SUCH PERSONS AND ENTITIES DO NOT WARRANT THAT THE FIX PROTOCOL WILL CONFORM TO ANY DESCRIPTION THEREOF OR BE FREE OF ERRORS. THE ENTIRE RISK OF ANY USE OF THE FIX PROTOCOL IS ASSUMED BY THE USER.

NO PERSON OR ENTITY ASSOCIATED WITH THE FIX PROTOCOL SHALL HAVE ANY LIABILITY FOR DAMAGES OF ANY KIND ARISING IN ANY MANNER OUT OF OR IN CONNECTION WITH ANY USER'S USE OF (OR ANY INABILITY TO USE) THE FIX PROTOCOL, WHETHER DIRECT, INDIRECT, INCIDENTAL, SPECIAL OR CONSEQUENTIAL (INCLUDING, WITHOUT LIMITATION, LOSS OF DATA, LOSS OF USE, CLAIMS OF THIRD PARTIES OR LOST PROFITS OR REVENUES OR OTHER ECONOMIC LOSS), WHETHER IN TORT (INCLUDING NEGLIGENCE AND STRICT LIABILITY), CONTRACT OR OTHERWISE, WHETHER OR NOT ANY SUCH PERSON OR ENTITY HAS BEEN ADVISED OF, OR OTHERWISE MIGHT HAVE ANTICIPATED THE POSSIBILITY OF, SUCH DAMAGES.

**DRAFT OR NOT RATIFIED PROPOSALS** (REFER TO PROPOSAL STATUS AND/OR SUBMISSION STATUS ON COVER PAGE) ARE PROVIDED “AS IS” TO INTERESTED PARTIES FOR DISCUSSION ONLY. PARTIES THAT CHOOSE TO IMPLEMENT THIS DRAFT PROPOSAL DO SO AT THEIR OWN RISK. IT IS A DRAFT DOCUMENT AND MAY BE UPDATED, REPLACED, OR MADE OBSOLETE BY OTHER DOCUMENTS AT ANY TIME. THE FPL GLOBAL TECHNICAL COMMITTEE WILL NOT ALLOW EARLY IMPLEMENTATION TO CONSTRAIN ITS ABILITY TO MAKE CHANGES TO THIS SPECIFICATION PRIOR TO FINAL RELEASE. IT IS INAPPROPRIATE TO USE FPL WORKING DRAFTS AS REFERENCE MATERIAL OR TO CITE THEM AS OTHER THAN “WORKS IN PROGRESS”. THE FPL GLOBAL TECHNICAL COMMITTEE WILL ISSUE, UPON COMPLETION OF REVIEW AND RATIFICATION, AN OFFICIAL STATUS (“APPROVED”) OF/FOR THE PROPOSAL AND A RELEASE NUMBER.

No proprietary or ownership interest of any kind is granted with respect to the FIX Protocol (or any rights therein), except as expressly set out in FIX Protocol Limited’s Copyright and Acceptable Use Policy.

© Copyright 2003-2023 FIX Protocol Limited, all rights reserved



FIX Recommended Practices by [FIX Protocol Ltd.](https://www.fixtradingcommunity.org/) are licensed under a [Creative Commons Attribution-NoDerivatives 4.0 International License](http://creativecommons.org/licenses/by-nd/4.0/).

**Table of Tables**

A Table of Tables is not required. If used, use styles to tag the captions and auto-generate the list here. If not used, remove this section.

[Table 1: SettlementStatusRequest(35=EC) 14](#_Toc140418521)

[Table 2: SettlementStatusRequestAck(35=ED) 15](#_Toc140418522)

[Table 3: SettlementStatusReport(35=EE) 16](#_Toc140418523)

[Table 4: SettlementStatusReportAck(35=EF) 18](#_Toc140418524)

[Table 5: Parties component 19](#_Toc140418525)

[Table 6: RegulatoryTradeIDGrp component 20](#_Toc140418526)

[Table 7: SettlTradeDetails component 20](#_Toc140418527)

[Table 8: Instrument component 22](#_Toc140418528)

[Table 9: NestedParties component 23](#_Toc140418529)

[Table 10: SettlInstructionsData component 23](#_Toc140418530)

[Table 11: DlvnInstGrp component 24](#_Toc140418531)

[Table 12: SettlParties component 25](#_Toc140418532)

[Table 13: SettlPtysSubGrp component 25](#_Toc140418533)

**Table of Figures**

A Table of Figures is not required. If used, use styles to tag the captions and auto-generate the list here. If not used, remove this section.

[Figure 1: High level activity flow 10](#_Toc140418534)

[Figure 2: Settlement Status Notification 11](#_Toc140418535)

[Figure 3: Settlement Status Request and Status Report 12](#_Toc140418536)

**Document History**

| Revision | Date | Author | Revision Comments |
| --- | --- | --- | --- |
| 0.1 | June 22, 2023 | L. Taikitsadaporn for GPTC | Initial draft |
| 0.2 | June 29, 2023 | L. Jones for FIX Trading Community | Editorial edits |
|  | July 15, 2023 | L. Taikitsadaporn for GPTC | Additional minor edits, reformatted to new template layout. |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# Preface

The purpose of the FIX Trading Community Post-Trade Processing via FIX Initiative is to define industry practices for common usage of the FIX Protocol for post-trade processing, for all asset classes, between buy-sides and sell-sides that can be used bi-laterally as well as through intermediary facilities.

This document is one of a series of Recommended Practices for Post-Trade Processing via FIX specifying recommended practices for industry usage of the FIX standard to facilitate parallel implementation across buy-sides, sell-sides and intermediaries.

This document assumes an understanding of the FIX Protocol and post-trade processing in general.

# Executive Summary

Background text to be added here as an introduction to the document.

This Recommended Practices document covers securities settlement status message flow for communicating the settlement status of a security transaction.

The securities settlement workflow, whether for domestic or cross-border settlement, has a workflow distinct to its operations as it involves the exchange of securities for money. Each step in this process has a status. Today this workflow is facilitated by the relevant parties in the settlement process using SWIFT MT or MX messages through the SWIFT network. In the environment where near real-time information is needed or required, Investment Managers wish to know about the settlement status of their trades as soon as possible rather than end of day or next day.

As an extension of the existing FIX Allocations and Confirmation workflows, this Recommended Practices document details the usage of four FIX message types that allow the broker, custodian or an outsourcer (e.g. vendor system) to communicate security settlement status back to the Investment Manager.

# Objectives

A few sentences to be added here on the main purpose of the document.

The FIX Global Post Trade Committee participants requested new FIX messages to provide a set of standards for a more automated workflow of securities settlement status, allowing for further continuation of FIX throughout the trade lifecycle. This document describes how FIX messages can be used to improve operational efficiencies through real-time status reporting.

# Scope

A bullet list of what is in scope to be added here.

This document includes the FIX Recommended Practices for automating the communication between Investment Managers and the party who is able to provide the security settlement status to the Investment Manager. The operational process of this Recommended Practices picks up where settlement has already been initiated via SWIFT MT or MX, and status from that process is available to be communicated.

The scope of the workflow also allows the Investment Manager to initiate either a one-time status request or subscription request based on transaction identifiers (e.g. using UTI) or by specifying detailed trade information that allows the receiving party to determine which trade's settlement status information is being requested.

It should be noted that while this Recommended Practices documents utilizes diagrams that shows the Custodian as the party providing this information, other parties such as the broker or a 3rd-party services can stand-in as the party providing the information. These general FIX message flows should apply regardless of the actual party involved, although some steps maybe skipped - for example, if the broker who confirmed the allocated trades with the Investment Manager is also the party who provides the security settlement status there may not be a need for the confirmation drop-copy.

## Out of Scope

The following aspect is out-of-scope:

* Initiating the settlement
* Initiating the movement of cash to fund a securities transaction

# Target Audience

Committee / Working Group to identify target audience as appropriate.

We encourage buy-side participants, such as Investment Managers, Hedge Funds and Corporates, Brokers/Dealers and Prime Brokers, Custodians and third-party service providers or platforms to use these Recommended Practices in order to automate via FIX the settlement status of securities transactions.

# Authors

Optional Item: Committee / Working Group to identify document authors.

The authors of this Recommended Practices document are the member participants in the FIX Global Post-trade Committee, with editorial participation from FIX Trading Community consultant Lisa Taikitsadaporn.

# Security Settlement Status - Recommended Practices

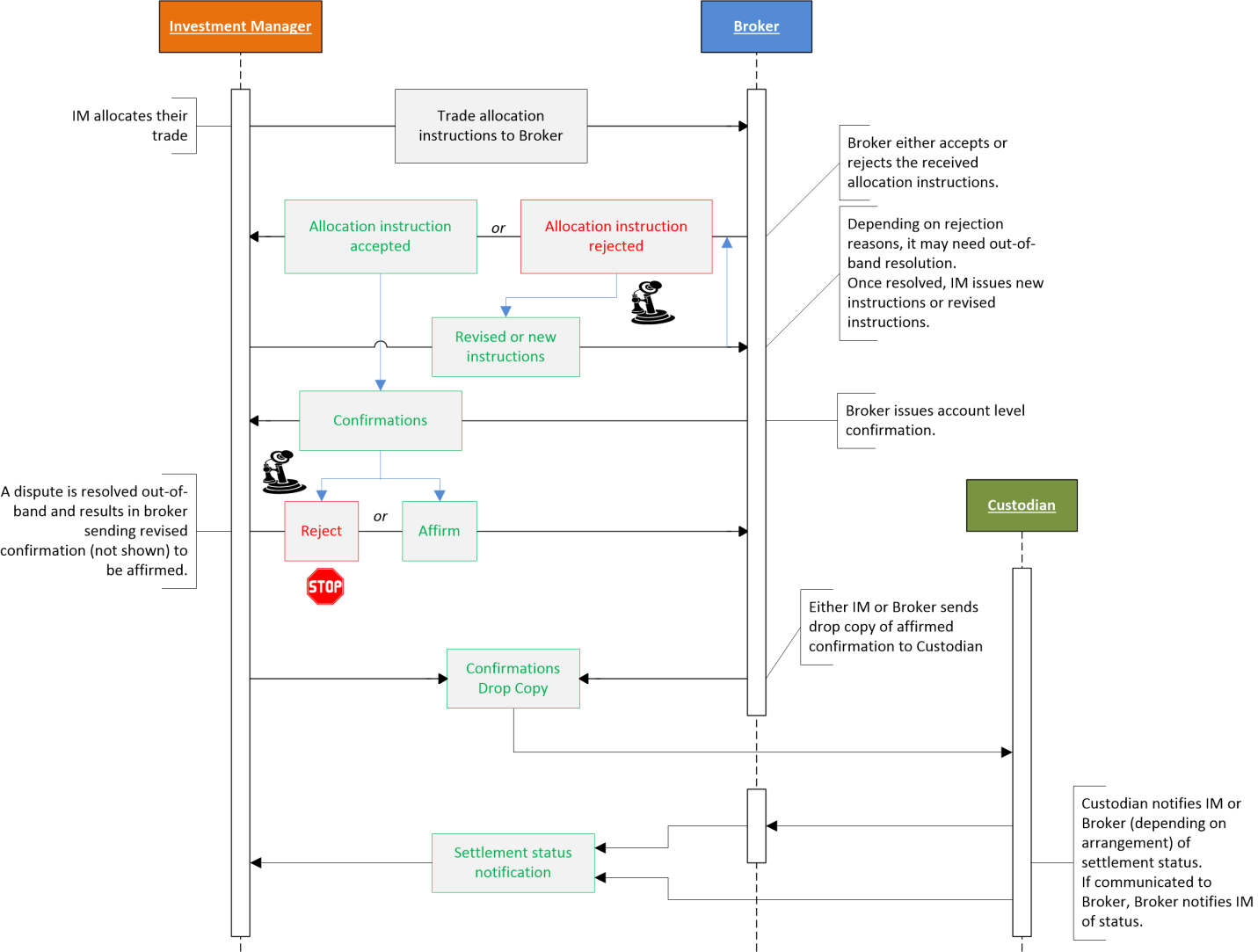
Main body of the document. Committee / Working Group to structure this as appropriate.

The diagram below illustrates at a high level the Custodian as the party communicating the security settlement status back to the investment manager directly. However this function may also be performed by the broker or a 3rd-party outsourcer, depending on the investment manager's setup.

At a high level this process begins after post-trade confirmation has been affirmed between the Investment Manager and the broker.

It should be noted that the existing FIX post-trade allocations and confirmation process as documented in the different FIX Recommended Practices are not affected. The communication of the settlement status picks up after the confirmation has been affirmed (or acknowledged as done in some markets) by the investment manager and the settlement process is initiated using SWIFT MT/MX messages.

Figure 1: High level activity flow



# FIX Message Workflows

Optional Item: Diagrams or verbose description of the workflows applicable to the scope of this document.

This section describes the specific FIX message flows for the different scenarios covered by this Recommended Practices document. Two broad workflows are detailed below that can be adapted to specific implementations.

## Affirmation Drop-copy for Settlement Status

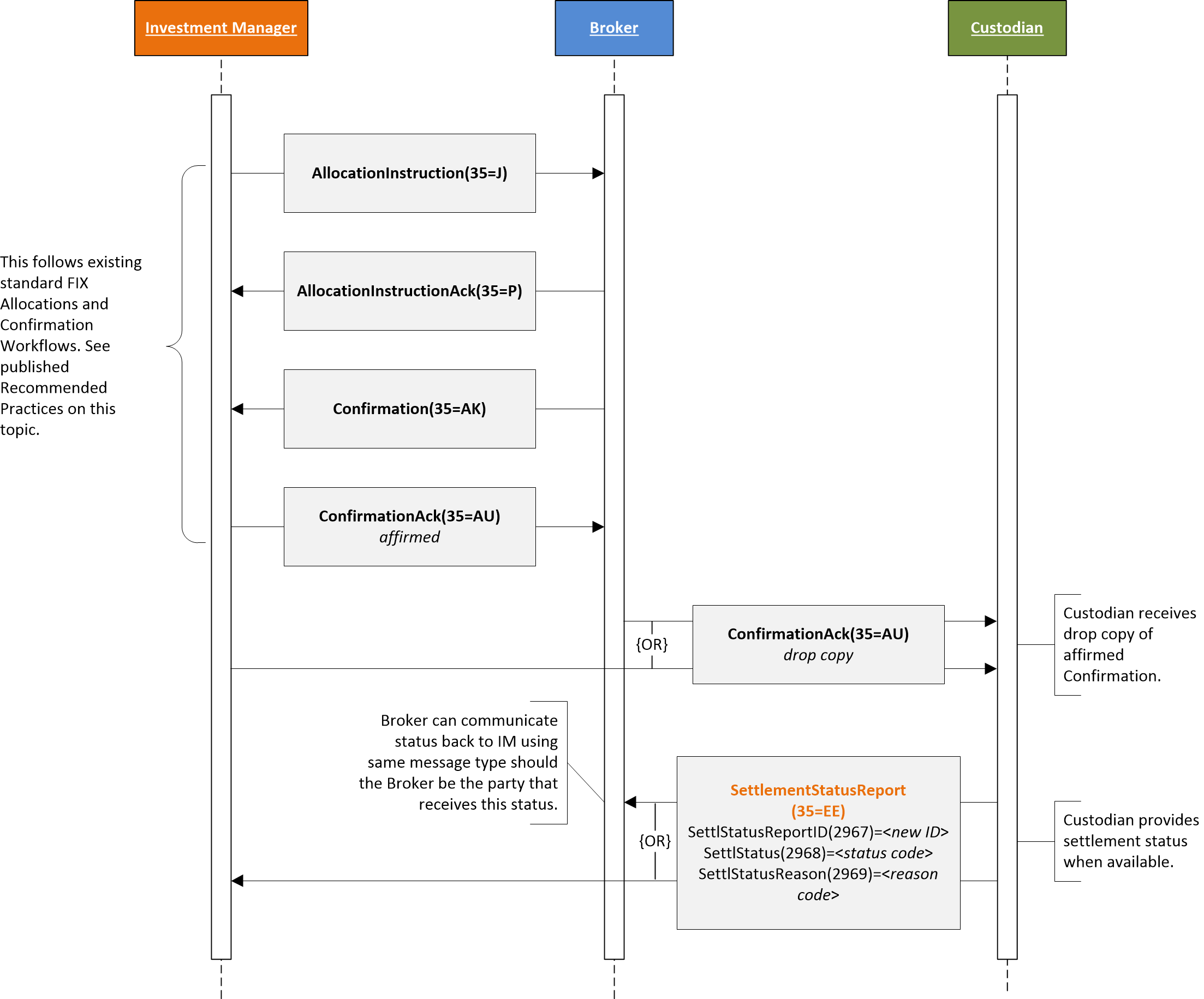
It is typical that investment managers will communicate a block trade's allocation instruction with their broker. Once the allocation to each account has been confirmed by the broker and affirmed (or acknowledged) by the investment manager it is deemed an affirmed trade (or acknowledged) for settlement purposes, i.e. the allocated amount to the specified account may proceed to settlement process. As an extension to this allocation/confirmation workflow, for investment managers who wish to receive updates to the settlement status of a given affirmed trade, a drop copy of the affirmed/acknowledged Confirmation(35=AK) message can sent to the party able to provide the settlement status for that trade - this step assumes the party providing settlement status don't already have confirmation information.

Figure 2 below illustrates this flow using the custodian as the party that is able to provide the settlement status to the investment manager. This role could be undertaken by any party such as a broker or outsourcer (e.g. vendor system) if they can provide the same information to the investment manager. In most cases, if the broker who provided the confirmation is the same party providing the settlement status, a drop copy may not be necessary. Upon receiving the drop-copy of the Confirmation(35=AK) message, settlement status updates is provided via the SettlementStatusReport(35=EE).

The settlement status should be sent to the party that provided the affirmed/acknowledged Confirmation(35=AK) drop-copy. For example, if the broker provided the custodian with the drop-copy of the Confirmation(35=AK) then the custodian provides the settlement status to the broker, and the broker may proceed to communicate that status to the investment manager using the same SettlementStatusReport(35=EE). The custodian may also directly communicate the status to the investment manager.

Depending on bilateral implementation agreement, under this scenario the investment manager may receive near real-time status updates without having to send a request for the status.

Figure 2: Settlement Status Notification



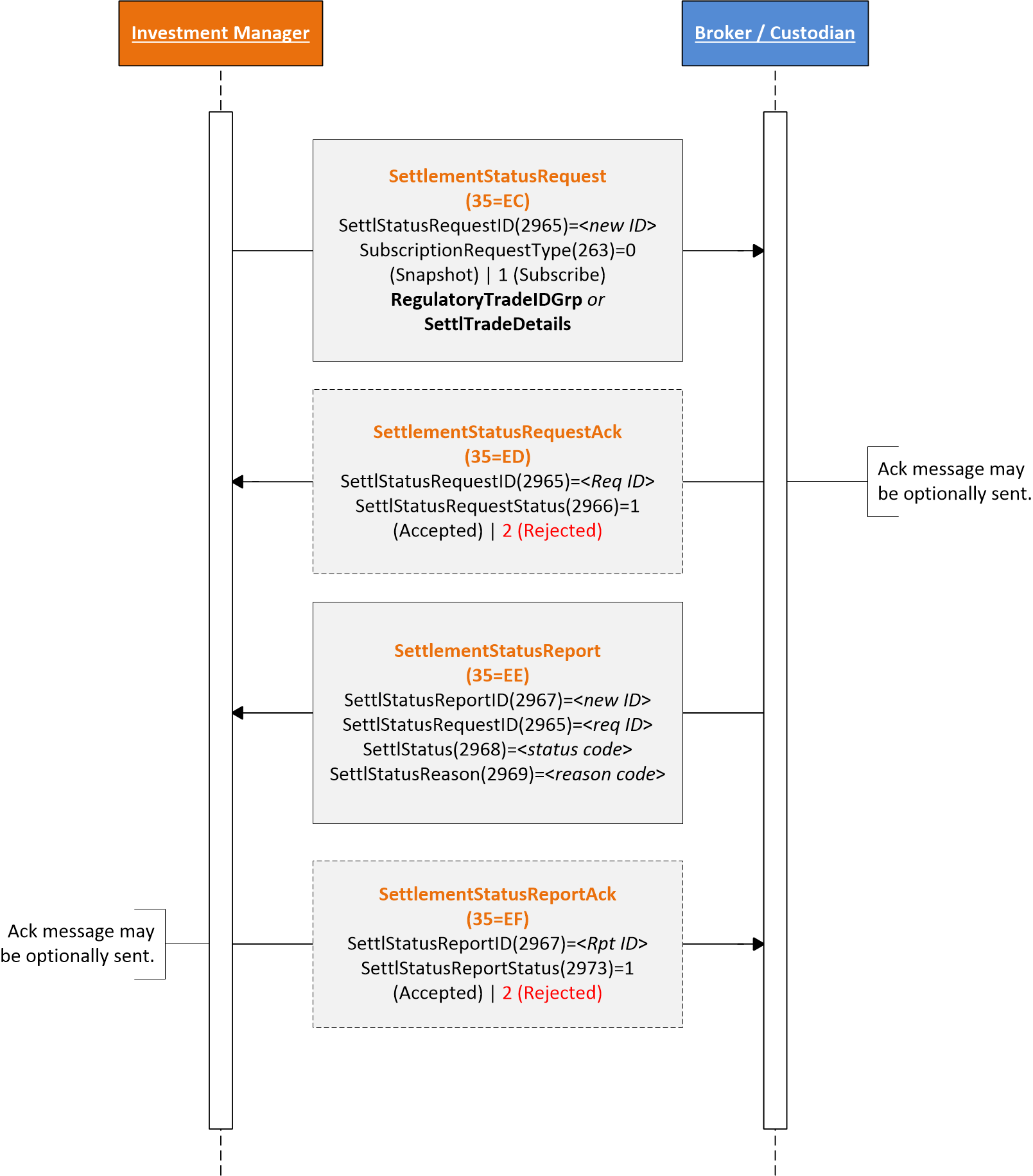
## Settlement Status Request and Report

This scenario encompasses an explicit request to receive settlement status information. A SettlementStatusRequest(35=EE) is sent to the party that is able to provide settlement status information. The request may either reference a Confirmation(35=AK) message known to the receiver of the request, or provide all the necessary transaction details.

The request may be a one-time request (i.e. SubscriptionRequestType(263)=0 (Snapshot) or a subscription (i.e. SubscriptionRequestType(263)=1 (Snapshot + update (subscribe))). As a "subscription" request, the receiver of the request responds back with the trade's settlement status as they become known in near real-time. As a "one-time" request, the receiver of the request responds back with the trade's settlement status currently known to the receiver.

Acknowledgement messages are optional.

Figure 3: Settlement Status Request and Status Report



## Usage of SettlStatusReason(2969) and SettlStatus(2968) fields

In implementations where the trade's settlement is initiated using SWIFT MT548 (or its equivalent MX message), the SettlementStatusReport(35=EE) contains two key fields that will carry the content from the following fields of the MT548:

* SettlStatus(2968) - This field contains the known status of the settlement. The content from MT548 mandatory sub-sequence A2 (Status) field 25D (Status code) should carry into this field.
* SettlStatusReason(2969) - This field further qualifies the reason for the status when it is needed. The content from MT548 optional subsequence A2a (Reason) field 24B (Reason code) should carry into this field.

In the MT548 message the field values may have a format of "aaaa/bbbb" in the mentioned fields above. The intent is for party that sends the SettlementStatusReport(35=EE) to send the value from the two fields as-is in the proposed FIX fields.

The list of 25D (Status code) possible values can be found here: <https://www.iso20022.org/15022/uhb/mt548-10-field-25d.htm>

The list of 24B (Reason code) possible values can eb found here: <https://www.iso20022.org/15022/uhb/mt548-12-field-24b.htm>

**Example 1:**

If the MT548 has the following in subsequence A2:

25D:MTCH

Then in the SettlementStatusReport(35=EE) the following would be sent in the SettlStatus(2968) field:

2968=MTCH

The example above states that the status of the settlement is "matched". As this state is pretty clear, further qualification is not needed.

**Example 2:**

If the MT548 has the following in subsequence A2 and A2a respectively:

25D:MTCH/NMAT

24B:NMAT/DTRD

70D:REAS/some text about why DTRD

Then in the SettlementStatusReport(35=EE) the above field content from the MT548 would be sent in the SettlStatus(2968), SettlStatusReason(2969) and optional SettlStatusReasonText(2970) fields **respectively** as follows:

2968=MTCH/NMAT

2969=NMAT/DTRD

2970="some text about why DTRD"

The example above states that the status of the settlement is "not matched" and the reason for the "not matched", in this example, is "DTRD" which means "disagreement trade date" (i.e. trade date does not match).

# Message and Component Tables

Introductory text applicable to all messages.

## Messages

### SettlementStatusRequest(35=EC)

The SettlementStatusRequest(35=EC) is sent by the party requesting settlement status of a specific trade. The trade can be identified using an identifier such as the UTI (specified in the RegulatoryTradeIDGrp component) or using trade details (specified in the SettlTradeDetails component). When the SettlTradeDetails component is used it is intended to serve as look-up criteria.

This message would typically be sent when the party receiving may not have received a drop-copy of the FIX Confirmation(35=AK) message or some other trade confirmation message. However, depending on bilaterally agreed implementation, the request may still be sent to initiate a "subscription" of status information for a trade.

**Response:**

The responses to this message are:

* SettlementStatusRequestAck(35=ED) to initially acknowledge, accept or reject the SettlementStatusRequest(35=EC) message itself
* After request has been accepted SettlementStatusReport(35=EE) is used to report back the trade's settlement status

Table 1: SettlementStatusRequest(35=EC)

| SettlementStatusRequest(35=EC) | | | |
| --- | --- | --- | --- |
| Tag | Name | Prescn | Description |
| **comp** | **StandardHeader** | Req | MsgType = EC |
| 2965 | SettlStatusRequestID | Req | Unique identifier assigned by the sender of this message. |
| 263 | SubscriptionRequestType | Req | Specifies whether the request is for a one-time (snapshot) status request or to receive status updates as status changes (subscription). |
| **comp** | **Parties** |  | May be used to identify the parties relevant to providing the settlement status, or additionally the parties of the trade when only the RegulatoryTradeIDGrp component is used to identify the trade. |
| **comp** | **RegulatoryTradeIDGrp** |  | May be used to specify the UTI (ISO 23897) of the trade this status request is for.  **Either RegulatoryTradeIDGrp or SettlTradeDetails component must be specified.** |
| **comp** | **SettlTradeDetails** |  | May be used to provide trade details to look-up the trade this settlement status request is for.  **Either RegulatoryTradeIDGrp or SettlTradeDetails must be present.**  The information in this component should closely carry over from the relevant parts of the AllocationInstruction(35=J) or from the Confirmation(35=AK) messages if FIX was used for the post-trade allocation/confirmation process.  If FIX was not used for post-trade allocation/confirmation, as much information should be provided to allow a "look up" of the trade. |
| 60 | TransactTime | Req | The time of the settlement status request message. |
| 58 | Text |  |  |
| 354 | EncodedTextLen |  |  |
| 355 | EncodedText |  |  |
| **comp** | **StandardTrailer** | Req |  |

### SettlementStatusRequestAck(35=ED)

The SettlementStatusRequsetAck(35=ED) is used to acknowledge, accept or reject the SettlementStatusRequest(35=EC) message. SettlStatusRequestStatus(2966)=1 (Accepted) means the request has been processed successfully and trade settlement status will be reported when available (using the SettlementStatusReport(35=EE). This message is **not** used to report on the trade's settlement status.

**Response:**

Should the recipient of this message have a need to reject this message (e.g. unknown SettlStatusRequestID(2965) value), the BusinessMessageReject(35=j) message shall be used.

Table 2: SettlementStatusRequestAck(35=ED)

| SettlementStatusRequestAck(35=ED) | | | |
| --- | --- | --- | --- |
| Tag | Name | Prescn | Description |
| **comp** | **StandardHeader** | Req | MsgType = EC |
| 2965 | SettlStatusRequestID | Req | Identifier of the SettlementStatusRequest(35=EC) being responded to. |
| 2966 | SettlStatusRequestStatus | Req | Status of the request message.  **Note** that this is not the settlement status of the trade.  0 = Received, not yet processed  1 = Accepted  2 = Rejected - Rejection reason provided in RejectText(1328) |
| 1328 | RejectText |  | Used to provide rejection reason when SettlStatusRequestStatus(2966)=2 (Rejected). |
| 1664 | EncodedRejectTextLen |  | Must be set if EncodedRejectText(1665) field is specified and must immediately precede it. |
| 1665 | EncodedRejectText |  | Encoded (non-ASCII characters) representation of the RejectText(1328) field in the encoded format specified via the MessageEncoding(347) field. |
| **comp** | **StandardTrailer** | Req |  |

### SettlementStatusReport(35=EE)

The SettlementStatusReport(35=EE) is sent by the party providing the settlement status of the trade to the party requesting information. This message may be sent as a result of an explicit request (SettlementStatusRequest(35=EC)) or unsolicited based on bilaterally agreed implementation.

**Response:**

The response to this message is the SettlementStatusReportAck(35=EF) to acknowledge, accept or reject the SettlementStatusReport(35=EE).

Table 3: SettlementStatusReport(35=EE)

| SettlementStatusReport(35=EE) | | | |
| --- | --- | --- | --- |
| Tag | Name | Prescn | Description |
| **comp** | **StandardHeader** | Req | MsgType = EC |
| 2967 | SettlStatusReportID | Req | Unique identifier assigned by the sender of this message. |
| 2965 | SettlStatusRequestID |  | Identifier of the SettlementStatusRequest(35=EC) this message is responding to. |
| 2968 | SettlStatus | Req | The current settlement status of the identified trade at the time of this message.  When reporting status based on MT548 this field should carry the status from MT548 sub-sequence A2a field 24B "Status code". |
| 2669 | SettlStatusReason |  | May be used to provide additional reason or qualify the reason for the settlement status specified in SettlStatus(2968).  When reporting status based on MT548 this field should carry the status reason from MT548 sub-sequence A2a field 24B "Reason code". |
| 2970 | SettlStatusReasonText |  | May be used to provide additional settlement status reason when available.  When reporting status based on MT548 this field should carry the status reason from MT548 sub-sequence A2a field 70D "Narrative text". |
| 2971 | EncodedSettlStatusReasonTextLen |  | Must be set if EncodedSettlStatusReasonText(2972) is specified and must immediately precede it. |
| 2972 | EncodedSettlStatusReasonText |  | Encoded (non-ASCII characters) representation of SettlStatusReasonText(2970) field in the encoded format specified via the MessageEncoding(347) field. |
| **comp** | **Parties** |  | In response to the SettlementStatusRequest(35=EC) may be used to echo back the information from the request message.  When this message is used unsolicted, this component may be used to identify the parties relevant to providing the settlement status, or additionally the parties of the trade when only the RegulatoryTradeIDGrp component is used to identify the trade. |
| **comp** | **RegulatoryTradeIDGrp** |  | In response to the SettlementStatusRequest(35=EC) may be used to echo back the information from the request message.  When this message is used unsolicited, this may be used to specify the UTI (ISO 23897) of the trade this settlement status is for.  **Either RegulatoryTradeIDGrp or SettlTradeDetails component must be specified.** |
| **comp** | **SettlTradeDetails** |  | In response to the SettlementStatusRequest(35=EC) may be used to echo back the information from the request message.  When this message is used unsolicited, this component specifies the trade details this settlement status for.  **Either RegulatoryTradeIDGrp or SettlTradeDetails must be present.** |
| 60 | TransactTime | Req | The time of the settlement status request message. |
| 58 | Text |  |  |
| 354 | EncodedTextLen |  |  |
| 355 | EncodedText |  |  |
| **comp** | **StandardTrailer** | Req |  |

### SettlementStatusReportAck(35=EF)

The SettlementStatusReportAck(35=EF) is used to acknowledge, accept or reject the SettlementStatusReport(35=EE) message.

**Response:**

Should the recipient of this message have a need to reject this message (e.g. unknown SettlStatusReportID(2967) value), the BusinessMessageReject(35=j) message shall be used.

Table 4: SettlementStatusReportAck(35=EF)

| SettlementStatusReportAck(35=EE) | | | |
| --- | --- | --- | --- |
| Tag | Name | Prescn | Description |
| **comp** | **StandardHeader** | Req | MsgType = EC |
| 2967 | SettlStatusReportID | Req | Identifier of the SettlementStatusReport(35=EE) being responded to. |
| 2973 | SettlStatusReportStatus | Req | Status of the report message.  0 = Received not yet processed  1 = Accepted  2 = Rejected - Rejection reason provided in RejectText(1328) |
| 1328 | RejectText |  | Used to provide rejection reason when SettlStatusReportStatus(2973)=2 (Rejected). |
| 1664 | EncodedRejectTextLen |  | Must be set if EncodedRejectText(1665) field is specified and must immediately precede it. |
| 1665 | EncodedRejectText |  | Encoded (non-ASCII characters) representation of the RejectText(1328) field in the encoded format specified via the MessageEncoding(347) field. If used, the ASCII (English) representation should also be specified in the RejectText(1328) field. |
| **comp** | **StandardTrailer** | Req |  |

Text followed by a table with the message layout.

## Components

### Parties

Table 5: Parties component

| <Parties> | | | |
| --- | --- | --- | --- |
| Tag | Name | Prescn | Description |
| 453 | NoPartyIDs |  |  |
| 🡪 448 | 🡪 PartyID |  | Required if NoPartyIDs(453) > 0. |
| 🡪 447 | 🡪 PartyIDSource |  | Required if NoPartyIDs(453) > 0.  *All standard enumerations of PartyIDSource(447).* |
| 🡪 452 | 🡪 PartyRole |  | Required if NoPartyIDs(453) > 0.  Relevant PartyRole(452) enumerations may include:  1 = Executing firm  4 = Clearing firm  11 = Order origination trader  12 = Executing trader  13 = Order origination firm  28 = Custodian |
| 🡪 **comp** | **🡪 PtysSubGrp** |  |  |

### RegulatoryTradeIDGrp

Table 6: RegulatoryTradeIDGrp component

| <RegulatoryTradeIDGrp> | | | |
| --- | --- | --- | --- |
| Tag | Name | Prescn | Description |
| 1907 | NoRegulatoryTradeIDs |  |  |
| 🡪 1903 | 🡪 RegulatoryTradeID |  | Required if NoRegulatoryTradeIDs(1907) > 0.  Contains the UTI or other trade identifier required for regulatory reporting purposes. |
| 🡪 1905 | 🡪 RegulatoryTradeIDSource |  | If RegulatoryTradeID(1903) contains a UTI (ISO 23897) value, this field shall contain the code value "1" (representing UTI), otherwise identifies the reporting entity that originated the value in RegulatoryTradeID(1903). |
| 🡪 1904 | 🡪 RegulatoryTradeIDEvent |  | Identifies the event that caused the origination of the RegulatoryTradeID(1903) value.  Relevant values:  0 = Initial block trade - Maybe applicable for single account allocation/confirmation  1 = Allocation |
| 🡪 1906 | 🡪 RegulatoryTradeIDType |  | Identifies the type of trade identifier in RegulatoryTradeID(1903).  Relevant values:  0 = Current - Default if not specified  5 = Trading venue transaction identifier |

### SettlTradeDetails

Table 7: SettlTradeDetails component

| <SettlTradeDetails> | | | |
| --- | --- | --- | --- |
| Tag | Name | Prescn | Description |
| **comp** | **SettlTradeDetails** |  | May be used to provide trade details to look-up the trade this settlement status request is for.  **Either RegulatoryTradeIDGrp or SettlTradeDetails must be present.**  The information in this component should closely carry over from the relevant parts of the AllocationInstruction(35=J) or from the Confirmation(35=AK) messages if FIX was used for the post-trade allocation/confirmation process.  If FIX was not used for post-trade allocation/confirmation, as much information should be provided to allow a "look up" of the trade. |
| 664 | ConfirmID |  | May be used to identify the trade via the known Confirmation(35=AK) message. |
| 70 | AllocID |  | May be used to identify the trade via the known AllocationInstruction(35=J) message. |
| 467 | IndividualAllocID |  | May be used to identify the trade via a specific allocated account instance of an AllocationInstruction(35=J) this IndividualAllocID(467) is part of. If specified AllocID(70) should be specified. |
| 793 | SecondaryIndividualAllocID |  | May be used to identify the trade via a specific allocated account instance of an AllocationInstruction(35=J) this SecondaryIndividualAllocID(793) is part of. If specified AllocID(70) should be specified. |
| 79 | AllocAccount |  | May be used to identify the account the trade was allocated to. |
| 75 | TradeDate |  | Date of the transaction. |
| **comp** | **Instrument** |  |  |
| 80 | AllocQty |  | Quantity of the trade. |
| 54 | Side |  | Side of the trade. |
| 6 | AvgPx |  | The price of the trade. |
| 423 | PriceType |  | Indicates the type of price in AvgPx(6).  *All standard enumerations of PriceType(423).* |
| 860 | AvgParPx |  | May be applicable for fixed income trades where AvgPx(6) is not percent-of-par price type. |
| 381 | GrossTradeAmt |  | The gross trade amount of the trade. |
| 118 | NetMoney |  | The net money of the trade. |
| 15 | Currency |  | Currency denomination use for prices and amounts of the trade. |
| 2897 | CurrencyCodeSource |  | The currency code source. By default, ISO 4217 Currency Code is used.  *All standard enumerations of CurrencyCodeSource(2897).* |
| 854 | QtyType |  | Type of quantity expressed in quantity fields. By default, market convention for the security is used, i.e. QtyType(854)=0 (Units - shares, par, currency).  *All standard enumerations of QtyType(854).* |
| **comp** | **NestedParties** |  | May be used to identify the parties to the trade when the SettlTradeDetails component is used instead of the RegulatoryTradeIDGrp component. |
| 64 | SettlDate |  | The settlement date of the trade. |
| 119 | SettlCurrAmt |  | The settlement amount when it is in a different currency. |
| 120 | SettlCurrency |  | Currency denomination of SettlCurrAmt(119) value. |
| 2899 | SettlCurrencyCodeSource |  | The currency code source. By default, ISO 4217 Currency Code is used.  *All standard enumerations of SettlCurrencyCodeSource(2899).* |
| **comp** | **SettlInstructionsData** |  | May be used to identify the trade's settlement instructions. |

### Instrument

Additional fields from the Instrument component should be included as bilaterally needed. The fields included here are the recommended minimum.

Table 8: Instrument component

| <Instrument> | | | |
| --- | --- | --- | --- |
| Tag | Name | Prescn | Description |
| 55 | Symbol |  | Common, "human understood" representation of the security. SecurityID(48) value can be specified if no symbol exists (e.g. non-exchange traded Collective Investment Vehicles). Use "[N/A]" for products which do not have a symbol. |
| 48 | SecurityID |  | Security identifier value of SecurityIDSource(22) type (e.g. CUSIP, SEDOL, ISIN, etc).  Requires SecurityIDSource(22). |
| 22 | SecurityIDSource |  | Identifies class or source of the SecurityID(48) value.  Requires SecurityID(48).  *All standard enumerations of SecurityIDSource(22).* |
| 167 | SecurityType |  | Indicates type of security. Security type enumerations are grouped by Product(460) field value.  *All standard enumerations of SecurityType(167).* |

### NestedParties

Table 9: NestedParties component

| <NestedParties> | | | |
| --- | --- | --- | --- |
| Tag | Name | Prescn | Description |
| 539 | NoNestedPartyIDs |  |  |
| 🡪 524 | 🡪 NestedPartyID |  | Required if NoNestedPartyIDs(539) > 0. |
| 🡪 525 | 🡪 NestedPartyIDSource |  | Required if NoNestedPartyIDs(539) > 0.  *All standard enumerations of NestedPartyIDSource(525).* |
| 🡪 538 | 🡪 NestedPartyRole |  | Required if NoPNestedartyIDs(539) > 0.  Relevant NestedPartyRole(452) enumerations may include:  1 = Executing firm  13 = Order origination firm |
| 🡪 **comp** | **🡪 NstdPtysSubGrp** |  |  |

### SettlInstructionsData

Table 10: SettlInstructionsData component

| <SettlInstructionsData> | | | |
| --- | --- | --- | --- |
| Tag | Name | Prescn | Description |
| 172 | SettlDeliveryType |  | Type of settlement/delivery.  *All standard enumerations of SettlDeliveryType(172).* |
| 169 | StandInstDbType |  | Identifies the standing settlement instruction databased used for SSI. Used with StandInstDbName(170) and StandInstDbID(171).  *All standard enumerations of SettlStandInstDbType(169).* |
| 170 | StandInstDbName |  | The SSI database name of the type indicated in StandInstDbType(169). |
| 171 | StandInstDbID |  | Identifier used in the SSI database to reference the SSI information for the trade. |
| **comp** | **DlvnInstGrp** |  | Used to identify the delivery instructions for the trade. Maybe used instead of identifying SSI information. |

### DlvnInstGrp

Table 11: DlvnInstGrp component

| <DlvnInstGrp> | | | |
| --- | --- | --- | --- |
| Tag | Name | Prescn | Description |
| 85 | NoDlvyInst |  |  |
| 🡪 165 | 🡪 SettlInstSource |  | Required if NoDlvyInst(85) > 0.  Source of the settlement instruction.  *All standard enumerations of SettlInstSource(165).* |
| 🡪 787 | 🡪 DlvyInstType |  | Indicates whether a delivery instruction is used for securities or cash settlement.  *All standard enumerations of DlvyInstType(787)* |
| **🡪 comp** | **SettlParties** |  | Used to identify the settlement parties for the trade. |

### SettlParties

Table 12: SettlParties component

| <SettlParties> | | | |
| --- | --- | --- | --- |
| Tag | Name | Prescn | Description |
| 781 | NoSettlPartyIDs |  |  |
| 🡪 782 | 🡪 SettlPartyID |  | Required if NoSettlPartyIDs(781) > 0. |
| 🡪 783 | 🡪 SettlPartyIDSource |  | Required if NoSettlPartyIDs(781) > 0.  *All standard enumerations of SettlPartyIDSource(783).* |
| 🡪 784 | 🡪 SettlPartyRole |  | Required if NoSettlPartyIDs(781) > 0.  Identifies role of SettlPartyID(782).  Relevant roles may include:  27 = Buyer/Seller  28 = Custodian  32 = Beneficiary |
| 🡪 2389 | 🡪 SettlPartyRoleQualifier |  | Used to further qualify the value of SettlPartyRole(784).  Relevant qualifiers may include:  7 = Bank - For use with party role 32 (Beneficiary)  24 = Natural person - For use with party role 32 (Beneficiary) |
| **🡪 comp** | **SettlPtysSubGrp** |  |  |

### SettlPtysSubGrp

Table 13: SettlPtysSubGrp component

| <SettlParties> | | | |
| --- | --- | --- | --- |
| Tag | Name | Prescn | Description |
| 801 | NoSettlPartySubIDs |  | Number of SettlPartySubID (785) entries |
| **🡪** 785 | **🡪** SettlPartySubID |  | Party sub-identifier value within a settlement parties component. |
| **🡪** 786 | **🡪** SettlPartySubIDType |  | Type of SettlPartySubID(785) value.  Relevant party sub-identifiers may include:  10 = Securities account number  15 = Cash account number  22 = Securities account name  23 = Cash account name |

A picture containing graphical user interface

Description automatically generated

Copyright © 2023 FIX Protocol Limited

**www.fixtrading.org**

Text followed by a table with the codes of the code set.