



# **FINANCIAL INFORMATION EXCHANGE (FIX)**

## **FIX APPLICATION LAYER**

### **Business Area: Infrastructure**

#### **FIX Latest**

*As of EP284, November 2023*

FIX Global Technical Committee

## **Table of Contents**

<b>Table of Contents</b> .....	<b>2</b>
<b>Table of Tables</b> .....	<b>4</b>
<b>Table of Figures</b> .....	<b>5</b>
<b>1 Introduction</b> .....	<b>7</b>
<b>2 List of Messages and Components for Infrastructure</b> .....	<b>8</b>
2.1 Messages .....	8
2.2 Components .....	8
<b>3 Category – Business Rejects</b> .....	<b>9</b>
3.1 Messages .....	9
3.1.1 Business Message Rejects .....	9
3.2 Components .....	9
3.3 Standard Responses for Pre-Trade Messages .....	9
3.4 Standard Responses for Trade Messages .....	11
3.5 Standard Responses for Post-Trade Messages .....	12
3.6 Key Fields for Application Message References .....	13
3.6.1 Pre-Trade .....	13
3.6.2 Trade .....	15
3.6.3 Post-Trade .....	15
3.7 Reject Codes for BusinessMessageReject(35=j) Message .....	17
<b>4 Category – Network Status Communication</b> .....	<b>18</b>
4.1 Messages .....	18
4.1.1 Network (Counterparty System) Status Requests .....	18
4.1.2 Network (Counterparty System) Status Responses .....	18
4.2 Components .....	19
4.2.1 CompIDReqGrp .....	19
4.2.2 CompIDStatGrp .....	19
<b>5 Category – User Management</b> .....	<b>20</b>
5.1 Messages .....	20
5.1.1 User Requests .....	20
5.1.2 User Responses .....	20
5.1.3 User Notifications .....	20
5.2 Components .....	21
5.2.1 ThrottleMsgTypeGrp .....	21
5.2.2 ThrottleParamsGrp .....	21
5.2.3 UsernameGrp .....	21
<b>6 Category – Application Sequencing</b> .....	<b>22</b>
6.1 Introduction .....	22
6.2 Background .....	22
6.2.1 Extends control over resent data .....	22
6.2.2 Support for secondary data distribution .....	22
6.3 Transaction usage is not recommended .....	23
6.4 Using Application Sequencing and Session Sequencing for Gap Detection .....	23

6.5	Messages.....	23
6.5.1	Application Message Requests .....	23
6.5.2	Application Message Request Acknowledgements.....	24
6.5.3	Application Message Reports .....	24
6.5.3.1	Using Application Message Reports to reset application-level sequence number.....	24
6.5.3.2	Using Application Message Reports to indicate last message sent.....	24
6.5.3.3	Using Application Message Report as keep-alive mechanism .....	24
6.5.3.4	Using Application Message Report to indicate completion of resent messages .....	24
6.6	Components .....	25
6.6.1	ApplIDReportGrp .....	25
6.6.2	ApplIDRequestAckGrp .....	25
6.6.3	ApplIDRequestGrp .....	25
<b>7</b>	<b>Appendix – Application Category .....</b>	<b>26</b>
7.1	Messages.....	26
7.1.1	ApplicationMessageRequest Message .....	26
7.1.2	ApplicationMessageRequestAck Message .....	26
7.1.3	ApplicationMessageReport Message .....	26
7.2	Components .....	27
7.2.1	ApplIDReportGrp .....	27
7.2.2	ApplIDRequestAckGrp .....	27
7.2.3	ApplIDRequestGrp .....	27
<b>8</b>	<b>Appendix – BusinessReject Category .....</b>	<b>28</b>
8.1	Messages.....	28
8.1.1	BusinessMessageReject Message.....	28
<b>9</b>	<b>Appendix – Network Category .....</b>	<b>29</b>
9.1	Messages.....	29
9.1.1	NetworkCounterpartySystemStatusRequest Message .....	29
9.1.2	NetworkCounterpartySystemStatusResponse Message .....	29
9.2	Components .....	29
9.2.1	CompIDReqGrp .....	29
9.2.2	CompIDStatGrp .....	30
<b>10</b>	<b>Appendix – UserManagement Category .....</b>	<b>31</b>
10.1	Messages.....	31
10.1.1	UserRequest Message .....	31
10.1.2	UserResponse Message .....	31
10.1.3	UserNotification Message.....	31
10.2	Components .....	32
10.2.1	ThrottleMsgTypeGrp.....	32
10.2.2	ThrottleParamsGrp .....	32
10.2.3	UsernameGrp.....	32

**Table of Tables**

Table 1: Messages for Infrastructure Business Area.....	8
Table 2: Components for Infrastructure Business Area.....	8
Table 3: Standard Responses for Pre-Trade Messages.....	9
Table 4: Standard Responses for Trade Messages.....	11
Table 5: Standard Responses for Post-Trade Messages.....	12
Table 6: Key Fields for Pre-Trade Application Message References.....	13
Table 7: Key Fields for Trade Application Message References.....	15
Table 8: Key Fields for Post-Trade Application Message References.....	15
Table 9: Reject Codes for BusinessMessageReject(35=j) Message.....	17

## **Table of Figures**

Figure 1: Message Diagram Templates .....	7
Figure 2: Message BusinessMessageReject(35=j) .....	9
Figure 3: Message NetworkCounterpartySystemStatusRequest(35=BC) .....	18
Figure 4: Message NetworkCounterpartySystemStatusResponse(35=BD) .....	18
Figure 5: Message UserRequest(35=BE) .....	20
Figure 6: Message UserResponse(35=BF) .....	20
Figure 7: Message UserNotification(35=CB) .....	20
Figure 8: Message ApplicationMessageRequest(35=BW) .....	23
Figure 9: Message ApplicationMessageRequestAck(35=BX) .....	24
Figure 10: Message ApplicationMessageReport(35=BY) .....	24

## 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.

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 Application Layer Specifications by [FIX Protocol Ltd.](#) are licensed under a [Creative Commons Attribution-NonDerivatives 4.0 International License](#). Based on a work at <https://github.com/FIXTradingCommunity/>.

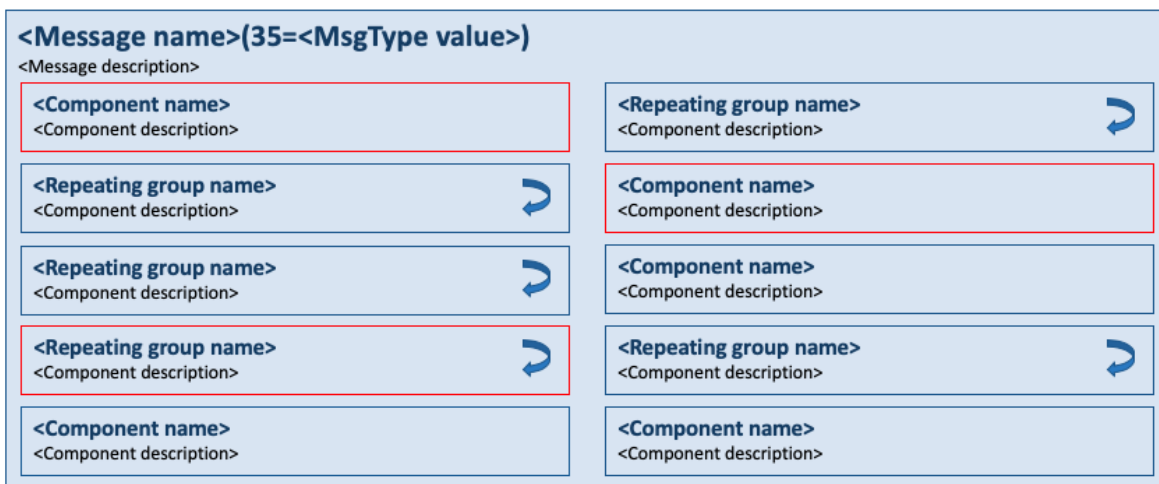
# 1 Introduction

Infrastructure messaging is characterized as messages which are common to the business areas pre-trade, trade and post-trade.

The specific FIX infrastructure messaging categories are:

1. [BUSINESS REJECTS](#)
2. [NETWORK STATUS COMMUNICATION](#)
3. [USER MANAGEMENT](#)
4. [APPLICATION SEQUENCING](#)

Descriptions of the specific FIX infrastructure application messages follow. There is a diagram for each of the messages depicting its components. Required components are shown with a red outline and repeating groups contain an arrow symbol. Some messages do not have any components. The detailed layout of all messages and components is provided in the [appendix](#).



Note: Components in red boxes are required.

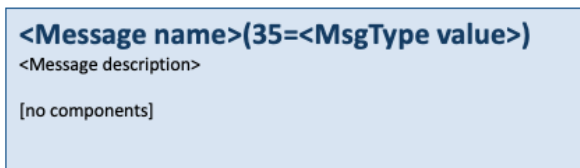


Figure 1: Message Diagram Templates

## 2 List of Messages and Components for Infrastructure

### 2.1 Messages

This section lists the infrastructure messages and the category each of them belongs to.

Table 1: Messages for Infrastructure Business Area

MsgType(35)	Name	Category
BY	<a href="#">ApplicationMessageReport</a>	<a href="#">Application Sequencing</a>
BW	<a href="#">ApplicationMessageRequest</a>	<a href="#">Application Sequencing</a>
BX	<a href="#">ApplicationMessageRequestAck</a>	<a href="#">Application Sequencing</a>
j	<a href="#">BusinessMessageReject</a>	<a href="#">Business Message Rejects</a>
BC	<a href="#">NetworkCounterpartySystemStatusRequest</a>	<a href="#">Network Status Communication</a>
BD	<a href="#">NetworkCounterpartySystemStatusResponse</a>	<a href="#">Network Status Communication</a>
BE	<a href="#">UserRequest</a>	<a href="#">User Management</a>
BF	<a href="#">UserResponse</a>	<a href="#">User Management</a>
CB	<a href="#">UserNotification</a>	<a href="#">User Management</a>

### 2.2 Components

This section lists components used by infrastructure messages defined in this part of the FIX specification. None of these are *Common Components* used by more than one category in this area. Messages may also reference *Global Components*, which are components used by messages across more than one area. *Global Components* are defined in the overall [Introduction](#) to the FIX specification.

Components can be either non-repeating or repeating (a.k.a. a “group”), i.e. contain multiple instances of a set of fields. Components can be nested to any level.

Table 2: Components for Infrastructure Business Area

Type	Name	Category
Repeating	<a href="#">ApplIDReportGrp</a>	<a href="#">Application Sequencing</a>
Repeating	<a href="#">ApplIDRequestAckGrp</a>	<a href="#">Application Sequencing</a>
Repeating	<a href="#">ApplIDRequestGrp</a>	<a href="#">Application Sequencing</a>
Repeating	<a href="#">CompIDReqGrp</a>	<a href="#">Network Status Communication</a> <sup>1</sup>
Repeating	<a href="#">CompIDStatGrp</a>	<a href="#">Network Status Communication</a> <sup>2</sup>
Repeating	<a href="#">ThrottleMsgTypeGrp</a>	<a href="#">User Management</a> <sup>3</sup>
Repeating	<a href="#">ThrottleParamsGrp</a>	<a href="#">User Management</a> <sup>4</sup>
Repeating	<a href="#">UsernameGrp</a>	<a href="#">User Management</a>

<sup>1</sup> CompIDReqGrp added as common with FIX 4.4 but only used in the category *Network Status Communication*.

<sup>2</sup> CompIDStatGrp added as common with FIX 4.4 but only used in the category *Network Status Communication*.

<sup>3</sup> ThrottleMsgTypeGrp added as common with EP116 but only used in the category *User Management*.

<sup>4</sup> ThrottleParamsGrp added as common with EP116 but only used in the category *User Management*.



## 3 Category – Business Rejects

### 3.1 Messages

#### 3.1.1 Business Message Rejects

<p><b>BusinessMessageReject(35=j)</b>          Rejection of an application message if no specific message applies</p> <p>[no components]</p>
--

Figure 2: Message BusinessMessageReject(35=j)

The BusinessMessageReject(35=j) message can reject an application-level message which fulfills session-level rules and cannot be rejected via any other means. Note if the message fails a session-level rule (e.g. body length is incorrect), a session-level Reject(35=3) message should be issued.

It should **NOT** be used whenever there is a session-level problem. In this case, use the session-level Reject(35=3) message if the FIX Session Protocol is being used. If the FIX Session Protocol is not being used, use an appropriate reject message of the given session protocol or the BusinessMessageReject(35=j) message may be used if no appropriate session protocol reject message is available. The message layout is available [here](#).

It should also **NOT** be used in response to application-level messages that have a dedicated response message as detailed in the following sections.

**Note the only exceptions to the appropriate responses defined below are:**

1. **in the event a business message is received, fulfills session-level rules, however, the message cannot be communicated to the business-level processing system.** In this situation a BusinessMessageReject(35=j) with BusinessRejectReason(380) = 4 (Application not available) can be issued if the system is unable to send the specific “reject” message listed above due to this condition.
2. **in the event a valid business message is received, fulfills session-level rules, however, the message type is not supported by the recipient.** In this situation a BusinessMessageReject(35=j) with BusinessRejectReason(380) = 3 (Unsupported Message Type) can be issued if the system is unable to send the specific “reject” message listed above because the receiving system cannot generate the related “reject” message.
3. **In the event a business message is received, fulfills session-level rules, but lacks a field conditionally required by the FIX specification.** In this situation a BusinessMessageReject(35=j) with BusinessRejectReason(380) = 5 (Conditionally required field missing) can be issued if the system is unable to send the specific “reject” message listed above. One example of this would be a stop order missing StopPx(99). However, a BusinessMessageReject(35=j) message **MUST NOT** be used to enforce proprietary rules more restrictive than those explicit in the FIX specification, such as a broker requiring an order to contain an Account(1), which the FIX specification considers an optional field.

### 3.2 Components

This category does not have any components.

### 3.3 Standard Responses for Pre-Trade Messages

Table 3: Standard Responses for Pre-Trade Messages

Category	FIX Message	Appropriate Response(s)
Indication	CrossRequest(35=DS)	CrossRequestAck(35=DT)

Category	FIX Message	Appropriate Response(s)
Quotation Negotiation	MassQuote(35=i)	MassQuoteAck(35=b)
Quotation Negotiation	QuoteRequest(35=R)	QuoteRequestReject(35=AG)
Quotation Negotiation	Quote(35=S)	Depending on workflow implemented: QuoteAck(35=CW) or QuoteStatusReport(35=AI)
Quotation Negotiation	QuoteCancel(35=Z)	Depending on workflow implemented: QuoteAck(35=CW) or QuoteStatusReport(35=AI)
Quotation Negotiation	QuoteStatusRequest(35=a)	QuoteStatusReport(35=AI)
Quotation Negotiation	QuoteResponse(35=AJ)	QuoteStatusReport(35=AI)
Market Data	MarketDataRequest(35=V)	MarketDataRequestReject(35=Y)
Market Data	MarketDataStatisticsRequest(35=DO)	MarketDataStatisticsReport(35=DP)
Market Data	StreamAssignmentReport(35=CD)	StreamAssignmentReportACK(35=CE)
Market Data	StreamAssignmentRequest(35=CC)	StreamAssignmentReport(35=CD)
Securities Reference Data	DerivativeSecurityListRequest(35=z)	DerivativeSecurityList(35=AA)
Securities Reference Data	SecurityDefinitionRequest(35=c)	SecurityDefinition(35=d)
Securities Reference Data	SecurityListRequest(35=x)	SecurityList(35=y)
Securities Reference Data	SecurityMassStatusRequest(35=CN)	SecurityMassStatus(35=CO)
Securities Reference Data	SecurityStatusRequest(35=e)	SecurityStatus(35=f)
Securities Reference Data	SecurityTypeRequest(35=v)	SecurityTypes(35=w)
Market Structure Reference Data	TradingSessionStatusRequest(35=g)	TradingSessionStatus(35=h)
Market Structure Reference Data	TradingSessionListRequest(35=BI)	TradingSessionList(35=BJ)
Parties Reference Data	PartyActionRequest(35=DH)	PartyActionReport(35=DI)

Category	FIX Message	Appropriate Response(s)
Parties Reference Data	PartyDetailsDefinitionRequest(35=CX)	PartyDetailsDefinitionRequestAck(35=CY)
Parties Reference Data	PartyDetailsListRequest(35=CF)	PartyDetailsListReport(35=CG)
Parties Reference Data	PartyEntitlementsDefinitionRequest(35=DA)	PartyEntitlementsDefinitionRequestAck(35=DB)
Parties Reference Data	PartyEntitlementsRequest(35=CU)	PartyEntitlementsReport(35=CV)
Parties Reference Data	PartyRiskLimitCheckRequest(35=DF)	PartyRiskLimitCheckRequestAck(35=DG)
Parties Reference Data	PartyRiskLimitsDefinitionRequest(35=CS)	PartyRiskLimitsDefinitionRequestAck(35=CT)
Parties Reference Data	PartyRiskLimitsReport(35=CM)	PartyRiskLimitsReportAck(35=DE)
Parties Reference Data	PartyRiskLimitsRequest(35=CL)	PartyRiskLimitsReport(35=CM)
Parties Reference Data	PartyRiskLimitsUpdateReport(35=CR)	PartyRiskLimitsReportAck(35=DE)

### 3.4 Standard Responses for Trade Messages

Table 4: Standard Responses for Trade Messages

Category	FIX Message	Appropriate Response(s)
Single General Order Handling	ExecutionReport(35=8)	DontKnowTrade(35=Q) or ExecutionAck(35=BN)
Single General Order Handling	NewOrderSingle(35=D)	ExecutionReport(35=8)
Single General Order Handling	OrderCancelReplaceRequest(35=G)	OrderCancelReject(35=9)
Single General Order Handling	OrderCancelRequest(35=F)	OrderCancelReject(35=9)
Single General Order Handling	OrderStatusRequest(35=H)	ExecutionReport(35=8)
Multileg Order Handling	MultilegOrderCancelReplace(35=AC)	OrderCancelReject(35=9)
Multileg Order Handling	NewOrderMultileg(35=AB)	ExecutionReport(35=8)
Cross Order Handling	CrossOrderCancelReplaceRequest(35=t)	OrderCancelReject(35=9)

Category	FIX Message	Appropriate Response(s)
Cross Order Handling	CrossOrderCancelRequest(35=u)	OrderCancelReject(35=9)
Cross Order Handling	NewOrderCross(35=s)	ExecutionReport(35=8)
Order Mass Handling	MassOrder(35=DJ)	MassOrderAck(35=DK)
Order Mass Handling	OrderMassActionRequest(35=CA)	OrderMassActionReport(35=BZ)
Order Mass Handling	OrderMassCancelRequest(35=q)	OrderMassCancelReport(35=r)
Order Mass Handling	OrderMassStatusRequest(35=AF)	ExecutionReport(35=8)
Program Trading	BidRequest(35=k)	BidResponse(35=l)
Program Trading	ListCancelRequest(35=K)	OrderCancelReject(35=9)
Program Trading	ListExecute(35=L)	ExecutionReport(35=8)
Program Trading	ListStatusRequest(35=M)	ListStatus(35=N)
Program Trading	NewOrderList(35=E)	ExecutionReport(35=8)

### 3.5 Standard Responses for Post-Trade Messages

Table 5: Standard Responses for Post-Trade Messages

Category	FIX Message	Appropriate Response(s)
Allocation	AllocationInstruction(35=J)	AllocationInstructionAck(35=P)
Allocation	AllocationInstructionAlertRequest(35=DU)	AllocationInstructionAlertRequestAck(35=DV)
Allocation	AllocationReport(35=AS)	AllocationReportAck(35=AT)
Confirmation	Confirmation(35=AK)	ConfirmationAck(35=AU)
Confirmation	ConfirmationRequest(35=BH)	Confirmation(35=AK)
Trade Capture	TradeCaptureReportRequest(35=AD)	TradeCaptureReportRequestAck(35=AQ)
Trade Capture	TradeCaptureReport(35=AE)	TradeCaptureReportAck(35=AR)
Trade Capture	TradeMatchReport(35=DC)	TradeMatchReportAck(35=DD)
Trade Management	TradeAggregationRequest(35=DW)	TradeAggregationReport(35=DX)
Position Maintenance	PositionMaintenanceRequest(35=AL)	PositionMaintenanceReport(35=AM)
Position Maintenance	PositionTransferInstruction(35=DL)	PositionTransferInstructionAck(35=DM)
Position Maintenance	RequestForPositions(35=AN)	RequestForPositionsAck(35=AO)
Collateral Management	CollateralAssignment(35=AY)	CollateralResponse(35=AZ)
Collateral Management	CollateralInquiry(35=BB)	CollateralInquiryAck(35=BG)
Collateral Management	CollateralReport(35=BA)	CollateralReportAck(35=DQ)
Collateral Management	CollateralRequest(35=AX)	CollateralAssignment(35=AY)

Category	FIX Message	Appropriate Response(s)
Margin Requirement Management	MarginRequirementInquiry(35=CH)	Depending on workflow implemented: MarginRequirementInquiryAck(35=CI) or MarginRequirementReport(35=CJ), PositionReport(35=AP)
Registration Instruction	RegistrationInstructions(35=o)	RegistrationInstructionsResponse(35=p)
Settlement Instruction	SettlementInstructionRequest(35=AV)	SettlementInstructions(35=T)
Pay Management	PayManagementReport(35=EA)	PayManagementReportAck(35=EB)
Pay Management	PayManagementRequest(35=DY)	Depending on workflow implemented: PayManagementRequestAck(35=DZ) or PayManagementReport(35=EA)
Settlement Status Management	SettlementStatusRequest(35=EC)	Depending on workflow implemented: SettlementStatusRequestAck(35=ED) or SettlementStatusReport(35=EE)
Settlement Status Management	SettlementStatusReport(35=EE)	SettlementStatusReportAck(35=EF)

### 3.6 Key Fields for Application Message References

Messages which do not have a standard response message and can only be referenced via the BusinessMessageReject(35=j) message are as follows:

#### 3.6.1 Pre-Trade

Table 6: Key Fields for Pre-Trade Application Message References

Category	FIX Message	BusinessRejectRefID(379)
Indication	IOI(35=6)	IOIID(23)
Indication	Advertisement(35=7)	AdvId(2)
Indication	CrossRequestAck(35=DT)	CrossRequestID(2672)
Event Communication	News(35=B)	Headline(148) or NewsID(1472)
Event Communication	Email(35=C)	EmailThreadID(164)
Quotation Negotiation	MassQuoteAck(35=b)	QuoteReqID(131) or QuoteID(117)
Quotation Negotiation	QuoteAck(35=CW)	QuoteReqID(131) or QuoteID(117) or QuoteMsgID(1166)
Quotation Negotiation	QuoteRequestReject(35=R)	QuoteReqID(131)
Quotation Negotiation	QuoteStatusReport(35=AI)	QuoteStatusReqID(649) or QuoteRespID(693) or QuoteID(117) or QuoteMsgID(1166)

Category	FIX Message	BusinessRejectRefID(379)
Quotation Negotiation	RFQRequest(35=AH)	RFQReqID(644)
Market Data	MarketDataIncrementalRefresh(35=X)	MDReqID(262)
Market Data	MarketDataReport(35=DR)	MDReportID(963)
Market Data	MarketDataSnapshotFullRefresh(35=W)	MDReqID(262)
Market Data	MarketDataStatisticsReport(35=DP)	MDStatisticRptID(2453) or MDStatisticReqID(2452)
Market Data	StreamAssignmentReportACK(35=CE)	StreamAsgnRptID(1501)
Securities Reference Data	DerivativeSecurityList(35=AA)	SecurityResponseID(322)
Securities Reference Data	SecurityDefinition(35=d)	SecurityResponseID(322) or SecurityReportID(964)
Securities Reference Data	SecurityDefinitionUpdateReport(35=BP)	SecurityResponseID(322) or SecurityReportID(964)
Securities Reference Data	SecurityList(35=y)	SecurityResponseID(322)
Securities Reference Data	SecurityListUpdateReport(35=BK)	SecurityResponseID(322) or SecurityReportID(964)
Securities Reference Data	SecurityMassStatus(35=CO)	SecurityStatusReqID(324)
Securities Reference Data	SecurityStatus(35=f)	SecurityStatusReqID(324)
Securities Reference Data	SecurityTypes(35=w)	SecurityResponseID(322)
Securities Reference Data	DerivativeSecurityListUpdateReport(35=BR)	SecurityResponseID(322)
Market Structure Reference Data	MarketDefinition(35=BU)	MarketReportID(1394)
Market Structure Reference Data	MarketDefinitionRequest(35=BT)	MarketReqID(1393)
Market Structure Reference Data	MarketDefinitionUpdateReport(35=BV)	MarketReportID(1394) or MarketReqID(1393)
Market Structure Reference Data	TradingSessionList(35=BJ)	TradSesReqID(335)
Market Structure Reference Data	TradingSessionListUpdateReport(35=BS)	TradSesReqID(335)
Market Structure Reference Data	TradingSessionStatus(35=h)	TradSesReqID(335)
Parties Reference Data	PartyDetailsDefinitionRequestAck(35=CY)	PartyDetailsListRequestID(1505)
Parties Reference Data	PartyDetailsListReport(35=CG)	PartyDetailsListReportID(1510) or PartyDetailsListRequestID(1505)

Category	FIX Message	BusinessRejectRefID(379)
Parties Reference Data	PartyDetailsListUpdateReport(35=CK)	PartyDetailsListReportID(1510) or PartyDetailsListRequestID(1505)
Parties Reference Data	PartyEntitlementsReport(35=CV)	EntitlementReportID(1771) or EntitlementRequestID(1770)
Parties Reference Data	PartyEntitlementsUpdateReport(35=CZ)	EntitlementReportID(1771) or EntitlementRequestID(1770)
Parties Reference Data	PartyRiskLimitsDefinitionRequestAck(35=CT)	RiskLimitRequestID(1666)
Parties Reference Data	PartyRiskLimitsReportAck(35=DE)	RiskLimitReportID(1667)
Parties Action	PartyActionReport(35=DI)	PartyActionReportID(2331) or PartyActionRequestID(2328)
Parties Action	PartyRiskLimitCheckRequestAck(35=DG)	RiskLimitCheckRequestID(2318)

### 3.6.2 Trade

Table 7: Key Fields for Trade Application Message References

Category	FIX Message	BusinessRejectRefID(379)
Single General Order Handling	DontKnowTrade(35=Q)	ExecID(17) <sup>5</sup>
Single General Order Handling	ExecutionAck(35=BN)	ExecID(17)
Single General Order Handling	OrderCancelReject(35=9)	ClOrdID(11)
Single General Order Handling	OrderMassCancelReport(35=r)	ClOrdID(11)
Order Mass Handling	MassOrderAck(35=DK)	MassOrderReportID(2424)
Order Mass Handling	OrderMassActionReport(35=BZ)	MassActionReportID(1369) or ClOrdID(11)
Program Trading	ListStatus(35=N)	ListID(66)
Program Trading	ListStrikePrice(35=m)	ListID(66)
Program Trading	BidResponse(35=l)	BidID(390)

### 3.6.3 Post-Trade

Table 8: Key Fields for Post-Trade Application Message References

Category	FIX Message	BusinessRejectRefID(379)
Account Reporting	AccountSummaryReport(35=CQ)	AccountSummaryReportID(1699)
Allocation	AllocationInstructionAck(35=P)	AllocID(70)
Allocation	AllocationReportAck(35=AT)	AllocID(70)
Allocation	AllocationInstructionAlert(35=BM)	AllocID(70)
Allocation	AllocationInstructionAlertRequestAck(35=DV)	AllocRequestID(2758)
Confirmation	ConfirmationAck(35=AU)	ConfirmID(664)
Trade Capture	TradeCaptureReportRequestAck(35=AQ)	TradeRequestID(568)

<sup>5</sup> May also respond with OrderCancelReject(35=9) message if attempting to cancel order identified with ExecID(17).

Category	FIX Message	BusinessRejectRefID(379)
Trade Capture	TradeCaptureReportAck(35=AR)	TradeReportID(571)
Trade Capture	TradeMatchReportAck(35=DD)	TradeMatchID(880)
Trade Management	TradeAggregationReport(35=DX)	TradeAggregationReportID(2792) or TradeAggregationRequestID(2786)
Position Maintenance	AdjustedPositionReport(35=BL)	PosMaintRptID(721)
Position Maintenance	AssignmentReport(35=AW)	AsgnRptID(833)
Position Maintenance	ContraryIntentionReport(35=BO)	ContIntRptID(977)
Position Maintenance	PositionMaintenanceReport(35=AM)	PosMaintRptID(721) or PosReqID(710)
Position Maintenance	PositionReport(35=AP)	PosMaintRptID(721) or PosReqID(710)
Position Maintenance	PositionTransferInstructionAck(35=DM)	TransferInstructionID(2436)
Position Maintenance	PositionTransferReport(35=DN)	TransferReportID(2438) or TransferInstructionID(2436)
Position Maintenance	RequestForPositionsAck(35=AO)	PosMaintRptID(721)
Collateral Management	CollateralResponse(35=AZ)	CollRespID(904)
Collateral Management	CollateralInquiryAck(35=BG)	CollInquiryID(909)
Collateral Management	CollateralReportAck(35=DQ)	CollRptID(908)
Margin Requirement Management	MarginRequirementInquiryAck(35=CI)	MarginReqmtInqID(1635)
Margin Requirement Management	MarginRequirementReport(35=CJ)	MarginReqmtRptID(1642)
Registration Instruction	RegistrationInstructionsResponse(35=p)	RegistID(513)
Settlement Instruction	SettlementInstructions(35=T)	SettInstMsgID(777)
Settlement Instruction	SettlementObligationReport(35=BQ)	SettObligMsgID(1160)
Pay Management	PayManagementReportAck(35=EB)	PayReportID(2799)
Pay Management	PayManagementRequestAck(35=DZ)	PayRequestID(2812)
Settlement Status Management	SettlementStatusReportAck(35=EF)	SettIStatusReportID(2967)



Category	FIX Message	BusinessRejectRefID(379)
Settlement Status Management	SettlementStatusRequestAck(35=ED)	SettlStatusRequestID(2965)

### 3.7 Reject Codes for BusinessMessageReject(35=j) Message

Table 9: Reject Codes for BusinessMessageReject(35=j) Message

BusinessRejectReason(380)
0 = Other
1 = Unknown ID
2 = Unknown Security
3 = Unsupported Message Type (receive a valid, but unsupported MsgType)
4 = Application not available
5 = Conditionally Required Field Missing
6 = Not Authorised
7 = DeliverTo firm not available at this time
8 = Throttle limit exceeded
9 = Throttle limit exceeded, session will be disconnected
10 = Throttled messages rejected on request
18 = Invalid price increment

Whenever possible, it is strongly recommended that the cause of the failure be described in Text(58) (e.g. “UNKNOWN SYBMOL: XYZ”).

## 4 Category – Network Status Communication

The Network Status messages are intended to be used by network services that facilitate FIX connectivity, allowing the service provider to convey to customers the FIX connection status of their counterparty(-ies).

It is envisaged these messages will be used in two scenarios:

### Scenario A

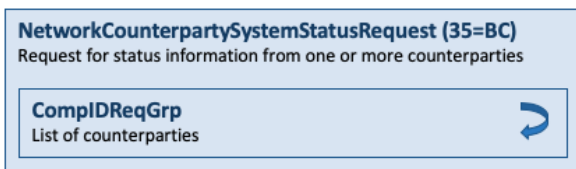
Allow one counterparty using a “hub and spoke” FIX network to know whether another counterparty is currently connected to the hub (i.e. whether the counterparty’s session to the hub is up or not).

### Scenario B

Allow a counterparty connecting to a global brokerage to know which regions within that brokerage are currently available as order routing destinations.

## 4.1 Messages

### 4.1.1 Network (Counterparty System) Status Requests

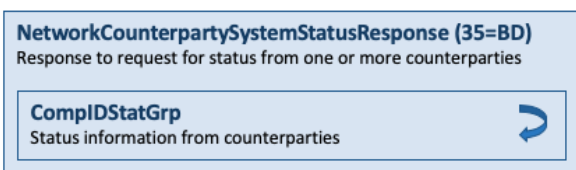


Note: Components in red boxes are required.

Figure 3: Message `NetworkCounterpartySystemStatusRequest(35=BC)`

The `NetworkCounterpartySystemStatusRequest(35=BC)` message is sent either immediately after logging on to inform a network (counterparty system) of the type of updates required or to, at any other time in the FIX conversation, to change the nature of the types of status updates required. It can also be used with `NetworkRequestType(935) = 1` (Snapshot) to request a one-off report of the status of a network (or counterparty) system. Finally this message can also be used to cancel a request to receive updates into the status of the counterparties on a network by sending a `NetworkCounterpartySystemRequestStatusMessage(35=BC)` message with `NetworkRequestType = 4` (Stop Subscribing). The message layout is available [here](#).

### 4.1.2 Network (Counterparty System) Status Responses



Note: Components in red boxes are required.

Figure 4: Message `NetworkCounterpartySystemStatusResponse(35=BD)`

The `NetworkCounterpartySystemStatusResponse(35=BD)` message is sent in response to a `NetworkCounterpartySystemStatusRequest(35=BC)` message with a list of counterparties and their status in the repeating group `CompIDStatGrp`.

If the network response payload is larger than the maximum permitted message size for that FIX conversation the response would be several `NetworkCounterpartySystemStatusResponse(35=BD)` messages, the first with a status of full and then as many messages, as updates to the first message, adding information as required. The message layout is available [here](#).

## **4.2 Components**

### **4.2.1 CompIDReqGrp**

This component is a repeating group used to convey one or more counterparties. The component layout is available [here](#).

### **4.2.2 CompIDStatGrp**

This component is a repeating group used to convey the connection status of one or more counterparties. The component layout is available [here](#).

## 5 Category – User Management

The messages in this category are provided to allow the passing of individual user information or user level settings between two counterparties.

### 5.1 Messages

#### 5.1.1 User Requests

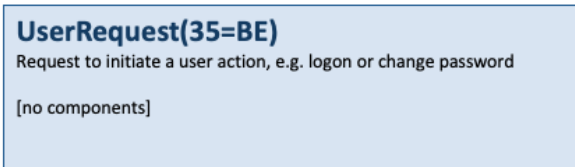


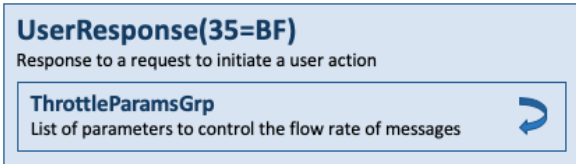
Figure 5: Message UserRequest(35=BE)

The UserRequest(35=BE) message is used to initiate a user action, e.g. logon, logout or password change. It can also be used to request a report on a user's status. The messages allow for specific functions by means of UserRequestType(924).

**NOTE: While this message may be used to change user password, it is not encouraged to transmit passwords in a FIX conversation unless there is guarantee of end to end security of both the FIX conversation and any intermediate routing hubs that are involved in routing the message.**

The message layout is available [here](#).

#### 5.1.2 User Responses

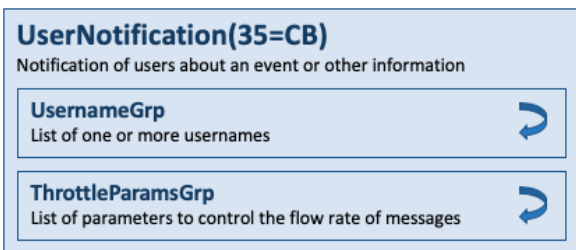


Note: Components in red boxes are required.

Figure 6: Message UserResponse(35=BF)

The UserResponse(35=BF) message is used to respond to a UserRequest(35=BE) message, it reports the status of the user after the completion of any action requested in the UserRequest(35=BE) message. The message layout is available [here](#).

#### 5.1.3 User Notifications



Note: Components in red boxes are required.

Figure 7: Message UserNotification(35=CB)

The UserNotification(35=CB) message is used to notify one or more users of an event or information from the sender of the message. This message is usually sent unsolicited from a marketplace (e.g. Exchange, ECN) to a market participant. The message layout is available [here](#).

## 5.2 Components

### 5.2.1 ThrottleMsgTypeGrp

This component is a repeating group that is part of the [ThrottleParamsGrp](#) and may be used to convey one or more message types subject to the specified throttling parameters. The component layout is available [here](#).

### 5.2.2 ThrottleParamsGrp

This component is a repeating group used to convey one or more parameters to control the flow rate of application messages. The component layout is available [here](#).

### 5.2.3 UsernameGrp

This component is a repeating group used to convey one or more usernames. The component layout is available [here](#).

## 6 Category – Application Sequencing

The Application Sequencing set of messages is used to manage application-level sequencing and address the segregation of retransmission of data over a session. This is primarily to address performance on a session when there is a need to retransmit data. These messages allow for the managing of this process based on specific upstream business systems and for the retransmission to occur over a separate FIX session or via other type of transport.

### 6.1 Introduction

FIX has a growing need to support application-level sequencing of messages in order to segregate the transmission of data over a session. The ability to identify and retransmit a subset of data by application and application sequence number range is an important feature in support of secondary distribution of data (see definition below). The current retransmission capabilities of the FIX session require that all messages on that session between the specified starting and ending message sequence number are resent rather than just those that have been produced by a specific upstream business process or application. This can pose capacity and performance problems for systems that need only a small set of messages related to an application. Secondary data distribution consists of a diverse set of data sourced from different applications; drop copy data, credit limit information, metrics, etc. It is **not** recommended that application sequencing is used over a conventional order routing or a transaction flow oriented connection. Standard FIX session capabilities should be used in this case.

Application sequencing greatly enhances the usefulness of FIX messages that are transmitted apart from the FIX session layer by making it possible for the receiver to detect and request missed messages on a specified feed. Market data sent over a broadcast or multicast transport is often in need of sequencing and retransmission. Application sequencing provides a means by which to sequence each message that is part of a broadcast stream such that the receiver can verify ordered delivery of the data. Application resends can then be requested when gaps are detected in the application sequence.

### 6.2 Background

The purpose of application-level sequencing is to allow messages being sent over a FIX session to be distinguished by the sending application that is upstream from the FIX engine. In the case that a session-level resend would result in an unnecessarily large number of messages being resent, application sequencing and recovery makes provision for the desired messages - and only the desired messages - to be seamlessly requested and resent while retaining the standard behaviors of the session protocol. It also provides the receiver with the flexibility to put off recovery of application level messages until a slow period or after the market has closed.

#### 6.2.1 Extends control over resent data

The primary intent of application sequencing and recovery is to allow receivers to avoid a retransmission of large quantities of unusable data which may result in receivers needing to glean the retransmission for the data they actually need - such as critical drop copy information that is used in risk management applications. Application sequencing allows the channeling of different types of data across a single FIX session. For example, application sequencing can allow drop copy data to be sent over the same FIX session with order flow data. While this may not be practical from a trading standpoint the flexibility that it introduces is compelling. This allows data which has a higher importance and priority to be identified by application ID thereby allowing requests for retransmission to be issued promptly and precisely.

#### 6.2.2 Support for secondary data distribution

Another goal of the proposal is to provide support for “secondary data” distribution. Application sequencing extends the capabilities of FIX such that secondary data can be distributed using a single channel. This data may be less time critical with less demanding latency requirements than order entry and market data, although this is not necessarily the case as drop copies are used for time sensitive risk management tasks. Secondary data may consist of drop copy fills, credit limit information, statistical data, trade confirmations, and best bids and offers for vendor consumption, etc. These are just a few of the possibilities. Application sequencing benefits data providers and their users by providing a common protocol which may be used to perform secondary data distribution. New applications transmitting data can be quickly introduced over an existing channel with minimal effort simply by introducing a new ApplID(1180) (application ID).

### 6.3 Transaction usage is not recommended

Application sequencing is not something that will be used in a normal order routing scenario. It has more relevance in large volume one-way connections in which the receiver would like to have some ability to control the data that is resent after a disconnect or data loss. There is no obvious advantage in using application sequencing with a regular trading connection since all data transmitted between sender and receiver is of equal importance in maintaining a viable trading session. Application sequencing should not be used to track broker connections that are in place for trading purposes. It should only be used for managing the flow of data when a FIX connection is used to deliver data in bulk and where there is a stated need to create classes of data.

### 6.4 Using Application Sequencing and Session Sequencing for Gap Detection

The use of ApplResendFlag(1352) on the [ApplicationSequenceControl](#) component (available on all FIX messages representing reports) is used to indicate that messages are being retransmitted as a result of an ApplicationMessageRequest(35=BW) message. When using the FIX Session Protocol, it is possible for both ApplResendFlag(1352) and PossDupFlag(43) to be set on the same message if the sender's cache size is greater than zero and the message is being resent due to a session level ResendRequest(35=2) message.

The sender and receiver may agree to use a limited cache in order to benefit from the convenience of session-level retransmission. In this case, a message that is dropped in response to an ApplicationMessageRequest(35=BW) message may have both fields present. This scenario depends on whether (1) the sender is maintaining a cache and (2) the sender and receiver have agreed to fill any gaps to the extent possible using the session level.

In this scenario, a combination of application and session-level sequencing will be used to recover missed messages. A limited cache of session-level messages may be retained by the sender in order to recover messages that have been dropped within an pre-stated window defined by time or number of messages. When a FIX session ResendRequest(35=2) message is issued within this window the sender's session will resend the messages. Once the window has been exceeded an ApplicationMessageRequest(35=BW) must be issued in order to recover dropped messages. The application level will not be aware that a gap has occurred until the session level has recovered what is available. Beyond this, the application will detect the gap according to the logic as described and issue a resend request at the application level using the ApplicationMessageRequest(35=BW) message.

**NOTE:** Gap detection and recovery with respect to the ApplicationMessageRequest(35=BW) message and response messages (e.g. ApplicationMessageRequestAck(35=BX) and resent application messages using the [ApplicationSequenceControl](#) component) may also need to take place at the application level since session level recovery may have been suspended.

## 6.5 Messages

### 6.5.1 Application Message Requests

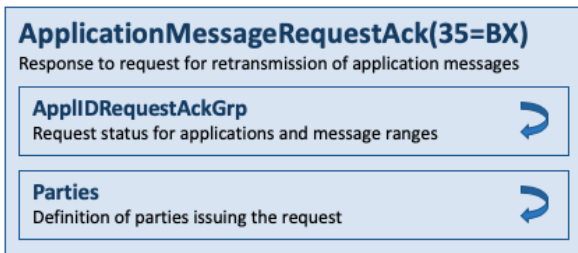


Note: Components in red boxes are required.

Figure 8: Message ApplicationMessageRequest(35=BW)

The ApplicationMessageRequest(35=BW) message is used to request a retransmission of a set of one or more messages generated by the application identified in the [ApplIDRequestGrp](#) component. The type of retransmission request is specified with ApplReqType(1347). The message layout is available [here](#).

## 6.5.2 Application Message Request Acknowledgements

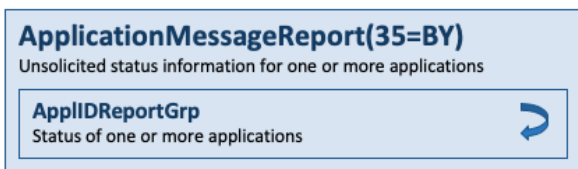


Note: Components in red boxes are required.

Figure 9: Message ApplicationMessageRequestAck(35=BX)

The ApplicationMessageRequestAck(35=BX) message is used to acknowledge an ApplicationMessageRequest(35=BW) message providing a status on the request (i.e. whether successful or not) with ApplResponseType(1348). This message does not provide the actual content of the messages to be resent. The message layout is available [here](#).

## 6.5.3 Application Message Reports



Note: Components in red boxes are required.

Figure 10: Message ApplicationMessageReport(35=BY)

The ApplicationMessageReport(35=BY) message is used for different purposes as indicated by ApplReportType(1426). The message layout is available [here](#).

### 6.5.3.1 Using Application Message Reports to reset application-level sequence number

The ApplicationMessageReport(35=BY) message with ApplReportType(1426) = 0 (Reset) is sent by the sender of an application to alert the receiver that the application sequence number of application RefApplID(1355) is being reset to ApplNewSeqNum(1399), for one or more RefApplID(1355) values, to the specified value(s). The next application message received will then conform to this value. In other words, ApplSeqNum(1181) in this message represents the next expected application sequence number the receiver will receive from the sender for the corresponding ApplID(1180). An ApplicationMessageReport(35=BY) message with ApplReportType(1426) = 0 (Reset) has no affect on, and is independent of, the FIX session sequence number in MsgSeqNum(34).

### 6.5.3.2 Using Application Message Reports to indicate last message sent

The ApplicationMessageReport(35=BY) message with ApplReportType(1426) = 1 (Last message) is sent by the sender of an application to indicate that the last message has been sent for one or more RefApplID(1355) values. Reception of this message mean the recipient can safely assume that no more message will be sent for that/or those RefApplID(1355) values. RefApplLastSeqNum(1357) should be set to ApplSeqNum(1181) on the last application-level message. RefApplID(1355) is set to ApplID(1180) on this message.

### 6.5.3.3 Using Application Message Report as keep-alive mechanism

For recipients of applications with infrequent message traffic it is a problem to detect a gap in the message flow. The gap cannot be detected until reception of the next message for that ApplID(1180). To mitigate this problem the ApplicationMessageReport(35=BY) message can be issued by the sender of an application at regular intervals. RefApplLastSeqNum(1357) should be set to the last ApplSeqNum(1181) sent for this application, identified by ApplID(1180) and referenced on the report by RefApplID(1355).

### 6.5.3.4 Using Application Message Report to indicate completion of resent messages

As part of a recovery scenario, the receiver (or consumer) may request all of the messages for one or more applications. Because of the potentially lengthy re-send situation, the request can be acknowledged with an



ApplicationMessageRequestAck(35=BX) prior to beginning the re-send of messages. In this case, the receiver or consumer will begin seeing re-sent messages until the re-send is complete. However, once the re-send is complete, the receiver or consumer will only know that the re-send has completed when they receive a new copied message from that application identified with ApplID(1180) that no longer has ApplResendFlag(1352) = Y. If the specified ApplID(1180) is only “heartbeating” and there are no new messages to send, the consumer will still not know the application message re-send has actually finished. It is in this case that an ApplicationMessageReport(35=BY) can be generated, which signals completion by setting ApplReportType(1426) = 3 (application message re-send completed).

## 6.6 Components

### 6.6.1 ApplIDReportGrp

This component is a repeating group used to convey a new application sequence number for a reset and/or the last application sequence number of one or more applications identified by RefApplID(1355). The component layout is available [here](#).

### 6.6.2 ApplIDRequestAckGrp

This component is a repeating group used to provide the status on a request to retransmit application messages for one or more applications identified by RefApplID(1355). In case of a successful request, it provides information such as the range of application sequence numbers that will actually be returned per application. In case of failure, it provides a reason for the rejection for the specified application. The component layout is available [here](#).

### 6.6.3 ApplIDRequestGrp

This component is a repeating group used to provide the range of application sequence numbers specified in ApplBegSeqNum(1182) and ApplEndSeqNum(1183) for retransmission from one or more applications identified by RefApplID(1355). The component layout is available [here](#).

## 7 Appendix – Application Category

### 7.1 Messages

#### 7.1.1 ApplicationMessageRequest Message

Tag	Name	Req'd	Description
<b>Component</b>	<b>StandardHeader</b>	Y	MsgType = BW
1346	ApplReqID	Y	Unique identifier for request
1347	ApplReqType	Y	Type of Application Message Request being made
<b>Component</b>	<b>ApplIDRequestGrp</b>	N	
<b>Component</b>	<b>Parties</b>	N	
58	Text	N	Allows user to provide reason for request
354	EncodedTextLen	N	
355	EncodedText	N	
<b>Component</b>	<b>StandardTrailer</b>	Y	

#### 7.1.2 ApplicationMessageRequestAck Message

Tag	Name	Req'd	Description
<b>Component</b>	<b>StandardHeader</b>	Y	MsgType = BX
1353	ApplResponseID	Y	Identifier for the Application Message Request Ack
1346	ApplReqID	N	Identifier of the request associated with this ACK message
1347	ApplReqType	N	
1348	ApplResponseType	N	
1349	ApplTotalMessageCount	N	Total number of messages included in transmission
<b>Component</b>	<b>ApplIDRequestAckGrp</b>	N	
<b>Component</b>	<b>Parties</b>	N	
58	Text	N	
354	EncodedTextLen	N	
355	EncodedText	N	
<b>Component</b>	<b>StandardTrailer</b>	Y	

#### 7.1.3 ApplicationMessageReport Message

Tag	Name	Req'd	Description
<b>Component</b>	<b>StandardHeader</b>	Y	MsgType = BY
1356	ApplReportID	Y	Identifier for the Application Message Report
1346	ApplReqID	N	If the application message report is generated in response to an ApplicationMessageRequest(MsgType=BW), then this tag contain the ApplReqID(1346) of that request.
1426	ApplReportType	Y	Type of report

Tag	Name	Req'd	Description
<b>Component</b>	<b>ApplIDReportGrp</b>	N	
58	Text	N	
354	EncodedTextLen	N	
355	EncodedText	N	
<b>Component</b>	<b>StandardTrailer</b>	Y	

## 7.2 Components

### 7.2.1 ApplIDReportGrp

Tag	Name	Req'd	Description
1351	NoApplIDs	N	
→1355	RefApplID	N	
→1399	ApplNewSeqNum	N	
→1357	RefApplLastSeqNum	N	

### 7.2.2 ApplIDRequestAckGrp

Tag	Name	Req'd	Description
1351	NoApplIDs	N	
→1355	RefApplID	N	
→1433	RefApplReqID	N	
→1182	ApplBegSeqNum	N	
→1183	ApplEndSeqNum	N	
→1357	RefApplLastSeqNum	N	
→1354	ApplResponseError	N	
→Component	<b>NestedParties</b>	N	

### 7.2.3 ApplIDRequestGrp

Tag	Name	Req'd	Description
1351	NoApplIDs	N	
→1355	RefApplID	N	
→1433	RefApplReqID	N	
→1182	ApplBegSeqNum	N	Message sequence number of first message in range to be resent
→1183	ApplEndSeqNum	N	Message sequence number of last message in range to be resent. If request is for a single message ApplBeginSeqNo = ApplEndSeqNo. If request is for all messages subsequent to a particular message, ApplEndSeqNo = "0" (representing infinity).
→Component	<b>NestedParties</b>	N	

## 8 Appendix – BusinessReject Category

### 8.1 Messages

#### 8.1.1 BusinessMessageReject Message

Tag	Name	Req'd	Description
<b>Component</b>	<b>StandardHeader</b>	Y	MsgType = j (lowercase)
45	RefSeqNum	N	MsgSeqNum of rejected message
372	RefMsgType	Y	The MsgType of the FIX message being referenced.
1130	RefAppVerID	N	Recommended when rejecting an application message that does not explicitly provide AppVerID ( 1128) on the message being rejected. In this case the value from the DefaultAppVerID(1137) or the default value specified in the NoMsgTypes repeating group on the logon message should be provided.
1406	RefAppExtID	N	Recommended when rejecting an application message that does not explicitly provide AppExtID(1156) on the rejected message. In this case the value from the DefaultAppExtID(1407) or the default value specified in the NoMsgTypes repeating group on the logon message should be provided.
1131	RefCstmAppVerID	N	Recommended when rejecting an application message that does not explicitly provide CstmAppVerID(1129) on the message being rejected. In this case the value from the DefaultCstmAppVerID(1408) or the default value specified in the NoMsgTypes repeating group on the logon message should be provided.
379	BusinessRejectRefID	N	The value of the business-level "ID" field on the message being referenced. Required unless the corresponding ID field (see list above) was not specified.
380	BusinessRejectReason	Y	Code to identify reason for a Business Message Reject message.
58	Text	N	Where possible, message to explain reason for rejection
354	EncodedTextLen	N	Must be set if EncodedText field is specified and must immediately precede it.
355	EncodedText	N	Encoded (non-ASCII characters) representation of the Text field in the encoded format specified via the MessageEncoding field.
<b>Component</b>	<b>StandardTrailer</b>	Y	

## 9 Appendix – Network Category

### 9.1 Messages

#### 9.1.1 NetworkCounterpartySystemStatusRequest Message

Tag	Name	Req'd	Description
<b>Component</b>	<b>StandardHeader</b>	Y	MsgType = "BC"
935	NetworkRequestType	Y	
933	NetworkRequestID	Y	
<b>Component</b>	<b>CompIDReqGrp</b>	N	Used to restrict updates/request to a list of specific CompID/SubID/LocationID/DeskID combinations. If not present request applies to all applicable available counterparties. EG Unless one sell side broker was a customer of another you would not expect to see information about other brokers, similarly one fund manager etc.
<b>Component</b>	<b>StandardTrailer</b>	Y	

#### 9.1.2 NetworkCounterpartySystemStatusResponse Message

Tag	Name	Req'd	Description
<b>Component</b>	<b>StandardHeader</b>	Y	MsgType = "BD"
937	NetworkStatusResponseType	Y	
933	NetworkRequestID	N	
932	NetworkResponseID	Y	
934	LastNetworkResponseID	N	Required when NetworkStatusResponseType=2
<b>Component</b>	<b>CompIDStatGrp</b>	Y	Specifies the number of repeating CompID's
<b>Component</b>	<b>StandardTrailer</b>	Y	

### 9.2 Components

#### 9.2.1 CompIDReqGrp

Tag	Name	Req'd	Description
936	NoCompIDs	N	
→930	RefCompID	N	Used to restrict updates/request to specific CompID
→931	RefSubID	N	Used to restrict updates/request to specific SubID
→283	LocationID	N	Used to restrict updates/request to specific LocationID
→284	DeskID	N	Used to restrict updates/request to specific DeskID

**9.2.2 CompIDStatGrp**

Tag	Name	Req'd	Description
936	NoCompIDs	N	
→930	RefCompID	Y	CompID that status is being report for. Required if NoCompIDs > 0,
→931	RefSubID	N	SubID that status is being report for.
→283	LocationID	N	LocationID that status is being report for.
→284	DeskID	N	DeskID that status is being report for.
→928	StatusValue	Y	
→929	StatusText	N	Additional Information, i.e. "National Holiday"

## 10 Appendix – UserManagement Category

### 10.1 Messages

#### 10.1.1 UserRequest Message

Tag	Name	Req'd	Description
<b>Component</b>	<b>StandardHeader</b>	Y	MsgType = "BE"
923	UserRequestID	Y	
924	UserRequestType	Y	
553	Username	Y	
554	Password	N	
925	NewPassword	N	
1400	EncryptedPasswordMethod	N	
1401	EncryptedPasswordLen	N	
1402	EncryptedPassword	N	
1403	EncryptedNewPasswordLen	N	
1404	EncryptedNewPassword	N	
95	RawDataLength	N	
96	RawData	N	Can be used to hand structures etc to other API's etc
<b>Component</b>	<b>StandardTrailer</b>	Y	

#### 10.1.2 UserResponse Message

Tag	Name	Req'd	Description
<b>Component</b>	<b>StandardHeader</b>	Y	MsgType = "BF"
923	UserRequestID	Y	
553	Username	Y	
926	UserStatus	N	
<b>Component</b>	<b>ThrottleParamsGrp</b>	N	
927	UserStatusText	N	Reason a request was not carried out
<b>Component</b>	<b>StandardTrailer</b>	Y	

#### 10.1.3 UserNotification Message

Tag	Name	Req'd	Description
<b>Component</b>	<b>StandardHeader</b>	Y	MsgType = CB
<b>Component</b>	<b>UsernameGrp</b>	N	List of users to which the notification is directed
926	UserStatus	Y	Reason for notification - when possible provide an explanation.
<b>Component</b>	<b>ThrottleParamsGrp</b>	N	
58	Text	N	Explanation for user notification.

Tag	Name	Req'd	Description
354	EncodedTextLen	N	Must be set if EncodedText field is specified and must immediately precede it.
355	EncodedText	N	Encoded (non-ASCII characters) representation of the Text field in the encoded format specified via the MessageEncoding field.
<b>Component</b>	<b>StandardTrailer</b>	Y	

## 10.2 Components

### 10.2.1 ThrottleMsgTypeGrp

Tag	Name	Req'd	Description
1618	NoThrottleMsgType	N	
→1619	ThrottleMsgType	N	Required when NoThrottleMsgType > 0.

### 10.2.2 ThrottleParamsGrp

Tag	Name	Req'd	Description
1610	NoThrottles	N	
→1611	ThrottleAction	N	Required when NoThrottles > 0.
→1612	ThrottleType	N	Required when NoThrottles > 0.
→1613	ThrottleNoMsgs	N	Number of messages per time interval, or number of outstanding requests. Required when NoThrottles > 0.
→1614	ThrottleTimeInterval	N	Can be used only when ThrottleType = Inbound Rate. Indicates, along with ThrottleTimeUnit, the interval of time in which ThrottleNoMsgs may be sent. Default is 1.
→1615	ThrottleTimeUnit	N	Can be used only when ThrottleType = Inbound Rate. Indicates, along with ThrottleTimeUnit, the interval of time in which ThrottleNoMsgs may be sent. Default is Seconds.
→Component	<b>ThrottleMsgTypeGrp</b>	N	Indicates MsgType values that this throttle counts. If not specified, the definition is implicit based upon bilateral agreement.

### 10.2.3 UsernameGrp

Tag	Name	Req'd	Description
809	NoUsernames	N	
→553	Username	N	Recipient of the notification