Futures Industry Association

Pre-Trade Credit Limit Check Enhancements

June 6, 2013

Revision 1.3

Proposal Status: Approved

For Global Technical Committee Governance Internal Use Only

Submission Date	January 3, 2013	Control Number	EP171
Submission Status	Approved	Ratified Date	July 2, 2013
Primary Contact Person	James Woods, FIA; Lisa Taikitsadaporn, Brook Path Partners	Release Identifier	5.0 SP3

r3

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).

Table of Contents

DOCUMENT HISTORY **6**

1 INTRODUCTION **9**

1.1 SUMMARY OF CHANGES 9

2 BUSINESS WORKFLOW **10**

2.1 PLUSONE MODEL 10
2.2 PING MODEL 11
2.3 PUSH MODEL 12
2.4 FIRM LEVEL TRADING HALT 13
2.5 HEARTBEAT MESSAGE 14

3 ISSUES AND DISCUSSION POINTS **14**

3.1 APPLICATION LEVEL HEARTBEAT 14
3.2 SPECIFYING OF LIMIT UTILIZED IN THE PING MODEL 15
3.3 RISKLIMITTYPE(1530) AND LIMITAMTTYPE(1631) 15
3.4 FIRM LEVEL TRADING HALT SUPPORT 16
3.5 SEMANTICS OF "KILL SWITCH" 17

4 PROPOSED MESSAGE FLOW **17**

4.1 LIMIT SETUP 17
4.1.1 Limit Setup *18*
4.1.2 Limit Definition Update *18*
4.2 PLUS ONE MODEL 19
4.2.1 Trade Breaches Credit Limit *19*
4.2.2 Clip Size Breached, Transition to Ping Model *20*
4.3 PING MODEL 21
4.3.1 New Order – Credit Check Accepted in Full *21*
4.3.2 New Order or Quote Request – Credit Limit Check Accepted Partially *22*
4.3.3 New Order or Quote Request– Credit Limit Check Rejected *23*
4.3.4 Order Canceled *24*
4.3.5 Order Executed *25*
4.3.6 New Quote Request – Accepted Full (Executing Broker Pre-Clearance) *26*
4.3.7 New Quote Request – Accepted Full (Executing Broker Cleared when post-trade) *27*
4.3.8 Voice Approval *30*
4.4 PUSH MODEL 31
4.5 FUEL GAUGE 32
4.5.1 Request/Subscribe *32*
4.5.2 Cancel Subscription *33*
4.6 FIRM LEVEL TRADE HALT 34
4.7 APPLICATION TEST 35

5 FIX MESSAGE TABLES **36**

5.1 PARTYRISKLIMITSREPORTACK(35=DE) 36
5.2 PARTYRISKLIMITCHECKREQUEST (35=DF) 38
5.3 PARTYRISKLIMITCHECKREQUESTACK (35=DG) 40
5.4 PARTYACTIONREQUEST(35=DH) 42
5.5 PARTYACTIONREPORT(35=DI) 44
5.6 EXECUTIONREPORT(35=8) 45

6 FIX COMPONENT BLOCKS 46

6.1 COMPONENT RISKLIMITTYPESGRP 46
6.2 COMPONENT REQUESTINGPARTYGRP 48
6.3 COMPONENT PARTYDETAILGRP 49
6.4 COMPONENT PARTYRISKLIMITSGRP 50
6.5 COMPONENT PARTYRISKLIMITSUPDATEGRP 51
6.6 COMPONENT PARTYRISKLIMITSACKGRP 52

7 CATEGORY CHANGES 57

APPENDIX A - DATA DICTIONARY 58

APPENDIX B - GLOSSARY ENTRIES 69

APPENDIX C - ABBREVIATIONS 70

APPENDIX D - USAGE EXAMPLES 70

APPENDIX E - PUBLIC COMMENT DISPOSITION 70

Comment A: 71
Comment B: 72
Comment C: 72
Comment D: 73
Comment E: 74
Comment F: 74

Table of Figures

Figure 1: PlusOne Model 11
Figure 2: Ping Model 12
Figure 3: Push Model 13
Figure 4: Limit Setup 18
Figure 5: Limit definition update 19
Figure 6: PlusOne to Ping model on credit breach 20
Figure 7: PlusOne to Ping on Clip size breach 21
Figure 8: Ping Model credit check accepted in full 22
Figure 9: Ping Model credit check partially accepted 23
Figure 10: Ping Model credit check rejected 24
Figure 11: Ping Model order cancelled 25
Figure 12: Ping Model order executed 26
Figure 13: Ping Model quote request accepted in full, executing broker credit checked pre-trade 27
Figure 14: Ping Model quote request accepted in full, executing broker credit checked post-trade 29
Figure 15: Ping Model off-SEF negotiated transaction 31
Figure 16: Subscription request 33
Figure 17: Subscription termination 34
Figure 18: "Halt trading" for a specified party 35
Figure 19: Application test message 36

Document History

Revision	Date	Author	Revision Comments
0.1	8/10/2012	L. Taikitsadaporn , Il Seo	Created Initial message flows for Plus One, Ping, and Push models using draft use case document.
0.2	8/23/2012	L. Taikitsadaporn , Il Seo	Updated message flows to reflect initial comments from ISDA/FIA Joint Committee.
0.3	9/5/2012	L. Taikitsadaporn , Il Seo	Created initial draft Gap Analysis
0.4	9/23/2012	L. Taikitsadaporn , Il Seo	Updated Message Flows to reflect final use cases from ISDA/FIA Joint Committee. These use cases are for the 3 different credit models, fuel gauge, and Kill switch.
0.5	10/2/2012	L. Taikitsadaporn , Il Seo	Created draft Gap Analysis that reflects the updated flows.
0.6	10/16/2012	L. Taikitsadaporn , Il Seo	Updated message flows and updated/created FIX messages.
0.7	10/30/2012	L. Taikitsadaporn , Il Seo	Updated FIX messages, components, and data definitions
0.8	11/29/2012	Il Seo	Updated FIX messages to contain application level failsafe check.
0.9	12/15/2012	L. Taikitsadaporn	Reviewed and updated for submission to GTC.
1.0	Jan. 4. 2013	L. Taikitsadaporn	Updated based on GTC review feedback. - replaced new PartySuspension messages (see Appdx E) with PartyDetailDefinition messages - added new boolean field to denote an "application test" message
1.1	Jan. 18, 2013	L. Taikitsadaporn	Updated based on GTC Jan. 17th review feedback.
1.2	Jan. 21, 2013 Feb. 28, 2013	L. Taikitsadaporn L. Taikitsadaporn	Updated based on further discussion with GTC Co-chair. New messages will be used for the "suspension" and "kill actions". The previously proposed PartySuspensionRequest and PartySuspensionReport was revisited and renamed to PartyActionRequest and PartyActionReport. Additional clean up edits prior to public comment, particularly around wording related to the term "Kill switch" vs. "halt" with the latter as preferred FIX semantic. Updated

Revision	Date	Author	Revision Comments
			sections 2.4, 2.5, 3.4, 3.5, 4.6, 4.7 Updated Figures 18, 19 Added RelatedPartyDetailGrp to new messages PartyActionRequest(35=TBD) and PartyActionReport(35=TBD).
1.3	Apr. 27, 2013 June 6, 2013	L. Taikitsadaporn	Changes made based on public comment feedback and disposition (see Appendix E): • Added new field RiskLimitCheckType to convey the type "submit" or "limit consumed" in PartyRiskLimitCheck messages • Changed RiskLimitCheckRequestMsgID to RiskLimitCheckRequestID and RiskLimitCheckRequestMsgRefID to RiskLimitCheckRequestRefID • PartyActionRequestID is made an optional field and added CopyMsgIndicator as an optional field in PartyActionReport message • Updated data dictionary with missing fields from new messages • Changed PartyActionRequestType to be PartyActionType • Add existing field RiskLimitID(1670) to PartyRiskLimitCheckRequest/Ack messages • Switched to Parties and RelatedPartyDetailGrp components in the RiskLimitCheckRequest and RiskLimitCheckRequestAck messages instead of PartyDetailGrp since the limit check is for a single party and the specified transaction reference. • Updated all the flow diagrams to reflect changes made • Corrected typos and abbreviations • removed RiskLimitBreachType from Summary of Changes as this is part of RiskLimitAction(1767)

Revision	Date	Author	Revision Comments
			• added component tables to show where to add RiskLimitCheckModelType as described in the Summary of Changes
ASBUILT	June 19, 2013	Lisa T. Rich S.	Created ASBUILT doc for EP171 build. Provisional tag, message and enum assignements.
	June 24, 2013	Lisa T.	Added additional rejection reason for "Exceeded CS01 limit" to OrdRejReason(103), QuoteRejectReason(300), QuoteREquestRejectReason(658) to allow for rejection based on RiskLimitType(1530) setting for CS01 limit.
	July 1, 2013	Lisa T.	Reclassified message category for PartyActionRequest, PartyActionReport, PartyRiskLimitCheckRequest, PartyRiskLimitCheckRequestAck to new category called "PartiesAction" Rectified FIXML abbreviation for RiskLimitID in the PartyRiskLimitCheckRequest, and PartyRiskLimitCheckRequestAck messages.
	July 8, 2013	R. Shriver	Added change to RiskWarningLevelAction(1769) to use enum values from RiskLimitAction(1767) as per discussion with Hanno and Lisa.
	Oct. 15, 2013	Lisa T.	Final QC: SPEC-1025 to SPEC-1036 created, SPEC-986 reopened. Adjustments made as follows: Data type for RiskLimitCheckAmount(2324) and RiskLimitApprovedAmount(2327) changed to Amt (previously Qty). Removed ApplicationSequenceControl component from PartyRiskLimitsReportAck msg - not needed in an ack msg. Removed duplicate proposed enum value to QuoteRequestRejectReason.

1 Introduction

This gap analysis is the result of analysis completed by the Messaging Sub-Group of the FIA/ISDA Joint Working Group that was created to address pre-trade clearing certainty for cleared swap transactions through all combinations of trade parties. This Joint Working Group was tasked with defining the requirements and use cases so FpML and FIA can define or enhance existing messaging standards to support the requirements. This work is a response to a regulatory requirement detailed in the CFTC's Rule 1.73 for Dodd-Frank Act.

This gap analysis proposal describes enhancements needed in FIX to support the pre-trade credit limit check workflows between customers, dealers, clearing members, execution venues (such as swaps execution facilities or SEFs), and the clearinghouse.

The following documents were the requirements documents provided by the FIA/ISDA Joint Working Group that formed the basis for this gap analysis proposal to enhance the FIX Protocol:

- "Pre-Execution Clearing Certainty: Messaging Protocol Use Cases" (Messaging Protocol Use Case Final Draft v3.pdf)
- "Conclusion of the ISDA/FIA Kill Switch Messaging Sub-Group" (20120831-Conclusion of the ISDA/FIA Kill Switch Messaging Sub-Group.pdf)
- "Recommendation of the ISDA/FIA Heartbeat Messaging Sub-Group" (20120803-Recommendation of the ISDAFIA Heartbeat Messaging Sub-Group.pdf)

1.1 Summary of changes

After reviewing the detailed use cases and requirements it was determined that the Parties Reference Data set of messages approved under Extension Packs EP105, EP128, EP129 and EP146[1] would be enhanced to support the requirements for pre-trade credit check. However, to fully satisfy the requirements set by the FIA/ISDA Joint Working Group the following new messages are also being proposed:

- PartyRiskLimitsReportAck(35=DE~~TBD~~) - used to acknowledge or nack the PartyRiskLimitReport(35=CM) message
- PartyRiskLimitCheckRequest(35=DF~~TBD~~) - used by the Ping Model to request for credit limit approval
- PartyRiskLimitCheckRequestAck(35=DG~~TBD~~) - used to respond to the PartyRiskLimitCheckRequest(35=DF~~TBD~~) message to either accept or reject the credit request
- PartyActionRequest(35=DH~~TBD~~) - use to request a specified action to be taken on the identified party
- PartyActionReport(35=DI~~TBD~~) - used to respond to the PartyActionRequest(35=DH~~TBD~~) message to either accept or reject the action request

[1] The Extension packs can be found at this URL
http://www.fixprotocol.org/specifications/FIX.5.0SP2#Extension_Packs_enhancing_FIX_5.0_SP2

Other enhancements include:

- Added to PartyDetailStatus(1672) a new enum value of "halted" to allow for specifying the status of the party.
- The RiskLimitType(1530) field in the RiskLimitTypesGrp is updated to include new values for clip size, maximum order quantity, DV01 and limit amount consumed by a trade transaction.
- New fields to support the ability to specify velocity: RiskLimitVelocityPeriod(2336~~tbd~~) and~~,~~ RiskLimitVelocityUnit(2337~~tbd~~)~~ and RiskLimitVelocityPeriod(tbd)~~. The clip size and velocity are used to together to define the amount that can be traded within a specified period of time.
- A new field in the PartyRiskLimitsGrp, PartyRiskLimitsUpdateGrp and PartyRiskLimitsAckGrp components provides a means to specify which credit limit check model to be setup for the given party.
    - RiskLimitCheckModelType(2339~~tbd~~) - specifies which credit check model to utili~~i~~ze for the party
- Added to RiskLimitAction(1767) several new enumeration values to allow specifying what action to take when limit is breached.
- A new RequestingPartyRoleQualifier(2338~~tbd~~) is being added to the RequestingPartyGrp to be able to specify that an Intermediary (RequestingPartyRole(1660)=29(Intermediary) is a "Hub". A new role qualifier value of "Hub" would be added.
- Add to the field descriptions for RefOrderID(1080) and RefOrderIDSource(1081) to allow these fields to be used to carry reference identifiers for credit limit check requests used in the Ping Model.
    - Additional RefOrderIDSource values added to allow for order, quote request and quote message reference identifiers to be specified

2 Business Workflow

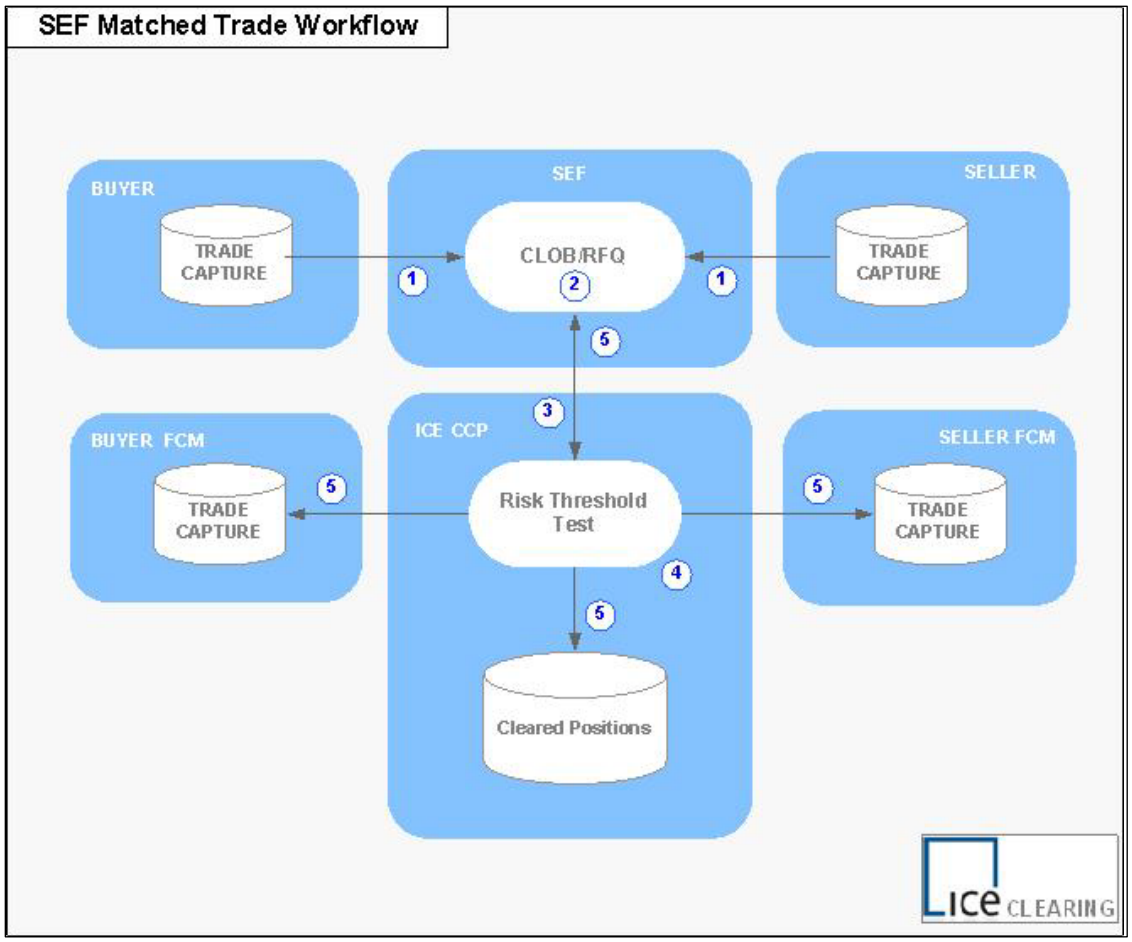
The Messaging Sub-Group of the FIA/ISDA Joint Working Group had defined three credit check models that would meet the requirements of credit check prior to trade completion for CFTC Rule 1.73. The three credit check models are described in the following sections.

2.1 PlusOne Model

In the PlusOne Model, the credit source (Central Counter Party (CCP) for Clearing Member (CM) limits or CM for Customer limits) indicates to the Swap Execution Facility (SEF) that the credit user (CM or Customer) is in good standing and to accept each trade for clearing that is done by the credit user for as long as the credit user remains in good standing. When a trade executed or guaranteed by a CM breaches the set limit at the CCP, or if a trade executed by Customer breaches the set limit by the CM, the credit source will accept that trade, but will have the right to prevent all other orders, quote requests or quotes in flight or standing associated with that credit user from being accepted, and will be rejected with an appropriate code.

The CCP may accept risk reducing trades for the CM or Customer after the party has breached their allowable credit limit. The allowable credit limit could be a maximum daily limit, a maximum trade size, or a maximum number of trades in a given time period. These risk reducing trades could be accepted using the Ping Model.

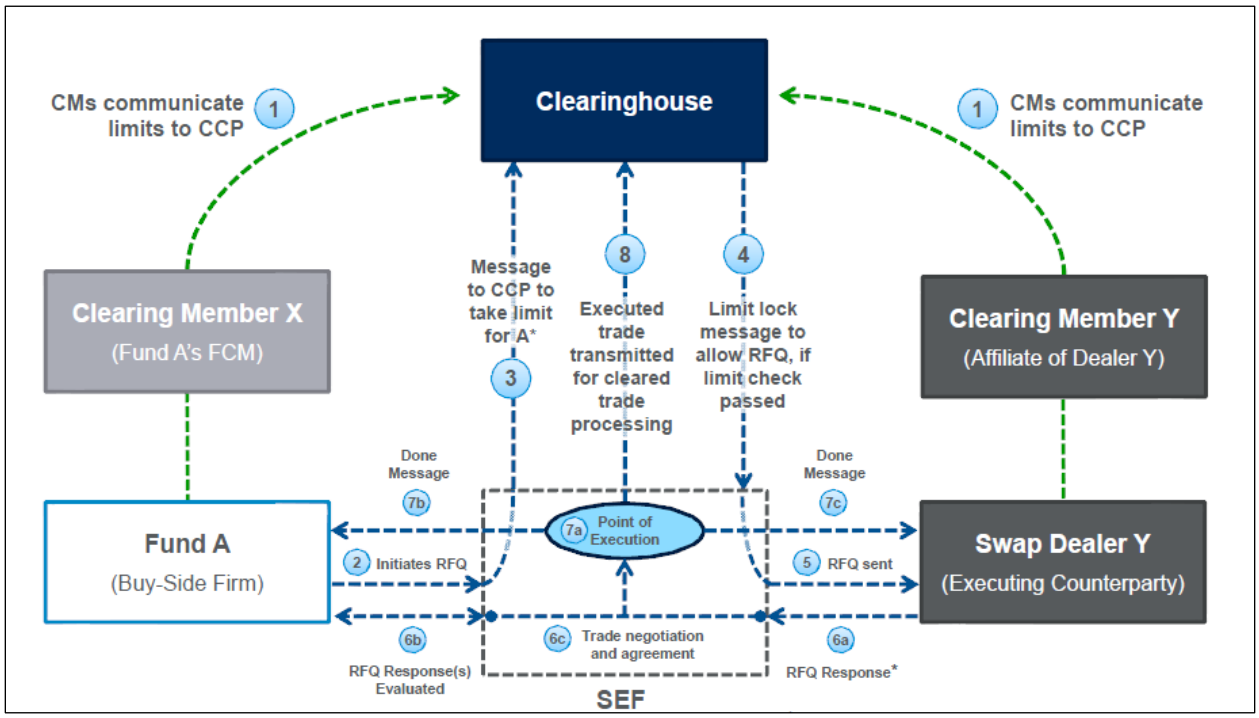
Figure 1: PlusOne Model



2.2 Ping Model

In the Ping Model, the credit user's credit limit is kept at the credit source. Messages sent from the SEF on-behalf-of the credit user to the credit source will reserve credit for each individual order, quote request, or quote submitted by the credit user. If credit is insufficient or not available, the initiated order, quote request, or quote is rejected and not visible to any other parties. If there is sufficient credit, the credit limit amount equal to the amount of the order, quote request or quote is reserved. Once the trade is done, the amount traded is credit that is consumed. If the order, quote request, or quote is not completed, times out, or is withdrawn (i.e. canceled), the limit that was reserved is unlocked.

Figure 2: Ping Model

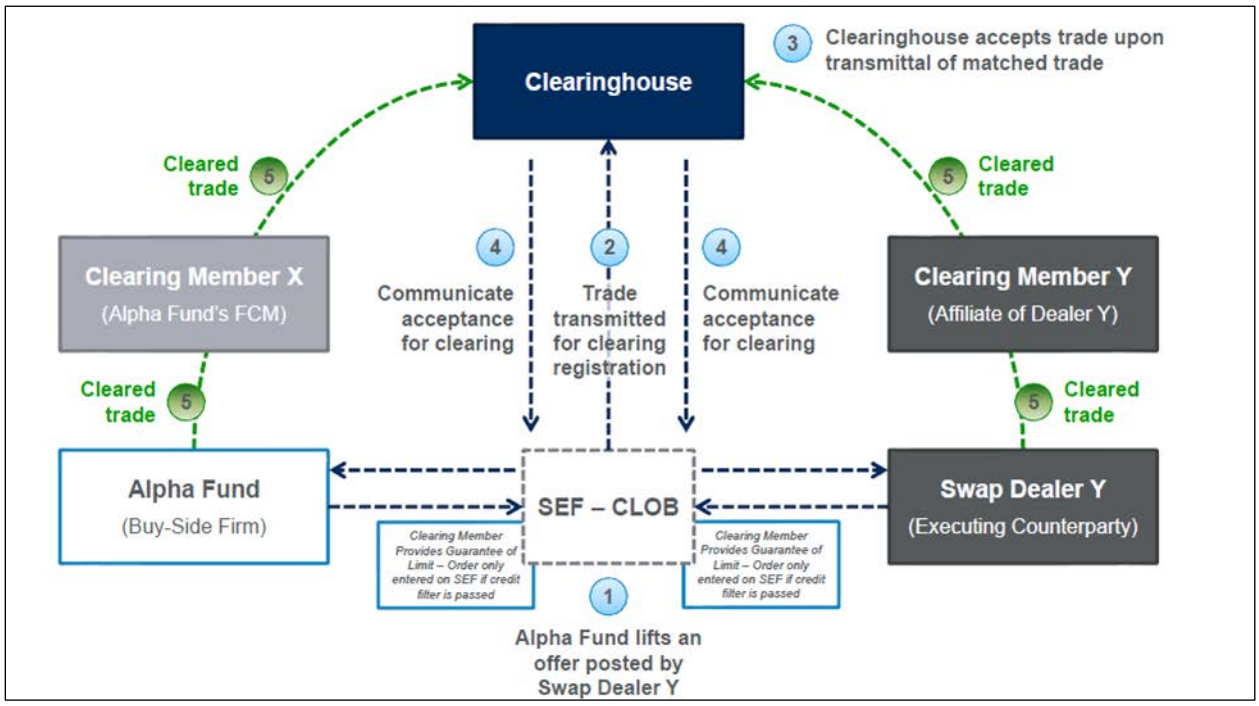


2.3 Push Model

Under the Push Model, CMs allocate a portion of their Customer's total risk limit to the SEF.

As each of the participants initiates a transaction, or seeks to respond to another party's quote request, or order initiation, before their activity is allowed by the SEF to be visible to other parties, the SEF checks internally that the relevant participant has sufficient credit for the transaction. If relevant participant has sufficient credit, the SEF locks the amount of credit needed for the transaction. If there is insufficient credit, the submitted message is rejected and is not displayed to other market participants. Once the credit is locked, the initiation or response may proceed until the order, quote request, or quote results in a trade, withdrawn, times out, or (highly exceptionally) the participant is killed through the application of a "Kill Switch".

Figure 3: Push Model



2.4 Firm level trading halt

The ISDA/FIA Kill Switch Messaging Sub-Group defined and issued a set of use cases that pertain to the triggering of a "Kill switch" against a Credit User (CM, Client, or EB). The trigger of a "kill switch" can be applied to a single or multiple platforms that the Credit User trades on and for a single or multiple asset classes that they trade. From a FIX semantics stand point FIX will not be using the term "kill switch" to express this type of activity[2], but will refer to this action as a "trading halt" action.

The ISDA/FIA Kill Switch Message Sub-group had defined two types of switches. The first is a "hard kill", or "trade halt". This is issued when the credit provider wishes intends to stop the ability of a credit user from trading any security on any SEF platform. This "hard kill" would be used only in highly exceptional scenarios. A "soft kill", or suspension, stops a Credit User's ability to, for example, trade on a specific SEF, clear through a specific CCP, or, in the case of an FCM, to suspend the FCM from accepting any new transactions for clearing. The scope of a "soft kill" is intended to be narrower than that of a "hard kill".

The proposal accommodates both of these concepts via the use of the PartyActionRequest(35=DH~~TBD~~) message. The use of the PartyAction~~Request~~Type(~~tbd~~2239)=1 (Halt trading) would effectively change the overall status of the party, or the status of their relationship to other parties, from "active" to "suspended" or "halted". Exactly what other actions (e.g. cancelling all outstanding orders/quotes or not) result from a party being "suspended" or "halted", will depend in whether there is a third party hub provider in the middle of the communication or not, and on specific SEF implementations. However, that implementation question will not be explicitly addressed in this proposal.

Additionally the sub-group also required the ability to reinstate a Credit User's ability to resume trading or clearing trades. This can be done via the PartyActionRequest(35=DH~~TBD~~) with

[2] See section 3.5 for further discussion around the semantics of the term "kill switch".

PartyAction~~Request~~Type(2239~~tbd~~) = 2 (reinstate) which would result in changing the party's status, or their relationship to another party, to "active".

2.5 Heartbeat Message

The ISDA/FIA Heartbeat Messaging Sub-Group issued a set of use cases that pertain to Heartbeat Messages. The concept of the "heartbeat" in this use is at the application level as oppose to the session level. The "heartbeat" message is used primarily to ensure that the receiving system can affect a party action should one be triggered. The purpose of this message~~s~~ is to obtain reasonable assurance from the receiving party that the application that would process the "trade halt" is available to respond to a "trade halt". To meet this requirement the new PartyActionRequest(35=DH~~TBD~~) message provides a ApplTestMessageIndicator(2230~~tbd~~) indicating that the PartyActionRequest(35=TBD) message is an "application test" message.

3 Issues and Discussion Points

3.1 Application level heartbeat

One of the requirements from the FIA/ISDA Joint Working Group is to have the ability for an application level heartbeat message to ensure that the system that would process and respond to a "kill switch" message is up and available.

The initial proposal ~~-~~is to utilize the new PartyActionRequest(35=DH~~TBD~~) message with a PartyAction~~Request~~Type(2239~~tbd~~) indicating that the message is an "application test" message.

Other possible alternatives are:

1. E~~e~~nhance the existing ApplicationMessageRequest(35=BW) with a new ApplReqType(1347) to request that the application periodically send a "heartbeat" indicating that the application is still alive. A "heartbeat interval" may need to be defined. Currently the ApplicationMessageReport(35=BY) already contain an enum value in ApplReportType(1426) of "2" to indicate a "Heartbeat message indicating that Application identified by RefApplID(1355) is still alive.", however this is further qualified with the text "Refer to RefApplLastSeqNum(1357) for the application sequence number of the previous message." The latter may not necessary be applicable under the credit check model requirements.

   The ApplicationMessageRequest message originally intended for only application level message sequencing and not for testing application availability, however, the ApplReportType(1426)=2 seems to imply that the ApplicationMessageRequest should have allowed for a means to request a status of the application's "health".

2. the other alternative is new messages specifically for application level heartbeat requests and response

Jan. 3, 2013 GTC Review: the participants on the call felt that it was best to use a boolean field at the main/root level of the message to indicate that the message is an "application test" message. The field should be added to the PartyXXXRequest and corresponding Ack messages. For the submission of this gap analysis, the new field will only be added to the PartyActionRequest(35=DH~~TBD~~) and

PartyActionReport(35=DH~~TBD~~). Future gap analysis submission may seek to add this new field to other messages as required.

3.2 Specifying of limit utilized in the Ping model

The Ping Model requires the SEF to report back to the Limit Checker or Credit Extender, the actual amount of the requested limit that was "consumed" or utilized by a trade. The proposal currently uses the new PartyRiskLimitCheckRequest(35=DF~~TBD~~) message with a RiskLimitCheckTransType(2320~~tbd~~)= "limit consumed" to report this information. It is possible that the amount of limit utilized may be less than the amount requested.

Should the PartyRiskLimitCheckRequest(35=DF~~TBD~~) message be used to report consumed limits? Or should another new message be considered that is used specifically to report this information back from the SEF to the Limit Checker?

Jan. 3, 2013 GTC Review: Additional background on this requirement. In reference to the workflow diagramed in Section 4.3.5, the requirement is that the SEF has to report back to the Credit Extender/Limit Checker the amount of previously approved and reserved credit was used up by the transaction (either in full or in part). While the ExecutionReport(35=8) has the LastLimitAmt(1632) within the LimitAmts component would be used to report back to the trading parties the limits utilized in that transaction, this field (and the component) is not being used as part of the proposed new PartyRiskLimitCheckRequest(35=TBD) message. As the proposed new message includes the RiskLimitTypesGrp component instead of the LimitAmts component, the proposal is to extend the RiskLimitType to include a type for identifying the limit utilized/consumed by a transaction.

It was pointed out that the RiskLimitTypesGrp also includes the RiskLimitUtilizationAmount(1766) field. The description of this field seems to imply that the party that manages the limits (i.e. the Credit Extender or Limit Checker) is the party that would provide the information for this field.

3.3 RiskLimitType(1530) and LimitAmtType(1631)

LimitAmtType(1631) was added as part of EP100 to the ExecutionReport while RiskLimitType(1530) was added as part of EP105 to the RequestedRiskLimitTypesGrp and RiskLimitTypesGrp components. The enumeration lists of these two fields are very similar yet slightly different. It appears that these two lists should be synced so that for the credit check model the limit can be reported back in an ExecutionReport(35=8) as well as part of a fill report. Currently the value lists are as follows:

LimitAmtType:

0	=	Credit limit	Added	FIX.5.0SP2	EP100	[CreditLimit]
1	=	Gross position limit	Added	FIX.5.0SP2	EP100	[GrossPositionLimit]
2	=	Net position limit	Added	FIX.5.0SP2	EP100	[NetPositionLimit]
3	=	Risk exposure limit	Added	FIX.5.0SP2	EP100	[RiskExposureLimit]
4	=	Long position limit	Added	FIX.5.0SP2	EP100	[LongPositionLimit]
5	=	Short position limit	Added	FIX.5.0SP2	EP100	[ShortPositionLimit]

RiskLimitType:

1	=	Gross limit	Added FIX.5.0SP2 EP105 Updated FIX.5.0SP2 EP128	[GrossLimit]
2	=	Net limit	Added FIX.5.0SP2 EP105 Updated FIX.5.0SP2 EP128	[NetLimit]
3	=	Exposure	Added FIX.5.0SP2 EP105	[Exposure]
4	=	Long limit	Added FIX.5.0SP2 EP105 Updated FIX.5.0SP2 EP128	[LongLimit]
5	=	Short limit	Added FIX.5.0SP2 EP105 Updated FIX.5.0SP2 EP128	[ShortLimit]
6	=	Cash margin	Added FIX.5.0SP2 EP128	[CashMargin]
7	=	Additional margin	Added FIX.5.0SP2 EP128	[AdditionalMargin]
8	=	Total margin	Added FIX.5.0SP2 EP128	[TotalMargin]

As part of this proposal the following values will need to be added to RiskLimitType:

0 = Credit limit (using the same enum value as in LimitAmtType)
9~~TBD~~ = Limit consumed (Elaboration: The limit used in the recent transaction.)
10~~TBD~~ = Clip size (Elaboration: The total amount allow to be traded within a defined period of time, or velocity.)
11~~TBD~~ = Maximum notional order size (Elaboration: The maximum amount allowed in any given submitted order, quote or quote request by the submitter.)
12~~TBD~~ = DV01/PV01 limit (Elaboration: The dollar or present value of one basis point~~s~~.)
13~~TBD~~ = CS01 limit (Elaboration: The credit spread value of one basis point~~s~~. The change in value of a CDS for a one basis point~~s~~ change in the credit spread.)

3.4 Firm level trading halt support

At the Jan. 3, 2013, GTC review of the proposal the participants suggested that a separate new set of messages is not needed for a ""trading halt" but to reuse the PartyDetailDefinitionRequest(35=CX) and PartyDetailDefinitionRequestAck(35=CY) instead with the appropriate enhancements to the message to support the "trading halt".

The original proposed new messages of PartySuspensionRequest and PartySuspensionRequestAck has been moved to Appendix E, and Sections 5 and 6 includes the enhanced PartyDetailDefinitionRequest(35=CX) and PartyDetailDefinitionRequestAck(35=CY), and related components, for discussion.

At the Jan. 17, 2013, GTC it was raised whether the concept of a "kill switch" should be a separate and distinct message instead of being in the now proposed part of the PartDetailDefinitionRequest (35=CX) message. Some members feel it should be a distinct message as it is an operational action as oppose to a reference data definition. It was proposed that a PartyActionRequest/Report message pair be introduced instead~~ad~~ that would allow an "action" to be taken for operational reasons. The participants on the call agreed to get larger community's comment on this particular topic.

Jan 21, 2013: After further discussion with GTC Co-chair, it was agreed to update the gap analysis to reflect a new message type for the functionality of a "trade half". The existing PartyDetailDefinitionRequest(35=CX) message will still be updated to allow for the conveyance of the party's current status, including a "halted" state. Two new messages, PartyActionRequest(35=DH~~TBD~~)

and PartyActionReport(35=DI~~TBD~~) are being proposed to support the ability to communicate the operational action of suspending or halting a party from trading.

3.5 Semantics of "kill switch"

At the Jan. 17th, 2012, GTC call there was discussion as to what the definition of a "kill switch" should mean. There a concern with the FIA/ISDA Joint~~g~~ Working Group's definition of a "kill switch" to mean the "stopping of a particular party from trading". It was raised that "kill switch" has primarily been discussed in the context of a more significant event of "stopping all trading at a market". This was raised as a matter of semantics of how FIX should not call the functionality proposed as a "kill switch".

Feb. 16th, 2013, external feedback: There was feedback after the GTC call from internan~~n~~l GTC Governance, exchanges and FIA Technology that the term "kill switch" is strongly discouraged in this context. A "kill switch" is clearly defined in systems engineering to be a highly exceptional system shut off to only be exercised when proper shutdown and control mechanisms of a system have failed. The use of "kill switch" within this context is a misuse and has a dilutive effect to the concept of the "kill switch". It appears the intent of the requirements from the ISDA/FIA Joint WG is not a "kill switch" that indicates a system failure that cannot be addressed, but instead a mechanism to be implemented within the core of the system that will serve to control party behavior within the system. As such for the purpose of the requirements related to pre-trade credit check, FIX will not use the ISDA/FIA Joint WG's terminology of "kill switch" but to refer to the stated required behavior as a "trade halt".

The recommended terminology:

Previous term	Recommendation
Kill	Halt
Suspend	Suspend

4 Proposed Message Flow

The FIA/ISDA Joint Working Group created a use case document which catalogs a number of use cases for each pre-execution credit/trading limit management model. Each entity (FCM, CCP, or Credit Hub) can play different roles in the process flow and the entity may opt to implement all or a subset of the use cases.

See Appendix B - Glossary Entries for the definition of the terms being used in the subsequent sections and in the message flow diagrams.

4.1 Limit Setup

For each of the above mentioned three credit check models the Credit Extender must convey to the SEF several different parameters to setup the Credit User's limits. These parameters may include the credit model used (Plus One, Ping, or Push), clip size and velocity limits, gross or net credit limits, or action to take when limit is breached (e.g. to transition to a different credit check model, or to suspend the Credit

User from further trading). The following flows will demonstrate the messages to setup and modify limit parameters.

4.1.1 Limit Setup

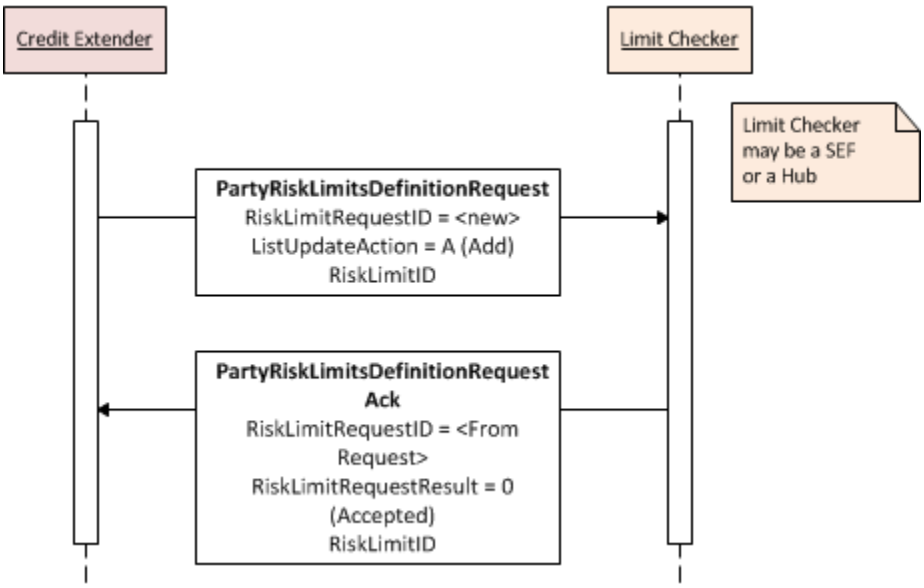
This use case allows a Credit Extender to define the limits for a given Credit User (e.g. the Customer or the FCM). The Credit Extender may define which model will be used to validate a Credit User along with the various different types of limits. Some of the setup information from the Credit Extender may include one or more of the following: product, CCP, clip size and velocity, etc.

The PartyRiskLimitsDefinitionRequest(35=CS) message is used by the Credit Extender to set the limits of the Credit User. The RiskLimitID(1670) in the mesesage may be used to uniquely identify the risk limit record for a given client between the Credit Extender and Limit Checker; the ListUpdateAction(1324) will indicate that the message is a new request; and the RiskLimitsUpdateGrp component will contain the Credit User and their various types of limits being set. The RiskLimitID(1670) may be used by the Credit Provider to provide a unique reference identifier that can be used for later referencing in updates to or deletion of the information. If RiskLimitID(1670) is not provided by the Credit Extender, the Limit Checker should provide an ID in RiskLimitID(1670) in the acknowledgement message.

The response would be a PartyRiskLimitsDefinitionRequestAck(35=CT) from the Limit Checker to indicate whether the definition request has been successfully processed.

More than one Credit User can be setup in a single message.

Figure 4: Limit Setup



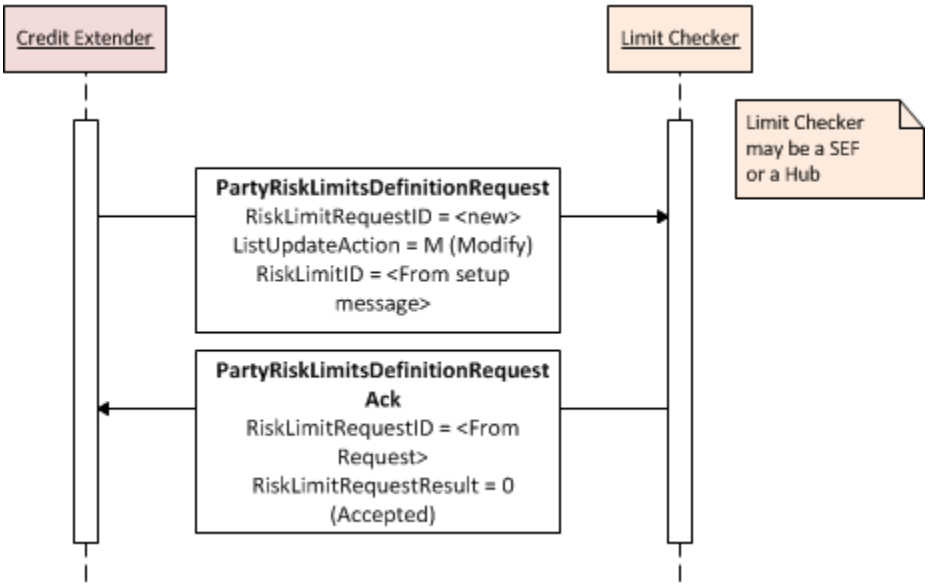
4.1.2 Limit Definition Update

This use case allows the Credit Extender to update previously created limits. The Credit Extender may update some or all of the information, including setting limits down to a zero amount.

The PartyRiskLimitsDefinitionRequest(35=CS) could be sent with the RiskLimitID(1670) containing the same ID used when the limit was setup by the Credit Extender. ListUpdateAction(1324)=M (modify) will define that this message is a "modify". The RiskLimitsUpdateGrp would contain the complete information related to the ID referenced in the RiskLimitID(1670).

More than one Credit User can be modified in a single message.

Figure 5: Limit definition update



4.2 Plus One Model

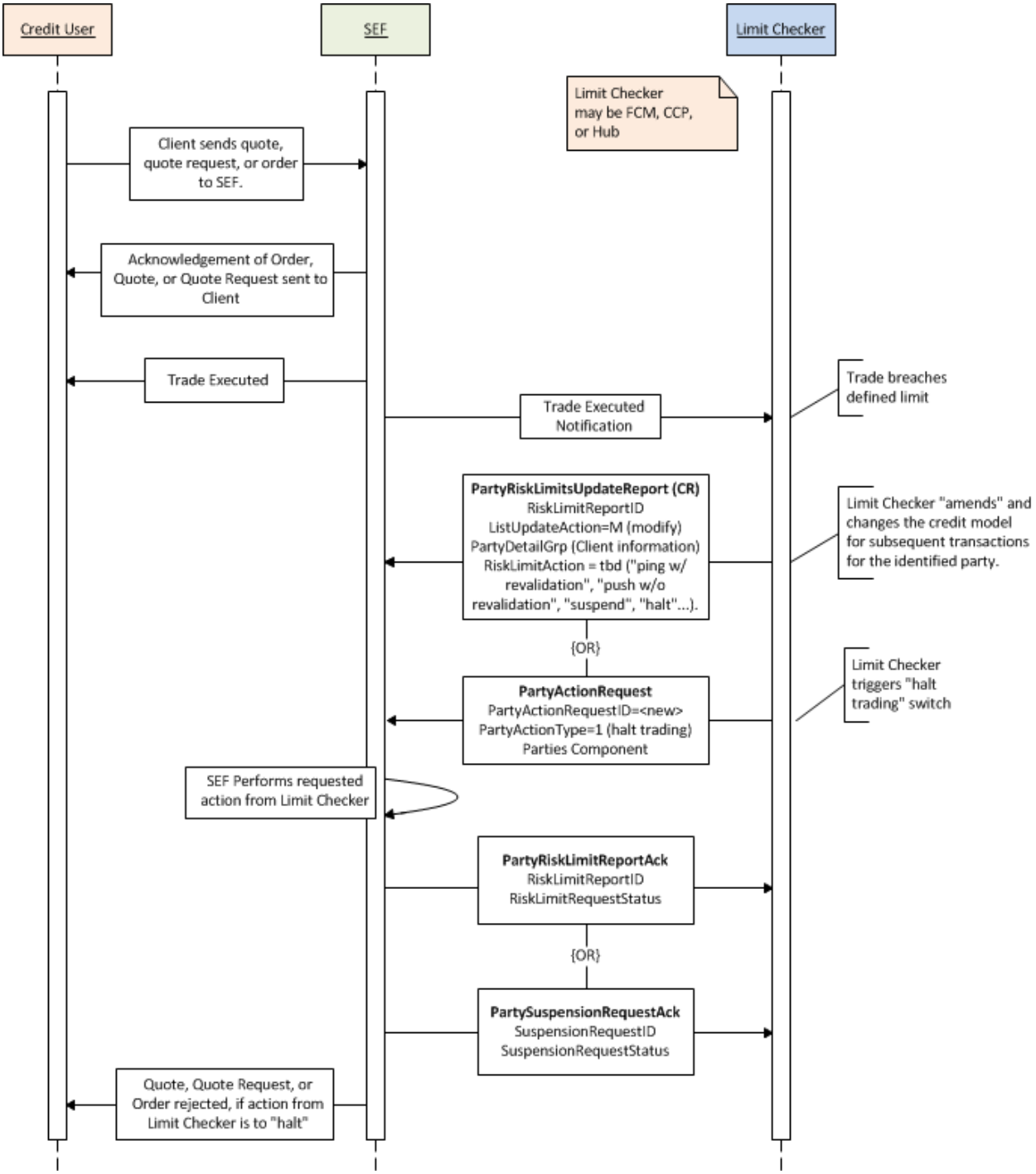
The Plus One Model assumes that all orders, quotes, and quote requests are to be accepted for clearing until the Credit Extender notifies the SEF otherwise or the limits have been breached. The flows in this section illustrates the use cases of a Credit User setup initially using the Plus One Model and the transition from the Plus One to another model (i.e. the Credit Extender specifies to the SEF whether to revert to a “Ping” credit model or “stop” all trading should limits be breached).

4.2.1 Trade Breaches Credit Limit

After every trade, the Credit User's credit utilization is checked by the Limit Checker, generally this is the Clearinghouse in the Plus One Model. In this scenario, when the Credit User breaches their limit, the Limit Checker will send PartyRiskLimitUpdateReport(35=CR) to the SEF indicating the type of action to be taken (e.g. changing to a different credit model or trigger the "kill" switch). The SEF will respond to the Limit Checker with an acknowledgement message.

The diagram below illustrates the scenario where once the Limit Checker has determined the Credit User has breached their limit, the Limit Checkers sends a PartyRiskLimitUpdateReport to change the credit check model for the Credit User (other parameters may also be changed) or send a PartySuspensionRequest message to trigger the "kill" switch on the Credit User.

Figure 6: PlusOne to Ping model on credit breach



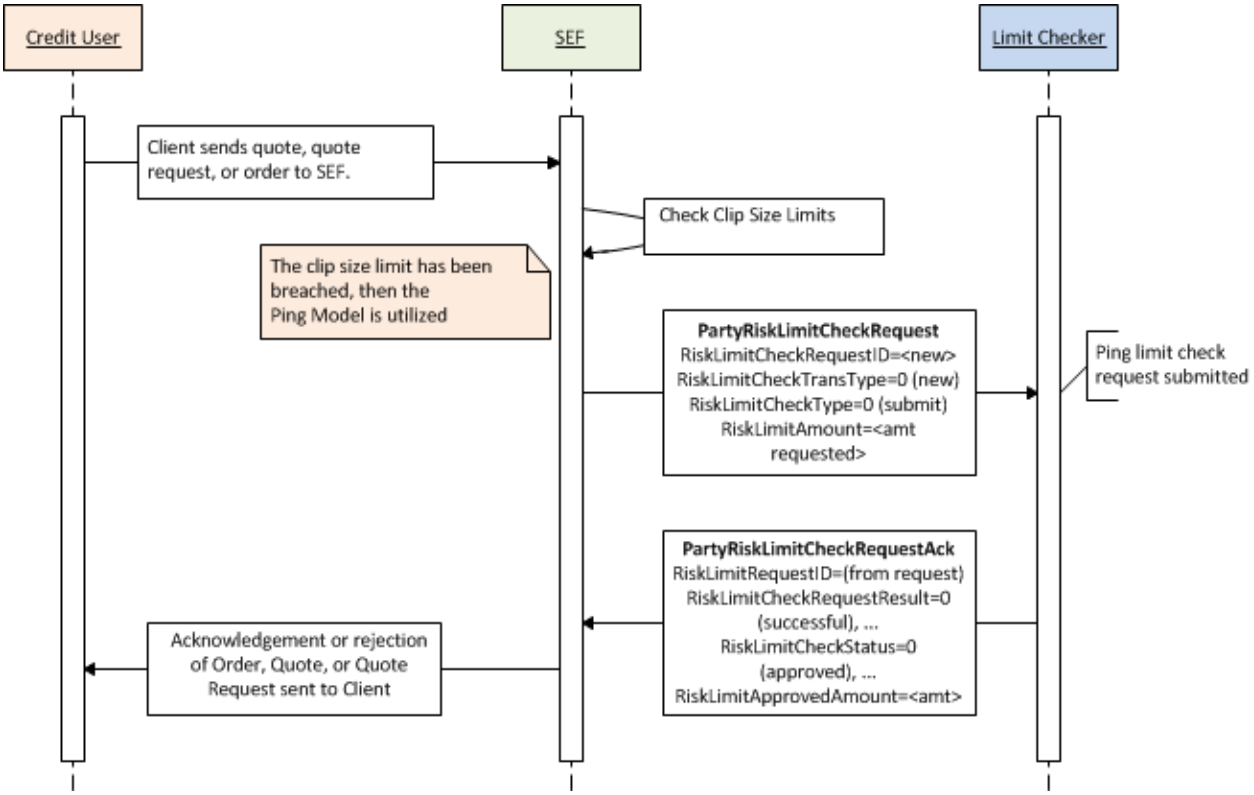
4.2.2 Clip Size Breached, Transition to Ping Model

In this scenario the Credit User's trade breaches the specified clip size. Depending on how the Credit User was setup by the Limit Checker or Credit Extender (see 4.1.1), the SEF takes the appropriate action

to either transition to either a ”Plus One to Ping” or “Plus One to Stop” credit model. The “Plus One to Stop” will cause all orders, quotes, and quote requests to be canceled. The SEF would be responsible for notifying the Credit User of cancellations of their orders, quotes, or quote requests. There is no need for the Limit Checker to send further messages to the SEF.

In the case of a transition to the Ping Model the following diagram gives an overview as to how the Ping Model will work. In this scenario, once the SEF determines that the Credit User breached their limit, the SEF would "ping" the Limit Checker for approval before displaying the order, quote or quote request to other market participants. See Ping Model section for more specifics.

Figure 7: PlusOne to Ping on Clip size breach



4.3 Ping Model

As summarized in Section 2.2, the Ping Model requires the SEF to seek approval from the Limit Checker or Credit Extender for every order, quote and quote request submitted by a Credit User. The following sections illustrate~~s~~ the various scenarios for the Ping Model and it's interaction with the quoting and trading workflows.

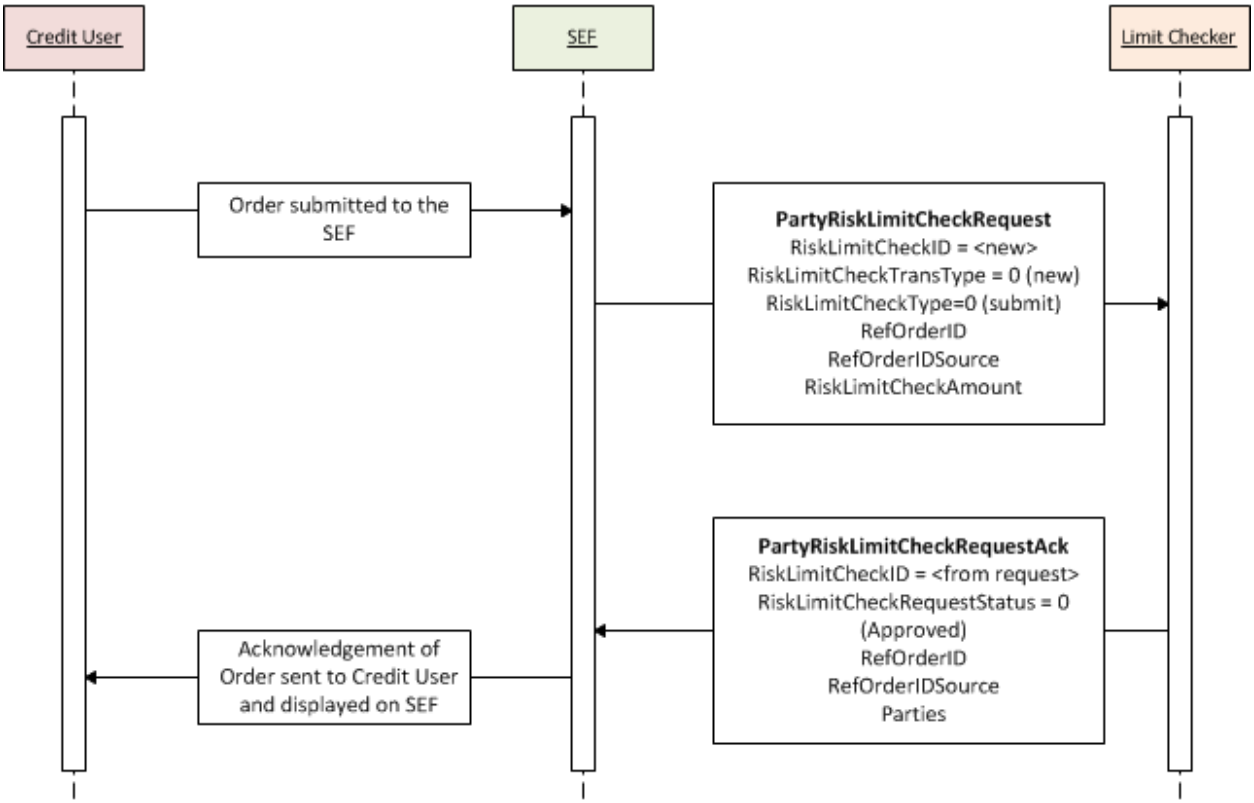
4.3.1 New Order – Credit Check Accepted in Full

This use case shows how a SEF would perform credit limit check for an order for a Credit User and receives full approval for the requested amount from the Limit Checker. (Since this gap analysis ~~-~~is not

intended to specifically discuss the order and execution report messages, these messages are shown generalized.)

The SEF sends the PartyRiskLimitCheckRequest(35=DF~~TBD~~) message to the Limit Checker (the Limit Checker may be an FCM or a Credit Hub) with reference to the order, some of the details of the order, and party to be verified. The Limit Checker will respond with PartyRiskLimitCheckRequestAck (35=DG~~TBD~~) with a result of the request and the proper references to the party and order.

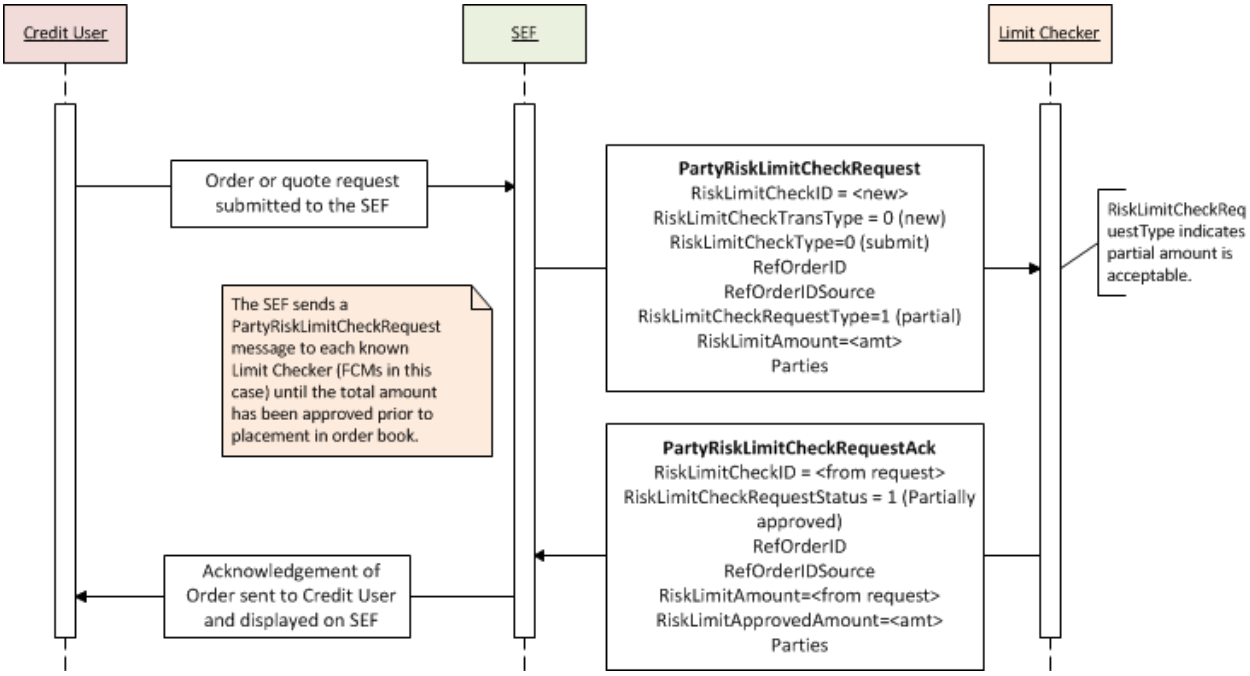
Figure 8: Ping Model credit check accepted in full



4.3.2 New Order or Quote Request – Credit Limit Check Accepted Partially

In this scenario, the SEF seeks credit approval for a submitted order for a Credit User, however, the Limit Checker only approval a partial amount. A PartyRiskLimitCheckRequest(35=DF~~TBD~~) message is sent to the Limit Checker with reference to the order or quote request, some of the details of the order, and party to be verified. The Limit Checker will respond with PartyRiskLimitCheckRequestAck(35=DG~~TBD~~) with a result of partially approved limit, the proper references to the party and order, and the amount approved. The SEF will send the PartyRiskLimitCheckRequest(35=DF~~TBD~~) message to each of the Credit User's Limit Checkers until the original requested amount is fully approved before the Credit User's order would be placed on the SEF.

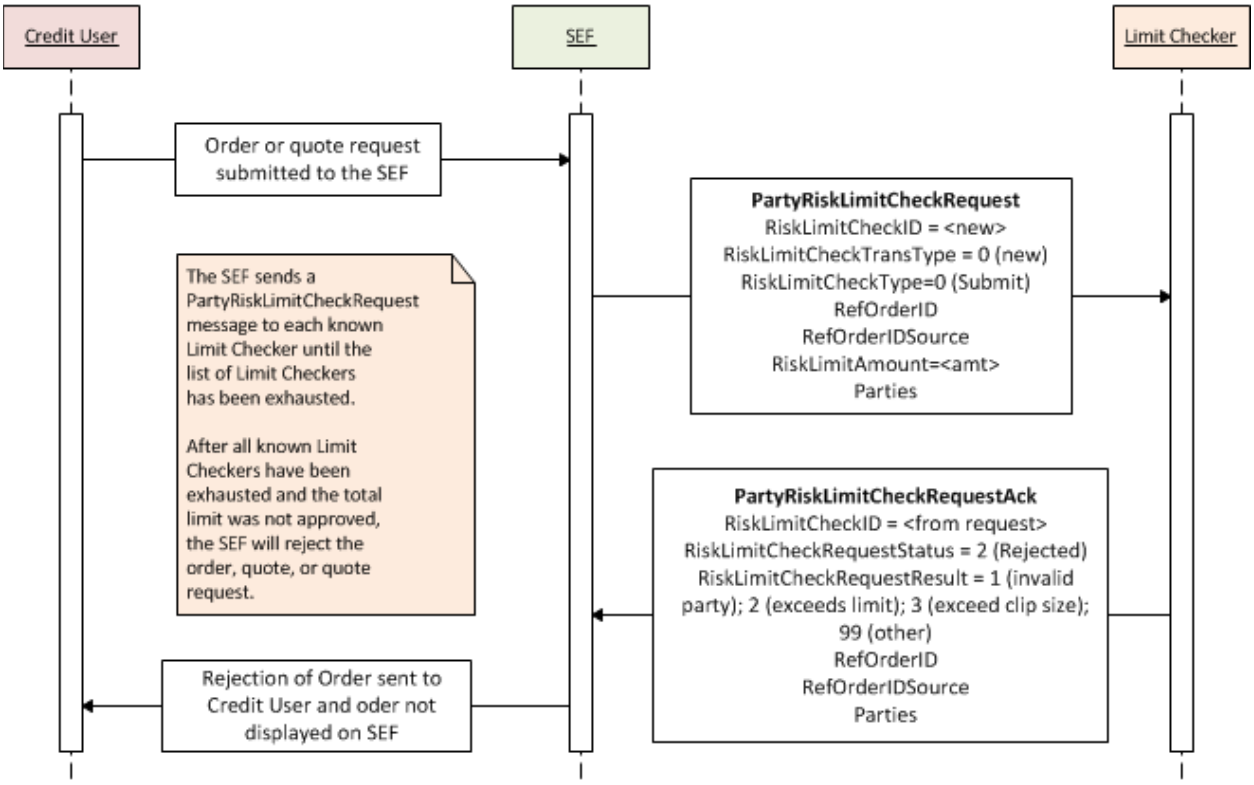
Figure 9: Ping Model credit check partially accepted



4.3.3 New Order or Quote Request– Credit Limit Check Rejected

In this scenario the SEF sends a PartyRiskLimitCheckRequest(35=DF~~TBD~~) message to the Limit Checker with references to the order, some of the details of the order or quote request, and party to be verified, seeking credit approval for the Credit User. The Limit Checker will respond with PartyRiskLimitCheckRequestAck(35=DG~~TBD~~) rejecting the credit request.

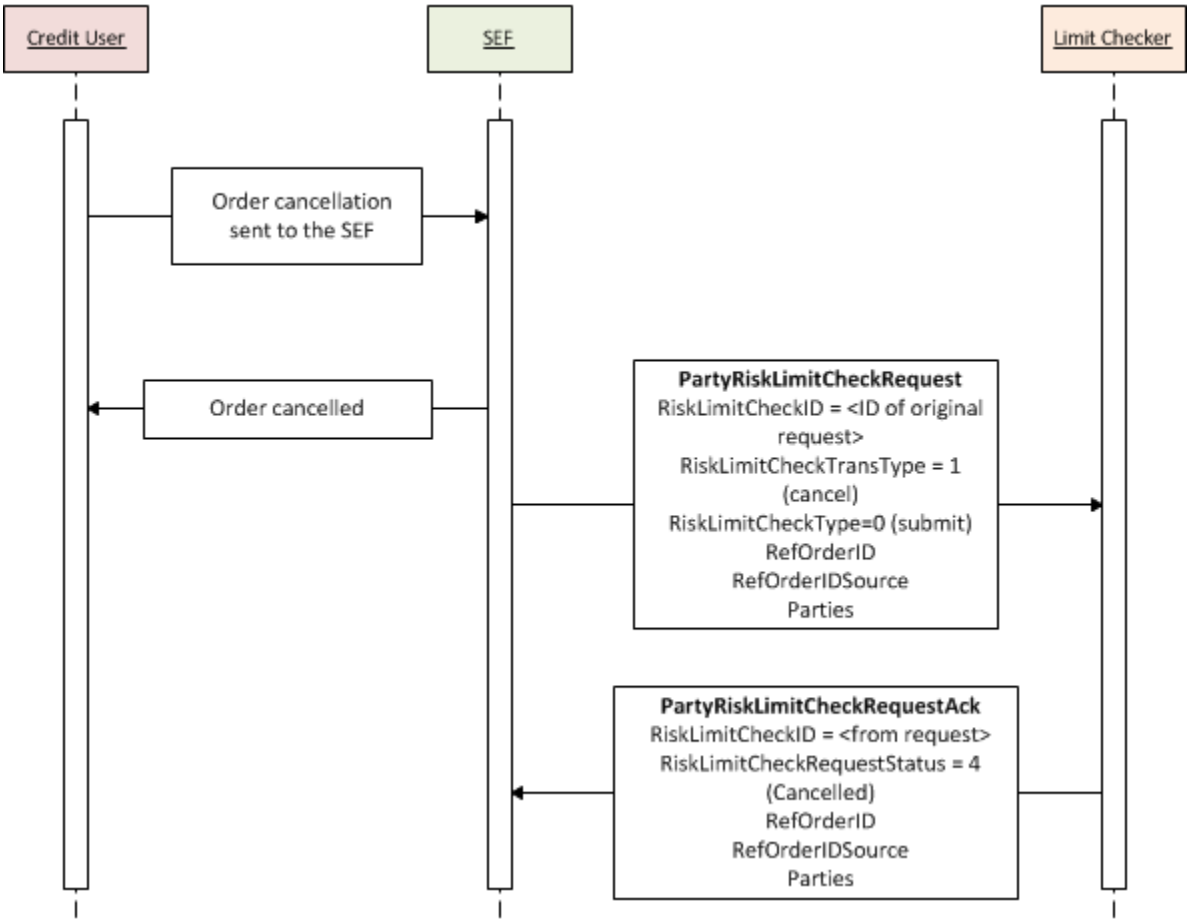
Figure 10: Ping Model credit check rejected



4.3.4 Order Canceled

When an order, quote, or quote request is canceled by the Credit User, the SEF sends a PartyRiskLimitCheckRequest(35=DF ~~TBD~~) to the Limit Checker to cancel the reserved credit for the referenced order, quote or quote request for the Credit User. The Limit Checker will respond with PartyRiskLimitCheckRequestAck(35=DG~~TBD~~) to acknowledge the reserved credit has been cancelled.

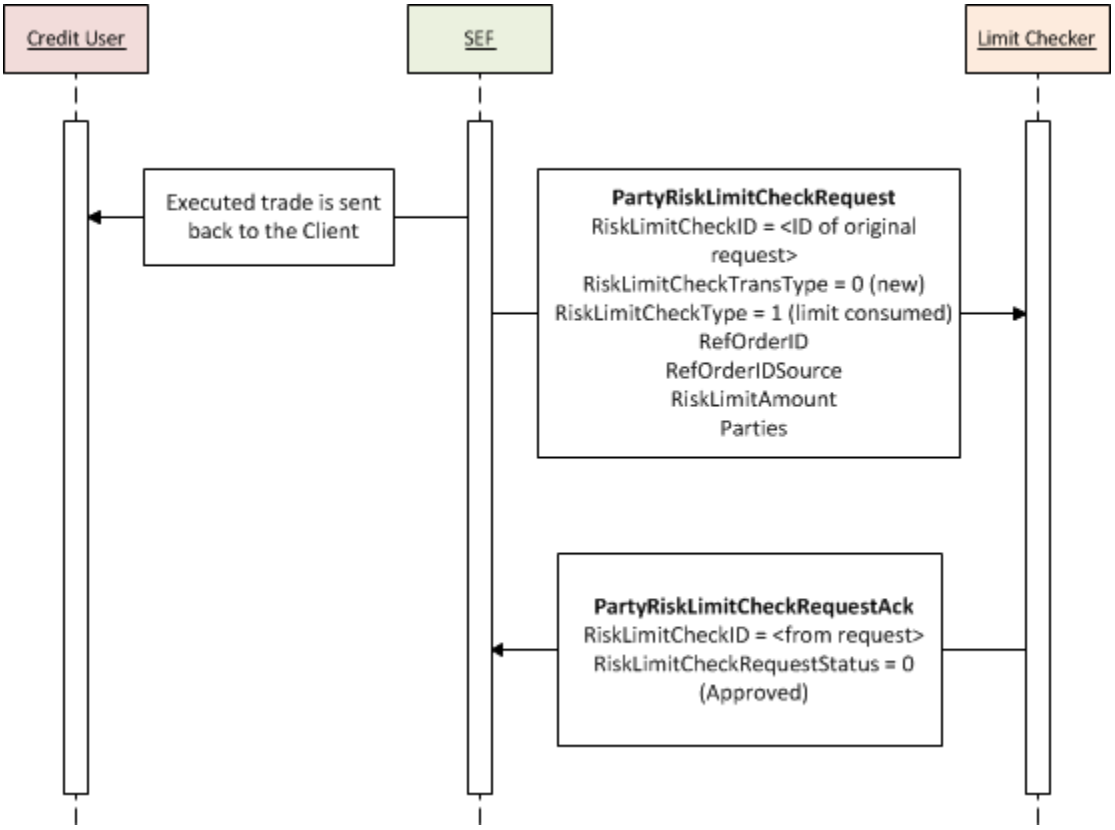
Figure 11: Ping Model order cancelled



4.3.5 Order Executed

When an order, quote, or quote request is executed, the SEF sends a PartyRiskLimitCheckRequest(35=DF~~TBD~~) to the Limit Checker to notify that the previously approved reserved credit for the Credit User has been consumed. The Limit Checker will respond with PartyRiskLimitCheckRequestAck(35=DG~~TBD~~) to acknowledge the information.

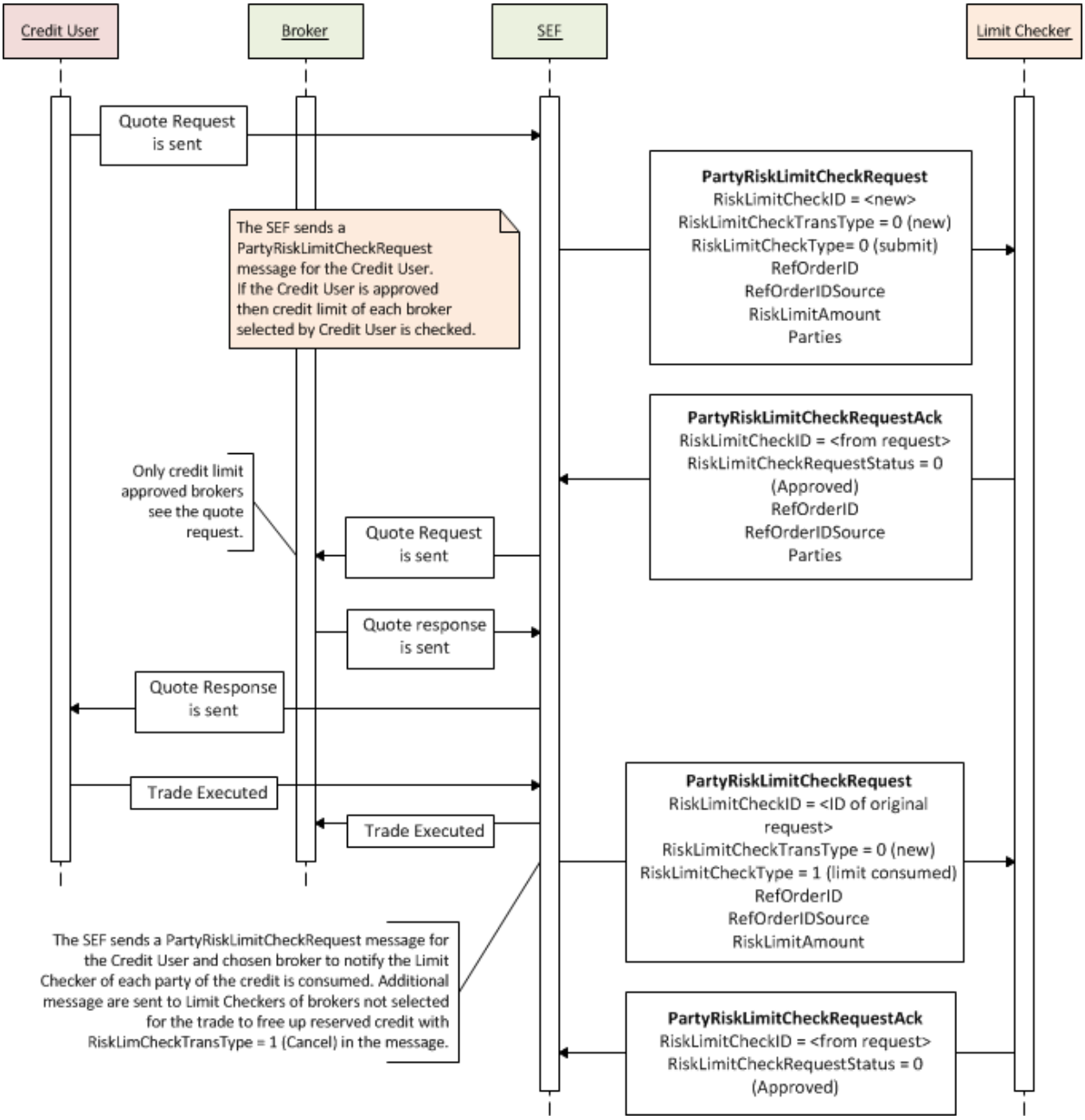
Figure 12: Ping Model order executed



4.3.6 New Quote Request – Accepted Full (Executing Broker Pre-Clearance)

In this scenario the Credit User trades on a SEF that utilizes the quote/negotiation trading model (also referred to as the "request for quote" or RFQ trading model). Under the quoting model the Credit User selects the brokers in which they would like to obtain a quote from. The SEF must do a credit check, not only of the Credit user, but also of each selected broker prior to releasing the quote request to the broker. The SEF sends a PartyRiskLimitCheckRequest(35=DF~~TBD~~) to the Limit Checker for the Credit User. If Credit User's the quote request's quantity limit is approved then the SEF will send a PartyRiskLimitCheckRequest(35=DF~~TBD~~) to the Limit Checker(s) for each of the brokers identified in the quote request by the Credit User. Once all the credit check results are obtained the SEF will send the quote request to all approved brokers. Once the trade is executed, the SEF sends a PartyRiskLimitCheckRequest(35=DF~~TBD~~) releasing the credit for all brokers that was not part of the trade, and additional PartyRiskLimitCheckRequest(35=DF~~TBD~~) messages are sent to the Limit Checker(s) of the two parties in the trade that reserved credit has been consumed by the Credit User and the broker.

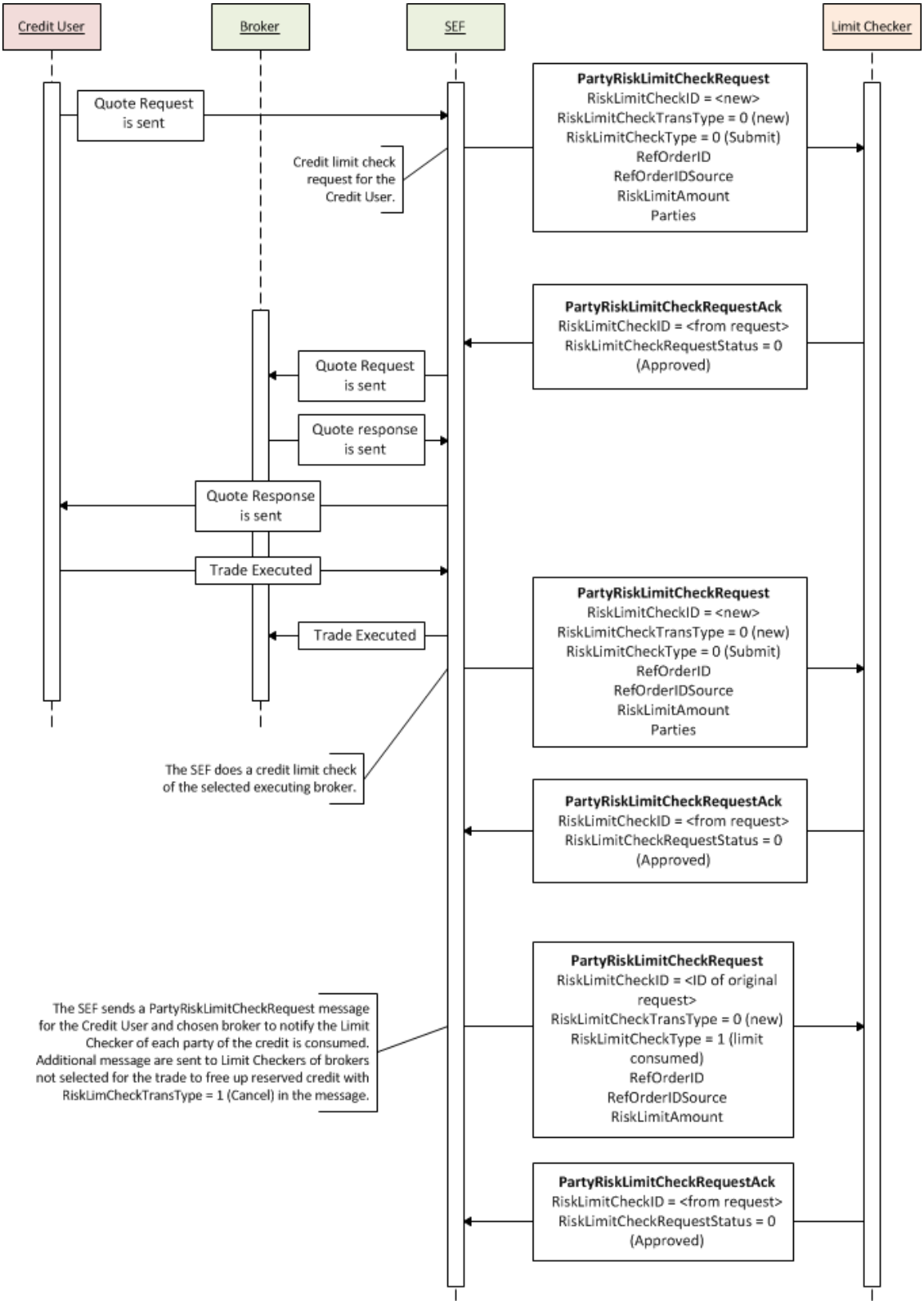
Figure 13: Ping Model quote request accepted in full, executing broker credit checked pre-trade



4.3.7 New Quote Request – Accepted Full (Executing Broker Cleared when post-trade)

This scenario is similar to the one described in Section 4.3.6 but with the difference in when the credit limit check for the selected broker is conducted. In this scenario the credit limit check for the broker selected as a result of a quote request, and resulting in a trade, is done after the trade is done. The same pre-trade credit check process as in Section 4.3.6 is done for the Credit User prior to displaying of the quote request to the candidate brokers.

Figure 14: Ping Model quote request accepted in full, executing broker credit checked post-trade

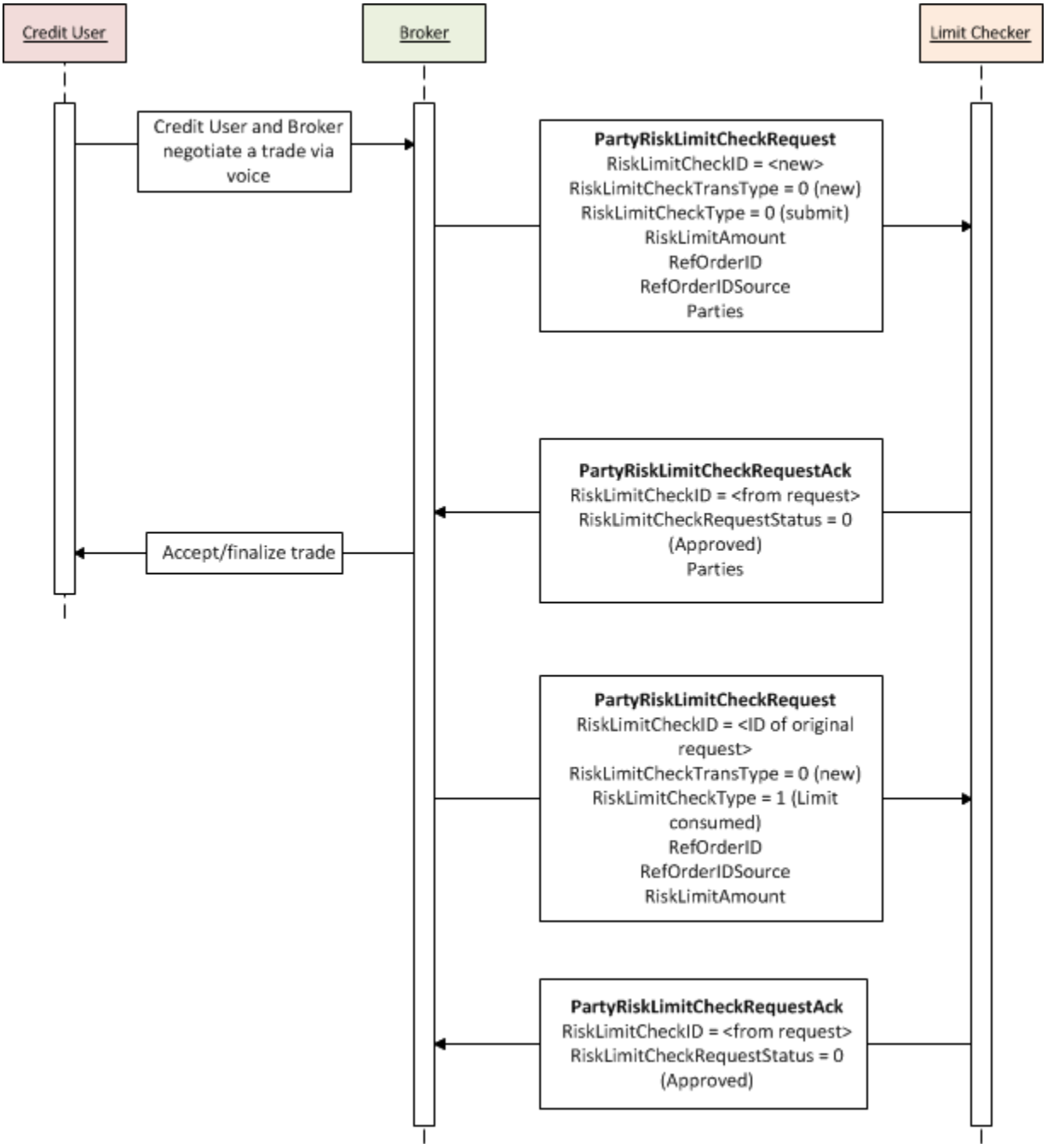


4.3.8 Voice Approval

This use case is a voice trade that is conducted away from a SEF and the credit limit is checked via a hub service provider or a CCP that provides such service for voice trades. After the terms of the trade ~~is~~ are negotiated between a Credit User (e.g. client) and the execution broker, the executing broker will perform a credit limit check with the Limit Checker (either a CCP or a hub service in this use case). The Limit Checker will check the credit limit for the Credit User and executing broker. The Credit User's requested quantity may be checked by the Limit Checker against more than one FCM if the first FCM would not approve the full amount. Once the Limit Checker informs the executing broker that both partie~~y~~s' credits are approved, the trade is accepted by the executing broker. The Limit Checker is informed of the consumed credit.

NOTE: the use case document from the FIA/ISDA Joint Working Group seems to indicate that it is the executing broker who sends a credit limit check request message to the Limit Checker for both parties. There is no indicatione in the use case description that the Credit User would send such a message. However, the FIX messages would not prevent a scenario where the Credit User sends such a request message to the Limit Checker.

Figure 15: Ping Model off-SEF negotiated transaction



4.4 Push Model

Under the Push credit check model, when the Limit Checker is the SEF, the SEF will have the credit information pushed to it from the Credit Extender stored locally in its own systems. The SEF simply has to internally check the Credit User's available limits prior to putting the quote request, quote or order into the market. It would also need to track the amount of credit consumed by the Credit User. Once the Credit User's credit limit is breached the then SEF will have to resort to whatever credit model was

initially setup by the Credit Extender for the Credit User - this may be reverting to a Ping model or to stop the Credit User from trading completely. In the case where the Limit Checker is a credit Hub provider, then the SEF will need to "ping" the Hub with limit check requests for approval prior to putting the quote request, quote or order into the market. The Credit Hub provider would be pushed the credit limit information of Credit Users to its systems by the Credit Extender.

Once the credit limits have been set up at the limit checker (SEF or Hub), message flows described in earlier sections will be used in the execution of orders, quotes, and quote requests.

These flows are all covered by previously described message flows.

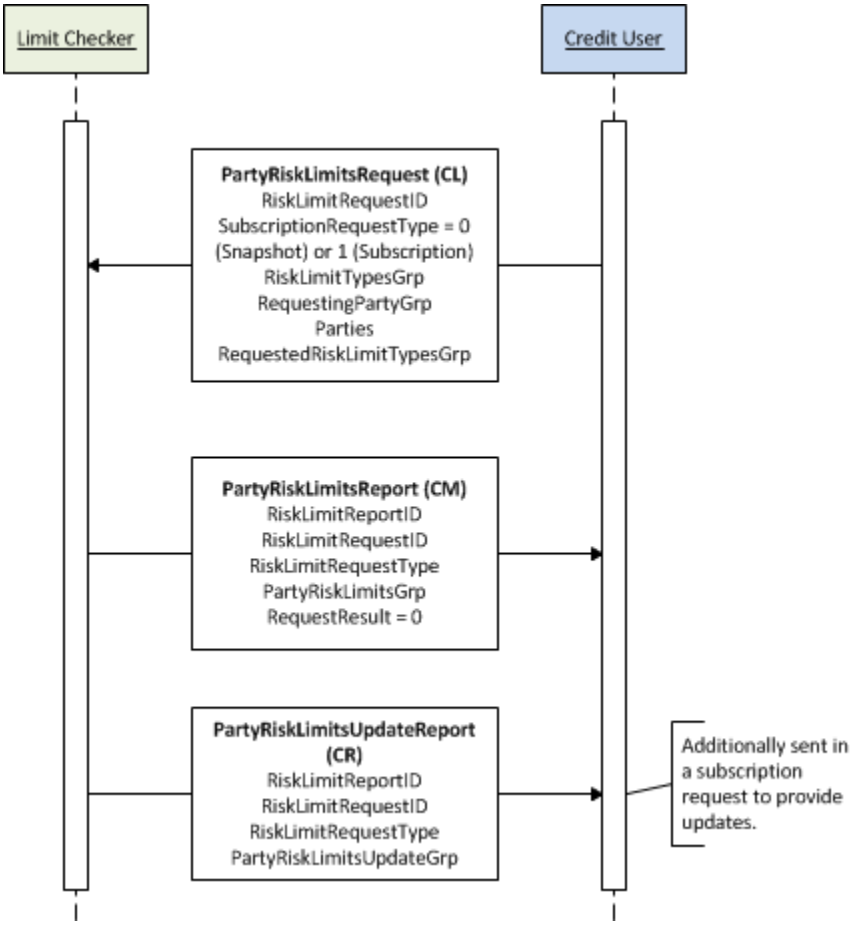
4.5 Fuel Gauge

Credit Users (clients or FCMs) may request from the Limit Checkers or Credit Extenders (CCP, FCM or Hub) how much available credit remains and how much credit have been consumed. The Credit User can request a snapshot or a subscription feed of their credit information.

4.5.1 Request/Subscribe

A PartyRiskLimitsRequest(35=CL) will be sent from the Credit User to the Limit Checker with a SubscriptionRequestType(263) of snapshot (0) or Subscription (1). The PartyRiskLimitsReport(35=CM) will be sent back to the Credit User with their amount of available and consumed credit limits in the RiskLimitTypesGrp. In the case of a subscription request, the PartyRiskLimitUpdateReport(35=CR) will be sent by the Limit Checker to the Credit User when there are any incremental updates to the credit limit information.

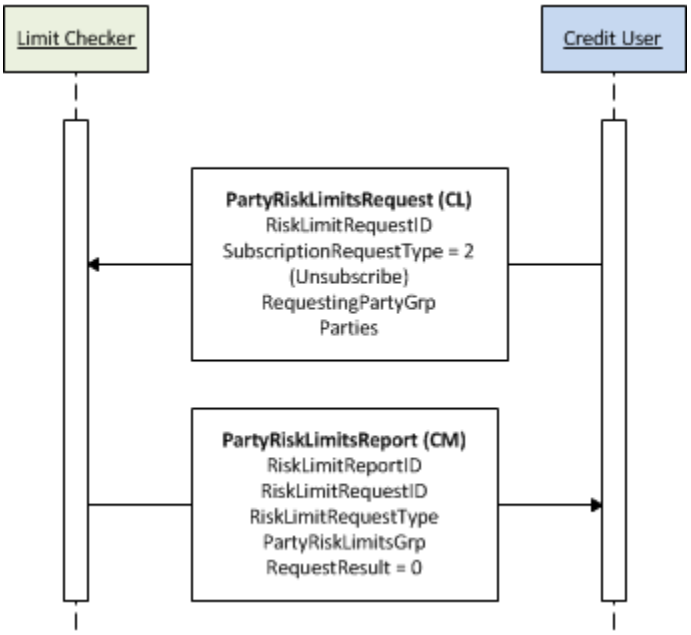
Figure 16: Subscription request



4.5.2 Cancel Subscription

A PartyRiskLimitsRequest(35=CL) will be sent from the Credit User to the Limit Checker with a SubscriptionRequestType(263)=2 (Unsubscribe). The PartyRiskLimitsReport(35=CM) will be sent back to the Credit User as a response confirming cancelling of the subscription with RequestResult(1511)=0 (Valid request).

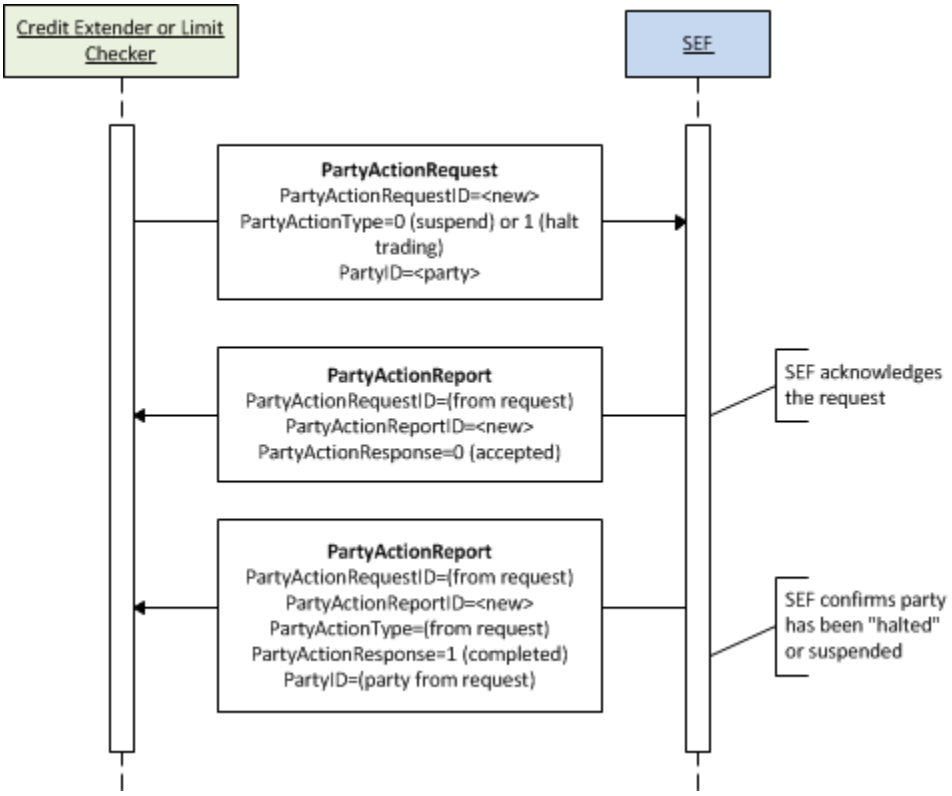
Figure 17: Subscription termination



4.6 Firm level Trade Halt

The firm level trading halt can be activated by a Credit Extender or a Limit Checker with the SEF to stop a Credit User from further trading. The PartyActionRequest(35=DH~~TBD~~) message will be sent from the party requesting the "halt trading" or "suspend" (CCP, FCM or Hub), identifying the party via the PartyDetailGrp. Upon receiving the "halt trading" or "suspend" message the SEF has to acknowledge the request with the PartyActionReport(35=DI~~TBD~~). Depending on market rules, ~~t~~if the request is to "halt trading" the identified party, the SEF may immediately make every effort to cancel all open or outstanding orders, quotes, and/or quote requests submitted by the party being halted and reject all future submissions.

Figure 18: "Halt trading" for a specified party



4.7 Application Test

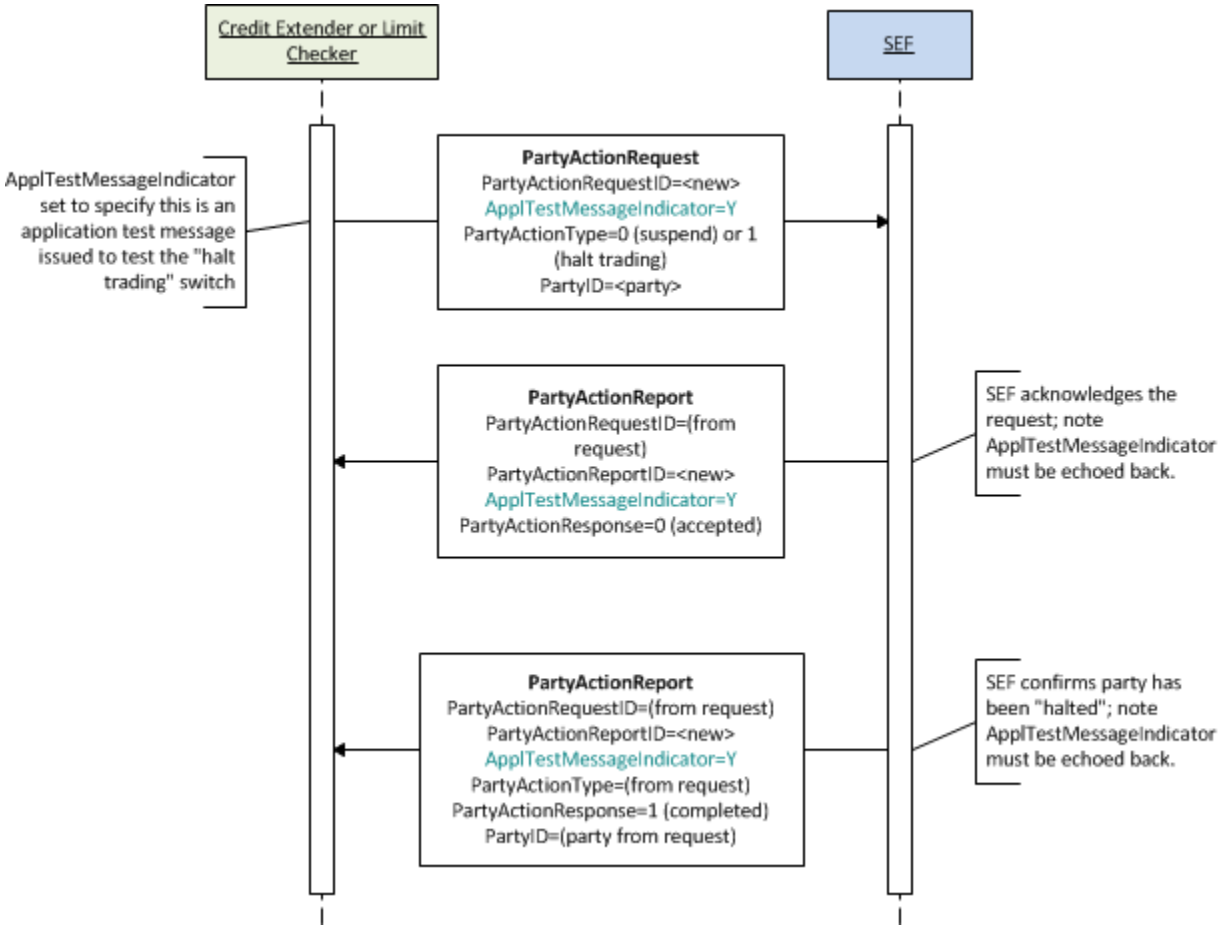
A requirement of the FIA/ISDA Joint Working Group is a requirement for the ability for the Credit Extender or Limit Checker to "ping" or request a test of the "halt trading" capability at the SEF. This requirement is utilized to periodically test the "halt trading" switch to ensure that when a real one is sent to "halt" a Credit User that it would be properly executed.

To meet this requirement a new field ApplTestMessageIndicator(2230~~tbd~~) is being added to the appropriate messages at the main level of the message. This allows a request message to be an application test message and the rec~~e~~ipient of the message must process the message as if it were a real request and respond appropriately to the success of the request in the corresponding response message.

In the case of the FIA/ISDA Join~~t~~g Working Group requirements, the ApplTestMessageIndicator(2230~~tbd~~) will be added to the PartyActionRequest(35=DH~~TBD~~) and PartyActionReport(35=DI~~TBD~~) messages at the main level of the message. When the Credit Extender or Limit Checker sends the PartyActionRequest(35=DH~~TBD~~) message with the ApplTestMessageIndicator(2230~~tbd~~)=Y this indicates to the SEF receiving the message that this is a test message for the action specified, in this case a "halt trading" of the identified party as specified by PartyID(448) set to the party being shut off from further trading, and PartyAction~~Request~~Type(2239~~tbd~~)=1 (halt trading).

The diagram below illustrates this flow and the responses from the SEF.

Figure 19: Application test message



5 FIX Message Tables

5.1 PartyRiskLimitsReportAck(35=DETBD)

To be completed at the time of the proposal – all information provided will be stored in the repository	
Message Name	PartyRiskLimitsReportAck
Message Abbreviated Name (for FIXML)	PtyRiskLmtReptAck
Category	PartiesReferenceData
Action	New
Message Synopsis	PartyRiskLimitsReportAck is an optional message used as a response to the ~~the~~ PartyRiskLimitReport(35=CM) or PartyRiskLimitUpdateReport(35=CR) messages to acknowledge or reject those messages.
Message Elaboration	

To be finalized by FPL Technical Office	
(MsgType(tag 35) Enumeration	DE
Repository Component ID	141

[Other additional text detailing usage of the message may be entered here]

Tag	*Field Name*	*Req'd*	*XMLName*	*FIX Spec Comments*	***Action***	***Mappings and Usage Comments***
StandardHeader		Y	BaseHeader	MsgType=DE~~TBD~~		**35=DE~~TBD~~**
~~***Component <ApplicationSequenceControl>***~~		~~***N***~~	~~***ApplSeqCtrl***~~			
1667	RiskLimitReportID	Y	RptID	The identifier of the PartyRiskLimitReport(35=CM~~tbd~~) or PartyRiskLimitUpdateReport(35=CR~~tbd~~) message.		
1666	RiskLimitRequestID	N	ReqID			
2316 ~~TBD~~	RiskLimitReportStatus	Y	RptStat	~~Status of the risk limit report~~		**0 = Accepted** **1 = Rejected**
2317 ~~TBD~~	RiskLimitReportRejectReason	N	RejRsn	Conditionally required when RiskLimitReportStatus(2316~~tbd~~)=1 (Rejected).		
Component <PartyRiskLimitsUpdateGrp>		***N***	***PtyRiskLmtUpdt***			
60	TransactTime	N	TxnTm			
1328	RejectText	N	RejTxt			
1664	EncodedRejectTextLen	N	EncRejTxtLen	Must be set if EncodedRejectText(1655) field is specified and must immediately precede it.		
1665	EncodedRejectText	N	EncRejTxt	Encoded (non-ASCII characters) representation of the RejectText(1328) field in the encoded format specified via the MessageEncoding(347) field.		
58	Text	N	Txt			
354	EncodedTextLen	N	EncTxtLen	Must be set if EncodedText(355) field is specified and must immediately precede it.		
355	EncodedText	N	EncTxt	Encoded (non-ASCII characters) representation of the Text(58) field in the encoded format		

				specified via the MessageEncoding(347) field.		
StandardTrailer		Y	Trlr			

5.2 PartyRiskLimitCheckRequest (35=DF~~TBD~~)

To be completed at the time of the proposal – all information provided will be stored in the repository	
Message Name	PartyRiskLimitCheckRequest
Message Abbreviated Name (for FIXML)	PtyRiskLmtChkReq
Category	PartiesAction~~ReferenceData~~
Action	New
Message Synopsis	PartyRiskLimitCheckRequest is used to request for approval of credit or risk limit amount intended to be used by a party in a transaction from another party that holds the information.
Message Elaboration	
To be finalized by FPL Technical Office	
(MsgType(tag 35) Enumeration	DF
Repository Component ID	142

[Other additional text detailing usage of the message may be entered here]

Tag	*Field Name*	*Req'd*	*XMLName*	*FIX Spec Comments*	***Action***	***Mappings and Usage Comments***
StandardHeader		Y	BaseHeader	MsgType=DF~~TBD~~		35=DF~~TBD~~
2318 ~~tbd~~	RiskLimitCheckRequestID	N	ChkReqID	Either RiskLimitCheckRequestID(2318~~tbd~~) or RiskLimitCheckID(2319~~tbd~~) must be specified. RiskLimitCheckRequestID(2318~~tbd~~) is conditionally required in a message-chaining model in which a subsequent message may refer to a prior message via RiskLimitCheckRequestRefID(2322~~tbd~~). The alt~~n~~ernative is an entity-based model in which RiskLimitCheckID(2319~~tbd~~) is used to statically identify a given request.		**Use to identify this request message.**

				In this case RiskLimitCheckID(2319~~tbd~~) is required and RiskLimitRequestID(1666~~tbd~~) can be optionally specified.		
2319 ~~tbd~~	RiskLimitCheckID	N	LmtChkID	Either RiskLimitCheckRequestID(2318~~tbd~~) or RiskLimitCheckID(2319~~tbd~~) must be specified.		**Used to identify at the business entity level the static identifier for the request.**
2320 ~~TBD~~	RiskLimitCheckTransType	Y	TransTyp			**0 = New** **1 = Cancel** **2 = Replace**
2321 ~~TBD~~	RiskLimitCheckType	Y	ChkTyp			**0 = Submit** **1 = Limit consumed**
2322 ~~TBD~~	RiskLimitCheckRequestRefID	N	ReqRefID	Conditionally required when RiskLimitCheckTransType(2320~~tbd~~) = 1 (Cancel) or 2 (Replace), and message-chaining model is used.		
1080	RefOrderID	N	RefOrdID	Used to specify the transaction reference for this limit check request.		
1081	RefOrderIDSource	N	RefOrdIDSrc	Identifies the type of reference specified in RefOrderID(1080) for this limit check request.		
2323 ~~TBD~~	RiskLimitCheckRequestType	N	ChkReqTyp			**Whether requested amount has to be approved in full or partial is acceptable.** **0 = All or none (default)** **1 = Partial**
2324 ~~TBD~~	RiskLimitCheckAmount	N	LmtChkAmt	Specifies the amount being requested or consumed, as indicated by RiskLimitCheckType(2321~~tbd~~).		
15	Currency	N	Ccy			
1670	RiskLimitID	N	RiskLmtID			
Component <RequestingPartyGrp>		***N***	***ReqPty***	***May be used to identify the party making the limit check request and their role.***		
Component <Parties>		***N***	***Pty***	***May be used to specify the trading party on which the limit check request is for. Each request is for a single trading party and the specified transaction reference.***		

Component <RelatedPartyDetailGrp>		***N***	***ReltdPtyDetl***			
Component <Instrument>		***N***	***Instrmt***			
Component <LegOrdGrp>		***N***	***Ord***			
Component <UndInstrm~~n~~tGrp>		***N***	***Undly***			
54	Side	N	Side			
60	TransactTime	N	TxnTm			
58	Text	N	Txt			
354	EncodedTextLen	N	EncTxtLen	Must be set if EncodedText(355) field is specified and must immediately precede it.		
355	EncodedText	N	EncTxt	Encoded (non-ASCII characters) representation of the Text(58) field in the encoded format specified via the MessageEncoding(347) field.		
StandardTrailer		Y	Trlr			

5.3 PartyRiskLimitCheckRequestAck (35=DG~~TBD~~)

To be completed at the time of the proposal – all information provided will be stored in the repository	
Message Name	PartyRiskLimitCheckRequestAck
Message Abbreviated Name (for FIXML)	PtyRiskLmtChkReqAck
Category	PartiesAction~~ReferenceData~~
Action	New
Message Synopsis	PartyRiskLimitCheckRequestAck is used to acknowledge a PartyRiskLimitCheckRequest(35=DF~~TBD~~) message and to respond whether the limit check request was approved or not. When used to accept the PartyRiskLimitCheckRequest(35=DF~~TBD~~) message the R~~r~~espondent may also include the limit amount that was approved.
Message Elaboration	
To be finalized by FPL Technical Office	
(MsgType(tag 35) Enumeration	DG
Repository Component ID	143

[Other additional text detailing usage of the message may be entered here]

Tag	*Field Name*	*Req'd*	*XMLName*	*FIX Spec Comments*	***Action***	***Mappings and Usage Comments***

StandardHeader		Y	BaseHeader	MsgType=DG~~TBD~~		**35=DG~~TBD~~**
2318 ~~tbd~~	RiskLimitCheckRequestID	N	ChkReqID	Either RiskLimitCheckRequestID(2318~~tbd~~) or RiskLimitCheckID(2319~~tbd~~) must be provided from the request message.		
2319 ~~tbd~~	RiskLimitCheckID	N	LmtChkID	Either RiskLimitCheckRequestID(2318~~tbd~~) or RiskLimitCheckID(2319~~tbd~~) must be provided from the request message.		
2325 ~~TBD~~	RiskLimitCheckRequestStatus	Y	ReqStat			**0 = Approved** **1 = Partially approved** **2 = Rejected** **3 = Approval pending** **4 = Cancelled**
2326 ~~TBD~~	RiskLimitCheckRequestResult	N	ReqRslt			**0 = Successful (default)** **1 = Invalid party** **2 = Requested amount exceeds overall limit** **3 = Requested amount exceeds clip size** **99 = Other**
2320 ~~TBD~~	RiskLimitCheckTransType	Y	TransTyp	Identifies the RiskLimitCheckTransType(2320~~tbd~~) this message is responding to as specified in the request message.		**0 = New** **1 = Cancel** **2 = Replace**
2321 ~~TBD~~	RiskLimitCheckType	Y	ChkTyp	Identifies the RiskLimitCheckType(2321~~tbd~~) this message is responding to as specified in the request message.		**0 = Submit** **1 = Limit consumed**
2322 ~~TBD~~	RiskLimitCheckRequestRefID	N	ReqRefID	Conditionally required when RiskLimitCheckTransType(2320~~tbd~~) = 1 (Cancel) or 2 (Replace)		
1328	RejectText	N	RejTxt			
1664	EncodedRejectTextLen	N	EncRejTxtLen	Must be set if EncodedRejectText(1665) field is specified and must immediately precede it.		
1665	EncodedRejectText	N	EncRejTxt	Encoded (non-ASCII characters) representation of the RejectText(1328) field in the encoded		

				format specified via the MessageEncoding(347) field.		
1080	RefOrderID	N	RefOrdID			
1081	RefOrderIDSource	N	RefOrdID Src			
54	Side	N	Side			
2327 ~~TBD~~	RiskLimitApproved Amount	N	LmtAprvd Amt	Conditionally required when RiskLimitCheckRequest Status(2325~~tbd~~)=1 (Partially approved)		
2324 ~~TBD~~	RiskLimitCheckAmount	N	LmtChkAmt			
1670	RiskLimitID	N	RiskLmtID			
15	Currency	N	Ccy			
126	ExpireTime	N	ExpireTm	Optionally used to specify when the approved credit limit being reserved will expire.		
Component <RequestingPartyGrp>		***N***	***ReqPty***			
Component <Parties>		***N***	***Pty***	***The trading party identified in the limit check request.***		
Component <RelatedPartyDetailGrp>		***N***	***ReltdPtyDetl***			
Component <Instrument>		***N***	***Instrmt***			
Component <LegOrdGrp>		***N***	***Ord***			
Component <UndInstrm~~n~~tGrp>		***N***	***Undly***			
60	TransactTime	N	TxnTm			
58	Text	N	Txt			
354	EncodedTextLen	N	EncTxtLen	Must be set if EncodedText(355) field is specified and must immediately precede it.		
355	EncodedText	N	EncTxt	Encoded (non-ASCII characters) representation of the Text(58) field in the encoded format specified via the MessageEncoding(347) field.		
StandardTrailer		Y	Trlr			

5.4 PartyActionRequest(35=DH~~TBD~~)

To be completed at the time of the proposal – all information provided will be stored in the repository

Message Name	PartyActionRequest
Message Abbreviated Name (for FIXML)	PtyActReq
Category	PartiesActionReferenceData
Action	New
Message Synopsis	The PartyActionRequest message is used suspend or "kill" the specified party from further trading activities at the Respondent. The Respondent must respond with a PartyActionReport(35=DI~~TBD~~) message.
Message Elaboration	
To be finalized by FPL Technical Office	
(MsgType(tag 35) Enumeration	DH
Repository Component ID	144

[Other additional text detailing usage of the message may be entered here]

Tag	*Field Name*	*Req'd*	*XMLName*	*FIX Spec Comments*	***Action***	***Mappings and Usage Comments***
StandardHeader		Y	BaseHeader	MsgType=DH~~TBD~~		**35=DH~~TBD~~**
2328 ~~TBD~~	PartyActionRequestID	Y	ActnReqID			
2329 ~~TBD~~	PartyActionType	Y	ActnTyp			**0 = Suspend** **1 = Halt trading** **2 = Reinstate**
2330 ~~TBD~~	ApplTestMessageIndicator	N	ApplTstMsgInd			
Component <RequestingPartyGrp>		***N***	***ReqPty***	***May be used to identify the party making the request and their role.***		
Component <Parties>		***N***	***Pty***	***Used to specify the trading party on which the action is applied to.***		
Component <RelatedPartyDetailGrp>		***N***	***ReltdPtyDetl***			
60	TransactTime	N	TxnTm			
58	Text	N	Txt			
354	EncodedTextLen	N	EncTxtLen	Must be set if EncodedText(355) field is specified and must immediately precede it.		
355	EncodedText	N	EncTxt	Encoded (non-ASCII characters) representation of the Text(58) field in the encoded format specified via the MessageEncoding(347) field.		

StandardTrailer	Y	Trlr			

5.5 PartyActionReport(35=DI~~TBD~~)

To be completed at the time of the proposal – all information provided will be stored in the repository	
Message Name	PartyActionReport
Message Abbreviated Name (for FIXML)	PtyActRpt
Category	PartiesAction~~ReferenceData~~
Action	New
Message Synopsis	Used to respond to the PartyActionRequest(35=DH~~tbd~~) message, indicating whether the request has been received, accepted or rejected.
Message Elaboration	
To be finalized by FPL Technical Office	
(MsgType(tag 35) Enumeration	DI
Repository Component ID	145

[Other additional text detailing usage of the message may be entered here]

Tag	*Field Name*	*Req'd*	*XMLName*	*FIX Spec Comments*	***Action***	***Mappings and Usage Comments***
StandardHeader		Y	BaseHeader	MsgType=DI~~TBD~~		**35=DI~~TBD~~**
2328 ~~TBD~~	PartyActionRequestID	N	ActnReqID	Conditionally required when responding to a PartyActionRequest(35=DH~~tbd~~) message.		
2331 ~~TBD~~	PartyActionReportID	Y	ActnRptID			
2329 ~~TBD~~	PartyActionType	Y	ActnTyp			
2332 ~~TBD~~	PartyActionResponse	Y	ActnRsp			**0 = Accepted** **1 = Completed** **2 = Rejected**
2333 ~~TBD~~	PartyActionRejectReason	N	RejRsn	Conditionally required when PartyActionResponse(2332~~tbd~~) = 2 (Rejected).		**0 = Invalid party(-ies)** **1 = Unknown requesting party** **98 = Not authorized** **99 = Other**
2330 ~~TBD~~	ApplTestMessageIndicator	N	ApplTstMsgInd	Conditionally required if present in the PartyActionRequest(35=		

				DH~~TBD~~) message.		
1328	RejectText	N	RejTxt	~~Reason description for rejecting the transaction request.~~		
1664	EncodedRejectTextLen	N	EncRejTxtLen	Must be set if EncodedRejectText(1665) field is specified and must immediately precede it.		
1665	EncodedRejectText	N	EncRejTxt	Encoded (non-ASCII characters) representation of the RejectText(1328) field in the encoded format specified via the MessageEncoding(347) field.		
Component <RequestingPartyGrp>		***N***	***ReqPty***	***May be used to identify the party making the request and their role.***		
Component <Parties>		***N***	***Pty***	***Used to specify the trading party on which the action is applied to. If in response to PartyActionRequest(35=DH~~tbd~~) message, this should echo back the values from the request.***		
Component <RelatedPartyDetailGrp>		***N***	***ReltdPtyDetl***			
60	Transacttime	N	TxnTm			
58	Text	N	Txt			
354	EncodedTextLen	N	EncTxtLen	Must be set if EncodedText(355) field is specified and must immediately precede it.		
355	EncodedText	N	EncTxt	Encoded (non-ASCII characters) representation of the Text(58) field in the encoded format specified via the MessageEncoding(347) field.		
797	CopyMsgIndicator	N	CopyMsgInd			
StandardTrailer		Y	Trlr			

5.6 ExecutionReport(35=8)

Add RefRiskLimitCheckID(2334~~tbd~~) and RefRiskLimitCheckIDType(2335~~tbd~~) with values 0= RiskLimitRequestID(1666~~tbd~~); 1 = RiskLimitCheckID(2319~~tbd~~) to allow linkage between the approved credit message and the fills. This addition allows for the linking of a fill report to the credit check approval in the Ping Model.

To be completed at the time of the proposal – all information provided will be stored in the repository	
Message Name	ExecutionReport
Message Abbreviated Name (for FIXML)	*(no change)*
Category	*(no change)*
Action	Change
Message Synopsis	*(no change)*
Message Elaboration	*(no change)*
To be finalized by FPL Technical Office	
(MsgType(tag 35) Enumeration	8
Repository Component ID	9

Tag	*Field Name*	*Req'd*	*XMLName*	*FIX Spec Comments*	***Action***	***Mappings and Usage Comments***
StandardHeader		Y	BaseHeader	MsgType=8		
<....truncated...>						
551	OrigCrossID	N				
549	CrossType	N				
2334 ~~TBD~~	RefRiskLimitCheckID	N	RefRiskLmtChkID		**NEW**	
2335 ~~TBD~~	RefRiskLimitCheckIDType	N	RefRiskLmtChkIDTyp	Conditionally required when RefRiskLimitCheckID(2334~~tbd~~) is specified.	**NEW**	**0 = RiskLimitRequestID** **1 = RiskLimitCheckID**
<…truncated…>						
StandardTrailer		Y	Trlr			

6 FIX Component Blocks

6.1 Component RiskLimitTypesGrp

To be completed at the time of the proposal – all information provided will be included in the repository	
Component Name	RiskLimitTypesGrp
Component Abbreviated Name (for FIXML)	RiskLmtTyp
Component Type	_X__ Block Repeating ___ Block
Category	PartiesReferenceData
Action	Change
Component Synopsis	Repeating group of risk limit types and values.
Component Elaboration	
To be finalized by ~~int~~FPL Technical Office	
Repository Component ID	2161

Component FIXML Abbreviation: <RiskLimitTypesGrp>								
Tag	*Field Name*			*Req'd*	*XMLName*	*FIX Spec Comments*	***Action***	***Mappings and Usage Comments***
1529	NoRiskLimitTypes			N				
→	→	***1530***	***RiskLimitType***	N	Typ	Required if NoRiskLimitTypes(1529) > 0.	**Change**	**New Enumeration Values** **9~~Tbd~~ = Limit consumed** **10~~Tbd~~ = Clip size**
→	→	***1531***	***RiskLimitAmount***	N	Amt			
→	→	***1767***	***RiskLimitAction***	N	Actn		**Change**	**New Enumeration Values** **5~~tbd~~ = Ping with revalidation** **6~~tbd~~ = Ping without revalidation** **7~~tbd~~ = Push with revalidation** **8~~tbd~~ = Push without revalidation** **9~~tbd~~ = Suspend** **10~~tbd~~ = Halt trading**
→	→	***1766***	***RiskLimitUtilizationAmount***	N	UtilztnAmt	Not applicable in a request.		
→	→	***1765***	***RiskLimitUtilizationPercent***	N	UtilztnPct	Not applicable in a request.		

→	→	***1532***	***RiskLimitCurrency***	N	Ccy			
→	→	***1533***	***RiskLimitPlatform***	N	Pltfm			
→	→	***2336 ~~TBD~~***	***RiskLimitVelocitPeriody***	N	Velcty	Conditionally required when RiskLimitType(1530) = 10~~bd~~ (Clip size)	**NEW**	
→	→	***2337 ~~TBD~~***	***RiskLimitVelocityUnit***	N	VelctyUnit		**NEW**	
→	→	***RiskWarningLevelGrp***		N	WarnLvl			
</RiskLimitTypesGrp>								

6.2 Component RequestingPartyGrp

To be completed at the time of the proposal – all information provided will be included in the repository	
Component Name	RequestingPartyGrp
Component Abbreviated Name (for FIXML)	ReqPty
Component Type	_X__ Block Repeating ___ Block
Category	~~PartiesReferenceData~~Common
Action	Change
Component Synopsis	Identifies the party making the request.
Component Elaboration	
To be finalized by ~~int~~FPL Technical Office	
Repository Component ID	2180

Component FIXML Abbreviation: <RequestingPartyGrp>							
Tag	*Field Name*		*Req'd*	*XMLName*	*FIX Spec Comments*	***Action***	***Mappings and Usage Comments***
1657	NoRequestingPartyIDs		N				
→	***1658***	***RequestingPartyID***	N	ID	Required when NoRequestingPartyIDs > 0.		

→	***1659***	***RequestingPartyIDSource***	N	Src	Required when NoRequestingPartyIDs > 0.		
→	***1660***	***RequestingPartyRole***	N	R	Required when NoRequestingPartyIDs > 0.		
→	***2338 ~~TBD~~***	***RequestingPartyRoleQualifier***	N	Qual		**NEW**	**For PartyRole = Intermediary. Used to specify that the requesting party is an intermediary hub system.** **8~~TBD~~ = Hub [Hub]**
→	***RequestingPartySubGrp***		N	Sub			
</RequestingPartyGrp>							

6.3 Component PartyDetailGrp

To be completed at the time of the proposal – all information provided will be included in the repository	
Component Name	PartyDetailGrp
Component Abbreviated Name (for FIXML)	PtyDetl
Component Type	_X__ Block Repeating ___ Block
Category	PartiesReferenceData
Action	Change
Component Synopsis	Contains details for a party, including related parties and alternative party identifiers.
Component Elaboration	
To be finalized by ~~int~~FPL Technical Office	
Repository Component ID	2156

Component FIXML Abbreviation: <PartyDetailGrp>							
Tag	*Field Name*		*Req'd*	*XMLName*	*FIX Spec Comments*	***Action***	***Mappings and Usage Comments***
1671	NoPartyDetails		N				
→	***1691***	***PartyDetailID***	N	ID	The identification of the party. Required when NoPartyDetails(1671) > 0.		
→	***1692***	***PartyDetailIDS***	N	Src	Used to identify source		

		ource			of PartyID value (e.g. BIC). Required when NoPartyDetails(1671) > 0.		
→	***1693***	***PartyDetailRole***	N	R	Identifies the type of PartyID (e.g. Executing Broker). Required when NoPartyDetails(1671) > 0.		
→	***1674***	***PartyDetailRoleQualifier***	N	Qual			
→	***PartyDetailSubGrp***		N	Sub			
→	***PartyDetailAltIDGrp***		N	AltPty	Optionally used to specify alternate IDs to identify the party specified.		
→	***RelatedPartyDetailGrp***		N	ReltdPtyDetl	May not be specified in PartyDetailsListUpdateReport(35=CK) if ListUpdateAction(1324) = D(Delete)		
→	***1672***	***PartyDetailStatus***	N	Stat	~~Specifies the status of the party information, whether active, or suspended (inactive) or "halted".~~	**Change**	**Add to enums new value:** **2~~tbd~~ = "Halted"**
</PartyDetailGrp>							

6.4 Component PartyRiskLimitsGrp

To be completed at the time of the proposal – all information provided will be included in the repository	
Component Name	PartyRiskLimitsGrp
Component Abbreviated Name (for FIXML)	PtyRiskLmt
Component Type	___ Block Repeating _X__ Block
Category	PartiesReferenceData
Action	Change
Component Synopsis	
Component Elaboration	
To be finalized by ~~int~~FPL Technical Office	

Repository Component ID	2184

Component FIXML Abbreviation: <PartyRiskLimitsGrp>							
Tag	*Field Name*		*Req'd*	*XMLName*	*FIX Spec Comments*	***Action***	***Mappings and Usage Comments***
1677	NoPartyRiskLimits		N				
→	***PartyDetailGrp***		N	PtyDelt	Required if NoPartyRiskLimits(1677) > 0.		
→	***RiskLimitsGrp***		N	RiskLmt	Required if NoPartyRiskLimits(1677) > 0. Omit to implicitly report removal of risk limits.		
→	1670	RiskLimitID	N	ID			
→	2339 ~~tbd~~	RiskLimitCheckModelType	N	ChkModelTyp		**NEW**	
</PartyRiskLimitsGrp>							

6.5 Component PartyRiskLimitsUpdateGrp

To be completed at the time of the proposal – all information provided will be included in the repository	
Component Name	PartyRiskLimitsUpdateGrp
Component Abbreviated Name (for FIXML)	PtyRiskLmtUpdt
Component Type	___ Block Repeating _X__ Block
Category	PartiesReferenceData
Action	Change
Component Synopsis	
Component Elaboration	
To be finalized by ~~int~~FPL Technical Office	
Repository Component ID	2193

Component FIXML Abbreviation: <PartyRiskLimitsUpdateGrp>							
Tag	*Field Name*		*Req'd*	*XMLName*	*FIX Spec Comments*	***Action***	***Mappings and Usage Comments***
1677	NoPartyRiskLimits		N				
→	1324	ListUpdateAction	N	ListUpdActn	Required if NoPartyRiskLimits(1677) > 0		
→	***PartyDetailGrp***		N	PtyDelt	Conditionally required when ListUpdateAction(1324) = A(Add). Conditionally required when ListUpdateAction(1324) = M(Modify) or D(Delete) and RiskLimitID(1670) is not provided.		
→	***RiskLimitsGrp***		N	RiskLmt	Conditionally required when ListUpdateAction(1324) = A(Add) or M(Modify).		
→	1670	RiskLimitID	N	ID	Conditionally required when PartyDetailGrp component is not provided.		
→	2339 ~~tbd~~	RiskLimitCheckModelType	N	ChkModelTyp		**NEW**	
</PartyRiskLimitsUpdateGrp>							

6.6 Component PartyRiskLimitsAckGrp

To be completed at the time of the proposal – all information provided will be included in the repository	
Component Name	PartyRiskLimitsAckGrp
Component Abbreviated Name (for FIXML)	PtyRiskLmtAck
Component Type	_X__ Block Repeating ___ Block
Category	PartiesReferenceData
Action	Change
Component Synopsis	
Component Elaboration	

To be finalized by ~~int~~FPL Technical Office	
Repository Component ID	2194

Component FIXML Abbreviation: <PartyRiskLimitsAckGrp>							
Tag	*Field Name*		*Req'd*	*XMLName*	*FIX Spec Comments*	***Action***	***Mappings and Usage Comments***
1677	NoPartyRiskLimits		N				
→	1324	ListUpdateAction	N	ListUpdActn	Required if NoPartyRiskLimits(1677) > 0		
→	1763	RiskLimitStatus	N	Stat	Required if NoPartyRiskLimits(1677) > 0		
→	1764	RiskLimitResult	N	Rslt			
→	***PartyDetailGrp***		N	PtyDelt	Conditionally required when RiskLimitID(1670) is not provided. Changes to party or related party(-ies) defined in the request are not permitted.		
→	***RiskLimitsGrp***		N	RiskLmt	Conditionally required when RiskLimitStatus(1763) = 1(Accepted with changes) and must then be complete, i.e. omissions compared to the request represent risk limits that were removed, additional risk limits are possible.		
→	1670	RiskLimitID	N	ID	Conditionally required when PartyDetailGrp component is not provided.		
→	2339 ~~tbd~~	RiskLimitCheckModelType	N	ChkModelTyp		**NEW**	
→	1328	RejectText	N	RejTxt			
→	1664	EncodedRejectTextLen	N	EncRejTxtLen	Must be set if EncodedRejectText(1665) field is specified and must immediately precede it.	**CHANGE**	
→	1665	EncodedRejectText	N	EncRejTxt	Encoded (non-ASCII characters) representation of the RejectText(1328)	**CHANGE**	

					field in the encoded format specified via the MessageEncoding field.		
</PartyRiskLimitsAckGrp>							

6.7 Component LegOrdGrp

To be completed at the time of the proposal – all information provided will be included in the repository	
Component Name	LegOrdGrp
Component Abbreviated Name (for FIXML)	Ord
Component Type	X Implicit Block Repeating Block
Category	Common
Action	Change
Component Synopsis	
Component Elaboration	
To be finalized by FPL Technical Office	
Repository Component ID	2025

6.8 Component PartyRelationshipGrp

To be completed at the time of the proposal – all information provided will be included in the repository	
Component Name	PartyRelationshipGrp
Component Abbreviated Name (for FIXML)	Rltnshp
Component Type	X Implicit Block Repeating Block
Category	Common
Action	Change
Component Synopsis	Repeating group of party relationships.
Component Elaboration	
To be finalized by FPL Technical Office	
Repository Component ID	2154

6.9 Component RelatedPartyDetailGrp

To be completed at the time of the proposal – all information provided will be included in the repository	
Component Name	RelatedPartyDetailGrp
Component Abbreviated Name (for FIXML)	ReltdPtyDetl
Component Type	X Block Repeating Block
Category	Common
Action	Change
Component Synopsis	Party details for parties related to the Party specified in the PartyDetailGrp.
Component Elaboration	
To be finalized by FPL Technical Office	
Repository Component ID	2166

6.10 Component RelatedPartyDetailSubGrp

To be completed at the time of the proposal – all information provided will be included in the repository	
Component Name	RelatedPartyDetailSubGrp
Component Abbreviated Name (for FIXML)	ReltdPtyDetl
Component Type	X Implicit Block Repeating Block
Category	Common
Action	Change
Component Synopsis	Party details for parties related to the Party specified in the PartyDetailGrp.
Component Elaboration	
To be finalized by FPL Technical Office	
Repository Component ID	2167

6.11 Component RelatedPartyDetailAltIDGrp

To be completed at the time of the proposal – all information provided will be included in the repository	
Component Name	RelatedPartyDetailAltIDGrp
Component Abbreviated Name (for FIXML)	AltPty
Component Type	X Implicit Block Repeating Block
Category	Common
Action	Change
Component Synopsis	Alternative identifiers for parties related to the party specified in the PartyDetailGrp.
Component Elaboration	
To be finalized by FPL Technical Office	
Repository Component ID	2168

6.12 Component RelatedPartyDetailAltIDSubGrp

To be completed at the time of the proposal – all information provided will be included in the repository	
Component Name	RelatedPartyDetailAltIDSubGrp
Component Abbreviated Name (for FIXML)	AltPty
Component Type	X Implicit Block Repeating Block
Category	Common
Action	Change
Component Synopsis	Sub identifiers for related parties alternate identifiers.
Component Elaboration	
To be finalized by FPL Technical Office	
Repository Component ID	2169

6.13 Component RequestingPartySubGrp

To be completed at the time of the proposal – all information provided will be included in the repository	
Component Name	RequestingPartySubGrp
Component Abbreviated Name (for FIXML)	Sub
Component Type	_X_ Block Repeating ___ Block
Category	Common
Action	Change
Component Synopsis	Sub identifiers for the requesting party.
Component Elaboration	
To be finalized by FPL Technical Office	
Repository Component ID	2181

7 Category Changes

To be completed at the time of the proposal – all information provided is stored in the repository	
Category Name	PartiesAction
Section	_X_ PreTrade __ Trade __ PostTrade __ Infrastructure
Category Synopsis	The PartiesAction category of messages is a set of messages that are used to take an action on party information as a result of risk management decisions made during the trading day.
Category Elaboration	[enter the category elaboration here]
To be finalized by FPL Technical Office	
Category Filename	

Appendix A - Data Dictionary

Tag	Field Name	Action	Data Type	Description	FIXML Abbreviation	Add to / Deprecate from Message type or Component block
2316 ~~TBD~~	RiskLimitReportStatus	New	Int	Status of risk limit report Valid values are: 0 = Accepted 1 = Rejected	@RptStat	Add to message: PartyRiskLimitsReportAck
2317 ~~TBD~~	RiskLimitReportReject Reason	New	int	The reason for rejecting the PartyRiskLimitsReport(35=CM) or PartyRiskLimitsUpdateReport(35=CR). Valid values: 0 = Unknown RiskLimitReportID(1667) 1 = Unknown party 99 = Other	@RejRsn	Add to message: PartyRiskLimitsReportAck
2318 ~~TBD~~	RiskLimitCheckRequestID	New	String	The unique identifier of the PartyRiskLimitCheckRequest(35=DF~~TBD~~) message.	@ChkReqID	Add to message: PartyRiskLimitCheckRequest PartyRiskLimitCheckRequestAck
2319 ~~TBD~~	RiskLimitCheckID	New	String	The unique and static identifier, at the business entity level, of a risk limit check request.	@LmtChkID	Add to message: PartyRiskLimitCheckRequest PartyRiskLimitCheckRequestAck
2320 ~~TBD~~	RiskLimitCheckTransType	New	int	Specifies the transaction type of the risk limit check request. Valid values: 0 = New 1 = Cancel 2 = Replace	@TransTyp	Add to message: PartyRiskLimitCheckRequest PartyRiskLimitCheckRequestAck
2321 ~~TBD~~	RiskLimitCheckType	New	int	Specifies the type of limit check message. Valid values: 0 = Submit (Elaboration: Indicates a submission for a limit check. The RiskLimitCheckTransType(2320~~tbd~~) indicates whether the submission is a new request, a cancel or replace/amend of a prior	@ChkTyp	Add to message: PartyRiskLimitCheckRequest PartyRiskLimitCheckRequestAck

				submission.) 1 = Limit consumed (Elaboration: Indicates that the limit reserved by a prior request has been used or consumed by a transaction that occurrued.)		
2322 TBD	RiskLimitCheckRequestRefID	New	String	Specifies the message reference identifier of the risk limit check request message.	@ReqRefID	Add to message: PartyRiskLimitCheckRequest PartyRiskLimitCheckRequestAck
2323 TBD	RiskLimitCheckRequestType	New	int	Specifies the type of limit amount check being requested. Valid value: 0 = All or none (default if not specified) (Elaboration: The limit check request is for the full amount requested or none at all. Request can only be responded to with a full approval of the amount requested or a rejection of the request.) 1 = Partial (Elaboration: The requester will accept a partial approval of the requested credit limit amount.)	@ChkReqTyp	Add to message: PartyRiskLimitCheckRequest
2324 TBD	RiskLimitCheckAmount	New	~~Qty~~Amt	Specifies the amount being requested for approval.	@LmtChkAmt	Add to message: PartyRiskLimitCheckRequest PartyRiskLimitCheckRequestAck
2325 TBD	RiskLimitCheckRequestStatus	New	int	Indicates the status of the risk limit check request. Valid values: 0 = Approved (Elaboration: Request has been accepted and processed. The credit amount requested has been reserved for the transaction.) 1 = Partially approved (Elaboration: Only a partial amount of the credit amount requested has been approved and has been reserved for the transaction.) 2 = Rejected	@ReqStat	Add to message: PartyRiskLimitCheckRequestAck

				3 = Approval pending 4 = Cancelled		
2326 ~~TBD~~	RiskLimitCheckRequestResult	New	int	Result of the credit limit check request. Valid values: 0 = Successful (default) 1 = Invalid party(-ies) 2 = Requested amount exceeds credit limit 3 = Requested amount exceeds clip size limit 4 = Request exceeds maximum notional order amount 99 = Other	@ReqRslt	Add to message: PartyRiskLimitCheckRequestAck
2327 ~~TBD~~	RiskLimitApprovedAmount	New	~~Qty~~Amt	The credit/risk limit amount approved.	@LmtAprvdAmt	Add to message: PartyRiskLimitCheckRequestAck
2328 ~~TBD~~	PartyActionRequestID	New	String	The unique identifier of the PartyActionRequest(35=DH~~TBD~~) message.	@ActnReqID	Add to message: PartyActionRequest PartyActionReport
2329 ~~TBD~~	PartyActionType	New	int	Specifies the type of action to take or was taken for a given party. Valid values: 0 = Suspend 1 = Halt trading 2 = Reinstate	@ActnTyp	Add to message: PartyActionRequest PartyActionReport
2330 ~~TBD~~	ApplTestMessageIndicator	New	Boolean	Used to indicate whether the message being sent is to test the receiving application's availability to process the message. When set to "Y" the message is a test message. If not specified, the message is by default not a test message.	@ApplTstMsgInd	Add to message: PartyActionRequest PartyActionReport
2331 ~~TBD~~	PartyActionReportID	New	String	The unique identifier of the PartyActionReport(35=DI ~~TBD~~) message as assigned by the message sender.	@ActnRptID	Add to message: PartyActionReport
2332 ~~TBD~~	PartyActionResponse	New	int	Specifies the action taken as a result of the PartyActionType(2239~~tbd~~) of the PartyActionRequest(35=DH~~tbd~~) message. Valid values: 0 = Accepted (Elaboration: The action request is accepted	@ActnRsp	Add to message: PartyActionReport

				for processing.) 1 = Completed (Elaboration: The processing of the requested action has been successfully completed.) 2 = Rejected (Elaboration: The action request was rejected. PartyActionRejectReason(2233~~tbd~~) should be used to specify the rejection reason.)		
2333 ~~TBD~~	PartyActionRejectReason	New	int	Specifies the reason the PartyActionRequest(35=DH~~TBD~~) was rejected. Valid values: 0 = Invalid party or part(~~-~~ies~~)~~ 1 = Unknown requesting party 98 = Not authorized 99 = Other	@RejRsn	Add to message: PartyActionReport
2334 ~~TBD~~	RefRiskLimitCheckID	New	String	The reference identifier to the PartyRiskLimitCheckRequest(35=DF~~TBD~~) message that contained the approval or rejection for risk/credit limit check.	@RefRiskLmtChkID	Add to message: ExecutionReport
2335 ~~TBD~~	RefRiskLimitCheckIDType	New	int	Specifies which type of identifier is specified in RefRiskLimitCheckID(2334~~tbd~~) field. Valid values: 0 = RiskLimitRequestID(1666~~tbd~~) 1 = RiskLimitCheckID(2319~~tbd~~)	@RefRiskLmtChkIDTyp	Add to message: ExecutionReport
2336 ~~TBD~~	RiskLimitVelocityPeriod	New	Int	The time interval for which the clip size limit applies. The velocity time unit is expressed in RiskLimitVelocityUnit(2337~~tbd~~).	@Velcty	Add to Component: RiskLimitTypesGrp
2337 ~~TBD~~	RiskLimitVelocityUnit	New	String~~int~~	Unit of time in which RiskLimitVelocityPeriod(2336~~tbd~~) is expressed. *(Uses values from TimeUnit(997))*	@VelctyUnit	Add to Component: RiskLimitTypesGrp
2338 ~~TBD~~	RequestingPartyRoleQualifier	New	Int	Qualifies the value of RequestingPartyRole(1660). For PartyRole = Intermediary 8~~TBD~~ = Hub [Hub] (Elaboration: ~~i~~Indicates that the Intermediary party is a hub system or	@Qual	Add to Component: RequestingPartyGroup

				service provider.) *(Uses values from PartyDetailRoleQualifier(1674). Add new value to this field.)*		
2339 ~~TBD~~	RiskLimitCheckModelType	New	Int	Specifies the type of credit limit check model workflow to apply for the specified party. Valid values: 0 = None (default if not specified) (Elaboration: No specified limit check model is defined. Limit checks for the party will be based on parameters defined.) 1 = PlusOne model (Elaboration: A pre-trade credit limit check model which allows trades to occur until it is determined by the clearinghouse or other designated limit checker that the party's limit(s) was breached by the most recent trade executed.) 2 = Ping model (Elaboration: A pre-trade credit limit check model which requires the execution venue to obtain limit approval from the Credit Provider for every transaction about to be conducted by the Credit User.) 3 = Push model (Elaboration: A pre-trade credit limit check model in which the Credit Provider "pushes" to the execution venue the credit limit information allocated to each of the Credit Provider's counterparty or customer.)	@ChkModelTyp	Add to Components: PartyRiskLimitsGrp PartyRiskLimitsUpdateGrp PartyRiskLimitsAckGrp
35	MsgType	Change	String	Defines message type ALWAYS THIRD FIELD IN MESSAGE. (Always unencrypted) Note: A "U" as the first character in the MsgType field (i.e. U, U2, etc.) indicates that	@MsgTyp	

				the message format is privately defined between the sender and receiver. *** Note the use of lower case letters *** Valid values: 0 = Heartbeat 1 = Test request ..... DE~~TBD~~ – PartyRiskLimitsReportAck DF~~TBD~~ – PartyRiskLimitCheckRequest DG~~TBD~~ – PartyRiskLimitCheckRequestAck DH~~TBD~~ – PartyActionRequest DI~~TBD~~ – PartyActionReport		
103	OrdRejReason	Change	int	Code to identify reason for order rejection. Note: Values 3, 4, and 5 will be used when rejecting an order due to pre-allocation information errors. Valid values: 0 = Broker / Exchange option 1 = Unknown symbol .... *add new values* 25~~tbd~~ = Insufficient credit limit 26~~tbd~~ = Exceeded clip size limit 27~~tbd~~ = Exceeded maximum notional order amount 28~~tbd~~ = Exceeded DV01/PV01 limit 29 = Exceeded CS01 limit		
126	ExpireTime	Add	UTCTimestamp	*Add additional description:* Time/Date of order expiration (always expressed in UTC (Universal Time Coordinated, also known as "GMT") The meaning of expiration is specific to the context where the field is used.		Add to message: PartyRiskLimitCheckRequestAck

				For orders, this is the expiration time of a Good Til Date TimeInForce. For Quotes - this is the expiration of the quote. Expiration time is provided across the quote message dialog to control the length of time of the overall quoting process. For collateral requests, this is the time by which collateral must be assigned. For collateral assignments, this is the time by which a response to the assignment is expected. For credit/risk limit checks, this is the time when the reserved credit limit will expire for the requested transaction.		
300	QuoteRejectReason	Change	int	Reason quote was rejected. Valid values: 1 = Unknown symbol (security) 2 = Exchange (security) closed .... ***deprecate (duplicate added in EP144)*** **15 = Price exceeds current price band** *add new values* 17~~tbd~~ = Insufficient credit limit 18~~tbd~~ = Exceeded clip size limit 19~~tbd~~ = Exceeded maximum notional order amount 20~~tbd~~ = Exceeded DV01/PV01 limit 21 = Exceeded CS01 limit		
658	QuoteRequestRejectReason	Change	int	Reason quote was rejected Valid values: 1 = Unknown symbol (security) 2 = Exchange (security) closed		

				.... 11 = Insufficient credit *add new values* ~~12tbd = Insufficient credit limit~~ (not added as duplicate existing enum value 11) ~~13~~12~~tbd~~ = Exceeded clip size limit ~~14~~13~~tbd~~ = Exceeded maximum notional order amount ~~15~~14~~tbd~~ = Exceeded DV01/PV01 limit 15 = Exceeded CS01 limit		
1080	RefOrderID	Change		*Add additional description:* The ID reference to the order being hit or taken. For pre-trade credit/risk limit check process, this is the reference to the placed order, quote request or quote for the credit/risk limit check.		
1081	RefOrderIDSource	Change		*Add to the description and new enums:* Used to specify what identifier, provided in order depth market data, to use when hitting (taking) a specific order or to identify what type of order or quote reference is being provided when seeking credit limit check. 0 = SecondaryOrderID(198) 1 = OrderID(37) 2 = MDEntryID(278) 3 = QuoteEntryID(299) 4 = Original order ID 5~~tbd~~ = QuoteID(117) 6~~tbd~~ = QuoteReqID(131)		
1530	RiskLimitType	Change	Int	Used to specify the type of risk limit amount of position limit quantity or margin requirement amounts. Valid values: 1 = Gross limit	@Typ	

				2 = Net limit .... Add values: 0 = Credit limit (Elaboration: The credit limit provided by one party to another for trading.) 9~~TBD~~ = Limit consumed (Elaboration: The limit used in the recent transaction.) 10~~TBD~~ = Clip size (Elaboration: The total amount allowed to be traded within a defined period of time, or velocity. The defined period of time is specified by the RiskLimitVelocityPeriod(2336~~tbd~~) and RiskLimitVelocityUnit(2337~~tbd~~).) 11~~TBD~~ = Maximum notional order size 12~~TBD~~ = DV01/PV01 limit (Elaboration: The maximum dollar value change resulting from a move of 1 basis point in the yield curve. This limits the interest rate risk exposure. Also known as "basis point value" or BPV.) 13 = CS01 limit (Elaboration: ~~The credit spread value of one basis point. T~~Credit spread sensitivity. Represents the change in market value of a CDS for a one basis point change in the credit spread. This limits the credit risk exposure of a CDS. Also known as "risky-DV01".)		

1670	RiskLimitID	Change	String	Unique reference identifier for a specific ~~NoPartyRiskLimits(1677) repeating group instance~~ risk limit defined for the specified party.	@RiskLmtID @ID in PartiesReference Data	
1672	PartyDetailStatus	Change	int	Indicates the status of the party identified with PartyDetailID(1691). 0 = Active (default if not specified) 1 = Suspended 2~~tbd~~ = Halted	@Stat	
1767	RiskLimitAction	Change	Int	Identifies the action to take or risk model to assume should risk limit be exceeded or breached for the specified party. 5~~tbd~~ = Ping credit check model with revalidation (Elaboration: Each subsequent order, quote request or quote submission by the Credit User must obtain pre-approval. Any open orders, quote requests or quotes are to be cancelled.) 6~~tbd~~ = Ping credit check model without revalidation (Elaboration: Each subsequent order, quote request or quote submission by the Credit User must obtain pre-approval. Any open orders, quote requests or quotes will remain active.) 7~~tbd~~ = Push credit check model with revalidation (Elaboration: Each subsequent order, quote request or quote sub~~n~~mission by the Credit User must be checked against the limit amounts pushed to the trading platform. Any	@Actn	

				open orders, quote requests or quotes are to be cancelled.) 8~~tbd~~ = Push credit check model without revalidation (Elaboration: Each subsequent order, quote request or quote sub~~n~~mission by the Credit User must be checked against the limit amounts pushed to the trading platform. Any open orders, quote requests or quotes will remain active.) 9~~tbd~~ = Suspend (Elaboration: Sus~~s~~pend the Credit User from trading once limit(s) is breached. This is considered a "soft" stop.) 10~~tbd~~ = Halt trading (Elaboration: Halt or stop the Credit User from trading once limit(s) is breached. This is considered a "hard" stop and may require more involved actions to reinstate the Credit User's ability to trade.)		
1769	RiskWarningLevelAction	CHANGE		*(Uses values from RiskLimitAction(1767))*		

Appendix B - Glossary Entries

Term	Definition	Field where used
Clip size	The amount (or clip) threshold under the velocity limit. The total amount allowed to be traded within a defined period of time. For example a clip and velocity limit may be defined as "$100 million per 10 minutes" the clip size is $100 million and the velocity of the clip is 10 minutes. See also "velocity".	RiskLimitType
Credit Extender Credit Provider Credit Source	An entity that provides credit to another party with whom they have a contractual relationship for the purposes of facilitating the clearing of that party's trades. For example, FCMs provide credit to end users; CCPs provide credit to FCMs.	
Credit Hub	A third party entity that provides credit limit check services on behalf of market participants.	
Credit Limit	A maximum value, which could be of various types (e.g. initial margin, dv01/cs01, notional, etc.), that an entity is willing to provide to another party with whom they have a contractual relationship for the purposes of clearing. The limit extended by the Credit Extender to the Credit User on a specific SEF or for specific, or classes of, instruments. This amount maybe a smaller portion of a Credit User's overall allowed risk limit set by the Credit Extender.	
Credit Model	The method jointly employed by the Credit Extender and Limit Checker for a given Credit User through which the credit value of a trade is verified to be within the credit limit prior to the placement of an order and the execution of a trade. the FIA/ISDA Joint Working group currently defined 3 pre-execution models: 1) Push; 2) Ping and 3) Plus One	
Credit User	An entity that receives credit from a Credit Provider and leverages that credit to clear trades intraday. For example: an asset manager with limits at an FCM; a FCM with limits at a CCP.	
Limit Checker	An entity that employs a systematic method to determine	

	whether the credit value of a swap to be executed and cleared by a Credit User is within the acceptable maximum credit limit value provided by the Credit Extender. For example: FCM checking their client's limit utilization; CCP checking the FCM's limit utilization; CCP checking an FCM's client's limit utilization on behalf of the FCM; a Credit Hub checking an FCM's client's limit utilization on behalf of the FCM.	
Velocity	The time interval for which the clip size limit applies. For instance one may set a limit which says "$100 million per 10 minutes". The velocity of the $100 million is 10 minutes. See also "Clip size".	RiskLimitType

Appendix C - Abbreviations

Term	Proposed Abbreviation	Proposed Messages, Components, Fields where used
Approved	Aprvd	RiskLimitApprovedAmount
Check	Chk	RiskLimitCheckID
Expire	Expire	ExpireTime
Test	Tst	ApplTestMessageIndicator
Velocity	Velcty	RiskLimitVelocity

Appendix D - Usage Examples

Appendix E - Public Comment Disposition

Comment A:

Re: PUBLIC COMMENT PERIOD – Pre-Trade Credit Check Proposal
Hanno Klein / Deutsche Börse
7 Mar 2013 7:39AM ET

Review comment #1:
RiskLimitCheckTransType value 2="Limit consumed". XXXTransType fields should have purely technical values, e.g. New/Replace/Cancel. Only exception is PosTransType(709). Additional XXXType fields can capture non-technical information about the transaction, e.g. TradeReportTransType(487) in combination with TradeReportType(856) is an example of that. Suggest to add optional field RiskLimitCheckType with valid values "0=Submit" and 1="Limit consumed" and add value 2="Replace" to RiskLimitCheckTransType. RiskLimitCheckType 0="Submit" would be used to maintain credit reservations (with RiskLimitCheckTransType New/Cancel/Replace) whereas 1="Limit consumed" would consume the reserved credit (with RiskLimitCheckTransType 0=New), reverse the consumed credit (with RiskLimitCheckTransType 1=Cancel) or correct the consumed credit (with RiskLimitCheckTransType 2=Replace).

Comment disposition:

Added RiskLimitCheckType to convey "limit consumed" and "submit". However, the notion of "maintain credit reservations" is not a requirement nor a notion within the requirement stated by the joint FIA/ISDA working group. Once a limit is reserved, it is for a specified order/quote, and either it is consumed or not (cancelled). If not consumed it the reserved credit is released, not "maintained". Currently, there is also no stated business case for a "correct the consumed credit". This particular use case will probably need to be raised with the joint WG as it was not documented.

Review comment #2:
Field name RiskLimitCheckRequestMsgID. Identifiers of request messages should be XXXRequestID, e.g. RiskLimitRequestID(1666). Suggest to change name to RiskLimitCheckRequestID.

Comment disposition:

Changed field name. There is agreement that the RiskLimitCheckRequestID is the message entity identifier while the RiskLimitCheckID is the business entity identifier. There is a requirement for the more "static" business entity identifier that can be carried through to the transaction (e.g. order/quote and execution) to link the limit check with the order/quote executed. Add field usage comment to distinguish between the message entity and business entity ID in the field usage text in the message.

Review comment #3:
PartyRiskLimitCheckRequest (and other party messages) use <PartyDetailGrp> which already includes a group of related parties. PartyActionRequest has deviated from that and used <Parties> together with <RelatedPartyDetailGrp> on the same level. Suggest to align with other party messages and only use <PartyDetailGrp>.

Comment disposition:

The PartyDetailGrp contains the PartyDetailStatus field which is not needed in the PartyActionRequest. Additionally there was no requirement stated by the joint WG to be able to specify more than one party in a message to "halt trading", "reinstate" or "suspend" trading when an electronic message is sent to request the SEF to immediately take an action on a party.

Comment B:

Re: PUBLIC COMMENT PERIOD – Pre-Trade Credit Check Proposal
Hanno Klein / Deutsche Börse
15 Mar 2013 6:26AM ET

PartyActionReport currently only allows to respond to a request (PartyActionRequestID is a required field). The message should also allow an unsolicited notification of a party other than the submitter of the request, e.g. to the credit user that has been halted.

Comment disposition:

This was not a use case requirement from the joint FIA/ISDA WG, but there is potential for sending the PartyActionReport as a "drop copy" to the party halted. Added the CopyMsgIndicator field to allow for optionally indicating that the msg is a drop copy.

Comment C:

Re: PUBLIC COMMENT PERIOD – Pre-Trade Credit Check Proposal
James Crosson / Intercontinental Exchange
15 Mar 2013 12:09PM ET

Should the document cover use-cases including:

1) Ping mechanism for partial fills
2) Ping mechanisms for Order Cancel Replace Request scenarios

Comment disposition:

The gap analysis was not intended to be a "recommended practices" document. It only illustrated the use cases that was documented in the use case document published by the joint FIA/ISDA working group. That use case document did not include any scenario for ping scenarios for partially filled orders, or for OrderCancel/Replace scenarios.

The gap anaslysis document is only intended to propose changes to FIX to support the stated requirements. For the scenarios pointed out in the comment, there should not be any additional gaps in what's proposed that would not support the scenarios above.

No further action required.

Comment D:

Re: PUBLIC COMMENT PERIOD – Pre-Trade Credit Check Proposal
Hanno Klein / Deutsche Börse
20 Mar 2013 6:43AM ET

New fields in PartyActionRequest/Report messages are missing in the data dictionary and need to be added.

Comment disposition:

Data Dictionary updated accordingly.

Field name PartyActionRequestType should simply be PartyActionType to better support unsolicited PartyActionReport messages. Same approach has been used for MassActionType(1373) in the OrderMassActionRequest/Report messages.

Comment disposition:

Changed field name.

Valid values of PartyActionType should include 3=Warning to be able to use PartyActionRequest/Report to issue or convey warnings for breaching a risk limit. Add field RiskLimitID(1670) to be able to identify the limit that has been breached and that has led to an action on the party.

Comment dispositon:

The requirement was for messages that would be focused on taking the action to either suspend, shut off or reinstate the party's ability to trade, thus the reason why the PartyActionRequest/Report are currently of limited scope.

Point taken on the "warning" but isn't that already covered by the PartyRiskLimitsReport's and PartyRiskLimitsUpdateReport's RiskLimitGrp?

If this is a use case exchanges are considering could this be deferred to a subsequent gap analysis in order to flush this out further, along with any other use cases/requirements that may come up that wasn't part of the original use cases from the joint FIA/ISDA WG?

Agreed to defer the "warning" use cases to a second phase gap analysis.

Comment E:

Re: PUBLIC COMMENT PERIOD – Pre-Trade Credit Check Proposal
Hanno Klein / Deutsche Börse
20 Mar 2013 6:55AM ET

Suggest to add RiskLimitID(1670) to PartyRiskLimitCheckRequest(Ack) messages to allow to reference a specific limit that has been defined with PartyRiskLimitsDefinitionRequest.

Comment disposition:

Added.

Comment F:

Re: PUBLIC COMMENT PERIOD – Pre-Trade Credit Check Proposal
James Crosson / Intercontinental Exchange
25 Mar 2013 3:01PM ET

The PartyRiskLimitCheckRequest message does not contain a currency field for the limit being requested.

Comment disposition:

Added to the main message level.

Couldn't we re-use the RiskLimit component block for this message instead of defining distinct RiskLimitRequest fields?

Comment disposition:

Unclear about the comment above or which specific message or field(s) the comment may be referring to. If this is referring to the PartyRiskLimitCheckRequest message, this message is used to request for and reserve limit for each transaction (ping model). There is no stated requirement to specify the RiskLimitType in this message when reserving limit for a submitted order or quote.

The RiskLimitTypesGrp component is being enhanced with the different limit types specified in the requirements when setting/defining limits for the party by the FCM or CCP.

Additional clarification from commenter:

Within the Party Risk Limit Check Request message you define several fields related to the Risk Limit Request ie. Risk Limit Check Request Message ID, Risk Limit Check ID, Risk Limit Check Transaction Type, etc.
Why not just use the Risk Limits component group and instead define an enumerated value on the RiskLimitType or RiskLimitAction to qualify it as a Risk Limit Request?

Response:

These fields being referred to (the first 7 fields in the proposed message) are part of the FIX message design to identify the message, link the request for credit approval to a specific order/quote/quote request, the trans type (is this a new request, update of a prior request, cancel the prior request), and to convey whether request is for full amount specific or partial approval of the requested amount is acceptable, etc. These were all stipulated requirements from the joint FIA/ISDA WG.

The RiskLimitTypesGrp can't be used in the RiskLimitCheckRequest message because it is a repeating group, whose purpose of the RiskLimitTypesGrp is used to define limits when a client is setup, e.g. this client is allowed a clip size of 10, and an order max size of 15, has total margin limit of 100, etc.. The RiskLimitCheckRequest message is intended to request the receiving party (e.g. FCM) to either approve or not the amount of credit limit for a given transaction (order, quote, quote request from the customer). No action required.