



# Global Exchanges and Markets Committee

## NGM Logon Extension

**June 17, 2010**

**Revision 0.7**

**Proposal Status: Approved**

---

**For Global Technical Committee Governance Internal Use Only**

Submission Date:	May 1, 2010	Control Number:	EP124
Submission Status	Approved	Ratified Date	July 28, 2010
Primary Contact Person:	Mikael Brännström NGM	Release Identifier:	5.0 SP3

## **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") TO 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).

Copyright 2003-2010 FIX Protocol Limited, all rights reserved

## Table of Contents

Document History.....	5
1 Introduction .....	6
2 Business Requirements.....	6
2.1 Message Recovery .....	6
2.2 Recovery From Mismatching Sequence Numbers in Logon .....	6
2.3 Logon Reset.....	7
3 Issues and Discussion Points .....	7
3.1 Message Recovery – Gap Fill or Reset? .....	7
3.2 Recovery From Mismatching Sequence Numbers in Logon .....	8
3.3 Logon Reset.....	8
4 Proposed Message Flow .....	8
4.1 Recovery From Mismatching Sequence Numbers in Logon .....	8
4.2 Logon Reset.....	10
5 FIX message tables.....	10
5.1 Logout.....	10
5.2 Sequence Reset .....	11
6 FIX component blocks.....	11
7 Appendix A - Data Dictionary.....	12
8 Appendix B - Glossary Entries .....	13
9 Appendix C - Usage Examples.....	13
10 Appendix D – Documentation Changes .....	13

## Table of Figures

## Document History

Revision	Date	Author	Revision Comments
0.1	March 12, 2010	Mikael Brännström, NGM	Initial
0.2	April 13, 2010	Mikael Brännström, NGM	Added message scenarios
0.3	May 18, 2010	Mikael Brännström, NGM	Added proposed changes to the FIXT 1.1 document
0.4	May 20, 2010	Mikael Brännström, NGM	Cleanup of discussion points.
0.5	May 28, 2010	Mikael Brännström, NGM	Changes after public feedback and the GEMC meeting at May 27.
0.6	June 9, 2010	Mikael Brännström, NGM	Changes after the GEMC meeting at June 3.
0.7	June 17, 2010	Mikael Brännström, NGM	Minor changes after the GTC meeting at June 17.
0.7	2011-08-22	Jim N	As Built
	2012-01-11	L. Taikitsadaporn	Replaced "TBD" with assigned values, corrected missing tag reference in descriptions, regenerated TOC for publishing

## 1 Introduction

Nordic Growth Market (NGM) is requesting changes to allow the initiator to reset the `MsgSeqNum` in the Logon message (not only as part of 24-hour connectivity). The Logout message also needs to be extended with some error codes to convey that the session state is corrupt and a recovery can be initiated.

Summary of proposed changes:

- Allow field `ResetSeqNumFlag` to be set in the first Logon message.
- Update `SessionStatus` in Logout with additional error codes.
- Add field `NextExpectedMsgSeqNum` to the Logout message.
- New field `AppLevelRecoveryIndicator` is added to the Sequence Reset message.

Terminology used in this document (FIXT 1.1 Errata March 2008, page 8):

- **Initiator** establishes the telecommunications link and initiates the session via transmission of the initial Logon message.
- **Acceptor** is the receiving party of the FIX session. This party has responsibility to perform first level authentication and formally declare the connection request “accepted” through transmission of an acknowledgment Logon message.
- **FIX Connection** is comprised of three parts: logon, message exchange, and logout.
- **FIX Session** is comprised of one or more FIX Connections, meaning that a FIX Session spans multiple logins.

## 2 Business Requirements

### 2.1 Message Recovery

FIXT have a message recovery mechanism that can be triggered by either a Resend Request message or by the `NextExpectedMsgSeqNum` processing in the Logon message. In case the requested message history is not available, for example if one party only keeps a message history of X messages, the request must be gap-filled. A way of indicating that some important business messages could have been lost is needed. If this happens, it is recommended that application level recovery be initiated (e.g. request snapshots).

### 2.2 Recovery From Mismatching Sequence Numbers in Logon

A robust way of recovering from mismatching sequence numbers in a FIX session is needed.

The acceptor may detect the following sequence number mismatches and must be able to indicate this back to the initiator (e.g. in the Logout message):

- **Initiator’s `MsgSeqNum` is too low.** The acceptor expected a higher `MsgSeqNum`, i.e. the session state is corrupt.
- **Initiator’s `NextExpectedMsgSeqNum` is too high.** The acceptor has never sent a message with `MsgSeqNum` equal to `NextExpectedMsgSeqNum-1` before, i.e. the session state is corrupt.
- **Acceptor has lost the session state.** E.g. some kind of failure of the acceptor has occurred.

Note that the first two errors do not necessarily mean that the error is on the initiator side. The errors are only described as seen from the acceptor’s perspective. For example:

- The acceptor did not reset its' session as it should (e.g. at midnight) for some reason, and is unaware of this. When the initiator sends a Logon with `MsgSeqNum=1` the acceptor will complain with the first error code above.
- The acceptor did reset its' session due to some failure (e.g. was restarted) but is unaware of this. When the initiator sends a Logon to resume the session, then the acceptor will complain with the second error since `NextExpectedMsgSeqNum=1` was expected.

If the acceptor *is* aware of a session state failure on its side, then the third error case should be conveyed in some way. The reason for this differentiation is to simplify any manual investigation of why the error did occur.

## 2.3 Logon Reset

From FIXT 1.1, Errata March 2008, page 8, first paragraph:

Connecting parties must bi-laterally agree as to when sessions are to be started/stopped based upon individual system and time zone requirements.

NGM wishes to extend this with a more flexible agreement, which states that the acceptor agrees to restart the session whenever the initiator requests it. During the logon part of a FIX connection, the initiator can request the FIX session to be restarted when Logon has `ResetSeqNumFlag=Y`.

The motivation of this requirement is that otherwise the time of session restart need to be agreed on out of band and clocks need to be in sync (minor issue, but still a timing issue). Also, whenever the session is restarted the possibility to retransmit messages from the previous session is lost. For example if the initiator need to retransmit the last few messages from yesterday (e.g. due to some failure at the initiator side at end of yesterday) the initiator simply do not need to reset the session when it connects in the morning. If on the other hand, the session would be reset e.g. at midnight every day no matter what, then those messages would not be easily recoverable.

## 3 Issues and Discussion Points

### 3.1 Message Recovery – Gap Fill or Reset?

Two alternatives are proposed to the message recovery requirement when important business messages cannot be retransmitted.

Assume that a Resend Request with `BeginSeqNo=100` and `EndSeqNo=200` is received, but only messages 150 through 200 are available.

The first proposal is to respond with a Sequence Reset with `GapFill=Y`, `NewSeqNo=201` and with `ApplLevelRecoveryIndicator=1` (new field). The `ApplLevelRecoveryIndicator=1` should be interpreted as application level state need to be recovered in some way, e.g. by requesting snapshots.

The second proposal is to respond with a Sequence Reset with `GapFill=N` and `NewSeqNo=201`, i.e. an ordinary reset. From FIXT 1.1, Errata March 2008, page 29:

**Sequence Reset – Reset should NOT be used as a normal response to a Resend Request (use Sequence Reset – Gap Fill mode). The Sequence Reset – Reset should ONLY be used to recover from a disaster situation which cannot be recovered via the use of Sequence Reset – Gap Fill. Note that the use of Sequence Reset – Reset may result in the possibility of lost messages.**

The formulation “ONLY be used to recover from a disaster situation” is quite strong, but “may result in the possibility of lost messages” fits the requirements.

The first proposal was chosen (`ApplLevelRecoveryIndicator=1`) to avoid changing the semantics of a Sequence Reset – Reset (“disaster situation”).

## 3.2 Recovery From Mismatching Sequence Numbers in Logon

If the session state could be reset by the Logon initiator with `ResetSeqNumFlag=Y`, then that could be used to recover from the mismatching sequence number situation without adding `NextExpectedMsgSeqNum` to the Logout message. Thus, instead of inspecting `MsgSeqNum` and `NextExpectedMsgSeqNum` in the Logout, the initiator could simply restart with Logon with `ResetSeqNumFlag=Y` and `MsgSeqNum=1`. The `ResetSeqNumFlag` is already present in the Logon message for usage in 24-hour connectivity, but cannot be used is otherwise according to the specification (it does not explicitly say so, but it seem to be the intention).

However, the Logon reset solution was not chosen since the acceptor would be forced to reset its' state upon request of the initiator and this could cause problems for some FIX engine implementations that have relied on that the `MsgSeqNum` cannot decrease.

## 3.3 Logon Reset

Currently several FIX engine implementations already support reset at Logon, and it is thus becoming a common practice. Logon reset could be an optional feature to support if some FIX engine implementations cannot easily support it.

# 4 Proposed Message Flow

## 4.1 Recovery From Mismatching Sequence Numbers in Logon

Two new error codes are added to the `SessionStatus` field to convey the following errors:

- **Received `MsgSeqNum` is too low.** The receiver expected a higher `MsgSeqNum`, i.e. the session state is corrupt.
- **Received `NextExpectedMsgSeqNum` is too high.** The receiver has never sent a message with `MsgSeqNum` equal to `NextExpectedMsgSeqNum-1` before, i.e. the session state is corrupt.

The `NextExpectedMsgSeqNum` field is added to the Logout message to convey the expected `MsgSeqNum` in case of the first error above. For the second case the initiator can observe the `MsgSeqNum` of the Logout message to find out what is the maximum `NextExpectedMsgSeqNum` (i.e. `MsgSeqNum+1`).

When the initiator receives a Logout with any of these errors it can automatically logon again with adjusted values in `MsgSeqNum` and/or `NextExpectedMsgSeqNum` and start the recovery procedure at the application level (e.g. request snapshots).

If the acceptor has lost its' session state this need not be a reason for Logout, instead this could be conveyed using `Sequence Reset`.

In all of the scenarios below the Logon is sent at the beginning of a FIX connection.

### **Acceptor detects `MsgSeqNum` is too low:**

Acceptor state: `next MsgSeqNum=500, NextExpectedMsgSeqNum=1000`

Initiator sends Logon:

`MsgSeqNum=100`  
`NextExpectedMsgSeqNum=200` (optional)

Acceptor reply, Logout (and then disconnect):

`MsgSeqNum=500`  
`SessionStatus=Initiators' MsgSeqNum is too low`  
`NextExpectedMsgSeqNum=1000`

The initiator can recover from this situation by connecting again and send a new Logon:

`MsgSeqNum=1000` (taken from Logout `NextExpectedMsgSeqNum`)  
`NextExpectedMsgSeqNum=200` (optional)



**Acceptor detects NextExpectedMsgSeqNum is too high:**

Acceptor state: next MsgSeqNum=500, NextExpectedMsgSeqNum=1000

Initiator sends Logon:

MsgSeqNum=1000  
NextExpectedMsgSeqNum=700

Acceptor reply, Logout (and then disconnect):

MsgSeqNum=500  
SessionStatus=Initiators' NextExpectedMsgSeqNum is too high  
NextExpectedMsgSeqNum=1000 (optional)

The initiator can recover from this situation by connecting again and send a new Logon:

MsgSeqNum=1000  
NextExpectedMsgSeqNum=**501** (taken from Logout MsgSeqNum+1)

**Acceptor detects both MsgSeqNum is too low and NextExpectedMsgSeqNum is too high:**

It is proposed that the error code for "MsgSeqNum is too low" takes precedence over "NextExpectedMsgSeqNum is too high".

Acceptor state: next MsgSeqNum=500, NextExpectedMsgSeqNum=1000

Initiator sends Logon:

MsgSeqNum=100  
NextExpectedMsgSeqNum=700

Acceptor reply, Logout (and then disconnect):

MsgSeqNum=500  
SessionStatus=Initiators' MsgSeqNum is too low  
NextExpectedMsgSeqNum=1000

The initiator can recover from this situation by connecting again and send a new Logon:

MsgSeqNum=**1000** (taken from Logout NextExpectedMsgSeqNum)  
NextExpectedMsgSeqNum=**501** (must be less than or equal to Logout MsgSeqNum+1)

**Acceptor has lost the session state (with Logon NextExpectedMsgSeqNum processing):**

Acceptor state: next MsgSeqNum=1 (unknown), NextExpectedMsgSeqNum=1 (unknown)

Initiator sends Logon:

MsgSeqNum=100  
NextExpectedMsgSeqNum=200

Acceptor reply, Logon:

MsgSeqNum=**200** (from NextExpectedMsgSeqNum)  
NextExpectedMsgSeqNum=**101** (from MsgSeqNum+1)

Acceptor sends Sequence Reset:

MsgSeqNum=201  
GapFillFlag=Y  
ApplLevelRecoveryIndicator=1  
NewSeqNo=202

Initiator realizes that application level recovery is needed (e.g. request snapshots)

**Acceptor has lost the session state (without Logon NextExpectedMsgSeqNum processing):**

Acceptor state: next MsgSeqNum=1 (unknown), NextExpectedMsgSeqNum=1 (unknown)

Initiator state: next MsgSeqNum=100, NextExpectedMsgSeqNum=200

Initiator sends Logon:

MsgSeqNum=100

Acceptor reply, Logon:

MsgSeqNum=1 (or guess arbitrary large number)

NextExpectedMsgSeqNum=**101** (from MsgSeqNum+1, optional)

Initiator reply, Logout (and then disconnect):

MsgSeqNum=101

SessionStatus=Received MsgSeqNum is too low

NextExpectedMsgSeqNum=200

Updated acceptor state: next MsgSeqNum=**200** (from Logout NextExpectedMsgSeqNum),  
 NextExpectedMsgSeqNum=**102** (from Logout MsgSeqNum+1). After next logon (which will most likely succeed)  
 the acceptor will send a Sequence Reset with GapFill=Y and ApplLevelRecoveryIndicator=1 to indicate that the  
 initiator should start application level recovery.

## 4.2 Logon Reset

Logon with MsgSeqNumResetFlag=Y processing is an optional feature that enables the initiator to request the session to be restarted (MsgSeqNum set to 1). The session can only be reset when a new FIX connection is initiated, i.e. not during an already established connection (which is covered by the 24-hour connectivity Logon procedure).

The Logon with MsgSeqNumRestFlag=Y works as follows:

Acceptor state: next MsgSeqNum=<any>, NextExpectedMsgSeqNum=<any>. .

Initiator (connects and) sends Logon:

MsgSeqNum=1

NextExpectedMsgSeqNum=1 (optional)

ResetSeqNumFlag=Y

Acceptor reply, Logon:

MsgSeqNum=1

NextExpectedMsgSeqNum=2 (optional)

ResetSeqNumFlag=Y

Both parties will continue with MsgSeqNum=2. It is recommended that application level recovery be initiated (e.g. request snapshots).

## 5 FIX message tables

### 5.1 Logout

Tag	FieldName	Req'd	Comments	Action	Mapping Usage and Comments
	StandardHeader	Y	MsgType = 5		
1409	SessionStatus	N			
789	NextExpectedMsgSeqNum	N	Can be used to provide the expected MsgSeqNum in case of an error.	ADD	
58	Text	N			
354	EncodedTextLen	N			
355	EncodedText	N			
	StandardTrailer	Y			

## 5.2 Sequence Reset

<i>Tag</i>	<i>FieldName</i>	<i>Req'd</i>	<i>Comments</i>	<i>Action</i>	<i>Mapping Usage and Comments</i>
StandardHeader		Y	MsgType = 4		
123	GapFillFlag	N			
36	NewSeqNo	Y			
1744	ApplLevelRecoveryIndicator	N	Indicates that some application level messages were gapfilled and application level recovery is needed.	ADD	
StandardTrailer		Y			

## 6 FIX component blocks

No changes.

## 7 Appendix A - Data Dictionary

Tag	Field Name	Action	Data type	Description	FIXML Abbreviation	Add to / Deprecate from Message type or Component block
1409	SessionStatus	New enum	int	9 = Received MsgSeqNum(34) is too low. 10 = Received NextExpectedMsgSeqNum(789) is too high.	@SessStat	
789	NextExpectedMsgSeqNum		SeqNum			Add to Logout
1744	ApplLevelRecoveryIndicator	New	int	Indicates whether application level recovery is needed. 0=Application level recovery is not needed (default) 1=Application level recovery is needed		Add to Sequence Reset

## 8 Appendix B - Glossary Entries

Term	Definition	Field where used

## 9 Appendix C - Usage Examples

## 10 Appendix D – Documentation Changes

The following are the proposed changes to the FIXT 1.1 documentation, Errata March 2008. New text is formatted with **yellow background** and discarded text is formatted as ~~red color and overstrike~~.

### Changes to section SESSION PROTOCOL, Logon at page 11:

- When using the ResetSeqNumFlag to maintain 24 hour connectivity [..]
- The ResetSeqNumFlag can also be used to reset sequence numbers when a FIX connection is initiated. The initiator should send a Logon with ResetSeqNumFlag=Y, MsgSeqNum=1 and optionally NextExpectedMsgSeqNum=1. The acceptor should respond with a Logon with ResetSeqNumFlag=Y, MsgSeqNum=1 and optionally NextExpectedMsgSeqNum=2. Both sides should proceed with MsgSeqNum=2. It is strongly recommended that snapshots be requested for data (orders, quotes, trades, market data etc.) that need to be in sync between the parties.

### Changes to section SESSION PROTOCOL, Message Recovery at page 13:

NOTE: In \*ALL\* cases except the Sequence Reset - Reset message **or Logon with ResetMsgSeqNumFlag=Y**, the FIX session should be terminated if the incoming sequence number is less than expected and the PossDupFlag is not set.