Date: September 12, 2016

Revision: 1

Status: Approved

FIX TECHNICAL ADDENDUM

FIX Technical Addendum Topic

Time Precision Enhancement

Effective Date:

September 12, 2016

FIX Standard Impact

The impact of this technical addendum includes the Data Type definitions of UTCTimestamp, UTCTimeOnly, TZTimestamp and TZTimeOnly in FIX versions 4.2, 4.4, 5.0, 5.0 SP1 and 5.0SPs as identified in the table below.

Table 1 - Standard Impact Table

Standard	Version	Affected Section and/or Subsection
FIX	4.2	Data Types: UTCTimestamp and UTCTimeOnly (page 13).
FIX	4.4	Data Types: UTCTimestamp and UTCTimeOnly (Volume 1 page 16).
FIX	5.0	Data Types: UTCTimestamp, UTCTimeOnly, TZTimestamp and TZTimeOnly (Volume 1 pages 15 and 16).
FIX	5.0 SP1	Data Types: UTCTimestamp, UTCTimeOnly, TZTimestamp and TZTimeOnly (Volume 1 pages 15, 16 and 17).
FIX	5.0SP2	Data Types: UTCTimestamp, UTCTimeOnly, TZTimestamp and TZTimeOnly (Volume 1 pages 16, 17 and 18).

Synopsis

Regulatory requirements, such as MiFID II from ESMA, necessitate a precision of timestamps greater than 3 decimal places of fractional seconds (milliseconds). As such, this technical addendum provides enhanced definitions for the Data Types UTCTimestamp, UTCTimeOnly, TZTimestamp and TZTimeOnly in support of the increased, sub-second time precision requirements. This change is understood to potentially impact FIX Engines adhering to the FIX version 4.2, 4.4, 5.0, 5.0 SP1 and 5.0 SP2 specifications. Extension Pack 206 (EP206) was approved and ratified January 21, 2016 and is part of the latest FIX application level specifications. This technical addendum applies these data type changes to prior versions of FIX in support of the new requirements. Bilateral

© Copyright, 2016, FIX Protocol, Limited

agreement in the FIX Rules of Engagement are necessary for time precision greater than 3 decimal places of fractional seconds (milliseconds).

Addendum Details:

The following table identifies the data type definitions for UTCTimestamp, UTCTimeOnly, TZTimestamp and TZTimeOnly data types as of EP206. These enhanced data type definitions are applied to and replace the corresponding data type definitions in FIX Version 4.2, 4.4, 5.0, 5.0 SP1 and 5.0 SP2 with this technical addendum.

Type Name	Description	Added in FIX version

© Copyright, 2016, FIX Protocol, Limited

Type Name	**Description**	**Added in FIX version**
UTCTimestamp	string field representing time/date combination represented in UTC (Universal Time Coordinated, also known as "GMT") in either YYYYMMDD-HH:MM:SS (whole seconds) or YYYYMMDD-HH:MM:SS.sss* format, colons, dash, and period required. Valid values: YYYY = 0000-9999, MM = 01-12, DD = 01-31, HH = 00-23, MM = 00-59, SS = 00-60 (60 only if UTC leap second), sss* fractions of seconds. The fractions of seconds may be empty when no fractions of seconds are conveyed (in such a case the period is not conveyed), it may include 3 digits to convey milliseconds, 6 digits to convey microseconds, 9 digits to convey nanoseconds, 12 digits to convey picoseconds; Other number of digits may be used with bilateral agreement. Leap Seconds: Note that UTC includes corrections for leap seconds, which are inserted to account for slowing of the rotation of the earth. Leap second insertion is declared by the International Earth Rotation Service (IERS) and has, since 1972, only occurred on the night of Dec. 31 or Jun 30. The IERS considers March 31 and September 30 as secondary dates for leap second insertion, but has never utilized these dates. During a leap second insertion, a UTCTimestamp field may read "19981231-23:59:59", "19981231-23:59:60", "19990101-00:00:00". (see http://tycho.usno.navy.mil/leapsec.html) Examples: TransactTime(60)="20011217-09:30:47.123" millisecond TransactTime(60)="20011217-09:30:47.123456" microseconds TransactTime(60)="20011217-09:30:47.123456789" nanoseconds TransactTime(60)="20011217-09:30:47.123456789123" picoseconds	FIX.4.2

© Copyright, 2016, FIX Protocol, Limited

Type Name	**Description**	**Added in FIX version**
UTCTimeOnly	string field representing time-only represented in UTC (Universal Time Coordinated, also known as "GMT") in either HH:MM:SS (whole seconds) or HH:MM:SS.sss* (milliseconds) format, colons, and period required. This special-purpose field is paired with UTCDateOnly to form a proper UTCTimestamp for bandwidth-sensitive messages. Valid values: HH = 00-23, MM = 00-59, SS = 00-60 (60 only if UTC leap second), sss* fractions of seconds. The fractions of seconds may be empty when no fractions of seconds are conveyed (in such a case the period is not conveyed), it may include 3 digits to convey milliseconds, 6 digits to convey microseconds, 9 digits to convey nanoseconds, 12 digits to convey picoseconds; Other number of digits may be used with bilateral agreement. Examples: MDEntryTime(273)="13:20:00.123"milliseconds MDEntryTime(273)="13:20:00.123456" microseconds MDEntryTime(273)="13:20:00.123456789" nanoseconds MDEntryTime(273)="13:20:00.123456789123" picoseconds	FIX.4.2
TZTimeOnly	string field representing the time represented based on ISO 8601. This is the time with a UTC offset to allow identification of local time and time zone of that time. Format is HH:MM[:SS][Z | [+ | - hh[:mm]]] where HH = 00-23 hours, MM = 00-59 minutes, SS = 00-59 seconds, hh = 01-12 offset hours, mm = 00-59 offset minutes. Examples: "07:39Z" is 07:39 UTC "02:39-05" is five hours behind UTC, thus Eastern Time "15:39+08" is eight hours ahead of UTC, Hong Kong/Singapore time "13:09+05:30" is 5.5 hours ahead of UTC, India time	FIX.4.4 EP21

Type Name	**Description**	**Added in FIX version**
TZTimestamp	string field representing a time/date combination representing local time with an offset to UTC to allow identification of local time and time zone offset of that time. The representation is based on ISO 8601. Format is YYYYMMDD-HH:MM:SS.sss*[Z | [+ | - hh[:mm]]] where YYYY = 0000 to 9999, MM = 01-12, DD = 01-31 HH = 00-23 hours, MM = 00-59 minutes, SS = 00-59 seconds, hh = 01-12 offset hours, mm = 00-59 offset minutes, sss* fractions of seconds. The fractions of seconds may be empty when no fractions of seconds are conveyed (in such a case the period is not conveyed), it may include 3 digits to convey milliseconds, 6 digits to convey microseconds, 9 digits to convey nanoseconds, 12 digits to convey picoseconds; Other number of digits may be used with bilateral agreement Examples: "20060901-07:39Z" is 07:39 UTC on 1st of September 2006 "20060901-02:39-05" is five hours behind UTC, thus Eastern Time on 1st of September 2006 "20060901-15:39+08" is eight hours ahead of UTC, Hong Kong/Singapore time on 1st of September 2006 "20060901-13:09+05:30" is 5.5 hours ahead of UTC, India time on 1st of September 2006 Using decimal seconds: "20060901-13:09.123+05:30" milliseconds "20060901-13:09.123456+05:30" microseconds "20060901-13:09.123456789+05:30" nanoseconds "20060901-13:09.123456789123+05:30" picoseconds "20060901-13:09.123456789Z" nanoseconds UTC time zone	FIX.4.4 EP21

All FIX services, via rules of engagements, identify the level of support provided for high resolution time (greater than 3 decimal places of seconds or milliseconds). FIX Service providers are encouraged to make the high-resolution precision levels configurable. The recommended levels for high resolution time precision support are:

0. No Support – FIX Service will likely experience a processing exception if high resolution time data is encountered.
1. High Resolution Tolerant – FIX Service can receive higher resolution time data, but will not transmit higher resolution time data and will not guarantee persistence of any higher resolution time data received.
2. Application Level Support for Higher resolution time data – FIX Service supports sending and receiving higher resolution time data and will persist higher resolution time data that is encountered within the application messages. The maximum precision shall be agreed upon out of band by counterparty agreement.

© Copyright, 2016, FIX Protocol, Limited

3. Full Support for Higher resolution time data – FIX Service supports sending and receiving higher resolution time data and will persist higher resolution time data at both the session layer and the application layer. The maximum precision shall be agreed upon out of band by counterparty agreement.