FIX Digital Asset WG

Extensions for Digital Asset Trading

May 18, 2022

v0.4

Proposal Status: Approved

For Global Technical Committee Governance Internal Use Only

Submission Date	May 19, 2022	Control Number	EP273
Submission Status	Submitted	Ratified Date	Aug 15, 2022
Primary Contact Person	Ryan Pierce	Release Identifier	FIX.Latest

r3.2

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.

Table of Contents

Table of Tables

Document History

Revision	Date	Author	Revision Comments
v0.1	Apr 26, 2022	Hanno Klein, GTC Shraddha Gohad, IIT	Initial draft
v0.2	May 11, 2022	Hanno Klein, GTC	Updates after WG review Apr 28 Updates of Chapters 5 and 6
V0.3	May 12, 2022 May 18, 2022 May 20, 2022	Hanno Klein, GTC	Updates after WG review May 12 Updated Data Dictionary based on the requirements. Added examples from Digital Asset WG.
V0.4	June 27, 2022 July 7, 2022	GTC Admin	Added Appendix E – Disposition of Public Comment Added final WG resolutions to the Public Comments based on Jul 7, 2022, WG call. Removed "wallet" concept throughout based on agreement on the WG call.
	July 19, 2022	GTC Admin	Assigned TBD and generated the ASBUILT document.
	August 18, 2022	GTC Admin	Minor changes based on comments from GTC Governance Board

1 Introduction

FIX provides support for a large number of asset classes. The FIX Digital Asset Working Group was created to provide standardization for digital assets including cryptocurrencies like Bitcoin and any kind of digital tokens that can be traded.

The FIX Digital Asset Working Group further wants to facilitate industry-wide implementation of standardized trading of digital assets by providing recommended practices. This work is going on in parallel to this gap analysis, which seeks to propose extensions to FIX to cover new business requirements arising from the nature of digital assets trading. The primary focus of this proposal is on the trading area with the objective to address the areas of pre- and post-trade with future proposals.

The proposal suggests FIX to support two different trading styles due to different existing market practices for digital asset trading.

- FX-style – currency pairs (e.g. BTC/USD, BTC/ETH) of fiat and/or digital currencies
- Securities-style – for trading any digital asset, priced and/or settling in either a fiat or digital currency

The identification of fiat currencies has been standardized a long time ago with ISO 4217, whereas the identification of digital assets (including digital currencies) has only recently been standardized with ISO 24165 *Digital Token Identifier*. One of the main proposals in this gap analysis is to extend the FIX datatype "Currency" to support both ISO 4217 and ISO 24165. This requires a number of additional extensions to support the ability to distinguish between the two code schemes when using FIX currency fields such as Currency(15) and SettlCurrency(120).

2 Business Requirements

2.1 Security identifiers for digital assets

Digital assets are traded either like a currency exchange or a security. FIX is able to support both approaches of how market participants trades digital assets.

In traditional currency exchange or Foreign Exchange trading the Symbol(55) field carries the currency pair using the "CCY1/CCY2" convention, e.g. EUR/USD. Many digital asset market participants who have adopted FIX have been using this FX-style convention, e.g. "BTC/USD"

An alternative convention adopted by some digital asset market participants is trading a digital asset like a security similar to trading of stocks. This style of trading simply provides the digital asset "symbol" in the Symbol(55) field, e.g. "BTC". This approach is popular when the digital asset is priced in terms of a fiat (government issued or sovereign) currency, such as US Dollars.

Note that how a digital asset trades have no bearing on the function or classification of a digital asset. Digital assets that are considered securities can trade using Foreign Exchange convention, and digital

assets that are considered virtual currencies can trade like stocks. FIX makes no recommendation regarding which of the two approaches should be used; this is left to bilateral agreement.

The business requirements are to support the identification and trading of digital assets under either one of these styles. Recently, existing symbologies (e.g. ISIN and FIGI) have or soon will support digital asset identification, and new symbologies (e.g. ISO 24165 Digital Token Identifier (DTI)) exclusively support digital assets. They provide more precise and unique identification of the digital asset and reduce the need to depend upon exchange symbology, which frequently is not consistent across exchanges.

The sections below go into details based on the trading style and proposed solution.

2.1.1 Security Identifier for securities-style trading

Securities-style trading is already supported by FIX with the exception of the ability to identify a value in SecurityID(48) as a DTI.

SecurityIDSource(22) conveys the identifier scheme of the value in SecurityID(48). It is proposed to add a new value for ISO 24165 (Digital Token Identifier – DTI). The values of SecurityIDSource(22) are identified in the Orchestra XML file with SecurityIDSourceCodeSet that is also used by SecurityAltIDSource(456) and many other fields. The extension of SecurityIDSourceCodeSet automatically extends all fields using this code set.

Y = Digital Token Identifier (DTI) – ISO 24165

2.1.2 Security identifier for FX-style trading

Currency symbols used for FX-style trading use ISO 4217 3-character codes which are created only under the framework of the standard. Digital assets did not have the benefit of an ISO standard at their inception, which has led to different symbols or labels to identify the same digital asset, e.g. "BTC" as well as "XBT" for bitcoin. ISO 24165 (Digital Token Identifier – DTI) was recently published to address the ambiguity by providing a unique identifier for cryptocurrencies as well as other digital assets where one or more random but human-readable labels can be attached. For example, the DTI assigned to bitcoin is 4H95J0R2X.

In order to disambiguate Symbol(55) for FX-style trading of digital assets, the requirement is to provide an ability to specify the DTI associated with the non-unique digital asset symbol(s) used in Symbol(55). To meet this requirement, it is proposed that the current concept for SecAltIDGrp be extended, which is currently limited to specifying alternate security identifiers to SecurityID(48) and SecurityIDSource(22), to allow for specifying alternative identifiers associated with the symbology in Symbol(55).

The proposed solution would support an explicit DTI or other security identifier that references either the first or the second symbol specified in the Symbol(55) field in FX-style trading via an ordinal reference within the SecAltIDGrp. This avoids the need for the definition of an implicit ordering of repeating group instances for SecAltIDGrp. It is proposed to add a new field to SecAltIDGrp for an ordinal reference as follows.

SymbolPositionNumber(2957) Reference to the first or second currency or digital asset in Symbol(55) for FX-style trading. Conditionally required when one or both symbols in Symbol(55) represent a digital asset.

The values are proposed as 1 = First currency or digital asset in Symbol(55) and 2 = Second currency or digital asset in Symbol(55).
SecurityAltID(455) and SecurityAltIDSource(456) are both required when using SecAltIDGrp. SecurityAltIDSource(456) has the same values as SecurityIDSource(22) and is used to denote the fiat currency with 6 = ISO Currency Code or another symbology that supports digital assets, such as Y=Digital Token Identifier (see section 2.1.1). The following table provides some examples. Note that Symbol(55) uses exchange ticker or other bilaterally agreed symbology, which is not consistent. In the example below, Bitcoin could be represented as either BTC or XBT in the Symbol(55) field, but the DTI for Bitcoin is the same in all cases.

Field	Buying 2.5 Bitcoin in USD	Buying $100,000 of Bitcoin (or selling $100,000 for Bitcoin)	Buying 10 Ether in Bitcoin
Symbol(55)	BTC/USD	BTC/USD	ETH/XBT
NoSecurityAltID(454)	2	2	2
> SecurityAltID(455)	4H95J0R2X (Bitcoin's DTI)	4H95J0R2X (Bitcoin's DTI)	X9J9K8725 (Ether's DTI)
> SecurityAltIDSource(456)	Y = DTI	Y = DTI	Y = DTI
> SymbolPositionNumber(2957)	1	1	1
> SecurityAltID(455)	USD	USD	4H95J0R2X (Bitcoin's DTI)
> SecurityAltIDSource(456)	6 = ISO Currency	6 = ISO Currency	Y = DTI
> SymbolPositionNumber(2957)	2	2	2

2.2 Clarifying Currency for digital asset trading

2.2.1 Currency field definition

The current description of the FIX field Currency(15) is as follows:

Identifies currency used for price. Absence of this field is interpreted as the default for the security. It is recommended that systems provide the currency value whenever possible. See Appendix 6-A: Valid Currency Codes for information on obtaining valid values.

This description has a securities trading focus due to the historical beginnings of FIX for equities trading. It is proposed to update this definition to clarify how this field applies to FX (fiat currencies and digital assets) and the securities type of trading digital assets. The FIX community of FX users have understood that the use of Currency(15) denotes the dealing currency of quantity fields such as OrderQty(38) (i.e. the value in the quantity field is denominated in that specified currency code) and Price(44) represents the conversion rate between the two currencies. The new definition should include text to explain the relationship with the new CurrencyCodeSource(2897) field (see 2.2.3 *Identifying Currency source scheme*) and add specific information about digital asset trading.

Orchestra allows a high-level definition (a.k.a synopsis or description) as well an elaboration to provide additional detail. It is proposed to use that capability and change the text as follows.

Synopsis/Description – Identifies currency used for price or quantity fields, depending on the asset class being traded. CurrencyCodeSource(2897 may be used to disambiguate the code source scheme used, and ISO 4217 is the default scheme if absent.

Elaboration – For securities trading and digital assets traded securities-style, identifies the currency used to denote the price. Absence of this field is interpreted as the default for the security. For Foreign Exchange (FX) and digital assets traded FX-style, identifies the dealt currency used to denominate (the) quantity related field(s).

2.2.2 Currency datatype definition

The FIX datatype "Currency" is used for a large number of fields and historically used to convey fiat currencies (ISO 4217). It currently is defined as follows:

String field representing a currency type using ISO 4217 Currency code (3 character).

This historical definition precludes the use of the FIX currency fields for digital asset trading. It is proposed to extend the definition of the datatype as follows.

String field representing a currency type using ISO 4217 CurrencyCode (3 character) or a digital asset. The digital asset may be denoted in one of several schemes, including but not limited to ISO 24165 Digital Token Identifier. The default is ISO 4217 in the absence of a second field identifying the source scheme.

2.2.3 Identifying Currency source scheme

The extension of the scope of the FIX datatype "Currency" (see section 2.2) introduces a possible ambiguity related to the domain of values in the FIX currency fields. Whilst the absence of any further information in a message should indicate the use of ISO 4217 as a default, an optional explicit approach is needed for digital asset trading.

It is proposed to add new fields for those FIX currency related fields that are relevant for the entire life-cycle of digital assets to be able to explicitly define the source scheme used. See section 3 for further elaborations on scope of this gap analysis.

The following table proposes new fields for currency source schemes and the messages they are to be added to. It shows the existing currency related field that is already part of the listed messages followed by the proposed new field.

Table 1 Fields to identify currency source schemes in messages

Field	Add to message(s)
Currency(15) CurrencyCodeSource(2897)	• Advertisement(35=7) • IOI(35=6) • BidRequest(35=k) • Quote(35=S) • QuoteResponse(35=AJ) • QuoteStatusReport(35=AI) • SecurityDefinitionRequest(35=c) • SecurityDefinition(35=d) • SecurityDefinitionUpdateReport(35=BP) • SecurityListRequest(35=x) • DerivativeSecurityListRequest(35=z) • SecurityStatusRequest(35=e) • SecurityStatus(35=f) • MarketDefinition(35=BU) • MarketDefinitionUpdateReport(35=BV) • MarketDataStatisticsReport(35=DP) • PartyRiskLimitCheckRequest(35=DF) • PartyRiskLimitCheckRequestAck(35=DG) • NewOrderSingle(35=D) • NewOrderMultileg(35=AB) • OrderCancelReplaceRequest(35=G) • MultilegOrderCancelReplace(35=AC) • ExecutionReport(35=8) • NewOrderCross(35=s) • CrossOrderCancelReplaceRequest(35=t) • TradeCaptureReport(35=AE) • TradeCaptureReportAck(35=AR) • TradeAggregationRequest(35=DW) • RequestForPositions(35=AN) • RequestForPositionsAck(35=AO) • PositionMaintenanceRequest(35=AL) • PositionMaintenanceReport(35=AM) • PositionReport(35=AP) • PositionTransferInstruction(35=DL) • PositionTransferReport(35=DN) • AllocationInstruction(35=J) • AllocationInstructionAlert(35=BM) • AllocationReport(35=AS) • Confirmation(35=AK) • AssignmentReport(35=AW) • CollateralRequest(35=AX) • CollateralAssignment(35=AY) • CollateralResponse(35=AZ) • CollateralReport(35=BA) • CollateralInquiry(35=BB) • CollateralInquiryAck(35=BG) • MarginRequirementReport(35=CJ)

Field	Add to message(s)
SettlCurrency(120) SettlCurrencyCodeSource(2899)	• Quote(35=S) • NewOrderSingle(35=D) • NewOrderMultileg(35=AB) • OrderCancelReplaceRequest(35=G) • MultilegOrderCancelReplace(35=AC) • ExecutionReport(35=8) • TradeCaptureReport(35=AE) • TradeCaptureReportAck(35=AR) • RequestForPositions(35=AN) • RequestForPositionsAck(35=AO) • PositionMaintenanceRequest(35=AL) • PositionMaintenanceReport(35=AM) • PositionReport(35=AP) • Confirmation(35=AK) • SettlementInstructionRequest(35=AV)
SettlPriceUnitOfMeasureCurrency(1887) SettlPriceUnitOfMeasureCurrencyCodeSource(2960)	• PositionReport(35=AP)

The following table proposes new fields for currency source schemes and the components they are to be added to. It shows the existing currency related field that is already part of the listed components followed by the proposed new field.

Table 2 Fields to identify currency source schemes in components

Field	Add to component(s)
Currency(15) CurrencyCodeSource(2897)	• InstrmtMDReqGrp • InstrmtMatchSideGrp • InstrmtStrkPxGrp • ListOrdGrp • MDFullGrp • MDIncGrp • QuotEntryAckGrp • QuotEntryGrp • QuotReqGrp • QuotReqRjctGrp • RelSymDerivSecGrp • RelSymDerivSecUpdGrp • SecListGrp • SecLstUpdRelSymGrp • SettlObligationInstructions
LegCurrency(556) LegCurrencyCodeSource(2898)	• InstrumentLeg
SettlCurrency(120) SettlCurrencyCodeSource(2899)	• AllocGrp • InstrmtMatchSideGrp • ListOrdGrp • MDFullGrp • MDIncGrp • QuotReqGrp • SettlInstGrp

	• SettlObligationInstructions • SideCrossOrdModGrp
LegSettlCurrency(675) LegSettlCurrencyCodeSource(2900)	• InstrmtLegExecGrp • LegOrdGrp • SideCrossLegGrp • TrdInstrmtLegExecGrp • TrdInstrmtLegGrp
SideCurrency(1154) SideCurrencyCodeSource(2901)	• TrdCapRptSideGrp • TrdCapRptAckSideGrp • TrdMatchSideGrp
SideSettlCurrency(1155) SideSettlCurrencyCodeSource(2902)	• TrdCapRptSideGrp • TrdCapRptAckSideGrp • TrdMatchSideGrp
SettlementAmountCurrency(1702) SettlementAmountCurrencyCodeSource(2903)	• SettlementAmountGrp
StrikeCurrency(947) StrikeCurrencyCodeSource(2904)	• Instrument
UnitOfMeasureCurrency(1716) UnitOfMeasureCurrencyCodeSource(2905)	• Instrument
PriceUnitOfMeasureCurrency(1717) PriceUnitOfMeasureCurrencyCodeSource(2906)	• Instrument
PriceQuoteCurrency(1524) PriceQuoteCurrencyCodeSource(2907)	• Instrument
LegStrikeCurrency(942) LegStrikeCurrencyCodeSource(2908)	• InstrumentLeg
LegUnitOfMeasureCurrency(1720) LegUnitOfMeasureCurrencyCodeSource(2909)	• InstrumentLeg
LegPriceUnitOfMeasureCurrency(1721) LegPriceUnitOfMeasureCurrencyCodeSource(2910)	• InstrumentLeg
LegPriceQuoteCurrency(1528) LegPriceQuoteCurrencyCodeSource(2911)	• InstrumentLeg
DerivativeStrikeCurrency(1262) DerivativeStrikeCurrencyCodeSource(2912)	• DerivativeInstrument
DerivativeUnitOfMeasureCurrency(1722) DerivativeUnitOfMeasureCurrencyCodeSource(2913)	• DerivativeInstrument
DerivativePriceUnitOfMeasureCurrency(1723) DerivativePriceUnitOfMeasureCurrencyCodeSource(2914)	• DerivativeInstrument
DerivativePriceQuoteCurrency(1576) DerivativePriceQuoteCurrencyCodeSource(2915)	• DerivativeInstrument
UnderlyingCurrency(318) UnderlyingCurrencyCodeSource(2916)	• UnderlyingInstrument
UnderlyingStrikeCurrency(941) UnderlyingStrikeCurrencyCodeSource(2917)	• UnderlyingInstrument
UnderlyingUnitOfMeasureCurrency(1718) UnderlyingUnitOfMeasureCurrencyCodeSource(2918)	• UnderlyingInstrument
UnderlyingPriceUnitOfMeasureCurrency(1719) UnderlyingPriceUnitOfMeasureCurrencyCodeSource(2919)	• UnderlyingInstrument
UnderlyingPriceQuoteCurrency(1526) UnderlyingPriceQuoteCurrencyCodeSource(2920)	• UnderlyingInstrument
UnderlyingNotionalCurrency(2615) UnderlyingNotionalCurrencyCodeSource(2921)	• UnderlyingInstrument

CommCurrency(479) CommCurrencyCodeSource(2922)	• CommissionData
CommissionCurrency(2643) CommissionCurrencyCodeSource(2923)	• CommissionDataGrp
CommissionUnitOfMeasureCurrency(2645) CommissionUnitOfMeasureCurrencyCodeSource(2924)	• CommissionDataGrp
AllocCommissionCurrency(2657) AllocCommissionCurrencyCodeSource(2925)	• AllocCommissionDataGrp
AllocCommissionUnitOfMeasureCurrency(2659) AllocCommissionUnitOfMeasureCurrencyCodeSource(2926)	• AllocCommissionDataGrp
AllocSettlCurrency(736) AllocSettlCurrencyCodeSource(2927)	• PreAllocGrp • PreAllocMlegGrp • AllocGrp • TrdAllocGrp
LegAllocSettlCurrency(1367) LegAllocSettlCurrencyCodeSource(2928)	• LegPreAllocGrp
CollateralCurrency(1705) CollateralCurrencyCodeSource(2929)	• CollateralAmountGrp
SideCollateralCurrency(2695) SideCollateralCurrencyCodeSource(2930)	• SideCollateralAmountGrp
CollateralReinvestmentCurrency(2843) CollateralReinvestmentCurrencyCodeSource(2931)	• CollateralReinvestmentGrp
SideCollateralReinvestmentCurrency(2866) SideCollateralReinvestmentCurrencyCodeSource(2932)	• SideCollateralReinvestmentGrp
TradeAllocCurrency(1847) TradeAllocCurrencyCodeSource(2933)	• TradeAllocAmtGrp
TradingCurrency(1245) TradingCurrencyCodeSource(2934)	• BaseTradingRules
LimitAmtCurrency(1634) LimitAmtCurrencyCodeSource(2935)	• LimitAmts
PosQtyUnitOfMeasureCurrency(1835) PosQtyUnitOfMeasureCurrencyCodeSource(2936)	• PositionQty
PositionCurrency(1055) PositionCurrencyCodeSource(2937)	• PositionAmountData
LegPosCurrency(1589) LegPosCurrencyCodeSource(2938)	• LegPositionAmountData
RiskLimitCurrency(1532) RiskLimitCurrencyCodeSource(2939)	• RiskLimitTypesGrp
EntitlementAttribCurrency(1781) EntitlementAttribCurrencyCodeSource(2940)	• EntitlementAttribGrp
ComplexOptPayoutCurrency(2122) ComplexOptPayoutCurrencyCodeSource(2941)	• ComplexEvents
ComplexEventCurrencyOne(2124) ComplexEventCurrencyOneCodeSource(2942)	• ComplexEvents
ComplexEventCurrencyTwo(2125) ComplexEventCurrencyTwoCodeSource(2943)	• ComplexEvents
LegComplexOptPayoutCurrency(2226) LegComplexOptPayoutCurrencyCodeSource(2944)	• LegComplexEvents
LegComplexEventCurrencyOne(2233) LegComplexEventCurrencyOneCodeSource(2945)	• LegComplexEvents
LegComplexEventCurrencyTwo(2234)	• LegComplexEvents

LegComplexEventCurrencyTwoCodeSource(2946)	
UnderlyingComplexOptPayoutCurrency(2266) UnderlyingComplexOptPayoutCurrencyCodeSource(2947)	• UnderlyingComplexEvents
UnderlyingComplexEventCurrencyOne(2268) UnderlyingComplexEventCurrencyOneCodeSource(2948)	• UnderlyingComplexEvents
UnderlyingComplexEventCurrencyTwo(2269) UnderlyingComplexEventCurrencyTwoCodeSource(2949)	• UnderlyingComplexEvents
BenchmarkCurveCurrency(220) BenchmarkCurveCurrencyCodeSource(2950)	• SpreadOrBenchmarkCurveData
LegBenchmarkCurveCurrency(676) LegBenchmarkCurveCurrencyCodeSource(2951)	• LegBenchmarkCurveData
AgreementCurrency(918) AgreementCurrencyCodeSource(2952)	• FinancingDetails
LegAgreementCurrency(2495) LegAgreementCurrencyCodeSource(2953)	• LegFinancingDetails
FundingSourceCurrency(2847) FundingSourceCurrencyCodeSource(2954)	• FundingSourceGrp
PayCollectCurrency(1709) PayCollectCurrencyCodeSource(2955)	• PayCollectGrp
PostTradePaymentCurrency(2818) PostTradePaymentCurrencyCodeSource(2956)	• PostTradePayment

All currency fields for OTC trading are out of scope of this document but may be subject to a future gap analysis.

All of these new fields should use a subset of the values available for SecurityIDSource(22) (defined by SecurityIDSourceCodeSet), depending on their applicability to digital assets. The code set values themselves are proposed to be identical to those used in SecurityIDSource(22), e.g. SecurityIDSource(22) and CurrencyCodeSource(2897) both use 6 = ISO Currency Code for fiat currencies. The following table proposes the list applicable values for currency source schemes.

Value	Name
1	CUSIP
2	SEDOL
4	ISIN
6	ISO Currency Code
S	Financial Instrument Global Identifier (FIGI)
Y	Digital Token Identifier (DTI)

2.3 Extending security types for digital assets

FIX provides fields such as Product(460), CFICode(461), SecurityType(167), SecuritySubType(762) in the Instrument component to identify products and instruments, i.e. describing "what" is being traded. It is proposed to add a new security type for digital assets trading by adding the following value to SecurityType(167) (and related fields using SecurityTypeCodeSet).

DIGITAL = Digital Asset
Elaboration: Asset that exists only in digital form or which is the digital representation of another asset (Source: ISO 24165 - Terms and Definitions).

3 Issues and Discussion Points

3.1 Pre- and post-trade

This gap analysis is focused on digital asset trading. Pre-trade and post-trade workflows for digital assets have not been discussed or considered as part of this gap analysis.

The Working Group decided to simply provide the new fields to express currency code schemes for all relevant fields including fields used in pre-trade and post-trade workflows. Discussions specific to pre-trade and post-trade workflows will be conducted in the future. Fields used for OTC (e.g. OTC swaps) are also excluded in this gap analysis.

3.2 Multileg instruments

The gap analysis has a focus on single leg instruments and does not include messages or components for the trading of multileg instruments. The issue is whether any of these should be included in this initial gap analysis or deferred to a future gap analysis related to digital asset trading.

The working group decided to include leg level currency fields into the scope of new fields to express currency code schemes. There is no reason to believe that digital assets will not be traded in conjunction with strategies.

4 Proposed Message Flow

NONE

5 FIX Message Tables

This gap analysis impacts a large number of messages as it adds fields to identify currency source schemes to their root level. However, the new fields are always positioned right after the existing currency field. Both the existing and the new fields and the corresponding messages are listed in *Table 1 Fields to identify currency source schemes in messages*.

6 FIX Component Blocks

This gap analysis impacts a large number of components as it adds fields to identify currency source schemes. However, the new fields are always positioned right after the existing currency field. Both the existing and the new fields and the corresponding components are listed in *Table 2 Fields to identify currency source schemes in components*. The following table shows the components and the existing and new field(s) for each of them.

Component	Currency field(s)
Instrument	StrikeCurrency(947) StrikeCurrencyCodeSource(2904) UnitOfMeasureCurrency(1716) UnitOfMeasureCurrencyCodeSource(2905) PriceUnitOfMeasureCurrency(1717) PriceUnitOfMeasureCurrencyCodeSource(2906) PriceQuoteCurrency(1524) PriceQuoteCurrencyCodeSource(2907)
InstrumentLeg	LegCurrency(556) LegCurrencyCodeSource(2898) LegStrikeCurrency(942) LegStrikeCurrencyCodeSource(2908) LegUnitOfMeasureCurrency(1720) LegUnitOfMeasureCurrencyCodeSource(2909) LegPriceUnitOfMeasureCurrency(1721) LegPriceUnitOfMeasureCurrencyCodeSource(2910) LegPriceQuoteCurrency(1528) LegPriceQuoteCurrencyCodeSource(2911)
UnderlyingInstrument	UnderlyingCurrency(318) UnderlyingCurrencyCodeSource(2916) UnderlyingStrikeCurrency(941) UnderlyingStrikeCurrencyCodeSource(2917) UnderlyingUnitOfMeasureCurrency(1718) UnderlyingUnitOfMeasureCurrencyCodeSource(2918) UnderlyingPriceUnitOfMeasureCurrency(1719)

Component	Currency field(s)
	UnderlyingPriceUnitOfMeasureCurrencyCodeSource(2919) UnderlyingPriceQuoteCurrency(1526) UnderlyingPriceQuoteCurrencyCodeSource(2920) UnderlyingNotionalCurrency(2615) UnderlyingNotionalCurrencyCodeSource(2921)
DerivativeInstrument	DerivativeStrikeCurrency(1262) DerivativeStrikeCurrencyCodeSource(2912) DerivativeUnitOfMeasureCurrency(1722) DerivativeUnitOfMeasureCurrencyCodeSource(2913) DerivativePriceUnitOfMeasureCurrency(1723) DerivativePriceUnitOfMeasureCurrencyCodeSource(2914) DerivativePriceQuoteCurrency(1576) DerivativePriceQuoteCurrencyCodeSource(2915)
CommissionDataGrp	CommissionCurrency(2643) CommissionCurrencyCodeSource(2923) CommissionUnitOfMeasureCurrency(2645) CommissionUnitOfMeasureCurrencyCodeSource(2924)
CommissionData	CommCurrency(479) CommCurrencyCodeSource(2922)
AllocCommissionDataGrp	AllocCommissionCurrency(2657) AllocCommissionCurrencyCodeSource(2925) AllocCommissionUnitOfMeasureCurrency(2659) AllocCommissionUnitOfMeasureCurrencyCodeSource(2926)
PreAllocGrp	AllocSettlCurrency(736) AllocSettlCurrencyCodeSource(2927)
PreAllocMlegGrp	AllocSettlCurrency(736) AllocSettlCurrencyCodeSource(2927)
AllocGrp	AllocSettlCurrency(736) AllocSettlCurrencyCodeSource(2927) SettlCurrency(120) SettlCurrencyCodeSource(2899)
TrdAllocGrp	AllocSettlCurrency(736) AllocSettlCurrencyCodeSource(2927)
LegPreAllocGrp	LegAllocSettlCurrency(1367) LegAllocSettlCurrencyCodeSource(2928)
CollateralAmountGrp	CollateralCurrency(1705) CollateralCurrencyCodeSource(2929)
SideCollateralAmountGrp	SideCollateralCurrency(2695) SideCollateralCurrencyCodeSource(2930)
CollateralReinvestmentGrp	CollateralReinvestmentCurrency(2843) CollateralReinvestmentCurrencyCodeSource(2931)
SideCollateralReinvestmentGrp	SideCollateralReinvestmentCurrency(2866) SideCollateralReinvestmentCurrencyCodeSource(2932)

Component	Currency field(s)
TradeAllocAmtGrp	TradeAllocCurrency(1847) TradeAllocCurrencyCodeSource(2933)
BaseTradingRules	TradingCurrency(1245) TradingCurrencyCodeSource(2934)
LimitAmts	LimitAmtCurrency(1634) LimitAmtCurrencyCodeSource(2935)
PositionQty	PosQtyUnitOfMeasureCurrency(1835) PosQtyUnitOfMeasureCurrencyCodeSource(2936)
PositionAmountData	PositionCurrency(1055) PositionCurrencyCodeSource(2937)
LegPositionAmountData	LegPosCurrency(1589) LegPosCurrencyCodeSource(2938)
RiskLimitTypesGrp	RiskLimitCurrency(1532) RiskLimitCurrencyCodeSource(2939)
EntitlementAttribGrp	EntitlementAttribCurrency(1781) EntitlementAttribCurrencyCodeSource(2940)
ComplexEvents	ComplexOptPayoutCurrency(2122) ComplexOptPayoutCurrencyCodeSource(2941) ComplexEventCurrencyOne(2124) ComplexEventCurrencyOneCodeSource(2942) ComplexEventCurrencyTwo(2125) ComplexEventCurrencyTwoCodeSource(2943)
LegComplexEvents	LegComplexOptPayoutCurrency(2226) LegComplexOptPayoutCurrencyCodeSource(2944) LegComplexEventCurrencyOne(2233) LegComplexEventCurrencyOneCodeSource(2945) LegComplexEventCurrencyTwo(2234) LegComplexEventCurrencyTwoCodeSource(2946)
UnderlyingComplexEvents	UnderlyingComplexOptPayoutCurrency(2266) UnderlyingComplexOptPayoutCurrencyCodeSource(2947) UnderlyingComplexEventCurrencyOne(2268) UnderlyingComplexEventCurrencyOneCodeSource(2948) UnderlyingComplexEventCurrencyTwo(2269) UnderlyingComplexEventCurrencyTwoCodeSource(2949)
SpreadOrBenchmarkCurveData	BenchmarkCurveCurrency(220) BenchmarkCurveCurrencyCodeSource(2950)
LegBenchmarkCurveData	LegBenchmarkCurveCurrency(676) LegBenchmarkCurveCurrencyCodeSource(2951)
FinancingDetails	AgreementCurrency(918) AgreementCurrencyCodeSource(2952)
LegFinancingDetails	LegAgreementCurrency(2495) LegAgreementCurrencyCodeSource(2953)
FundingSourceGrp	FundingSourceCurrency(2847) FundingSourceCurrencyCodeSource(2954)
PayCollectGrp	PayCollectCurrency(1709)

Component	Currency field(s)
	PayCollectCurrencyCodeSource(2955)
PostTradePayment	PostTradePaymentCurrency(2818) PostTradePaymentCurrencyCodeSource(2956)
TrdCapRptSideGrp	SideSettlCurrency(1155) SideSettlCurrencyCodeSource(2902) SideCurrency(1154) SideCurrencyCodeSource(2901)
TrdCapRptAckSideGrp	SideSettlCurrency(1155) SideSettlCurrencyCodeSource(2902) SideCurrency(1154) SideCurrencyCodeSource(2901)
TrdMatchSideGrp	SideSettlCurrency(1155) SideSettlCurrencyCodeSource(2902) SideCurrency(1154) SideCurrencyCodeSource(2901)
InstrmtLegExecGrp	LegSettlCurrency(675) LegSettlCurrencyCodeSource(2900)
LegOrdGrp	LegSettlCurrency(675) LegSettlCurrencyCodeSource(2900)
SideCrossLegGrp	LegSettlCurrency(675) LegSettlCurrencyCodeSource(2900)
TrdInstrmtLegExecGrp	LegSettlCurrency(675) LegSettlCurrencyCodeSource(2900)
TrdInstrmtLegGrp	LegSettlCurrency(675) LegSettlCurrencyCodeSource(2900)
InstrmtMatchSideGrp	Currency(15) CurrencyCodeSource(2897) SettlCurrency(120) SettlCurrencyCodeSource(2899)
ListOrdGrp	Currency(15) CurrencyCodeSource(2897) SettlCurrency(120) SettlCurrencyCodeSource(2899)
MDFullGrp	Currency(15) CurrencyCodeSource(2897) SettlCurrency(120) SettlCurrencyCodeSource(2899)
MDIncGrp	Currency(15) CurrencyCodeSource(2897) SettlCurrency(120) SettlCurrencyCodeSource(2899)
QuotReqGrp	Currency(15) CurrencyCodeSource(2897)

Component	Currency field(s)
	SettlCurrency(120) SettlCurrencyCodeSource(2899)
SettlInstGrp	SettlCurrency(120) SettlCurrencyCodeSource(2899)
SettlObligationInstructions	Currency(15) CurrencyCodeSource(2897) SettlCurrency(120) SettlCurrencyCodeSource(2899)
SideCrossOrdModGrp	SettlCurrency(120) SettlCurrencyCodeSource(2899
InstrmtMDReqGrp	Currency(15) CurrencyCodeSource(2897)
InstrmtStrkPxGrp	Currency(15) CurrencyCodeSource(2897)
QuotEntryAckGrp	Currency(15) CurrencyCodeSource(2897)
QuotEntryGrp	Currency(15) CurrencyCodeSource(2897)
QuotReqRjctGrp	Currency(15) CurrencyCodeSource(2897)
RelSymDerivSecGrp	Currency(15) CurrencyCodeSource(2897)
RelSymDerivSecUpdGrp	Currency(15) CurrencyCodeSource(2897)
SecListGrp	Currency(15) CurrencyCodeSource(2897)
SecLstUpdRelSymGrp	Currency(15) CurrencyCodeSource(2897)

6.1 Component SecAltIDGrp

To be completed at the time of the proposal – all information provided will be included in the repository	
Component Name	SecAltIDGrp
Component Abbreviated Name (for FIXML)	AID
Component Type	_X_ Block Repeating ___ Block
Category	Common
Action	__New _X_Change
Component Synopsis	*(no change)*
Component Elaboration	*(no change)*

To be finalized by FPL Technical Office	
Repository Component ID 2071	

Component FIXML Abbreviation: <*SecAltID*>						
Tag	*Field Name*		*Req'd*	*Action*	*Mappings and Usage Comments*	*FIX Spec Comments*
454	NoSecurityAltID					
→	455	SecurityAltID				
→	456	SecurityAltIDSource				
→	2957	SymbolPositionNumber	N	NEW		
</*SecAltID*>						

6.2 Component LegSecAltIDGrp

To be completed at the time of the proposal – all information provided will be included in the repository	
Component Name	LegSecAltIDGrp
Component Abbreviated Name (for FIXML)	LegAID
Component Type	_X_ Block Repeating ___ Block
Category	Common
Action	__New _X_Change
Component Synopsis	*(no change)*
Component Elaboration	*(no change)*
To be finalized by FPL Technical Office	
Repository Component ID 2072	

Component FIXML Abbreviation: <*LegSecAltIDGrp*>						
Tag	*Field Name*		*Req'd*	*Action*	*Mappings and Usage Comments*	*FIX Spec Comments*
604	NoLegSecurityAltID					
→	605	LegSecurityAltID				
→	606	LegSecurityAltIDSource				

→	2958	LegSymbolPositionNumber	N	NEW		
</LegSecAltIDGrp>						

6.3 Component UndSecAltIDGrp

To be completed at the time of the proposal – all information provided will be included in the repository	
Component Name	UndSecAltIDGrp
Component Abbreviated Name (for FIXML)	UndAID
Component Type	_X_ Block Repeating ___ Block
Category	Common
Action	__New _X_Change
Component Synopsis	*(no change)*
Component Elaboration	*(no change)*
To be finalized by FPL Technical Office	
Repository Component ID 2073	

Component FIXML Abbreviation: <*UndSecAltIDGrp*>						
Tag	*Field Name*		*Req'd*	*Action*	*Mappings and Usage Comments*	*FIX Spec Comments*
457	NoUnderlyingSecurityAltID					
→	458	UnderlyingSecurityAltID				
→	459	UnderlyingSecurityAltIDSource				
→	2959	UnderlyingSymbolPositionNumber	N	NEW		
</*UndSecAltIDGrp*>						

7 Category Changes

NONE

Appendix A – Data Dictionary

Tag	FieldName	Action	Datatype	Description	FIXML Abbreviation	Add to / Deprecate from Message type or Component block
2897	CurrencyCodeSource	NEW	string	Identifies class or source of the Currency(15) value. 1 = CUSIP 2 = SEDOL 4 = ISIN 6 = ISO Currency Code (ISO 4217) S = Financial Instrument Global Identifier Elaboratrion- An Object Management Group (OMG) standard. Also referred to as FIGI. Formerly known as "Bloomberg Open Symbology BBGID". Y = Digital Token Identifier (ISO 24165)	@CcySrc	Please see *Table 1* and *Table 2*
2898	LegCurrencyCodeSource	NEW	string	Identifies class or source of the LegCurrency(556) value. *(Uses enums from CurrencyCodeSource(2897))*	@CcySrc	Please see *Table 1* and *Table 2*
2899	SettlCurrencyCodeSource	NEW	string	Identifies class or source of the SettlCurrency(120) value. *(Uses enums from CurrencyCodeSource(2897))*	@SettlCcySrc	Please see *Table 1* and *Table 2*

2900	LegSettlCurrencyCodeSource	NEW	string	Identifies class or source of the LegSettlCurrency(675) value. *(Uses enums from CurrencyCodeSource(2897))*	@SettlCcySrc	Please see *Table 1* and *Table 2*
2901	SideCurrencyCodeSource	NEW	string	Identifies class or source of the SideCurrency(1154) value. *(Uses enums from CurrencyCodeSource(2897))*	@CcySrc	Please see *Table 1* and *Table 2*
2902	SideSettlCurrencyCodeSource	NEW	string	Identifies class or source of the SideSettlCurrency(1155) value. *(Uses enums from CurrencyCodeSource(2897))*	@SettlCcySrc	Please see *Table 1* and *Table 2*
2903	SettlementAmountCurrencyCodeSource	NEW	string	Identifies class or source of the SettlementAmountCurrency(1702) value. *(Uses enums from CurrencyCodeSource(2897))*	@CcySrc	Please see *Table 1* and *Table 2*
2904	StrikeCurrencyCodeSource	NEW	string	Identifies class or source of the StrikeCurrency(947) value. *(Uses enums from CurrencyCodeSource(2897))*	@StrkCcySrc	Please see *Table 1* and *Table 2*
2905	UnitOfMeasureCurrencyCodeSource	NEW	string	Identifies class or source of the UnitOfMeasureCurrency(1716) value. *(Uses enums from CurrencyCodeSource(2897))*	@UOMCcySrc	Please see *Table 1* and *Table 2*

2906	PriceUnitOfMeasureCurrencyCodeSource	NEW	string	Identifies class or source of the PriceUnitOfMeasureCurrency(1717) value. *(Uses enums from CurrencyCodeSource(2897))*	@PxUOMCcySrc	Please see *Table 1* and *Table 2*
2907	PriceQuoteCurrencyCodeSource	NEW	string	Identifies class or source of the PriceQuoteCurrency(1524) value. *(Uses enums from CurrencyCodeSource(2897))*	@PxQteCcySrc	Please see *Table 1* and *Table 2*
2908	LegStrikeCurrencyCodeSource	NEW	string	Identifies class or source of the LegStrikeCurrency(942) value. *(Uses enums from CurrencyCodeSource(2897))*	@StrkCcySrc	Please see *Table 1* and *Table 2*
2909	LegUnitOfMeasureCurrencyCodeSource	NEW	string	Identifies class or source of the LegUnitOfMeasureCurrency(1720) value. *(Uses enums from CurrencyCodeSource(2897))*	@UOMCcySrc	Please see *Table 1* and *Table 2*
2910	LegPriceUnitOfMeasureCurrencyCodeSource	NEW	string	Identifies class or source of the LegPriceUnitOfMeasureCurrency(1721) value. *(Uses enums from CurrencyCodeSource(2897))*	@PxUOMCcySrc	Please see *Table 1* and *Table 2*
2911	LegPriceQuoteCurrencyCodeSource	NEW	string	Identifies class or source of the LegPriceQuoteCurrency(1528) value. *(Uses enums from CurrencyCodeSource(2897))*	@PxQteCcySrc	Please see *Table 1* and *Table 2*

2912	DerivativeStrikeCurrencyCodeSource	NEW	string	Identifies class or source of the DerivativeStrikeCurrency(1262) value. *(Uses enums from CurrencyCodeSource(2897))*	@StrkCcySrc	Please see *Table 1* and *Table 2*
2913	DerivativeUnitOfMeasureCurrencyCodeSource	NEW	string	Identifies class or source of the DerivativeUnitOfMeasureCurrency(1722) value. *(Uses enums from CurrencyCodeSource(2897))*	@UOMCcySrc	Please see *Table 1* and *Table 2*
2914	DerivativePriceUnitOfMeasureCurrencyCodeSource	NEW	string	Identifies class or source of the DerivativePriceUnitOfMeasureCurrency(1723) value. *(Uses enums from CurrencyCodeSource(2897))*	@PxUOMCcySrc	Please see *Table 1* and *Table 2*
2915	DerivativePriceQuoteCurrencyCodeSource	NEW	string	Identifies class or source of the DerivativePriceQuoteCurrency(1576) value. *(Uses enums from CurrencyCodeSource(2897))*	@PxQteCcySrc	Please see *Table 1* and *Table 2*
2916	UnderlyingCurrencyCodeSource	NEW	string	Identifies class or source of the UnderlyingCurrency(318) value. *(Uses enums from CurrencyCodeSource(2897))*	@CcySrc	Please see *Table 1* and *Table 2*
2917	UnderlyingStrikeCurrencyCodeSource	NEW	string	Identifies class or source of the UnderlyingStrikeCurrency(941) value. *(Uses enums from CurrencyCodeSource(2897))*	@StrkCcySrc	Please see *Table 1* and *Table 2*

2918	UnderlyingUnitOfMeasureCurrencyCodeSource	NEW	string	Identifies class or source of the UnderlyingUnitOfMeasureCurrency(1718) value. *(Uses enums from CurrencyCodeSource(2897))*	@UOMCcySrc	Please see *Table 1* and *Table 2*
2919	UnderlyingPriceUnitOfMeasureCurrencyCodeSource	NEW	string	Identifies class or source of the UnderlyingPriceUnitOfMeasureCurrency(1719) value. *(Uses enums from CurrencyCodeSource(2897))*	@PxUOMCcySrc	Please see *Table 1* and *Table 2*
2920	UnderlyingPriceQuoteCurrencyCodeSource	NEW	string	Identifies class or source of the UnderlyingPriceQuoteCurrency(1526) value. *(Uses enums from CurrencyCodeSource(2897))*	@PxQteCcySrc	Please see *Table 1* and *Table 2*
2921	UnderlyingNotionalCurrencyCodeSource	NEW	string	Identifies class or source of the UnderlyingNotionalCurrency(2615) value. *(Uses enums from CurrencyCodeSource(2897))*	@NotlCcySrc	Please see *Table 1* and *Table 2*
2922	CommCurrencyCodeSource	NEW	string	Identifies class or source of the CommCurrency(479) value. *(Uses enums from CurrencyCodeSource(2897))*	@CcySrc	Please see *Table 1* and *Table 2*
2923	CommissionCurrencyCodeSource	NEW	string	Identifies class or source of the CommissionCurrency(2643) value. *(Uses enums from CurrencyCodeSource(2897))*	@CcySrc	Please see *Table 1* and *Table 2*

2924	CommissionUnitOfMeasureCurrencyCodeSource	NEW	string	Identifies class or source of the CommissionUnitOfMeasureCurrency(2645) value. (*Uses enums from CurrencyCodeSource(2897)*)	@UOMCcySrc	Please see *Table 1* and *Table 2*
2925	AllocCommissionCurrencyCodeSource	NEW	string	Identifies class or source of the AllocCommissionCurrency(2657) value. *(Uses enums from CurrencyCodeSource(2897))*	@CcySrc	Please see *Table 1* and *Table 2*
2926	AllocCommissionUnitOfMeasureCurrencyCodeSource	NEW	string	Identifies class or source of the AllocCommissionUnitOfMeasureCurrency(2659) value. *(Uses enums from CurrencyCodeSource(2897))*	@UOMCcySrc	Please see *Table 1* and *Table 2*
2927	AllocSettlCurrencyCodeSource	NEW	string	Identifies class or source of the AllocSettlCurrency(736) value. *(Uses enums from CurrencyCodeSource(2897))*	@AllocSettlCcySrc	Please see *Table 1* and *Table 2*
2928	LegAllocSettlCurrencyCodeSource	NEW	string	Identifies class or source of the LegAllocSettlCurrency(1367) value. *(Uses enums from CurrencyCodeSource(2897))*	@AllocSettlCcySrc	Please see *Table 1* and *Table 2*
2929	CollateralCurrencyCodeSource	NEW	string	Identifies class or source of the CollateralCurrency(1705) value. *(Uses enums from CurrencyCodeSource(2897))*	@CcySrc	Please see *Table 1* and *Table 2*

2930	SideCollateralCurrencyCodeSource	NEW	string	Identifies class or source of the SideCollateralCurrency(2695) value. *(Uses enums from CurrencyCodeSource(2897))*	@CcySrc	Please see *Table 1* and *Table 2*
2931	CollateralReinvestmentCurrencyCodeSource	NEW	string	Identifies class or source of the CollateralReinvestmentCurrency(2843) value. *(Uses enums from CurrencyCodeSource(2897))*	@CcySrc	Please see *Table 1* and *Table 2*
2932	SideCollateralReinvestmentCurrencyCodeSource	NEW	string	Identifies class or source of the SideCollateralReinvestmentCurrency(2866) value. *(Uses enums from CurrencyCodeSource(2897))*	@CcySrc	Please see *Table 1* and *Table 2*
2933	TradeAllocCurrencyCodeSource	NEW	string	Identifies class or source of the TradeAllocCurrency(1847) value. *(Uses enums from CurrencyCodeSource(2897))*	@CcySrc	Please see *Table 1* and *Table 2*
2934	TradingCurrencyCodeSource	NEW	string	Identifies class or source of the TradingCurrency(1245) value. *(Uses enums from CurrencyCodeSource(2897))*	@TrdCcySrc	Please see *Table 1* and *Table 2*
2935	LimitAmtCurrencyCodeSource	NEW	string	Identifies class or source of the LimitAmtCurrency(1634) value. *(Uses enums from CurrencyCodeSource(2897))*	@LmtAmtCcySrc	Please see *Table 1* and *Table 2*

2936	PosQtyUnitOfMeasureCurrencyCodeSource	NEW	string	Identifies class or source of the PosQtyUnitOfMeasureCurrency(1835) value. *(Uses enums from CurrencyCodeSource(2897))*	@UOMCcySrc	Please see *Table 1* and *Table 2*
2937	PositionCurrencyCodeSource	NEW	string	Identifies class or source of the PositionCurrency(1055) value. *(Uses enums from CurrencyCodeSource(2897))*	@CcySrc	Please see *Table 1* and *Table 2*
2938	LegPosCurrencyCodeSource	NEW	string	Identifies class or source of the LegPosCurrency(1589) value. *(Uses enums from CurrencyCodeSource(2897))*	@CcySrc	Please see *Table 1* and *Table 2*
2939	RiskLimitCurrencyCodeSource	NEW	string	Identifies class or source of the RiskLimitCurrency(1532) value. *(Uses enums from CurrencyCodeSource(2897))*	@CcySrc	Please see *Table 1* and *Table 2*
2940	EntitlementAttribCurrencyCodeSource	NEW	string	Identifies class or source of the EntitlementAttribCurrency(1781) value. *(Uses enums from CurrencyCodeSource(2897))*	@CcySrc	Please see *Table 1* and *Table 2*
2941	ComplexOptPayoutCurrencyCodeSource	NEW	string	Identifies class or source of the ComplexOptPayoutCurrency(2122) value. *(Uses enums from CurrencyCodeSource(2897))*	@OptCcySrc	Please see *Table 1* and *Table 2*

2942	ComplexEventCurrencyOneCodeSource	NEW	string	Identifies class or source of the ComplexEventCurrencyOne(2124) value. *(Uses enums from CurrencyCodeSource(2897))*	@Ccy1Src	Please see *Table 1* and *Table 2*
2943	ComplexEventCurrencyTwoCodeSource	NEW	string	Identifies class or source of the ComplexEventCurrencyTwo(2125) value. *(Uses enums from CurrencyCodeSource(2897))*	@Ccy2Src	Please see *Table 1* and *Table 2*
2944	LegComplexOptPayoutCurrencyCodeSource	NEW	string	Identifies class or source of the LegComplexOptPayoutCurrency(2226) value. *(Uses enums from CurrencyCodeSource(2897))*	@OptCcySrc	Please see *Table 1* and *Table 2*
2945	LegComplexEventCurrencyOneCodeSource	NEW	string	Identifies class or source of the LegComplexEventCurrencyOne(2233) value. *(Uses enums from CurrencyCodeSource(2897))*	@Ccy1Src	Please see *Table 1* and *Table 2*
2946	LegComplexEventCurrencyTwoCodeSource	NEW	string	Identifies class or source of the LegComplexEventCurrencyTwo(2234) value. *(Uses enums from CurrencyCodeSource(2897))*	@Ccy2Src	Please see *Table 1* and *Table 2*
2947	UnderlyingComplexOptPayoutCurrencyCodeSource	NEW	string	Identifies class or source of the UnderlyingComplexOptPayoutCurrency(2266) value. *(Uses enums from CurrencyCodeSource(2897))*	@OptCcySrc	Please see *Table 1* and *Table 2*

2948	UnderlyingComplexEventCurrencyOneCodeSource	NEW	string	Identifies class or source of the UnderlyingComplexEventCurrencyOne(2268) value. *(Uses enums from CurrencyCodeSource(2897))*	@Ccy1Src	Please see *Table 1* and *Table 2*
2949	UnderlyingComplexEventCurrencyTwoCodeSource	NEW	string	Identifies class or source of the UnderlyingComplexEventCurrencyTwo(2269) value. *(Uses enums from CurrencyCodeSource(2897))*	@Ccy2Src	Please see *Table 1* and *Table 2*
2950	BenchmarkCurveCurrencyCodeSource	NEW	string	Identifies class or source of the BenchmarkCurveCurrency(220) value. *(Uses enums from CurrencyCodeSource(2897))*	@CcySrc	Please see *Table 1* and *Table 2*
2951	LegBenchmarkCurveCurrencyCodeSource	NEW	string	Identifies class or source of the LegBenchmarkCurveCurrency(676) value. *(Uses enums from CurrencyCodeSource(2897))*	@CcySrc	Please see *Table 1* and *Table 2*
2952	AgreementCurrencyCodeSource	NEW	string	Identifies class or source of the AgreementCurrency(918) value. *(Uses enums from CurrencyCodeSource(2897))*	@AgmtCcySrc	Please see *Table 1* and *Table 2*
2953	LegAgreementCurrencyCodeSource	NEW	string	Identifies class or source of the LegAgreementCurrency(2495) value. *(Uses enums from CurrencyCodeSource(2897))*	@AgmtCcySrc	Please see *Table 1* and *Table 2*

2954	FundingSourceCurrencyCodeSource	NEW	string	Identifies class or source of the FundingSourceCurrency(2847) value. *(Uses enums from CurrencyCodeSource(2897))*	@CcySrc	Please see *Table 1* and *Table 2*
2955	PayCollectCurrencyCodeSource	NEW	string	Identifies class or source of the PayCollectCurrency(1709) value. *(Uses enums from CurrencyCodeSource(2897))*	@CcySrc	Please see *Table 1* and *Table 2*
2956	PostTradePaymentCurrencyCodeSource	NEW	string	Identifies class or source of the PostTradePaymentCurrency(2818) value. *(Uses enums from CurrencyCodeSource(2897))*	@CcySrc	Please see *Table 1* and *Table 2*
2957	SymbolPositionNumber	NEW	int	Reference to the first or second currency or digital asset in Symbol(55) for FX-style trading. Conditionally required when one or both symbols in Symbol(55) represent a digital asset.	@SymPosNum	Add to SecAltIDGrp component
2958	LegSymbolPositionNumber	NEW	int	Reference to the first or second currency or digital asset in LegSymbol(600) for FX-style trading. Conditionally required when one or both symbols in LegSymbol(600) represent a digital asset.	@SymPosNum	Add to LegSecAltIDGrp component
2959	UnderlyingSymbolPositionNumber	NEW	int	Reference to the first or second currency or digital asset in UnderlyingSymbol(311) for FX-style trading. Conditionally required when one or both symbols in UnderlyingSymbol(311) represent a digital asset.	@SymPosNum	Add to UndSecAltIDGrp component

2960	SettlPriceUnitOfMeasureCurrencyCodeSource	NEW	String	Identifies the class or source of the SettlPriceUnitOfMeasureCurrency(1887) value. (Uses enums from CurrencyCodeSource(2897))	@SetPxUOMCcySrc	Add to PositionReport(35=AP) message
15	Currency	CHANGE	Currency	*Correct/append description and add elaboration:* Identifies currency used for price or quantity fields, depending on the asset class being traded. CurrencyCodeSource(2897) may be used to disambiguate the code source scheme used, and ISO 4217 is the default scheme if absent. Absence of this field is interpreted as the default currency for the security as defined by the respective reference data. It is recommended that systems provide the currency value whenever possible. ~~See "Appendix 6-A: Valid Currency Codes" for information on obtaining valid values.~~ [Elaboration: For securities trading and digital assets traded securities-style, identifies the currency used to denote the price. Absence of this field is interpreted as the default for the security. For Foreign Exchange (FX) and digital assets traded FX-style, identifies the dealt currency used to denominate the quantity related field(s).]	@Ccy	

22	SecurityIDSource	CHANGE	String	Identifies class or source of the SecurityID(48) value. *Change enumeration:* *6 = ISO Currency Code (ISO 4217)* *Add enumeration:* Y = Digital Token Identifier (ISO 24165)	@Src	
167	SecurityType	CHANGE	String	Indicates type of security. *Add enumeration to **Other**:* DIGITAL = Digital Asset [Elaboration: Asset that exists only in digital form or which is the digital representation of another asset (Source: ISO 24165 - Terms and Definitions).]	@SecTyp	
220	BenchmarkCurveCurrency	CHANGE	Currency	*Correct and append description:* Specifies currency used for benchmark curve. ~~See "Appendix 6-A; Valid Currency Codes" for information in obtaining valid values.~~ BenchmarkCurveCurrencyCodeSource(2950) may be used to disambiguate the code source scheme used, and ISO 4217 is the default scheme if absent.	@Ccy	
318	UnderlyingCurrency	CHANGE	Currency	*Correct description:* Underlying security's ~~C~~currency. ~~See Currency (5) field for description and valid values.~~	@Ccy	

479	CommCurrency	CHANGE	Currency	*Correct and append description:* Specifies currency to be used for Commission(12) if the commission currency is different from the deal currency. ~~See "Appendix 6-A; Valid Currency Codes".~~ CommCurrencyCodeSource(2922) may be used to disambiguate the code source scheme used, and ISO 4217 is the default scheme if absent.	@Ccy	
1634	LimitAmtCurrency	CHANGE	Currency	*Correct description:* Indicates the currency that the limit amount is specified in. ~~See Currency(15) for additional description and valid values.~~	@Ccy	
2122	ComplexOptPayoutCurrency	CHANGE	Currency	*Correct and append description:* Specifies the currency of the payout amount. ~~Uses ISO 4217 currency codes.~~ ComplexOptPayoutCurrencyCodeSource(2941) may be used to disambiguate the code source scheme used, and ISO 4217 is the default scheme if absent.	@Ccy	
2124	ComplexEventCurrencyOne	CHANGE	Currency	*Correct and append description:* Specifies the first or only reference currency of the trade. ~~Uses ISO 4217 currency codes.~~ ComplexEventCurrencyOneCodeSource(2942) may be used to disambiguate the code source scheme used, and ISO 4217 is the default scheme if absent.	@Ccy	

2125	ComplexEventCurrencyTwo	CHANGE	Currency	*Correct and append description:* Specifies the second reference currency of the trade. ~~Uses ISO 4217 currency codes.~~ ComplexEventCurrencyTwoCodeSource(2943) may be used to disambiguate the code source scheme used, and ISO 4217 is the default scheme if absent.	@Ccy	
2226	LegComplexOptPayoutCurrency	CHANGE	Currency	*Correct and append description:* Specifies the currency of the payout amount. ~~Uses ISO 4217 currency codes.~~ LegComplexOptPayoutCurrencyCodeSource(2944) may be used to disambiguate the code source scheme used, and ISO 4217 is the default scheme if absent.	@Ccy	
2233	LegComplexEventCurrencyOne	CHANGE	Currency	*Correct and append description:* Specifies the first or only reference currency of the trade. ~~Uses ISO 4217 currency codes.~~ LegComplexEventCurrencyOneCodeSource(2945) may be used to disambiguate the code source scheme used, and ISO 4217 is the default scheme if absent.	@Ccy	
2234	LegComplexEventCurrencyTwo	CHANGE	Currency	*Correct and append description:* Specifies the second reference currency of the trade. ~~Uses ISO 4217 currency codes.~~ LegComplexEventCurrencyTwoCodeSource(2946) may be used to disambiguate the code source scheme used, and ISO 4217 is the default scheme if absent.	@Ccy	

2266	UnderlyingComplexOptPayoutCurrency	CHANGE	Currency	*Correct and append description:* Specifies the currency of the payout amount. ~~Uses ISO 4217 currency codes.~~ UnderlyingComplexOptPayoutCurrencyCodeSource(2947) may be used to disambiguate the code source scheme used, and ISO 4217 is the default scheme if absent.	@Ccy	
2268	UnderlyingComplexEventCurrencyOne	CHANGE	Currency	*Correct and append description:* Specifies the first or only reference currency of the trade. ~~Uses ISO 4217 currency codes.~~ UnderlyingComplexEventCurrencyOneCodeSource(2948) may be used to disambiguate the code source scheme used, and ISO 4217 is the default scheme if absent.	@Ccy	
2269	UnderlyingComplexEventCurrencyTwo	CHANGE	Currency	*Correct and append description:* Specifies the second reference currency of the trade. ~~Uses ISO 4217 currency codes.~~ UnderlyingComplexEventCurrencyTwoCodeSource(2949) may be used to disambiguate the code source scheme used, and ISO 4217 is the default scheme if absent.	@Ccy	

2615	UnderlyingNotionalCurrency	CHANGE	Currency	*Correct and append description:* Specifies the currency denomination of the notional value. ~~Uses ISO 4217 currency codes.~~ UnderlyingNotionalCurrencyCodeSource(2921) may be used to disambiguate the code source scheme used, and ISO 4217 is the default scheme if absent.	@Ccy	
2643	CommissionCurrency	CHANGE	Currency	*Correct and append description:* Specifies the currency denomination of the commission amount if different from the trade's currency. ~~Uses ISO 4217 currency codes.~~ CommissionCurrencyCodeSource(2923) may be used to disambiguate the code source scheme used, and ISO 4217 is the default scheme if absent.	@Ccy	
2657	AllocCommissionCurrency	CHANGE	Currency	*Correct and append description:* Specifies the currency denomination of the commission amount if different from the trade's currency. ~~Uses ISO 4217 currency codes.~~ AllocCommissionCurrencyCodeSource(2925) may be used to disambiguate the code source scheme used, and ISO 4217 is the default scheme if absent.	@Ccy	

2695	SideCollateralCurrency	CHANGE	Currency	*Correct and append description:* Specifies the currency of the collateral; optional, defaults to settlement currency if not specified. ~~Uses ISO 4217 currency codes.~~ SideCollateralCurrencyCodeSource(2930) may be used to disambiguate the code source scheme used, and ISO 4217 is the default scheme if absent.	@Ccy	
2818	PostTradePaymentCurrency	CHANGE	Currency	*Correct and append description:* Specifies the currency in which PostTradePaymentAmount(2817) is denominated. ~~Uses ISO 4217 currency codes.~~ PostTradePaymentCurrencyCodeSource(2956) may be used to disambiguate the code source scheme used, and ISO 4217 is the default scheme if absent.	@Ccy	
2843	CollateralReinvestmentCurrency	CHANGE	Currency	*Correct and append description:* The currency denomination of the re-invested cash amount. ~~Uses ISO 4217 currency codes.~~ CollateralReinvestmentCurrencyCodeSource(2931) may be used to disambiguate the code source scheme used, and ISO 4217 is the default scheme if absent.	@Ccy	

2847	FundingSourceCurrency	CHANGE	Currency	*Correct and append description:* Currency denomination of the market value of the funding source. ~~Uses ISO 4217 currency codes.~~ FundingSourceCurrencyCodeSource(2954) may be used to disambiguate the code source scheme used, and ISO 4217 is the default scheme if absent.	@Ccy	
2866	SideCollateralReinvestmentCurrency	CHANGE	Currency	*Correct and append description:* The currency denomination of the re-invested cash amount. ~~Uses ISO 4217 currency codes.~~ SideCollateralReinvestmentCurrencyCodeSource(2932) may be used to disambiguate the code source scheme used, and ISO 4217 is the default scheme if absent.	@Ccy	

Appendix B - Glossary Entries

Term	Definition	Field where used

Appendix C - Abbreviations

Term	Proposed Abbreviation	Proposed Messages, Components, Fields where used

Appendix D - Usage Examples

Fiat ccy vs. fiat ccy, "FX-style"

	Buying 1M EUR	Selling 1.4M USD
ORDERS & EXECUTIONS		
Symbol(55)	EUR/USD	EUR/USD
Side(54)	Buy	Sell
Direction - PriceType(423)	20 (Normal - CCY1 * rate)	21 (Inverse - CCY1 / Rate)
ORDERS		
Rate – Price(44)	1.4	1.4
Dealing Qty – OrderQty(38)	1000000	1400000
Dealing Ccy - Currency(15)	EUR	USD
Optional unless settling in a third currency:		
SettlCurrency(120)	USD	EUR
EXECUTIONS		
Rate – LastPx(31) and AvgPx(6)	1.4	1.4
Dealt Qty – LastQty(32)	1000000	1400000
Dealt Ccy - Currency(15)	EUR	USD
CalculatedCcyLastQty(1056)	1400000	1000000
Optional unless settling in a third currency:		
SettlCurrency(120)	USD	EUR
SettlCurrAmt(119)	1400000	1000000

Digital, "FX-Style"

	Buying 2.5 Bitcoin in USD	Buying $100,000 of Bitcoin *(or selling $100,000 for Bitcoin)*	Buying 10 Ether in Bitcoin
ORDERS & EXECUTIONS			
Symbol (55)	BTC/USD	BTC/USD	ETH/BTC
NoSecurityAltID(454)	2	2	2
SecurityAltID(455)	4H95J0R2X (Bitcoin's DTI)	4H95J0R2X (Bitcoin's DTI)	X9J9K8725 (Ether's DTI)
SecurityAltIDSource(456)	*Y=Digital Token Identifier*	*Y=Digital Token Identifier*	*Y=Digital Token Identifier*
SymbolPositionNumber(2957)	*1*	*1*	*1*
SecurityAltID(455)	USD	USD	4H95J0R2X (Bitcoin's DTI)
SecurityAltIDSource(456)	6	6	New value to represent DTI
SymbolPositionNumber(2957)	*2*	*2*	*2*
Side(54)	Buy	Sell (because we are selling USD)	Buy
Direction - PriceType(423)	20 (Normal - CCY1 * rate)	21 (Inverse - CCY1 / rate)	20 (Normal - CCY1 * rate)
ORDERS			
Rate – Price(44)	40,000 (price of 1 BTC in USD)	40,000 (price of 1 BTC in USD)	0.075 (price of 1 ETH in BTC)
Dealing Qty – OrderQty(38)	2.5 (in Bitcoin)	100,000 (in USD)	10 (in Ether)
Dealing Ccy - Currency(15)	4H95J0R2X (Bitcoin)	USD	X9J9K8725 (Ether)
CurrencyCodeSource(2897)	*Y=Digital Token Identifier*	*6=ISO Currency Code*	*Y=Digital Token Identifier*
Optional unless settling in a third currency:			
SettlCurrency(120)	USD	4H95J0R2X (Bitcoin's DTI)	4H95J0R2X (Bitcoin's DTI)

	Buying 2.5 Bitcoin in USD	**Buying $100,000 of Bitcoin** *(or selling $100,000 for Bitcoin)*	**Buying 10 Ether in Bitcoin**
SettlCurrencyCodeSource(2899)	*6=ISO Currency Code*	*Y=Digital Token Identifier*	*Y=Digital Token Identifier*
EXECUTIONS			
Rate – LastPx(31) and AvgPx(6)	40,000 (price of 1 BTC in USD)	40,000 (price of 1 BTC in USD)	0.075 (price of 1 ETH in BTC)
Dealt Qty – LastQty(32)	2.5 (in Bitcoin)	100,000 (in USD)	10 (in Ether)
Dealt Ccy - Currency(15)	4H95J0R2X (Bitcoin)	USD	X9J9K8725 (Ether)
CurrencyCodeSource(Y)	*Y=Digital Token Identifier*	*6=ISO Currency Code*	*Y=Digital Token Identifier*
CalculatedCcyLastQty(1056)	100,000 (in USD)	2.5 (in Bitcoin)	0.75 (in Bitcoin)
Optional unless settling in a third currency:			
SettlCurrency(120)	USD	4H95J0R2X (Bitcoin)	4H95J0R2X (Bitcoin's DTI)
SettlCurrencyCodeSource(2899)	*6=ISO Currency Code*	*Y=Digital Token Identifier*	*Y=Digital Token Identifier*
SettlCurrAmt(119)	100,000 (in USD)	2.5 (in Bitcoin)	0.75 (in Bitcoin)

Digital, "Securities-Style"

	Buying 250 MSFT in USD	Buying 5 'crypto-Apple shares' in USD	Buying $100,000 of Bitcoin	Buying 10 Ether in Bitcoin
ORDERS AND EXECUTIONS				
Symbol(55)	MSFT	CAAP	BTC	ETH
SecurityID(48)	594918104	A1B2C3D4E	4H95J0R2X (Bitcoin's DTI)	X9J9K8725 (Ether's DTI)
SecurityIDSource(22)	*1 (CUSIP)*	*Y=Digital Token Identifier*	*Y=Digital Token Identifier*	*Y=Digital Token Identifier*
Side(54)	Buy	Buy	Buy	Buy
ORDERS				
Price – Price(44)	311	500	40,000	0.075 (price of 1 ETH in BTC)
Quantity – OrderQty(38)	250	5		10
Quantity – CashOrderQty(152)			100,000	
Dealing Ccy - Currency(15)	USD	USD	USD	4H95J0R2X (Bitcoin)
CurrencyCodeSource(2897)	*6=ISO Currency Code*	*6=ISO Currency Code*	*6=ISO Currency Code*	*Y=Digital Token Identifier*
Optional unless settling in a third currency:				
SettlCurrency(120)	USD	USD	4H95J0R2X (Bitcoin)	4H95J0R2X (Bitcoin)
SettlCurrencyCodeSource(2899)	*6=ISO Currency Code*	*6=ISO Currency Code*	*Y=Digital Token Identifier*	*Y=Digital Token Identifier*
EXECUTIONS				
Price – LastPx(31) and AvgPx(6)	311	500	40,000	0.075 (price of 1 ETH in BTC)
Quantity – LastQty(32)	250	5	2.5	10
Dealt Ccy - Currency(15)	USD	USD	USD	4H95J0R2X (Bitcoin)
CurrencyCodeSource(2897)	*6=ISO Currency Code*	*6=ISO Currency Code*	*6=ISO Currency Code*	*Y=Digital Token Identifier*

	Buying 250 MSFT in USD	Buying 5 'crypto-Apple shares' in USD	Buying $100,000 of Bitcoin	Buying 10 Ether in Bitcoin
Optional unless settling in a third currency:				
SettlCurrency(120)	USD	USD	4H95J0R2X (Bitcoin)	4H95J0R2X (Bitcoin)
SettlCurrencyCodeSource(2899)	*6=ISO Currency Code*	*6=ISO Currency Code*	*Y=Digital Token Identifier*	*Y=Digital Token Identifier*
SettlCurrAmt(119) – on ERs	77750	2500	2.5 (in Bitcoin)	0.75

Appendix E – Disposition of Public Comments

Public comment #1 from Li Zhu May 26, 2022

Another proposal: Section 2.2.1 describes how to use Currency(15) and OrderQty(38) fields for dealing currency of quantity fields. But this explanation has only touched half of the currency/digital assets involved in a trading. It is advised to provide a complete example for this description.

Response from jkaye May 29, 2022:
Hi - many thanks for your comments. We are in the process of writing a 'Recommended Practices' document which will include a lot more detail on how to use these fields under different trading scenarios. If you're a member of our Digital Assets Working Group, we'll be reviewing this in meetings of that group. There will also be a public review.
Jim.

DWAG Resolution:
July 7, 2022 WG call: Group agrees that this would be covered by the Recommended Practices in more detail. Appendix D of this Gap Analysis also includes some examples regarding use of OrderQty(38) and Currency(15) fields along with the new fields proposed.

Public comment #2 from Li Zhu May 26, 2022

Section 2.4 Wallet identifier.

It is not a good term to use WALLET here. In the most popular permissionless blockchains, such as Bitcoin and Ethereum, one uses "address" for reception purpose. And in the crypto communities, wallet is used for some software/hardware which manages the privates keys and signs transactions. But there can be a many to many relationships between wallet and address(which is useally derived from private keys). That is , a wallet will manage many private keys(and thus, many address), and a private key(and thus address） can be imported into different wallets and controlled by any one of the wallets.

Perhaps the most popular wallet is MetaMask, as a chrome plugin. Hardware wallets, such as LedgerNanoS are also widely used.

By the way, ISO has published ISO 22739 （Blockchain and distributed ledger technologies - vocabulary ）, defined common terms used in blockchain and DLT community. It is advised to reuse the terms defined in an ISO standard, instead of inventing ones own.

In ISO 22739(sorry I can not upload the documents due to copyright), we can find the following 3 related terms:

entry 3.24
DLT account
distributed ledger technology account

representation of an entity (3.34) participating in a transaction (3.77)

entry 3.25
DLT address
distributed ledger technology address
value that identifies a DLT account (3.24) participating in a transaction (3.77)

entry 3.84
wallet
application used to generate, manage, store or use private (3.62) and public keys (3.65)

I believe it is better to use DLT address instead of wallet in our standard, both to keep aligned with established ISO standard and crypto community daily practice.

Response from jkaye May 29, 2022:
Hi,
Thanks for your comments on use of wallet identifier. I support removing this reference for this gap analysis and analysing further for a future gap analysis to cover post-trade workflows.
Your comments on ISO 22739 are also helpful - I wasn't aware of that. It's not clear to me whether the concept of 'Account' in FIX (tag 1) maps to a DTL account or a DLT address, or even quite what the difference between the two is. We may need to dip into our Swiss franc money box to find out…
Jim.

DWAG Resolution:
July 7, 2022 WG call: Agreed to remove the proposed request to add the concept of "wallet" to PartySubIDType. This concept will be revisited at a later time when further requirements are understood. Section 2.4 has been removed as well.

Public comment #3 from jkaye June 1, 2022

I have reviewed the Gap Analysis and have no concerns other than to restate my response to a comment raised earlier on this forum regarding Wallet Identifier. My recommendation is that the paragraph referring to Wallet Identifier be removed and that this be addressed as part of the post-trade analysis scheduled to be undertaken by the Digital Assets Working Group in the second half of this year.
Jim.

DWAG Resolution:
July 7, 2022 WG call: no action needed

